Semiconductors

UM10113

User manual for the PNX2015 family

Rev. 01 — 6 May 2005 User manual

Document information

Info	Content
Keywords	TV810 platform, PNX8550, ATV, ATSC, Jaguar, HD subsystem, AVIP1, AVIP2, Columbus, TV microcontroller.
Abstract	This user manual provides a functional overview of PNX2015, together with detailed descriptions of major functional blocks.

PHILIPS

Revision history

Rev	Date	Description
01	20050506	Objective data

Contact information

For additional information, please visit: **http://www.semiconductors.philips.com**

For sales office addresses, please send an email to: **sales.addresses@www.semiconductors.philips.com**

1. Introduction

The Functional Overview section of this User Manual contains three IC level overviews:

- Interfaces

  IC interfaces and configuration of AVIP 1 and 2 in the PNX2015
- Top level, structure

  IC wide signal and power requirement
- Subsystems

  Summary of IP block functions

The main body of the manual describes the IP blocks that the PNX2015 contains:

- **AVIP**

  Dlink audio video input interface IP block. This section is applicable to both AVIP 1 and AVIP 2
- **Columbus**

  3D comb filter and spatial/temporal noise reduction block description
- **HD subsystem**

  This module subdivides to further IP block descriptions
- **LVDS transmitter**

  30 bit input (from PNX8550 QVCP) IP block description
- **TV microcontroller**

  Standby micro (containing an 80C51 CPU core) description

The end sections include Limitations, Glossary and Contents.

This user manual provides:

This User Manual contains a description of the PNX2015 components and its internal IP blocks. The Internal IP bock descriptions contain register listings and show how the IP blocks perform their function. There are IC inter connectivity descriptions to the PNX8550 and Video coprocessor. There are explanations of the video audio and programming interfaces.

This user manual does not provide:

Since the PNX2015 is a component within a system this document is not a Use Case based programming guide. The register descriptions do not contain recommended programming values. The PNX2015 has a memory bases architecture which provides for flexible configuration to meet Use Case requirements. There are therefore no configuration descriptions for the PNX2015.

2. Functional overview

2.1 Introduction

The PNX2015 is a prime component for the mid-to high-end market segment of the ATSC, DVB and Flat Panel markets. Through the addition of various hardware and software components, systems can meet high-end requirements of connectivity and picture performance.

The PNX2015 accompanies the PNX8550 and provides the following functionality, in addition to that provided by PNX8550:

- Dual analog audio/video decoding.
- Full audio processing.
- Analog video improvement, e.g. 3D comb filter, noise reduction and noise measurement (Columbus).
- Second channel HD decode with memory based scaler.
- Interface to additional external video improvement IC’s.
- RGB via standard LVDS interface for LCD panels.
- Embedded TV microcontroller.

An example system diagram is shown in Figure 1

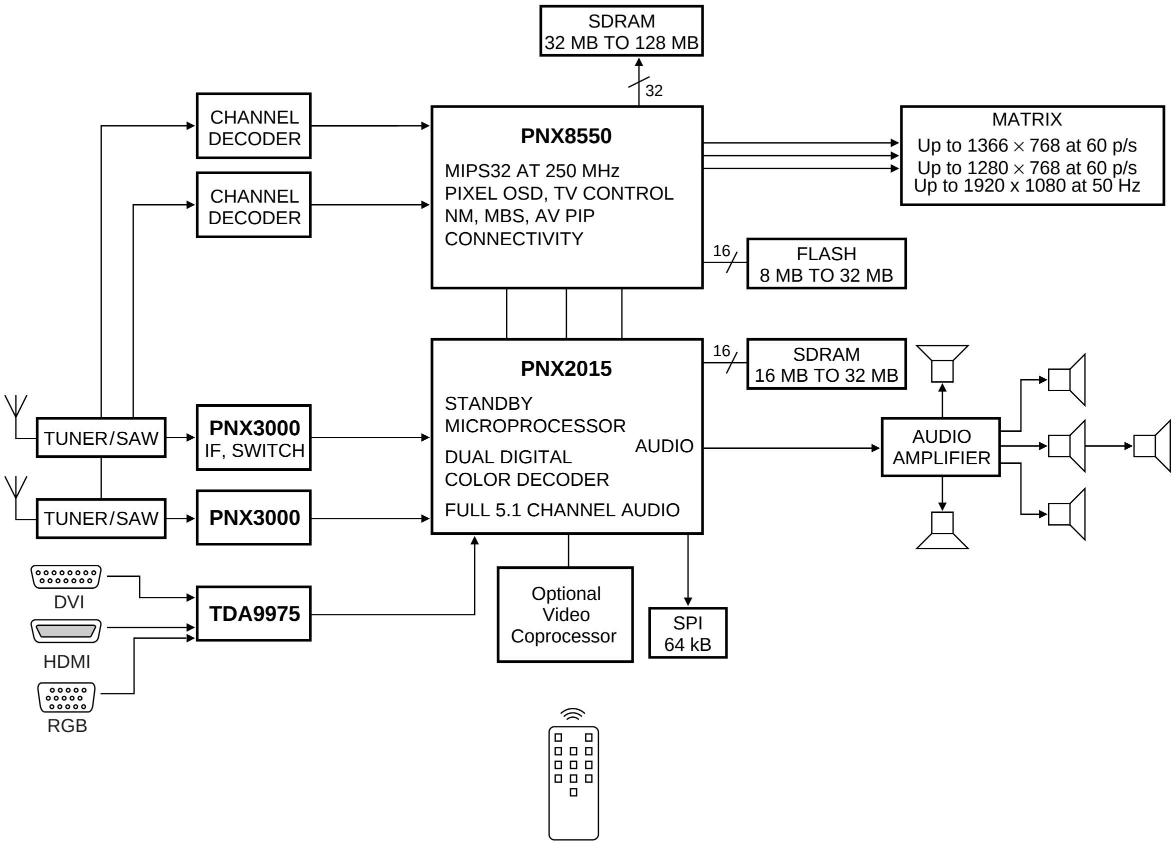


Fig 1. System diagram

In terms of functional blocks the PNX2015 can be described as follows:

- Audio Video Input Processor 1 (AVIP1).
- Audio Video Input Processor 2 (AVIP2).
- 3D comb filter with spatial and temporal noise reduction (Columbus).
- High Definition MPEG decoder (HD Subsystem).
- Standby microcontroller for low-power control.

The PNX2015 block diagram is shown in Figure 2.

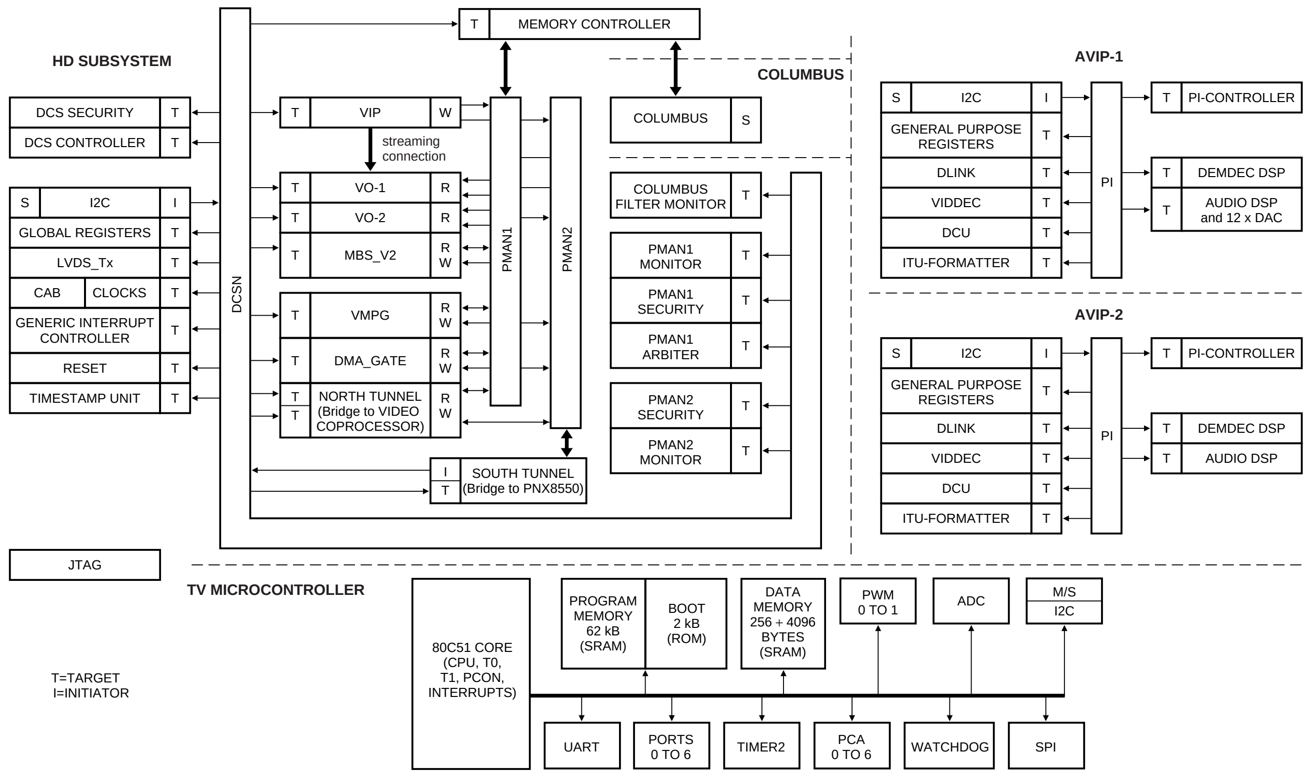


Fig 2. PNX2015 block diagram

2.2 Feature summary

- Detection of PAL, NTSC or SECAM analog sources and 1_{fH} and 2_{fH} component video input sources. (1_{fH} = 13.3 MHz, 2_{fH} = 27 MHz).
- Support for 1_{fH} and 2_{fH} video sources; progressive and interlaced.
- 3D comb filters for PAL/NTSC standards.
- Spatial and temporal noise reduction for PAL, NTSC and SECAM analog sources.
- Decoding for global VBI Standards (WST, WSS, VPS, CC, VITC and Gemstar™).
- Global multi-standard audio demodulation and decoding (FM, AM, NICAM, BTSC, A2).
- Dolby® ProLogic® II multi-channel audio decoding and post-processing.
- Advanced, fully programmable audio post-processing functions (VDS, VDD, bass management and graphic equalizer).
- RGB via standard LVDS interface for National™ (18/24-bit) and THine™ (18/24/30-bit).
- High and Standard Definition digital video input interface.
- Hardware decoding of two SD streams (MPEG2 MP@ML) or one HD (MPEG2 MP@HL).
- Embedded TV Microcontroller for TV control and system power management.
- Two independent tunnel interfaces for PNX8550 and video coprocessor.

- Memory interface for DDR/GDDR SDRAM and SPI™ for external flash.
- Three digital video outputs 8/10-bit YUV 4:2:2.

Three versions of the PNX2015 are available, the features are shown in Table 1.

Table 1: PNX2015 versions

Type number	Features				
	3D NTSC	3D PAL	DPL2	BBE™	dbx™
PNX2015E/M1C02		3	3		
PNX2015E/M1C03	3		3		3

2.3 Interfaces

2.3.1 Video

Figure 3 shows the main video interfaces of PNX2015, and how the video signals are connected between subsystems.

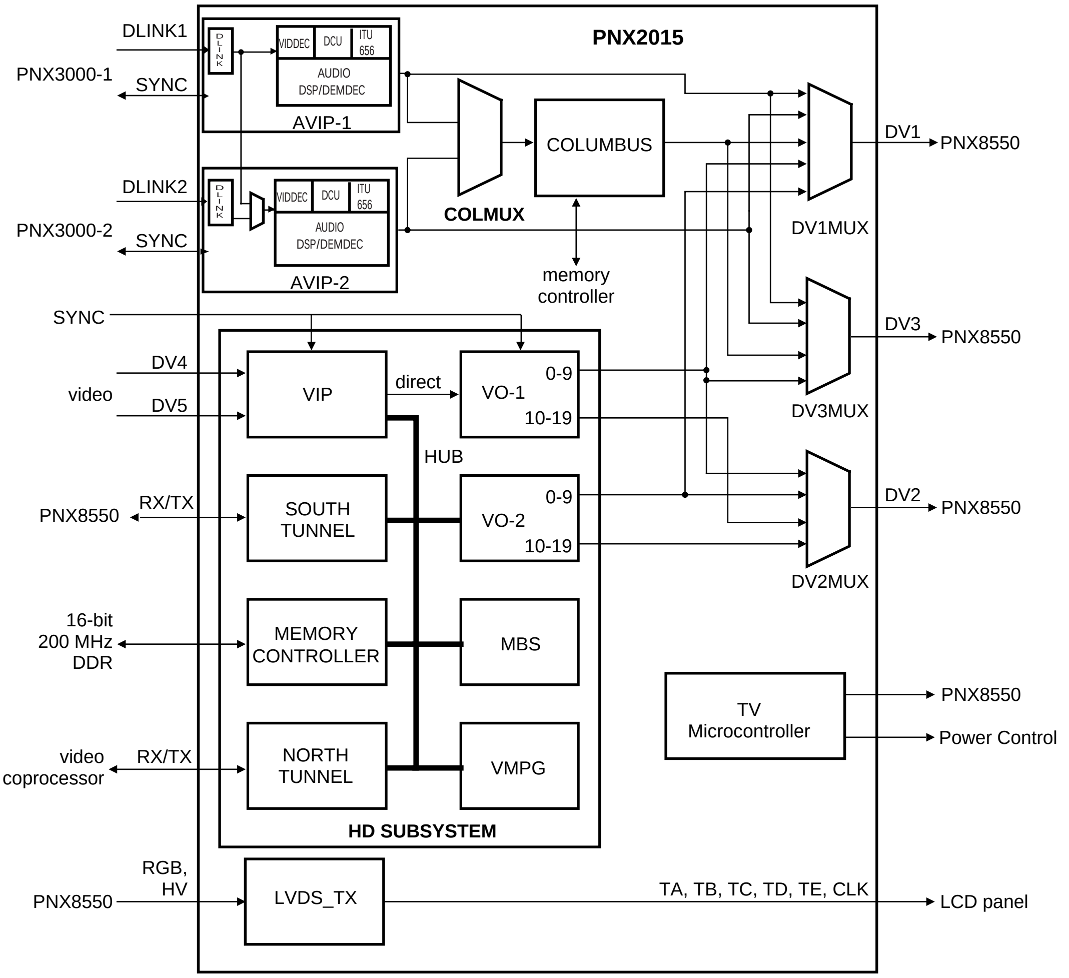


Fig 3. PNX2015 subsystems

Table 2 to Table 4 show the outputs available on the three Digital Video output ports. Each DV port can independently select the required (8/10-bit YUV) output source. There is no link between the DV outputs.

The DV1 and DV2 ports can be used in combination to output a 16/20-bit semi-planar signal. In this mode DV1 outputs the Y (lower 10-bits) component, and DV2 the UV (upper 10-bits) component.

Table 2: Output port DV1 - signal combinations

Select	DV1
0 Default	AVIP1
1	AVIP2
2	Columbus
3	VO-1 (lower) 656 mode = YUV; semi-planar = Y
4	VO-2 (lower) 656 mode = YUV; semi-planar = Y

Table 3: Output port DV2 - signal combinations

Select	DV2
0	VO-1 656 mode = YUV; semi-planar = N/A
1 Default	VO-2 656 mode = YUV; semi-planar = N/A
2	VO-1 (upper) 656 mode = N/A; semi-planar = UV
3	VO-2 (upper) 656 mode = N/A; semi-planar = UV

Table 4: Output port DV3 - signal combinations

Select	DV3
0	AVIP1
1 Default	AVIP2
2	Columbus
3	VO-1 656 mode = YUV; semi-planar = N/A

The DV Input ports can be used as a standard 656 interface 8/10-bit with DV4 only, or as a 16/20-bit semi-planar mode with the addition of DV5. One clock is provided for DV4 and DV5, so they may not be used independently.

Table 5: DV input port signal combinations (DV4, DV5)

Input port	656 mode	Semi-planar mode
DV4	YUV	Y
DV5	-	UV

The use of video data bus bits on DV1 to DV5, and their connections for the SD and HD capture modes are shown in Table 7 and Table 6. DV1, DV2 and DV3 are outputs, DV4 and DV5 are inputs.

In all modes, the MSB of the data words is always placed on pin [9], i.e. in any 8-bit mode the valid data occupies the upper 8 most significant bit positions of the 10-bit input.

2.3.1.1 Multi-mode video output

Table 6: Multi-mode video output connections (DV1, DV2, DV3)

Signal	Port	656 mode		Semi-planar	
		10-bit	8-bit	20-bit	16-bit
DV1_DATA [9:2]	DV1	YUV [9:0]	YUV [7:0]	Y [9:0]	Y [9:2]
DV1_DATA [1:0]			not used		not used
DV2_DATA [9:2]	DV2	YUV [9:0]	YUV [7:0]	UV [9:0]	UV [9:2]
DV2_DATA [1:0]			not used		not used
DV3_DATA [9:2]	DV3	YUV [9:0]	YUV [7:0]	not used	not used
DV3_DATA [1:0]			not used		

2.3.1.2 Multi-mode video input

Table 7: Multi-mode video input connections (DV4, DV5)

Signal	Port	656 mode		Semi-planar	
		10-bit	8-bit	20-bit	16-bit
DV4_DATA [9:2]	DV4	YUV [9:0]	YUV [7:0]	Y [9:0]	Y [9:2]
DV4_DATA [1:0]			not used		not used
DV5_DATA [9:2]	DV5	not used	YUV [7:0]	UV [9:0]	UV [9:2]
DV5_DATA [1:0]			not used		not used

2.3.2 Audio

The audio interfaces allow for up to six I^2S[9] digital inputs, and six I^2S digital outputs. All I^2S interfaces share a common clock system and Word Select signal. The I^2S clock system can be configured as Master or Slave.

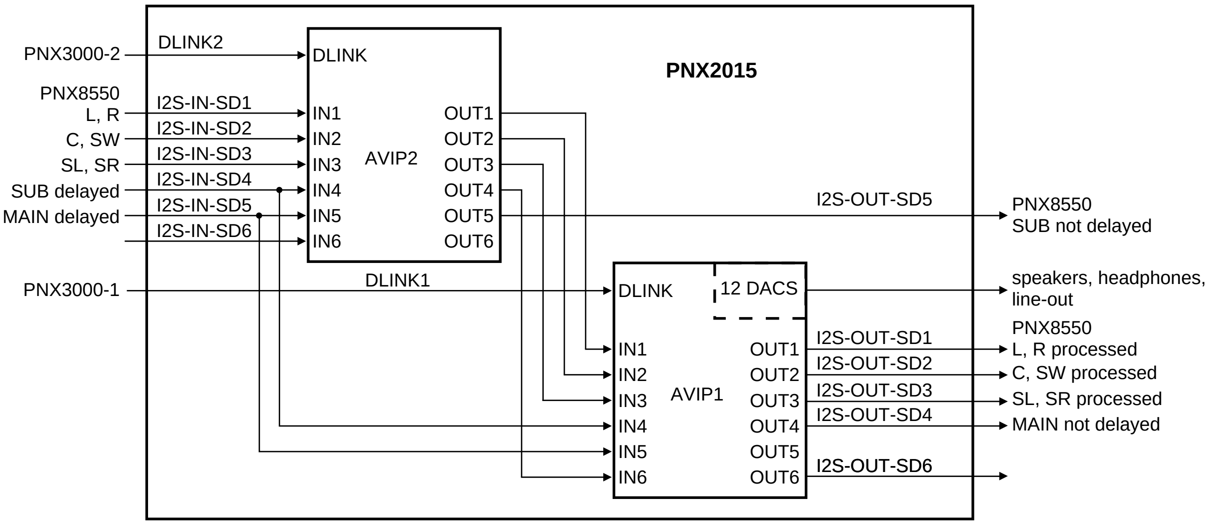


Fig 4. Audio interface

Figure 4 shows the interconnections between the two AVIP instances and the connections to the I^{2}S device pins. The legend shows the suggested signal content when used with PNX8550, although the assignment of signals to the I^{2}S channels and DAC outputs is fully flexible. Audio DACs are only present on AVIP1, otherwise the two AVIP audio sections are identical.

In single MPIF AvPip mode the suggested routing of the sub channel is to I2S_OUT_SD4.

2.3.3 Audio/video flow

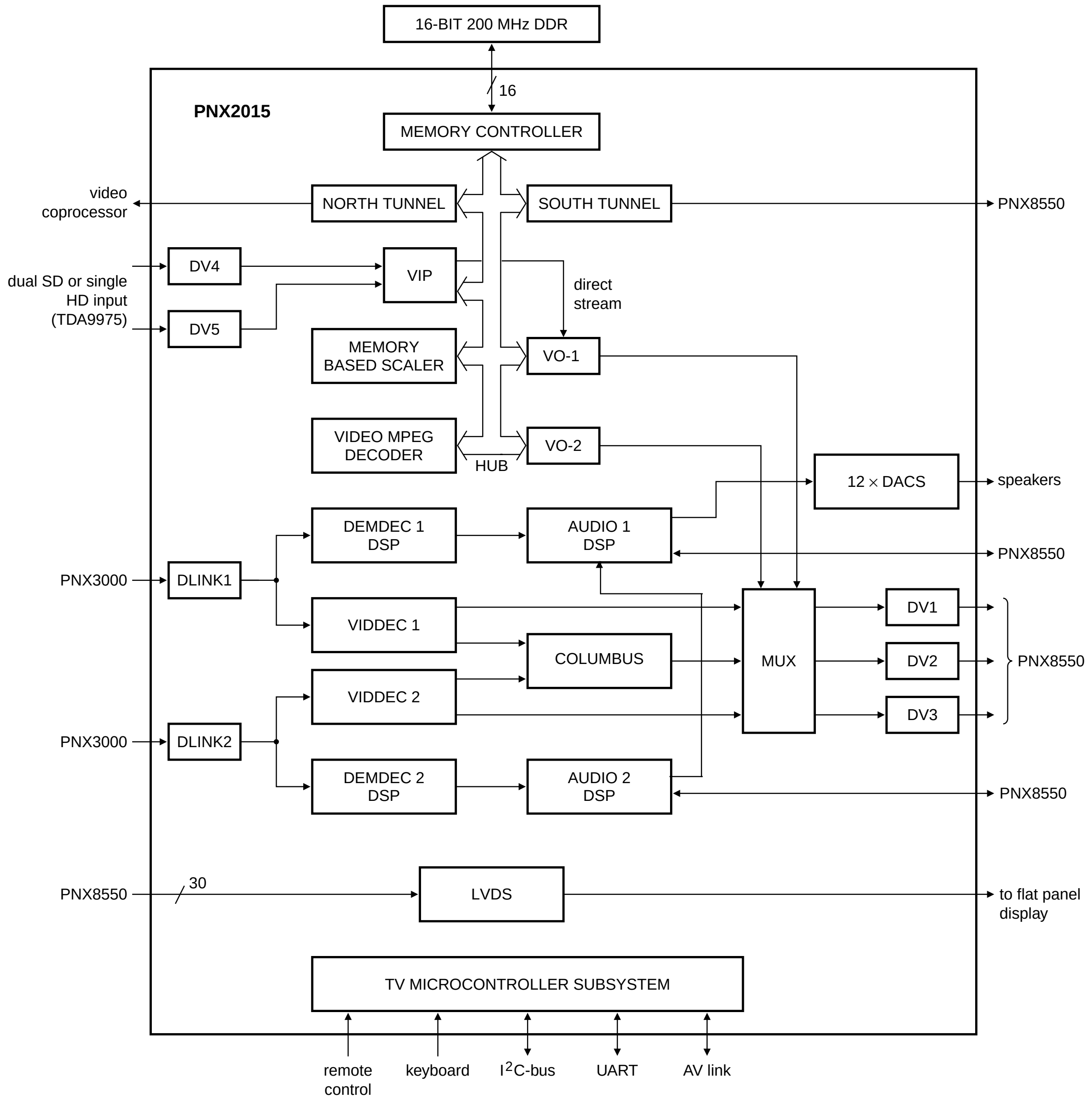


Fig 5. Audio/video flow diagram

2.3.4 Control

The AVIP and Columbus subsystems are controlled via I^{2}C interfaces as shown in Figure 6.

The main method of control for the HD subsystem is via MMIO commands across the south tunnel interface, once the tunnel is initialized using the I^{2}C.

The TV Microcontroller has a master/slave I^{2}C interface, which is mainly used by the TV Microcontroller to control the devices in the system.

The I^{2}C module implements a master/slave I^{2}C interface with integrated shift register timing generation and slave address recognition. It is compliant to the I^{2}C specification Ref. 1. The I^{2}C standard mode (100 KHz SCL) and fast mode (400 KHz SCL) are supported. Extended 10-bit addressing is not supported.

Table 8: Control interface

Block	Sub-address	Sub-address width	Data width
AVIP1	8A(W), 8B(R)	32-bit	32-bit
AVIP2	88(W), 89(R)	32-bit	32-bit
Columbus	B0(W), B1(R)	8-bit	8-bit
HD subsystem	A0(W), A1(R)	32-bit	32-bit
TV Microcontroller	via SW config	8-bit	8-bit

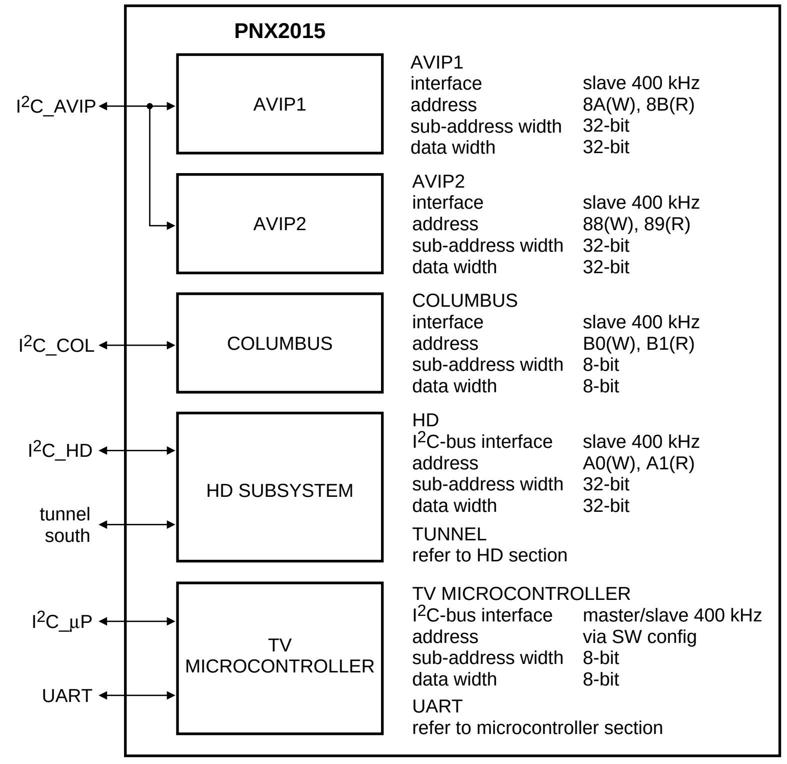


Fig 6. Control interface

2.3.5 Test (JTAG)

A JTAG interface is used for both IC testing via a TAP controller and boundary scan. A single boundary scan chain is provided for the whole of the PNX2015. Pads that are not included in the boundary scan chain are:

- Power supplies.
- Analog input and output.
- DDR/GDDR SDRAM Interface.
- LVDS interface.

The Boundary Scan Description Language (BSDL) file detailing the TAP Controller ID and Instructions is available on request from Philips Semiconductors.

2.4 Top level structure

The following paragraphs describe the top-level structure of the PNX2015 with respect to clocking, reset, interrupts and power supplies. Details of the subsystem implementation can be found in Section 2.5.

2.4.1 Clocking

The PNX2015 requires two clock sources, one for the TV Microcontroller subsystem and one for the remaining subsystems. These may be from a clock generator source, or from a crystal. See Figure 8.

The TV Microcontroller supports either 16 MHz or 24 MHz external crystal.

The HD, AVIP1 and AVIP2 require a clock of 27 MHz. Columbus receives a clock from either AVIP1 or AVIP2, depending on the selected input source. To ensure synchronization of video streams processed across the PNX8550 and PNX2015 devices, the 27 MHz is be supplied from PNX8550.

See *PNX2015* Limitations Section 8.1 for voltage divider network.

Remark: An external crystal may be used for testing purposes only.

2.4.2 Reset

The PNX2015 has two external reset signals, one for the TV Microcontroller subsystem, and one for the remaining subsystems. This is shown in Figure 7 and Figure 8. The TV Microcontroller reset is active high and held high for at least 30 clock cycles after the crystal frequency has become stable.

The reset for AVIP1 and AVIP2, Columbus and the HD subsystem is derived from a single input. The signal is active low and held low for at least 30 clock cycles after the 27 MHz clock is applied. The reset signal is extended within the subsystems for up to 1.2 ms to allow for all blocks to initialize correctly. After the 1.2 ms period communication with the subsystems can commence.

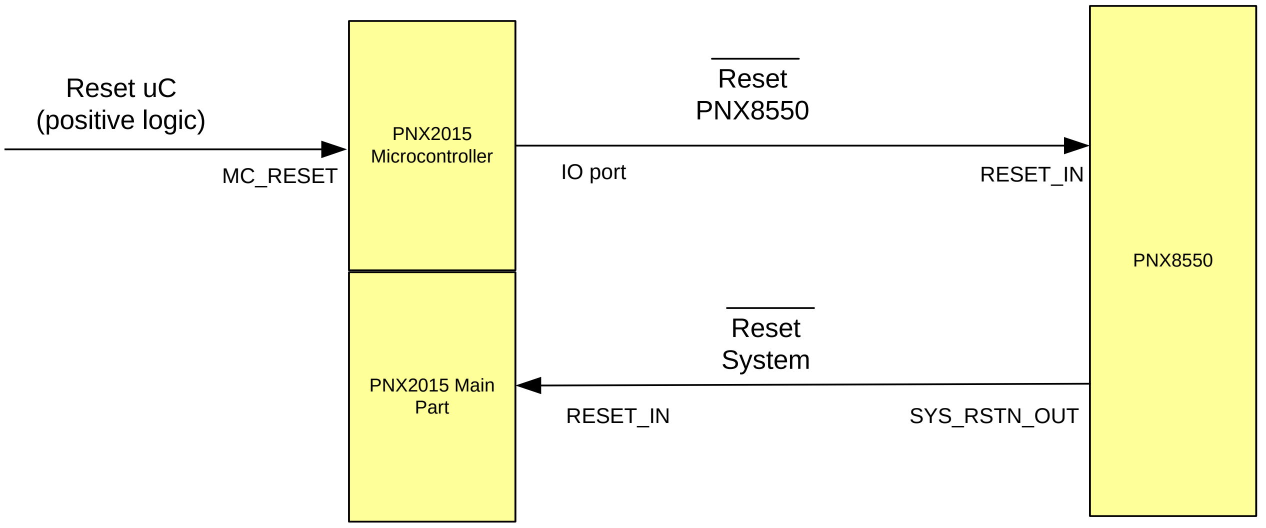


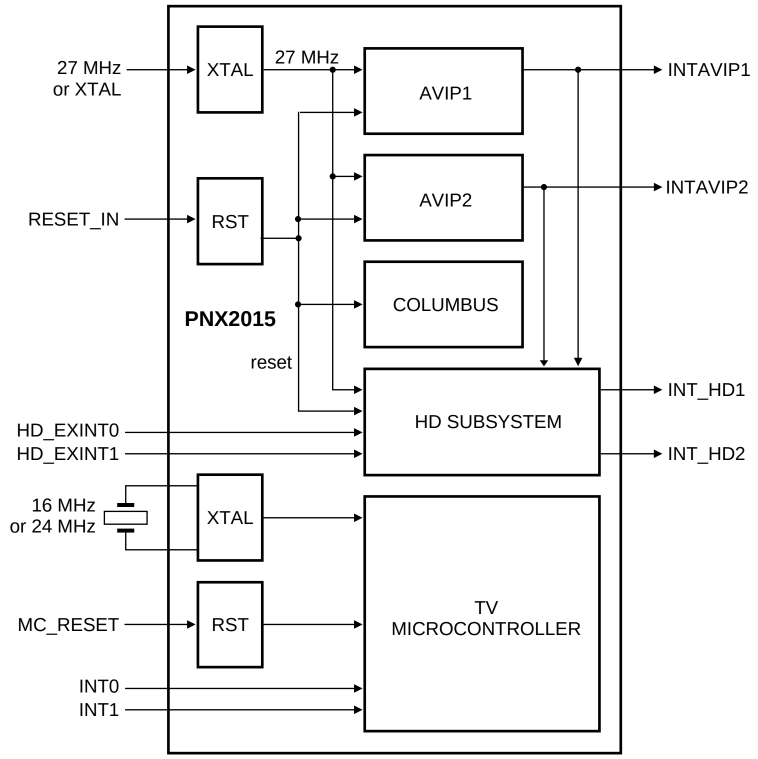
Fig 7. Reset path for PNX2015

2.4.3 Interrupts

The PNX2015 has an interrupt structure based on the individual subsystems. The details of all interrupt sources and control can be found in the subsystem descriptions.

The PNX2015 provides:

- Four interrupt outputs, one each from AVIP1 and AVIP2 and two from the HD
- Four interrupt inputs, two for the TV Microcontroller and two for the HD



Columbus input video data is clocked by AVIP1/2

Fig 8. Clock, reset and interrupts

2.4.4 Power supplies

The PNX2015 requires the following:

- 1.2 V for core logic
- 2.5 V for DDR-SDRAM and tunnel interfaces
- 3.3 V for periphery (excluding DDR and Tunnel)

and for the TV Microcontroller:

- 1.2 V for MC core
- 3.3 V for MC periphery

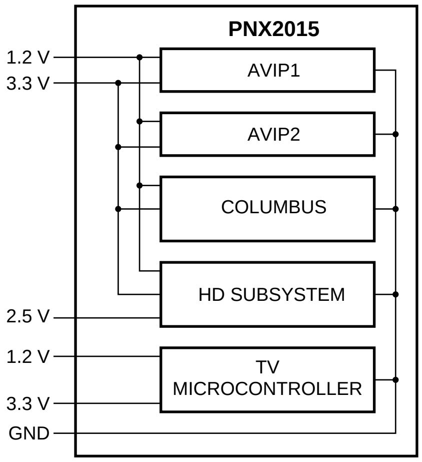


Fig 9. Subsystem power supplies

2.5 Subsystems

The following paragraphs describe in more detail the functions of the various subsystems.

The processor referred to by the term CPU has different meanings depending on where it occurs in this User Manual:

- In the DDR SDRAM section of the HD subsystem, CPU refers to a low latency MTL port of DDR SDRAM controller.
- In the TV microcontroller section CPU refers to the PNX2015 TV micro core processor.
- In the remainder of the document, CPU refers to a processor in the PNX8550 (MIPS or Trimedia).

2.5.1 AVIP1 and AVIP2

Each AVIP performs input decoding of single stream analog audio and single stream analog video signals. In addition, AVIP1 provides processing and presentation of all audio output streams in the system.

The AVIPs support the following features:

- Detection of PAL, NTSC or SECAM, and various $1f_H$ and $2f_H$ component video input sources.
- Full support for $1f_H$ and $2f_H$ video sources; progressive and interlaced.
- Decoding for global VBI Standards (WST, WSS, VPS, CC, VITC).
- ITU-656 output interface.
- Global multi-standard audio demodulation and decoding.
- Dolby® Pro Logic® II multi-channel audio decoding and post-processing.
- Advanced fully programmable audio post-processing functions, including psychoacoustic spatial algorithms for optimal loudspeaker matching.

2.5.2 Columbus

This block provides the following picture improvement functions:

- Enhanced 2D comb for PAL and NTSC.
- 3D field comb for PAL and NTSC.
- 3D frame comb for PAL and NTSC.
- Spatial noise reduction for all component video standards.
- Temporal noise reduction for all component video standards.

The comb filter is controlled via a separate I^2C interface on the PNX2015, this is to ensure registers containing measurement are accessed at appropriate times. The measurement information is also available as ancillary data within the video stream (ITU-656).

For certain features of the comb filter access to external memory is required. The PNX2015 has a unified memory that both comb filter and HD subsystem's share concurrently.

A functional block diagram is shown in Figure 140.

2.5.3 HD subsystem (High Definition)

The HD subsystem performs MPEG video decoding for up to MP@HL streams. It interfaces with PNX8550 and video coprocessor via tunnel interfaces (high bandwidth, low pin count interface), HD/SD video using DV4 and DV5 inputs and PNX8550 using DV1, DV2 and DV3 outputs.

The HD subsystem can also perform horizontal and vertical scaling of video images, and perform a range of video measurements on a stream.

A functional block diagram of the HD subsystem is shown in Figure 164.

2.5.4 LVDS transmission interface

The LVDS transmission interface connects to compatible LCD and flat panel displays. The block uses the LVDS technology as per the TIA/EIA standard. The protocol for video output encoding is selectable and can be THine™ or National™ semiconductors format.

The following features are supported in the LVDS transmitter IP:

- Single-link transmission of RGB video pixel data.

  Up to 30 bits of RGB pixel data, synchronization signals (HS and VS), data valid indication signal (DE), and up to two user-defined control bits (UD1 and UD2) sampled at the input using a 13.5 MHz to 86 MHz input clock.

- Transmission of two user defined control bits with four selectable values for each bit.
- Either 30, 24 or 18 bpp (i.e. 10, 8 or 6 bits-per-component) selectable video data transmission formats.
- Selectable polarity for the data valid (DE) input signal - programmable high or low level to designate (qualify) active video data.
- Support for selectable output transmission formats:

National™ semiconductor and THine™ formats for 18, 24 and 30 bits per pixel.

- Optional substitution of invalid pixel data by zero (RGB) values.
- Selectable diagnostic modes (stress test and pattern test).
- Selectable clock strobe - positive or negative edge of the clock for input data sampling.

An example of a system set-up is shown in Figure 245.

2.5.5 TV Microcontroller

The TV Microcontroller is the system power controller and supports the following functions:

- System power management; control and detection of power supplies.
- User interfacing via keyboard or remote control.
- Communication with external sources via AV_LINK Ref. 4.

The TV Microcontroller subsystem consists of an 80C51 core, together with extended memory and peripherals required for TV systems.

The TV Microcontroller is isolated from the other (Main) subsystems within the PNX2015, having its own power domain with 1.2 V and 3.3 V supplies, together with separate clocking and reset. This allows the TV Microcontroller to be active while all other subsystems are inactive, by either clock being disabled, or powered down, or by being held indefinitely in reset as shown in Table 9.

Table 9: Power supply connections for standby and other subsystem power saving modes

Signal/Supply description	Signal/Supply Pin name	Power down (Standby)	Clk disabled (27MHz)	Indefinite Reset	Main PNX2015 Operational
3.3V Main	VDDD3.3V	Not powered	Powered	Powered	Powered
2.5V Main	VDDD2.5V	Not powered	Powered	Powered	Powered
1.2V Main	VDDD1.2V	Not powered	Powered	Powered	Powered
27 MHz CLK	XTALI	Disabled	Disabled	Enabled	Enabled
Reset Main (active low)	RESET_IN	Low	Low	Low	High
3.3V Micro	VDDD(MCIO)	Powered	Powered	Powered	Powered
1V2 Micro	VDDD(MC_CORE)	Powered	Powered	Powered	Powered
Reset Micro (active high)	MC_Reset	Low	Low	Low	Low

The above table is the only legal reference to Power Down modes. This table supersedes all other imported IP references to Power Down modes that may be embedded within the User Manual. Any references outside of this table should be considered illegal. The Standby and Power Down modes embedded in IP blocks should not be used.

A functional block diagram is shown in Figure 253.

2.6 External memory requirements

The PNX2015 requires two external memories:

- 512 k bit (64 × 8) SPI Flash connected to the TV Microcontroller via the Motorola SPI™ interface. This enables the Microcontroller to download code during the initial boot sequence.The SPI is a master only, and cannot be used in slave mode. (See Section 7.14.3).
- 64 Mbits, 128 Mbits or 256 Mbits (8 MB, 16 MB or 32 MB) DDR[1] SDRAM connected to the DDR memory controller interface. See memory controller Section 5.1.14.

2.7 Power management

The PNX2015 has two separate power domains for the device. The first is for the TV Microcontroller subsystem the second is for the remaining AVIP, Columbus and HD subsystems. There are various modes with associated levels of power dissipation for the PNX2015. The following lists the modes in increasing level of power consumption.

Mode 1

Power is applied to the TV Microcontroller only. The TV Microcontroller is running from the 16 (or 24) MHz XTAL. All TV Microcontroller functions are available.

Mode 2

27 MHz clock is applied to the system and all subsystems are active.

2.7.1 Power supply sequencing

On power-up, the clock module will output all clocks to modules, defaulting to the 27 MHz xtal_clk clock. Reset of all modules and the boot-up sequence executed by the Boot block will run on the 27 MHz clock. During boot up the boot block will program all PLLs with their operating frequencies. After the 300 us PLL settling time, the boot block will set the exit_reset registers for these clocks and the clock module will switch these bus clocks from 27 MHz xtal_clk to their individual frequencies.

Power ON sequence

1. Apply power to $V_{DDD1.2V}$.
2. Allow $V_{DDD1.2V}$ to stabilize (approx.100 ms).
3. Apply power to $V_{DDD2.5V}$.
4. Apply power to $V_{DDD3.3V}$.

When the power supply to the TV Microcontroller is applied first, the same ordering of power supply sequence is used; firstly for the TV Microcontroller and subsequently the remainder of the PNX2015.

Power OFF sequence

1. Power may be removed simultaneously from $V_{DDD3.3V}$, $V_{DDD2.5V}$ and $V_{DDD1.2V}$.
2. Otherwise, remove $V_{DDD3.3V}$ followed by $V_{DDD2.5V}$ and $V_{DDD1.2V.}$

[1] For frequencies equal to or below 200Mhz use DDR, for frequencies above 200MHz GDDR should be used.

Remark: If supply sequencing is used and other parts of the system are powered ahead of the PNX2015, it is important to ensure that the 3.3 V supply to the PNX2015 is cleanly applied. i.e. no back-feeding of 3.3 V supply from other devices (or higher voltage rails that are powered first). This can occur with pull-ups or termination resistors to a system 5 V supply (e.g. I^{2}C) or to an inappropriate 3.3 V rails.

2.7.2 Low-power modes

The TV Microcontroller has two power-saving modes: Idle and Power-down, which are controlled by the PCON SFR. AVIP blocks are put into a low-power standby mode by disabling all PLLs. See Section 3.10.9.2.

3. AVIP1 and AVIP2

3.1 Introduction

Figure 10 shows a diagram of the AVIP block.

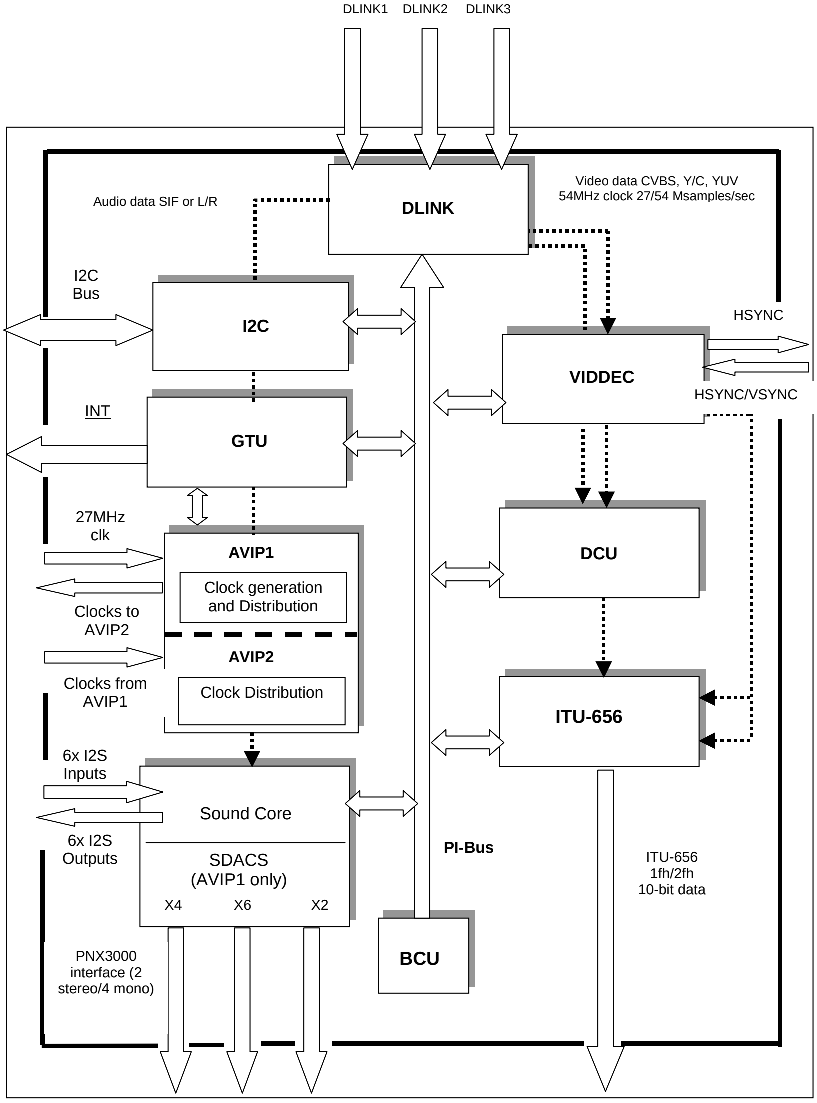


Fig 10.AVIP block diagram

Each AVIP provides the following functionality:

- Detection of PAL, NTSC or SECAM, and various 1f$_H$ and 2f$_H$ component video input sources.
- Full support for 1f$_H$ and 2f$_H$ video sources; progressive and interlaced.
- Decoding for global VBI Standards (WST, WSS, VPS, CC, VITC).
- ITU-656 output interface.

- Global multi-standard audio demodulation and decoding.
- Dolby® Pro Logic® II multi-channel audio decoding and post-processing.
- Advanced fully programmable audio post-processing functions, including psychoacoustic spatial algorithms for optimal loudspeaker matching.

3.2 Dual AVIP configuration in PNX2015

The AVIP1 sub system has the following differences compared to the standalone PNX2000.

- Fixed I^{2}C Address. Refer to Table 12
- Restricted I^{2}S input and Output capability. Refer to Section 2.3.2.
- Provides clocking sources for AVIP2

The AVIP2 sub system has the following differences compared to the standalone PNX2000.

- Fixed I^{2}C Address. Refer to Table 12
- Restricted I^{2}S input and Output capability. Refer to Section 2.3.2
- No dedicated PLL/Clock source, slaved from AVIP1
- No dedicated DACs, shared with AVIP1

AVIP modules are IP blocks.

To utilize a single MPIF AvPip mode, the AVIP2 incorporates a video mux controlled by register bit AVIP2_I2D_SEL.

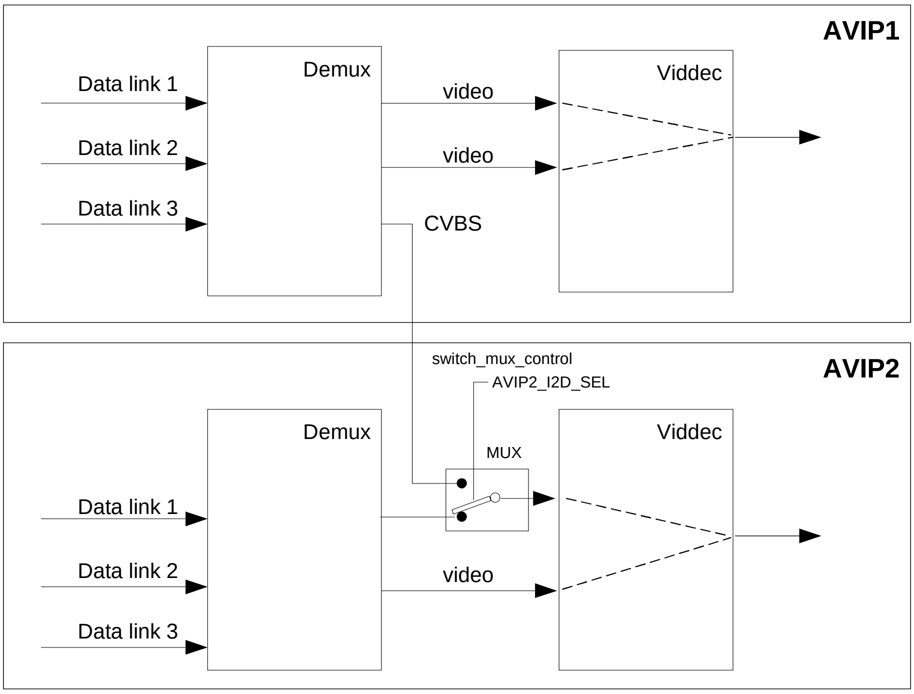


Fig 11. Single MPIF AvPip

3.3 Functional specification

3.3.1 Overview of functional partitioning

Table 10 shows the major functions mapped to hardware blocks (Figure 10).

Table 10: Major functions

Function	Block	Description
High speed data-link	DLINK	Receives data in 3 streams from PNX3000
Video decoder processor	VIDDEC	Decodes and processes CVBS, YUV or Y/C in YUV stream
Serial interface	I^2C	To access all the internal registers
Global Task Unit	GTU	Generates all the internal clocks, Reset and Power management
TV sound decoder	DEMDEC DSP	Demodulation, decoding of terrestrial TV audio standards
Audio processor	AUDIO DSP	Processing analogue and digital audio sources
Data Capture Unit	DCU	Acquires VBI data (Teletext, CC, VPS) and formats in a stream
Formatter unit	ITU-656	Formats YUV, VBI data and CVBS data in ITU-656
Bus Control Unit	BCU	Bus arbitration among all the internal blocks

3.3.2 AVIP interfaces

Table 11: Interfaces

Interface	Description
I^2C	Each AVIP is controlled using I^2C. It performs like an I^2C to PI bridge, i.e. translates I^2C-slave received commands to PI master commands.
DLINK	Receives data in three streams from PNX3000.
I^2S	Serial digital audio interface (6 stereo inputs and 6 stereo outputs) for connection to other devices that support the I^2S standard. Are used to receive decoded sound from a multi-channel digital audio decoder, provide additional ADCs and DACs, or loop audio signals through an external processor or delay line.
ITU656	Mainly intended to transfer output data streams externally to the PNX8550, but may also be read by other ITU 656 input devices that implement data valid signalling.
DACS	Digital-analogue converters used to generate analogue outputs from Sound Core

3.3.3 AVIP features

3.3.3.1 Video features

- Automatic Gain Control (AGC) to correct amplitude errors at input source.
- Synchronisation identification (used for channel search).
- Sync processing for 1FH and 2FH video input source.
- Standard detection of PAL, NTSC or SECAM and various 1FH/2FH component video input sources.

1FH video

- Color decoding (ITU-601) for PAL, NTSC or SECAM input sources.

- 2D comb filter.
- Supports component video sources with sync on CVBS or green.
- Fastblank insertion of RGB signals onto CVBS input.

2FH video

- Supports various progressive and interlaced component video sources.
- Synchronisation of video sources with sync on Y or external H/V inputs.

VBI data capture

- Decodes 525 line standards - WST, WSS, VPS, CC, VITC.
- Decodes 625 line standards - WST, WSS, CC, VITC.

ITU656 output interface

- Video and VBI formatted into an ITU-style output data stream, compliant to ITU-656/1364 REF [10] (exception being the use of a data valid signal).
- Interfaces to PNX8550 IC.
- Supports CVBS/C mode to interface to external picture improvement devices.

3.3.3.2 Audio features

Demodulator and decoder

- Demodulator and Decoder Easy Programming (DDEP)
- Auto standard detection (ASD)
- Static Standard Selection (SSS)
- DQPSK demodulation for different standards, simultaneously with 1-channel FM demodulation
- NICAM decoding (B/G, I, D/K and L standard)
- Two-carrier multistandard FM demodulation (B/G, D/K and M standard)
- Decoding for three analogue multi-channel systems (A2, A2+ and A2*) and satellite sound
- Adaptive de-emphasis for satellite FM
- Optional AM demodulation for system L, simultaneously with NICAM
- Identification A2 systems (B/G, D/K and M standard) with different identification time constants
- FM pilot carrier present detector
- Monitor selection for FM/AM DC values and signals, with peak and quasi peak detection option
- BTSC MPX decoder
- SAP decoder
- dbx noise reduction
- Japan (EIAJ) decoder
- FM radio decoder

- Soft-mute for DEMDEC outputs DEC, MONO and SAP
- FM overmodulation adaptation option to avoid clipping and distortion
- Sample rate conversion (SRC) for up to three demodulated terrestrial audio signals. It is possible to process SCART signals together with demodulated terrestrial signals.

Audio multi-channel decoder

- Dolby® Pro Logic® II Surround (DPL2); Trademark of Dolby Laboratories Licensing Corporation.
- Six channel processing for Main Left and Main Right, Subwoofer, Center, Surround Left and Surround Right.

Volume and tone control

- Automatic Volume Level (AVL) control
- Smooth volume control
- Master volume control and Balance
- Soft-mute
- Loudness
- Bass, Treble
- Dynamic Bass Enhancement (DBE)
- Dynamic Ultra Bass (DUBII)
- Non processed subwoofer
- 5 band equalizer
- Acoustical compensation
- Programmable beeper
- Noise generator for loudspeaker level trimming

Reflection and delay

- Dolby® Pro Logic® Delay
- Pseudo hall/matrix function

Psychoacoustic spatial algorithms, downmix and split

- Incredible Mono
- Incredible Stereo
- Virtual Dolby® Surround (VDS 522,523)
- Virtual Dolby® Digital (VDD 522,523)
- Bass Redirection according to Dolby specifications

Interfaces and switching

- Digital audio input interface (stereo I^{2}S input interface)
- Digital audio output interface (stereo I^{2}S output interface)
- Digital crossbar switch for all digital signal sources and destinations

- Output crossbar for exchange of channel processing functionality
- Audio monitor for level detection
- 8 audio DACs for six channel loudspeaker outputs and stereo headphones output
- 4 audio DACs for stereo SCART output and stereo LINE output.
- Serial data link interface for interfacing with the analogue multi-purpose interface IC PNX3000.

3.4 Control interface

3.4.1 AVIP control interface

The AVIP blocks are controlled via an I^{2}C interface. Internally, an I^{2}C-to-PI Bus bridge converts I^{2}C accesses into read and write transactions on the internal PI. This PI provides access to the control and status registers for all the modules in the AVIP design. The operation of the internal PI bus is controlled by the BCU block.

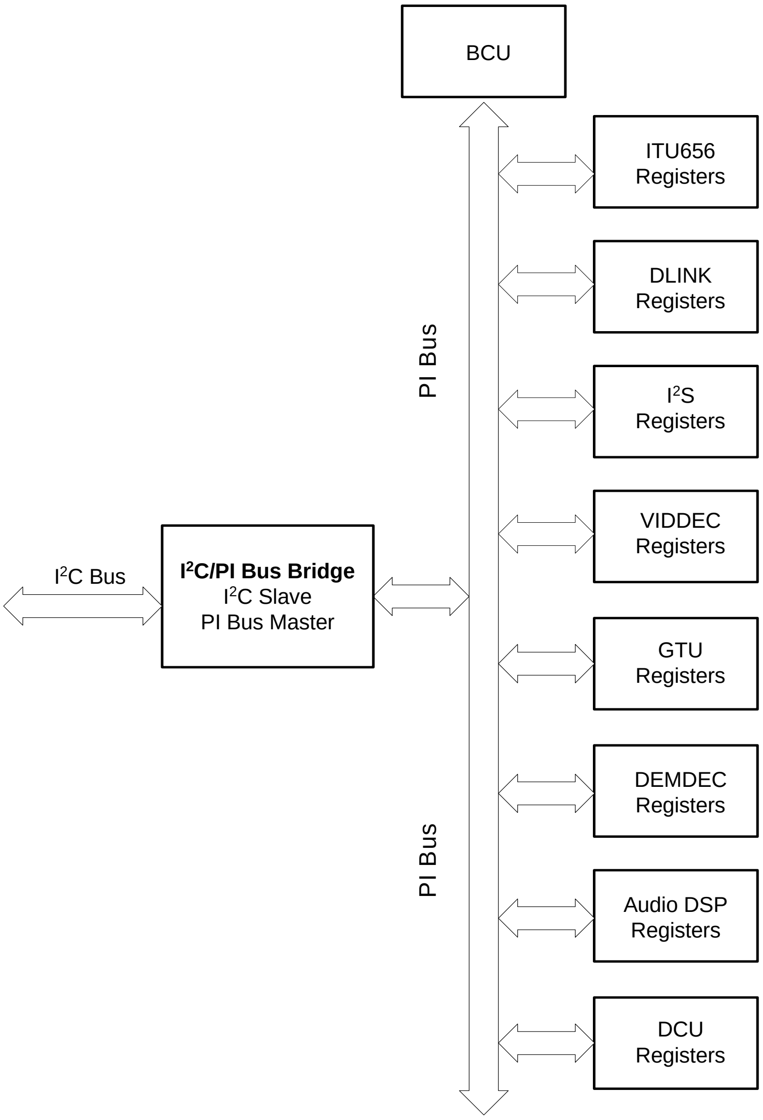


Fig 12.Control interface

3.4.2 I^{2}C interface

3.4.2.1 I^{2}C features

The I^{2}C module has the following features:

- 7-bit I^{2}C slave address.
- LSB of I^{2}C address selectable from external pin, to allow two AVIP blocks to coexist on a shared I^{2}C.
- PI Bus data width 32 bits.
- PI bus address width 32 bits.
- I^{2}C data transmitted in big endian format (MSB transmitted first).
- Up to 400 kHz I^{2}C speed.

3.4.2.2 AVIP I^{2}C address

Table 12: AVIP I^{2}C address

Block	Sub-address	Sub-address width	Data width
AVIP1	8A(W), 8B(R)	32-bit	32-bit
AVIP2	88(W), 89(R)	32-bit	32-bit

3.4.2.3 I^{2}C register access protocol

The following diagrams illustrate the procedure used to access register locations over the I^{2}C.

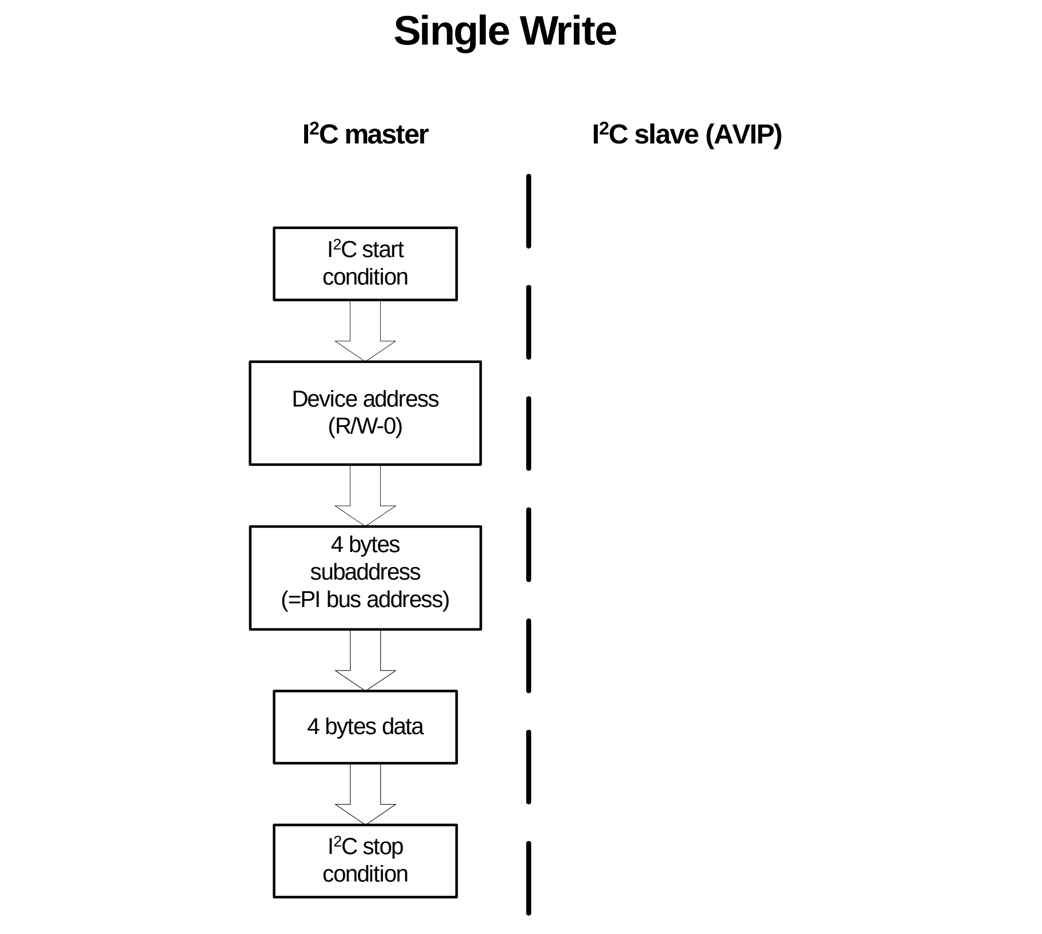


Fig 13. Single write

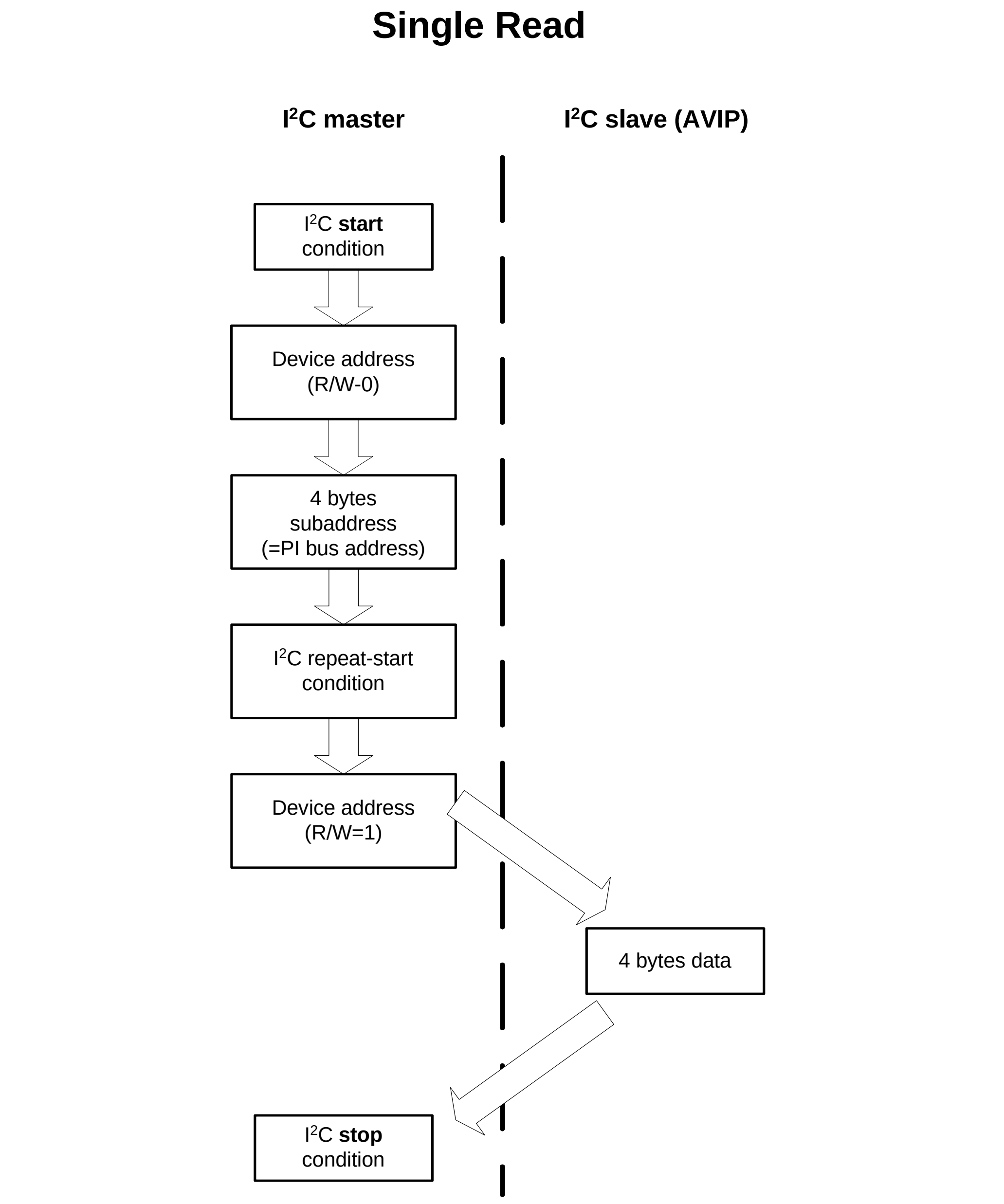


Fig 14. Single read

3.4.2.4 I²C interface block

The I²C interface module contains no software-accessible status or configuration registers.

If the internal PI locks up, the I²C interface will lock the external I²C bus by holding the SCL signal low. The only way to break the lock-up is to reset the entire AVIP block. In order to avoid this condition, the BCU timeout register must be configured by software early in the AVIP initialization process.

The AVIP I^{2}C module will not respond to a ‘general call’ on the I^{2}C, i.e. when a slave address of 0000000 is sent by a master. In case of any illegal address, transmission of the data that follows is not acknowledged, and the transmission is aborted.

The I^{2}C slave devices are capable of operating at a maximum speed of 400 kbits/s in accordance with the I^{2}C fast-mode specification.

3.4.3 BCU module

3.4.3.1 BCU features

The BCU module performs the following functions:

- Address space mapping and slave selection
- Bus error notification and logging
- Bus timeout monitoring, with software programmable timeout threshold
- Interrupt generation on bus error and timeout

3.4.3.2 Registers

The BCU contains eight software accessible registers which are listed in the following table. Note that the base address of the BCU is 0x07fe8000.

The “reset” values given in the tables in the following subsections correspond to the state of a variable after PI reset.

Table 13: BCU register summary

Address	Name	Description
0x0	INT_STATUS	BCU interrupt status
0x4	INT_SET	BCU interrupt set
0x8	INT_CLEAR	BCU interrupt clear
0xC	FAULT_STATUS	Bus fault status
0x10	FAULT_ADDRESS	Bus fault address
0x14	INT_ENABLE	BCU interrupt enable
0x18	TOUT	Time-out control
0x1C	SNOOP	Memory coherency control

Table 14: PI_1003_BCU registers

Bit	Symbol	Access	Reset value	Description
Offset 0x0 - INT_STATUS				
31:2	RSD[31:2]	R	0x00000000	Reserved
1	BCU_TO_STATUS	R	0x0	Time-out status: '0' = no time-out error has occurred. '1' = time-out error has occurred
0	BCU_BE_STATUS	R	0x0	Bus error status: '0' = no bus error has occurred. Fault logging enabled if BCU_TO=0 and BCU_BE=0; '1' = bus error has occurred, Fault logging stopped.
Offset 0x4 - INT_SET				
31:2	RSD[31:2]	R	0xXXXXXXXX	Reserved

Table 14: PI_1003_BCU registers *...continued*

Bit	Symbol	Access	Reset value	Description
1	BCU_TO_SET	W	0xX	Write '1' to set Time-Out interrupt; '0' has no effect
0	BCU_BE_SET	W	0xX	Write '1' to set Bus-Error interrupt; '0' has no effect
Offset 0x8 - INT_CLEAR				
31:2	RSD[31:2]	R	0xXXXXXXXX	Reserved
1	BCU_TO_CLEAR	W	0xX	Write '1' to clear Time-Out interrupt; '0' has no effect
0	BCU_BE_CLEAR	W	0xX	Write '1' to clear Bus-Error interrupt; '0' has no effect
Offset 0xC - FAULT_STATUS				
31:15	RSD[31:15]	R	0xXXXXX	Reserved
14:7	BCU_MASTER[7:0]	R	0xXX	Bus master of failed bus operation: 1 = Bus master 0; 2 = Bus master 1; 3 = Bus master 2; 4 = Bus master 3; &H10 = Bus master 4; &H20 = Bus master 5; &H40 = Bus master 6; &H80 = Bus master 7
6	BCU_LOCK	R	0xX	Lock status of failed bus operation: '0' = LOCK was 0; '1' = LOCK was 1
5	BCU_READ	R	0xX	Data direction of failed bus operation: 0 = write operation; 1 = read operation
4:0	BCU_OPC[4:0]	R	0xXX	Opcode of failed operation
Offset 0x10 - FAULT_ADDRESS				
31:2	BCU_ADDR[29:0]	R	0xXXXXXXXX	Address of failed bus operation
1:0	RSD[1:0]	R	0x0	Reserved. Returns '0' on read.
Offset 0x14 - INT_ENABLE				
31:1	RSD[31:1]	R	0x00000000	Reserved
0	BCU_INT_EN	R/W	0x0	'0' = disable BCU interrupt request; '1' = enable interrupt request - an interrupt is generated when BCU_TO and/or BCU_BE flag is set
Offset 0x18 - TOUT				
31:0	BCU_TO_THRESHOLD[31:0]	R/W	0x00000000	Time-out threshold: '0' = never time-out; '1' = time-out after 1st data cycle in bus operation; ‚Ä¶; 255 = time-out after 256th data cycle in bus operation
Offset 0x1C - SNOOP				
31:10	RSD[31:10]	R	0x000000	Reserved
9:2	BCU_SNOOP_MASTERS[7:0]	R/W	0x00	Snoop on masters: Bit 0 corresponds to master 0, ‚Ä¶, bit 7 corresponds to master 7; '1' on each bit to enable snooping
1	BCU_SNOOP_WRITE	R/W	0x0	Snoop on write: '0' = disable snoop on write; '1' = enable snoop on write
0	BCU_SNOOP_READ	R/W	0x0	Snoop on read: '0' = disable snoop on read; '1' = enable snoop on read

3.5 DLINK (also known as I^{2}D)

3.5.1 Introduction

This section provides an overview of DLINK functions and guidance in its use. The DLINK allows the transmission of data from PNX3000 (MPIF) to AVIP1 and AVIP2.

The communication between one datalink transmitter and one datalink receiver consists of two signals, a data signal and a strobe signal. The strobe signal contains the data, bit-sync and word-sync information. For optimal EMC performance both data and strobe are low voltage differential signals. The voltage swing on the wires is about 300mV.

In the PNX3000 the video and audio data to be transmitted is multiplexed in an output register of 42 bits. The content of that 42-bit register is serial transmitted on one of the three datalinks. In the AVIPs the serial data is demultiplexed into parallel streams. With a software selection in the PNX3000 you can choose which data you want to set in the output register for the datalink and in the AVIPs you have to make a selection which data from the datalink you want to use. The data on the datalink is divided in several groups of signals (video, audio and strobe_signals). It is important that the transmitter and receiver are in the same transmitting mode.

3.5.2 Functional capabilities of the links

The DLINK has the following characteristics.

- The datalink runs at 297 MHz / 594 Mbs.
- The driver rise/fall time is around 200 pS.
- The datalink uses differential signals.

The receiver has an internal termination resistor of 100 Ω differential connected.

- The differential threshold is 50 mV.
- The signalling voltages are between 200 - 500 mV.
- The datalink traces, both pairs should be of equal length and internally terminated with 100 Ω (The PCB-lines also characteristic).
- The max length of the datalink tracks is 20 cm (equal length), normal advice is 5 cm maximum
- The maximum capacitance on the line is approximately 15 pF.

3.5.3 Transmitter

In the PNX3000 the data coming from the A/D converters (digital video and audio) is multiplexed and put in data words. Each data word on the data links consists of 44 bits (4 video samples of 10 bits each, 2 audio samples of 2 bits and 2 word-sync bits). The word clock is 13.5 MHz. The data rate on each of the three data links is 44 bits/cycle*13.5*106 Cycle/s = 594 Mbit/s. For detailed transmission information see Table 16.

3.5.4 Receiver

The DLINK receiver module consists of three datalink receivers, and three Data Strobe receivers. The data receiver regenerates the serial data bit-streams, and converts them to parallel words of 42 bits (picture 4x10 bits and 2 bits of audio). When the data is ready for output a valid Word Sync pulse is generated in the DLINK receiver module. The Word Sync pulses are used by the clock domain separator to take over the 42 bits wide data from the DLINK receivers to the AVIP clock domain.

The clock domain separator module converts the data from the transmitter clock (13.5 or 27 MHz) domain (PNX3000) to the AVIP clock (13.5 / 54 MHz) domain. There is a clock domain separator necessary because the signals in the PNX3000 are processed via different paths and then multiplexed on a serial data line with a Data Strobe (and Word Sync). This leads to a static but unknown phase difference between the AVIP and PNX3000 clock. In addition, the duration of the serial data differs according the link length and group of data on the link and the different processing in the PNX3000. That is why a clock domain separator is necessary.

The data from the clock domain separator module is passed to the de-multiplexer module. This module formats the data into several audio and video streams (parallel data) together with accompanying VALid pulses derived from the clock domain separator (ready for takeover) to the Viddec and Demdec modules.

When the expected Word Sync pulse is not detected in the DLINK receiver, the clock domain separator still generates a DV pulse. The previous data is still on the parallel output lines of the DLINK receiver. When the Word Sync pulse is not detected, the counter counts the missing DataValids. This internal counting continues until it reaches the DV_MISS_MAX value. When the limit of DV_MISS_MAX is reached, an interrupt DVx_MISS_STAT is generated and a synchronisation action will follow. When the limit is reached the internal counter is frozen.

When the max value of DV_MISS_MAX is not reached and a Word Sync pulse arrives in the receiver window, the counter DV_MISS_MAX is reset.

When there is a situation in which the expected Word Sync pulse is detected in the DLINK receiver, but not within the data valid window (receiver window) of the clock domain separator, the pulse is Out Of Window (OOW). The clock domain separator generates a Data Valid (DV) pulse on the time that the clock domain separator expects to receive a Word Sync pulse from the receiver. The data can still be valid if the pulse comes to early, but if the pulse comes to late, the previous data can be on the output when the clock domain separator takes the data over. When the Word Sync pulse is out of his window detected, it generates an Out Of Window (OOW) pulse (referring to I2D_REC_SYNC_LOST). This Out Of Window pulse increments the counter (it counts the OOW pulses), the counter value itself cannot be read. This counting continues till it reaches the OOW_MAX value (register I2D _REC_SYNC_LOST). When the limit of OOW_MAX is reached, an interrupt SYNCx_LOST_STAT is generated, ref to I2D _INT_STATUS for more details, and a synchronisation action will follow. When the max value of OOW_MAX is not reached and a Word Sync pulse arrives in the receiver window, the counter OOW_MAX is reset.

At the end of the receiver is a de-multiplexer, the de-multiplexer reformats the data into several audio and video streams (parallel data) to the Viddec and Demdec.The functional block diagram of the receiver is shown in Figure 15.

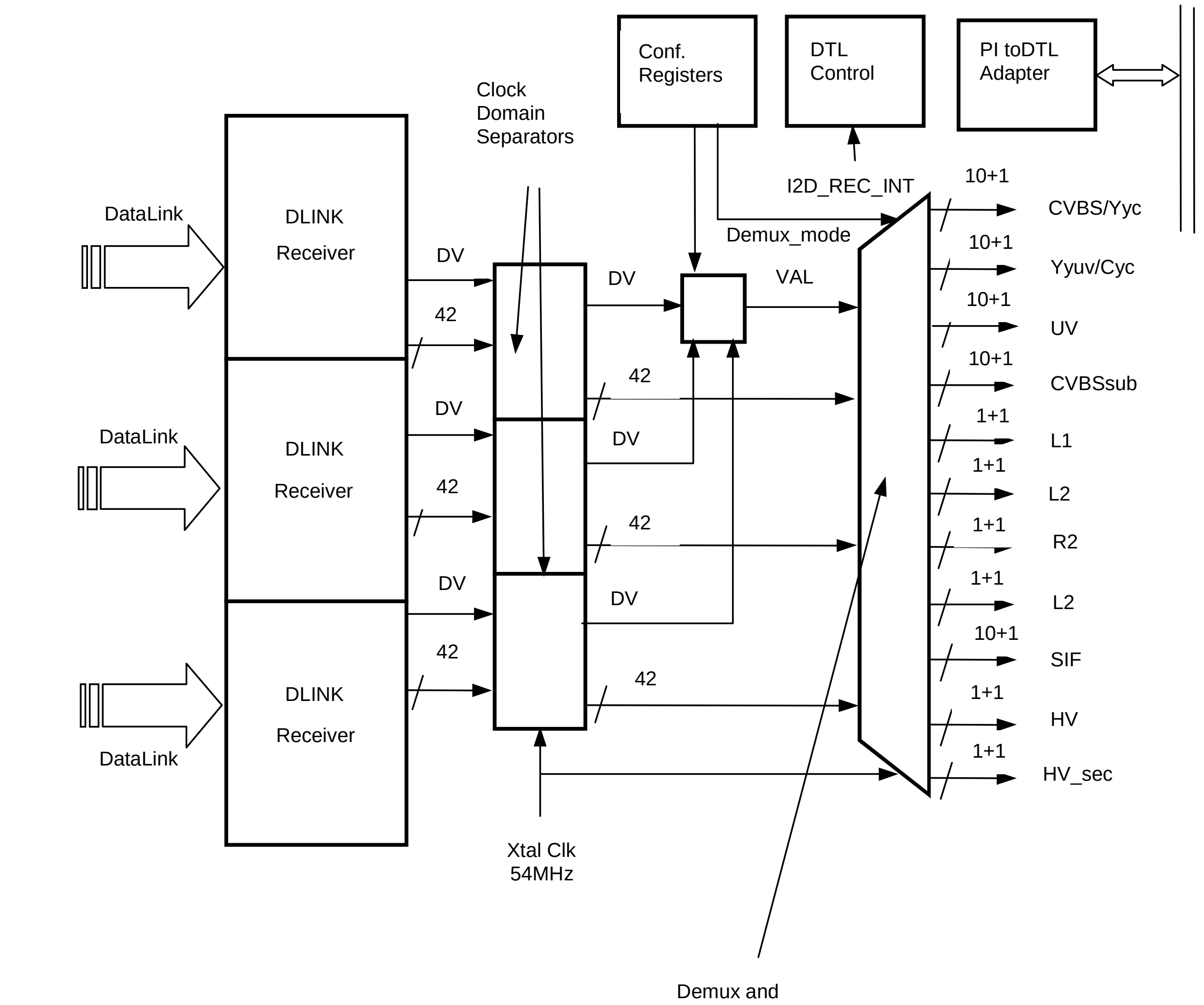


Fig 15.Simplified receiver AVIP1 and AVIP2

3.5.4.1 Transmitter / receiver transmission modes

The data from the PNX3000 can be sent in two modes (0,1) to the receiver in the AVIP. Table 15 describes the data that can be extracted from the datalink. The transmitter in the PNX3000 has to be set by the external I^{2}C communication link. The microprocessor in the AVIP transmits the mode settings and other multiplexer settings by the I^{2}C to the PNX3000. The receiver in the AVIP has to be configured in the same mode by the PI bus in the AVIP. Table 15 describes the dataflow and possible modes on each link. The software for the receiver runs in the MIPS processor in the AVIP. The software takes care of the boot sequences, interrupts and the use of the data on the datalink.

When the transmitter is in mode 0 (all three transmitters are in mode 0), the receiver has the possibility to extract data in mode 0a en 0b (for all three links together). This is possible due to the group of 42 bits send together, see Table 16.

When Viddec uses Y and C from mode 0 datalink 1, it can’t use YUV from datalink 2 (no sync available, they use the same bus in the demultiplexer output). If the Viddec use YUV from datalink 2 (input from RGB in PNX3000) it use the CVBS datalink 1 for sync.

Table 15: Content of data links

		Bits					
Mode setting	Mode reg value	41	40	39:30	29:20	19:10	9:0
Datalink 1							
mode 0a	0x0	R1	L1	Cn+1	CVBS or Yn+1	Cn	CVBS or Yn
mode 0b	0x1	R1	L1	-	CVBS or Yn+1	-	CVBS or Yn
mode 1	0x2	R1	L1	Yyuvn+3	Yyuvn+2	Yyuvn+1	Yyuvn
Datalink 2							
mode 0a	0x0	R2	L2	-	-	-	-
mode 0b	0x1	R2	L2	Vn	Yyuvn+1	Un	Yyuvn
mode 1	0x2	R2	L2	Vn+2	Un+2	Vn	Un
Datalink 3							
mode 0a	0x0	HVsec	HV	SIFn+1	CVBSsecn+1	SIFn	CVBSsecn
mode 0b	0x1	HVsec	HV	SIFn+1	CVBSsecn+1	SIFn	CVBSsecn
mode 1	0x2	HVsec	HV	SIFn+1	CVBSsecn+1	SIFn	CVBSsecn

A CVBS or Y signal may be connected to the inputs of the PNX3000. The type of signal on Datalink 1, in Mode 0 (a or b) is not known, but the preferred is shown **bold underlined** .

If from datalink 1 (mode 0b) the CVBS is used, via fast insertion the Viddec can use the YUV (1fH-mode) signals from datalink2 if the signal contains a sync signal.

The following figures show the use modes (video) in the receiver at the output of the multiplexer.

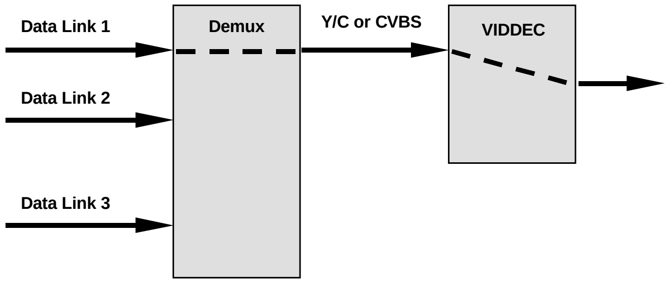


Fig 16.Mode 0a transmission

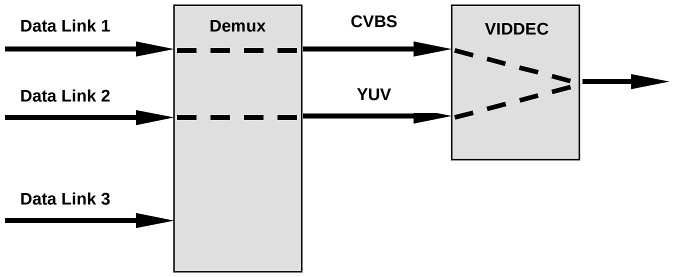


Fig 17.Mode 0b transmission (default)

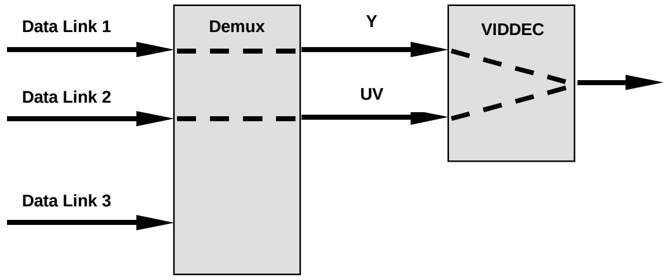


Fig 18.Mode 1 transmission (2fh on main channel, on sub is 1fh)

The control software has to set the correct settings in the PNX3000 and AVIP.

3.5.4.2 Data rate and timing output signals

The output rate of the data from the datalink receiver is shown in Table 16.

The HV_PRIM and HV_SEC are for the horizontal and vertical sync for the primary and secondary channel (timing pulses in IF part). These are clamping signals, which are coming from the VIDDEC. The frequency of the signals is dependent of the selected mode. These signals are not needed for DLINK.

Table 16: Data rate output signals

Pin name	Sample rate (Msamples/s)	
	1FH	2FH
CVBS_YYC_OUT	27	-
YYUV_CYC_OUT	27	54
UV_OUT	27	54
CVBS_SEC_OUT	27	-
LEFT1_OUT	6.75	
RIGHT1_OUT	6.75	
LEFT2_OUT	6.75	
RIGHT2_OUT	6.75	
HV_OUT	54	
HV_SEC_OUT	54	
SIF_OUT	27	

All data is generated on the negative edge of the 54 MHz clock.

3.5.5 Configuration registers.

The DLINK configuration registers are used to control the DLINK receiver module. For description of the de-multiplexer outputs see Figure 19.

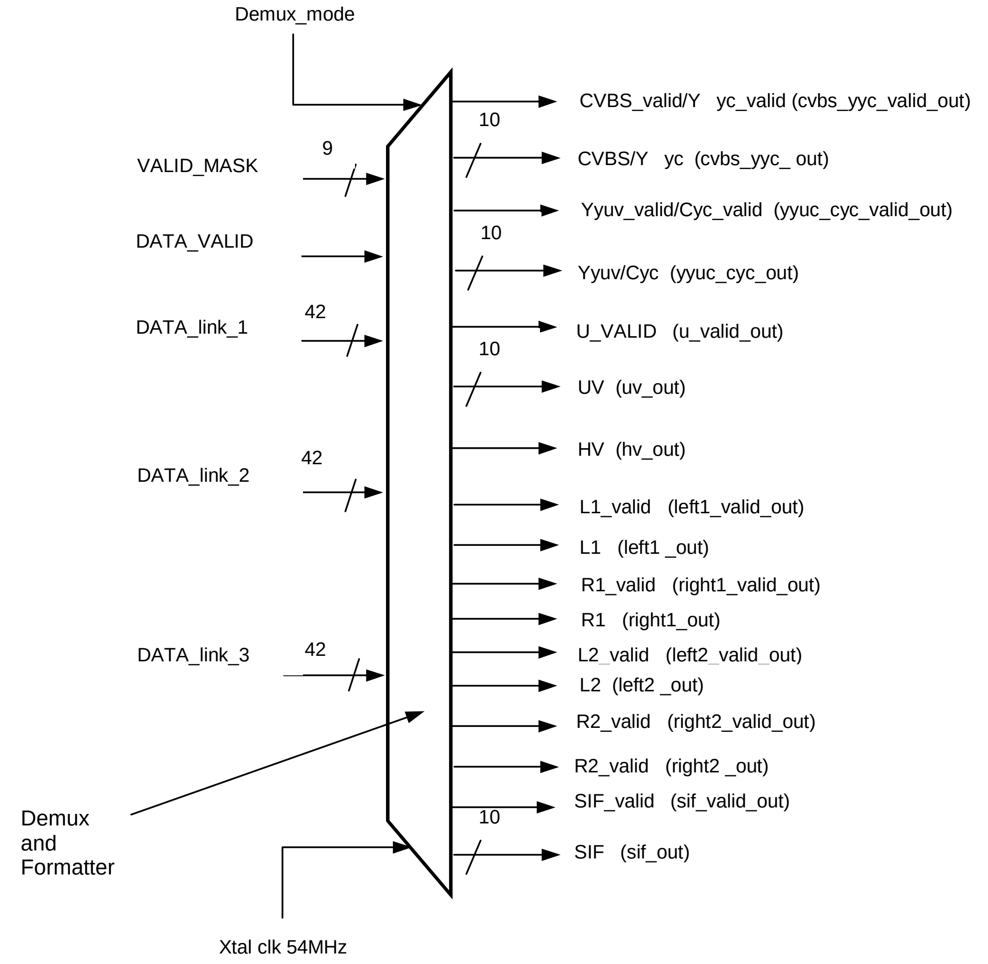


Fig 19.Read write and control flow

The DLINK configuration register block contains the control registers which are used to configure the DLINK receiver block, address decoder and two state machines. One to synchronize the write request and the register write enable, and the other one to synchronize the read request and read enable.

3.5.5.1 DLINK register map

This section provides information on the DLINK configuration. Each of the registers within the receiver block is described separately below. The base address for the DLINK is set to 0x07FF8000 (32 bits), the last 3 digits (12 bits) are for the DLINK control register.

Table 17: DLINK register summary

Address	Name	Description
0x0	RX_CTRL	Settings for analogue receiver
0x4	RX_STATUS	Status of analogue receiver
0x10	I2D_DTM_M_STAB	Number of flip flops for synchronization of the read and write access to DLINK config register.

Table 17: DLINK register summary *…continued*

Address	Name	Description
0x14	I2D_MTD_M_STAB	DLINK number of flip flops for synchronization of the read and write access from DLINK config register to the DTL target Controller.
0x18	REC_DEMUX_MODE	Demux settings
0x1C	REC_SYNC_LOST	Sync lost timer before generating an interrupt
0x20	PRBS_STAT	Pseudo Random Bit Sequence checksum status
0x24	PRBS_CTRL	Pseudo Random Bit Sequence checksum settings
0xFE0	I2D_INT_STATUS	Status of (possible) DVP interrupt requests
0xFE4	I2D_INT_ENABLE	Enable the DVP interrupt for request to the system irq controller
0xFE8	I2D_INT_CLEAR	Clear a DVP interrupts
0xFEC	I2D_INT_SET	Set a DVP interrupt
0xFFC	I2D_MOD_ID	Block information

Table 18: DLINK registers

Bit	Symbol	Access	Reset value	Description
Offset 0x0 - RX_CTRL				
31:1	RSD[31:1]	R/W	0x00000000	reserved
0	RX_APPL_PD	R/W	0x1	Power down for analog receiver in application mode. '0' : The analog receiver is active (normal mode) '1' : The analog receiver is in power down mode (For ADOC sleep/coma modes)
Offset 0x4 - RX_STATUS				
31:28	RSD[31:28]	R	0x0	reserved
27:16	RX_DC_MON[11:0]	R	0xXXX	Internal DC testmode monitor point
15:2	RSD[15:2]	R	0x0000	reserved
1	PD_STAT_STIMGEN	R	0x1	Power down monitor of internal testmode '1' : internal tester is powered down '0' : internal tester is active
0	PD_STAT_RX	R	0x1	Power down monitor of analog receivers '1' : analogue receiver is powered down '0' : analog receiver is active
Offset 0x10 - I2D_DTM_M_STAB				
31:2	RSD[31:2]	R/W	0x00000000	reserved
1:0	DTL_I2D_CTRL[1:0]	R/W	0x0	Number of flip flops used for synchronization during validation. Located between DTL target Ctrl to I2D '00' Have two metastability registers. '01' Have one register, bypass the other one. '1x' Bypass both metastability registers.
Offset 0x14 - I2D_MTD_M_STAB				
31:2	RSD[31:2]	R/W	0x00000000	reserved

Table 18: DLINK registers *...continued*

Bit	Symbol	Access	Reset value	Description
1:0	I2D_DTL_CTRL[1:0]	R/W	0x0	Number of flip flops used for synchronization during validation. Located between I2D to DTL ctrl '00' Have two metastability registers. '01' Have one register, bypass the other one. '1x' Bypass both metastability registers.
Offset 0x18 - REC_DEMUX_MODE				
31:18	RSD[31:18]	R/W	0x0000	reserved
17	I2D_SOFT_RESET	W	0x0	Softreset of the clock domain separator
16	I2D_RX_DATA_VALID_MASK	R/W	0x1	Mask the overall data valid flag inside the I2D (dv from the Clock Domain Separator). Advised not to use. '1' Enable '0' Hide
15:12	RSD[15:12]	R/W	0x0	reserved
11:3	I2D_DEMUX_VALID_MASK[8:0]	R/W	0x1FF	Mask data valid of the several I2D outputs. Each bit: '0' to hide the valid signal. [11] SIF [10] Right 2 [9] Left 2 [8] Right 1 [7] Left 1 [6] CVBS sec [5] U [4] Y [3] CVBS
2:0	DEMUX_MODE[2:0]	R/W	0x1	Select the I2D content format to output mode '000' mode 0a '001' mode 0b '010' mode 1 '100' mode 2 '110' mode 3
Offset 0x1C - REC_SYNC_LOST				
31:16	DV_MISS_MAX[15:0]	R/W	0x03E8	Number of consecutive valid pulses missing before generating an dv error interrupt '0' disables detection and/or resets the counter To ensure proper counting during lowering this value, first write a value of 0 into the max value, before lowering this marker. Otherwise the hold counter marker might be shifted over the counter what will result in a 16 bits overcount (afterwards it will continue at 0).
15:0	OOW_MAX[15:0]	R/W	0x03E8	Number of consecutive valid pulses out of the catching window before generating an 'out of sync' (sync lost) interrupt '0' disables detection and/or resets the counter To ensure proper counting during lowering this value, first write a value of 0 into the max value, before lowering this marker. Otherwise the hold counter marker might be shifted over the counter what will result in a 16 bits overcount (afterwards it will continue at 0).
Offset 0x20 - PRBS_STAT				
31:7	RSD[31:7]	R	0x0000000	reserved
6	DV_UNDET	R	0x1	Global data valid undetected '1' DV undetected yet '0' DV has been detected
5	DV3_UNDET	R	0x1	Data valid of datalink 3 undetected '1' DV undetected yet '0' DV has been detected
4	DV2_UNDET	R	0x1	Data valid of datalink 2 undetected '1' DV undetected yet '0' DV has been detected

Table 18: DLINK registers *…continued*

Bit	Symbol	Access	Reset value	Description
3	DV1_UNDET	R	0x1	Data valid of datalink 1 undetected '1' DV undetected yet '0' DV has been detected
2	DLINK3_ERROR	R	0x0	Error on datalink 3
1	DLINK2_ERROR	R	0x0	Error on datalink 2
0	DLINK1_ERROR	R	0x0	Error on datalink 1
Offset 0x24 - PRBS_CTRL				
31:8	RSD[31:8]	R/W	0x000000	reserved
7	PRBS_ENABLE	R/W	0x0	Enable check on Pseudo Random Bit Sequence
6	DV_UNDET_SET	W	0x0	Set data valid detection status to undetected for global dv
5	DV3_UNDET_SET	W	0x0	Set data valid detection status to undetected for datalink 3
4	DV2_UNDET_SET	W	0x0	Set data valid detection status to undetected for datalink 2
3	DV1_UNDET_SET	W	0x0	Set data valid detection status to undetected for datalink 1
2	DL3_ERR_RST	W	0x0	Clear error status bit of datalink 3
1	DL2_ERR_RST	W	0x0	Clear error status bit of datalink 2
0	DL1_ERR_RST	W	0x0	Clear error status bit of datalink 1
Offset 0xFE0 - I2D_INT_STATUS				
31:6	RSD[31:6]	R	0x0000000	reserved
5	DV3_MISS_STAT	R	0x0	Data valids are missing for datalink 3 The max value dv_miss_max has been reached.
4	SYNC3_LOST_STAT	R	0x0	Data valid out of sync indication for datalink 3 The max value oow_max for out of window dv pulses has been reached.
3	DV2_MISS_STAT	R	0x0	Data valids are missing for datalink 2 The max value dv_miss_max has been reached.
2	SYNC2_LOST_STAT	R	0x0	Data valid out of sync indication for datalink 2 The max value oow_max for out of window dv pulses has been reached.
1	DV1_MISS_STAT	R	0x0	Data valids are missing for datalink 1 The max value dv_miss_max has been reached.
0	SYNC1_LOST_STAT	R	0x0	Data valid out of sync indication for datalink 1 The max value oow_max for out of window dv pulses has been reached.
Offset 0xFE4 - I2D_INT_ENABLE				
31:6	RSD[31:6]	R/W	0x0000000	reserved
5	DV3_MISS_ENA	R/W	0x0	Enable interrupt for out of window indication for datalink 3
4	SYNC3_LOST_ENA	R/W	0x0	Enable interrupt for lost of sync of datalink 3

Table 18: DLINK registers *…continued*

Bit	Symbol	Access	Reset value	Description
3	DV2_MISS_ENA	R/W	0x0	Enable interrupt for out of window indication for datalink 2
2	SYNC2_LOST_ENA	R/W	0x0	Enable interrupt for lost of sync of datalink 2
1	DV1_MISS_ENA	R/W	0x0	Enable interrupt for out of window indication for datalink 1
0	SYNC1_LOST_ENA	R/W	0x0	Enable interrupt for lost of sync of datalink 1
Offset 0xFE8 - I2D_INT_CLEAR				
31:6	RSD[31:6]	R	0x0000000	reserved
5	DV3_MISS_CLR	W	0x0	Clear indication for data valid out of window for datalink 3
4	SYNC3_LOST_CLR	W	0x0	Clear indication for lost of sync of datalink 3
3	DV2_MISS_CLR	W	0x0	Clear indication for data valid out of window for datalink 2
2	SYNC2_LOST_CLR	W	0x0	Clear indication for lost of sync of datalink 2
1	DV1_MISS_CLR	W	0x0	Clear indication for data valid out of window for datalink 1
0	SYNC1_LOST_CLR	W	0x0	Clear indication for lost of sync of datalink 1
Offset 0xFEC - I2D_INT_SET				
31:6	RSD[31:6]	R	0x0000000	reserved
5	DV3_MISS_SET	W	0x0	Set indication for data valid out of window for datalink 3
4	SYNC3_LOST_SET	W	0x0	Set indication for lost of sync of datalink 3
3	DV2_MISS_SET	W	0x0	Set indication for data valid out of window for datalink 2
2	SYNC2_LOST_SET	W	0x0	Set indication for lost of sync of datalink 2
1	DV1_MISS_SET	W	0x0	Set indication for data valid out of window for datalink 1
0	SYNC1_LOST_SET	W	0x0	Set indication for lost of sync of datalink 1
Offset 0xFFC - I2D_MOD_ID				
31:16	I2D_MODULE_ID[15:0]	R	0x0141	Module identifier
15:12	I2D_MAJOR_REV[3:0]	R	0x0	Major Revision. Any revision that may break SW compatibility.
11:8	I2D_MINOR_REV[3:0]	R	0x0	Minor Revision. Any revision that still keep SW compatibility.
7:0	I2D_APERTURE[7:0]	R	0x00	Aperture Size.

3.5.6 Interrupt procedure

When the DLINK core detects an interrupt condition, i.e. a situation that requires software interaction, it sets the corresponding Internal Interrupt Status bit in the Interrupt Status register. Then the DLINK checks whether this interrupt condition is enabled by inspecting the corresponding bit in the Interrupt Enable register. If this bit is ‘1’, a system interrupt request is generated.

The software should remove the cause of the interrupt condition by taking the appropriate action. As soon as the cause is removed, the Internal Interrupt Status bit in the Interrupt Status register should be cleared by writing a ‘1’ to the corresponding bit of the Interrupt Clear register.

For debugging purposes the software can also generate ‘fake’ interrupt conditions by writing a ‘1’ into bit i of the Interrupt Set register. The result is that the Interrupt Status bit i will go high.

3.5.6.1 Interrupt behavior

The DLINK interrupt architecture contains 4 registers: status, enable, set and clear. Activation of an interrupt request starts as soon as the interrupt condition becomes true. The interrupt condition can be read from the status register, its name indicates the interrupt generated. Every interrupt condition (set interrupt or write action) can set bits in the status register. The status register indicates one or more pending interrupt conditions.

To disable interrupts:

- via register INT_ENABLE to enable/disable interrupts on a line
- via setting OOW_MAX and DV_MISS_MAX to 0x0.

To clear interrupts:

- write 0x0 in DV_MISS_MAX and OOW_MAX register
- clear the interrupts via register INT_CLEAR (write 0x3F to it)
- set default value 0x50 in DV_MISS_MAX and OOW_MAX register

3.5.6.2 Software action with registers

When an interrupt occurs, the reaction of the system is customer dependent. It is recommended to execute the soft-reset procedure. A software loop should check that the fault situation will not occur again (polling or via interrupts). There are several operating situations that can occur:

1. Start up.
2. Normal operation.
3. Soft_reset.
4. Change of source selection.
5. Sync lost on a datalink.
6. Missing of data valid pulses.
7. Test mode: pseudo random mode, set interrupt status bit for lost data and/or sync.

Start up: During start up, registers will have a default value after releasing the reset to the DLINK receiver registers.

The bit 0 in I2D_RX_CTRL has to activate the receiver. A 0 has to be written to it, by startup the bit is: 1 (power down), needed to power down the HF datalink receiver in sleep and coma modes. Enabling is needed for normal operations.

During start up the clock domain separator has to lock on the Data Valid signals. After power up all the DLINK interrupt sources are disabled. The clock domain separator block is waiting for Data Valids (validity the data of the corresponding datalink) coming from the three analog datalinks. If there are Data Valids on at least two data_links, which are in the same clock period, the clock domain separator is locked on these pulse rates.

Procedure at start up:

- Activate the receiver in reg: I2D _RX_CTRL, bit 0 (RX_APPL_PD) write 0.
- Disable the DV_MISS_MAX and OOW_MAX counter, by writing a 0, (disable int)
- Give a Soft_Reset, in I2D _REC_DEMUX_MODE write bit 17.
- Write 3F to the INT_CLEAR,
- Enable the DV_MISS_MAX and OOW_MAX counter and write default 0x50, the minimum value is 2.

After this soft-reset the clock domain receiver locks again and Data and Strobe signals become stable. The receiver should now enter Normal operation, if not refer Section “Missing data_valid pulses”.

Normal operation: During normal operation AVIP sends an operating status signal to the PNX3000 receiver (1 per second). When the receiver is in lock, the clock domain separator continues to check that the Data Valid pulses are coming in the right window from the datalinks. If the clock domain separator does not get Data Valid pulses within the desired window, the number of missing Data Valid, or Out Of Window pulses, from the corresponding link is incremented. If the number missing Data valid pulses to the de-multiplexer is larger than OOW_MAX, or DV_MISS_MAX, the interrupt status is set and an interrupt can be generated. When this happens, operating condition 5 or 6 occur.

Remark: If the clock domain separator does not get Data Valid signal within the desired window, the data can still be valid.

Soft_reset: When a fault condition appears the clock domain separator gets out of lock (no data pulses are detected within the window), when the limit of OOW_MAX or DV_MISS_MAX is reached. However, the software resets the Rec_Demux_mode register 07FF8018 and bit 17 from this register resets the clock domain separator and the receiver can again lock on the data-stream.

Such a fault condition appears during start-up, for this reason the receiver is powered down during switch on.

A temperature change, or start-up transient can cause fast phase shift of the datalink and the clock domain receiver does not receive DV pulses within the catching or locking window.

Change of source selection: When there is a request to change the video source, the AVIP gives a command by the I^{2}C to the PNX3000 to change source. When the PNX3000 changes the source, the Data Valids generated in the receiver are still right, but the contend from a packet is not right (due to asynchronous switch over). The MIPS_software itself has to find out where the PNX3000 has changed over. The change over is in one packet.

Sync lost on datalink (Out of Sync): If there is a Sync_lost indicator (one of the internal counters reaches the OOW_MAX value on one of the datalinks), the data can be disturbed. When there are in the counter OOW_MAX too many word sync out of the locked window, the output data is not stable. The software has to soft_reset to recalibrate the clock domain.

In this situation carry out the following:

1. Re-calibrate the clock domain with a soft_reset from the I2D_REC_DEMUX_MODE register.
2. Clear the internal Out Of Window counter by writing 0 into OOW_MAX and afterwards write back the chosen value (recommended is 0x50).
3. Clear the interrupt by writing 0x3F into I2D_INT_CLEAR.

This calibration loop is needed to ensure a proper picture on the output. There can appear speckles on the screen and sound can be disturbed, if this interrupt is raised. Software can decide whether they regular poll the status register or enable the interrupt.

Missing data_valid pulses: When the clock domain separator doesn't receive word sync pulses, the DV_MISS_MAX counter for a respective link is incremented. However each consecutive time a word sync (or data valid) is received, the DV_MISS_MAX counter of the appropriated datalink is reset. When the counter reaches the programmed value, defined in DV_MISS_MAX register, the corresponding bit of the INT_STATUS register (DV3_MISS_STAT or DV2_MISS_STAT or DV1_MISS_STAT) is set to 1 and the datalink receiver generates an interrupt flag.

If there is an indicator of data valid missing raised, meaning DV3_MISS_STAT DV2_MISS_STAT DV1_MISS_STAT is set to '1', it is likely that the output data is invalid. Change in value of a data valid missing indicator from '0' to '1' may be an indicator for software to perform a recovery to improve picture and sound quality.

The following steps can be executed in this situation:

1. Check the status of the datalink receivers. The RX_APPL_PD of the I2D_RX_CTRL register should be set 0 and the PD_STAT_RX (bit 0) of I2D_RX_STATUS should be equal. If this status bit is 1, there is an internal hardware problem and should be stored in the Error register.
2. Set a soft_reset item from the I2D_REC_DEMUX_MODE register to '1' in order to calibrate the clock domain again.
3. Write '0' into DV_MISS_MAX and afterwards write back again the chosen value (recommended is 0x50). This action is necessary to clear the internal DV_MISS counter.
4. Clear the interrupts by writing 0x3F into I2D_INT_CLEAR.
5. Put PPRS_ENABLE to '1' to start measurements on the DLINK Receiver.
6. If the DV_UNDET bit remains high in 100ms, the I^{2}D can not lock anymore to the input. If the DVx_UNDET bits remain high in 100 ms, the corresponding data valid did not arrive at all. In this situation, there is an external (hardware) problem and this should be logged in the Error register.
7. If the interrupt returns within 1 second, there is an external (hardware) problem of bad reception and this should be logged in the Error register.

Software can decide whether they regularly poll the status register or enable the interrupt. Under normal circumstances this interrupt should not appear, but if it does (due to a hardware defect or external factors) software should act as proposed in this section. It is up to the customer whether this software loop is implemented and if errors are logged in the Error register.

It is advised to blank the picture output when this condition appears, since the data is corrupted. Most likely Viddec won't be able to lock and sound is disturbed. Check whether the PNX3000 is booted up and functioning properly.

Test mode: It is not possible to do a boundary scan of the datalink transmitters in the PNX3000 or the inputs of the AVIP receiver, which makes it difficult to test the ICs on those points. Therefore it is possible to bring the PNX3000 in pseudo-random mode (from version PNX3000 N1D). In this mode the manufacturer can evaluate the transmitter-receiver link on various data profiles and analyze the link behavior.

This test mode can be used to evaluate the data transfer from PNX3000 to AVIP.

The procedure is described below.

1. Blank the picture on the screen and switch off the sound output, to avoid noise on the screen and noise out of the speakers.
2. Set the PNX3000 in pseudo-random mode via the PRND bit from the PNX3000 Datalink_mode register. See the PNX3000 user manual.
3. Write a '1' to Soft_reset from the I2D_REC_DEMUX_MODE register to calibrate the clock domain again.
4. Set the OOW_MAX and DV_MISS_MAX counter in register: REC_SYNC_LOST to 0x0. (Disables counter and interrupts generation) and write back again the chosen value (recommended is 0x50).
5. Clear the I2D_INT_CLEAR register by writing 0x3F.
6. Activate the Pseudo-Random Bit Sequence check. Write 1 to PRBS_ENABLE (bit 8) of the I2D_PRBS_CTRL register.
7. When the PRBS mode is activated, the circuit checks the data coming from the PNX3000. Afterwards poll the Pseudo Random Bit Sequence status (PRBS_STAT) register regular. The value of this register should be 0x0.

- When the DVx_UNDET did not become '0', the datalink did not receive any datavalid from the HF datalink receivers.
- When the DLINKx_ERROR is '1', it means the pseudo-random data was not right on the line the corresponding error bit. This bit stays high until it is cleared by toggling PRBS_ENABLE.
- When Interrupt DVx_MISS_STAT is high, the data valid does not appear regular.
- When Interrupt SYNCx_LOST_STAT is high, the datalink is not calibrated well. A soft_reset should be executed to calibrate again.

The working of PRBS registers is independent of the OOW and DV_MISS counters. The data speed on the line is very high, so you know immediately if the line is good.

If no errors are observed during execution of the test, software can again switch off the PRBS mode of PNX3000 and the DLINK receiver and release the system (i.e. enable sound and picture output).

If there are errors it means that:

- The transmitter in the PNX3000 is not functioning,
- The receiver in the AVIP is not functioning,
- The wire connection is not good (with one open wire it was found that the transmission was still good. (The wire connection can be checked with the DCF status bit of PNX3000).
- There are outside disturbances (e.g. EMC, power stability).

3.6 Video processing (VIDDEC)

3.6.1 VIDDEC overview

VIDDEC has the following features:

- AGC on all inputs to ensure optimum use of the bit range
- CVBS and Y/C input
- Multi-standard color decoder including PAL M and PAL N
- 2D Comb Filter
- YPrPb / RGB processing, both 1Fh and 2Fh
- Sync on CVBS, Y or external (external 2Fh only)
- Fast blanking for RGB on SCART (1Fh only)

The input can handle CVBS, Y/C and YUV signals.

Remark: The system can also handle RGB signals because the PNX3000 converts RGB signals to YUV. The Y from YUV and C from Y/C share the same channel. In practice Y/C and YUV are not present at the same time, so this is no limitation.

The signals from the DLINK receiver block are fed to the input of a data synchroniser block. All incoming data streams are 10 bits wide and have the same sample frequency. U and V, which are sampled at half the sample frequency of CVBS, C and Y, are combined in one data stream.

The U and V stream is demultiplexed in separate U,V streams for further processing. In the sample rate converter the data streams are transferred from the free running sample clock to a line locked clock domain. At the same time the data sample size is increased to 13 bit.

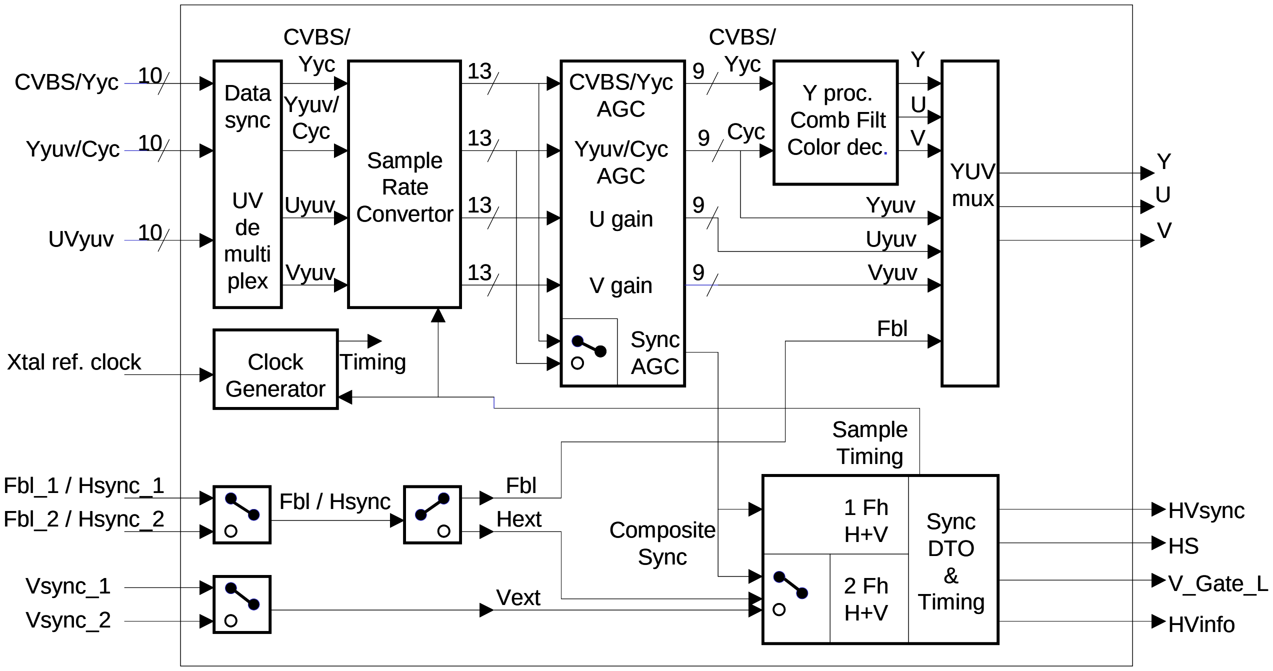


Fig 20.Block diagram VIDeo DECoder

The 13 bit wide data streams enter the AGC block. This block takes care to fit the incoming signals optimally in the available 9 bit space for further processing in the chain. Level deviations at the PNX3000 inputs from +3 dB to -3dB are corrected. In this way no excessive headroom needs to be reserved which improves the signal to noise in the chain. The sync signal has a separate AGC block. The input for the Sync AGC can be taken from the CVBS/Y channel or from the Y channel from YUV.

The CVBS and Y/C data are fed to a multi-standard color decoder. This decoder can handle all world standards of PAL, SECAM and NTSC including Latin America. All necessary filtering and traps are included. The input of the color demodulator can be switched between the CVBS signal and the C signal to enable Y/C processing. The decoder also incorporates a 2D comb filter for PAL (4 lines) and NTSC (2 lines) for improved luminance and chrominance separation.

The YUV at the output of the color decoder connect to a YUV switch. At the other input of this switch the YUV signals from the YUV input are connected. The switch can be controlled by an external voltage (Fast Blanking on insertion pin) or forced by software. A formatter combines the U and V stream again to one data stream with the same sample frequency as the Y stream The sync output from the AGC goes to the synchronisation block. This block generates the Horizontal and Vertical pulses for further processing (HVsync), as well as timing information for the PNX3000 for correct black level clamping (HVinfo).

The YUV path and the synchronisation can handle both 1Fh signals as 2Fh signals. For 2 Fh signals, the sampling frequency for YUV is doubled and also the synchronisation uses a special 2 Fh part for sync processing.

The sync can be derived from the Y signals or from external H and V pulses. Also ATSC YUV signals (tri level sync and Fh = 33.75 kHz) can be handled in 2 Fh mode. To process 2Fh signals, set the VIDDEC in 2Fh mode by doubling one of its clock frequencies coming from another block in AVIP.

3.6.2 Data input, sample rate converter and timing

Figure 21 shows typical input and sample rate conversion.

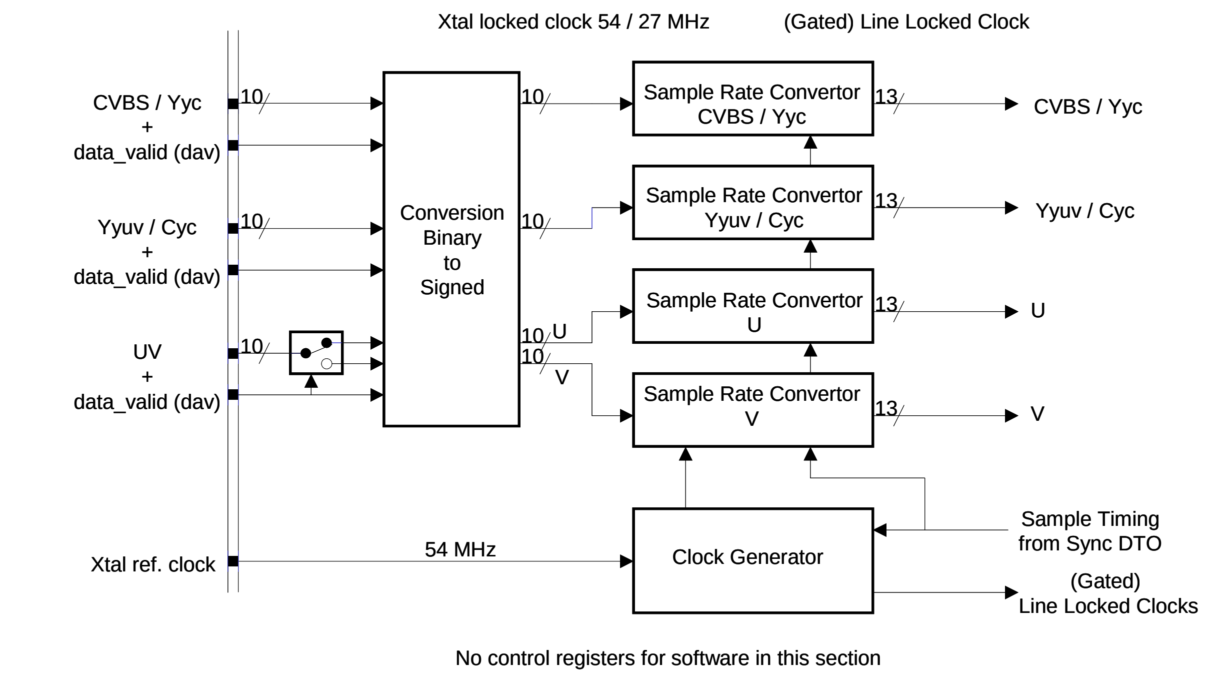

Xtal locked clock 54 / 27 MHz
(Gated) Line Locked Clock
CVBS / Yyc
+
data_valid (dav)
Yyuv / Cyc
+
data_valid (dav)
UV
+
data_valid (dav)
Xtal ref. clock
10
10
10
10
10
10
10
U
V
Conversion Binary to Signed
Sample Rate Convertor CVBS / Yyc
Sample Rate Convertor Yyuv / Cyc
Sample Rate Convertor U
Sample Rate Convertor V
13
13
13
13
CVBS / Yyc
Yyuv / Cyc
U
V
54 MHz
Clock Generator
Sample Timing from Sync DTO
(Gated) Line Locked Clocks
No control registers for software in this section

Fig 21.Input and sample rate conversion

3.6.2.1 Short description

The input can handle CVBS, Y/C and YUV. To distinguish the Y from Y/C and from YUV the first is called Yyc and the second Yyuv. Cyc and Yyuv share the same data path. The selection which signal is routed to the input is made by the DLINK receiver block.

Figure 22 shows the data streams from DLINK to VIDDEC for different modes.

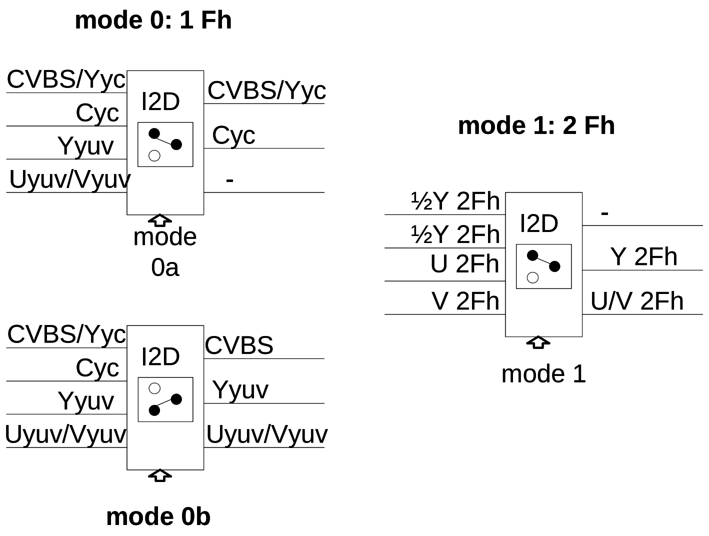


Fig 22.Selection input data streams for VIDDEC in DLINK

For 1 Fh, all input data streams are 10 bit, sampled with 27 MHz derived from a free running system clock. For 2 Fh, the Y and multiplexed UV data stream have a sample rate of 54 MHz.

In the first block the data is converted from unsigned to signed and the UV data stream is demultiplexed in separate U and V streams. The data streams are then fed to a sample rate converter. The samples are converted from the free running system clock domain to a (gated) line locked clock domain. (see PNX8550 for more information)

The number of bits per sample is increased to 13 bits at the output to enable optimal processing in the next AGC block.

3.6.3 AGC

Figure 23 shows the AGC stage block diagram.

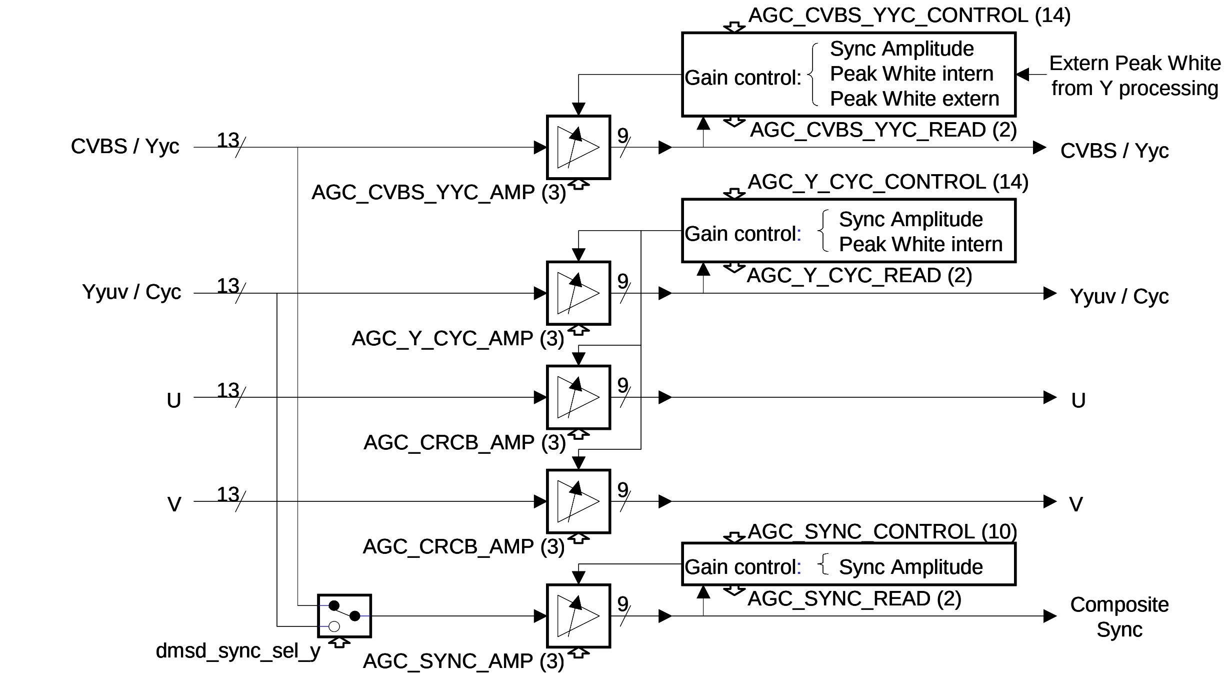


Fig 23.AGC block diagram

3.6.3.1 Brief description

The up converted 13 bit wide signals, coming from the sample rate converter, are passed through an AGC stage to utilize the full 9 bits resolution of the color decoder.

The CVBS/Y, U and V signal path have their own AGC circuit, the Cyc and Yyuv share the AGC circuit because these signals are not available at the same time. Selection between Cyc and Yyuv is done in the DLINK receiver with the bits mode1..0.

At the input of the Sync AGC circuit, it is possible to select between the CVBS/Yyc signal or the Yyuv signal for sync processing.

The AGC stage consists of a general programmable gain stage and a control circuit.

The gain stage is identical for all input signals. It features:

- Programmable black level for the input stage
- Programmable black level for the output stage
- Programmable gain range

For gain stages carrying one type of signal (CVBS/Yyc, U, V, Sync), the settings are fixed. For gain stages carrying different signals (Cyc or Yyuv) the settings should be adapted for the selected signal. There are 3 control circuits, one for CVBS/Yyc, one for Cyc or Yyuv, one for Sync signal, each adapted for the specific signal properties. The control options are:

- Control on Sync amplitude
- Setting target sync amplitude
- Control on Peak White (Only CVBS/Yyc, Cyc or Yyuv)
- Setting target Peak White amplitude (Only CVBS/Yyc, Cyc or Yyuv)
- Minimal gain
- Maximal gain
- Fixed gain (No AGC)
- Hold momentary gain

In addition, the CVBS/Yyc control circuit can also use the (external) Peak White Limiter of the Color Decoder to adapt the gain. The U and V gain stages are slaved to the Cyc/Yyuv gain stage and the Cyc/Yyuv control circuit. At the output the streams are 9 bits wide. The signals are routed to the Color decoder (CVBS/Yyc and Cyc), to the YUV switch (Yyuv, U, V) and to the Sync circuit (CVBS/Yyc or Yyuv).

3.6.3.2 AGC gain stages

Figure 24 shows the AGC gain stages.

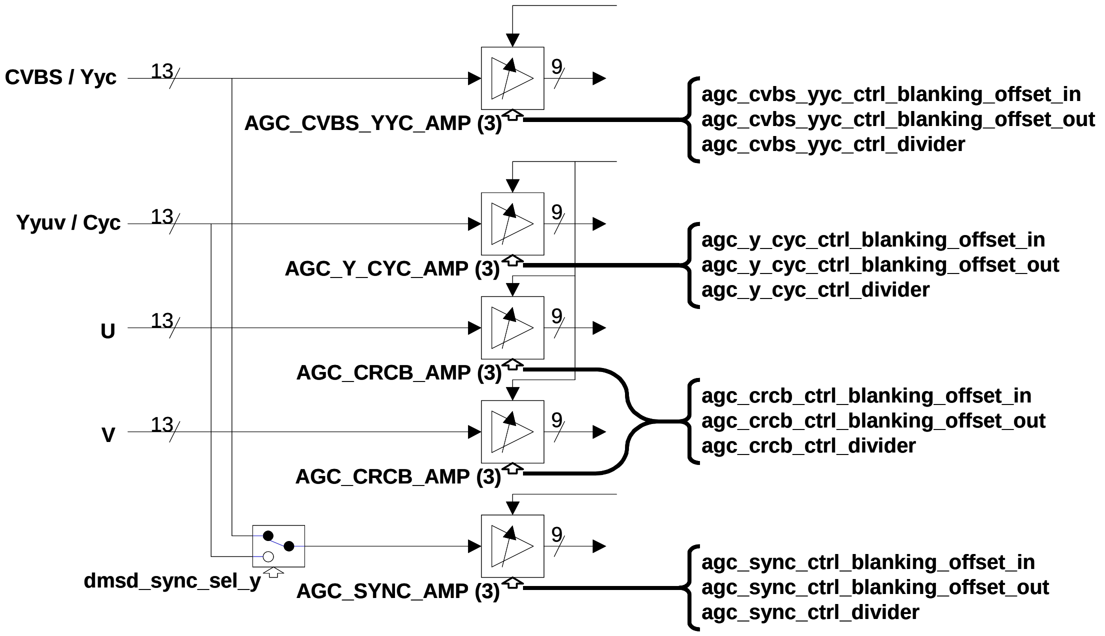

CVBS / Yyc
13
9
AGC_CVBS_YYC_AMP (3)
agc_cvbs_yyc_ctrl_blanking_offset_in
agc_cvbs_yyc_ctrl_blanking_offset_out
agc_cvbs_yyc_ctrl_divider
Yyuv / Cyc
13
9
AGC_Y_CYC_AMP (3)
agc_y_cyc_ctrl_blanking_offset_in
agc_y_cyc_ctrl_blanking_offset_out
agc_y_cyc_ctrl_divider
U
13
9
AGC_CRCB_AMP (3)
agc_crcb_ctrl_blanking_offset_in
agc_crcb_ctrl_blanking_offset_out
agc_crcb_ctrl_divider
V
13
9
AGC_CRCB_AMP (3)
dmsd_sync_sel_y
9
AGC_SYNC_AMP (3)
agc_sync_ctrl_blanking_offset_in
agc_sync_ctrl_blanking_offset_out
agc_sync_ctrl_divider

Fig 24. AGC gain stages

The 10 bits wide data, coming from the DLINK receiver, are up converted to 13 bits by the sample rate converter before entering the AGC gain stage. This implies that the incoming data is multiplied with a factor of 8.

The gain stage is made universal for all channels, and to adapt the stage to the specific input/output requirements, the black level at the input (ctrl_blanking_offset_in), the gain (ctrl_divider) and the black level at the output (ctrl_blanking_offset_out) are programmable.

After the gain stage, the data width is brought back to 9 bits wide to fit the data width of the processing by the color decoder. Beside the data width, the black level and signal format (signed or unsigned) are adapted for the next stage. Figure 25 shows input data versus output data of the AGC stage. The maximum and minimum data value, and the blanking level is indicated.

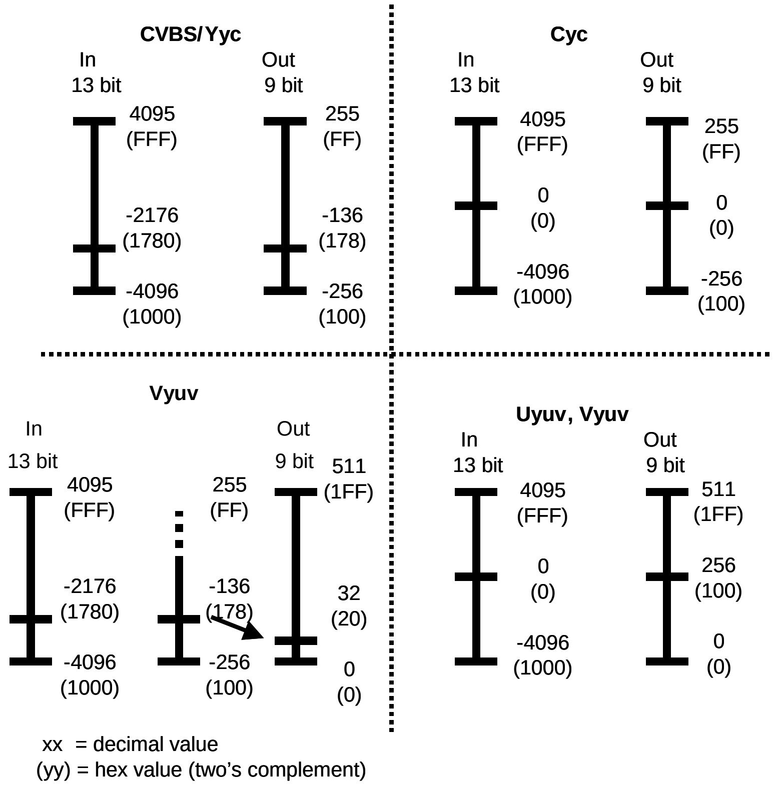


Fig 25.Input format vs. output format of AGC

As indicated, different signals need different conversion. Especially the AGC gain stage in the Cyc / Yyuv path needs attention, because it has to be configured differently depending on the selected signal path.

Table 19: AGC gain stages - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D [1]	R/W
040	2	dmsd_sync_sel_y	Selects sync input 2 Sync from CVBS/Yyc path (1Fh) 3 Sync from Yyuv path (1Fh or 2Fh)	0/x	R/W
084	31..29	agc_cvbs_yyc_divider	Sets amplification range of the AGC block	3	R/W
	24..16	agc_cvbs_yyc_blanking_offset_out	Sets output blanking level	138	R/W

Table 19: AGC gain stages - address 0X7FF9xxx *…continued*

add xxx	Bits	Name	Function	R/D [1]	R/W
	7..0	agc_cvbs_yyc_blanking_offset_in	Sets input blanking level	80	R/W
088	31..29	agc_y_cyc_divider	Sets amplification range of the AGC block	4/3	R/W
	24..16	agc_y_cyc_blanking_offset_out	Sets output blanking level	138/1C0	R/W
	7..0	agc_y_cyc_blanking_offset_in	Sets input blanking level	80/0	R/W
08C	31..29	agc_crcb_divider	Sets amplification range of the AGC block	3	R/W
	24..16	agc_crcb_blanking_offset_out	Sets output blanking level	C0	R/W
	7..0	agc_crcb_blanking_offset_in	Sets input blanking level	0	R/W
080	31..29	agc_sync_divider	Sets amplification range of the AGC block	6	R/W
	24..16	agc_sync_blanking_offset_out	Sets output blanking level	138	R/W
	7..0	agc_sync_blanking_offset_in	Sets input blanking level	80	R/W

[1] In all Bit Description tables the R/D column indicates Reset/Default. If default value is not given it is the same as the reset.

dmsd_sync_sel_y: The sync for the synchronisation block can be taken from the CVBS/Yyc path or the Cyc/Yyuv path. The Cyc/Yyuv path can also be used for 2Fh signals with sync on Yyuv. Note that for 1Fh RGB signals, the sync is often taken from the (accompanying) CVBS signal.

agc_xxx_divider: The divider sets the maximum gain of the stage. The first part of the stage is a pre-amplifier, whose gain can range between a minimum of 0 and a maximum of 1023. The gain can be controlled by the AGC loop or programmed for a fixed gain. See Section 3.6.3.3.

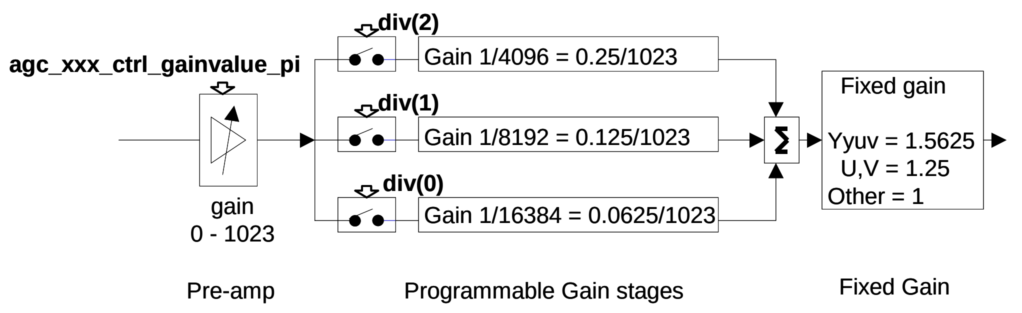


Fig 26. AGC universal programmable gain stage

The programmable part of the gain stage consists of 3 parallel branches (dividers), which can be selected individually or in combination. The amplification of the 3 branches at maximum pre-amplifier gain (1023) is 0.2500, 0.1250 and 0.0625 respectively and are controlled by agc_xxx_divider bits 2..0 in the same order.

The gain range from the 13-bit input to the 9-bit output of the summation block can be set between:

- Minimal gain, agc_xxx_divider = 0 0 1 [binary] is 0.0625
- Maximum gain, agc_xxx_divider = 1 1 1 [binary] is 0.4375

Remark: At least one branch has to be enabled.

This gain is finally multiplied by the fixed gain of the following block. This fixed gain is not equal for all AGC gain stages. In formula:

gain = pre-amp gain (dec) * ((div(2)*0.25 + div(1)*0.125 + div(0)*0.0625) / 1023) * fixed gain

Programming: For the AGC divider stages of CVBS/Yyc, CrCb (Uyuv/VYuv) and Sync, the reset value is the required value. No need to program these values.

Only for the Cyc/Yyuv stage, programming is needed, depending on the selected signal path:

- Yyuv signal (YPrPb or RGB): agc_y_cyc_divider = 3 hex (bit 011)
- Cyc signal: agc_y_cyc_divider = 4 hex (bit 100)

agc_xxx_blanking_offset_in: Looking at the input format, there are two standard blanking levels at the input, 0 (0 hex) and -2176 (1780 hex two's complement). The selection between these two levels is made by the most significant bit of AGC_xxx_blanking_offset_in (bit 7):

- agc_xxx_blanking_offset_in (7) = 0 selects offset 0 (0 hex)
- agc_xxx_blanking_offset_in (7) = 1 selects offset -2176 (1780 hex, two's complement)

The bits agc_xxx_blanking_offset_in (6..0) control the offset in two's complement mode. Resolution per step is 4 hex on the 13 bit wide input data, i.e. one step increases the 13 bit data by 4 hex. Calculating back the resolution to the 10 bits data at the output of the DLINK receiver (or input of the sample rate converter), the resolution is ½ LSB on the 10 bits data, which is fine enough for this purpose.

The offset control is mainly to correct problems of black level offsets in previous stages like the AD conversion. For AVIP, no black level offset correction is needed, so only the most significant bit 7 is used to select between the two standard blanking levels at the input.

Programming: For the AGC stages of CVBS/Yyc, CrCb (Uyuv/VYuv) and Sync, the reset value is the required value. No need to program these values. Only for the Cyc/Yyuv stage, programming is needed, depending on the selected signal path:

- Yyuv signal: agc_y_cyc_blanking_offset_in = 80 hex (bit 7 = 1)
- Cyc signal: agc_y_cyc_blanking_offset_in = 0 hex (bit 7 = 0)

agc_xxx_blanking_offset_out: At the output, there is greater difference in required format. The signals fed to the color decoder are formatted in two's complement, while signals following the YUV path, are unsigned. The blanking level has now to be programmed correctly for each signal path. The offset value is 9 bits wide and has a resolution of 1 LSB/step related to the 9 bits wide output. The range is (in two's complement) -256 to 255. To have enough range for the offset programming, an offset of 64 dec (40 hex) is added. This shifts the range to - 192 to 319 The formula to calculate the needed offset value is:

(decimal): agc_xxx_blanking_level_offset_out(dec) = required blanking level(dec) - 64

(hex) : agc_xxx_blanking_level_offset_out(hex) = required blanking level(hex) - 40

Except for agc_y_cyc_blanking_offset_out, the needed value equals the reset value:

agc_sync_blanking_offset_out = 138 (hex)

agc_cvbs_yyc_blanking_offset_out = 138 (hex)

agc_crcb_blanking_offset_out = C0 (hex)

agc_y_cyc_blanking_offset_out

The gain stage for the (combined) Cyc / Yyuv path contains an extra formatter stage which contains a fixed gain and transforms the output levels of Yyuv to the levels, suited to feed to the YUV switch. For the calculation of the agc_y_cyc_blanking_level_offset_out for use with Yyuv use the levels before the formatter.

The blanking level for Yyuv out is -136 dec (178 hex two's complement).

Using the formula the value for Yyuv becomes -200 dec (138 hex).

The blanking level for Cyc is 256.

The value for use with Cyc becomes 192 dec (C0 hex)

The range of the agc_xxx_blanking_offset_out is larger than needed for the application in AVIP.

It is possible to program the offset so high that the value of the output is higher than the output range of 9 bits. In that case, the value is not clipped but folds over. Because these high settings are not practical for AVIP other than for testing, this is no limitation. In practice the described values should be used.

Programming: For the AGC stages of CVBS/Yyc, CrCb (Uyuv/VYuv) and Sync, the reset value is the required value. No need to program these values.

Only for the Cyc/Yyuv stage, programming is needed, depending on the selected signal path:

- Yyuv signal: agc_y_cyc_blanking_offset_out = -200 dec (138 hex, two's complement)
- Cyc signal: agc_y_cyc_blanking_offset_out = 192 dec (C0 hex)

3.6.3.3 AGC control circuit

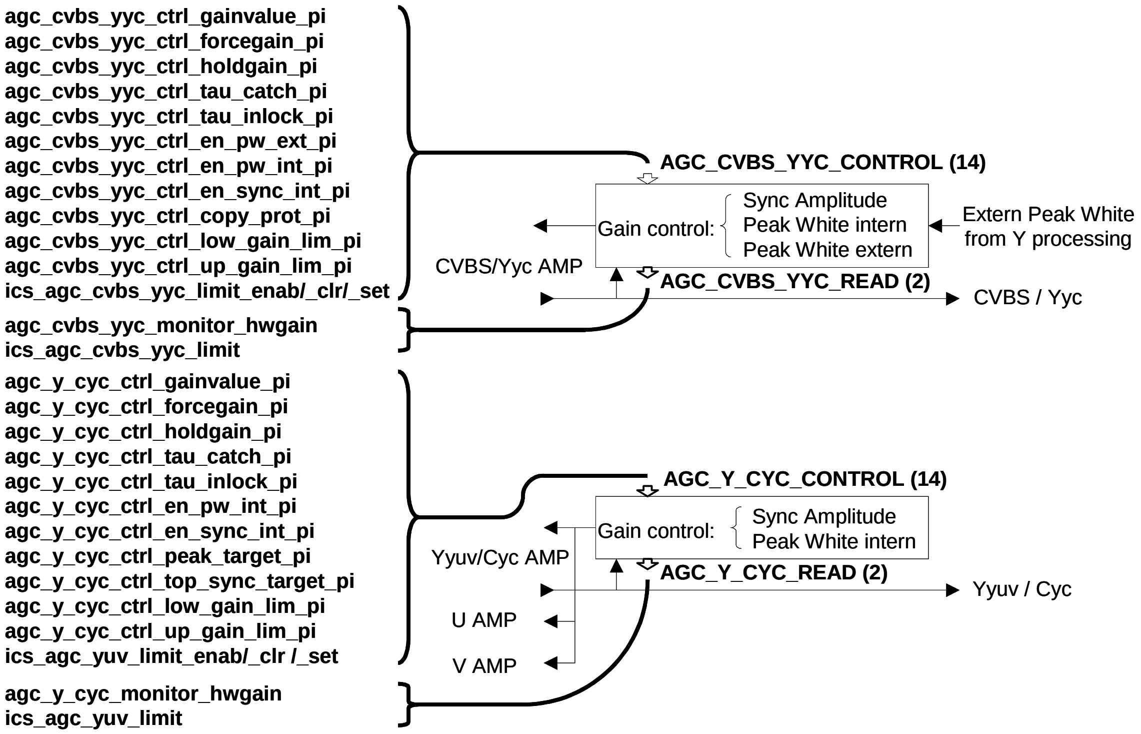

agc_cvbs_yyc_ctrl_gainvalue_pi
agc_cvbs_yyc_ctrl_forcegain_pi
agc_cvbs_yyc_ctrl_holdgain_pi
agc_cvbs_yyc_ctrl_tau_catch_pi
agc_cvbs_yyc_ctrl_tau_inlock_pi
agc_cvbs_yyc_ctrl_en_pw_ext_pi
agc_cvbs_yyc_ctrl_en_pw_int_pi
agc_cvbs_yyc_ctrl_en_sync_int_pi
agc_cvbs_yyc_ctrl_copy_prot_pi
agc_cvbs_yyc_ctrl_low_gain_lim_pi
agc_cvbs_yyc_ctrl_up_gain_lim_pi
ics_agc_cvbs_yyc_limit_enab/_clr/_set
agc_cvbs_yyc_monitor_hwgain
ics_agc_cvbs_yyc_limit
AGC_CVBS_YYC_CONTROL (14)
Gain control:
Sync Amplitude
Peak White intern
Peak White extern
Extern Peak White
from Y processing
CVBS/Yyc AMP
AGC_CVBS_YYC_READ (2)
CVBS / Yyc
agc_y_cyc_ctrl_gainvalue_pi
agc_y_cyc_ctrl_forcegain_pi
agc_y_cyc_ctrl_holdgain_pi
agc_y_cyc_ctrl_tau_catch_pi
agc_y_cyc_ctrl_tau_inlock_pi
agc_y_cyc_ctrl_en_pw_int_pi
agc_y_cyc_ctrl_en_sync_int_pi
agc_y_cyc_ctrl_peak_target_pi
agc_y_cyc_ctrl_top_sync_target_pi
agc_y_cyc_ctrl_low_gain_lim_pi
agc_y_cyc_ctrl_up_gain_lim_pi
ics_agc_yuv_limit_enab/_clr /_set
agc_y_cyc_monitor_hwgain
ics_agc_yuv_limit
AGC_Y_CYC_CONTROL (14)
Gain control:
Sync Amplitude
Peak White intern
Yyuv/Cyc AMP
AGC_Y_CYC_READ (2)
Yyuv / Cyc
U AMP
V AMP

Fig 27.AGC control circuit CVBS/Yyc and Yyuv/Cyc

AGC control circuit for CVBS / Yyc path: The control circuit for the CVBS / Yyc path has a very flexible set-up. It is designed to work in an automatic mode. In this case, the sync amplitude is used to determine the needed gain factor for amplification of the total signal to the nominal level. To cope with signals having compressed sync, a peak white limiter will take care that no clipping occurs. In the CVBS / Yyc control circuit it is possible to use the AGC internal peak white limiter or the "external" peak white limiter of the color decoder.

To improve the behavior for non-standard conditions, the maximum and minimum gain can be programmed to prevent excessive adaptation. It is also possible to set a fixed gain for test purposes. Also the gain of an active loop can be frozen for measuring or testing.

The time constant of the AGC loop can be programmed differently for the situation when there is horizontal lock (usually a fast time constant required) or when there is no horizontal lock (usually slower time constant to prevent pumping). For this purpose, the horizontal lock of the sync circuit is used. For monitoring, an interrupt can be programmed to signal when the programmed gain limits are exceeded.

Table 20: AGC gain control - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
094	9.0	agc_cvbs_yc_ctrl_gainvalue_pi	Gain value when forcegain_pi = 1	1CD	R/W
	10	agc_cvbs_yc_ctrlagc_cvbs_yc_ctrl_forcegain_pi	0 Gain controlled by AGC loop. 1 Fixed gain, determined by gainvalue_pi	0	R/W
	11	agc_cvbs_yc_ctrl_holdgain_pi	0 Normal AGC loop operation1 Freezes the momentary gain of the loop	0	R/W
	18..16	agc_cvbs_yc_ctrl_tau_catch_pi	AGC loop time constant when there is no H-lock 000 Fast AGC time constant. 111 Slow AGC time constant	1	R/W
	22..20	agc_cvbs_yc_ctrl_tau_inlock_pi	AGC loop time constant when there is H-lock000 Fast AGC time constant111 Slow AGC time constant	6	R/W
	23	agc_cvbs_yc_ctrl_pw_ext_pi	Enables the "external" (= outside AGC block) peak white limiter of the multi-standard color decoder to influence the AGC loop. 0 External pk wh lim not enabled 1 External pk wh lim enabled	1	R/W
	24	agc_cvbs_yc_ctrl_pw_int_pi	Enables the internal peak white limiter of the AGC block to influence the AGC loop0 Internal pk wh lim not enabled1 Internal pk wh lim enabled	1	R/W
	25	agc_cvbs_yc_ctrl_sync_int_pi	Enables the sync AGC loop to apply the same multiplication to the CVBS / YYC amplifier as needed to bring the sync amplitude to nominal level 0 Control by sync AGC loop not enabled1 Control by sync AGC loop enabled	1	R/W
	26	agc_cvbs_yc_ctrl_copy_prot_pi	Corrects the nominal sync amplitude to 80% for Macrovision signals. Should be used when Macrovision is detected (dmsd_copro = 1) 0 Normal operation1 Use reduced sync amplitude(80%)	0	R/W
0A0	19..10	agc_cvbs_yyc_ctrl_low_gain_lim_pi	Sets lowest possible gain for the AGC loop	142	R/W
0A4	19..10	agc_cvbs_yyc_ctrl_up_gain_lim_pi	Sets maximum possible gain for the AGC loop	39A	R/W
08	19..10	agc_cvbs_yc_monitor_hwgain	Readout of the momentary gain value of the AGC loop		R
FE0	13	ics_agc_cvbs_yc_gain_limit	Interrupt flag set to 1 when upper or lower gain limit is exceeded. See chapter about interrupts for details		R

agc_cvbs_yc_ctrl_gainvalue_pi / _forcegain_pi

When agc_cvbs_yc_ctrl_forcegain_pi is set to 1, the gain of the AGC stage is determined by the setting of register agc_cvbs_yc_ctrl_gainvalue_pi. The 10 bits of this register determine the amplification of the pre-amplifier, discussed in AGC Gain Stages bit agc_xxx_divider.

The gain can be set from 0 to 1023.

The total amplification from input (13 bits) to output (9 bits) can be calculated using the formula, given in the description of the agc_xxx_divider bit:

gain = pre-amp gain (dec) * ((div(2)*0.25 + div(1)*0.125 + div(0)*0.0625) / 1023) * fixed gain. Taking into account that the fixed gain in the sync path is 1:

gain = gainvalue_pi (dec) * ((div(2)*0.25 + div(1)*0.125 + div(0)*0.0625) / 1023) * 1.

div(w) = agc_xxx_divider(w) where w is the bit number.

To scale the gain from 13 bit to the 10 bits wide output of the DLINK receiver (or input of the sample rate converter), the found gain value has to be multiplied by 8.

agc_cvbs_yc_ctrl_holdgain_pi

Implemented for test purposes to freeze the momentary gain.

agc_cvbs_yc_ctrl_ctrl_tau_catch_inlock_pi

While catching signals, the AGC time constant has to be fast to adapt quickly to varying signals conditions during e.g. search tuning. When in lock, the AGC time constant can better be chosen larger to prevent unstable behavior like pumping on video content. Because it is possible to program different time constants when not in lock (fast) and when in lock (less fast) the loop values do not need reprogramming depending on the signal condition. The values for these registers after a reset should perform correctly.

agc_cvbs_yc_ctrl_pw_int_pi / _pw_ext_pi / sync_int_pi

These bits enable the different control loops.

When the amplitude of the total signal is attenuated proportionally but the relative amplitude ratios are kept correct, the sync amplitude is the most ideal signal part to determine the needed amplification to bring the signal back to nominal level.

Only for compressed sync, the amplification would become too large. The external peak white limiter (present in the color decoder / Y processing part) can be used to reduce the gain below peak white level. Also the internal peak white limiter in the AGC can perform this task, but this peak white limiter clips immediately signals when coming above peak white level, while the external peak white limiter is more sophisticated in behavior. It is advised to enable all three control circuits for the best performance. This is also the status after a reset.

agc_cvbs_yc_ctrl_copy_prot_pi

When Macrovision is present in the signal, the sync amplitude is reduced to 80% of the nominal sync level. Writing this bit to '1' adapts the sync target value to prevent that the CVBS signal is amplified too much (to 125% in stead of 100%) in case of signals containing Macrovision on sync. The sync processing contains a bit, dmsd_copro, which becomes 1 when Macrovision in sync is detected.

This bit has to be monitored on a regular basis and agc_cvbs_yc_ctrl_copy_prot_pi has to follow the value as indicated by dmsd_copro for correct behavior.

agc_cvbs_yyc_ctrl_low_gain_lim_pi / _up_gain_lim_pi

Limits the minimum and maximum gain of the AGC loop to prevent strange behavior under abnormal signal conditions. Default reset values are such that they should not require changing after a reset.

agc_cvbs_yc_monitor_hwgain

Reads out the momentary gain when the control loop is active. For test purposes.

ics_agc_cvbs_yyc_limit / _enab / _clr / _set

It is possible to enable an interrupt when the programmed lowest gain or upper gain is reached.

Programming

All registers bar one have a default value after reset such that no additional programming is required. That register is described below.

agc_cvbs_yc_ctrl_copy_prot_pi

Software has to monitor regularly (or can enable an interrupt to be signalled) the bit dmsd_copro, which indicates whether Macrovision is detected in the sync. The value of agc_cvbs_yc_ctrl_copy_prot_pi has to follow the value of dmsd_copro.

AGC control circuit for Yyuv / Cyc path: The basic control mechanisms for this path are identical to those for the CVBS / Yyc path. However, here the settings need adapting depending on the type of signal:

- Yyuv with sync on Y
- Yyuv, converted from RGB without sync
- Cyc

The required settings are discussed in more detail after the survey of the control bits.

Table 21: AGC control circuit for Yyuv / Cyc path - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
098	9..0	Agc_y_cyc_ctrl_gainvalue_pi	Gain value when forcegain_pi =1	0E8/[1]	R/W
	10	Agc_y_cyc_ctrl_forcegain_pi	0 Gain controlled by AGC loop 1 Fixed gain, determined by gainvalue_pi	0	R/W
	11	Agc_y_cyc_ctrl_holdgain_pi	0 Normal AGC loop operation 1 Freezes the momentary gain of the loop	0	R/W
	18..16	Agc_y_cyc_ctrl_tau_catch_pi	AGC loop time constant when there is no H-lock 000 Fast AGC time constant 111 Slow AGC time constant	1	R/W
	22..20	Agc_y_cyc_ctrl_tau_inlock_pi	AGC loop time constant when there is H-lock 000 Fast AGC time constant 111 Slow AGC time constant	6	R/W
	24	agc_y_cyc_ctrl_enable_pw_int_pi	Enables the internal peak white limiter of the AGC block to influence the AGC loop 0 Internal peak white limiter not enabled 1 Internal peak white limiter enabled	1	R/W
	25	agc_y_cyc_ctrl_enable_sync_int_pi	Enables the sync AGC loop to apply the same multiplication to the CVBS /YYC amplifier as needed to bring the sync amplitude to nominal level 0 Control by sync AGC loop not enabled 1 Control by sync AGC loop enabled	1	R/W
09C	8..0	Agc_y_cyc_ctrl_peak_target_pi	Sets the peak level the AGC loop uses for gain control	1FF/13B	
	24..16	Agc_y_cyc_ctrl_top_sync_target_pi	Sets the top sync level the AGC loop uses for gain ctrl	100/[2]	R/W

Table 21: AGC control circuit for Yyuv / Cyc path - address 0X7FF9xxx *…continued*

add xxx	Bits	Name	Function	R/D	R/W
0A0	29..20	Agc_y_cyc_ctrl_low_gain_lim_pi	Sets lowest possible gain for the AGC loop	11F/0F5	R/W
0A4	29..20	Agc_y_cyc_ctrl_up_gain_lim_pi	Sets maximum possible gain for the AGC loop	332/2BA	R/W
008	29..20	Agc_y_cyc_monitor_hwgain	Readout of the momentary gain value of the AGC loop		R/W
0E0	14	Ics_agc_y_cyc_gain_limit	Interrupt flag set to 1 when upper or lower gain limit is exceeded		R
0E4	14	int_ena_agc_y_cyc_gain_limit	Disables/enables the interrupt generation 0 Disabled 1 Enabled	0	R/W
0E8	14	int_clr_agc_y_cyc_gain_limit	Writing "1" to this position clears the interrupt flag Interrupt flag should be cleared by software after acknowledge of the interrupt.		W
0EC	14	int_set_agc_y_cyc_gain_limit	Writing "1" to this position forces the interrupt flag to 1. For test purposes.		W

[1] 15D (YPrPb), 100 (Y/C)

[2] 100 or 118 when Macrovision present

Most of the bits are identical to the ones, discussed with the CVBS / Yyc path and will not be discussed here again. Note that when YUV processing is selected, the gain of the AGC amplifiers in the UV path is slaved to the amplification of the Yyuv AGC stage.

Differences with the CVBS / Yyc path:

Missing control option

There is no option to use an external peak white limiter, because the Yyuv is not routed through the color decoder part. So for peak white control, only the internal peak white clipper can be used.

Agc_y_cyc_ctrl_peak_target_pi / _top_sync_target_pi

Because the Yyuv / Cyc path can handle a variety of signals, the peak white level and the top sync level can be programmed. These values are used when the agc_y_cyc_ctrl_enable_pw_int_pi / _sync_int_pi enable AGC control by the (internal) peak white limiter or sync amplitude.

When the gain of the Yyuv / Cyc stage is forced to manual (agc_y_cyc_ctrl_forcegain_pi = 1) and the setting of the gain value (agc_y_cyc_ctrl_gainvalue_pi) is set too high (3FF), fold over may occur. Because this is not a practical situation and it is advised to use automatic settings, this is not a problem in practice.

Settings for different signal streams

The subsequent paragraphs describe the control register settings for the three possible input signals:

1. YPrPb
2. RGB with sync on CVBS
3. Cyc

To start with the first two, the levels are given of a Y signal before and after the AGC stage. These levels are needed to calculate amplification factors.

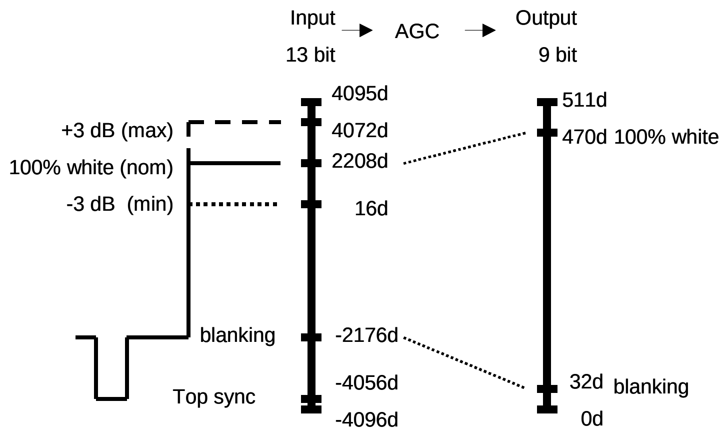


Fig 28.Levels before and after the AGC in the Yyuv path

1. YPrPb

The setting for YPrPb are comparable with the settings for CVBS/Yyc. The signal contains a nominal sync which can be used for gain control and the peak white limiter can be enabled to limit the gain for compressed sync. The top sync target should be set for a nominal sync, which is -256d (100 hex two's complement).

These levels are explained in Section "AGC control circuit for the sync path". The peak target can be set for the maximum level at the 9 bits output, which is 511d (1FF hex).

Table 22: AGC Yyuv / Cyc for YPrPb signals

Name	Setting for YPrPb	Reset/Default
Agc_y_cyc_ctrl_gainvalue_pi	15D	0E8/[1]
Agc_y_cyc_ctrl_forcegain_pi	0	0
Agc_y_cyc_ctrl_holdgain_pi	0	0
Agc_y_cyc_ctrl_tau_catch_pi	1	1
Agc_y_cyc_ctrl_tau_inlock_pi	6	6
agc_y_cyc_ctrl_enable_pw_int_pi	1	1
agc_y_cyc_ctrl_enable_sync_int_pi	1	1
Agc_y_cyc_ctrl_peak_target_pi	13B	1FF/13B
Agc_y_cyc_ctrl_top_sync_target_pi	100/118	100/[2]
Agc_y_cyc_ctrl_low_gain_lim_pi	0F5	11F/0F5
Agc_y_cyc_ctrl_up_gain_lim_pi	2BA	332/2BA
Agc_y_cyc_divider	3	4

[1] 15D (YPrPb), 100 (Y/C)

[2] 100 or 118 when Macrovision present

As can be seen, the reset values are not correct for a number of control bits for processing of YPrPb signals. This is related to an adaptation of the AGC block design without adapting the reset values. The bits, needing another default value, are:

agc_y_cyc_ctrl_peak_target_pi = 13B (Reset: 1FF)

agc_y_cyc_ctrl_top_sync_target_pi = 100 / 118

agc_y_cyc_ctrl_low_gain_lim_pi = 0F5 (Reset: 11F)

agc_y_cyc_ctrl_up_gain_lim_pi = 2BA (Reset: 332)

The value of the bits below are different for YPrPb / RGB processing and Cyc processing. For YPrPb / RGB the value becomes:

agc_y_cyc_ctrl_gainvalue_pi = 15D (Reset: 0E8)

agc_y_cyc_divider = 3 (Reset: 4).

Macrovision in sync of YPrPb signals.

Also YPrPb signals can contain Macrovision in the sync. The presence of Macrovision in sync can be read out by the detection bit dmsd_copro for 1Fh signals and copro_2fh for 2Fh signals, just like with CVBS / Yyc, provided that for the sync processing the Yyuv signal path is selected by setting dmsd-sync_sel_y to 1.

Only the Yyuv / Cyc control block does not contain a special bit like in the CVBS / Yyc control block (agc_cvbs_yc_ctrl_copy_prot_pi) which compensates for the 80% sync levels.

However, the agc_y_cyc_ctrl_top_sync_target_pi can be used to adapt the target level to compensate for the 80% sync levels with Macrovision. For calculation of the value, see Figure 29.

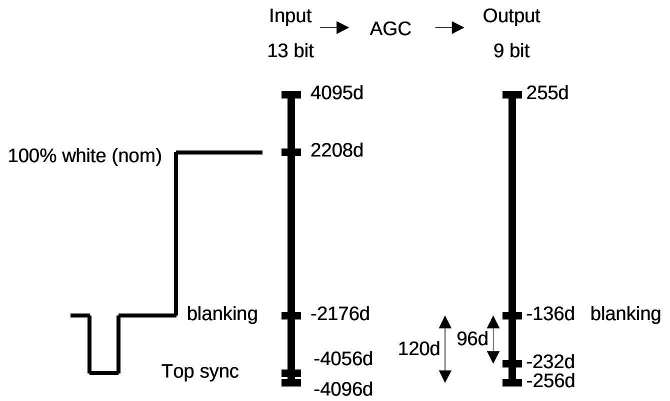


Fig 29.Levels before and after the AGC in the sync path

A nominal sync at the AGC output has an amplitude of 120d, while the blanking level is at -136d. The value for agc_y_cyc_ctrl_top_sync_target_pi is the level of the sync bottom. So for a nominal sync this is -256 (100 hex two's complement) For 80% sync level, the amplitude becomes 0.8 * 120 = 96. Taking the blanking level as reference, the sync bottom becomes -232d, which is 118 in hex two's complement. So also for the Yyuv path it is important to check for Macrovision and adapt the setting of agc_y_cyc_ctrl_top_sync_target_pi accordingly.

Programming

Take care to program the following registers with another value after reset:

agc_y_cyc_ctrl_peak_target_pi = 13B (Reset: 1FF)

agc_y_cyc_ctrl_top_sync_target_pi = 100 / 118 (See below Macrovision)

agc_y_cyc_ctrl_low_gain_lim_pi = 0F5 (Reset: 11F)

agc_y_cyc_ctrl_up_gain_lim_pi = 2BA (Reset: 332)

For YPrPb processing, program:

agc_y_cyc_ctrl_gainvalue_pi = 15D (Reset: 0E8)

agc_y_cyc_ctrl_top_sync_target_pi has to be adapted according the status of

dmsd_copro (for 1Fh) or copro_2fh (for 2Fh), indicating Macrovision on sync:

dmsd_copro / copro_2fh = 0 -> agc_y_cyc_ctrl_top_sync_target_pi = 100 hex

dmsd_copro / copro_2fh = 1 -> agc_y_cyc_ctrl_top_sync_target_pi = 118 hex

2. RGB with sync on CVBS

The signal levels for RGB input signals are the same as for Yyuv, because the RGB signals are converted to YUV format in the PNX3000 before further processing. Only the RGB signals do not contain sync, which excludes the possibility to use the sync for AGC control.

The performance is optimized for nominal signals. By setting the maximum gain (agc_y_cyc_ctrl_up_gain) so that nominal signals are remaining just below the peak white target, nominal signals are linear processed. Too small signals will lead to smaller levels at the RGB outputs, but for most scenes this is compensated by the beam current limiter. For too large signals, the peak white limiter is enabled to reduce the gain when needed. The minimum gain can be set to accommodate signals up to +3 dB.

Describing the gain stages, for the bits agc_xxx_divider a formula is given which can be used to calculate the gain in the Yyuv stage:

gain = pre-amp gain (dec) * ((div(2)*0.25 + div(1)*0.125 + div(0)*0.0625) / 1023)*fixed gain

For the Yyuv /Cyc stage in Yyuv mode, div(1) and div(0) are programmed and the fixed gain is 1.5625, so the formula becomes:

gain Yyuv / Cyc = pre-amp gain (dec) * ((0.125 + 0.0625) / 1023) * 1.5625

For nominal signal, the gain is the black to white pk-pk level at the output divided by the black to white pk-pk level at the input (see Figure 29) .

Gain = (470 - 32) / (2208 + 2176) = 0.1 -> The pre-amp gain becomes 349 and the setting of agc_y_cyc_ctrl_up_gain becomes 349 dec (15D hex).

The setting for agc_y_cyc_ctrl_low_gain should be 3dB lower, 247 dec (F7 hex). This last value is almost identical to the default value (not the reset value) of agc_y_cyc_ctrl_low_gain (F5 hex) and needs no change.

The table with all settings for RGB with sync on CVBS is given below:

Table 23: AGC Yyuv / Cyc for RGB signals with sync on CVBS

Name	Setting for YPrPb	Reset/Default
Agc_y_cyc_ctrl_gainvalue_pi	15D	0E8/*
Agc_y_cyc_ctrl_forcegain_pi	0	0
Agc_y_cyc_ctrl_holdgain_pi	0	0
Agc_y_cyc_ctrl_tau_catch_pi	1	1
Agc_y_cyc_ctrl_tau_inlock_pi	6	6
agc_y_cyc_ctrl_enable_pw_int_pi	1	1
agc_y_cyc_ctrl_enable_sync_int_pi	0	1
Agc_y_cyc_ctrl_peak_target_pi	13B	1FF/13B
Agc_y_cyc_ctrl_top_sync_target_pi	100	100/1)
Agc_y_cyc_ctrl_low_gain_lim_pi	0F5	11F/0F5
Agc_y_cyc_ctrl_up_gain_lim_pi	15D	332/2BA
Agc_y_cyc_divider	3	4

Programming

The agc_y_cyc_ctrl_enable_sync_int_pi has to be switched off and the agc_y_cyc_ctrl_up_gain has to be programmed to another value. Note also that the setting for agc_y_cyc_divider has to be set different from the reset value.

agc_y_cyc_ctrl_sync_int_pi = 0

agc_y_cyc_ctrl_up_gain = 15D hex

agc_y_cyc_divider = 3

Remaining registers, to be programmed different from reset value after reset:

agc_y_cyc_ctrl_peak_target_pi = 13B (Reset: 1FF)

agc_y_cyc_ctrl_top_sync_target_pi = 100 / 118 (When Macrovision present)

agc_y_cyc_ctrl_low_gain_lim_pi = 0F5 (Reset: 11F).

3. Cyc

Processing of Cyc is different from the other discussed signals. The color decoder has its own AGC to adapt the gain of the color carrier, using the burst as reference. So for the Cyc signal, a fixed gain can be set. The gain should be chosen such that for nominal Cyc levels the input at the color decoder is the same as when nominal CVBS signals are fed to the color decoder. Taking into account the whole path from PNX3000 to input of AGC, an attenuation from input to output of the AGC block from 0.0625 (1/16) is required. Using the formula for the gain of the Yyuv / Cyc stage the required pre-amp gain value can be calculated:

gain = pre-amp gain (dec) * ((div(2)*0.25 + div(1)*0.125 + div(0)*0.0625) / 1023)*fixed gain

For the Yyuv / Cyc stage in Cyc mode, only div(2) should be programmed 1. In Cyc mode, the fixed gain is also 1. The formula becomes:

gain Yyuv / Cyc = pre-amp gain (dec) * 0.25 / 1023 .

So: pre-amp gain = 0.0625 * 1023 / 0.25 = 256d

The pre-amp gain has to be set to 256 dec (100 hex).

The table with settings for Cyc becomes:

Table 24: AGC Yyuv / Cyc for Cyc signals

Name	Setting for YPrPb	Reset/Default
Agc_y_cyc_ctrl_gainvalue_pi	100	0E8/*
Agc_y_cyc_ctrl_forcegain_pi	1	0
Agc_y_cyc_ctrl_holdgain_pi	0	0
Agc_y_cyc_ctrl_tau_catch_pi	1	1
Agc_y_cyc_ctrl_tau_inlock_pi	6	6
agc_y_cyc_ctrl_enable_pw_int_pi	1	1
agc_y_cyc_ctrl_enable_sync_int_pi	1	1
Agc_y_cyc_ctrl_peak_target_pi	13B	1FF/13B
Agc_y_cyc_ctrl_top_sync_target_pi	100	100/1)
Agc_y_cyc_ctrl_low_gain_lim_pi	0F5	11F/0F5
Agc_y_cyc_ctrl_up_gain_lim_pi	2BA	332/2BA
Agc_y_cyc_divider	4	4

By forcing the gain (programming agc_y_cyc_ctrl_gainvalue_pi and forcing fixed gain setting with agc_y_cyc_ctrl_forcegain_pi all other settings can remain as the default settings for the YPrPb and RGB mode. In fact, they are ‘don't cares’ in this mode.

Programming

Only registers to be set different from default value for YPrPb and RGB:

agc_y_cyc_ctrl_gainvalue_pi = 100 hex

agc_y_cyc_ctrl_forcegain_pi = 1

agc_y_cyc_divider = 4

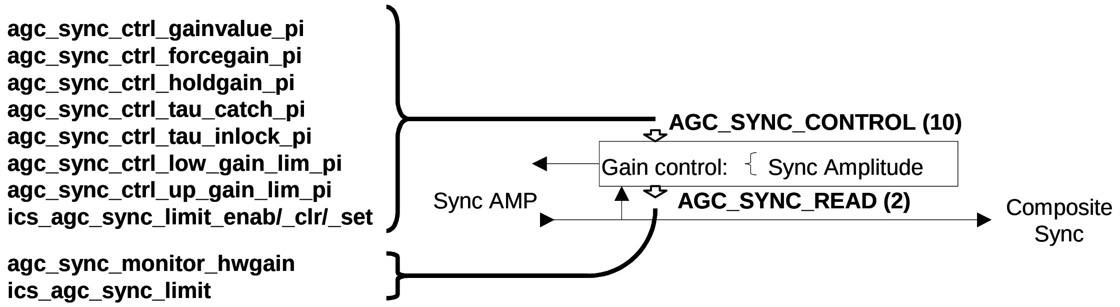

agc_sync_ctrl_gainvalue_pi
agc_sync_ctrl_forcegain_pi
agc_sync_ctrl_holdgain_pi
agc_sync_ctrl_tau_catch_pi
agc_sync_ctrl_tau_inlock_pi
agc_sync_ctrl_low_gain_lim_pi
agc_sync_ctrl_up_gain_lim_pi
ics_agc_sync_limit_enab/_clr/_set
agc_sync_monitor_hwgain
ics_agc_sync_limit
AGC_SYNC_CONTROL (10)
Gain control: { Sync Amplitude
Sync AMP
AGC_SYNC_READ (2)
Composite Sync

Fig 30.AGC control circuit sync

AGC control circuit for the sync path: The AGC for the sync amplifies the sync portion of the CVBS / Yyc or Yyuv signal to a level, suitable for the sync slicer of the synchronisation block. The sync amplitude is less critical (as long as it is large enough, the sync circuit will work correctly), however the control options are limited.

Table 25: AGC control circuit for the sync path - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
090	9..0	agc_sync_ctrl_gainvalue_pi	Gain value when forcegain_pi = 1	0E8	R/W
	10	agc_sync_ctrl_forcegain_pi	0 Gain controlled by AGC loop 1 Fixed gain, determined by gainvalue_pi	0	R/W
	11	agc_sync_ctrl_holdgain_pi	0 Normal AGC loop operation 1 Freezes the momentary gain of the loop	0	R/W
	18..16	agc_sync_ctrl_tau_catch_pi	AGC loop time constant when there is no H-lock 000 Fast AGC time constant 111 Slow AGC time constant	1	R/W
	22..20	agc_sync_ctrl_tau_inlock_pi	AGC loop time constant when there is no H-lock 000 Fast AGC time constant 111 Slow AGC time constant	6	R/W
0A0	9..0	agc_sync_ctrl_low_gain_lim_pi	Sets lowest possible gain for the AGC loop	AB	R/W
0A4	9..0	agc_sync_ctrl_up_gain_lim_pi	Sets maximum possible gain for the AGC loop	2DB	R/W
008	9..0	agc_sync_monitor_hwgain	Readout of the momentary gain value of the AGC loop		R
0E0	12	ics_agc_sync_gain_limit	Interrupt flag set to 1 when upper or lower gain limit is exceeded		R

The working of all bits has been explained already in the CVBS / Yyc path, and will not be described again here. The sync AGC loop is always put in automatic mode, there is no need to change the settings for different input signals. The reset values are correct and do not need to be changed.

For understanding and to enable calculation of the top sync target value in the Yyuv / Cyc path, the levels are given before and after the sync AGC in Figure 31.

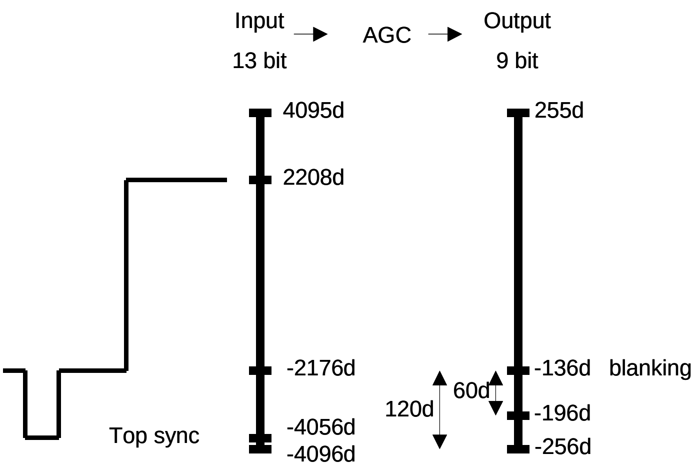


Fig 31.Levels before and after sync AGC

For the sync AGC, the part below blanking level of CVBS / Yyc or Yyuv is used.

At the input, the blanking level is -2176 dec, at the output the level is -136 dec. (see also Section 3.6.3.2).

A nominal sync at the input has a pk-pk amplitude of 120d. The sync input has its own gain stage and can handle levels from 60d to 120d. The settings in the AGC block are chosen such, that for each sync input including compressed sync these pk-pk levels are reached.

When in the Yyuv / Cyc AGC stage the value agc_y_cyc_ctrl_top_sync_target_pi is programmed, the value should be set between -256d (100 hex two's complement) to -196d (13C hex two's complement).to guarantee a pk-pk sync amplitude between 60d and 120d. For calculations, the knowledge that a nominal sync has just a pk-pk amplitude of 120d can be used.

Programming

As indicated, The sync AGC loop is always put in automatic mode, there is no need to change the settings for different input signals. The reset values are correct and need not to be changed.

3.6.4 Digital Multi Standard Decoder (DMSD)

The CVBS or Yyc signal first enters an adjustable line delay stage. This stage compensates for the line delay when the combfilter is used. Next, the color information is removed from the CVBS signal by subtracting the remodulated color carrier from the U,V demodulation. The Y delay compensates for time differences of the Y signal and the demodulated U and V signals.

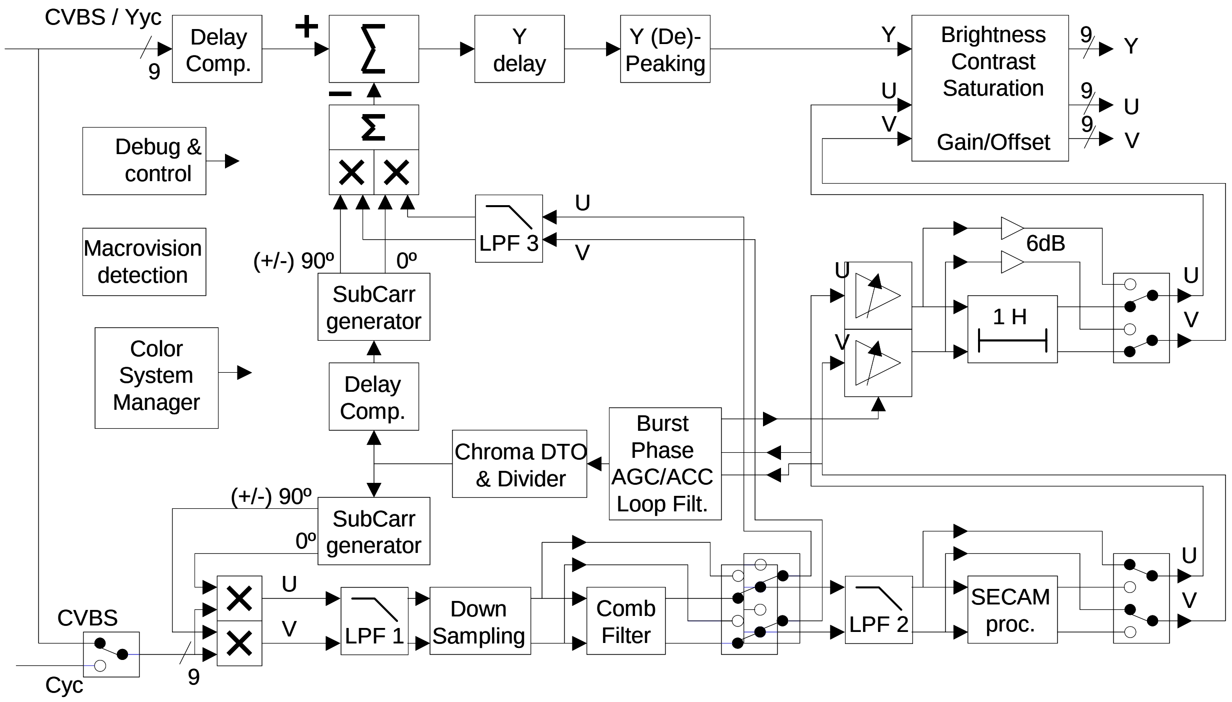


Fig 32.Block diagram Digital Multi Standard Decoder (DMSD)

The (de)peaking circuit not only controls the (de)peaking of the Y signal but also contains some traps for filtering unwanted residual components from Y. The color decoder input can select either the CVBS signal or the C signal as input. The color carrier is demodulated and the U, V signals are low pass filtered and down sampled to match the U, V required bandwidth.

For better luminance and chrominance separation a 2D combfilter can be used for PAL (4 lines) and NTSC (2 lines). It can also be bypassed. After the combfilter selection switch the U and V signals split. One branch passes the programmable Low Pass Filter 3. After the filter the U and V signals are remodulated on the regenerated color carrier. This color carrier is then subtracted from the CVBS signal to obtain the luminance information (Y).

The other branch passes another programmable Low Pass Filter. In case of SECAM, the signals pass a SECAM decoder block. The U and V signals split again. One stream goes to the control block, which contains the color phase detector, the loop filter and auxiliary functions like Hue, Automatic Gain control (color AGC) and Automatic Color Control (ACC). The loop filter output controls the Chroma Discrete Time Oscillator (DTO) which controls two Sub carrier generators, one for demodulation of the incoming color carrier, one (including a delay compensation for exact matching) for the remodulation of U and V to remove the color carrier information from the CVBS.

The other stream passes an adjustable gain amplifier, which is controlled by the AGC and ACC from the above described control block. A delay line section, needed for PAL and SECAM completes the U, V processing. The delay line can be bypassed, in which case 6 dB gain is added to the U,V to match the output levels of the delay line section.

The processed Y, U and V then enter the control stage in which brightness, contrast and saturation can be adjusted. For the AVIP, these controls have a fixed setting, because these items are controlled in other processing blocks. Further some compensation in gain and offset can be made to compensate for errors in the processing. These need not to be used in AVIP.

A color system manager block, Macrovision detection block and a Debug and control block complete the DMSD.

3.6.4.1 Y processing

The CVBS / Yyc signal first passes a line delay compensation (dmsd_ldel). This compensates for the two (PAL) or one (NTSC) line delay in U and V when the comb filter is used for chrominance and luminance. An identical delay compensation is used for the Sub carrier generator. The color information is removed from the CVBS by subtracting the remodulated U and V from the decoder from the CVBS. The Y signal at the output passes a delay section, which can be used when the transitions in Y and U / V are not coincident. The (de)peaking section not only adapts the peaking in the Y channel but also controls a number of traps to remove unwanted residual components.

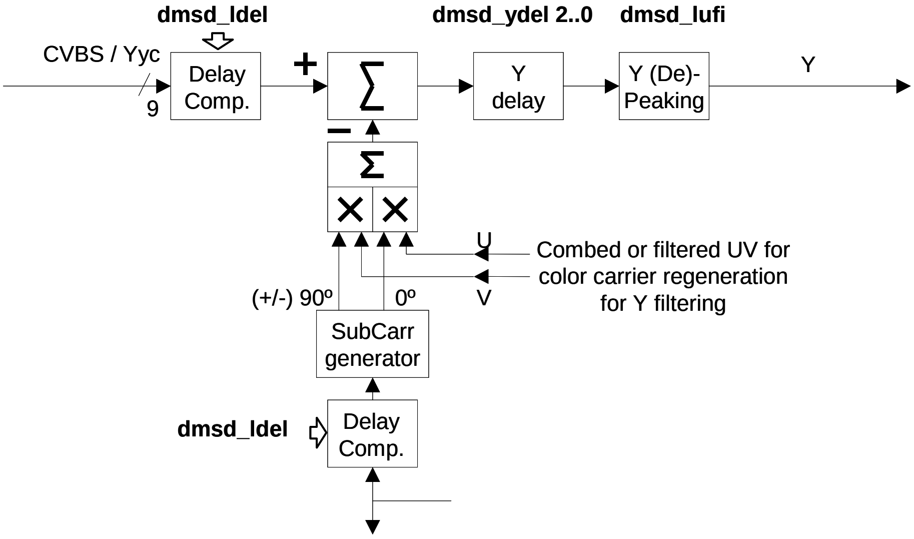


Fig 33. Y Processing

Table 26: Y processing - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
190	14	dmsd_ldel	Extra number of lines delay after vertical sync in NON-combfilter mode 0 No lines delay (recommended) 1 SECAM: No lines delay, NTSC: 1 line delay, PAL: 2 lines delay	0/x	R/W
	13..11	dmsd_ydel	Luminance delay with respect to chroma	0	R/W
	10..7	dmsd_lufi	Luminance peaking 0000 Flat (recommended) 0001 Peaking 8.0 dB at 4.1 MHz 0010 Peaking 6.8 dB at 4.1 MHz 0011 Peaking 5.1 dB at 4.1 MHz 0100 Peaking 4.1 dB at 4.1 MHz 0101 Peaking 3.0 dB at 4.1 MHz 0110 Peaking 2.3 dB at 4.1 MHz 0111 Peaking 1.6 dB at 4.1 MHz 1000 LPF -2 dB at 4.1 MHz 1001 LPF -3 dB at 4.1 MHz 1010 LPF -3 dB at 3.3 MHz, -4 dB at 4.1 MHz 1011 LPF -3 dB at 2.6 MHz, -8 dB at 4.1 MHz 1100 LPF -3 dB at 2.4 MHz, -14 dB at 4.1 MHz 1101 LPF -3 dB at 2.2 MHz, notch at 3.4 MHz 1110 LPF -3 dB at 1.9 MHz, notch at 3.0 MHz 1101 LPF -3 dB at 1.7 MHz, notch at 2.5 MHz	0	

dmsd_ldel: The bit controls the number of lines delay when the combfilter is switched off.

In combfilter mode, the signals of PAL are 2 lines delayed and the signals for NTSC 1 line. To have the same delay when combfilter is switched on or off, for PAL and NTSC this bit should be set to 1. For SECAM it is required that this bit is set to 0 (note that for SECAM no comb filtering is possible). So advised setting is:

- Set 1 when PAL or NTSC color system is detected.
- Set 0 when SECAM is detected (for SECAM this bit has to be 0, it is not allowed to be 1).

dmsd_ydel: Controls the delay of Y with respect to chroma (U and V). When transitions of luminance (Y) and Chrominance (U and V) are not at the same horizontal position, this register can delay the Y until the transitions fit. Depending on the color system and the combfilter setting, the delay has to be adapted.

dmsd_lufi: Because peaking is done in another block, no peaking should be applied here. However, the peaking is also used to compensate for the Y trap when no combfilter is used. Therefore peaking has to be applied when no combfilter is used. The advised settings are pending on the combfilter setting and the color system:

- 3 when the combfilter is switched off and PAL is detected0 when the combfilter is active (Only possible for PAL and NTSC).

- 0 when the combfilter is active (only possible for PAL and NTSC).
- 6 when the combfilter is switched off and NTSC is detected.
- 11 when SECAM is detected. (No combfilter possible and no Y/C available for SECAM).

Programming: dmsd_ldel needs to be set according to the found color system:

- 1 when PAL or NTSC is detected.
- 0 when SECAM is detected.

dmsd_ydel has to be set according to the found color system or is fixed.

dmsd_lufi has to be set according to the activation of the combfilter and the color system.

- 0 when the combfilter is active (only possible for PAL and NTSC).
- 3 when the combfilter is switched off and PAL is detected.
- 6 when the combfilter is switched off and NTSC is detected.
- 11 when SECAM is detected.

3.6.4.2 Demodulator, filtering (combfilter) and SECAM decoder

A switch selects between the incoming CVBS signal or Cyc signal (**chr_inp_del**). The selected signal is then demodulated. The sub carrier generator for demodulation is controlled by the color PLL (see Figure 34).

Demodulator

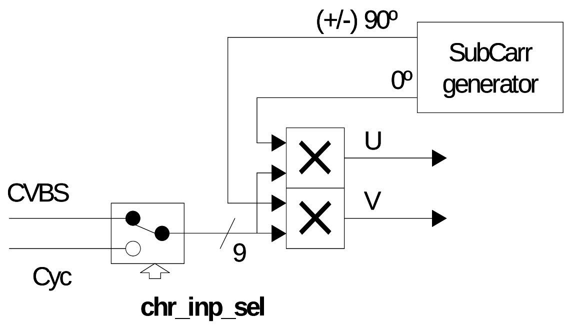


Fig 34.Demodulator

Table 27: Demodulator - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
040	3	chr_inp_del	Selects CVBS or Cyc for the color decoder: 0 CVBS 1 Cyc	0/x	R/W

Programming

CVBS/YC detection

Detection whether a CVBS signal or Y/C signal is connected, when the CVBS path and Y/C path are shared, can be done in three ways:

1. Use two menu items for the combined CVBS/YC connector, one configured for CVBS and one configured for YC. The user can decide whether the picture has color or not and make the appropriate selection.
2. Use a mechanical switch to indicate whether a cable is connected to the CVBS input or Y/C input (only possible when connectors are cinch for CVBS and 4-pin mini-din for Y/C). Software can readout the pin status via an I/O port and configure the correct settings.
3. Automatic detection via software. An algorithm is described in Section “CVBS or Y/C input selection”.

Though reliable detection is possible in this way, the time needed to guarantee a reliable detection can run up to 2 seconds after selecting the input.

Filtering: The demodulated U and V pass through a programmable Low Pass Filter 1. The selected bandwidth of this filter determines the bandwidth of the U and V signals. A high U,V bandwidth will result after remodulation and subtraction of the chroma from the CVBS signal in a lower Luminance bandwidth. The U,V signals are then down sampled to bring the sampling rate in line with the U,V bandwidth.

The down sampled U and V signals can go through a combfilter section or bypass the combfilter. The 2D combfilter contains a number of registers to control the performance. After the combfilter a switch section selects whether the non-combed or combed U and V signals are used for further processing. Note that for Luma processing and Chroma processing the selection can be made independently. In practice the selection should be synchronized for Chroma and Luma of course.

For Y/C signals it is possible to bypass the filtering completely. The U,V signals after the combfilter selection switch for the Chroma path pass the programmable Low Pass Filter 2. The selected bandwidth determines the final U,V bandwidth for further processing.

The U,V signals after the combfilter selection switch for the Luma path are fed to the programmable Low Pass Filter 3. The selected bandwidth determines the resulting notch width in the Luma path after the remodulation and subtraction of the color information from the CVBS.

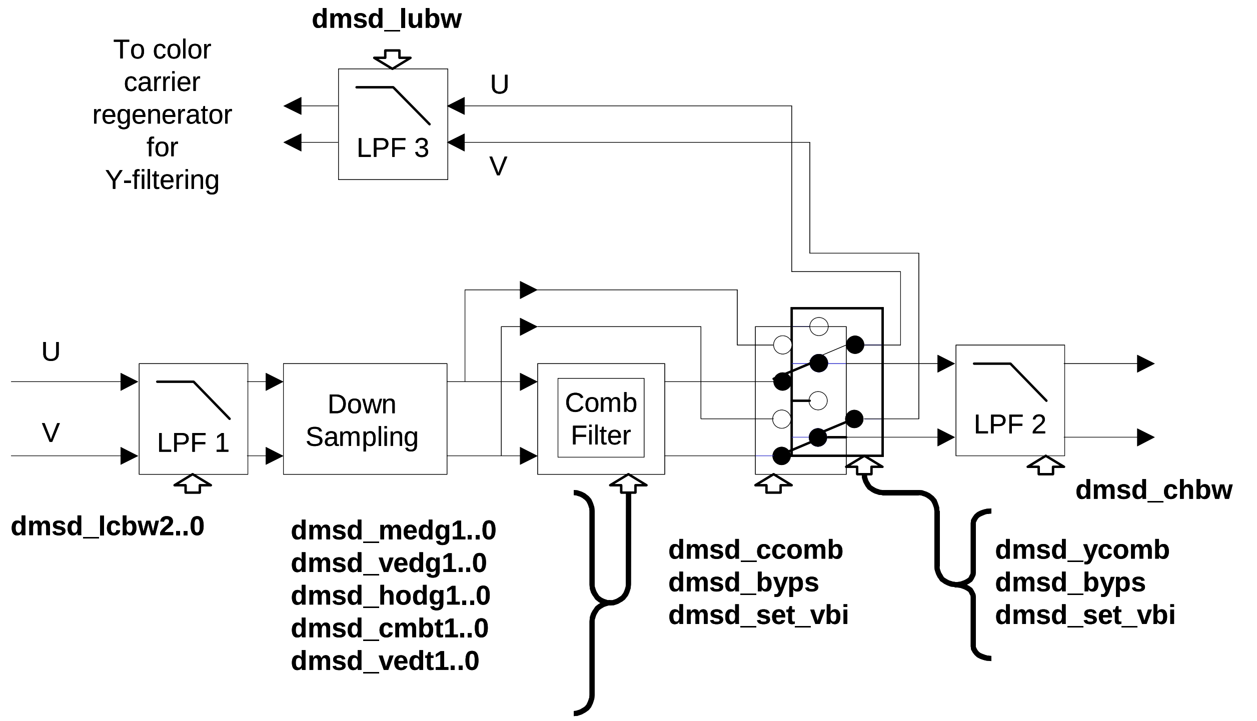


Fig 35.Demodulator and Filter

Table 28: Filters - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
190	2..0	dmsd_lcbw	Luminance bandwidth versus Chroma bandwidth 000 Highest Luma bandw / Lowest Chroma bandw 111 Lowest Luma bandw / Highest Chroma bandw	6/x	R/W
	17..16	dmsd_medg	Comb median filter gain 00 Highest Luma bandwidth at high color saturation 10 Recommended setting 11 Lowest Luma bandwidth at high color saturation	2/x	R/W
	19..18	dmsd_vedg	Comb vertical difference gain 00 Highest Luma bandwidth at vertical transients 10 Recommended setting 11 Lowest Luma bandwidth at vertical transients	2/x	R/W
	21..20	dmsd_hodg	Comb horizontal difference gain 00 Highest Luma bandwidth at horizontal transients 10 Recommended setting 11 Lowest Luma bandwidth at horizontal transients	2/x	R/W
	23..22	dmsd_cmbt	Comb amplitude threshold to adjust the comb strength for signals with small chroma content 00 Lowest comb strength (High threshold) 01 Recommended setting 11 Highest comb strength (Low threshold)	1/x	R/W

Table 28: Filters - address 0X7FF9xxx *…continued*

add xxx	Bits	Name	Function	R/D	R/W
	25..24	dmsd_vedt	Comb vertical difference threshold to adjust the comb strength for signals with large vertical chroma difference 00 Highest strength (High threshold) 01 Recommended setting 11 Lowest strength (Low threshold)	1/x	R/W
	6	dmsd_ccomb	Disable / Enable combfilter in Chroma path 0 Disable combfilter in Chroma path 1 Enable combfilter in Chroma path	1/x	R/W
	5	dmsd_ycomb	Disable / Enable combfilter in Luma path 0 Disable combfilter in Luma path 1 Enable combfilter in Luma path	1/x	R/W
	15	dmsd_byps	Bypass chroma trap / YComb 0 Chroma trap / YComb active 1 Chroma trap / YComb bypassed (for Y/C mode)	0/[1]	R/W
198	20	dmsd_set_vbi	Bypass Luma and Chroma filtering during Vertical Blanking Interval (VBI). Only intended for test purposes 0 No bypass during VBI (recommended) 1 Bypass during VBI	0	R/W
190	3	dmsd_chbw	Select Chroma bandwidth 0 Small, related to setting of dmsd_lcbw 1 Wide, related to setting of dmsd_lcbw	0/x	R/W
	4	dmsd_lubw	Select Luminance bandwidth 0 Narrow Chroma notch -> Maximum Luma bandw 1 Wide Chroma notch -> Less Luma bandw	0/x	R/W

[1] Should be set to 1 for Y/C mode, set to 0 for all other modes

dmsd_lcbw

Determines the balance between Luma and Chroma bandwidth. A high U, V (Chroma) bandwidth will result in a lower Luma bandwidth after subtraction of the remodulated color carrier of the CVBS signal. Recommended setting is 6, which is also the reset value.

2D comb filter settings

The reset value of the combfilter settings is equal to the most optimal settings from design point of view. Depending on user preference, it is possible to select another balance, improving one parameter at the cost of another. The bits to control the performance are:

dmsd_medg, dmsd_vedg, dmsd_hodg, dmsd_cmbt, dmsd_vedt. Table 28 briefly describes which parameter the bits control.

dmsd_ccomb, dmsd_ycomb

Enables the combfilter function in the Chroma and/or Luma path. Though the combfilter function can be selected independently for the Chroma and Luma path, in practice both selections should be synchronized for correct result.

dmsd_byps

Bypasses the combfilter and normal filters in both Chroma and Luma path. Should be set to 1 when Y/C input signal is selected and no filtering in Chroma or Luma path is needed.

dmsd_set_vbi

Only intended for test purposes. Leave at default value.

dmsd_chbw

Selects the bandwidth in the Chroma path by selecting the bandwidth of Low Pass Filter 2. Leave at default value.

dmsd_lubw

Selects the bandwidth of U and V for filtering in the Luma path by selecting the bandwidth of Low Pass Filter 3. A high bandwidth of U and V means a wide trap in the Luma path after remodulation of U and V and subtraction from CVBS and vice versa. Leave at default value.

Programming

In principle, only the registers dmsd_ccomb, dmsd_ycomb and dmsd_byps have to be set according to the selection for combfilter on/off and Y/C signal processing.

The value for dmsd_lcbw, dmsd_chbw and dmsd_lubw are OK and can be left at their default value which equals the reset value. No changes for these register settings are expected.

The design and the register settings for the 2D combfilter are new. Though from design the most optimal values are selected for the reset values, it is possible that in practice or due to different requirements from the customer other values are needed. Dynamic values are not expected.

SECAM decoder: The SECAM decoder only has a few control settings (dmsd_sthr and dmsd_fctc), which are discussed in the next section. When SECAM is detected, the combfilter is automatically switched off.

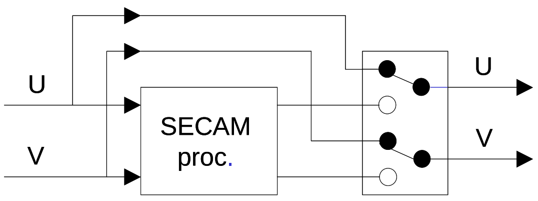


Fig 36.SECAM detector

3.6.4.3 Color PLL and delay line

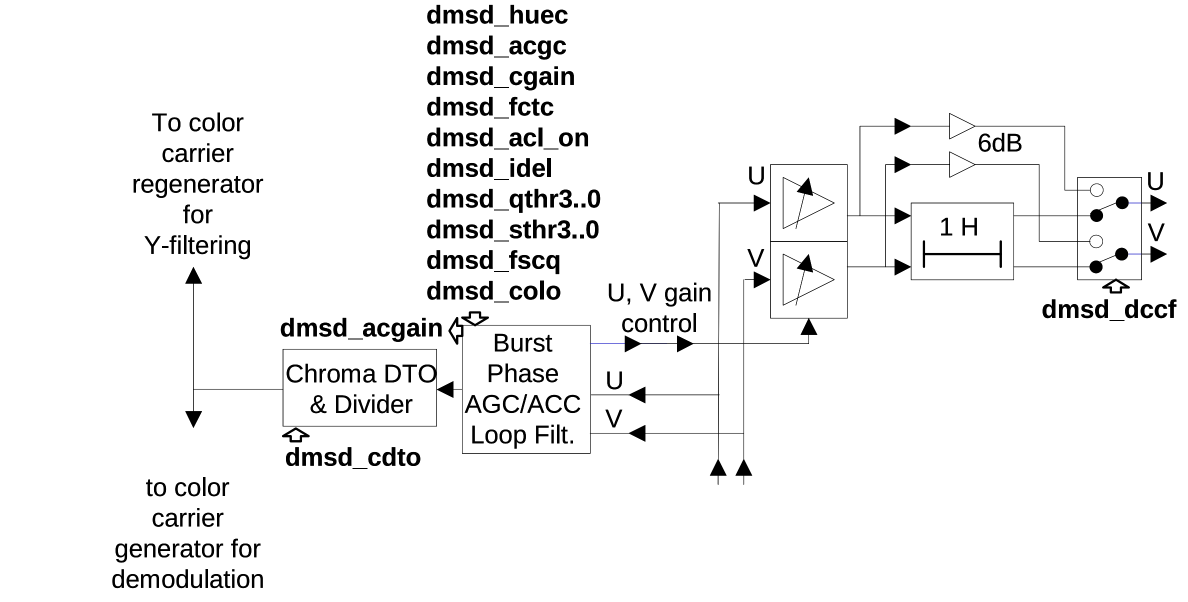


Fig 37.Color PLL and delay line

Color PLL

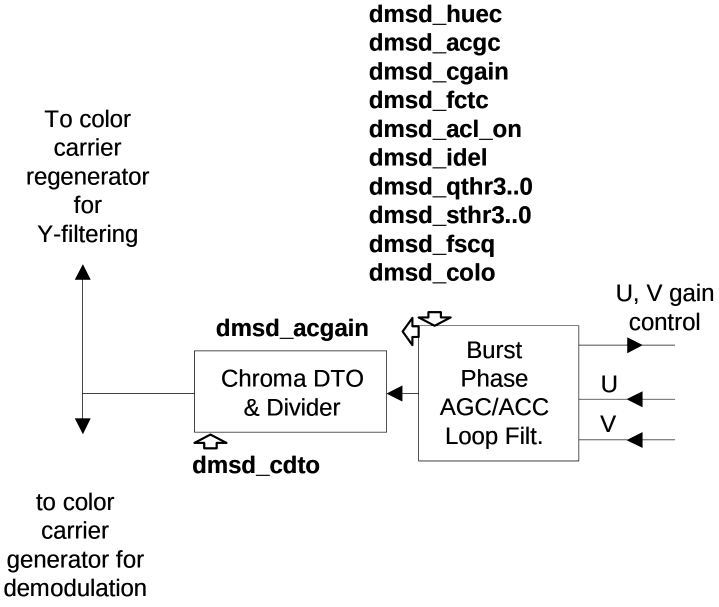


Fig 38.Color PLL

The filtered U and V signals enter the phase detector of the color PLL. In this block, the following functions are implemented:

- Adjustable demodulation phase to be used as HUE for NTSC (dmsd_huec)
- AGC which adapts the gain for the incoming color carrier to bring the burst amplitude to nominal level. Compensates for overall amplitude variation at the Color Carrier frequency. For test purposes the AGC can be switched off (dmsd_acgc), in which case the gain is set by dmsd_cgain. The momentary value of the amplifier in the AGC path can be read from register dmsd_acgain. This works regardless the AGC is enabled or not.
- Selectable fast color PLL time constant for special signal conditions (dmsd_fctc).
- ACL, Automatic Color Limiting. Prevents over saturation when the ratio between chroma and burst is disturbed and due to a too small burst the saturation would increase too much. Can be switched on or off using dmds_acl_on.
- Horizontal Incremental delay setting to match the phase detector output signal with the timing of the incoming CVBS / C samples. Is controlled by dmsd_idel. Value is determined by design and fixed.
- Color killing. The killer levels can be adapted for special signal conditions using dmsd_qthr for PAL and NTSC and dmsd_sthr for SECAM. Note that changing these registers from the reset value (= default value) increases the chance of misidentification.
- Selectable fast PAL/SECAM flip flop phase correction (dmsd_fscq).
- Option to switch off the color killer. (dmsd_colo).

The output of the phase detector is fed into a Discrete Time Oscillator (DTO) which controls the color sub carrier generators, one for demodulation of the incoming color signals (from CVBS or C), one for remodulation of the demodulated U, V signals for subtraction of CVBS to obtain the Y (Luma) signal. To ensure the correct phase of the DTO, the bit dmsd_cdto has to be toggled (from 0 ->1 and back from 1 -> 0) each time after regaining horizontal sync lock after loss of sync and when another setting is selected for dmsd_auto, dmsd_auto_short or dmsd_cstd. This ensures a reset of the DTO and correct behavior.

Table 29: DMSD_COL_DEC control/status - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
18C	0	dmsd_cdto	Clear Chrominance DTO (for remodulation and ‘cleaning’ luma from chroma components) 0 disabled, normal operation mode in automatic mode 1 clear DTO when automatic off and color standard changed	0	R/W
	8..1	dmsd_huec	Hue control (NTSC only) Range 0...+359 degrees (linear)	0	R/W
	9	dmsd_dccf	0 enabled, normal operation 1 disable PAL delay line control	0	R/W
	10	dmsd_acgc	Chroma AGC 0 enabled, recommended 1 disabled, see dmsd_acgain	0	R/W
	17.. 11	dmsd_cgain	Chroma Gain Value range 0.5...7.5 Not required if dmsd_acgc=0	0	R/W

Table 29: DMSD_COL_DEC control/status - address 0X7FF9xxx *…continued*

add xxx	Bits	Name	Function	R/D	R/W
	18	dmsd_fctc	Fast Chroma PLL Time Constant 0 normal mode, recommended 1 fast phase error correction (lower damping factor)	0	R/W
	19	dmsd_acl_on	Automatic Color Limiter 0 no limiting 1 limiting enabled, recommended to prevent over-saturation	1	R/W
	23..20	dmsd_idel	Horizontal Incremental Delay 0111 Value determined by design	0111	R/W

dmsd_heuc

Controls the HUE for NTSC. The range is 0 degrees (0x0) to 359 degrees (0xFF). The range is too large for practical use. Table 30 details the range.

Table 30: Range - dmsd_heuc

Range (Hex)	Range (Dec)	Range (degrees)
0E-FF	224-255	-44 to -1
00-1F	00-31	0 to 44

Remark: The HUE control is still active when a PAL is detected and therefore it should be set to 0 for PAL.

dmsd_acgc

Disables the Chroma AGC. The chroma AGC should always be left on. This is also the reset value.

dmsd_cgain

Not used in AVIP. Leave at reset value.

dmsd_fctc

The reset value, which selects the normal time constant for the color PLL, is correct. The fast filter time constant may be a solution for special signal conditions (e.g. VCR trick modes), but should never be selected as alternative setting for normal use.

dmsd_acl_on

The Automatic Color Limiting prevents over saturation (too large amplitude of U and V signals) when the burst is too small in relation to the color carrier during the active video. It is advisable to leave the ACL always on (which is also the reset value), it also prevents clipping of U and V signals under these conditions.

dmsd_idel

Adjusts the phase of the phase detector output (and the color carrier generator) with respect to the incoming CVBS / C samples. Value determined by design and fixed. Fixed value is equal to the reset value.

dmsd_qthr, dmsd_sthr

These bits control the threshold level of the color killer for PAL/NTSC (dmsd_qthr) and SECAM (dmsd_sthr). Several tests have lead to an optimal value which balances color sensitivity and reliable system recognition. This value (9) is also the register value after reset. For special signal conditions, it is possible to change the threshold value. One should be very careful doing this, because it has negative influence on the overall detection performance. Lowering the threshold value of one of the color killers increases the chance to get under very weak signal conditions a colored picture from the color systems, belonging to that threshold. However, the possibility for wrong color system detection increases. At the same time the sensitivity of the color system with the unchanged killer level decreases. This leads to an imbalance in system recognition performance. Lowering both thresholds increases the chance of misidentification. Increasing both threshold levels just decreases the color sensitivity.

dmsd_fscq

Determines the speed of correction of the PAL / SECAM Flip-Flop when a wrong phase is detected. It is recommend to correct the Flip_Flop once per field, which setting (1) is also the register content after reset. A fast correction can be useful for VCR trick modes, where at Fast Forward or Fast Reverse after each noise bar part of another field is displayed with different PAL / SECAM phase. To react to this trick mode is possible with TV/VCR ‘combis’, where the VCR mode is known, but is it is not recommended with connected VCRs.

dmsd_colo

Disables the color killers. For test purposes.

dmsd_cdto

This bit resets the color DTO and ensures the correct phase relations between all signals. It is advisable to use dmsd_cdto as follows (see also color System Manager).

First, the sequence for a Multi-System set is given:

- After start-up, put the set in automatic mode (dmsd_auto = 2, is reset value).
- Select at preference the short auto loop (dmsd_auto_short).
- Select the preferred system to start the search (dmsd_cstd).
- Toggle dmsd_cdto from 0 -> 1 and back from 1-> 0.

Each time after regaining horizontal sync lock after sync loss and after changing the setting of dmsd_auto dmsd_auto_short and/or the color standard selection (dmsd_cstd), dmsd should be set to 1 and back to 0. When a color system has to be forced the procedure is:

- Put the automatic mode off (dmsd_auto = 0).
- Force the required color system using dmsd_cstd.
- Set dmsd_cdto to 1 and back to 0.

Also in this condition, each time after regaining horizontal sync lock after sync loss, dmsd_cdto had to be set to 1 and back to 0. When going back to automatic mode, the first sequence can be used again. When a set is built for markets with one color system only, the procedure to force a color system should be used.

dmsd_acgain

Returns the value of the momentary gain of the color AGC amplifier. This can be used for an automatic software Y/C detection (see Section "CVBS or Y/C input selection") The dmsd_acgain register only returns a valid value when the color system is PAL or NTSC and the color system manager has recognized the system. For SECAM the value always reads maximum (8F hex or 143 dec) Also when no color carrier is present, the value reeds maximum. Because the color system has to be detected first, a reliable CVBS/YC detection algorithm may take up to 2 seconds.

Programming

dmsd_huec should be limited in range and set to 0 for other systems than NTSC.

dmsd_cdto has to be set to 1 and back to 0 after regaining horizontal sync lock after sync loss and after each change of registers dmsd_auto, dmsd-auto_short and dmsd_cstd

dmsd_acgain can be used in an algorithm for automatic CVBS/YC detection.

The bits dmsd_fctc, dmsd_qthr, dmsd_sthr and dmsd_fscq may be needed under bad signal conditions.

The bits dmsd_acgc, dmsd_cgain, dmsd_idel and dmsd_colo can be left at their reset value.

Delay Line

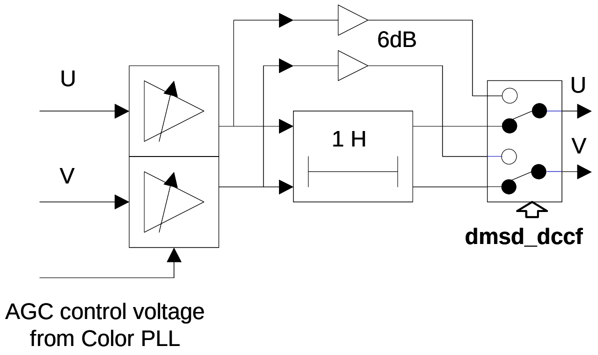


Fig 39.Delay line

The amplitude of U and V is controlled by the color AGC and ACL circuits

The delay line is used for PAL and SECAM color systems.

For NTSC, the delay line can be bypassed. The 6 dB amplification in the U,V path due to the addition at the output of the delay line, is compensated in the bypass path.

Table 31: Delay line - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
	9	dmsd_dccf	Disable PAL delay line. Is controlled automatically in auto modes. 0 Enable PAL delay line 1 Disable PAL delay line	0/x	R/W

dmsd_dccf

When the 2D combfilter is enabled, the delay line should be switched off for NTSC.

Programming

In Auto Search mode (see color System Manager) it is possible to select settings where the switching of the Delay Line is done automatically according to the found color system.

3.6.4.4 Color system manager

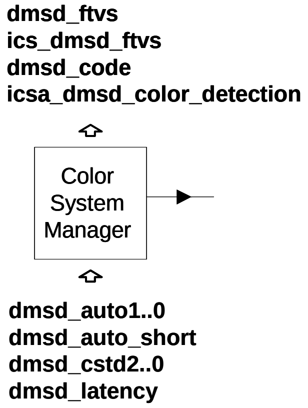


Fig 40.Color system manager

The color System Manager offers various possibilities to control the color search:

- Full search: Will search for all possible systems including Latin America systems like PAL M and PAL N. It is possible to define the preferred color system to start the search.
- Short search: Will only search for the most common systems (PAL 4.43, SECAM, NTSC 3.58 and NTSC 4.43). Shortens the color system recognition time. Also in this loop it is possible to define the preferred color system to start search.
- Forced mode: Possibility to force one color system only, suitable for market area's like USA and Philippines (NTSC 3.58)

In the automatic search mode, it is possible to define different levels of automatic setting of the filters, combfilter and traps optimized for the found color system. It is also possible to set the search time per color system. A bit indicates when a color system is found, this can also be signalled via an interrupt. It is also possible to read the found color system (PAL, SECAM or NTSC)

In view of the amount of information, the bit description is split in two parts:

- Control bits, which control the color System Manager
- Status bits, which can be read

Table 32: Color system manager - control bits - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
188	1..0	dmsd_auto	Automatic TV system detection mode 00 Level 0, disabled 01 Level 3, Active, all filters adapting automatically 10 Level 2, Active, some filters adapting automatically 11 Level 1, Active, all filters to be set by software See separate table for automatic filter settings	2/x	R/W
	2	dmsd_auto_short	Selects between all color system search loop and a limited color system search loop (for faster detection) 0 All color system search loop (LATAM) 1 Limited color system search loop (Europe, ROW) for faster detection. Only searches: PAL 4.43, SECAM, NTSC M, NTSC 4.43	0/x	R/W
	5..3	dmsd_cstd	Color standard selection. When dmsd_auto = 0, forces the color system according the table below: Fv = 50 Hz Fv = 60 Hz 000 PAL 4.43 NTSC M[1] 001 NTSC 4.43 PAL 4.43 010 PAL N NTSC 4.43 011 --------- PAL M 100 PAL 4.43 NTSC J[1] 101 SECAM --------- Other --------- -------- Note that the forced standard is pending on the vertical frequency, either automatic detected, either forced When auto detection is enabled (dmsd_auto 01,10,11) then selects the first color system to start the search, in 50 Hz also the 2nd system to search for is selected. Fv = 50 Hz Fv = 60 Hz 000 PAL 4.43 SECAM NTSC M[1] 100 PAL 4.43 SECAM NTSC J[1] 101 SECAM PAL 4.43 NTSC M[1] Other PAL 4.43 SECAM NTSC M[1]	0/x	R/W
	8..6	dmsd_latency	Number of fields before stepping to the next color standard in auto mode	11	

[1] NTSC M mode removes the pedestal of 7 IRE from the Y signal, while NTSC J mode leaves the pedestal unchanged.

Before discussing the bits in more detail, a note about NTSC M and NTSC J. Both systems refer to NTSC with a color carrier frequency of 3.58 MHz. The difference between these two color decoder modes is the processing of the pedestal. The NTSC 3.58 standard has a pedestal of 7 IRE. When for the color decoding NTSC J is selected, the pedestal will not be removed from the Y signal. When NTSC M is selected, the pedestal is removed from the Y signal. The presence or removal from the pedestal has influence on the behavior of Black Stretch. When the pedestal is removed, the blackstretch will not react on the signal. When the pedestal is present, black stretch will pull the 7 IRE level to black according a non-linear transfer curve. NTSC J or NTSC M can be selected for NTSC only countries, but for multisystem applications NTSC M is probably the best choice because features like blackstretch can be made equal for all color systems.

dmsd_auto, dmsd_cstd: To distinguish two operating modes:

1. Disabled, force the color system (dmsd_auto = 0 0)
2. Automatic (dmsd_auto = 0 1, 1 0 or 1 1)

Disabled

When disabled, the color system has to be forced using **dmsd_cstd**. The systems, which can be selected are given in the bit table. Because the forced system depends on the vertical frequency (50 or 60 Hz), this setting also has to be forced. This is possible in the Vertical Synchronisation part.. The procedure to force the field frequency is:

- Set dmsd_aufd = 0 (non-automatic field detection, 1 = automatic field detection)
- Select the vertical frequency using dmsd_fsel (0 = 50 Hz, 1 = 60 Hz).

See Section “Vertical sync processing 1 Fh” .

In forced mode, all settings of delay line and filters (dmsd_dccf, dmsd_chbw, dmsd_lcbw, dmsd_lubw and dmsd_lufi) need to be set by the software. The setting of the combfilter (dmsd_ycomb, dmsd_ccomb) and bypass mode (dmsd_byps for Y/C mode) have to be taken into account for the correct filter settings.

The table for dmsd_auto mode "3" can serve as input for the settings to be selected. After forcing the system, it is needed to set dmsd_cdto to 1 and back to 0. This guarantees the correct phase relationship between all samples.

Note that dmsd_cdto has to be toggled whenever dmsd_auto, dmsd_auto_short, or dmsd_cstd is changed (see also 2.3.3 color PLL and Delay Line).

Automatic

When automatic color system search is enabled (don't forget to toggle dmsd_cdto after selecting this mode) several modes can be selected. It is possible to control all settings of the delay line and filters by software, but to ease programming, a number of settings or even all settings can be done automatically, based upon the found color system and the user selection of the comb filter (dmsd_ycomb, dmsd_ccomb) and bypass mode (dmsd_byps for Y/C mode) The table below indicates which settings are controlled automatically and what value is set, depending on the selected automation level.

The settings, used in auto mode level 1 to 3.

Table 33: Auto mode - settings

Standard	User selection				Settings controlled automatically							
					Level1	Level 2+3				Level 3		
	Signal path	byps	ycomb	ccomb		dccf	lcbw	lubw	ycomb	ccomb	lufi	chbw
PAL	Comb	0	1	1		0	110	0	[2]	[2]	0000	0
PAL	Notch	0	0	0		0	000		[2]	[2]	0110	0
PAL	Flat (for YC)	1	[1]	0		0	110	0	0	0	0000	1
NTSC	Comb	0	1	1		1	110	0	[2]	[2]	0000	0
NTSC	Notch	0	0	0		0	000	0	[2]	[2]	0110	0
NTSC	Flat (for YC)	1	[1]	0		1	110	0	0	0	0000	1
SECAM	Notch	0	[1]	[1]		0	000	1	0	[1]	1011	0
SECAM	Flat (for YC)	1	[1]	[1]		0	000	[1]	0	[1]	0000	0
Bk/White	-	1	[1]	[1]	ycomb=0	[1]	[1]	[1]	0	[1]	0000	[1]

[1] No influence

[2] User Selection. Chosen in ycomb and ccomb, under User Selection is taken over.

Level 1 (dmsd_auto = 1 1)

Automatic color system detection, all filters / delay line have to be programmed. One exception: when no color system is found (Black & White), ycomb is forced to 0.

Level 2 (dmsd_auto = 1 0)

Automatic color system detection, some filters and delay line are programmed according the table. **dmsd_ccomb**, **dmsd_lufi** and **dmsd_chbw** still have to be programmed.

Level 3 (dmsd_auto = 0 1)

Search loop in Automatic Mode Automatic Color system detection, all filters adapt automatically. It is advisable to set in automatic mode dmsd_auto = 2. The automatic filter setting of dmsd_lufi and dmsd_chbw are not optimal, it is advisable to control these settings by software.

In automatic mode, dmsd_cstd determines the first (and in 50 Hz also the second) color system that is searched for. In the table below, the search order is given. Note that for a successful search, first the correct field frequency has to be detected (dmsd_fidt, 0 = 50 Hz, 1 = 60 Hz, see also Vertical Sync Processing 1 Fh)

Once programmed, the value of dmsd_cstd is valid for both 50 and 60 Hz.

Table 34: Full search loop

Order	50Hz (fidt=0)			60Hz (fidt=1)		
	cstd=000	cstd=100	cstd=101	cstd=000	cstd=100	cstd=101
1	PAL 4.43	PAL 4.43	SECAM	NTSC M	NTSC J	NTSC M
2	SECAM	SECAM	PAL 4.43	NTSC 4.43		
3	PAL N			PAL M		
4	NTSC 4.43			PAL 4.43		

dmsd_auto_short

The search loop can be shortened when the Latin America color systems like PAL M and PAL N are not needed.

When dmsd_auto_short = 0, the full loop in the table above is executed.

When dmsd_auto_short = 1, the loop is shortened to the first two systems, see table below.

Table 35: Short search loop

Order	50Hz (fidt=0)			60Hz (fidt=1)		
	cstd=000	cstd=100	cstd=101	cstd=000	cstd=100	cstd=101
1	PAL 4.43	PAL 4.43	SECAM	NTSC M	NTSC J	NTSC M
2	SECAM	SECAM	PAL 4.43	NTSC 4.43		

As can be seen, the settings of dmsd_cstd determine in the same way the search order preference as in the full search loop.

dmsd_latency

Determines the number of fields before the color system manager steps to the next color system in the loop. The standard setting is 3 (fields/color system).

Timing before a color system is recognized

The following items determine the time, before the color system is found.

1. The locking of the incoming samples to the new line phase and frequency.
2. The recognition of the field frequency (50 or 60 Hz) When after channel change or input change the field frequency is identical to the previous signal, the settle time is short (< 100 msec, depending on the phase difference of the vertical retrace).
3. The search loop itself, which takes 60 msec / color system (50 Hz, dmsd_latency = 3).

The most dominant one is the 50 / 60 Hz detection. The switch-over from 50 to 60 Hz or vice versa can take up to 600 msec. Due to all these variables, the time before a color system is recognized varies quite a lot. Worst case (switching from 50 to 60 Hz source or vice versa) and changing color standard the recognition time may run up to 800 msec. Best case (no change in vertical frequency, same color system as previous source) the time can be as short as 30 msec.

Table 36: Status bits - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
004	4	dmsd_code	color detected 0 No color detected 1 color system detected according dmsd_ftvs		R
FE0		ics_dmsd_code	Interrupt flag set to 1 when the status of the color detection bit dmsd_code changes See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		R
004	7..6	dmsd_ftvs	Found TV System, indicates the detected color standard 00 Black & White (no color system detected) 01 NTSC 10 PAL 11 SECAM		R
FE0	6	ics_dmsd_ftvs	Interrupt flag set to 1 when value of dmsd_ftvs is changed See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		R

dmsd_code

Indicates whether a color system is found. Can be used as a first indication to check whether the automatic search loop has recognized a color system or when a single color system is forced whether the forced system is found. A status change of this bit can also trigger an interrupt.

dmsd_ftvs

Indicates which color system is found. Note that "Black and White" and no color system found yet give the same reading. The three color systems, which can be indicated are PAL, SECAM and NTSC. Note that no information is available about the color carrier frequency. If needed, the color carrier frequency can be derived from the FM mono sound carrier, which frequency can be determined by the sound core. This of course only works for off-air signals. It is possible to generate an interrupt when the value of dmsd_ftvs changes.

Remark: Latency of the read-out bits - After changing channel or signal source, it can take up to 50 msec. before the bits dmsd_code and dmsd_ftvs change status. So if one immediately after channel or input change reads out the status bits, one might conclude a color system is found while the bits indicate the status from the previous signal. It has been observed when switching from PAL to PAL, the bits do not even change status. When switching to another color system, it takes 30 to 50 msec. max before the bits indicate color loss and the new color search starts. Take this behavior into account when designing the source switching and channel changing algorithms.

Programming

A short summary how to use the bits.

Forcing a color system:

- Set dmsd_auto to 0

- Select the wanted color system using dmsd_cstd
- Select the wanted settings for filter bypass and combfilter dmsd_byps, dmsd_ycomb, dmsd_ccomb
- Program all filters according the forced color system and selected bypass and combfilter settings. dmsd_dccf, dmsd_chbw, dmsd_lcbw, dmsd_lubw and dmsd_lufi
- Program dmsd_ldel according the forced system (See 2.3.1 Y processing)
- Set dmsd_cdto to 1 and back to 0.
- Select also the correct field frequency setting dmsd_aufd = 0 and selecting the frequency using dmsd_fsel.
- After channel change or source switching, read dmsd_code and/or dmsd_ftvs to check whether the system is found. Take the latency into account

Automatic color search:

- Set dmsd_auto to 1, 2 or 3, depending on the required level of automation
- Select whether the full loop or short loop is required, setting dmsd_auto_ short.
- Select the preferred search order using dmsd_cstd
- Set dmsd_cdto to 1 and back to 0
- Take care that also the field frequency selection is set to automatic (dmsd_aufd = 1)
- After channel change or source switching, read dmsd_code and dmsd_ftvs to check which color system is found. Take the latency into account
- Depending on the found system and the signal source program the filter bypass and combfilter dmsd_byps, dmsd_ycomb, dmsd_ccomb
- Program dmsd_ldel according to the found color system
- Depending on the level of automation, program the non-automatic programmed filters and the delay line bypass dmsd_dccf, dmsd_chbw, dmsd_lcbw, dmsd_lubw and dmsd_lufi

3.6.4.5 Signal controls, Macrovision and debug

Signal controls: The processed Y and demodulated U and V pass the control block before being sent to the fast YUV switch. The control block contains the video controls contrast, brightness and saturation. Because these items are controlled at another place in system (see PNX8550) these controls are set to a fixed level. A noise shaping function minimizes quantization at the output. A dither function enables to go from 10 bits to 9 bits at the output, while maintaining the 9 bits resolution for low frequency signals like ramps. This function is not used in AVIP. Finally, a small offset alignment is possible for the U and V signals to correct small design errors.

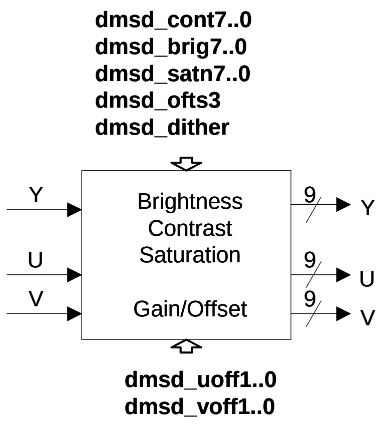


Fig 41.Color decoder output control

Table 37: Signal control - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
194	11..4	ddmsd_cont	Brightness control, not used in AVIP. Set to 44 hex.	44	R/W
	19..12	dmsd_brig	Contrast control, not used in AVIP, set to 80 hex	80	R/W
	27..20	dmsd_satn	Saturation control, not used in AVIP, set to 40 hex	40	R/W
	28	dmsd_ofts3	Selects output formatter mode and noise shaper mode 0 Linear mode, no noise shaping 1 Noise shaping activated (recommended)	1	R/W
	29	dmsd_dither	Dithers 10 bit output to 9 bits. Not used in AVIP 0 No dithering (Recommended value) 1 Dithering enabled	0	R/W
	1..0	dmsd_uoff	U offset to correct for rounding errors 00 No offset 01 + 1 LSB 10 + 2 LSB 11 + 3 LSB	0	R/W
	3..2	dmsd_voff	V offset to correct for rounding errors 00 No offset 01 + 1 LSB 10 + 2 LSB 11 + 3 LSB	0	R/W

dmsd_cont, dmsd_brig

Contrast and brightness are controlled outside of the AVIP.

Remark: If it is necessary (depending on success of removing PNX3000 peaking around 4.5 MHz) to set in the AGC block agc_cvbs_yyc_ctrl_copy_prot_pi to 1 to ensure sufficient headroom, the amplitude decrease should be compensated by contrast and brightness setting of the VIDDEC.

dmsd_satn

Saturation is controlled outside of the AVIP.

dmsd_ofts3

These bits enables dithering form internally used 11 bits to 10 bits at the decoder output. Set to 1 to minimize quantization and noise.

dmsd_dither

Dithers from 10 bits to 9 bits. This function is not used.

dmsd_uoff, dmsd_voff

Intended to correct for rounding errors in the processing. Not needed in AVIP.

Macrovision: The 1 Fh Macrovision block can detect the Macrovision in sync/white level during vertical retrace and two color stripe methods, which are defined for NTSC with DVD.

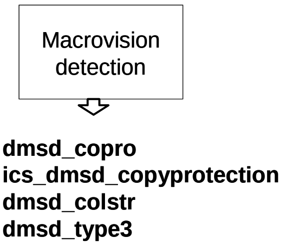


Fig 42.Macrovision detection

Table 38: Macrovision detection - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
004	5	dmsd_copro	Detects whether the input signal is Macrovision encoded 0 No Macrovision 1 Macrovision detected		R
FE0	5	ics_dmsd_copro	Interrupt flag set to 1 when dmsd_copro changes See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		R
000	8	dmsd_colstr	Detects Macrovision color stripe encoding 0 No color stripe 1 color stripe encoding detected		R
000	9	dmsd_type3	Detects Macrovision color stripe type 3 encoding 0 No color stripe type 3 1 color stripe type 3 encoding detected		R

dmsd_copro

Detects the Macrovision during vertical retrace in sync (reduced sync amplitude and false sync pulses). Because the sync amplitude changes, the AGC settings have to be adapted when Macrovision in sync is detected:

For CVBS / Yyc:

Normal – agc_cvbs_yyc_ctrl_copy_prot_pi = 0

Copro = 1 – agc_cvbs_yyc_ctrl_copy_prot_pi = 1

For YUV:

Normal – agc_y_cyc_ctrl_top_sync_pi = 100 hex

Copro = 1 – agc_y_cyc_ctrl_top_sync_pi = 118 hex

It is possible to generate an interrupt when dmsd_copro changes.

dmsd_colstr, dmsd_type3

Currently no action is expected relating to the recognition of these Macrovision standards for NTSC on DVD.

Debug & Control: Most bits are intended for debugging the design. and possibilities are offered to invert the phase of some control signals.

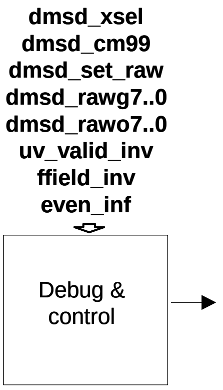


Fig 43.Debug and control

Table 39: Debug and control - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
198	1..0	dmsd_xsel	Selects clock (Xtal) frequency 00 24.576 MHz 01 32.11 MHz 10 27.00 MHz Used for AVIP	2	R/W
	2	dmsd_cm99	Selects compatibility with 7199 decoder 0 Normal mode, recommended for AVIP 1 Compatibility with 7199 decoder	0	R/W
	3	dmsd_set_raw	Raw data mode for debug 0 Normal mode 1 Bypass mode, bypasses Luma filtering, Comb, Brightness, Contrast, Chroma vertical filtering	0	R/W
	11..4	dmsd_rawg	Sets Luma gain for Raw Data mode	0	R/W
	19..12	dmsd_rawo	Sets Luma offset for Raw Data mode	0	R/W

Table 39: Debug and control - address 0X7FF9xxx *…continued*

add xxx	Bits	Name	Function	R/D	R/W
040	4	uv_valid_inv	Inverts U and V signals 0 Normal mode 1 U and V swapped	0	R/W
	5	ffield_inv	Inverts polarity of the First Field info signal for the ITU656 0 Normal mode 1 Inverted (swapping first and second field) for debug	0	R/W
	6	even_inv	Inverts polarity of the even_ccir_l signal for the Data Capture Unit (DCU) 0 Normal mode 1 Inverted (swapping first and second field) for debug	0	R/W

uv_valid_inv

U and V are multiplexed in one data stream. When the phase of the U, V multiplexing /demultiplexing clock is reversed, the U and V data are swapped. With this bit it is possible to correct a wrong inversion in the chain.

ffield_inv

Inverts the readout of the odd- and even field going to the ITU656. Can correct an internal wrong inversion in the processing.

even_inv

Same as bit above, but then for the signal, going to the DCU and ITU656.

3.6.4.6 Sync processing

The Sync processing block contains 4 building blocks (See Figure 44):

- 1 Fh horizontal and vertical sync processing
- 2 Fh horizontal and vertical sync processing
- A Discrete Time oscillator (DTO) and clock generator, shared for 1Fh and 2Fh
- Fast blanking, external 2Fh sync and timing info output

The 1 Fh sync block contains for horizontal sync processing the PHI-1 loop phase detector and loop filter. It is possible to read out whether the loop is locked and to set the time constant of the lop filter. A noise readout enables the user to adapt the loop filter time constant for noisy conditions. The position and width of the horizontal output pulses can also be programmed.

The 1 Fh vertical sync processing uses a counter to generate the internal vertical sync. The counter can be forced to direct sync (for fast catching) or programmed for noise rejection (for normal use with off-air signals). The field frequency can be determined automatically or forced to 50 or 60 Hz. It is possible to force an odd / even field sequence at the odd / even field output, independent of the properties of the incoming signal.

The 1Fh measurement block provides information over the phase deviation from incoming line to incoming line (can be used for VCR detection). It also indicates whether the field length has a standard number of lines (525 / 625). This can be used to detect trick mode in VCR's. The 2 Fh sync block contains almost the same functionality as the 1 Fh sync block. Some provisions for non-practical situations in 2 Fh like noisy signals are omitted. The 2 Fh block can handle also ATSC signals, including tri-level sync on Y. It is also possible to use an external H and V sync in 2 Fh mode.

The DTO is used as oscillator for both 1 Fh and 2 Fh sync processing. It is clear from this set-up that only one loop (1 Fh or 2 Fh) can be active at the same time. For 2 Fh, a number of clock frequencies for the VIDDEC have to be doubled. This is done outside the VIDDEC and controlled by programming some General Purpose I/O registers (GPIO) in the GTU. Depending on the selected clock frequency the 1 Fh loop or the 2 Fh loop is active.

There are two inputs, which serve as Fast Blanking inputs for RGB insertion in 1 Fh or as Horizontal Sync input in 2 Fh. Also two inputs for external vertical sync in 2 Fh mode are provided. H and V sync pulses are made available for the ITU656 block and the Data Capture Unit (DCU) which extracts TXT, Closed Caption, WSS and other data services from the incoming CVBS Finally, timing info is provided to the PNX3000 (HV_info) to enable correct black clamping of the locked incoming signal (CVBS, Y/C, YPrPb or RGB).

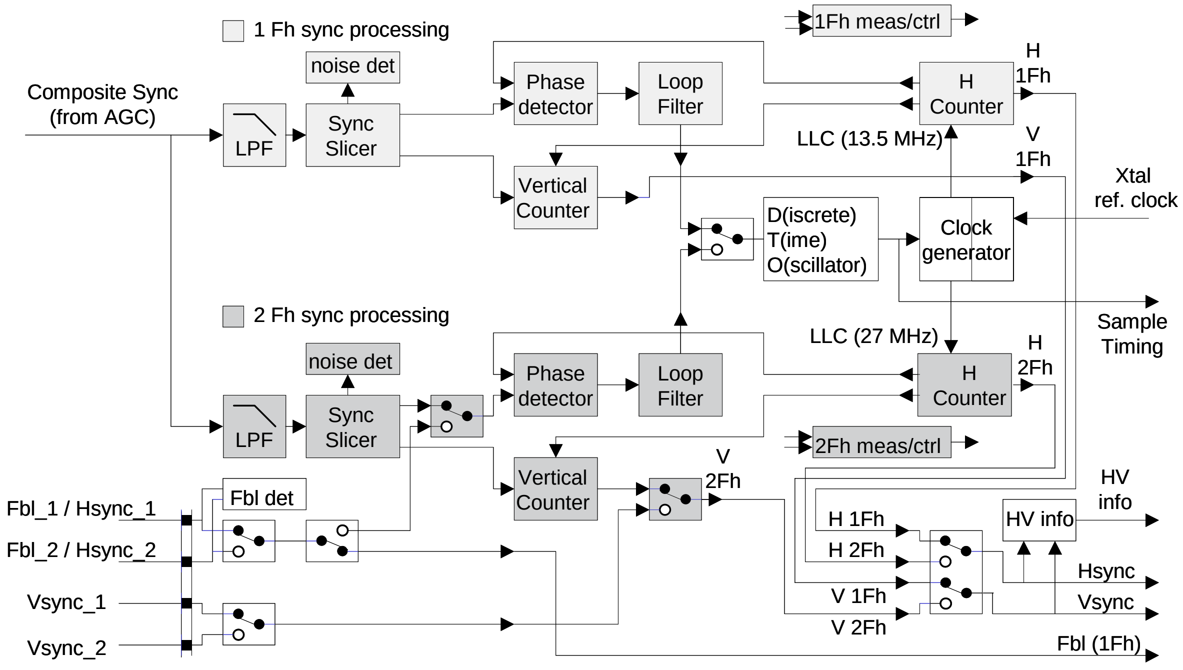


Fig 44.Sync processing

Horizontal sync processing 1 Fh and measurement/control: The incoming sync signal from the AGC first passes a low pass filter to remove high frequency components. (See Figure 45).

The sync slicer extracts the sync data from the signal. Also the noise is measured at sync bottom to determine the signal to noise ratio from the incoming signal. The noise info (dmsd_hnoise) can be used to adapt the time constant from the PHI-1 loop filter. The PHI-1 phase detector indicates whether the loop is locked to the incoming signal (dmsd_hl).

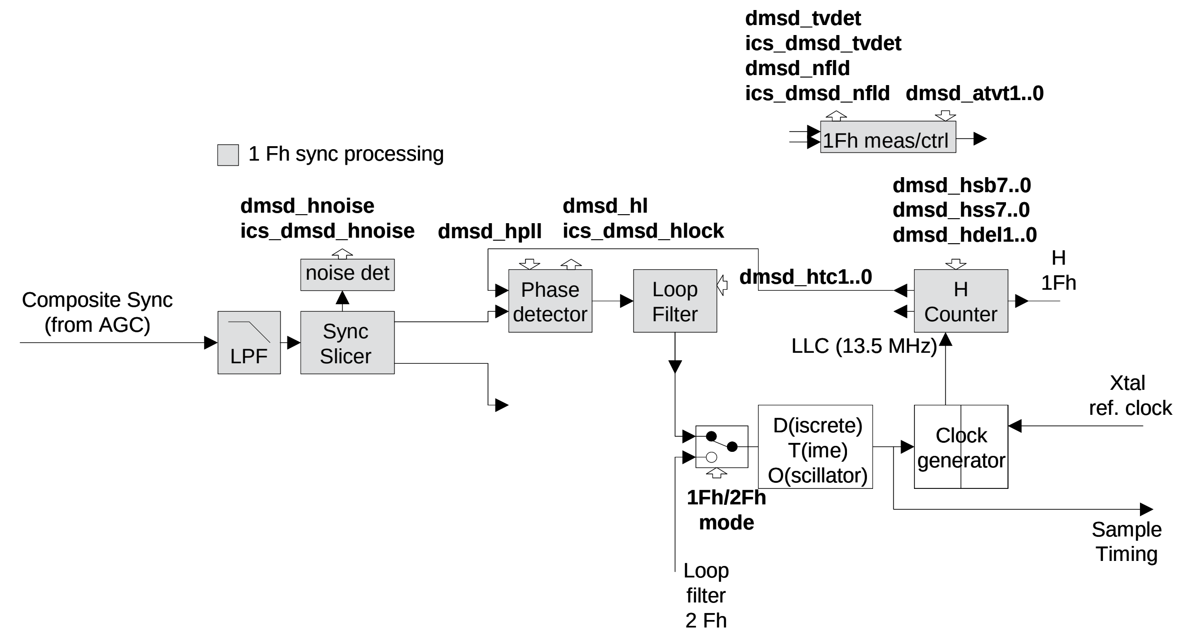


Fig 45. Horizontal sync processing 1 Fh

It is possible to generate an interrupt when the status of the h-lock readout changes. The time constant of the PHI-1 loop filter can be set to different values to optimize the behavior for various input signal conditions (dmsd_htc) The PHI-1 can be set in free running mode (dmsd_hpll). This is for test purposes and not used in AVIP.

The Start position, the stop position and a fine shift of horizontal pulses for the following blocks can be programmed (dmsd_hsb, dmsd_hss, dmsd_hdel). The measurement block contains circuits to detect whether a VCR is connected. It can measure whether horizontal phase jumps are present in the incoming signal (dmsd_tvdet). The sensitivity for the phase jump size can be set using dmsd_atvt. It is possible to switch automatically the PHI-1 time constant (dmsd_htc), pending on the status of tv_det (Fast for VCR, slow for stable sources) To detect trick modes, another circuit measures whether the field length is nominal (not deviating more than +/- 1/2 line from standard)

The Discrete Time oscillator is used for both the 1 Fh PLL and the 2 Fh PLL. The selection of 1 Fh and 2 Fh is done by switching the clocks for the VIDDEC using settings in the GPIO registers of the GTU. After clock switching, all register values have to be reset and the registers, needing a value different from reset have to be reprogrammed with the correct value.

Table 40: Horizontal sync - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
004	8	1 Fh: dmsd_hnoise	Is set when noise level on the sync bottom is detected. 0 No noise on sync bottom 1 Noise on sync bottom When 1, dmsd_htc should be set to 0		R
FE0	7	1 Fh: ics_dmsd_hnoise	Interrupt flag set to 1 when dmsd_hnoise changes. See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		R
180	0	1 Fh: dmsd_hpll	Sets horizontal PLL in free running (Xtal based) mode 0 Normal operation (locked to the input signal) 1 Free running on 1 Fh	0	R/W
004	0	1 Fh: dmsd_hl	Indicates that the horizontal PLL is in lock 0 No lock 1 In lock		R
FE0	0	1 Fh: ics_dmsd_hl	Interrupt flag set to 1 when dmsd_hl changes. See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		R
180	2..1	1 Fh: dmsd_htc	Horizontal Time Constant of the PHI-1loop 00 Slow mode for noise conditions (preferred when dmsd_hnoise = 1) 01 Normal mode with limited correction / line, preferred for standard condition. 10 Switches automatically between fast and slow time constant, depending on detection of phase errors by read-out bit dmsd_tvdet 11 Fast mode without limitation of correction/line	1/x	R/W
180	10..3	1 Fh: dmsd_hsb	Horizontal Sync output pulse Begin position. Range is -107 to +107 in 8 pixels/step. Is fixed value for AVIP of FA.	FA	R/W
	18..11	1 Fh: dmsd_hss	Horizontal Sync output pulse Stop position Range is -107 to +107 in 8 pixels/step. Fixed value for AVIP of FB	FB	R/W
	20..19	1 Fh: dmsd_hdel	Horizontal Sync output pulse delay. Range is 0 to 3 in 2 pixels/step. Fine control of above. Fixed value for AVIP of 0	0	R/W
		1Fh / 2Fh mode	Signal, coming from the GPIO (General Purpose I/0) block, controlling also the clocks.		
000	10	dmsd_tvdet	Indicates horizontal phase jumps (VCR detection) 0 Non-stable input (Phase jumps detected, e.g. VCR) 1 Stable input (e.g. broadcast, DVD) Sensitivity adjustable with dmsd_atvt1..0 When dmsd_htc is set to 2, the H-pll time constant is automatically switched between fast (non-stable input) and slow (stable input) when dmsd_tvdet toggles.		R
FE0	21	ics_dmsd_tvdet	Interrupt flag set to 1 when dmsd_tvdet changes. See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		R

Table 40: Horizontal sync - address 0X7FF9xxx *…continued*

add xxx	Bits	Name	Function	R/D	R/W
000	11	dmsd_nfld	Detection of nominal field length 0 Field length not nominal 1 Field length nominal Nominal is for 50 Hz from 312 to 313 lines / field Nominal is for 60 Hz from 262 to 263 lines / field		R
FE0	22	ics_dmsd_nfld	Interrupt flag set to 1 when dmsd_nfld changes. See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		R
180	22..21	dmsd_atvt	Sets detection threshold for phase jump detection of read-out bit dmsd_tvdet 00 High sensitivity for phase jumps 01 Recommended value ----- 11 low sensitivity for phase jumps	1	R/W

dmsd_hnoise

Indicates the amount of noise, measured on the sync bottom. When the signal to noise ratio gets below a certain threshold the bit is set to 1. In this condition it is advisable to set the time constant of the horizontal PLL (dmsd_htc) to slow ("0") to avoid a too high horizontal jitter under noisy conditions. See further the description of the dmsd_htc. The change of the dmsd_hnoise bit can be signalled via an interrupt.

dmsd_hl

This is the most important bit to indicate whether a valid video signal is present. No status reading of the VIDDEC is valid when there is no H-lock. The reason is that the total VIDDEC concept is based upon line locked samples, and only when there is H-lock, the line locked samples can be calculated on the correct position to enable decoding. The change of the status of this bit can be signalled via an interrupt. For a description of optimization of the catch time, see the part "Programming" after the bit description.

Read-out latency: Please note that after losing horizontal sync lock, it can take 50 msec. before this status is reflected in the dmsd_hl readout. This time has to be taken into account when switching channel or changing source to prevent that the H-lock status of the previous signal being regarded as H-lock on the new channel / source signal. This issue is valid for all readout bits of the VIDDEC.

Points to note when mode switching between 1FH/2FH modes. Ignor all bits from not used mode. Wait for for both H and V lock bits to be stable before accepting the appropriate status bits.

(See Section “CVI Input Selection”).

dmsd_htc

Sets the time constant of the PHI-1 loop. There are 4 settings, each are described below:

00 - Slow time constant. Optimal setting for noisy CVBS signals from RF. The presence of noise can be detected using the bit dmsd_hnoise described above. This setting should only be used for (noisy) off-air signals via the antenna.

01 - Normal mode. This mode reacts quite fast on phase changes in the input signal, but the correction per line is limited. This is a proper setting for normal use and provides a stable picture, even when losing sync.

10 - In this mode the PHI-1 time constant is automatically switched between slow mode and fast mode depending on the presence of the phase jumps, indicated by the bit dmsd_tvdet. This bit indicates the presence of a VCR by measuring the phase jumps in the input signal. The sensitivity for phase jumps can be set to 4 levels using dmsd_atvt. Can be used to switch automatically the PHI-1 time constant to fast for VCR input signals to get a proper performance under these conditions. Not used in AVIP.

11 - Fast mode. Follows very fast phase jumps in the input signal. Has optimal catching speed but the phase jumps from line to line are not limited.

See ‘Programming’ below.

dmsd_hsb, dmsd_hss and dmsd_hdel.

Not used in AVIP. Leave on reset value.

dmsd_tvdet, dmsd_atvt

(Readout only valid when dmsd_hl = 1, also note latency)

Not used in AVIP, backup to solve problems in performance if needed.

Can be used to detect whether the signal is coming from a VCR. The detection is done by measuring the size of the horizontal phase jumps in the incoming signal. When no phase jumps are detected, the signal is regarded as being stable from broadcast or DVD. The sensitivity of the detection can be set by dmsd_atvt (4 settings). The higher the setting, the larger the horizontal phase jump needs to be in order to be recognized as VCR signal by dmsd_tvdet.

When the horizontal pll time constant dmsd_htc is set to 2, the PHI-1 time constant is set to fast (for VCR) or slow (for stable sources) depending on the status of dmsd_tvdet. Status change of dmsd_tvdet can be indicated by an interrupt.

dmsd_nfld (readout only valid when dmsd_hl = 1, also note latency) Not used in AVIP, backup to solve problems in performance if needed. Indicates whether the input field length is nominal and can be used to identify whether a signal, coming from a VCR in trick mode is connected. The specification for nominal field length are:

- Nominal field length for 50 Hz is 312 to 313 lines.
- Nominal field length for 60 Hz is 262 to 263 lines.

All standard broadcast, DVD play mode and VCR normal play mode are considered standard length. Also non-interlaced signals like those from MPEG 1 video players are considered standard. All VCR trick modes with phase jumps deviating 1 or more lines per field are recognized as non-standard field length. These can be used to change settings to get better performance in VCR trick modes e.g. dmsd_htc, or dmsd_fscq of the color decoder. Can be used to distinguish between 1Fh and 2Fh input sources.

Remark: To be checked if dmsd_nfld also works under all settings of dmsd_vnoi. First check with AVIP showed that the bit only works when dmsd_vnoi = 0

Programming

A proposal to achieve a good system performance for the PHI-1 is given below.

Normal operation

The settings for the horizontal PLL become:

- dmsd_htc = 3 when dmsd_hnoise = 0
- dmsd_htc = 0 when dmsd_hnoise = 1

Optimizing H-lock catch time and PHI-1 settings

The most dominant factors, influencing the H-lock catch time, are:

- The dmsd_htc setting
- The setting of the input clamp mode in PNX3000
- The size of the input capacitor at the CVBS / Y input

For the input capacitor, a maximum value of 100 nF is advised. Higher values cause a too long settling time of the clamping circuit, leading to a longer H-lock time. Also the reaction time on DC jumps at the CVBS inputs is negatively influenced using higher input capacitor values.

The settings advised for search tuning are:

Select correct free running mode of the MBF-PLL:

Fnom = 1 (Set PNX8550 to cater for free running mode,)

Bypass = 0

Interlace = 1 (select 1 Fh input)

Fh_sel = 0 (select 1 Fh input)

Line_sel = 0 for 60 Hz (525 lines) or 1 for 50 Hz (625 lines)

(depending on the preferred free running display mode during search)

Set dmsd_htc = 3 (fast mode)

Set the clamp mode in the PNX3000 to top sync clamp (CLP = 0). This shortens the clamping time and as consequence the H-lock time at the cost of less performance on reduced sync. However, extreme reduced sync is not common on RF signals. Note to set always CLP = 1 (black clamping mode) under normal condition (non-search), else the correct black level of the input signal cannot be guaranteed.

Under these conditions, the H-pll lock time is below 80 msec. This does not mean that the minimum waiting time per frequency step has to be 80 msec. In practice it is allowed to add the waiting times of all frequency steps that the IF-PLL is locked to the picture carrier.

Example:

When stepping at 1 MHz, it is possible that you only get one step an IF-PLL lock. Then it is needed to have a stepping time of 80 msec. to guarantee H-lock. When stepping at 0,5 MHz, you can reduce the waiting time/step by a factor 2 to 40 msec. The total search performance for weak transmitters will still increase, because now it is possible you have 3 steps where the IF-PLL can lock to the picture carrier, adding up the possible lock time to 120 msec.

Changing channel or input source

The horizontal catching time is acceptable when changing channel or input source without changing the horizontal register settings. Take care of the readout-latency of 50 msec when reading dmsd_hl after changing channel or input source.

If you want to have the highest possible speed and accept limited performance for reduced sync, you can apply the algorithm below:

- Switch channel or input source
- Select top sync clamp (CLP = 0)
- Wait at least 50 msec. before starting to read dmsd_hl to account for the readout latency
- Wait for dmsd_hl = 1
- Set clamping back to black level clamp (CLP = 1).

Vertical sync processing 1 Fh: The vertical sync is fed to a counter. The vertical counter indicates several properties of the incoming signal:

- Vertical lock to the incoming signal (dmsd_vl). The values of the readout bits mentioned below are only reliable when there is Horizontal Lock and Vertical Lock.
- Interlace (dmsd_intl)
- Field frequency, 50 or 60 Hz (dmsd_fidt). A status change of each of these readout bits can be signalled by an interrupt.

  Many properties of the vertical counter can be programmed:
- The noise rejection level for wrong vertical sync pulses (dmsd_vnoi, dmsd_vnoi_rst, dmsd_vnoi_max)
- Automatic or forced vertical frequency (50 or 60 Hz), (dmsd_aufd, dmsd_fsel)
- Forced odd/even toggle or odd/even toggle according the input signal (dmsd_foet)
- Start and stop position of the vertical sync pulses (dmsd_vsta, dmsd_vsto)
- Extra offset insertion for the internal vertical gating pulse (dmsd_vgps), bit not used in AVIP.

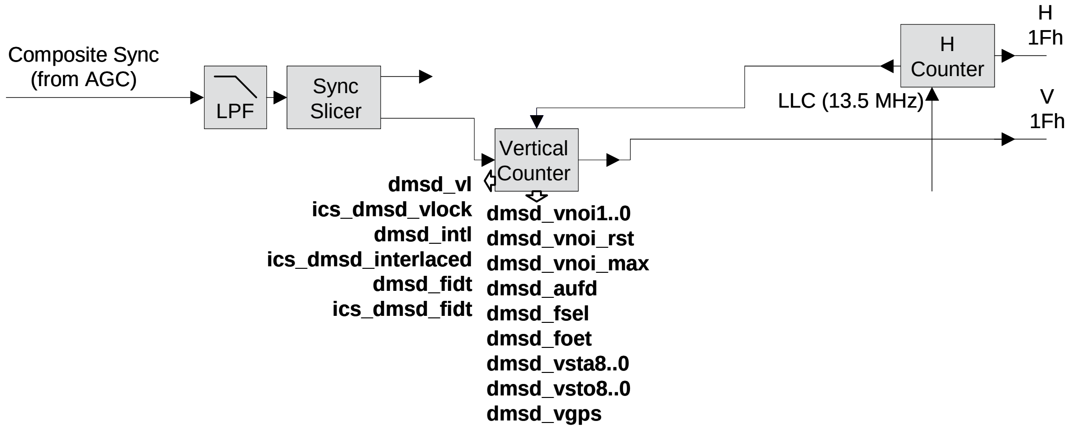


Fig 46.Vertical sync processing 1 Fh

Table 41: Status bit vertical sync - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
004	1	1 Fh: dmsd_vl	Vertical lock indication 0 No vertical lock 1 Vertical lock. Can take up to 27 fields.		R
FE0	1	1 Fh: ics_dmsd_vl	Interrupt flag set to 1 when dmsd_vl changes. See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		R
004	2	1 Fh: dmsd_intl	Indication input signal is interlaced 0 Not interlaced 1 Interlaced		R
FE0	2	1 Fh: ics_dmsd_intl	Interrupt flag set to 1 when dmsd_intl changes. See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		R
004	3	1 Fh: dmsd_fidt	Indicates the found field length or vertical frequency 0 50 Hz 1 60 Hz		R
FE0	3	1 Fh: ics_dmsd_fidt	Interrupt flag set to 1 when dmsd_fidt changes. See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		R

Remark: All readout bits have a latency of up to 50 msec. Take this latency into account when reading status bits after channel change or input source change.

dmsd_vl

The vertical lock indication can need up to 27 fields before indicating the correct status. Especially when the input signal changes from 50 to 60 Hz or vice versa, the lock time can be can be maximum. Changing from 50 Hz to 50 Hz or 60 Hz to 60 Hz, the lock time becomes much shorter. The status of dmsd_vl can be used to select the setting of the vertical noise suppression mode. It is possible to generate an interrupt when the status of dmsd_vl changes.

Remark: The reading of dmsd_vl is only reliable when dmsd_hl is 1, indicating horizontal lock. The reading of all other vertical status bits is only reliable when both dmsd_hl = 1 and dmsd_vl = 1.

dmsd_intl

The interlace bit indicates whether there is a half line shift with respect to the vertical retrace between two fields. All standard CVBS, Y/C and YPrPb signals are interlaced. The only exceptions are CVBS, Y/C and YPrPb from MPEG1 decoded video discs and the famous test pattern of a X-hatch. See also dmsd_foet and programming at the end of this chapter.

dmsd_fidt

The field length indicates whether the vertical frequency is 50 or 60 Hz. It is important to wait for dmsd_vl = 1 before reading this bit. The bit status also is needed to program the position of the vertical reference pulse (see bit description control bits below).

When changing from 50 to 60 Hz input signal or vice versa, the time before dmsd_fidt is valid may take 27 fields.

Table 42: Control bit vertical sync - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
184	1..0	1 Fh: dmsd_vnoi1..0	Vertical noise suppression mode 00 Normal mode, suppression of vertical sync pulses outside vertical window 01 Fast mode. Not used for AVIP 10 Free running mode on 50/60 Hz (pending on dmsd_fsel). Not used for AVIP 11 Debug mode	0/x	R/W
	2	1 Fh: dmsd_vnoi_rst	Vertical noise reduction reset 0 Normal mode 1 Reset to largest window. Should be used to enable fast vertical lock when changing source or channel	0/x	R/W
	3	1 Fh: dmsd_vnoi_max	Selects maximum vertical noise suppression 0 Normal mode (recommended) 1 Extreme high vertical noise suppression. Not useful in AVIP system.	0	R/W
	4	1 Fh: dmsd_aufd	Automatic field length detection (50/60 Hz) 0 Off, forced selection by dmsd_fsel (see below) 1 Active (Recommended for AVIP except NAFTA)	1/x [1]	R/W
	5	1 Fh: dmsd_fsel	Forced field length (only effective when dmsd_aufd = 0) 0 50 Hz (nominal 625 lines) 1 60 Hz (nominal 525 lines)	0/x [1]	R/W

Table 42: Control bit vertical sync - address 0X7FF9xxx *…continued*

add xxx	Bits	Name	Function	R/D	R/W
	6	1 Fh: dmsd_foet	Force odd / even toggle of First Field signal. 0 Only toggling when signal is interlaced 1 Interlaced signal: toggling each field, in phase Non-interlaced: toggling each field, phase random	1	R/W
	15..7	1 Fh: dmsd_vsta	Start position of vertical pulse VGATE_L Value pending on field length: 50 Hz: 2 60 Hz: 3	002/x	R/W
	24..16	1 Fh: dmsd_vsto	Stop position of vertical pulse VGATE_L Value pending on field length: 50 Hz: 12F hex 60 Hz: FE hex	12F/x	R/W
	25	1 Fh: dmsd_vgps	Influences the field offset for the start of the negative VGATE_L pulse. Recommended position: 0	0	R/W

[1] For North America and Philippines, where only 60 Hz signals are received, it is best to set aufd = 0 and fsel = 1.

dmsd_vnoi

Determines the vertical noise suppression mode. Set to 0 for normal operation. For fast vertical catching just after channel/input change or sync loss, it is possible to set to 1 and set back to 0 after vertical sync lock has been found (dmsd_hl and dmsd_vl = 1). Also for signals with varying vertical retrace position field by field (e.g. some VCR trick modes), position 1 may be used.

dmsd_vnoi_rst

When set to 1, the catch window is set to maximum value. This should be used to speed up vertical catching after changing channel/input signal or loss of sync. For fastest catching, should be combined by setting dmsd_vnoi = 1. Both dmsd_vnoi_rst and dmsd_vnoi should be set back to 0 when vertical sync lock has been found (dmsd_hl and dmsd_vl = 1).

dmsd_vnoi_max

Not applicable for TV use. Leave on default value 0.

dmsd_aufd, dmsd_fsel

Selects between automatic field length detection or forced field length. For all multi system countries, should be set to automatic (1 = reset value) Only when forcing color systems and for one color system only countries like USA, Canada, Mexico and Philippines the field frequency should be forced by setting dmsd_aufd to 0. Note that when forcing a color system, the forced field frequency is key to determine which color system is forced by dmsd_cstd (see Section 3.6.4.4) The field frequency in forced mode is selected by dmsd_fsel. When dmsd_aufd is in automatic mode, dmsd_fsel is don't care.

dmsd_foet

This bit is set to a 1 for AVIP. In this way, always an odd/even toggle is generated, also when the input is a non-interlaced signal. For interlaced signals, the phase of the odd/even signal will follow the input signal. For non-interlaced signals, the odd/even toggle changes in a random phase.

dmsd_vsta, dmsd_vsto

The setting of these registers for the internal vertical gating pulse is depending on the vertical frequency. This frequency can be read from dmsd_fidt. Note that this reading is only valid when dmsd_hl and dmsd_vl are 1.

dmsd_vgps

Not used in AVIP.

Programming

The bits of the vertical counter require quite some use and adaptations, depending on the readouts. Take into account the latency of the readout bits after channel/input change of 50 msec!

dmsd_vl - indicates that vertical lock is found, provided dmsd_hl = 1. All other vertical readout bits are only valid when both dmsd_hl and dmsd_vl are 1.

dmsd_intl - indicates whether the input signal is interlaced or not.

dmsd_fidt - indicates the field length, 0 = 50 Hz, 1 = 60 Hz. Values of dmsd_vsta and dmsd_vsto - should be adapted according to the found field frequency.

dmsd_vnoi and dmsd_vnoi_rst - should be set to 0 for normal operation.

After a channel / input change, to ensure fast catching, the best programming is:

Set dmsd_vnoi and dmsd_vnoi_rst = 1

When dmsd_hl and dmsd_vl are both 1, set dmsd_vnoi and dmsd_vnoi_rst back to 0.

dmsd_aufd is set to automatic (1) for multi-system destinations.

To force a color system:

- Set dmsd_aufd = 0
- Select the wanted field frequency with dmsd_fsel, 0 = 50 Hz, 1 = 60 Hz.
- Set dmsd_auto = 0
- Select the color standard using dmsd_cstd
- When dmsd_hl is 1 (note latency after channel / input change), set dmsd_cdto to 1 and back to 0.

For countries like USA, Canada and Mexico (NAFTA) and the Philippines, the TV can be forced to NTSC-M or NTSC-J only using the above procedure.

dmsd_foet

Should be programmed 1 always, is also the reset value.

dmsd_vsta, dmsd_vsto

To be programmed according to the found field frequency (dmsd_fidt, only valid when dmsd_hl and dmsd_vl are 1).

Table 43: Field frequency

Field Freq.	dmsd_fidt	dmsd_vsta	dmsd_vsto
50Hz	0	2	12F
60Hz	1	3	FE

Horizontal sync processing 2 Fh: The horizontal sync part for 2 Fh is for a large part identical to the 1 Fh part. The signals supported are 576p, 480p, ATSC 1080i, 1080i50Hz and 1152i50Hz either in Y Pr Pb format with in- or external sync or in RGB format with SOG (Sync On Green) or external sync.

The sync from the AGC passes a low pass filter to remove high frequency disturbances. The sync slicer does not have a noise detector, because it is not likely that 2 Fh input signals from DVD, PC or digital reception contain noise like 1 Fh terrestrial analogue reception.

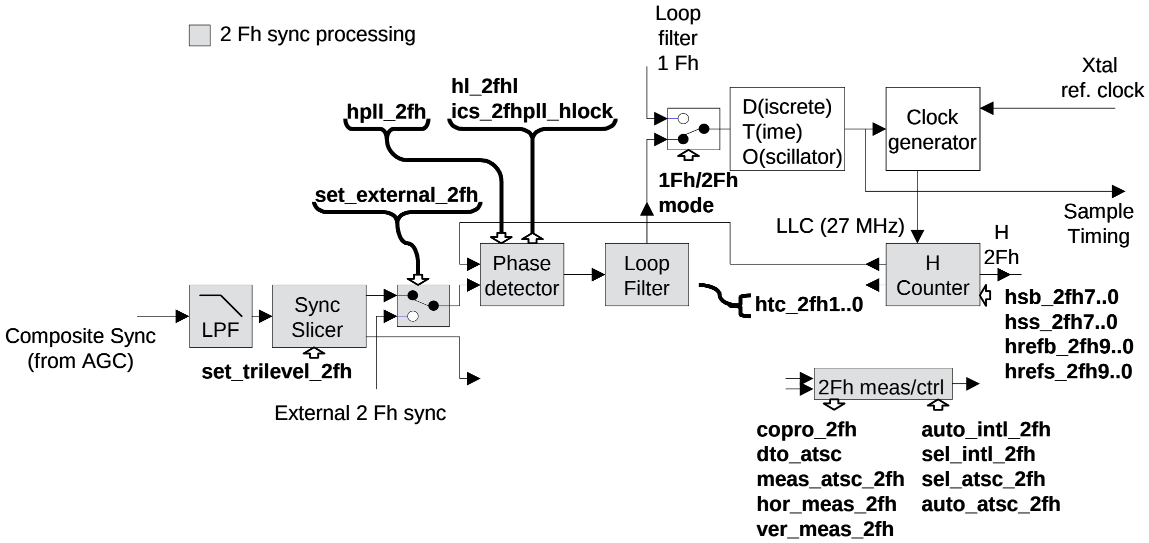


Fig 47.Horizontal sync processing 2 Fh

To accommodate ATSC, it is possible to set the sync slicer for tri-level sync on Y using set_3l_2fh. The 2Fh sync part also supports external sync input, selectable by sel_ext_2fh. Like for 1Fh, the PLL can be set in free running mode (hpll_2fh) and the lock status can be read (hl_2fh) The loop filter time constant can be set using htc_2fh. In these settings, the automatic time constant switching for VCR is omitted.

The settings hsb_2fh and hss_2fh for timing of internal sync pulses can be left on their reset value and need no change for all input signals. The value of hrefb_2fh and hrefs_2fh for the internal horizontal reference pulse should be adapted according the input signal.

The properties of the input signal can be read from the control/measurement block:

- Macrovision in sync (copro_2fh)

- Input signal is ATSC (dto_atsc, meas_atsc_2fh)
- The properties of the remaining two readout bits (hor_meas_2fh and ver_meas_2fh) are not defined.

Also it is possible to program the behavior

- Automatic or forced interlace (auto_intl-2fh, sel_intl_2fh)
- Automatic or forced setting for ATSC (auto_atsc_2fh, sel_atsc_2fh)
- Set 2fh Interlaced standard (atsc_mode_2fh)

Table 44: Horiz sync - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
140	10	2 Fh: set_3l_2fh	Selects tri-level mode for sync module 0 Normal mode 1 Tri-level mode, to be selected for ATSC	0/x	R/W
	11	2 Fh: sel_ext_2fh	Selects between internal sync (on Y) or external H and V sync inputs 0 Internal sync on Y 1 External sync on H and V sync inputs	0/x	R/W
144	21	2 Fh: hpll_2fh	Sets horizontal PLL in free running (Xtal based) mode 0 Normal operation (locked to the input signal) 1 Free running on 2 Fh	0	R/W
004	9	2 Fh: hl_2fh	Indicates that the horizontal PLL is in lock 0 No lock 1 In lock		R
FE0	8	2 Fh: ics_2fhpll_hlock	Interrupt flag set to 1 when hl_2fh changes. See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		R/W
144	20..19	2 Fh: htc_2fh	Horizontal Time Constant of the 2Fh PHI-1 loop 00 Slow mode for noise conditions 01 Normal mode with limited correction / line Preferred for standard condition 10 Reserved mode, do not use 11 Fast mode without limitation of correction/line	0/1	R/W
		1Fh / 2Fh mode	Signal, coming from the GPIO (General Purpose I/0) block, controlling also the clocks.		R/W
144	15..8	2 Fh: hsb_2fh	Horizontal Sync output pulse Begin position. Range is -107 to +107 in 8 pixels/step. Is fixed value for AVIP of FA.	FA	R/W
	7..0	2 Fh: hss_2fh	Horizontal Sync output pulse Stop position Range is -107 to +107 in 8 pixels/step. Fixed value for AVIP of FB	FB	R/W
14C		2 Fh: hrefb_2fh9..0	Begin position of horizontal reference pulse Value pending signal:		R/W
		2 Fh: hrefs_2fh9..0	Stop position of horizontal reference pulse Value pending signal:		R/W

set_3l_2fh

Y Pr Pb signals in ATSC format use a tri-level sync on Y. So when ATSC is detected, using the readout bits of the 2Fh measurement/control block, tri-level sync should be selected. For all other signals this bit should be set to 0.

sel_ext_2fh

Selects between sync on Y / G or external H and V sync.

hpll_2fh

Sets the 2Fh horizontal PLL in free running mode. For test purposes. Not used for AVIP.

hl_2fh

Indicates whether the horizontal 2Fh PLL is in lock. For the 2Fh circuit, the same applies as for the 1Fh. Only when hpll_2fh indicates lock (= 1), other status bit readouts are reliable.

Without horizontal lock, the VIDDEC is in an undefined state and readings cannot be trusted. Also the behavior regarding latency of status bits is comparable with the 1Fh circuit. A status change of hl_2fh can be signalled by an interrupt. Behavior hl_2fh for 1Fh/2Fh input signals in 2Fh mode of the VIDDEC. The bit hl_2fh can indicate false H-lock status for 1Fh signals while VIDDEC is in 2Fh mode. See CVI input selection

htc_2fh

Sets the time constant for the 2Fh horizontal PLL.

Below, the differences with the 1Fh PLL time constant settings are discussed.

00

Slow setting for signals with noise. In 2Fh mode, these signals (analogue off air reception) are not present. Therefore also the noise measurement module has been omitted in the 2Fh PLL part.

01

Normal mode with limited phase correction / line. Preferred mode when horizontal lock time is acceptable. Guarantees also stable behavior for no input signal conditions and disturbances on the input.

10

Not defined. Do not use. (For 1 Fh this setting provided the automatic switching between position 3 (VCR) and 0 (broadcast, stable source) depending on readout of dmsd_tvdet)

11

Fast mode, no gating or limitation for phase correction per line. May be used when the horizontal lock time in mode 01 is too long.

1Fh/2Fh mode

hsb_2fh, hss_2fh

Timing of internal horizontal sync pulse. Not used in AVIP. Leave on reset value.

hrefb_2fh, hrefs_2fh

Horizontal reference pulse. These settings should be programmed according to the found input format. The input format can be deducted from the status bit readings.

Table 45: Horizontal input format

Signal	Input Format	hrefb 2fh	hrefs 2fh	hsb_2fh	hss_2fh
YPrPb	480p	3A7	31	7	4
	576p	3A1	31	7	4
	1080i (ATSC)	3BE	28	7	4
	1080i/50Hz	3BE	3CC	7	4
	1152i/50HZ	39F	28	5	0
RGB Ext. pos Sync	480p	37D	7	7	4
	576p	375	5	7	4
	1080i (ATSC)	393	3FD	7	4
	1080i/50Hz	393	39D	7	4
	1152i/50HZ	371	1	5	0
RGB Ext. neg sync	480p	3B9	3B	7	4
	576p	3B5	45	7	4
	1080i (ATSC)	3B3	1D	7	4
	1080i/50Hz	3B1	3B9	7	4
	1152i/50HZ	3AF	3F	5	0

Programming

Sel_trilevel_2fh

Should be set when ATSC standard is detected (see bit description measurement / control below) and there is no external sync.

Sel_ext_2fh

Should be set according to the source for sync signals to be checked. Also in the 2Fh circuit, the bit hl_2fh is key. Without horizontal lock no other status bits are reliable. Take care of the latency of 50 msec. after loss of sync when changing channel / input.

htc_2fh

When horizontal lock-in time is acceptable, set to 01, else use 11.

Set hrefb_2fh and hrefs_2fh according to the table, depending on the format of the input signal. All other bits can be left in reset state.

Table 46: Measurement /control - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
140	17	2 Fh: auto_intl_2fh	Sets sync loop automatic for interlaced mode when interlace is detected by intl_2fh 0 Interlace / non-interlace set by sel_intl 1 Interlace set by status of intl_2fh	0	R/W
	18	2 Fh: sel_intl_2fh	Sets interlaced / non-interlaced mode when auto_intl_2fh = 0 0 Non-interlaced 1 Interlaced	0	R/W
	19	2 Fh: sel_atsc_2fh	Sets H counter / V window according ATSC norm 0 Normal mode 1 ATSC mode Not active when auto_atsc_2fh = 1, see below	0	R/W
	20	2 Fh: auto_atsc_2fh	Sets automatic normal / atsc mode, pending on dto_atsc (meas_atsc_2fh) Only controls H counter / V window, not tri-level sync!! 0 use mode, set by sel_atsc_2fh 1 Use value of dto_atsc (meas_atsc_2fh)	0	R/W
004	12	2 Fh: copro_2fh	Is set to 1 when Macrovision is detected on the sync.		R
FE0	11	2 Fh: ics_2fhpll_copro	Interrupt flag set to 1 when copro_2fh changes. See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		
004	13	2 Fh: dto_atsc	Detection of ATSC signal		R
	20	2 Fh: meas_atsc_2fh	Alternative detection of ATSC signal, validation to prove which is the most reliable		R
140	22..21	2 Fh: atsc_mode_2fh	00 US1 01 US2 10 Australia 11 China	0	R/W
FE0	18	2 Fh: ics_dto_atsc	Interrupt flag set to 1 when dto_atsc changes. See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		R
004	22..21	2 Fh: hor_meas_2fh	Not defined yet		R
	25..23	2 Fh: ver_meas_2fh	Not defined yet		R

auto_intl_2fh

Should be set to 0 to force the interlace status. The automatic mode can cause instabilities while catching the signal.

sel_intl_2fh

Set according to the properties of the input signal. The status bits used for determining the interlace of the input signal are intl_2fh, dto_atsc and meas_atsc_2fh. These bits are discussed below. For further info, see Programming.

auto_atsc_2fh, sel_atsc_2fh

It is recommended to set the automatic selection for ATSC to forced mode (ato_atsc_2fh = 0). To determine whether the input signal is ATSC, the status bits dto_atsc, meas_atsc_2fh and intl_2fh can be used.

copro_2fh

Indicates when Macrovision is detected in the sync. Also the 2Fh signals can contain Macrovision in the Y signal, as well as on the sync (80% sync amplitude and false sync pulses during vertical retrace) as false AGC pulses during the vertical retrace When copro_2fh = 1, indicating Macrovision, the setting of the agc_y_cyc_ctrl_top_sync_pi should be adapted, identical to the 1Fh case (dmsd_copro).

For YUV:

Normal agc_y_cyc_ctrl_top_sync_pi = 100 hex.

Copro_2fh = 1 agc_y_cyc_ctrl_top_sync_pi = 118 hex.

There is no Macrovision for RGB signals. A status change of copro_2fh can be signalled by an interrupt.

dto_atsc, meas_atsc_2fh

Two measurement circuits are built-in to detect whether the input signal is ATSC. It is recommended to have both indicating ATSC and to ensure that intl_2fh also indicates an interlaced signal before deciding to set sel_atsc_2fh in ATSC mode.

Note that intl_2fh only is reliable when vl_2fh indicates vertical lock, see ‘Vertical sync processing 2 Fh’ below. A status change of dto_atsc can be signalled by an interrupt.

hor_meas_2fh, ver_meas_2fh

These bits are reserved but have not been defined in the design. Do not use in AVIP.

Programming

sel_atsc_2fh, sel_intl_2fh, auto_atsc_2fh and auto_intl_2fh are discussed in the ‘programming’ part of the next section.

Set both auto_atsc_2fh and auto_intl_2fh in forced mode (0)

copro_2fh detects when there is Macrovision in the sync part. When Macrovision is detected (copro_2fh = 1), adapt the AGC setting for the Y channel:

For YUV:

Normal agc_y_cyc_ctrl_top_sync_pi = 100 hex.

Copro_2fh = 1 agc_y_cyc_ctrl_top_sync_pi = 118 hex.

Vertical sync processing 2 Fh: The vertical sync processing for 2Fh is almost identical to the one for 1Fh.

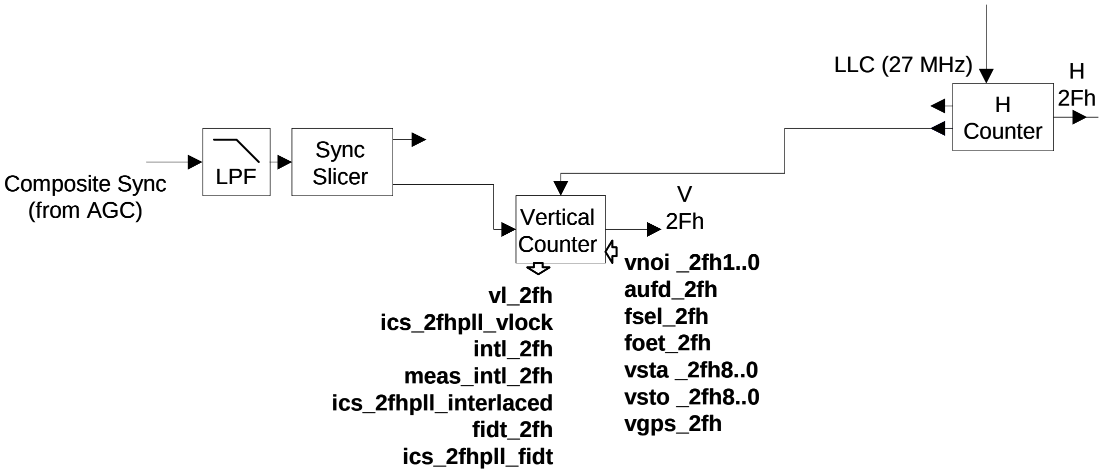


Fig 48.Vertical sync processing 2Fh

The following status bits are available:

- Vertical lock (vl_2fh). Both vertical lock and horizontal lock have to be true (vl_2fh and hl_2fh both 1) to guarantee that the other vertical status bits are reliable.
- Interlace (intl_2fh and meas_intl_2fh).
- 50 or 60 Hz field frequency (fidt_2fh).

It is possible to generate an interrupt when one of the status bit changes.

In the properties, which can be programmed, the option to reset the vertical search window to maximum and to select a very high vertical noise level suppression have been removed for 2Fh. In view of the signal properties of 2Fh sources, they were not relevant. The programmable items are:

- Noise suppression (vnoi_2fh)
- Automatic or forced field frequency (aufd_2fh, fsel_2fh)
- Automatic or forced odd/even toggle (foet_2fh)
- Position vertical reference pulse (vsta_2fh, vsto_2fh)
- Extra offset insertion for the internal vertical gating pulse (vgps_2fh), bit not used in AVIP.

The vertical blanking for 2Fh signals is not according the specification of the 576p, 480p and 1080i standard.

Table 47: Status bit - address OX7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
004	10	2 Fh: vl_2fh	Vertical lock indication 0 No vertical lock 1 Vertical lock		R
FE0	9	2 Fh: ics_2fhpll_vlock	Interrupt flag set to 1 when vl_2fh changes. See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		R
004	11	2 Fh: intl_2fh	Indication input signal is interlaced 0 Not interlaced 1 Interlaced		R
004	19	2 Fh: meas_intl_2fh	Alternative indication input signal is interlaced 0 Not interlaced 1 Interlaced Practice should prove which indication is most reliable		R
FE0	10	2 Fh: ics_2fhpll_interlaced	Interrupt flag set to 1 when intl_2fh changes. See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		R
004	16	2 Fh: fidt_2fh	Indicates the found field length or vertical frequency 0 50 Hz 1 60 Hz		R
FE0	17	2 Fh: ics_2fhpll_fidt	Interrupt flag set to 1 when fidt_2fh changes. See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		R

vl_2fh

The vertical lock status bit indicates that the vertical counter is synchronized with the incoming signal. The reading of vl_2fh is only reliable when also hl_2fh = 1. Both hl_2fh and vl_2fh have to be 1 before the other vertical status bits are reliable. The vertical lock-in time is comparable with 1Fh, it can take up to 27 fields. (more than 500 msec). The longest times will occur when the VIDDEC starts in 50 Hz after switching to 2Fh and a reset of all registers and the input signal is 60 Hz or vice versa. When the incoming frequency is known before and the system is forced to that frequency (see aufd_2fh, fsel_2fh) the lock time is shorter. The same latency of 50 msec is expected (keeping the last reading) as for 1Fh when switching input source, but this is less relevant for 2Fh input. A status change can be signalled by an interrupt.

intl_2fh, meas_intl_2fh

Two measurement circuits have been built-in to check whether the input signal is interlaced or not. Validation has to prove whether which status bit is the most reliable. It is recommended to use both bits together to determine the interlace status. A status change of one of the interlace status bits, intl_2fh, can be signalled by an interrupt.

fidt_2fh

The field length (50 or 60 Hz) is important to program the correct settings for the position of the vertical reference pulse (vsta_2fh, vsto_2fh), see below. The bit is also used in the recognition of ATSC. A status change can be signalled by an interrupt.

Table 48: Control bit vert sync - address OX7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
144	23..22	2 Fh: vnoi_2fh1..0	Vertical noise suppression mode 00 Normal mode, suppression of vertical sync pulses outside vertical window 01 Fast mode. Can speed vertical catching 10 Free running mode on 50/60 Hz (pending on fsel_2fh). Not allowed for AVIP 11 Debug mode, not allowed for normal operation	0	R/W
	16	2 Fh: aufd_2fh	Automatic field length detection (50/60 Hz) 0 Off, forced selection by fsel_2fh (see below) 1 Active (Recommended for AVIP except NAFTA)	1/x 1)	R/W
	17	2 Fh: fsel_2fh	Forced field length (only effective when aufd_2fh = 0) 0 50 Hz (576 progressive) 1 60 Hz (480 progressive)	0/x	R/W
	18	2 Fh: foet_2fh	Force odd / even toggle of First Field signal. 0 Only toggling when signal is interlaced 1 Interlaced signal: toggling each field, in phase Non-interlaced: toggling each field, phase random	0	R/W
148	9..0	2 Fh: vsta_2fh9..0	Start position of vertical pulse VGATE_L Value pending on signal:	4	R/W
	19..10	2 Fh: vsto_2fh9..0	Stop position of vertical pulse VGATE_L Value pending on signal:	209	R/W
140	12	2 Fh: vgps_2fh	Influences the field offset for the start of the negative VGATE_L pulse. Recommended position: 0	0	R/W

vnoi_2fh

Set to 0 for normal use. For fast vertical catching, it is possible to set vnoi_2fh to 1 after selecting 2Fh mode for the VIDDEC till there is vl_2fh (note that also hl_2fh has to be 1)

Remark: To ensure stable behavior, settings 2 and 3 are not allowed.

For 2Fh, some settings for vertical noise suppression have been omitted. The option to reset the search window to maximum size for fast catching after input change, is not available. As two 2Fh inputs are unlikely, this is not a problem in practice. Also a setting for maximum noise rejection is omitted, but this setting was not useful for TV application.

aufd_2fh, fsel_2fh

The vertical frequency can follow the input signal or can be forced to 50 or 60 Hz only. For most countries, the automatic setting is optimal. For countries with only one TV system like USA, Canada, Mexico and Philippines, the vertical frequency can be forced to 60 Hz only. Forcing to one frequency can speed up the vertical catching time. The use is identical to the 1Fh bits. For automatic, set aufd_2fh to 1. For forced, set aufd to 0 and set fsel_2fh = 0 for 50 Hz and to 1 for 60 Hz.

foet_2fh

For 2Fh signals, the bit foet_2fh has to follow the interlace status of the incoming signal. The setting becomes:

- 0 for 576p, 50 Hz
- 0 for 480p, 60 Hz
- 1 for ATSC, 1080i, 60 Hz

vsta_2fh, vsto_2fh

The position of the vertical reference pulse has to be programmed according to the format of the input signal:

Table 49: Vertical input format

Signal	Input format	vsto_2fh	vsta_2fh
YPrPb	480p	209	4
	576p	26D	3
	1080i (ATSC)	22F	4
	1080i/50Hz	22F	4
	1152i/50HZ	22F	4
RGB Ext. pos Sync	480p	209	4
	576p	26D	3
	1080i (ATSC)	22F	4
	1080i/50Hz	22F	4
	1152i/50HZ	22F	4
RGB Ext. neg sync	480p	209	4
	576p	26D	3
	1080i (ATSC)	22F	4
	1080i/50Hz	22F	4
	1152i/50HZ	22F	4

vgps_2fh

Not used in AVIP.

Programming

Both hl_2fh and vl_2fh have to be 1 to guarantee that the other vertical status bits are reliable.

The lock time of vl_2fh can run up to 27 fields.

Forcing the vertical frequency (for one TV system countries like USA, Canada, Mexico and Philippines) can shorten the vertical lock-in time. vnoi_2fh should be set to 0 for normal operation. For fast vertical catching, it is possible to set vnoi_2fh = 1 when there is no horizontal sync lock and put it back to 0 when there is both hl_2fh and vl_2fh.

- Set aufd_2fh = 0

  For most countries, the vertical field frequency is set to automatic (aufd_2fh = 1). For one TV system countries (USA, Canada, Mexico and Philippines) it is possible to force the vertical frequency.
- Select the desired vertical frequency setting fsel_2fh = 0 for 50 Hz or 1 for 60 Hz.

In this part, also the control bits auto_atsc_2fh, sel_atsc_2fh, auto_intl_2fh, and sel_intl_2fh and the status bits dto_atsc and measatsc_2fh of the 2Fh measurement/control block is discussed. As indicated in the previous chapter, set the selection mode for ATSC and interlace to forced.

auto_atsc_2fh = 0

auto_intl_2fh = 0

For the setting of the other bits, depending on the format of the input signal and the readout of the status bits, see the table below:

Table 50: Setting of bits

Input	hl	vl	fidt	intl	meas	dto	meas	foet	sel	sel	sel	vsta	vsto
	2fh	2fh	2fh	2fh	intl	atsc	atsc	2fh	atsc	trilevel	intl	2fh	2fh
					2fh		2fh		2fh	2fh	2fh		
R R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
576P	1	1	0	0	0	0	0	0	0	0	0	3	26D
480P	1	1	1	0	0	0	0	0	0	0	0	4	209
1080i ATSC	1	1	1	1	1	1	1	1	1	1	1	4	22F
1080i/ 50Hz	1	1	0	1	1	1	1	1	1	1	1	4	22F
1152i/ 50Hz	1	1	0	1	0	0	0	1	1	0	1	4	22F

Fast blanking / external 2 fh sync / clamp info: The circuit combines three functions:

- The fast blanking input for 1Fh mode
- The external H and V inputs for 2Fh mode
- Timing information for the clamping in the PNX3000 (HVinfo)

The Fast blanking input for 1Fh shares the pins with the inputs for Horizontal Sync in 2Fh mode.

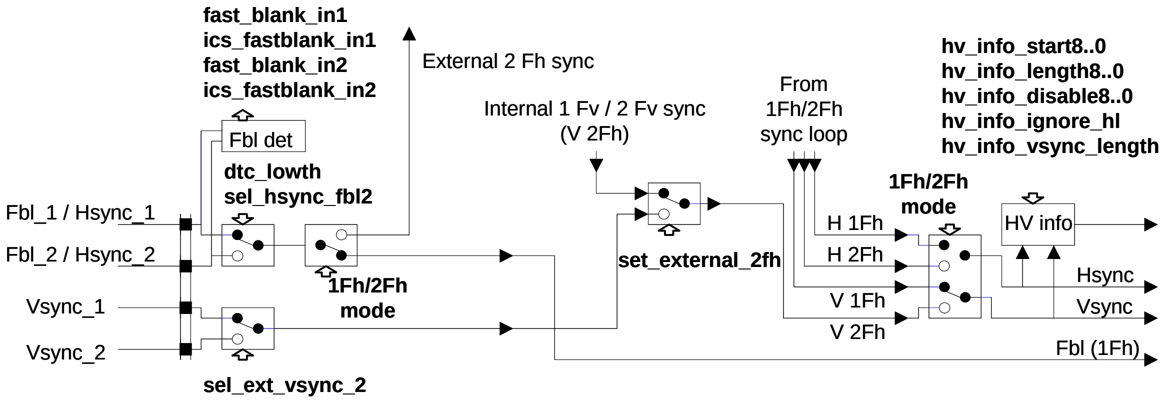


Fig 49.Fast blanking, external 2 Fh composite sync input and clamp info

Fast blanking

In 1Fh mode, the Fast Blanking input controls in VIDDEC switch between the output of the color decoder (decoded CVBS or Y/C) and CVI input direct from PNX3000. It is possible to select one of the two Fast Blanking inputs with sel_hsync_fbl2. The slicing level of the input can be set on 1.65 Volt or 0.65 Volt. To fulfil the Fast Blanking spec for SCART (insertion for > 0.9 Volt), 0.65 volt slicing level should be selected. The status of both Fast Blanking inputs can be read (fast_blank_in1/2). A change in status can be signalled by an interrupt.

2Fh H and V input

In 2 Fh mode, the two Fast Blanking Inputs are configured as inputs for external Horizontal Sync. Also for these signals, the desired input can be selected with sel_hsync_fbl2. The slicing level is also set to 1.65 or 0.65 Volt, depending on dtc_lowth.

There are also two vertical inputs. Selection between the inputs is made by sel_ext_vsync_2. Selection in 2Fh mode between internal sync on Y and external sync is controlled by sel_ext_2fh for both horizontal and vertical sync.

HV info

The HVinfo block delivers the timing for correct clamping in the PNX3000 for the incoming CVBS, Y/C and CVI signals. The registers have to be programmed according to the signal property (1Fh, 2Fh, ATSC, vertical frequency).

Table 51: Fast blanking - address OX7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
004	17	fast_blank_in1	Reflects level of Fbl_1 input during vertical retrace Can be used to determine full RGB insertion on SCART		R
FE0	19	ics_fastblank_in1	Interrupt flag set to 1 when fast_blank_in1 changes See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		
004	18	fast_blank_in2	Reflects level of Fbl_2 input during vertical retrace Can be used to determine full RGB insertion on SCART		R
FE0	20	ics_fastblank_in2	Interrupt flag set to 1 when fast_blank_in2 changes See AGC part for explanation of the related interrupt control bits _enab, _clr and _set		
040	8	dtc_lowth	Selects the slicing level on the Fast Blanking/2Fh_Hsync inputs 0 Slicing level 1.65 Volt 1 Slicing level 0.65 Volt	1	R/W
0C0	4	sel_hsync_fbl2	Selects between the two Fbl / Hsync inputs: 0 Selects Fbl_1 / Hsync_1 1 Selects Fbl_2 / Hsync_2	0	R/W
040	7	sel_ext_vsync_2	Selects between the two Vsync inputs 0 Selects Vsync_1 1 Selects Vsync_2	0	R/W
		1Fh / 2Fh mode	Signal, coming from the GPIO (General Purpose I/0) block, controlling also the clocks Selects also the function from the Fbl / Hsync pins: 1 Fh: Fbl input 2 Fh: Hsync input.		
140	11	2 Fh: sel_ext_2fh	Selects between internal sync (on Y) or external H and V sync inputs 0 Internal sync on Y 1 External sync on H and V sync inputs	0/x	R/W
100	8..0	hv_info_start8..0	Feedback signal to PNX3000 for black level clamping Start position. Value pending on signal: All 1 Fh: 060 576p 2B 480p 2C ATSC 4B 1080i 50Hz 4 1152i 50Hz 2A	060/x	R/W

Table 51: Fast blanking - address OX7FF9xxx *…continued*

add xxx	Bits	Name	Function	R/D	R/W
104	8..0	hv_infolength8..0	width horizontal timing pulse. Value pending on signal: All 1 Fh: 048 576p 024 480p 024 ATSC 4B 1080i 50Hz 14 1152i 50Hz 24	048/x	R/W
108	8..0	hv_info_disable8..0	width during lines with clamp disabled. Value pending on signal : All 1 Fh: 080 576p 040 480p 040 ATSC 2C 1080i 50Hz 2C 1152i 50Hz 2C	80	R/W
	24	hv_info_ignore_hl	Set to 1 for all modes	1	R/W

Fast_Blank_in1/2

Reflects the status of the fast blanking input. When the input is above the slicing level set by **dtc_lowth** during vertical retrace, the bit is set to 1.

Fast Blanking is part of the SCART spec.

- A "low" level (< 0.3 Volt) on the Fast Blanking pin of the SCART connector connects the decoded CVBS to the display.
- A "high" level on the Fast Blanking pin (> 0.9 Volt) of the SCART connector should connect the RGB signals on the SCART connector to the display.

In both modes, the sync is always derived from the CVBS signal, CVBS and RGB should coincide.

The Fast Blanking input on the SCART is used for:

- Insertion of RGB for OSD (e.g. menus for control of external connected set top boxes) Fast Blanking only pulled high during insertion of the OSD itself.
- Insertion of RGB full screen for high quality pictures (e.g. DVD players in Europe) Fast Blanking pulled high continuously.

To distinguish between Full Insertion and OSD insertion, the level on the Fast Blanking pins is only checked during vertical retrace, which indicates full insertion.

Remark: Both inputs can be monitored independent of the selected Fast Blanking input. A change in status of both bits can be signalled by a separate interrupt. Only valid in 1Fh mode.

dtc_lowth

Works in both 1Fh and 2Fh mode. Selects the slicing level of the Fast Blanking / External Horizontal Sync inputs. The available levels are 1.65 Volt (dtc_lowth = 0) and 0.65 Volt (dtc_lowth = 1). To fulfil SCART norm (see above) in 1Fh, the slicing level should be below 0.9 Volt, so dtc_lowth should be set to 1 For External Horizontal Sync input in 2Fh mode, selection depends on the input signals. Usually the "low" level of an external sync should be below 0.5 Volt, so the slicing level can remain on 0.65 Volt.

sel_hsync_fbl2

Selects between the two Fast Blanking (1Fh) / External Horizontal Sync (2Fh) inputs.

sel_ext_vsync_2

Selects between the two External Vsync (2Fh) input pins. Is independent from the selection for the Horizontal Sync inputs.

1Fh/2Fh mode

See Section 3.6.4.7.

sel_ext_2fh

Selects in 2Fh mode between horizontal and vertical sync, derived from the sync on Y and external H and V sync.

HV info block

For proper Analogue to Digital conversion, it is required that the input signals in PNX3000 are clamped on black level. The HV info block provides the timing signals derived from the horizontal PLL and vertical counter to set the correct clamping timing in the PNX3000 for the CVBS, Y/C and CVI signals. The timing is coded in an analogue serial protocol, the properties are:

- Start of the clamping pulse for horizontal clamping
- Length of the horizontal clamping pulse
- No valid clamping info available (no horizontal lock)
- Vertical retrace, no proper black level clamping possible.

The information for disabling clamping and vertical retrace is coded in the length of the pulse after start of the clamping pulse. The HV_info signals can be measured at the appropriate pins of the AVIP and PNX3000.

The bit hv_info_ignore_hl is for test purposes and should be set to 1.

Programming

1Fh mode

For full SCART input, the fast YUV switch in the VIDDEC should be enabled to follow the signal on Fast Blanking input. (fbl_switch_ctrl, see next paragraph). For details of the set-up and programming see full SCART input.

CVBS + RGB insert via SCART.

The correct Fast Blanking input is selected using sel_hsync_fbl2(0= input 1, 1 = input 2). The slicing level should be set to 0.65 volt, dtc_lowth = 1 (= reset value) The reading of the Fast Blanking inputs (Fast_Blank_in1/2) can be used to check when full RGB insertion is performed, to adapt the settings in the processing chain for a good quality high bandwidth noise free input signal.

2Fh mode

Select for internal sync on Y or external H and V sync using sel_ext_2fh (0 = internal, 1 = external) In case of external H and V sync, select the correct H and V inputs using sel_hsync_fbl2 (0 = input 1, 1 = input 2) and sel_ext_vsync_2 (0 = input1, 1 = input 2).

HV_info

The bits hv_info_start, hv_info_length, hv_info_disable and hv_info_vsync_length Should be set according to the properties of the found input signal. (see table in bit description) For all 1Fh signals, there is one setting. For 2 Fh signals, different settings are needed for 576p, 480p and 2Fh Interfaced Modes. The bit hv_info_ignore_hl should be kept 1 (= reset value).

YUV switch + formatter: The YUV switch is a fast switch, which can switch on pixel base between the internal YUV from the color decoder and the external YUV from the CVI inputs in the PNX3000.

The switch can be controlled by an external signal on the Fast Blanking input or forced by software to display YUV from the color decoder or the external YUV from the CVI inputs from PNX3000 (fbl_switch_ctrl) The polarity of the Fast Blanking signal can be set using fbl_polarity.

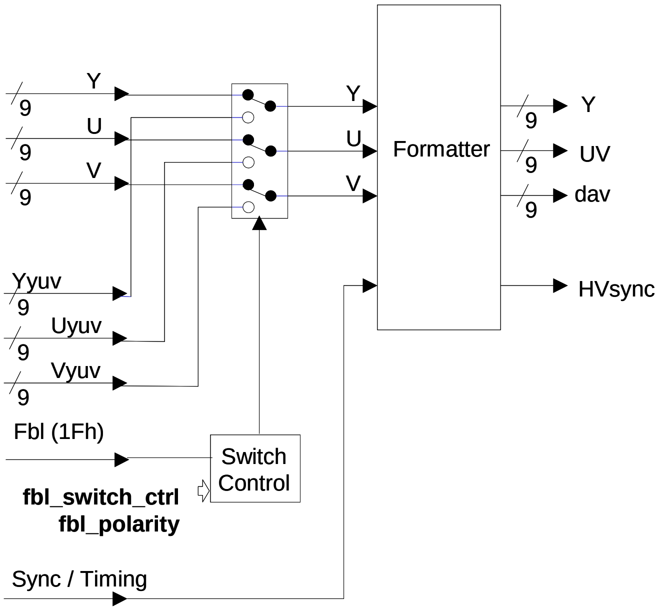


Fig 50.YUV switch and formatter

Table 52: YUV switch - address 0X7FF9xxx

add xxx	Bits	Name	Function	R/D	R/W
0C0	11..10	fbl_switch_ctrl	Fast Blanking switch control 00 Fast blanking input controls the switch 01 Switch forced to YUV output of color decoder 10 Switch forced to external YUV input signal 11 Fast blanking input controls the switch	0/x	R/W
	12	fbl_polarity	Determines the polarity of the Fast Blanking input 0 Fast Blank input low switches to external YUV 1 Fast Blank input high switches to external YUV	1	R/W

fbl_switch_ctrl

Different settings are needed for different input signals conditions.

00

The switch is controlled by the Fast Blanking signal, connected to the selected Fast Blanking input. This mode should only be enabled when a full SCART socket is selected with RGB insertion capability. The Fast Blanking input should be selected where the Fast Insertion pin of the SCART connector is connected to. The selected CVBS input should be connected to the CVBS input pin of the same SCART connector, providing the sync for full RGB insertion or the picture with via RGB inserted OSD. When the SCART input is no longer selected, the fbl_switch_control should not be kept 00, because in that case the Fast Blanking signal on that SCART still controls the YUV switch. Because there are two Fast Blanking inputs, it is possible to have two Full Scart configurations with their own Fast Blanking input. Note that in this case no input is left over for an external 2Fh horizontal sync

01

Normal setting when CVBS or Y/C signal is connected. Forces the switch to display the decoded YUV from the color decoder.

10

Forces the switch to display the YUV signals coming from the CVI inputs of the PNX3000. Setting for YPrPb input signals without a Fast Blanking signal to control the switch. Also the setting for all 2Fh CVI input signals (YPrPb and RGB with internal or external sync).

11

Same as 00, do not use.

fbl_polarity

For flexibility. When working according to SCART norm, should be kept 0.

Programming

Program fbl_switch_ctrl according to the selected input. For details of full SCART use, see CVBS + RGB insert via SCART. For different designs, it might be useful to use the reverse polarity of the fast blanking input using fbl_polarity.

3.6.4.7 Switching VIDDEC between 1Fh and 2Fh

The VIDDEC can be switched in 1Fh and 2Fh mode.

In 1 Fh mode, the functionality available is:

- CVBS
- Y/C
- CVBS + RGB insert
- YPrPb, sync on Y

In 2Fh mode, the functionality available is:

- YPrPb with sync on Y or external H/V sync (576p, 480p, 2Fh Interfaced Modes)
- RGB with Sync On Green (SOG) or external H/V sync (576p, 480p)

To switch the VIDDEC from 1Fh to 2Fh or vice versa, a clock frequency outside the VIDDEC has to be doubled (or halved) and all registers of the VIDDEC have to be reset. The reset is needed to ensure a reliable working of the VIDDEC after clock switching. This means that after switching 1Fh/2Fh mode, all registers with a value, different from the reset value, have to be initialised by software.

The control registers for switching and resetting the VIDDEC are located in the GPR (General Purpose Registers) block of the AVIP.

Table 53: 1Fh/2Fh switching - address OX7FF7xxx

add xxx	Bits	Name	Function	R/D	R/W
010	All	gp_vidclksel	Selects clock frequency for the blocks BEF, primary and secondary VIDDEC		R/W
	1	gp_dtofreqsel_vid	Selects 1Fh or 2Fh mode for primary VIDDEC 0. 1 Fh mode (clock = 27 MHz) 1. 2 Fh mode (clock = 54 MHz)	0	R/W
064	All	gp_vid_resets	Controls reset of all Video Core blocks R/W		
	12	gp_vid_reset_vd1_n	Resets the primary VIDDEC block (Active low) 0. Reset 1. Normal mode		

Hints to switch 1Fh/2Fh mode: Because the switching of mode resets all registers, first all registers which contain User Control settings which cannot be derived from the planned signal source or signal properties have to be saved.

The only practical value for 1Fh is the customer HUE setting (dmsd_huec). For 2Fh, it is not expected that customer preferences have to be saved for VIDDEC. Because the control bits are only 1 of the 32 bits of the register, first the register has to be read to ensure that all other bits are not changed.

To switch the clock:

- First force the VIDDEC in reset by setting gp_vid_reset_vd1_n = 0
- Switch clock by setting gp_dtofreqsel_vid1 (0 for 1Fh, 1 for 2Fh)
- End the reset by writing gp_vid_reset_vd1_n = 1

After releasing the reset all registers needing a value, different from the reset value, have to be programmed by software.

Only a few registers need programming, because their default value is different from the reset value. The value for most registers is determined by the properties of the selected channel or by the properties of the detected input signal. Finally, some registers have to be programmed according to the diversity settings. Depending on the use of interrupts, the interrupt section has to be cleared and enabled, else these registers can maintain their reset value and need no programming.

System aspects: Besides the VIDDEC, the whole video path in AVIP and PNX3000 have to be set in 2Fh mode. The most important items are:

- PNX3000, Selection of the mode bit (0 = 1Fh mode, 1 = 2Fh mode)
- I2D receiver, Selection of the receiver mode, demux_mode2..0 (0 / 1 = 1Fh mode 0a / 0b 2 = 2Fh mode)
- PNX8550

Though this list is certainly not complete, it helps to get at least the path in such mode, you can see a picture on the screen. For more info about selecting 1Fh and 2Fh sources, see also CVI input selection.

3.6.4.8 Use of interrupt bits

Most status bits are coupled to interrupt bits. When enabled, these interrupt bits can generate an interrupt each time the status bit changes. To control the interrupt behavior, for each interrupt one bit in 4 registers is available.

Table 54: Interrupt bits

Name	Function	R/W
ics_xxxx (xxxx = register name	Indicates that the status of the corresponding status register is changed, or from 0 ->1 or from 1->0. 0. No change 1. Changed	R
ics_xxxx_enab	Enables the interrupt from the corresponding status register	R/W
ics_xxxx_clr	Sets the corresponding interrupt bit to 0	
ics_xxxx_set	Forces the corresponding interrupt bit to 1 Meant for testing. Can be used to check whether the software reacts correct upon generating an interrupt.	R/W

The software can use the interrupt bits in two ways:

Interrupt driven structure: The interrupts of the relevant status bits are enabled using ics_xxx_enab. When an interrupt is received, the interrupt registers are read to find which status bit generated the interrupt (ics_xxx) When the interrupt source is found, software can execute the needed activities.

Remark: The interrupt bit should be cleared by software using the corresponding ics_xxx_clr bit, it is not cleared by reading. After clearing and serving the interrupt the software can continue its main loop.

Polled structure: Also when the interrupts are not used, the interrupt bits still can fulfil a useful function. When polling the status bits every XX msec, it is possible that a short pulse is not detected. By reading the corresponding interrupt bit (ics_xxx), the interrupt bit will tell whether the status has changed of the corresponding status bit after the last clearance. In this way, the interrupt bit can fulfil a latch function.

3.6.4.9 Automatic selection of different input signal formats

This section describes the methods used to distinguish automatically between different input signal formats. There are three different use cases, which can be combined when appropriate.

- CVBS - Y/C input
- CVBS - RGB input
- CVI (Component Video Input) signals like YPrPb 1 Fh, YPrPb 2 Fh, RGB + external H/V sync 2 Fh, ATSC with tri-level sync on Y

Each case is discussed in the sections on the following pages. The last use case (CVI) still needs to be updated after measurements with AVIP to optimize the correct recognition of 1Fh and 2Fh input signals.

CVBS or Y/C input selection

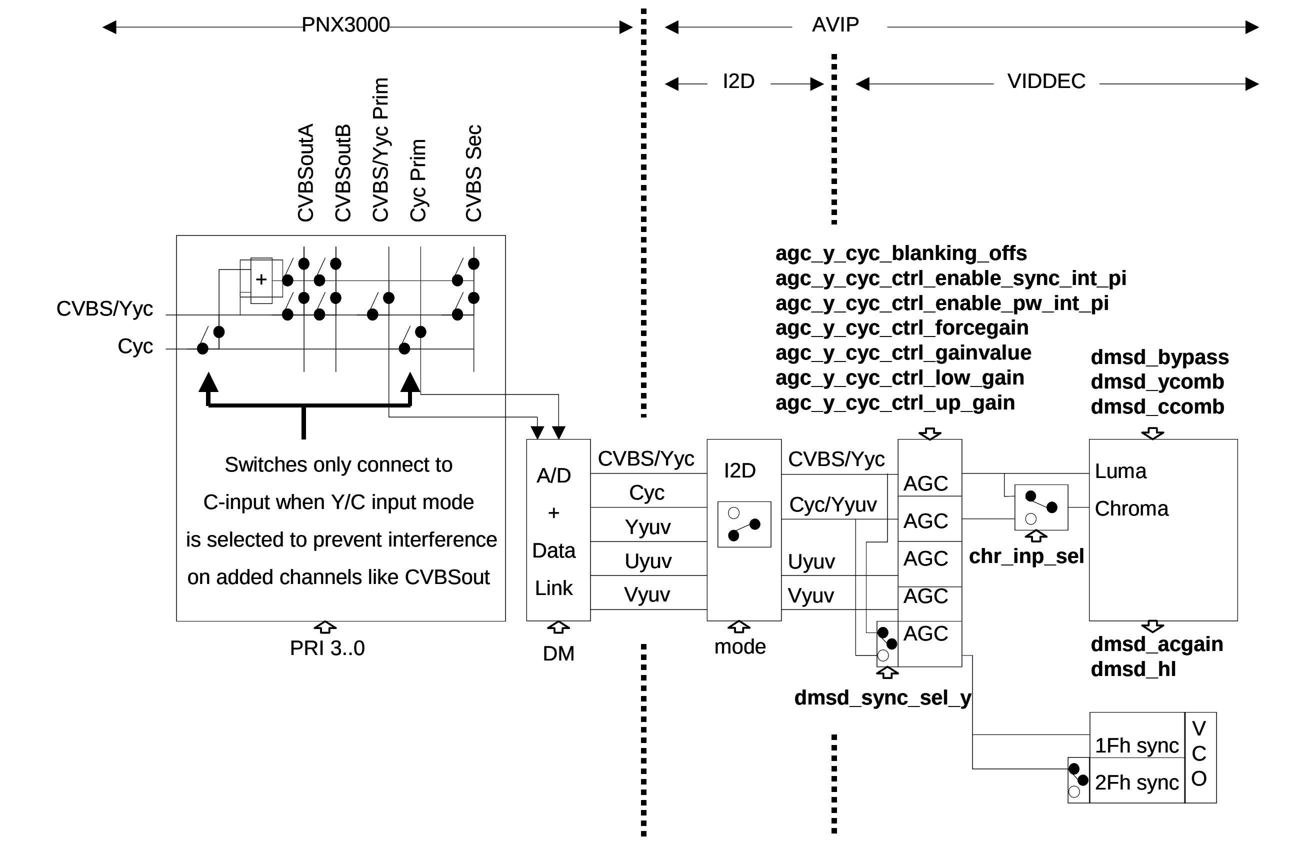


Fig 51.Block diagram CVBS / YC selection

The algorithm required to distinguish between these two inputs is quite simple.

The dmsd_acgain register reads the amplification of the color AGC. This measurement can be done for the CVBS/Y channel and for the C channel. The channel with the lowest value in dmsd_acgain has the largest chroma amplitude and will therefore be the channel carrying the color information.

The proposed algorithm works fine, but the detection time for a reliable automatic detection may run up to 2 seconds. The long detection time is related to the long time needed to ensure reliable color system recognition, especially when switching from a 50Hz source to a 60 Hz source or vice versa. The value of dmsd_acgain is not reliable unless the color system is recognized. If automatic detection of 2 seconds is considered too long, possible alternatives are:

- Use two menu items for the combined CVBS/YC connector, one configured for CVBS and one configured for YC. The user can decide which is the correct selection by checking whether the picture is black and white or color.
- Use a mechanical switch to indicate whether a cable is connected to the CVBS input or Y/C input (only possible when connectors are cinch for CVBS and 4-pin mini-din for Y/C). Software can readout the pin status via an I/O port and configure the correct settings.

Algorithm example:

- PNX3000: Select for the Primary Channel an input with Y/C capacity (YC3 or YC4) (PRI3..0 = 1 0 1 1 for YC3 or 1 1 0 0 for YC4). You have to select the input YC mode and not CVBS mode for the inputs, because only in the YC mode the C input is connected to the Primary C channel. The CVBS/Y channel of the Primary Channel will carry CVBS or Y, regardless CVBS mode or YC mode has been selected.
- PNX3000: Ensure that the correct Data Mode is chosen (1Fh) by setting DM = 0.
- I2D: Select in the I2D decoder mode 0a, I2D_mode = 0 0. In this mode, the I2D receiver selects the C channel for the (combined) C/Yyuv channel. Both CVBS/Y and C channel are now available at the VIDDEC inputs.
- VIDDEC: Select Y/C mode by setting the AGC settings for the C/Yyuv channel in Y/C mode (see chapter XXX) and setting chr_inp_del = 1. This connects the C/Yyuv channel to the color decoder input.
- VIDDEC: Wait first 50 msec. to avoid problems with latency. Check whether there is horizontal lock (dmsd_hl = 1) to guarantee proper timing and gating.
- VIDDEC Wait until a color system is found. If found, continue to next step, else wait maximum 1 second to proceed to the next step.
- VIDDEC: Read out dmsd_acgain and store the value. This reflects the chroma content on the C input.
- VIDDEC: Switch over to CVBS mode by connecting the CVBS/Y channel to the color decoder input, setting chr_inp_del = 0.
- VIDDEC Wait first 50 msec. to avoid latency problems. Wait until a color system is found. If found, continue to next step, else wait maximum 1 second to proceed to the next step. If the found color system is SECAM, select CVBS mode with Yyuv, U, V available. SECAM leads to maximum reading of dmsd_acgain and no SECAM Y/C exists.

- VIDDEC: Read out dmsd_acgain and compare the value with the stored value for the C input. The lowest value indicates the channel having the highest burst amplitude and will therefore carry the color information. If both read out maximum, the signal is Black and White (or SECAM) and CVBS mode with Yyuv u, V should be selected.
- I2D, PNX3000: If this is the CVBS/Y channel, it is CVBS. Switch back the I2D receiver to 1Fh default mode 0b (mode = 0 1) to have both CVBS and Yyuv, U and V available at the VIDDEC inputs and set the PNX3000 input selector back to CVBS input (PRI3..0 = 0 0 1 1 for CVBS3 or 0 1 0 0 for CVBS4) to prevent that interference on the C channel is added to the CVBS/Y channel for CVBSout A or B.
- VIDDEC: If this is the C/Yyuv channel, it is Y/C. Then it is needed to set chr_inp_del = 1 and set dmsd_byps = 1, dmsd_ycomb = 0 and dmsd_ccomb = 0 for a flat response.

General/specific use of the algorithm in software

For different blocks, the possible use of the algorithm is:

- PNX3000 Common

  Selection of 1 Fh mode for data link should be "standard".
- PNX3000 (Source select): Specific

  Of course, Y/C mode can only be selected when the input supports this mode. For PNX3000, these are only the inputs CVBS/Y3 + C3 and CVBS/Y4 + C4. The separate YC input for 3D combfilter is always in YC mode, but there are no objections to using the algorithm for this input.
- I2D receiver: Specific

  Each time the CVBS/YC source is changed, the algorithm can be applied, but in view of the long detection time it is best to limit the algorithm to only those sources that need it.
- VIDDEC: Specific

  Each time the CVBS/YC source is changed, the algorithm can be applied, but in view of the long detection time it is best to limit the algorithm to only those sources that need it.

CVBS + RGB insert via SCART

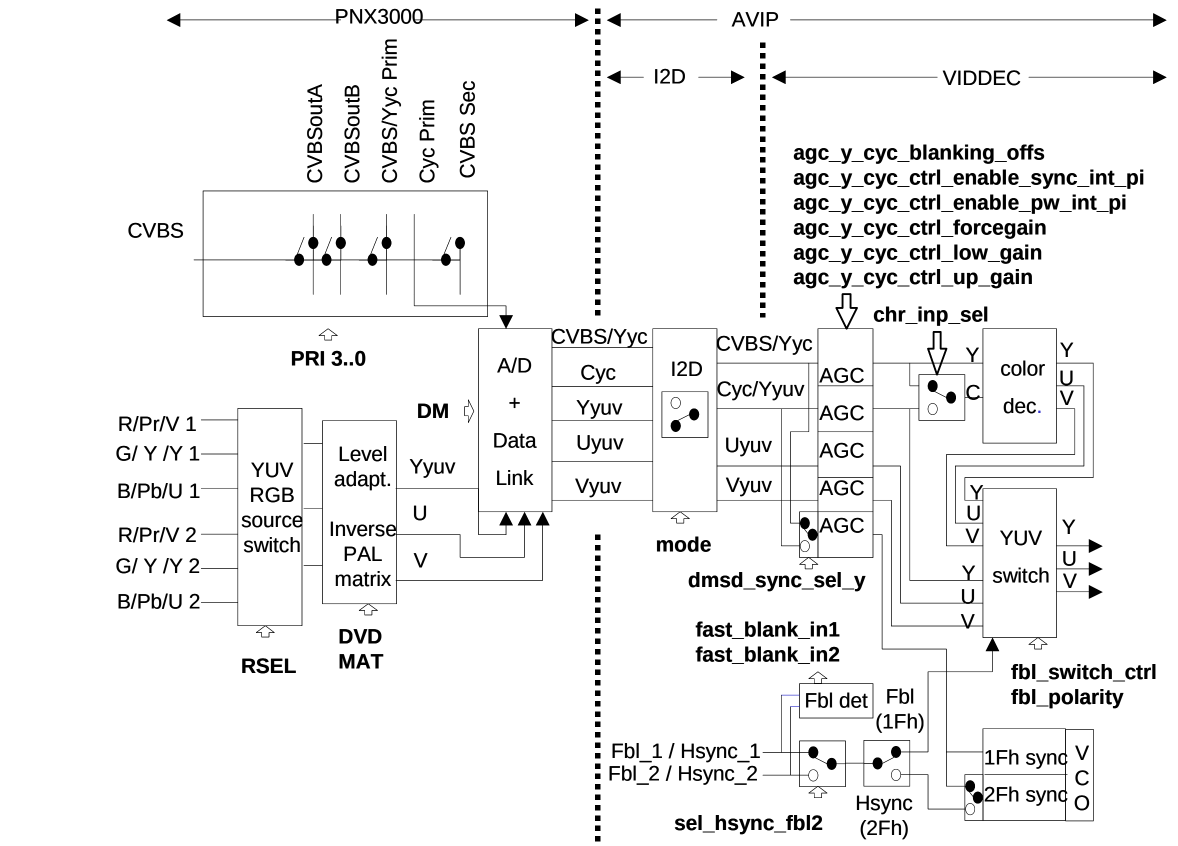


Fig 52.Block diagram full SCART input (CVBS + RGB + Fast Blanking)

This condition is meant for "full" SCART inputs according to the official "Peritel" norm. These SCART connectors provide a CVBS input and a RGB input. To switch automatically to the RGB signals, a Fast Blanking signal is provided on pin 16 of the SCART connector. When this signal is "high" (> 0.3 Volt) then RGB input should be selected. The Fast Blanking signal bandwidth is high enough to enable CVBS with an overlay of inserted OSD via the RGB input. Some de scramblers use this option, mainly Channel plus decoders in France.

Today, European DVD players provide an RGB output for better signal quality, but then the Fast Insertion is made continuously high. In both cases (OSD insertion or full picture insertion) the synchronisation is taken from the (accompanying) CVBS input. To handle such input, both CVBS and the YUV converted RGB signals should be routed to the VIDDEC. The Fast Blanking input, connected to pin 16 of the selected SCART should be enabled.

Remark: The CVBS and CVI inputs are located on the PNX3000, while the Fast Blanking Inputs are part of the VIDDEC in the AVIP.

Algorithm example:

- Check (via e.g. a properties list) that the selected input is a "full" SCART with RGB insertion possibility. Needed info:
- Connector has RGB insertion possibility
- To which FBL input is the Fast Blanking pin connected (1 or 2).
- PNX3000: Select with PRI_3..0 for the Primary Channel the correct CVBS input (for "Full" SCART Y/C is not possible)
- PNX3000: Select the CVI input where the RGB signals are connected (RSEL) Select as format RGB input (MAT, DVD = 1 0)
- PNX3000: Ensure that the correct Data Mode is chosen (1Fh) by setting DM = 0.
- I2D: Select mode 0b, I2D_mode = 0 1. In this mode Yyuv channel is selected for the combined C/Yyuv channel. Now CVBS, Yyuv, Uyuv and Vyuv are available at the input of the VIDDEC.
- VIDDEC: Set the AGC settings of the C/Yyuv channel for Yyuv signal in RGB mode. Because there is no sync on RGB, a combination of peak white limiting and AGC range limitation is used. The range is limited by setting the appropriate values for upper and lower gain of the AGC in the C/Yyuv channel.
- VIDDEC: Ensure that the sync from CVBS is selected for synchronisation setting dmsd_sync_sel_y = 0 (not needed if this is default mode)
- VIDDEC: Ensure that CVBS mode is selected for the color decoder by setting chr_inp_del = 0 (not needed if this is the 'default" mode)
- VIDDEC: Select the Fast Blanking input where the Fast Blanking signal of that SCART is connected (sel_hsync_fbl2)
- VIDDEC: Enable the Fast Blanking input by setting fbl_switch_ctrl = 0 0.
- VIDDEC: Optionally read back the status of fast_blank_in1 or fast_blank_in2 (depending on the selected Fast Blanking Input pin) to check whether full RGB insertion is applied (to know the displayed signal has RGB quality to optimize settings)

General/Specific use of the algorithm in software

For different blocks, the possible use of the algorithm is:

- PNX3000 (Source select): Specific set-up of CVBS input together with the RGB input is only needed when a full specified SCART connector is selected. So only apply when the connector property indicates full SCART.
- I2D: Common Selection of mode 0b should be the "standard" setting for the I2D receiver
- VIDDEC: Common Selection of sync on CVBS should be "standard" Selection of CVBS mode for the color decoder should be "standard"
- VIDDEC: Specific All items below are related to the selection of a "full" SCART input:

  AGC for Yyuv channel in RGB mode

  Selection of Fast Blanking input

  Enabling of Fast Blanking input

  Read-back of RGB insertion status

CVI Input Selection

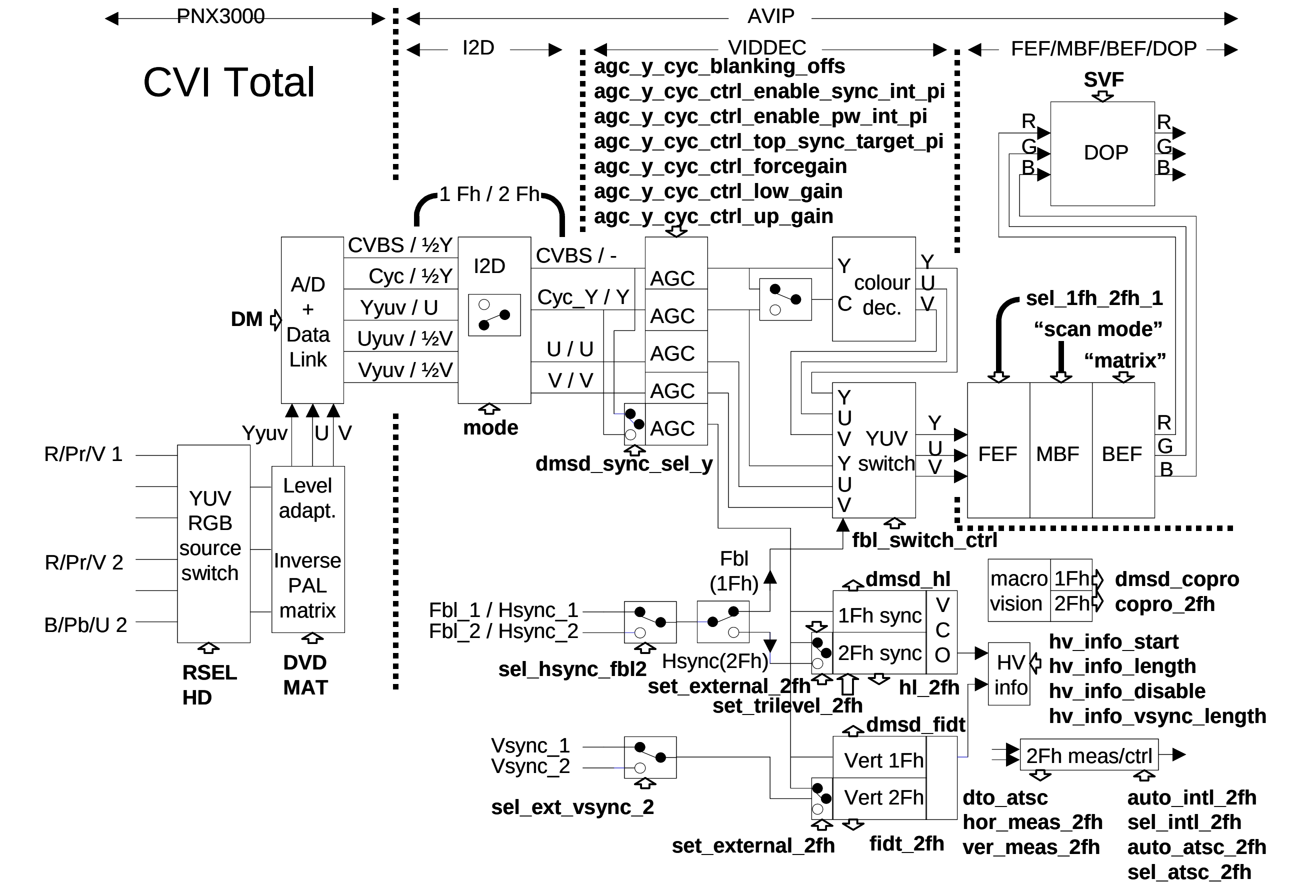


Fig 53.Block diagram CVI input selection

The detection between 1Fh and 2Fh signals is more complex than described in this part. The reason is that the 1Fh Hlock and 2Fh Hlock both can indicate incorrect lock. The 1 Fh lock, dmsd_hl, also can indicate false lock for 2Fh signals with VIDDEC in 1Fh mode and also for 1Fh signals with VIDDEC in 2Fh mode. The 2Fh lock can indicate false lock for 1Fh signals with VIDDEC in 2Fh mode.

The following CVI (Component Video Input) signals are supported by the system:

List 1: Supported formats

1 Fh:

- RGB with sync on CVBS and Fast Blanking (SCART norm, see previous chapter)
- RGB with Sync On Green (SOG)
- YPrPb with sync on Y

2 Fh:

- YPrPb 576p, progressive, Sync on Y, Fhor. = 31250 Hz (2 Fh), Fvert = 50 Hz (1 Fv)
- YPrPb 480p, progressive, Sync on Y, Fhor. = 31500 Hz (2 Fh), Fvert. = 60 Hz (1 Fv)

- RGB 576p, progressive, Sync On Green, Fhor. = 31250 Hz (2 Fh), Fvert = 50 Hz (1 Fv)
- RGB 480p, progressive, Sync On Green, Fhor. = 31500 Hz (2 Fh), Fvert. = 60 Hz (1 Fv)
- YPrPb 576p, progressive, Ext. H / V Sync, Fhor. = 31250 Hz (2 Fh), Fvert = 50 Hz (1 Fv)
- YPrPb 480p, progressive, Ext. H / V Sync, Fhor. = 31500 Hz (2 Fh), Fvert. = 60 Hz (1 Fv)
- RGB 576p, progressive, Ext. H / V sync, Fhor. = 31250 Hz (2 Fh), Fvert. = 50 Hz (1 Fv)
- RGB 480p, progressive, Ext. H / V sync, Fhor. = 31500 Hz (2 Fh), Fvert. = 60 Hz (1 Fv)
- ATSC 1080i, interlaced, Sync on Y, Fhor. = 33750 Hz ("2Fh"), Fvert. = 60 Hz (1 Fv)
- ATSC 1080i, interlaced, Ext. H / V sync, , Fhor. = 33750 Hz (2Fh), Fvert. = 60 Hz (1 Fv)
- 1080i, 50Hz, interlaced, Sync on Y, Fhor. = 28125 Hz ("2Fh"), Fvert. = 50 Hz (1 Fv)
- 1080i, 50Hz, interlaced, Ext. H / V sync, Fhor. = 28125 Hz ("2Fh"), Fvert. = 50 Hz (1 Fv)
- 1152i, 50Hz, interlaced, Sync on Y, Fhor. = 31250 Hz ("2Fh"), Fvert. = 50 Hz (1 Fv)
- 1152i, 50Hz, interlaced, Ext. H / V sync, Fhor. = 31250 Hz ("2Fh"), Fvert. = 50 Hz (1 Fv)

Not all these formats are used in practice by video equipment. Already excluded is the proprietary Philips YUV format, because it is not used by any external video equipment to establish an auto search algorithm for CVI inputs, formats should be limited to those used in practice. For auto detection, there can be no distinguishing between an RGB format and a YPrPb format, provided both have "internal" sync (on G for RGB or Y for YPrPb) or External sync.

So the following signals are the same for the system and cannot be uniquely identified:

List 2: Formats, which cannot be uniquely identified

1 Fh:

- RGB with Sync On Green (SOG)
- YPrPb with sync on Y

2 Fh:

- YPrPb 576p, progressive, Sync on Y, Fhor. = 31250 Hz (2 Fh), Fvert = 50 Hz (1 Fv)
- RGB 576p, progressive, Sync On Green, Fhor. = 31250 Hz (2 Fh), Fvert = 50 Hz (1 Fv)
- YPrPb 480p, progressive, Sync on Y, Fhor. = 31500 Hz (2 Fh), Fvert. = 60 Hz (1 Fv)
- RGB 480p, progressive, Sync On Green, Fhor. = 31500 Hz (2 Fh), Fvert. = 60 Hz (1 Fv)
- YPrPb 576p, progressive, Ext. H / V Sync, Fhor. = 31250 Hz (2 Fh), Fvert = 50 Hz (1 Fv)

- RGB 576p, progressive, Ext. H / V sync, Fhor. = 31250 Hz (2 Fh), Fvert. = 50 Hz (1 Fv)
- YPrPb 480p, progressive, Ext. H / V Sync, Fhor. = 31500 Hz (2 Fh), Fvert. = 60 Hz (1 Fv)
- RGB 480p, progressive, Ext. H / V sync, Fhor. = 31500 Hz (2 Fh), Fvert. = 60 Hz (1 Fv)

From each of the pairs mentioned, one should be selected for the search loop. Best is to exclude those signals, which are not commonly used in practice. Therefore it is best to omit the following formats:

- All YPrPb formats with External H and V sync.
- All RGB formats with Sync On Green

Further it is practical to leave out RGB with Sync on CVBS and Fast Blanking (SCART norm), this is a special case when a "full SCART" input is selected. The procedure for such input is described in Section “CVBS + RGB insert via SCART”. This leaves the list below for an automatic CVI search algorithm:

List 3: Proposed formats for an automatic search loop for CVI signals

1 Fh:

- YPrPb with sync on Y

2 Fh:

- YPrPb 576p, progressive, Sync on Y, Fhor. = 31250 Hz (2 Fh), Fvert = 50 Hz (1 Fv)
- YPrPb 480p, progressive, Sync on Y, Fhor. = 31500 Hz (2 Fh), Fvert. = 60 Hz (1 Fv)
- RGB 576p, progressive, Ext. H / V sync, Fhor. = 31250 Hz (2 Fh), Fvert. = 50 Hz (1 Fv)
- RGB 480p, progressive, Ext. H / V sync, Fhor. = 31500 Hz (2 Fh), Fvert. = 60 Hz (1 Fv)
- ATSC 1080i, interlaced, Sync on Y, Fhor. = 33750 Hz ("2Fh"), Fvert. = 60 Hz (1 Fv)
- ATSC 1080i, interlaced, Ext. H / V sync, , Fhor. = 33750 Hz (2Fh), Fvert. = 60 Hz (1 Fv)
- 1080i, 50Hz, interlaced, Sync on Y, Fhor. = 28125 Hz ("2Fh"), Fvert. = 50 Hz (1 Fv)
- 1080i, 50Hz, interlaced, Ext. H / V sync, Fhor. = 28125 Hz ("2Fh"), Fvert. = 50 Hz (1 Fv)
- 1152i, 50Hz, interlaced, Sync on Y, Fhor. = 31250 Hz ("2Fh"), Fvert. = 50 Hz (1 Fv)
- 1152i, 50Hz, interlaced, Ext. H / V sync, Fhor. = 31250 Hz ("2Fh"), Fvert. = 50 Hz (1 Fv)

Now each signal has at least one unique parameter, which enables to identify the correct format.

The formats, not included in the automatic search loop can still be manually selected if required. It is possible to make another automatic search list, provided form each of the pairs mentioned in List 2 only one is chosen.

The distinguishing parameters for detection are:

- 1 Fh or 2 Fh

and further in case it is 2 Fh:

- Internal sync (sync on Y) or External sync
- 50 Hz or 60 Hz
- ATSC or non-ATSC

Algorithm example:

An example for an auto detection algorithm is given below. The search order in this algorithm is:

1 Fh:

- YPrPb with sync on Y

2 Fh:

- ATSC 1080i, interlaced, Sync on Y, Fhor. = 33750 Hz ("2Fh"), Fvert. = 60 Hz (1 Fv)
- YPrPb 576p, progressive, Sync on Y, Fhor. = 31250 Hz (2 Fh), Fvert = 50 Hz (1 Fv)
- YPrPb 480p, progressive, Sync on Y, Fhor. = 31500 Hz (2 Fh), Fvert. = 60 Hz (1 Fv)
- ATSC 1080i, interlaced, Ext. H / V sync, , Fhor. = 33750 Hz (2Fh), Fvert. = 60 Hz (1 Fv)
- RGB 576p, progressive, Ext. H / V sync, Fhor. = 31250 Hz (2 Fh), Fvert. = 50 Hz (1 Fv)
- RGB 480p, progressive, Ext. H / V sync, Fhor. = 31500 Hz (2 Fh), Fvert. = 60 Hz (1 Fv)

General

- Check (e.g. via a property list) what formats the selected CVI input supports. Non-supported input formats do not have to be checked. Further, in case of support of External Sync input, check which H and V inputs are used.
- PNX3000 Select using RSEL the appropriate CVI input on the PNX3000
- VIDDEC Select the external YUV path by setting fbl_switch_ctrl = 1 0
- It is assumed that VIDDEC is in 1Fh mode. Because 2Fh is only possible on the CVI inputs and external 2Fh signals are not so common, 1 Fh should be the 'default" setting for VIDDEC. This implies that the VIDDEC should always be in 1 Fh mode before switching to another input.

1 Fh detection

- PNX3000 Set MAT,DVD to 0 1 to select YPrPb format
- PNX3000 Ensure that the correct Data Format is chosen (1 Fh) by setting DM = 0
- I2D: Select mode 0b, I2D_mode = 0 1. In this mode Yyuv channel is selected for the combined C/Yyuv channel. Now Yyuv, Uyuv and Vyuv are available at the input of the VIDDEC
- VIDDEC : Set the AGC settings of the C/Yyuv channel for standard Yyuv signal in YprPb mode without Macrovision.

- VIDDEC Select to take the sync from the Yyuv channel by setting dmsd_sync_sel_y = 1.
- VIDDEC: To be updated. More bits needed to detect unique 1Fh!! Check for horizontal lock by checking for dmsd_hl = 1 for 40 msec. If horizontal lock is found, VIDDEC Check for 50 or 60 Hz, reading dmsd_fidt.
- If 0, signal is 50 Hz
- VIDDEC Set dmsd_vsta and dmsd_vsto (see Table 42 Fh YPrPb 50 Hz

  If 1, signal is 60 Hz.
- VIDDEC Set dmsd_vsta and dmsd_vsto (see Table 42 1Fh YPrPb 60 Hz

VIDDEC Check for Macrovision reading dmsd_copro. If dmsd_copro = 1, (Macrovision detected), adapt the value of the sync AGC register agc_y_cyc_ctrl_top_sync_target_pi to 118h (instead of 100h for normal operation)

Exit the auto detection algorithm. If no horizontal lock is found, continue.

2 Fh detection

Switch to 2Fh mode

- PNX3000 Select 2 Fh YUV throughput by setting DM = 1
- I2D Select 2 Fh YUV throughput by selecting mode 2 setting mode = 1 0
- VIDDEC Switch VIDDEC0 to 2 Fh mode. See separate procedure.
- PNX3000 Set MAT,DVD to 0 1 to select YPrPb format
- VIDDEC : Set the AGC settings of the C/Yyuv channel for Yyuv signal in 2Fh YprPb 50/60 Hz mode.
- VIDDEC Select to take the sync from the Yyuv channel by setting dmsd_sync_sel_y = 1
- VIDDEC Ensure that automatic adaptation for interlace is selected (auto_intl_2fh = 1) and automatic adaptation of the H and V counter for ATSC is enabled (auto_atsc_2fh = 1). Both values are set correct after a reset.
- VIDDEC To be updated. More bits needed to detect unique 2Fh!! Check for horizontal lock by checking for hl_2fh = 1 for 40 msec.

When horizontal lock is found:

- VIDDEC Check for ATSC by reading dto_atsc or meas_atsc_2fh.= 1 If 1, the source is identified as ATSC 1080i input with tri-level sync on Y.
- PNX3000 Select ATSC clamping mode by setting HD = 1
- VIDDEC Select tri-level sync by setting set_3l_2fh = 1
- VIDDEC Set interlaced mode by setting foet_2fh = 1
- VIDDEC Set hv-info_start/length/disable/vsync_length and vsto_2fh / vsta_2fh, hrefs_2fh / hrefs_2fh according table XX for ATSC format

Exit the auto detection algorithm. If 0, source is not ATSC, continue.

- Check that intl_2fh or meas_intl_2fh = 0 to indicate a progressive input. If progressive input, signal is identified as YPrPb 576p or 480p with sync on Y.

- VIDDEC Set progressive mode by setting foet_2fh = 1
- VIDDEC Check for the vertical frequency by reading fidt_2fh If fidt_2fh = 0, signal is YPrPb 576p, sync on Y, 50 Hz.
- VIDDEC Set hv_info_start/length/disable/vsync_length and vsto_2fh / vsta_2fh, hrefs_2fh / hrefs_2fh according table XX 2Fh YPrPb 576p.
- VIDDEC Set hv_info_start/length/disable/vsync_length and vsto_2fh / vsta_2fh, hrefs_2fh / hrefs_2fh according table XX 2Fh YPrPb 480p
- VIDDEC Check for Macrovision reading copro_2fh. If copro_2fh = 1, (Macrovision detected), adapt the value of the sync AGC register agc_y_cyc_ctrl_top_sync_target_pi to 118 (in stead of 100 for normal operation)

Exit the auto detection algorithm. If interlaced but non-ATSC, signal format is not supported. Exit and display an error message. When no lock is found, continue.

- VIDDEC Select external sync input. Set sel_hsync_fbl2 and sel_ext_vsync2 to select the correct external H and V input Select external sync by setting sel_ext_2fh = 1
- VIDDEC Check for horizontal lock by checking for hl_2fh = 1 for 40 msec.
- When horizontal lock is found:
- VIDDEC Check for ATSC by reading dto_atsc or meas_atsc_2fh.= 1 If 1, the source is identified as ATSC 1080i input with external H and V sync..
- PNX3000 Select ATSC clamping mode by setting HD = 1
- VIDDEC Select forced odd/even toggle (interlace) by setting foet_2fh = 1
- VIDDEC Set hv-info_start/length/disable/vsync_length and vsto_2fh / vsta_2fh, hrefs_2fh / herefs_2fh according table XX for ATSC format

Exit auto detection algorithm. If 0, source is not ATSC, continue.

- Check that intl_2fh or meas_intl_2fh = 0 to indicate a progressive input. If progressive input, signal is identified as RGB 576p or 480p with external H and V sync.
- PNX3000 Select RGB input by setting MAT,DVD = 1 0
- VIDDEC : Set the AGC settings of the C/Yyuv channel for Yyuv signal in RGB mode according table XX 2Fh RGB 576p/480p
- VIDDEC Select progressive mode by setting foet_2fh = 1
- VIDDEC Check for the vertical frequency by reading fidt_2fh If fidt_2fh = 0, signal is RGB 576p, external H and V sync, 50 Hz
- VIDDEC Set hv_info_start/length/disable/vsync_length and vsto_2fh / vsta_2fh, hrefs_2fh / herefs_2fh according table XX 2Fh RGB 576p
- VIDDEC Set hv_info_start/length/disable/vsync_length and vsto_2fh / vsta_2fh, hrefs_2fh / herefs_2fh according table XX 2Fh RGB 480p
- Exit the auto detection algorithm. If interlaced, signal format is not supported. Exit and display an error message. When no lock is found, continue.
- No 2Fh source found. Switch back to 1Fh mode and restore default settings.
- PNX3000 Select 1 Fh YUV throughput by setting DM = 0
- I2D Select 1 Fh YUV throughput by selecting mode 0b setting mode = 0 1
- VIDDEC Switch VIDDEC to 1 Fh mode. See separate procedure.

The total procedure can be repeated 10 times from 1 Fh detection till here. If not successful, it is best to stop in 1 Fh mode (YPrPb, sync on Y) and display a message. When switching to another input, the VIDDEC should be set in 1 Fh mode and the settings of the affected bits be restored.

To speed up the detection once a signal format is found, the found format can be stored. When the CVI input is again selected, the loop might start at the format, found the previous time. The check on Macrovision (dmsd_copro and copro_2fh) should be carried out at regular intervals because the customer can change the disk/tape without changing the input.

3.6.5 VIDDEC registers

See Ref. 8 for detailed register descriptions.

3.7 Data Capture Unit (DCU)

3.7.1 Functional Overview

The purpose of the Data Capture Unit (DCU) is to acquire digital data (containing Teletext, Closed Captions, etc.) from a CVBS video input source or the Y component, perform processing on the received data and stream it out to the ITU formatter unit.

The CVBS signal or Y component from the DLINK receiver are supplied to the DCU in digitised form, and are processed by the data slicer section (MULVIP) to extract the serial data and its associated clock. The serial data is assembled into bytes by the SERPAR section, and subsequently the bytes into packets which are then fed out to the ITU 656 formatter. Received packets of certain data types can be optionally processed to perform error checking and decoding functions.

Control of the DCU is via programmable registers, which are accessible through a PI Slave port. This also produces two configurable interrupts, one at field rate and one to signal that a data packet has been received.

Remark: 2fH video signals are not supported by the DCU

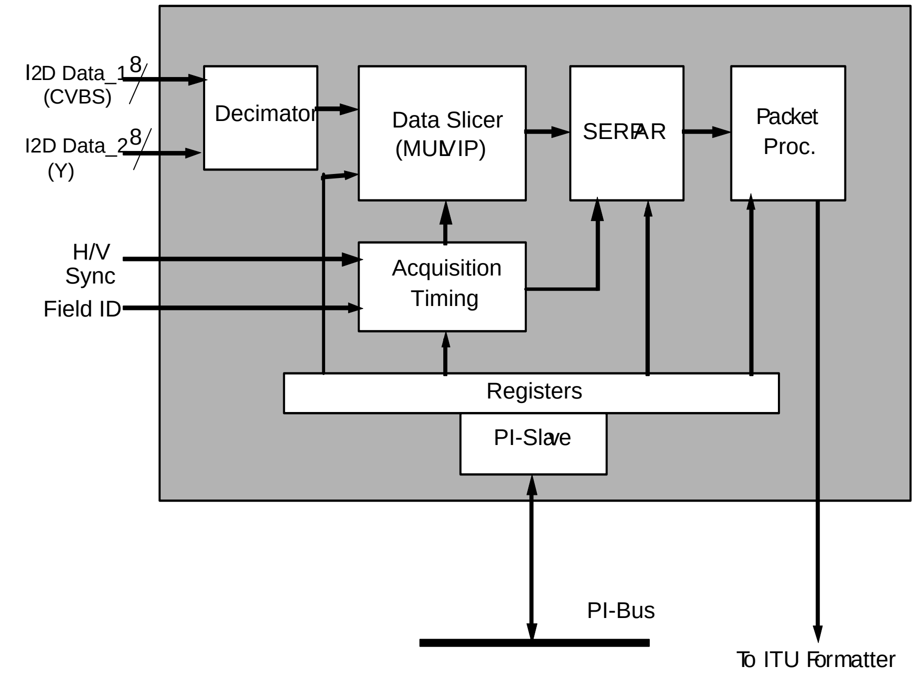


Fig 54.Block diagram DCU

The blocks are described as:

- Decimator: Reduces sample rate (from 27MHz to 13.5MHz) of incoming Digitised CVBS or Y data stream from DLINK Receiver
- Data Slicer: Reconstructs the transmitted bit stream and associated clock from the video input
- Serpar: Serial to Parallel converter - converts serial data into bytes
- Acquisition Timing: Locks onto sync signals, and provides timing information to other sections of data capture circuitry
- Packet Formatter: Performs data decoding and some error correction; assembles received bytes into a packet structure and streams out data to ITU 656 Formatter
- PI-Slave: Interfaces control and status registers to the PI Bus

3.7.2 Design Specification

The fundamental principle of the DCU is that of specifying a data type to be captured for each line and field of the incoming video signal. The data types supported are listed in Table 73.

VBI mode text can be transmitted multiplexed on a 525 or 625 line composite video transmission; the video lines available for text and related services are specified in the table below. Full field text uses all available TV lines to transmit teletext data, giving a maximum of 305 packets per field.

Table 55: VBI Mode Text - Line Numbers

Data Type	525 Line	625 line
Teletext	10 to 21,273 to 284	6 to 22, 318 to 335
VPS	Not Available	16
Gemstar™	10 to 25, 273 to 288	Not available
WSS	20 and 283	23
Line 21 Data (CC)	21 and 284	22 and 335

3.7.3 Data Packet Formats

This section describes the contents of the data packets of the various data types supported by the DCU.

Table 56: Data Packet Structure

Data Type	Data Packet Byte Number								
	0 - 1	2 - 3	4	5 - 10	11-15	16-27	28 - 35	36	37 - 43
Euro WST	S [1]	Mag/Pkt [2] (2)	Teletext Data (40)						
US WST	S	Mag/Pkt (2)	Teletext Data (32)						
NABTS	S	Teletext Data (34)							
WSS625	S	WSS625 Data (14)							
WSS525	S	WSS525 Data (3)							
VPS	S	VPS Data (26)							
CC	S	CC Data (2)							
Moji	S	Prefix (14 bits)	Information Data (22 bytes) + Parity Check (82 bits)						
VITC	S	VITC Data (9)							
Gemstar(2x)	S	Gemstar data (4)							

[1] Status Bytes

[2] Magazine and packet numbers, as WST spec.

The structure of the Open data type packets follows the same pattern; all packets start with 2 status bytes, followed by the appropriate number of captured data bytes. See Table 73 for a complete list of supported data types.

3.7.3.1 Status Bytes

The first two bytes of each packet form the Status Bytes, which contain information about the data in the rest of that packet.

Table 57: Status Bytes

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	OP	field_id	LN8	LN7	LN6	LN5	LN4	LN3
Byte 0	OP	LN2	LN1	LN0	DT3	DT2	DT1	DT0

Table 58: Status Bytes Bit Definitions

Bits	Bit Description
OP	Odd Parity of bits 6 to 0.
field_id	field_id = 0 -> field1; field_id = 1 -> field2.
LN8..0	Line Number in current field (1..313).
DT3..0	Data Type according to Table 73.

3.7.3.2 Euro WST, US WST and NABTS Data

Stored in transmission order; first received bit becomes LSB of byte in packet.

3.7.3.3 WSS625 Data

Table 59: Assembly of WSS625 Data into Data Packet

WSS Bit No.	Bit 0	Bit 1	Bit 2	Bit 3	. . .	Bit 11	Bit 12	Bit 13
Bit No. in Packet	7 0	7 0	7 0	7 0	. . .	7 0	7 0	7 0
Data in Packet	xxdddddd	xxdddddd	xxdddddd	xxdddddd	. . .	xxdddddd	xxdddddd	xxdddddd
Byte No.in Packet	2	3	4	5	. . .	13	14	15

In Table 59 xx is undefined by the WSS625 transmission and is set to zero by the hardware. Each dddddd are a group of 6 bits representing a single symbol (a WSS625 bit) bi-phase coded and then oversampled at 3 times the baud rate.

To decode the individual bits, it is usual to take a majority decision on each group of 3 bits (majority of 0s or 1s), then compare the first and second three-bit groups to do bi-phase decoding. This is illustrated in the table below.

This decoding is not actually performed by the Data Capture hardware.

Table 60: WSS625 Biphase Decoding

Stored bits b5..b0	1st bit	2nd bit	Biphase Decoded	Biphase Error
111 000	1	0	1	No
000 111	0	1	0	No
101 010	1	0	1	No
101 011	1	1		Yes
000 011	0	1	0	No
010 000	0	0		Yes

3.7.3.4 WSS525 Data

The received data contains 20 bits including 6 bits of CRC code; all 20 bits are packed into 3 bytes and written to the packet with the first received bit becoming LSB of byte 4. CRC checking is performed on the received data, and a flag to indicate the result is stored in the last data byte - see Table 61.

Table 61: WSS525 Data in Data Capture Memory

WSS525 word	**2**		**1**				**0**		**CRCC**		**2**						**-**				**CRCC**			
WSS525 Bit No.	8	7	6	5	4	3	2	1	16	15	14	13	12	11	10	9	C[1]	-[2]	-	-	20	19	18	17
Bit No. in Byte	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Byte No. in Packet	2								3								4							

[1] Result of CRC check - ‘0’ if no errors, ‘1’ otherwise

[2] Unused bits written as ‘0’ by hardware

3.7.3.5 VPS Data

Table 62: Assembly of VPS Data into Data Packet

VPS word	**Word 3**								**Word 4**								**…**	**Word 14**								**Word 15**							
Bit Number	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	…	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4
Data in Packet	ddccbbaa				ddccbbaa				ddccbbaa				ddccbbaa				. . .	ddccbbaa				ddccbbaa				ddccbbaa				ddccbbaa			
Bit Number in Byte	7			0	7			0	7			0	7			0	. . .	7			0	7			0	7			0	7			0
Byte No. in Packet	2				3				4				5				. . .	24				25				26				27			

Each pair of bits dd, cc, bb or aa is a single symbol, biphase coded. 01 represents a 1 symbol, 10 represents a 0 symbol. 00 and 11 are biphase errors.

The data can be decoded in minimum processor time by using a look-up table (256 bytes) using the received data as index, which gives the correct decoded biphase data in the ls 4 bits of each byte and 4 corresponding error flags in the ms bits; e.g.: a stored byte with hex value 0x1B (binary 00.01.10.11) would be decoded as 1001.0100 (i.e.: the middle two pairs 01 and 10 decode correctly to 1 and 0, but the outer two pairs 00 and 11 are errors).

Table 63: VPS Biphase Decoding

Stored bit pair msb lsb	**Biphase Decoded**	**Biphase Error**
0 0	-	Yes
0 1	1	No
1 0	0	No
1 1	-	Yes

3.7.3.6 Closed Caption

Stored in transmission order - first received bit becomes LSB of byte 2

3.7.3.7 Gemstar™ (2x)

Stored in transmission order - LSB of each byte is transmitted first.

A special feature of the Gemstar data type is that the DCU is able to detect and automatically switch between Gemstar(2x) data and Line-21 data (e.g. Closed Captions). This detection is performed for each line on which the Gemstar data type is requested, by analyzing the received framing code sequence and comparing it with two fixed patterns. If it matches the Gemstar(2x) pattern, the line is sampled at the Gemstar(2x) data rate, resulting in 4 bytes of sliced data. If it matches the Line-21 pattern, the remainder of the line is sampled at the CC525 data rate (which is exactly half of the Gemstar(2x) data rate), resulting in 2 bytes of sliced data; the packet structure in this case is identical to the normal Line-21 (Closed Caption) packet structure. To indicate to the application software that a Line-21 packet has been captured, the DT3..0 bit field in the packet Status Bytes is modified from a value of 6 (0110) to a value of 5 (0101), this being the Data Type code for US Closed Caption (Line-21) Data as listed in Table 73.

This automatic detection feature was provided to support the (now obsolete) “Gemstar(1x)” data format. It may still have a useful application, however, since it enables the DCU to search for Gemstar(2x) data on all VBI lines, as required by the Gemstar specification [17], whilst still acquiring any Line-21 data that is being transmitted at the time. This is important to enable continuous monitoring of V-Chip data as well as providing an uninterrupted Closed Caption service.

3.7.3.8 Moji (Japanese Text)

The Moji Data Line contains 272 bits, made up of a 14-bit prefix, 22 Information Data bytes (176 bits) and an 82-bit Parity Check. The captured bits are constructed into bytes, LSB first, in transmission order, but the first byte of the packet (after the Status Bytes) contains only 6 bits of transmitted data in bits 2-7. Bit 2 corresponds to the first transmitted bit; bits 1 and 0 are filled with zeros.

This is done in order to align the Information Data bytes to byte boundaries in the constructed data packet. It also means that the last data byte in the packet contains only 2 transmitted bits (in positions 0 - 1) - the remaining 6 bits are undefined and should be ignored.

3.7.3.9 VITC Data

The VITC data line in both 625- and 525-line video formats contains 90 bits, which can be divided into nine 10-bit groups. The first two bits of each group are defined as Synchronizing Bits, and consist of a fixed ‘1’ followed by a fixed ‘0’. These Synchronizing Bits are excluded from the data packet constructed by the Data Capture unit, leaving exactly nine 8-bit bytes of useful data. The bytes are presented in transmission order, with the LSB of each corresponding to the first transmitted bit - the structure is shown in the table below.

Table 64: VITC Data Packet Contents

VITC Bit No.	0	1	2 - - 9	10	11	12 - -19	- - -	80	81	82 - -89
Byte No. in Packet			2			3	- - -			10
Bit No. in Packet			0 - - - 7			0 - - - 7	- - -			0 - - - 7

Remark: This behavior is believed to be different from a previous version of the Data Slicer that supported the VITC data types.

3.7.3.10 Open Data Types

The Open data types are provided primarily to allow capture of low bit-rate data types that are not specifically supported by the Data Capture Unit, by oversampling the transmitted data and leaving software to extract the individual bytes.

Acquisition starts when a match is found for the programmable framing code (taking into account the FCE control bit); bytes are then captured, LSB first, in transmission order at the specified bit- rate. The search window for the framing code is open between approximately 8 and 16s into the line, referenced to the falling edge of the H-Sync pulse.

The number of bytes captured in the open data types depends on when in this period the framing code match is found: the numbers shown in Table 73 are the maximum assuming earliest possible detection of the framing code.

If the framing code is detected any later, fewer bytes may be captured; also, data capture may extend into the line-reset region of the following line, in which case the last few data bytes in the packet are meaningless. It is left up to the software to process the appropriate amount of data from the packet, as defined by the application for which the open data type is being used.

3.7.4 Packet Processing Capabilities

A number of hardware-supported packet processing options are available. These are primarily intended to assist with WST data capture and processing. Hardware data processing can be enabled or disabled by software.

Acquired packets may optionally be decoded. The data processing operations available are: 8/4 hamming decoding of the magazine and packet number (bytes 2 and 3 of every WST teletext packet); and page header decoding for bytes 4 to 11 of WST packets X/0.

3.7.4.1 Magazine and Packet Number Decoding

When magazine and packet number decoding is enabled, bytes 2 and 3 of WST teletext packets are decoded by hardware during the acquisition process. The decoded result is written back in the place of the original data.

Magazine and packet number decoding is only possible for data types EuroWST (data type 0), USWST (data type 4), and Teletext (data type 8).

Input Data Format

Table 65: Input Data Format

	d7							d0
Byte 2 [1]	Packet Number Bit 1	Protection	Magazine Number Bit 3 [2]	Protection	Magazine Number Bit 2	Protection	Magazine Number Bit 1	Protection
Byte 3	Packet Number Bit 5	Protection	Packet Number Bit 4	Protection	Packet Number Bit 3	Protection	Packet Number Bit 2	Protection

[1] Transmitted byte layout.These two bytes are 8/4 Hamming encoded.

[2] Magazine number bit 3 is used as the ‘Tabulation’ bit for USWST.

Output Data Format

Table 66: Output Data Format

	d7							d0
Byte 2 [1]	Error Flag 1[2]	0 [3]	0	0	0	Magazine Number Bit 3 [4]	Magazine Number Bit 2	Magazine Number Bit 1
Byte 3	Error Flag 2 [5]	0	0	Packet Number Bit 5	Packet Number Bit 4	Packet Number Bit 3	Packet Number Bit 2	Packet Number Bit 1 [6]

[1] Byte numbers are according to the WST packet byte numbers.

[2] When set high, Error Flag 1 indicates non-correctable errors in incoming byte 2 of Teletext packet.

[3] Other bits set to zero.

[4] Magazine number bit 3 is used as the ‘Tabulation’ bit for USWST.

[5] When set to high, Error Flag 2 indicates non-correctable errors in incoming byte 3 of Teletext packet.

[6] During decoding, packet number bit1 is read from byte 2, written back in byte 3.

3.7.4.2 Page Header Decoding

Page header decoding provides automatic processing for the 8/4 hamming encoded page control bytes of WST teletext packets X/0.

Page header decoding is only possible for data types EuroWST (data type 0), USWST (data type 4), and Teletext (data type 8).

Enable magazine and packet number decoding for page headers to be identified and decoded.

Table 67: Page Header Byte Sequence

Byte Number	4	5	6	7	8	9	10	11
Byte Function	Page Number Units	Page Number Tens	Subcode S1	Subcode S2	Subcode S3	Subcode S4	Control Bits C7-C10	Control Bits C11-C14

Bytes 4-11 of the WST packet X/0 are all 8/4 hamming. When page header decoding is enabled, the decoded bytes are written back in the place of the encoded bytes, overwriting the original data.

Table 68: Hamming 8/4 Data Format

	d7							d0
Encoded Byte	D4	P4	D3	P3	D2	P2	D1	P1
Decoded Byte	Error Flag	0	0	0	D4	D3	D2	D1

3.7.4.3 WSS525 CRC Checking

CRC checking is performed on every WSS525 packet received, and the result of the CRC check is written into the packet along with the captured data as shown in Table 61. Software can examine the CRC check bit for data integrity and reject packets that fail.

3.7.4.4 Packet Validity Checking

Error checking can be performed on packets of certain data types to determine the validity of the packets. Although all captured data packets are sent to the output regardless of any errors, certain features of the DCU are affected by the result of these checks. These include:

- Generation of the pktrx (packet-received) interrupt
- WSSV bit in the Status Bytes/DCS register
- Data amplitude tracking/searching

In the case of text packets, the magazine and packet number (bytes 2 and 3 of WST teletext) may be 8:4 hamming checked as a validation of data integrity. Hamming checking is always performed on WST packets (data types 0 and 4), and is optional for other

Teletext packets (data types 8 and C) depending on the HAM bit in register DCR1. In the case of WSS525 packets (date type 7), the result of the CRC check is used to determine packet validity only if the WSS_CRC bit is set in register DCR2; otherwise all captured WSS525 packets are considered valid, relying on the Start Bit detection alone.

Teletext Hamming checking for data types 8 and C should be disabled if these data types are being used to capture data that does not employ Hamming encoding.

3.7.5 Registers

Table 69: Register summary

Address	Name	Description
0x0	DCR1	Data capture control register 1
0x4	LCR2_5	Line Control Registers 2 to 5
0x8	LCR6_9	Line Control Registers 6 to 9
0xC	LCR10_13	Line Control Registers 10 to 13
0x10	LCR14_17	Line Control Registers 14 to 17
0x14	LCR18_21	Line Control Registers 18 to 21
0x18	LCR22_24	Line Control Registers 22 to 24
0x1C	DCS	Data Capture Status
0x2C	DCR2	Data Capture Register 2
0xFCC	Debug_Ctrl	Debug Control Register
0xFE0	INT_STATUS	Interrupt Status
0xFE4	INT_ENABLE	Interrupt Enable
0xFE8	INT_CLEAR	Interrupt Clear
0xFEC	INT_SET	Interrupt Set
0xFFC	MOD_ID	Module ID

3.7.5.1 DCR1: Data Capture Register (Write)

Table 70: DCR1 Registers

Bit	Symbol	Access	Reset value	Description
Offset 0x0 - DCR1				
31	INV_MSB	R/W	0x0	Invert MSB of incoming CVBS data (used for signed data). '0' = no invert; '1' = invert MSB
30:22	V[8:0]	R/W	0x000	Vertical position of CVFLD interrupt, start of line number 1-312 (must be non-zero for interrupt to be issued)
21:16	H[5:0]	R/W	0x00	Horizontal position of packet received interrupt, in microseconds (32-63, 0-3)
15:8	FC[7:0]	R/W	0x00	Framing code for 'open' data types
7	DPH	R/W	0x0	'0' = do not decode page header; '1' = hamming 8/4 decode page header (bytes 6 to 13 of packets X/0). This also requires DMP to be set.
6	DMP	R/W	0x0	'0' = do not decode magazine and packet; '1' = decode magazine and packet (hamming 8/4)

Table 70: DCR1 Registers *...continued*

Bit	Symbol	Access	Reset value	Description
5	HAM_DISABLE	R/W	0x0	'0' = hamming check magazine and packet address of open data types; '1' = do not hamming check. Hamming checking is only optional for data types 8 to F
4	FCE	R/W	0x0	'0' = one error allowed in framing code; '1' = no errors allowed in framing code
3	VID_525	R/W	0x0	'0' = expect 625 line transmissions; '1' = expect 525 line transmissions
2	HUNT_DISABLE	R/W	0x0	'0' = amplitude searching allowed; '1' = amplitude searching disabled. Amplitude searching should not be disabled in normal operation.
1	VCR	R/W	0x0	'0' = normal PLL mode (no integral path); '1' = VCR PLL mode (integral path enabled)
0	ACQ_EN	R/W	0x0	'0' = All lines treated as 'do-not-acquire'; '1' = normal acquisition mode

3.7.5.2 DCR2: Data Capture Register 2 (Write)

Table 71: DCR2 Registers

Bit	Symbol	Access	Reset value	Description
Offset 0x2C - DCR2				
31:24	RSD[31:24]	R	0x00	Reserved
23	WSS_CRC	R/W	0x0	WSS525 CRC control. '0' = WSS525 CRC check is ignored for packet validity checking ; '1' = WSS525 packet validity depends on result of CRC check.
22	INPUT_SEL	R/W	0x0	Video input selection 0: Selects first video (I2D) input stream 1: Selects second video (I2D) input stream
21:18	RSD[21:18]	R/W	0x0	Reserved
17	FPOS	R/W	0x0	Field Input Polarity Control 0: Field input polarity 0 = odd, 1 = even 1: Field input polarity 0 = even, 1 = odd
16:8	VSPOS[8:0]	R/W	0x000	Timing of vertical sync input relative to CVBS, in lines
7:0	HSPOS[7:0]	R/W	0x00	Timing of horizontal sync input relative to CVBS, in increments of 0.25us (1/256 of line period)

Aligning the internal line counter to PAL or NTSC line counting standard, can be done by appropriately setting the VSPOS field in the DCR2 register.

3.7.5.3 LCR2..LCR24: Line Control Registers (Write)

Table 72: Structure of LCR Registers

	31..28	27..24	23..20	19..16	15..12	11..8	7..4	3..0
LCR2_5	Line 5 Field 1 DT[3:0]	Line 5 Field 2 DT[3:0]	Line 4 Field 1 DT[3:0]	Line 4 Field 2 DT[3:0]	Line 3 Field 1 DT[3:0]	Line 3 Field 2 DT[3:0]	Line 2 Field 1 DT[3:0]	Line 2 Field 2 DT[3:0]
LCR6_9	Line 9 Field 1 DT[3:0]	Line 9 Field 2 DT[3:0]	Line 8 Field 1 DT[3:0]	Line 8 Field 2 DT[3:0]	Line 7 Field 1 DT[3:0]	Line 7 Field 2 DT[3:0]	Line 6 Field 1 DT[3:0]	Line 6 Field 2 DT[3:0]
LCR10_13	Line 13 Field 1 DT[3:0]	Line 13 Field 2 DT[3:0]	Line 12 Field 1DT[3:0]	Line 12 Field 2 DT[3:0]	Line 11 Field 1 DT[3:0]	Line 11 Field 2 DT[3:0]	Line 10 Field 1 DT[3:0]	Line 10 Field 2 DT[3:0]
LCR14_17	Line 17 Field 1 DT[3:0]	Line 17 Field 2 DT[3:0]	Line 16 Field 1 DT[3:0]	Line 16 Field 2DT[3:0]	Line 15 Field 1 DT[3:0]	Line 15 Field 2 DT[3:0]	Line 14 Field 1 DT[3:0]	Line 14 Field 2 DT[3:0]
LCR18_21	Line 21 Field 1 DT[3:0]	Line 21 Field 2 DT[3:0]	Line 20 Field 1 DT[3:0]	Line 20 Field 2 DT[3:0]	Line 19 Field 1 DT[3:0]	Line 19 Field 2 DT[3:0]	Line 18 Field 1 DT[3:0]	Line 18 Field 2 DT[3:0]
LCR22_24	-	-	Line 24+ Field 1 DT[3:0]	Line 24+ Field 2 DT[3:0]	Line 23 Field 1 DT[3:0]	Line 23 Field 2 DT[3:0]	Line 22 Field 1 DT[3:0]	Line 22 Field 2 DT[3:0]

Reset value:0x00000000h

Reset value for LCR22_24:0x00ff0000h

This array of registers tells the frontend what data type to receive on video lines: 2, 3, 4, ... 23, (24 and after), on each field. At the start of each incoming video line, the relevant register is read, and the data slicer and SERPAR set up to slice the required type of data. On RESET, the LCR registers should be set by software to acquire nothing (DT=1111). Registers LCR2 to LCR23 apply to lines 2 to 23 respectively. Register LCR24 applies to all lines from line 24 to the end of the field. LCR2_5 is provided to cater for non-standard video signals, which may contain data from line number 2 onwards.

DT3 f1 - DT0 f1: Data type to be received on line n, first field (odd).

DT3 f2 - DT0 f2: Data type to be received on line n, second field (even).

Table 73: Data Types

DT	Data Type	Data Rate (Mb/s)	Framing Code[1]	FC Window	Hamming Check[2]	Data Bytes Output (inc Status)
0000	European Teletext (WST625), Chinese Teletext (CCST)	6.9375	0x27	WST625	Always	44
0001	European Closed Caption	0.500	001 binary	CC625	-	4
0010	VPS	5	0x9951[3]	VPS	-	28
0011	European Wide Screen Signalling (WSS625)	5	0x1E3C1F[4]	WSS625	-	16
0100	US Teletext (WST525)	5.7272	0x27	WST525	Always	36

Table 73: Data Types *…continued*

DT	Data Type	Data Rate (Mb/s)	Framing Code[1]	FC Window	Hamming Check[2]	Data Bytes Output (inc Status)
0101	US Closed Caption (Line 21)	0.503	001 binary	CC525	-	4
0110	Gemstar(2x)	1.007	0011101101	Gemstar(2x)	-	6
0111	US Wide Screen Signalling (WSS525)[5]	0.447443	10 binary	WSS525	-	5
1000	Teletext	6.9375	programmable	gen_text	Optional[6]	44
1001	VITC-625	1.8125	10 binary	VITC625	-	11
1010	VITC-525	1.7898	10 binary	VITC525	-	11
1011	Open 1	5	programmable	8-16ms	-	386(max.)[7]
1100	US NABTS	5.7272	programmable	NABTS	Optional	36[8]
1101	Moji (Japanese Text)	5.7272	programmable[9]	MOJI	-	37
1110	Open 2	5.7272	programmable	8-16ms	-	41 (max.)
1111	Do not acquire	N/A	none	disable	-	0

[1] Teletext, programmable and VPS framing codes are specified in reverse transmission order (transmitted LSB first). Closed Caption and Wide Screen Signalling framing codes are specified in transmission order (transmitted MSB first).

[2] 8/4 Hamming check of magazine and packet address contained in bytes 2 and 3 of WST teletext. Note that this overrides the HAM bit in DCR for data types 0 to 7.

[3] The transmitted bit sequence for the VPS framing code is “10 00 10 10 10 01 10 01”.

[4] The transmitted bit sequence for the WSS framing code is “0001 1110 0011 1100 0001 1111”.

[5] Used for Copy Generation Management System (CGMS) and Japanese wide screen signalling.

[6] For data types where hamming checking is optional, hamming checking is enabled or disabled via the HAM bit in DCR.

[7] Exact number of bytes captured depends on when framing code is detected - see Open Data Types Section 3.7.3.10.

[8] NABTS data actually contains only 33 bytes, but 34 bytes are captured so that this Data Type may also be used for WST525 or Moji, if necessary.

[9] Should be set to 0xA7 for Moji.

3.7.5.4 DCS: Data Capture Status (Read)

This register gives information about the acquisition performance of the DCU.

There is no reset value as the register is updated at the end of each line regardless of any other register setting. The register is updated even if the data type is “do not acquire”, or if the ACQ_EN bit of DCR1 is set to ‘0’.

All bits are active high unless otherwise indicated. Signals from the front-end set the relevant bits in a holding latch during the field. On the field boundary the contents are transferred to the status register and the holding latch is reset to 0.

Table 74: DCS Registers

Bit	Symbol	Access	Reset value	Description
Offset 0x1C - DCS				
31	F2	R	0xX	Set to '1' at the start of field 2 of frame; set to '0' at the start of field 1. Field 1 is 'odd', field 2 is 'even'.
30:25	RSD[30:25]	R	0x00	Reserved
24:16	LN[8:0]	R	0xXXX	Line number in current field (1..313). Updated at the start of each line even if the data type is 'do not acquire'
15	FC8V	R	0xX	Teletext data received with no errors in framing code within last field
14	FC7V	R	0xX	Teletext data received with one error in framing code within last field
13	VPSV	R	0xX	Video Programming Signal (VPS) data received within last field
12	WSSV	R	0xX	WSS data received within last field
11	CCV	R	0xX	CC data received within last field
10	RSD_Bit10	R	0x0	Reserved
9	RSD_Bit9	R	0x1	Reserved
8	525R	R	0xX	525-line transmission detect by acquisition line counter
7	DPH	R	0xX	Decoding of Page Headers enabled
6	DMP	R	0xX	Decoding of Magazine and Packet numbers enabled
5:4	RSD[5:4]	R	0x0	Reserved
3:0	DT[3:0]	R	0xX	Type of data received

Remark: The bits FC8V, FC7V, VPSV, WSSV and CCV do not contain information about the current packet; they are intended to give a general indication of acquisition performance on the data types requested by the Line Control Registers during the last field.

3.7.5.5 Interrupt Registers (Read/Write)

Only the least-significant 2 Bits are implemented in each of the 4 registers; bit 0 corresponds to the cvfld (field- rate) interrupt, and bit 1 corresponds to the pktrx (packet-received) interrupt, as shown in table below.

Table 75: Interrupt Registers

Bit	Symbol	Access	Reset value	Description
Offset 0xFE0 - INT_STATUS				
31:2	RSD[31:2]	R	0x00000000	Reserved
1	PKTRX_STATUS	R	0x0	Status of packet received interrupt
0	CVFLD_STATUS	R	0x0	Status of CVBS field sync interrupt
Offset 0xFE4 - INT_ENABLE				
31:2	RSD[31:2]	R	0x00000000	Reserved
1	PKTRX_ENABLE	R/W	0x0	Packet received interrupt enable
0	CVFLD_ENABLE	R/W	0x0	CVBS field sync interrupt enable
Offset 0xFE8 - INT_CLEAR				
31:2	RSD[31:2]	R	0x00000000	Reserved
1	PKTRX_CLEAR	W	0x0	Write '1' to clear named interrupt. Write '0' has no effect

Table 75: Interrupt Registers *...continued*

Bit	Symbol	Access	Reset value	Description
0	CVFLD_CLEAR	W	0x0	Write '1' to clear named interrupt. Write '0' has no effect
Offset 0xFEC - INT_SET				
31:2	RSD[31:2]	W	0x00000000	Reserved
1	PKTRX_SET	W	0x0	Write '1' to set named interrupt. Write '0' has no effect
0	CVFLD_SET	W	0x0	Write '1' to set named interrupt. Write '0' has no effect

INT_ENABLE is read/write. A ‘1’ will enable the interrupt pin corresponding to the actual bit set. Any interrupts which occur while disabled will still set the INT_STATUS register, indicating that an interrupt is pending.

INT_STATUS is read-only and can be interrogated to see which interrupt(s) are pending.

Writing a ‘1’ to the appropriate bit of INT_SET causes an interrupt to become pending and sets the corresponding bit of INT_STATUS, also generating an interrupt on that pin if the corresponding interrupt is enabled. INT_CLEAR is used to reset INT_STATUS and the corresponding interrupt pin by writing ‘1’ to the appropriate bit. INT_SET and INT_CLEAR are not typical registers in that they do not hold the values written to them; they will always read as ‘0’.

Note that the pktrx interrupt is set by the acquisition circuitry whenever a valid packet is captured, regardless of the state of the ACQ_EN control bit or the top-level dt_enable inputs.

Remark: Only the cvfld interrupt is used on the AVIP blocks; the pktrx interrupt is not internally connected. However, it is still possible to detect whether this interrupt would have occurred (i.e. whether any packets have been received) by reading the INT_STATUS register.

3.7.5.6 MODULE_ID (Read)

The MODULE_ID register always returns a fixed value made up of the fields shown in table below, and cannot be written to.

Table 76: Module ID Registers

Bit	Symbol	Access	Reset value	Description
Offset 0xFFC - MOD_ID				
31:16	Id[15:0]	R	0xA00F	Module Identifier.
15:12	Major_rev[3:0]	R	0x3	Major revision. Any revisions that may break software compatibility.
11:8	Minor_rev[3:0]	R	0x0	Minor revision. Any revisions that still keep software compatibility.
7:0	Aperture[7:0]	R	0x00	Aperture size. Encoded as (aperture size/4K)-1; so 0 means 4K (the default)

3.8 ITU656

3.8.1 Functional Overview

The ITU output formatter combines video data from the VIDDEC block and VBI data (e.g. Teletext) from the DCU into a ITU 656 style data stream. It is mainly intended to transfer the output data stream externally to the PNX8550 or 3D comb filter device, but the output data stream may also be readable by other ITU 656 input devices.

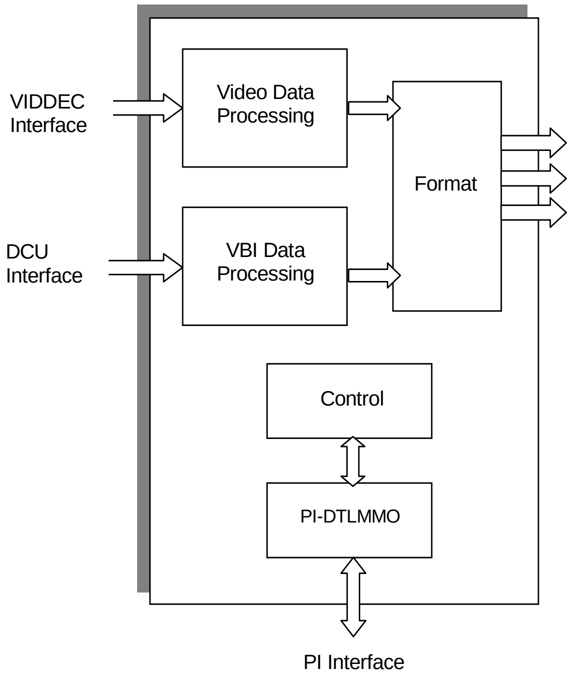


Fig 55.Formatter Block Diagram

3.8.2 ITU656 Formatter Data Interfaces

The ITU formatter receives data from 3 sources:

- YUV data as video input signals, sourced from the VIDDEC block. This YUV data is either decoded CVBS signals, YC or YUV input signals.
- VBI sliced data sourced from the DCU in packets between 4 and 46 Bytes depending on the format of the data packet (e.g. Euro WST, Closed Caption).
- CVBS data, sourced from the VIDDEC (after SRC and AGC).

This data is formatted into an output data stream, which is ITU 601/656/1364 style, this will contain video and VBI data.

VBI data, text etc., is formatted into the VBI region and is preceded by an ANC header.

In normal mode video data, luminance and chrominance, is formatted into the active video region. It is preceded by the SAV timing reference and terminated by the EAV timing reference.

In 3D comb filter mode the CVBS color burst data is formatted into the HBI interval and preceded by an ANC header. Video data in this mode will consist of CVBS and chrominance samples and is formatted into the active video region.

The ANC data packet is a Type 2 format.

The ITU output is compliant to ITU-R BT.656-4[1] and ITU-R BT.1364[2] with the exception of:

- A data valid signal is used to validate the data (not in stuttered clock mode).
- No checksum is added to the ANC data stream

The ITU 656 formatter implementation outputs VBI (text) in the VBI interval within the ITU 656 stream but doesn’t guarantee maintaining the relationship of where the line was captured in the original input analogue stream to the output ITU 656 stream.

i.e. The text data may have been captured from line 5 but may be output in the ITU-656 stream on line 6, however it will always be output in the vertical blanking interval. If the VBI data is captured in the last line of the VBI, it may in some modes be transmitted in the first line of the next of the field.

The PNX2015 ITU656 formatter can only carry over 4 lines of data to the next field, that is sufficient for Gemstar.

Field numbering for NTSC and PAL have opposite odd even field allocations as shown in Table 77. Like NTSC this User Manual, refers to the odd field as field 1 and the even field as field 2.

Table 77: Odd even field allocation for NTSC and PAL

Field_1 Signal	Field_0 Signa	HSYNC# and VSYNC# Timing Relationship or BT.656 F Bit	NTSC Field Number		PAL Field Number	
0		field 1	1	odd field	1	even field
0		field 2	2	even field	2	odd field
0	1	field 1	3	odd field	3	even field
0	1	field 2	4	even field	4	odd field
1		field 1	-	-	5	even field
1		field 2	-	-	6	odd field
1	1	field 1	-	-	7	even field
1	1	field 2	-	-	8	odd field

AVIP Formatter Registers

VF_SYNC registers (SYNC_VALUE_F0/1 bitfields)

- when a vertical sync is received from VIDDEC then the Formatter line-counter is set to the SYNC_VALUE_F0/1 (according to which field). F0 and F1 are Frame 0 and Frame 1.

FIELD 2 registers (start/stop bitfields)

- these registers determine the field indication in the output ITU656 stream
- when the line-counter matches either the 'start' values then it identifies that as the start of field 2

- when the line-counter matches either of the 'stop' values then it identifies that the end of field 2, so start of field 1
- under stable video conditions then the FIELD1 register 'start' and 'stop' values are enough, and the field bit will toggle appropriately
- under unstable video conditions (e.g. VCR trick modes) then (similar to VBI registers) the line-counter may not always reach the FIELD1 start and end values.

VBI1/2/3/4 registers (start/end bitfields)

- when the line-counter matches any of the 'start' values then the next line is identified as being a VBI line
- when the line-counter matches any of the 'end' values then the next line is identified as being a video line (not VBI)

The VBI1 and VBI2 registers are enough to identify the VBI in stable video, however something else is needed if the number of lines per field is unstable (VCR trick-modes). This is where VBI3 and VBI4 are needed.

In VCR fast forward mode, there are fewer lines in the field.

- This means that when the VSYNC is received by the formatter, the line-counter is advanced from it's current value and set to the next field-start line number.
- If this skips past the VBI start line number then the formatter will not know that this should be VBI.
- The way to solve this is by using the VBI3/4 registers.
- These give an additional opportunity to set a VBI start line number, and should be set to the same line numbers as the F0/F1 in the VF_SYNC.
- This ensures that even with short fields in VCR fast-forward, the line-counter will still match a VBI 'start' value and know to set the timing-codes appropriately.

To avoid synchronisation errors during VCR fast-forward set:

- Field2_Start ==VF_SYNC_F1 == VBI4 _start
- Field2_Stop ==VF_SYNC _F0 == VBI3 _start

3.8.3 Control Registers

3.8.3.1 ITU656 Formatter Registers

Table 78: ITU656 Formatter Register Summary

Address	Name	Description
0x0	CONFIG	Provides IP configuration
0x4	DATA_IDToVBI	ANC VBI Data ID
0x8	DATA_IDToHBI	ANC HBI Data ID
0xC	CAPTURE	Capture parameters
0x10	FIFO	FIFO Register
0x14	VF_CONTROL	VF CONTROL Register
0x18	VF_SYNC	VF SYNC Register
0x1C	FIELD_1	FIELD 1 register

Table 78: ITU656 Formatter Register Summary *…continued*

Address	Name	Description
0x20	FIELD_2	FIELD 2 register
0x24	VBI_1	VBI 1 register
0x28	VBI_2	VBI 2 register
0x2C	VBI_3	VBI 3 register
0x30	VBI_4	VBI 4 register
0x34	PROG_HBI	Prog HBI register
0x38	YUV_Offset	Prog HBI register
0xFC4	DTM_SYNC	DTL to module Metastability
0xFC8	MTD_SYNC	Module to DTL Metastability
0xFCC	DEBUG	Debug control
0xFE0	INT_STATUS	Interrupt Status
0xFE4	INT_ENABLE	Interrupt Enable
0xFE8	INT_CLEAR	Interrupt Clear
0xFEC	INT_SET	Interrupt Set
0xFFC	MODULE_ID	ITU656 formatter ID

3.8.3.2 CONFIG Register

Table 79: CONFIG Register

Bit	Symbol	Access	Value	Description
Offset 0x0 - CONFIG				
31:16	RSD[31:16]	R	0x0000	Unused
15	DVO_ENABLE	R/W	0x0	1=DVO outputs enabled
14	INPUT_TEST_MODE	R/W	0x0	1=Output mono bar test pattern
13	OUTPUT_TEST_MODE	R/W	0x0	1=Output colour bar test pattern
12	PROGESSIVE_MODE	R/W	0x0	1=Progressive mode, timing flags always 1st field
11	CLOCK_STUTTER	R/W	0x0	1=ITU clock stuttered
10	CLOCK_INVERT	R/W	0x0	1=ITU data clocked from Formatter on rising edge
9	DC_JUSTIFIED	R/W	0x0	1=VBI ANC Data Count justified to an integer number of 4 blocks, for usage in 8-bit ITU
8	DITHER	R/W	0x0	1=LSB of 9-bit video will be dithered into 8-bit ITU
7	UV_COMPL	R/W	0x0	1=MSB of 9-bit UV video will be inverted
6	CVBS_COMPL	R/W	0x0	1=MSB of 9-bit CVBS will be inverted
5	VBI_ONLY	R/W	0x0	1=Even during active video (non-vertical blanking) text is transmitted from the DCU and inserted into ITU data stream.

Table 79: CONFIG Register *...continued*

Bit	Symbol	Access	Value	Description
4:3	VBI_CONTROL[1:0]	R/W	0x0	Modes for Avoidance of '00' and 'FF' in data stream during VBI transmission are: '00'=Pure Text - VBI bytes are shifted left (left aligned). The risk to get an unwanted sequence with 'FF', '00', EAV/SAV is high for 8-bit recognition, but zero for 10-bit recognition, as the two LSBs are modified to prevent that. '01'=One bit wrong - (in case of 8-bit ITU) VBI bytes will be modified in the LSB bit to prevent from getting '00' or 'FF' in the data. '10'=No Test Shift - VBI bytes will not be shifted left and the two MSBs are modified to prevent from getting '00' or 'FF' (for 10-bit use). '11'=Nibble - VBI data bytes will be transmitted via two nibbles, low nibble first filled with '1010'.
2	COLUMBUS	R/W	0x0	1=656 output data stream contains CVBS/Chrominance samples.
1:0	MODE[1:0]	R/W	0x0	See Table 1 (FRS V1.0) Supported Video Standards

Mode: Bits 1:0 of the CONFIG register selects the mode of operation for the formatter.

Table 80: Supported Video Standards

Mode	Total Pixels per Line/Active Pixels per Line	CONFIG(1:0)
0	864/720	00
1	858/720	01
2	825/720	10
3	990/720 (default)	11

In mode 3 it is possible to modify the total pixels per line from 722 to 1021 but the active pixels per line will always be 720. This is achieved via the PROG_HBI register.

3D comb filter: Bit 2 of the CONFIG register selects the ITU 656 stream format

Normal mode (CONFIG(2) = 0)

Video data output stream format in normal mode.

The video data words are conveyed as a 27Mwords/s in 1fh mode and 54Mwords/s in 2fh mode and shall be multiplexed in the following order:

C_{B0}, Y_0, C_{R0}, Y_1, C_{B2}, Y_2, C_{R2}, Y_3, C_{B4}, Y_4, C_{R4}, …

3D comb filter mode (CONFIG(2) = 1)

Video data output stream format in 3D comb filter mode.

The video data words are conveyed as a 27Mwords/s in 1fh mode and 54Mwords/s in 2fh mode and shall be multiplexed in the following order:

CB0, CVBS0, CR0, CVBS1, CB2, CVBS2, CR2, CVBS3, CB4, CVBS4, CR4, …

In addition to replacing luminance samples with CVBS samples, the color burst information is also transmitted in the HBI in 3D comb filter mode.

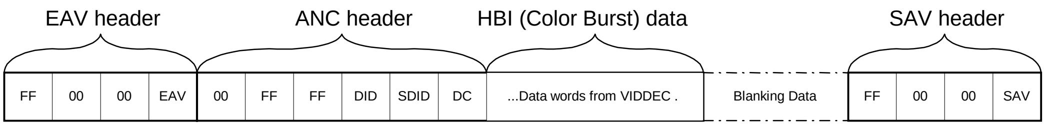


Fig 56.Insertion of HBI data in ITU data stream

VBI_CONTROL: Bits (4:3) of the CONFIG register select the mode for avoidance of ‘00’ and ‘FF’ in the data stream during VBI data transmission.

Pure Text Mode (CONFIG(4:3) = 00)

As VBI data has only 8 bits a mapping has to be performed prior to data transmission. The default mapping is shown in Figure 57. This mode of operation is for use in 10-bit ITU where signalling is detected across the full 10 bits of data.

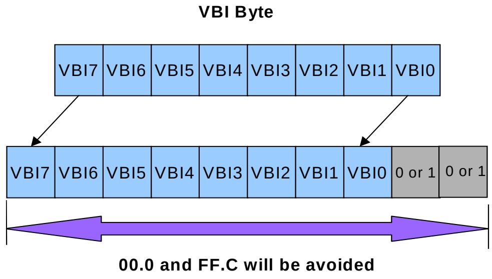


Fig 57.Shifting of bits in pure text mode

Nibble Mode (CONFIG(4:3) = 11)

Nibble mode is performed to ensure 00 or FF timing codes are not generated by the VBI data. Each byte is split in two nibbles. The remaining nibble in each byte is filled with ‘1010’. The lower nibble is transmitted first. The nibbles are output on the upper 8 bits of the data bus. This mode of operation is for use in 10-bit ITU where signalling is detected across only the upper 8 bits or 8-bit ITU and the possible data corruptions of 1 Wrong bit mode are unacceptable.

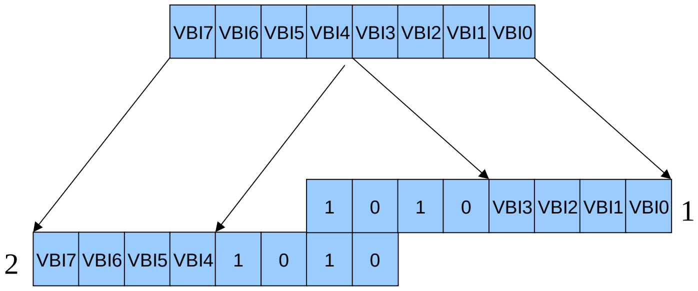


Fig 58.Generation of bytes in nibble mode

Wrong Bit Mode (CONFIG(4:3) = 01)

For use in 8 bit ITU, VBI bytes are shifted left then the LSB (bit2 of the 10 bits) modified to prevent 00 or FF in the data bits 9:2. This mode of operation is for use in 10-bit ITU where signalling is detected across only the upper 8 bit or 8-bit ITU and possible data corruptions are acceptable.

No text Shift (CONFIG(4:3) = 10)

VBI bytes will not be shifted left and the 2 MSB's are modified to prevent from getting 00.0-00.C and FF.0-FF.C. This mode of operation is for use in 10-bit ITU use where VBI data is transmitted non-standard position of ITU_OUT(7:0).

VBI_ONLY: Bit 5 of the CONFIG register selects VBI_ONLY mode (CONFIG(5) = 1). In this mode the formatter will not transmit any video data from the VIDDEC and only transmits VBI data from the DCU. In this mode the formatter can utilize any video line for VBI transmission.

CVBS_COMPL: Bit 6 of the CONFIG register selects CVBS_COMPL mode (CONFIG(6) = 1). In this mode the formatter will invert the MSB of the 9 bit incoming CVBS data. This mode can be used to convert from signed to unsigned data.

UV_COMPL: Bit 7 of the CONFIG register selects UV_COMPL mode (CONFIG(7) = 1). In this mode the formatter will invert the MSB of the 9 bit incoming UV data. This mode can be used to convert from signed to unsigned data.

DITHER: Bit 8 of the CONFIG register selects dithering of Y and UV data (CONFIG(8) = 1). The dithering function can be selected for the video data input to reduce the bus width from 9 to 8 bits. The 8 bits output shall be in the upper 8 bits (9:2) of the ITU output stream.

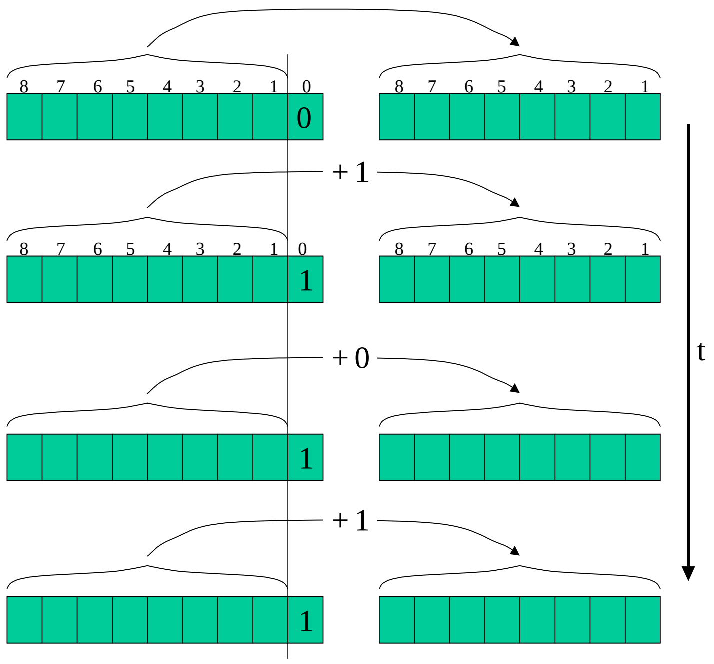


Fig 59.Dithering of 9 bit video data to 8 bits

If the LSB is ‘0’, the highest 8 bits are simply mapped to the destination (top line). If the LSB is ‘1’, an alternately ‘1’ or ‘0’ is added to the highest 8 bits before mapping.

DC_JUSTIFIED: Bit 9 of the CONFIG register justifies the data count into an integer number of 4 words (CONFIG(9) = 1). This mode is for use in 8-bit ITU where the data count is active only on ITU_OUT(9:2), hence the data count requires rounding up to the next multiple of 4.

CLOCK_INVERT: Bit 10 of the CONFIG register selects an inversion of the ITU_CLOCK

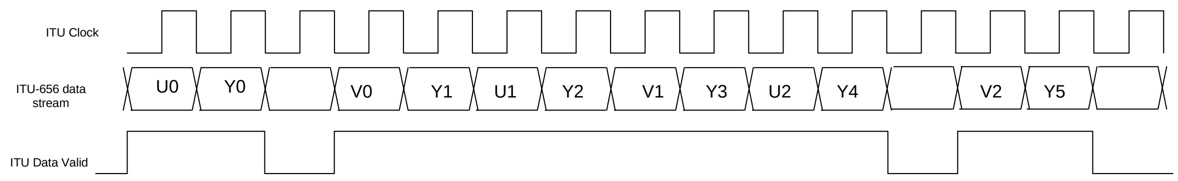


Fig 60.ITU output data stream (CONFIG(10) = 0)

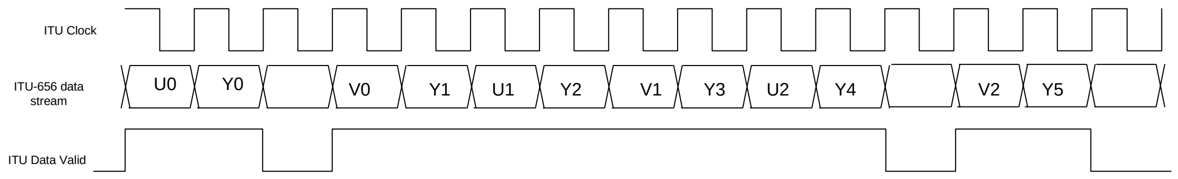


Fig 61.ITU output data stream (CONFIG(10) = 1)

CLOCK_STUTTER: Bit 11 of the CONFIG register selects a stuttered ITU_CLOCK.

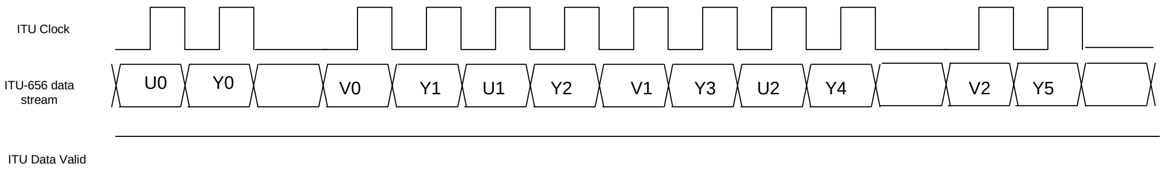


Fig 62.ITU output data stream (CONFIG(11) = 1)

PROGRESSIVE_MODE: Bit 12 of the CONFIG register selects PROGRESSIVE_MODE (CONFIG(12) = 1). In this mode the SAV and EAV flags will always indicate first field. The VBI and HBI DID bytes will also only transmit the indicator for first field.

OUTPUT_TEST_MODE: Bit 13 of the CONFIG register will select a color bar test pattern (CONFIG(13) = 1). This pattern is generated on the output of the formatter. To utilize this test mode ensure the formatter is not in 3D comb filter mode.

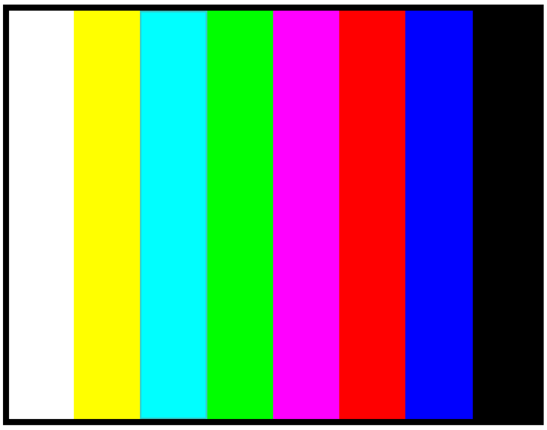


Fig 63.Color Bar Test Pattern

INPUT_TEST_MODE: Bit 14 of the CONFIG register will select a mono bar test pattern (CONFIG(14) = 1). This pattern is generated on the input of the formatter. Ensure the formatter is not in 3D comb filter mode, or the OUTPUT_TEST_MODE active, in order to utilize this test mode.

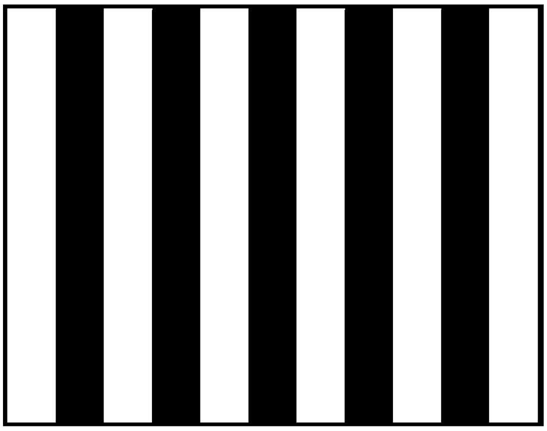


Fig 64.Mono Bar Test Pattern

DVO_ENABLE: Used to enable the DVO output pins DVO_DATA (9.0), DVO_CLK and DVO_VALID

3.8.3.3 Data Identification Register - VBI data

Table 81: Data Identification Register - VBI data

Bit	Reset Value	Name	Description
31:30		RSD	Unused.
29:20	0	SDID_VBI	ANC VBI SDID word.
19:10	0	DID2_VBI	ANC VBI DID word for field 2 (even field).
9:0	0	DID1_VBI	ANC VBI DID word for field 1 (odd field).

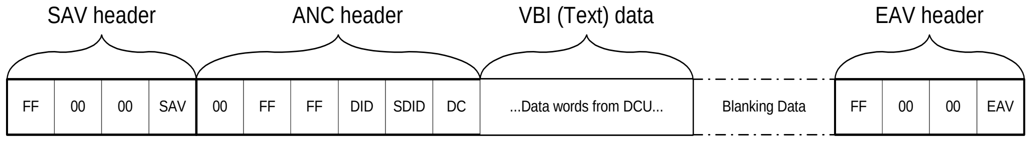


Fig 65.Insertion of VBI data in ITU data stream

The data identification register for the VBI data will control the values for the DID and SDID for the ANC VBI header. The DID is selected is DID1_VBI or DID2_VBI depending on the field transmitted except when PROGESSIVE_MODE is selected then DID1_VBI is always used.

3.8.3.4 Data Identification Register - HBI data

Table 82: Data Identification Register - HBI data

Bit	Reset Value	Name	Description
31:30	-	RSD	Unused.
29:20	0	SDID_HBI	ANC HBI SDID word.
19:10	0	DID2_HBI	ANC HBI DID word for field 2 (even field).
9:0	0	DID1_HBI	ANC HBI DID word for field 1 (odd field).

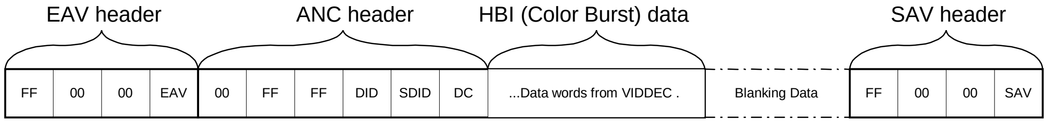


Fig 66.Insertion of HBI data in ITU data stream

The data identification register for the HBI data will control the values for the DID and SDID for the ANC HBI header. The DID is selected for DID1_HBI or DID2_HBI depending on the field transmitted except when PROGESSIVE_MODE is selected then DID1_HBI is always used.

3.8.3.5 CAPTURE Register

Table 83: CAPTURE Register

Bit	Reset Value	Name	Description
31:30		RSD	Unused.
29:24	0	YUV_LATENCY	YUV latency from 1st byte to start of frame
23:21	0	RSD	Unused.
20	0	SYNC_TO_HSYNC	1 = Sync timing to HSYNC. 0 = Sync to HSY_OUT.
19	0	RSD	Unused.

Table 83: CAPTURE Register *…continued*

Bit	Reset Value	Name	Description
18:8	0x03E	CVBS_LATENCY	CVBS latency from 1st byte to start of frame
7	0	RSD	Unused.
6:0	0x74	CVBS_FIFO_OFFSET	CVBS offset for 1st byte in buffer

This register is used to determine:

- The first CVBS sample of a line.
- The CVBS latency and hence buffer use.
- The YUV latency and hence buffer use.
- Timing sync to HSY_OUT or HSYNC.

3.8.3.6 FIFO Register

Table 84: FIFO Register

Bit	Reset Value	Name	Description
31:25		RSD	Unused.
24	0	FIFO_CONTROL	Operate FIFO buffering
23	0	RSD	Unused.
22:12	0	FIFO_WIN_STOP	FIFO window stop value
11	0	RSD	Unused.
10:0	0	FIFO_WIN_START	FIFO window start value

The FIFO register is for use when inputting a line locked clock into the formatter (AVIP) in design proving and test modes. It should not be used for functional operation and therefore should be set to 0x00000000.

3.8.3.7 VF CONTROL Register

Table 85: VF Control Register

Bit	Reset Value	Name	Description
31:14		RSD	Unused.
13	0	EAV_UPDATE	1 = Update VBI and field indicators with EAV. Only use when using internally generated VBI/field indicators and syncing to hsync.
12	0	VBI_FIELD_CONTROL	1 = Use VBI and field indicators generated within the ITU656 formatter.
11		RSD	Unused.
10:0	0x271	LINE_NUMBER	Quantity of lines per frame

Registers VF_CONTROL, VF_SYNC, FIELD_1/2, VBI1/4 are used to generate field and VBI indicators within the formatter design so the external indicators, from VIDDEC, do not have to be used.

3.8.3.8 VF SYNC Register

Table 86: VF Sync Register

Bit	Reset Value	Name	Description
31:23		RSD	Unused.
22:12	0x13C	SYNC_VALUE_F1	Sync value for field one (F1)
11		RSD	Unused.
10:0	0x003	SYNC_VALUE_F0	Sync value for field zero (F0)

3.8.3.9 FIELD 1 Register

Table 87: Field 1 Register

Bit	Reset Value	Name	Description
31:23		RSD	Unused.
22:12	0x271	FIELD_STOP_1	Field stop value
11		RSD	Unused.
10:0	0x138	FIELD_START_1	Field start value

Used to set field indicator start and stop line. The start condition will set the field indicator high indicating an even (second) field, the stop condition will set the field indicator low indicating an odd (first) field.

3.8.3.10 FIELD 2 Register

Table 88: Field 2 Register

Bit	Reset Value	Name	Description
31:23		RSD	Unused.
22:12	0x003	FIELD_STOP_2	Field stop value
11		RSD	Unused.
10:0	0x13C	FIELD_START_2	Field start value

Used to set the field indicator start and stop line under synchronisation conditions.

3.8.3.11 VBI 1 Register

Table 89: VBI 1 Register

Bit	Reset Value	Name	Description
31:23		RSD	Unused.
22:12	0x14F	VBI_STOP_1	VBI stop value
11		RSD	Unused.
10:0	0x136	VBI_START_1	VBI start value

Used to set VBI indicator start and stop line. The start condition will set the VBI indicator high indicating a VBI region, the stop condition will set the VBI indicator low indicating a video region.

3.8.3.12 VBI 2 Register

Table 90: VBI 2 Register

Bit	Reset Value	Name	Description
31:23		RSD	Unused.
22:12	0x016	VBI_STOP_2	VBI stop value
11		RSD	Unused.
10:0	0x26F	VBI_START_2	VBI start value

Used to set VBI indicator start and stop line. The start condition will set the VBI indicator high indicating a VBI region, the stop condition will set the VBI indicator low indicating a video region.

3.8.3.13 VBI 3 Register

Table 91: VBI 3 Register

Bit	Reset Value	Name	Description
31:23		RSD	Unused.
22:12	0x7FF	VBI_STOP_3	VBI stop value
11		RSD	Unused.
10:0	0x003	VBI_START_3	VBI start value

Used to set the VBI indicator start and stop line under synchronisation conditions.

3.8.3.14 VBI 4 Register

Table 92: VBI 4 Register

Bit	Reset Value	Name	Description
31:23		RSD	Unused.
22:12	0x7FF	VBI_STOP_4	VBI stop value
11		RSD	Unused.
10:0	0x13C	VBI_START_4	VBI start value

Used to set the VBI indicator start and stop line under synchronisation conditions.

3.8.3.15 PROG HBI Register

Table 93: PROG HBI Register

Bit	Reset Value	Name	Description
31:22	0	HBI_DC	Horizontal blanking interval data count value
21:11	0	CB_OFF_SLOT	color burst off slot
10:0	0x7B8	SAV_SLOT	First SAV slot

Used to construct a programmable HBI when mode 3 is selected, see Table 80.

3.8.3.16 YUV Offset Register

Table 94: YUV Offset Register

Bit	Reset Value	Name	Description
31:22		RSD	Unused.
21:16	0	V_OFFSET	V offset value
15:14		RSD	Unused.
13:8	0	U_OFFSET	U offset value
7:6		RSD	Unused.
5:0	0	Y_OFFSET	Y offset value

Used to offset the YUV samples that may be mis-aligned i.e.

To delay (retard) the Y sample by 4 wrt to the U V sample:

Y_OFFSET = 64 - 4 = 60 = 0x3C.

To delay (retard) the U sample by 13 wrt to the Y V sample:

U_OFFSET = 64 - 13 = 51= 0x33.

The maximum delay is 63, a value of 0 results in no delay. Pixels at the start and end of a line are invalidated by the same value as the offset used.

3.8.3.17 Interrupt Registers

The interrupt status is controlled by a set of four registers. The interrupts registers provide a means of indicating DCU buffer overflows within the ITU656 formatter. A buffer overflow will indicate too many lines of VBI data from the DCU wrt to VBI indicators. Interrupts should be cleared after the ITU656 Formatter has been reset, or after the Viddec has achieved lock.

The interrupt can be enabled by means of setting the corresponding bit in the INT_ENABLE register. If an interrupt condition is not enabled the bit is still set in the INT_STATUS register but the global ITU656 formatter interrupt line will not be raised. The interrupt bits in the INT_STATUS register should be cleared manually by writing a 1 to the same bit position in the INT_CLEAR register. The INT_SET register can be used to manually force an interrupt.

INT_STATUS Register

Table 95: INT_STATUS Register

Bit	Reset Value	Name	Description
31:1		RSD	Unused.
0	0	DCU_BUF_INT_STATUS	Interrupt status of DCU buffers

INT_ENABLE Register

Table 96: INT_ENABLE Register

Bit	Reset Value	Name	Description
31:1		RSD	Unused.
0	0	DCU_BUF_INT_ENABLE	Enable interrupt for DCU buffers

INT_CLEAR Register

Table 97: INT_CLEAR Register

Bit	Reset Value	Name	Description
31:1		RSD	Unused.
0	0	DCU_BUF_INT_CLEAR	Clear interrupt for DCU buffers

INT_SET Register

Table 98: INT_SET Register

Bit	Reset Value	Name	Description
31:1		RSD	Unused.
0	0	DCU_BUF_INT_SET	Set interrupt for DCU buffers

3.8.3.18 MODULE_ID Register

Table 99: MODULE_ID Register

Bit	Reset Value	Name	Description
31:16	A05F	ID	Module ID. This field identifies the block as type ITU 656 formatter.
15:12	0	MAJ_REV	Major Revision ID. This field is incremented by 1 when changes introduced in the block result in software incompatibility with the previous version of the block. First version default = 0.
11:8	0	MIN_REV	Minor Revision ID. This field is incremented by 1 when changes introduced in the block result in software compatibility with the previous version of the block. First version default = 0.
7:0	0	APERTURE	Aperture Size. Identifies the MMIO aperture size in units of 4kB. The ITU 656 formatter has an aperture size of 4kB. Aperture = 0; 4kB.

3.8.3.19 Debug Control Register

It is possible within AVIP to access internal signals in the ITU656 Formatter module via the testrail output. The DEBUG_OUTPUT_SELECT control bit will select the debug outputs and DEBUG_SEL selects the various debug rails as detailed in Table 100.

Table 100: DEBUG Register

Bit	Reset Value	Name	Description
31:3		RSD	Unused.
2:1	0	DEBUG_SEL	Debug rail selection: 00: tst_rail_out = core_debug 01: tst_rail_out = yuv_debug 10: tst_rail_out = cvbs_debug 11: tst_rail_out = dcu_debug
0	0	DEBUG_OUTPUT_SELECT	1= debug outputs 0 = scan chain outputs

Table 101: DEBUG Signals

TestRail	core_debug	yuv_debug	cvbs_debug	dcu_debug
23	sav_tx_req(0)	hsy_out_itu	line_sync	buffer_full_itu(3)
22	eav_tx_req(0)	evenfield_itu	cvbs_tx_ack	buffer_full_itu(2)
21	vbi_anc_tx_req(0)	vblank_itu	cvbs_valid_itu	buffer_full_itu(1)
20	hbi_anc_tx_req(0)	chroma	w_address_start_itu(6)	buffer_full_itu(0)
19	uv_tx_req	uv_dither_history	w_address_start_itu(5)	buffer_emptied(1)
18	y_tx_req	y_dither_history	w_address_start_itu(4)	buffer_emptied(0)
17	cvbs_tx_req	line_sync	w_address_start_itu(3)	dcu_r_buffer(1)
16	cvbs_cb_tx_req	hsync_itu	w_address_start_itu(2)	dcu_r_buffer(0)
15	dbc_tx_req	y_tx_ack	w_address_start_itu(1)	dcu_data(9)
14	dbl_tx_req	uv_tx_ack	w_address_start_itu(0)	dcu_data(8)
13	dcu_tx_req	uv_valid_itu	w_address_itu(6)	dcu_data(7)
12	hsync_itu	y_valid_itu	w_address_itu(5)	dcu_data(6)
11	hsy_out_itu	w_address_itu(5)	w_address_itu(4)	dcu_data(5)
10	pixel slot(10)	w_address_itu(4)	w_address_itu(3)	dcu_data(4)
9	pixel slot(9)	w_address_itu(3)	w_address_itu(2)	dcu_data(3)
8	pixel slot(8)	w_address_itu(2)	w_address_itu(1)	dcu_data(2)
7	pixel slot(7)	w_address_itu(1)	w_address_itu(0)	dcu_data(1)
6	pixel slot(6)	w_address_itu(0)	r_address(6)	dcu_data(0)
5	pixel slot(5)	r_address(5)	r_address(5)	r_address(5)
4	pixel slot(4)	r_address(4)	r_address(4)	r_address(4)
3	pixel slot(3)	r_address(3)	r_address(3)	r_address(3)
2	pixel slot(2)	r_address(2)	r_address(2)	r_address(2)
1	pixel slot(1)	r_address(1)	r_address(1)	r_address(1)
0	pixel slot(0)	r_address(0)	r_address(0)	r_address(0)

3.8.4 Video Line Interface Signal Structure

Table 102 to Table 107 detail the ITU frame structure for the various mode of operation.

3.8.4.1 AVIP (Mode 0) in 3D Comb Filter Mode

Table 102: AVIP (Mode 0) in 3D Comb Filter Mode

656 data output	Data Input Type
Slot 1440	End of active Video - FF.C
Slot 1441	End of active Video - 00.0
Slot 1442	End of active Video - 00.0
Slot 1443	End of active Video - EAV
Slot 1444	Ancillary Data Flag - 00.0
Slot 1445	Ancillary Data Flag - FF.C
Slot 1446	Ancillary Data Flag - FF.C
Slot 1447	Data ID Type 2 (DID)
Slot 1448	Secondary data ID (SDID)

Table 102: AVIP (Mode 0) in 3D Comb Filter Mode *…continued*

656 data output	Data Input Type
Slot 1449	Data Count (DC)
Slot 1450	Horizontal Digital Blanking Sample 0
Slot 1451	Horizontal Digital Blanking Sample 1
Slot 1452 - Slot 1591	Horizontal Digital Blanking Samples 2 -141
Slot 1592	Horizontal Digital Blanking Sample 142
Slot 1593	Horizontal Digital Blanking Sample 143
Slot 1594	Digital Blanking Chrominance - 80.0
Slot 1595	Digital Blanking Luminance - 10.0
Slot 1596 - Slot 1721	Digital Blanking Chrominance / Luminance
Slot 1722	Digital Blanking Chrominance - 80.0
Slot 1723	Digital Blanking Luminance - 10.0
Slot 1724	Start of active Video - FF.C
Slot 1725	Start of active Video - 00.0
Slot 1726	Start of active Video - 00.0
Slot 1727	Start of active Video - SAV
Slot 0	Chrominance Blue Sample 0 = CB0
Slot 1	CVBS Sample 0 = CVBS0
Slot 2	Chrominance Red Sample 0 = CR0
Slot 3	CVBS Sample 1 = CVBS1
Slot 4 - Slot 1435	CVBS Samples - Chrominance Samples
Slot 1436	Chrominance Blue Sample 359 = CB359
Slot 1437	CVBS Sample 718 = CVBS718
Slot 1438	Chrominance Red Sample 359 = CR359
Slot 1439	CVBS Sample 719 = CVBS719

3.8.4.2 AVIP (Mode 1) in 3D Comb Filter Mode

Table 103: AVIP (Mode 1) in 3D Comb Filter Mode

656 data output	Data Input Type
Slot 1440 - 1443	End of active Video
Slot 1444 - 1449	Ancillary Data Header
Slot 1450 - 1587	Horizontal Digital Blanking Samples
Slot 1588 - 1711	Digital Blanking Chrominance/Luminance
Slot 1712 - 1715	Start of active Video
Slot 0 - 1439	Chrominance/CVBS Samples

3.8.4.3 AVIP (Mode 0) in PNX8550 mode

Table 104: AVIP (Mode 0) in PNX8550 mode

656 data output	Data Input Type
Slot 1440	End of active Video - FF.C
Slot 1441	End of active Video - 00.0
Slot 1442	End of active Video - 00.0
Slot 1443	End of active Video - EAV
Slot 1444	Digital Blanking Chrominance - 80.0
Slot 1445	Digital Blanking Luminance - 10.0
Slot 1446 - Slot 1721	Digital Blanking Chrominance / Luminance
Slot 1722	Digital Blanking Chrominance - 80.0
Slot 1723	Digital Blanking Luminance - 10.0
Slot 1724	Start of active Video - FF.C
Slot 1725	Start of active Video - 00.0
Slot 1726	Start of active Video - 00.0
Slot 1727	Start of active Video - SAV
Slot 0	Chrominance Blue Sample 0 = CB0
Slot 1	Luminance Sample 0 = Y0
Slot 2	Chrominance Red Sample 0 = CR0
Slot 3	Luminance Sample 1 = Y1
Slot 4 - Slot 1435	Luminance Samples - Chrominance Samples
Slot 1436	Chrominance Blue Sample 359 = CB359
Slot 1437	Luminance Sample 718 = Y718
Slot 1438	Chrominance Red Sample 359 = CR359
Slot 1439	Luminance Sample 719 = Y719

3.8.4.4 AVIP (Mode 1) in PNX8550 mode

Table 105: AVIP (Mode 1) in PNX8550 mode

656 data output	Data Input Type
Slot 1440 - 1443	End of active Video
Slot 1444 - 1711	Digital Blanking Chrominance/Luminance
Slot 1712 - 1715	Start of active Video
Slot 0 - 1439	Chrominance/Luminance Samples

3.8.4.5 AVIP (Mode 2) in PNX8550 mode

Table 106: AVIP (Mode 2) in PNX8550 mode

656 data output	Data Input Type : ATSC source
Slot 1440 - 1443	End of active Video

Table 106: AVIP (Mode 2) in PNX8550 mode *…continued*

656 data output	Data Input Type : ATSC source
Slot 1444 - 1645	Digital Blanking Chrominance/Luminance
Slot 1646 - 1649	Start of active Video
Slot 0 - 1439	Chrominance/Luminance Samples

3.8.4.6 AVIP (Mode 3) in PNX8550 mode

Table 107: AVIP (Mode 3 - default) in PNX8550 mode

656 data output	Data Input Type : ATSC source
Slot 1440 - 1443	End of active Video
Slot 1444 - 1975	Digital Blanking Chrominance/Luminance
Slot 1976 - 1979	Start of active Video
Slot 0 - 1439	Chrominance/Luminance Samples

3.9 Audio Processing

3.9.1 General Description

The Sound Processor Core in AVIP forms a significant part of the audio subsystem in a Digital TV application. It performs demodulation and decoding of a wide range of analogue terrestrial TV sound standards, and provides a wealth of audio processing and enhancement features relevant for use in a TV set. It interfaces to the PNX3000 device for analogue inputs including modulated audio from a tuner block, and contains 12 Audio D/A Converters for output of analogue audio to the set’s loudspeakers and external connectors. A standard I^{2}S format interface allows connection to a multi-channel digital audio decoder, the output of which can be processed and enhanced in the Sound Core before being rendered on the DAC outputs. The Sound Core has sufficient capacity to process audio signals for multiple outputs simultaneously, such as a multi-channel loudspeaker set, stereo headphones, and line-level outputs for recording or connection to an external amplifier.

An easy-to-use control concept is provided for easiest programming of the very complex functionality of the Sound Processor Core. Pre-defined setups are available for all implemented sound processing modes. A very flexible loudspeaker switching concept allows it to adapt the pre-defined setups to the specific loudspeaker application.The built-in intelligence for pre-defined standards and Auto Standard Detection (ASD) allows an easy setup of the demodulator and decoder part.

The control concept for the demodulator and decoder (DEMDEC) is based on the following features:

- Easy demodulator setup for all implemented standards with Demodulator and Decoder Easy Programming (DDEP) for a pre-selected standard or combined with Auto Standard Detection (ASD) for automatic detection of a transmitted standard
- Automatic decoder configuration and signal routing depending on the selected or detected standard
- FM overmodulation adaptation option to avoid clipping and distortion

The control concept for the audio processor is based on the following features:

- Pre-defined setups for the sound processing modes like Dolby® Pro Logic® II and Virtual Dolby® Surround
- Flexible configuration of audio outputs to the loudspeaker configuration with an additional output crossbar
- Master volume function

3.9.2 Supported Standards

The multistandard capability of the Sound Processor Core covers all terrestrial TV sound standards, FM Radio and satellite FM.

The AM sound of L/L' standard is normally demodulated in the 1st sound IF. The resulting audio signal has to be entered into the mono audio input of the Sound Processor Core. A second possibility is to use the internal AM demodulator stage; however, this may result in limited performance because of video crosstalk in the IF circuitry.

Korea has a stereo sound system similar to Europe. It is supported by the Sound Processor Core. Differences include deviation, modulation contents and identification. It is based on M standard.

Other features of the DEMDEC are:

- M/BTSC and N standards supported
- M/Japan (EIAJ) supported
- FM Radio stereo decoding
- Alignment-free, fully digital system
- For BTSC full dbx performance (non dbx version is possible by hardware programming)
- SAP demodulation (without dbx) simultaneously with stereo decoding, or mono plus SAP with dbx
- Line/pilot frequency selectable from 15.734 kHz and 15.625 kHz (or automatic detection / auto search)
- High selectivity for pilot detection, high robustness against high-frequency audio components
- Pilot lock indicator
- SAP detector
- Separate noise detectors for stereo and SAP with adjustable threshold levels, hysteresis, and automute function

An overview of the supported standards and sound systems and their key parameters is given in the following tables.

The analogue multi-channel sound systems (A2, A2+ and A2*) are sometimes also named 2CS (2 carrier systems).

3.9.2.1 Analogue 2-carrier systems

Table 108: Frequency modulation

Standard	Sound system	Carrier frequency (MHz)	FM deviation (kHz) nom./max./over	Bandwidth/ de-emphasis (kHz/ms)	Modulation	
					SC1	SC2
M	mono	4.5	15/25/50	mono	-	15/75
M	A2+	4.5/4.724	15/25/50	12(L + R)	12(L - R)	15/75 (Korea)
B/G	A2	5.5/5.742	27/50/80	12(L + R)	R	15/50
I	mono	6.0	27/50/80	mono	-	15/50
D/K (1)	A2	6.5/6.258	27/50/80	12(L + R)	R	15/50
D/K (2)	A2*	6.5/6.742	27/50/80	12(L + R)	R	15/50
D/K (3)	A2	6.5/5.742	27/50/80	12(L + R)	R	15/50

Table 109: Identification for A2 systems

Parameter	A2/A2*	A2+ (Korea)
Pilot frequency	54.6875 kHz = 3.5 x line frequency	55.0699 kHz = 3.5 x line frequency
Stereo identification frequency	$117.5\text{ Hz} = \frac{\text{line frequency}}{133}$	$149.9\text{ Hz} = \frac{\text{line frequency}}{105}$
Dual identification frequency	$274.1\text{ Hz} = \frac{\text{line frequency}}{57}$	$276.0\text{ Hz} = \frac{\text{line frequency}}{57}$
AM modulation depth	50%	50%

3.9.2.2 2-carrier systems with NICAM

Table 110: NICAM standards

Standard	Frequency (MHz)	TYPE	Index(%)nom./max.	Deviation(kHz)nom./max./over	SC2(MHz)NICAM	De-emphasis	Roll-off(%)	NICAMcoding
B/G	5.5	FM	-	27/50/80	5.85	J17	40	[1]
I	6.0	FM	-	27/50/80	6.552	J17	100	[1]
D/K	6.5	FM	-	27/50/80	5.85	J17	40	[1]
L	6.5	AM	54/100	-	5.85	J17	40	[1]

[1] See 'EBU specification' or equivalent specification.

3.9.2.3 Satellite systems

An important specification for satellite TV reception is the Astra specification. The Sound Processor Core is suited for the reception of Astra and other satellite signals.

Table 111: FM satellite sound

Carrier type	Carrier frequency (MHz)	Modulation index	Max. FM deviation (kHz)	Modulation	Bandwidth/ de-emphasis (kHz/ms)
Main	6.50 [1]	0.26	85	mono [1]	15/50 [2]
Sub	7.02/7.20	0.15	50	m/st/d [2]	15/adaptive [3]
Sub	7.38/7.56	0.15	50	m/st/d [2]	15/adaptive [3]
Sub	7.74/7.92	0.15	50	m/st/d [2]	15/adaptive [3]
Sub	8.10/8.28	0.15	50	m/st/d [2]	15/adaptive [3]

[1] For other satellite systems, frequencies of, for example, 5.80, 6.60 or 6.65 MHz can also be received. A de-emphasis of 60 ms, or in accordance with J17, is available.

[2] m/st/d = mono or stereo or dual language sound.

[3] Adaptive de-emphasis = compatible to transmitter specification.

3.9.2.4 BTSC/SAP, Japan (EIAJ) and FM Radio Systems

Table 112: Frequency modulation

Standard	Sound system	Carrier frequency (MHz)	FM deviation (kHz) nom./max./over	Modulation	Bandwidth/ de-emphasis (kHz/ms)
M	mono	4.5	15/25/50	mono	15/75
M	BTSC	4.5	50 max	MPX (FM/AM)	14/n.a. [1]
	SAP	5fh=78,67 kHz	15 max	SAP (FM)	8/n.a. [1]
M	Japan	4.5	15/25/50	MPX (FM/FM)	15/50
FM Radio	stereo	4.5...10.7	40/75/150	MPX (FM/AM)	15/75 or 15/50

[1] Not applicable due to dbx noise reduction.

Table 113: Identification for BTSC/SAP, Japan (EIAJ) and FM Radio systems

Parameter	Pilot Tone Frequency
BTSC	1fh=15.734 kHz
Japan/(EIAJ)	3.5fh= 55,069 kHz
FM Radio	19 kHz

3.9.3 Features

3.9.3.1 Demodulator and decoder

- Demodulator and Decoder Easy Programming (DDEP).
- Auto standard detection (ASD).
- Static Standard Selection (SSS).
- DQPSK demodulation for different standards, simultaneously with 1-channel FM demodulation.
- NICAM decoding (B/G, I, D/K and L standard).
- Two-carrier multistandard FM demodulation (B/G, D/K and M standard).
- Decoding for three analogue multi-channel systems (A2, A2+ and A2*) and satellite sound.
- Adaptive de-emphasis for satellite FM.
- Optional AM demodulation for system L, simultaneously with NICAM.
- Identification A2 systems (B/G, D/K and M standard) with different identification time constants.
- FM pilot carrier present detector.
- Monitor selection for FM/AM DC values and signals, with peak and quasi peak detection option.
- BTSC MPX decoder .
- SAP decoder.
- dbx noise reduction.

- Japan (EIAJ) decoder.
- FM radio decoder.
- Soft-mute for DEMDEC outputs DEC, MONO and SAP.
- FM overmodulation adaptation option to avoid clipping and distortion.
- Sample rate conversion (SRC) for up to three demodulated terrestrial audio signals. It is possible to process SCART signals together with demodulated terrestrial signals.

3.9.3.2 Audio Multi Channel Decoder

- Dolby® Pro Logic® II Surround (DPL2); Trademark of Dolby Laboratories Licensing Corporation.
- Six channel processing for Main Left and Right, Subwoofer, Center, Surround Left and Surround Right.

3.9.3.3 Volume and tone control

- Automatic Volume Level (AVL) control
- Smooth volume control
- Master volume control and Balance
- Soft-mute
- Loudness
- Bass, Treble
- Dynamic Bass Enhancement (DBE)
- Dynamic Ultra Bass (DUBII)
- Non processed subwoofer
- 5 band equalizer
- Acoustical compensation
- Programmable beeper
- Noise generator for loudspeaker level trimming

3.9.3.4 Reflection and delay

- Dolby® Pro Logic® Delay
- Pseudo hall/matrix function

3.9.3.5 Psychoacoustic spatial algorithms, downmix and split

- Incredible Mono
- Incredible Stereo
- Virtual Dolby® Surround (VDS 522,523)
- Virtual Dolby® Digital (VDD 522,523)
- Bass Redirection according to Dolby specifications

3.9.3.6 Interfaces and switching

- Digital crossbar switch for all digital signal sources and destinations
- Output crossbar for exchange of channel processing functionality

- Digital audio input interface (stereo I^2S input interface)
- Digital audio output interface (stereo I^2S output interface)
- Audio monitor for level detection
- 8 audio DACs for six channel loudspeaker outputs and stereo headphones output
- 4 audio DACs for stereo SCART output and stereo LINE output.
- Serial data link interface for interfacing with the analogue multi-purpose interface IC PNX3000 (PNX3000).

3.9.4 Functional overview of the Sound Core

The Sound Core contains two DSP cores as shown in Figure 67. The first core called DEMDEC-DSP is combined with DEMDEC hardware and the second core is the AUDIO-DSP. The DEMDEC-DSP is used for the decoder and demodulator tasks, plus the sample rate conversion.

The AUDIO-DSP is used for the sound features, from the level adjust unit up to the output cross bar. Audio DACs and I^2S hardware convert the processed signals to analogue or digital audio.

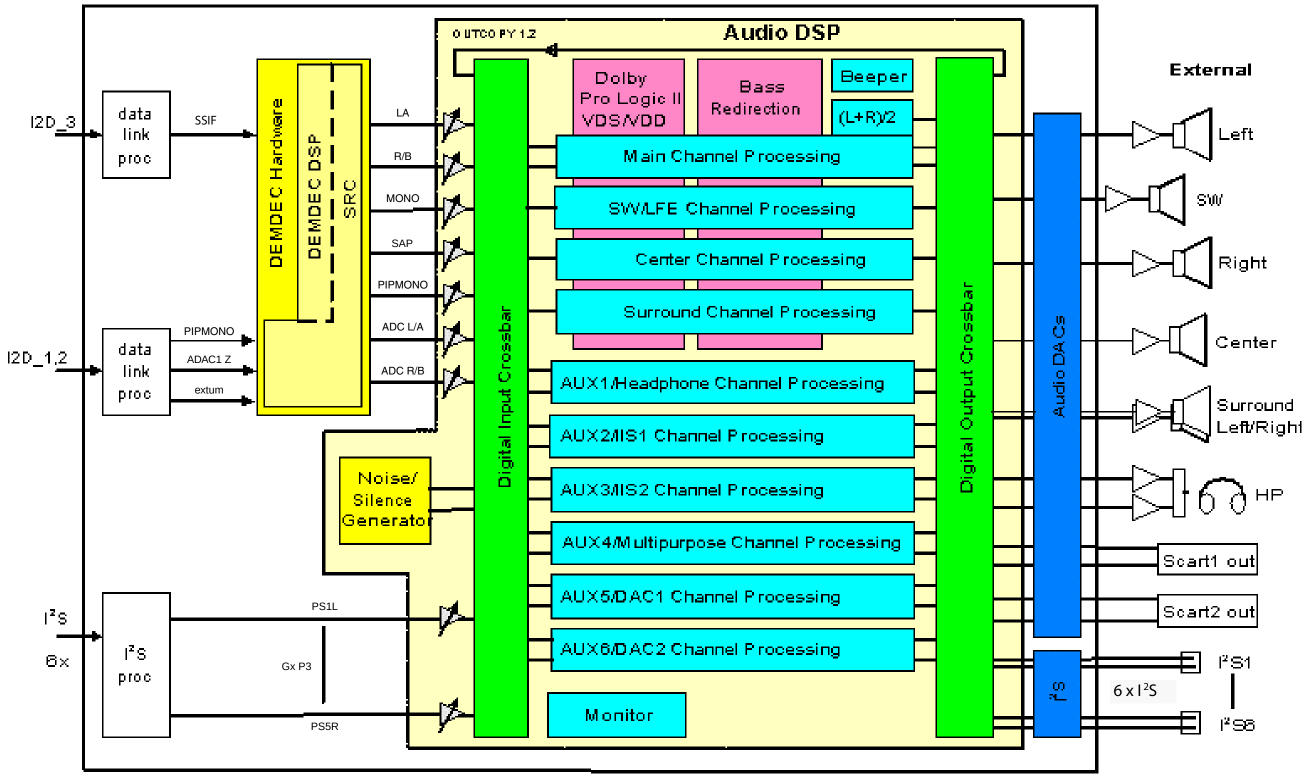


Fig 67.Sound Functions

All DLINK data links carry sound signals. The data link processing splits them from the other signals such as video so that the DEMDEC block receives the second sound IF (SSIF) and the audio signals from the audio ADCs of the PNX3000 IC.

The SSIF needs some hardware processing before it enters the DEMDEC DSP Section 3.9.8.

The audio signals from the audio ADCs of PNX3000 are passing the DEMDEC DSP only for source selection and sample rate conversion.

The Audio DSP block shows the structure of the audio processing. After level adjust all signals from the DEMDEC and the I^{2}S input are available at the Digital Input Crossbar. Special inputs are provided by the OUTCOPY 1, 2 signal, a feedback from the Digital Output Crossbar, and the Noise/Silence Generator needed for Dolby® Pro Logic® processing. Every audio channel can be connected to each of the inputs to the Digital Input Crossbar. All channel processing delivers signals to the Digital Output Crossbar which offers the facility to connect each of the channel signals to the appropriate DACs or to the I^{2}S outputs. The crossbar offers the freedom to the set maker to choose the purpose of the speaker DACs, whether they feed the L, R, C, SL, SR or SW power amplifier, according to the needs of his chassis layout.

In normal TV applications two of the DAC outputs are used to feed a headphone but they can also be dedicated to other purposes.

Two DAC stereo outputs are provided for the audio feedback to the PNX3000 IC.

Digital audio output signals are available from six I^{2}S outputs. The purpose of the outputs is decided by the customer.

Details of the audio processing is described in following sections.

3.9.5 Sound Core Control Interface

The Sound Core is controlled by writing to locations within its address range in the memory map, much like any other block in the AVIP blocks. The AVIP RSL contains the complete list and descriptions of all user-accessible registers in the Sound Core.

The registers are divided into two sections, one for each DSP unit (DEMDEC and AUDIO). From each DSP unit base address, the bottom 64 words (address offsets 0x00 - 0xFC) are the Software Control Registers. Physically these are not conventional registers, but locations in a RAM. Instead of controlling hardware directly, they are read by the software running internally on the DSP units, which alters its behavior according to the register contents. The DSPs can also write status information back to assigned memory addresses for polling by the application software.

From the user’s point of view, the only difference between the Software Control Registers and any other conventional register is the access time. The DSP software only allows control access to the DSP memories once per audio sample period, so there is a latency of up to 1/fS on accesses to the Software Control Registers (where fS is the current audio sample frequency in use on the I^{2}S interface). For this reason, the time-out value of the BCU should be programmed to a minimum of 1/fS to prevent spurious time-outs occurring during access to the Sound Core registers.

The Sound Core contains a unit called the Bus Allocation Minimizer (BAM), which is intended to prevent long wait-states on the PI caused by the Software Control Register access latency. This is not relevant in the AVIP since all control is via the (relatively low-bandwidth) I^{2}C interface, so the BAM should not be enabled in normal use.

3.9.6 I^{2}S

The Soundblock contains different digital serial audio inputs, serial digital outputs and associated clock signals. It can be used to supply, for example, audio signals from received TV programs to a digital audio output device (AES/EBU format), or import serial audio signals from other sources for reproduction through the TV set’s loudspeaker and/or head phone channels.

Three serial audio formats are supported at the feature interface:

- Philips I^{2}S format
- Sony I^{2}S format
- Japanese LSB justified format 24-bits

The differences of the formats are illustrated in Figure 68 to Figure 70.

In the Philips and Sony formats the left audio channel of a stereo sample pair is output first and is placed on the serial data line (SDI for input, SDO for output) when the word select line (WS) is LOW. Data is written at the trailing edge of SCK and read at the leading edge of SCK. The most significant bit is sent first.

In the Japanese LSB justified format the right audio channel of a stereo sample pair is output first. The most significant bit is sent first but data is LSB aligned to the falling edge of the word select line (WS).

The following is only applicable for Japanese LSB justified formats:

The input circuitry is limited in handling the number of SCK pulses per WS level. The maximum allowed number of bitclocks per WS is 64 (per mono audio word 32 bitclocks). Also the number of bitclocks during the low and high phase of WS should be equal, or more than, the selected format (24 bits).

All inputs and the output work with the same sampling frequency, formats and word sizes.

For input, six stereo channels per sampling frequency are transmitted.The number of significant bits is 24.

For output, six stereo channels (2*32 bits) per sampling frequency are transmitted.The number of significant bits on the output is 24.

The number of bitclock (SCK) pulses may vary in the application. When the applied word length is smaller than 24 bits, the LSB bits are set to “0” internally. When the applied word length exceeds 24 bits, the LSB’s are skipped.

During master mode the word select output is clocked with the audio sample frequency at 48 kHz. The serial clock output (SCK) is clocked at a frequency of 64*Fs. This means, that there are 64 clock pulses per pair of stereo output samples, or 32 clock pulses per sample. Depending again on the signal source, the number of significant bits on the serial data output SDO is 24. Apart from just feeding a digital audio device, such as a DAC or an AES/EBU transmitter, the serial data outputs can be connected directly to the serial inputs (loop-back connection) or first to an external device, e.g. a feature DSP such as the SAA7710 and then back to the serial inputs. In all of these configurations, the SCK and WS clocks are generated by the AVIP, which then is the I^{2}S master.

In slave mode, the external source is master and supplies the clock (SCK) and the word select (WS). The I^{2}S interface works on sampling frequencies of 32kHz to 48kHz.

3.9.6.1 Formats

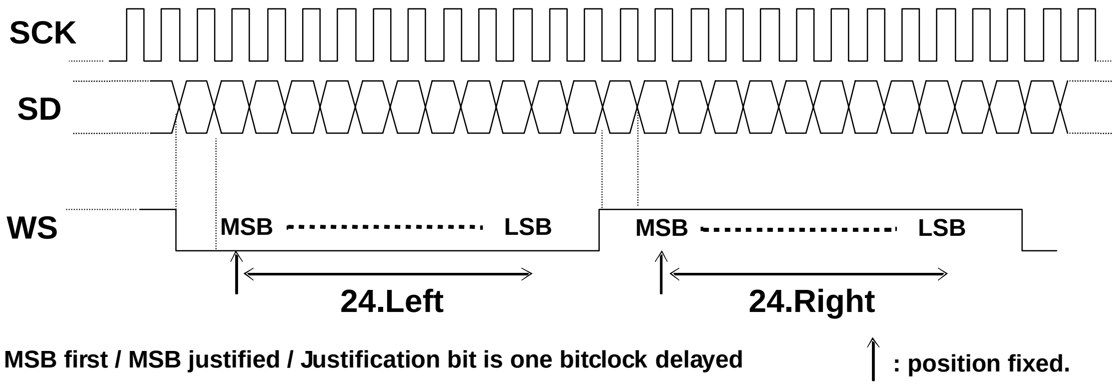

SCK
SD
WS
MSB
LSB
MSB
LSB
24.Left
24.Right
MSB first / MSB justified / Justification bit is one bitclock delayed
: position fixed.

Fig 68.Philips I^{2}S Format

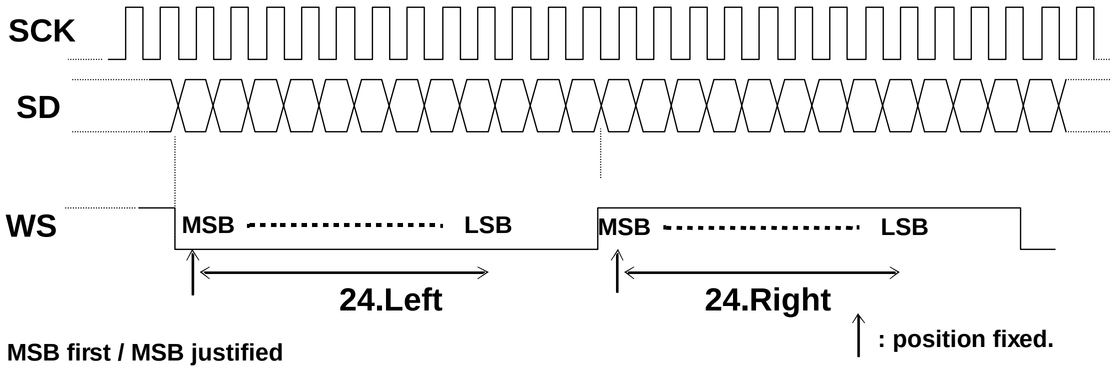


Fig 69.Sony I^{2}S Format

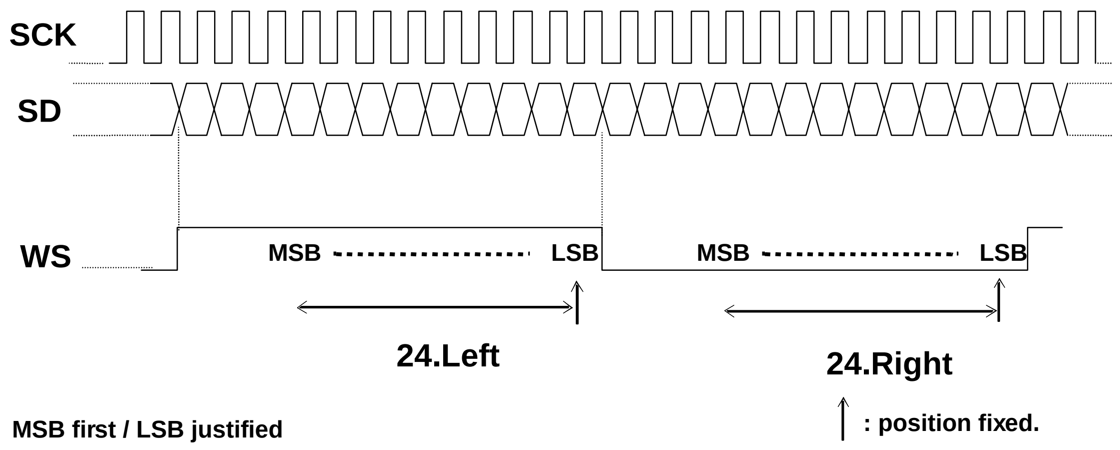


Fig 70.Japanese Format

3.9.7 Digital-Analogue Converters (AVIP1 only)

The Sound Processor Core contains twelve audio DACs. These DACs are typically used to provide six analogue loudspeaker outputs, two analogue headphone outputs and four analogue audio output connections from the digital sound processing core to an external analogue crossbar switch section.

Each of these audio DACs is based on the SDAC type meaning that it incorporates a switched resistor architecture. This special multi-stage bitstream digital-to-analogue converter requires a 128-fold oversampled input signal. The oversampled input signal is internally preprocessed by a 3rd order noise shaper to achieve a further noise reduction in the audio frequency band.

The DAC itself is capable to convert a 4 bit input signal into 16 different analog output levels. These levels are generated using 15 resistors of 15 kΩ each which are placed between an analogue switch either to supply or ground and the analogue voltage output pin. The AC output resistance of this circuit equals 1 kΩ.

Since the DAC itself does not contain any interpolating low-pass filter at its output an additional external capacitor is required. Together with the AC output resistance this capacitor represents a sufficient first order low-pass. Using a capacitor of approximately 3.3 nF results into a 3 dB roll-off at 48 kHz.

The SDAC analog voltage output has neither current source nor sink capability. Therefore an appropriate AC coupling is required in the application to connect to e.g. to a subsequent power amplifier.

3.9.8 Demdec DSP

DEMDEC Easy Programming (DDEP) is the name of the high-level control interface to the DEMDEC DSP in the SoundCore. The intention is to make the development of system control software for the DEMDEC as simple as possible, while optimally exploiting the available hardware and DSP resources.

The functionality of DDEP is divided into three main areas:

1. Demodulator and decoder configuration with optional standard and second carrier / subcarrier search;
2. Decoding, signal routing and switching for simple handling of broadcast sound signal types, plus encoding of the main status register;
3. FM overmodulation adaptation: optional adaptive reduction of levels and filter widening in case of overmodulation, in order to avoid distortions due to clipping or overflow.

The DDEP software controls both the demodulator hardware and the real-time signal processing software running on the same DSP, e.g. by changing filter coefficients, pointers etc., often depending on status information generated by hardware or software.

Most functions act like "background processes": small code sections are executed at a reduced rate (for instance every 32nd sample at 32 kHz = 1000 times per second), in order to accommodate a large amount of program code without consuming too much of the available processing power of the DSP. A "control timeslot" is reserved in the DSP software in which both control register decoding and background processing is performed.

3.9.8.1 DDEP in brief

DDEP can operate in one of two modes, which differ only in the type of standard handling. Additionally, a few options are available to the user.

In ASD (Auto Standard Detection) mode, an automatic TV sound standard and carrier search is performed at a channel switch, following preferences determined by the user or the system controller, such that a standard detection and identification (stereo / dual) result is obtained as fast as the hardware permits. If only the stereo system within a standard changes later, the search procedure adapts (e.g. B/G A2 to B/G NICAM or vice versa).

The SSS (Static Standard Selection) mode requires the user to select the sound standard (incl. stereo system) by means of a standard code (e.g. code 4 denotes "B/G A2", the European analog FM two-carrier standard) and no searching is done. This mode is like a subset of the ASD mode in that it acts similarly as the ASD mode if the standard detection has found the selected standard. However, in SSS mode the decoder never changes to a different standard, and the user should supply the settings that ASD selects by its own expertise (IDENT speed for A2 standards and line frequency for BTSC). The SSS mode can be used to enforce a certain sound standard in case ASD was unable to find a sound carrier (see Section “Channel switch procedure”) and is needed to select FM Radio decoding. The ASD routines operate as if using the SSS mode to select a certain standard.

In both of these modes, the DDEP system handles the other signal processing and settings automatically without a need for further interaction, and also allows the same options:

- It is possible not to use the default NICAM configuration for a detected or selected standard, but supply other settings via the NICAM configuration register (see Section “NICAM configuration”.
- The default thresholds and hysteresis sizes for noise-based automute and SAP detection can be overruled (see Section “SAP detection”.
- The optional overmodulation adaptation may be used in ASD as well as in SSS mode (see Section “Using the SSS mode”).

- A pre-scaling of the EXTAM signal is usually needed to obtain a correct level.
- As NICAM sound often seems softer than the FM sound, an additional level adjust for this signal path is possible.
- Levels of the DEMDEC output signals may be changed individually if a level other than the nominal -15 dBFS (with nominal modulation degrees) is desired: in the AUDIO DSP, all signal levels can be adjusted before the first digital crossbar.

DDEP can be switched off completely, allowing user access to all low-level settings. All automatisms are then disabled. This so called "DEMDEC expert mode" ("manual mode") requires detailed knowledge and understanding of the involved hardware and software . A satellite TV application unfortunately requires the expert mode since satellite sound is not supported by DDEP. As usual for this application, all configurations like carrier frequencies and de-emphasis types should be supplied by the set user or have to be pre-programmed. The expert mode is also needed for Philips internal evaluation purposes.

Figure 71 shows the management of the two different control register sets for DDEP and expert mode, and their translation into software and hardware settings.

All central DDEP functions are controlled by writing a single register, the DDEPR (see Section 3.9.8.5), located in the XRAM (data memory) of the EPICS DSP and accessible via the I^{2}C interface of AVIP.

3.9.8.2 What DDEP does NOT do

1. DDEP does not include handling the SIF frontend (input selection, AGC etc.) since this is application dependent and forms part of another chip or IP block (currently PNX3000). It is also lacking information about the SSIF input level, IF lock, video detection, or other parts of the system.
2. The DEMDEC expert mode registers are not overwritten by DDEP, so it is not possible to read out internal settings (such as the hardware configuration) from these registers. This is mainly for two reasons: control registers should not change their value by themselves (except at reset), and an encoding of the expert mode registers from the distinct DDEP variables and settings would involve too much complexity.
3. DDEP has nothing to do with controlling the audio features in the Audio DSP of the sound core.
4. It is currently not possible to exchange the standard detection routines by user-defined algorithms.

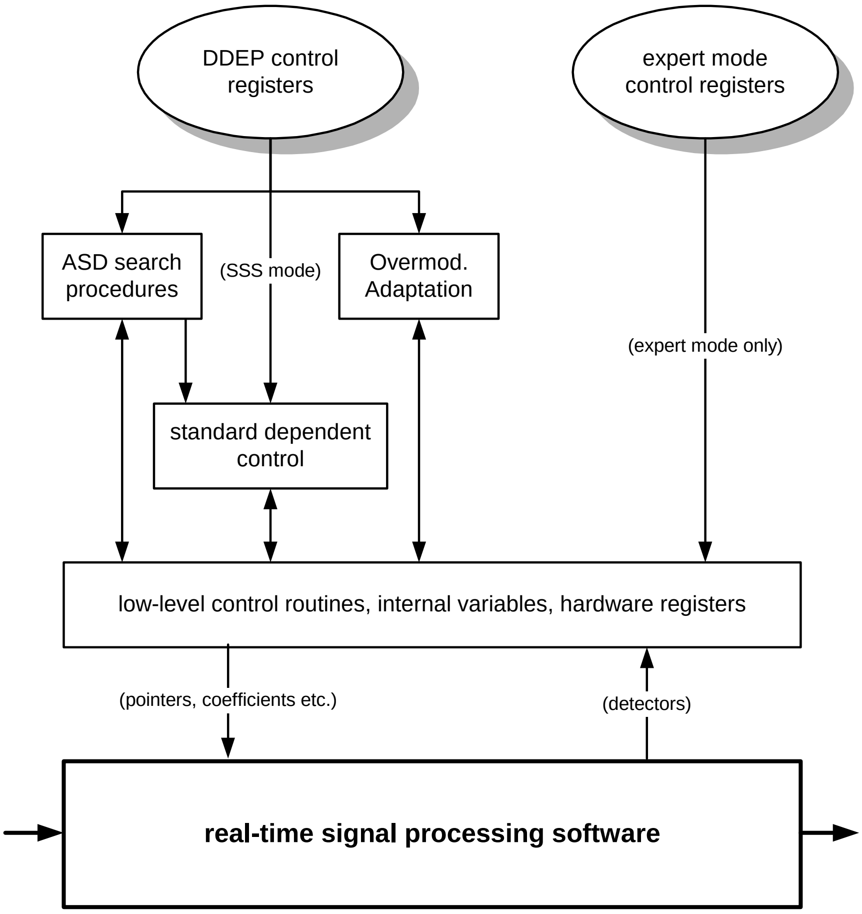

DDEP control registers
expert mode control registers
ASD search procedures
(SSS mode)
Overmod. Adaptation
(expert mode only)
standard dependent control
low-level control routines, internal variables, hardware registers
(pointers, coefficients etc.)
(detectors)
real-time signal processing software

Fig 71.Control Flow

3.9.8.3 Design considerations

The following requirements and priorities were identified for the design of the DDEP functionality.

1. Detection of sound standards and stereo/bilingual mode should be as fast as the hardware permits.
2. DEMDEC control should be possible with a minimum of transfers, preferably via a single control and one status register.
3. Except initialization and channel switching, no further controller interaction should be necessary.
4. The controller should have the option to change a number of default settings.
5. The control interface should help to minimize the risks involved in standard detection (see Section "Search procedures (ASD mode)").
6. The control (I^2C or PI bus) register map should be easy to overview, with related functions placed in one register or close to each other.
7. Registers are writable / readable with auto-increment of addresses (for access via I^2C)

8. It should be allowed to refresh registers without major restrictions.
9. All automatisms can be disabled, allowing control in the "old-fashioned" style via low-level settings (mainly for debugging, validation and internal purposes).
10. The complexity should not be too high in order to facilitate an economic implementation on a EPICS7A DSP.

3.9.8.4 DDEP basics and usage

The following paragraphs provide information that a control software developer should know to handle the DEMDEC DSP of the SoundCore in a terrestrial TV or VCR application.

DEMDEC hardware blocks and the sample rate problem: Due to the high bandwidth and computational requirements, the actual demodulation of the sound carrier(s) is implemented in dedicated digital hardware. These are:

1. Two FM / AM demodulator channels with programmable mixer frequencies and four different filter bandwidths, as present in all TDA987x(A) digital stereo decoders;
2. A NICAM demodulator and decoder for all NICAM standards, mostly identical to the TDA9874(A) and TDA9875(A), but adapted to 27 MHz system clock and using an internal timing (symbol tracking) loop instead of a DCXO;
3. An identification circuit for all standards as in TDA987x(A) but extended by the Japanese standard detectors;
4. An additional "BSJ" ("BTSC, SAP, Japan") block (PS restricted) provides MPX demodulation (for BTSC and FM Radio), FM subchannel demodulation plus matching filter (for SAP and EIAJ standard), and a noise detector. This block was not yet present in TDA987x(A) and adds "global TV sound" capability to the DEMDEC.

The dedicated demodulator and decoder hardware delivers "raw" signals that cannot be used without further processing in the DEMDEC DSP. Furthermore, each signal type comes in with a different sample rate: the analog sound carrier channels produce samples at n*35.2 kHz (derived from the 27 MHz clock by dividers 3Þ2n); NICAM audio is sampled at 32 kHz (synchronized to the transmitter), and the ADCs deliver signals with 52.7 kHz sample rate (rounded values). In order to achieve a common sample rate for all these signals, a flexible or asynchronous sample rate conversion (SRC) is necessary, as explained below.

The AUDIO DSP runs at a sample rate determined by the operating mode of the sound core:

- in I^{2}S master mode (the default), a fixed sample rate of 48 kHz is derived from the system clock (32 and 44.1 kHz are also possible).
- in I^{2}S slave mode, the WS (word select) signal of the external device is the clock master. The SoundCore supports 32, 44.1 and 48 kHz sample rates in I^{2}S slave mode.

Figure 72 shows the described hardware blocks and their interconnection to the DSP. The hardware is usually controlled by the DDEP software, or via the expert mode.

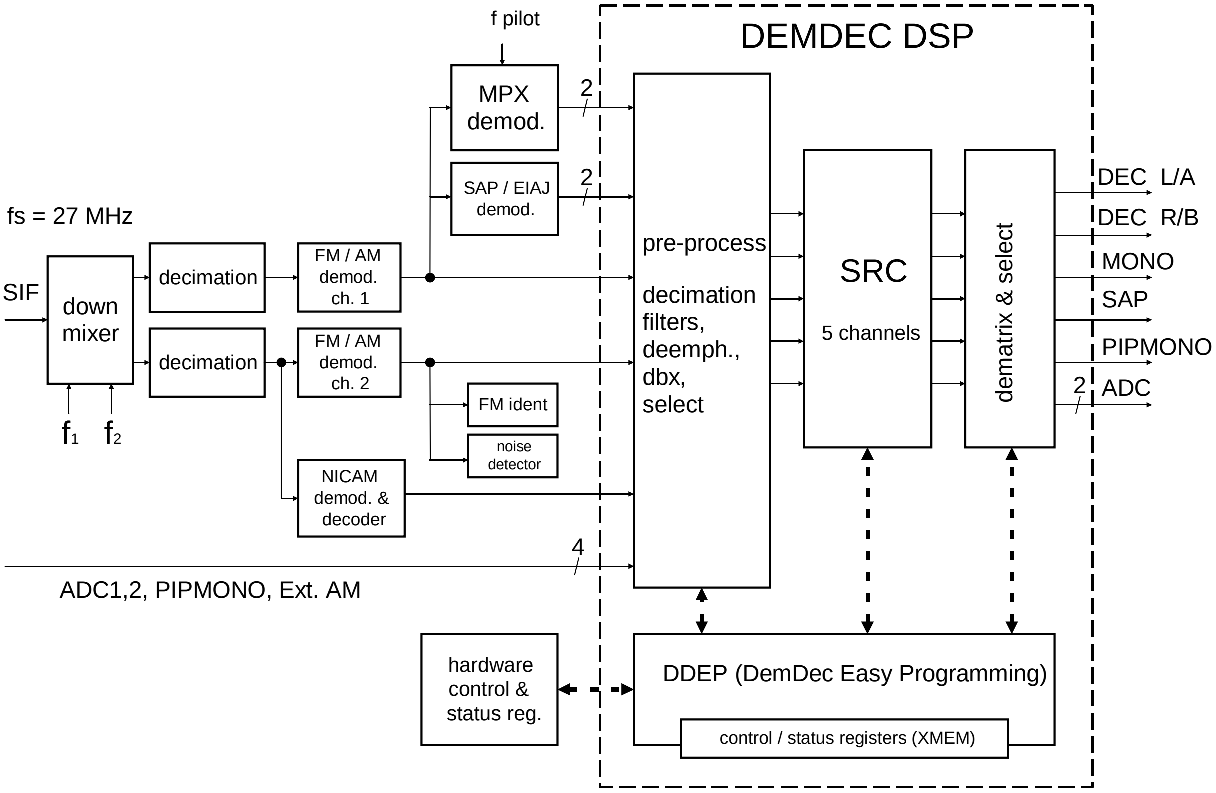


Fig 72.DEMDEC structure

Signal processing in DSP software: The output signals of the above-mentioned hardware blocks, plus four audio ADC channels are read in by the DSP, processed, converted to the current audio sample rate, "demultiplexed", and forwarded to the Audio DSP for further processing (volume, tone control, effects etc.). Figure 73 shows this signal flow in a simplified structure.

The DSP applies several filters, like down-sampling and de-emphasis, noise reduction processing (Wegener-Panda / dbx expanders), performs a sample rate conversion (SRC) to the current audio sample rate, and routs the decoded signals to the output channels. The first (topmost in pair of output channels, called DEC (from DECoder), is intended to carry the stereo or bilingual (dual) signal; an extra "MONO" channel always contains the mono signal from the first sound carrier (always FM or AM), or the main channel ([L+R]/2) of the MPX type standards BTSC, FM Radio and EIAJ. (This channel may contain different audio contents in case of NICAM.) Another single channel named "SAP" transports a SAP signal if detected during a BTSC reception. PIPMONO is the monaural sound belonging to the "picture-in-picture" feature and may come from a second tuner or some other source. The two "ADC" channels are originating from some other external device, usually the via the SCART stereo input connector. The “PIPMONO” and “ADC” signals are not further processed.

The "external AM" signal (EXTAM in Figure 73 and Figure 74) should be used for standard L/L' as the "internal" (related to the SoundCore) digital AM demodulator does usually not achieve the S/N performance of an analog demodulator operating on the first sound IF.

The EXTAM signal may come from the IF frontend's (PNX3000 or successor) internal AM demodulator or another external device. It is routed by DDEP to the MONO output of the DEMDEC (and to the DEC as well if no NICAM is detected), see also Table 115.

By means of this signal routing, the processing paths in the audio backend do not need to select a specific source depending on the currently active sound standard as it was required in earlier Philips stereo decoders (FM/AM, NICAM source). For every audio processing path, the controller can select the DEC, MONO etc. output like any other signal source (ADC, I^{2}S input,…). The information about the signal type (mono, stereo, dual) on the DEC channels is available by two status bits (see Section 3.9.8.7 “Status registers”). This also allows the audio backend to implement a "smart matrix" which selects one of the two languages in dual mode, or stereo in other cases.

The MONO output can be selected in case that stereo / dual is not wanted, while a two-channel output to another destination is still possible. A special case is a NICAM transmission with independent contents of analog and NICAM sound carriers (indicated by status flag RSSF=0) when the mono channel carries a different signal than the NICAM channels.

The switching structure of the output processing block, Figure 74, matches the requirements of the various standards and allows simple handling of auto-mute issues etc.

Internal scalings are applied in DDEP mode (not all shown in the figures due to limited space) such that all outputs signals have a level of -15 dBFS for nominal modulation degrees (e.g. 54% full scale sine wave = 27 kHz FM deviation of a B/G FM carrier). Additional level adjustments can be performed at the digital crossbar in the audio DSP. In expert mode the internal scalings (switches etc.) should be controlled via the expert mode registers.

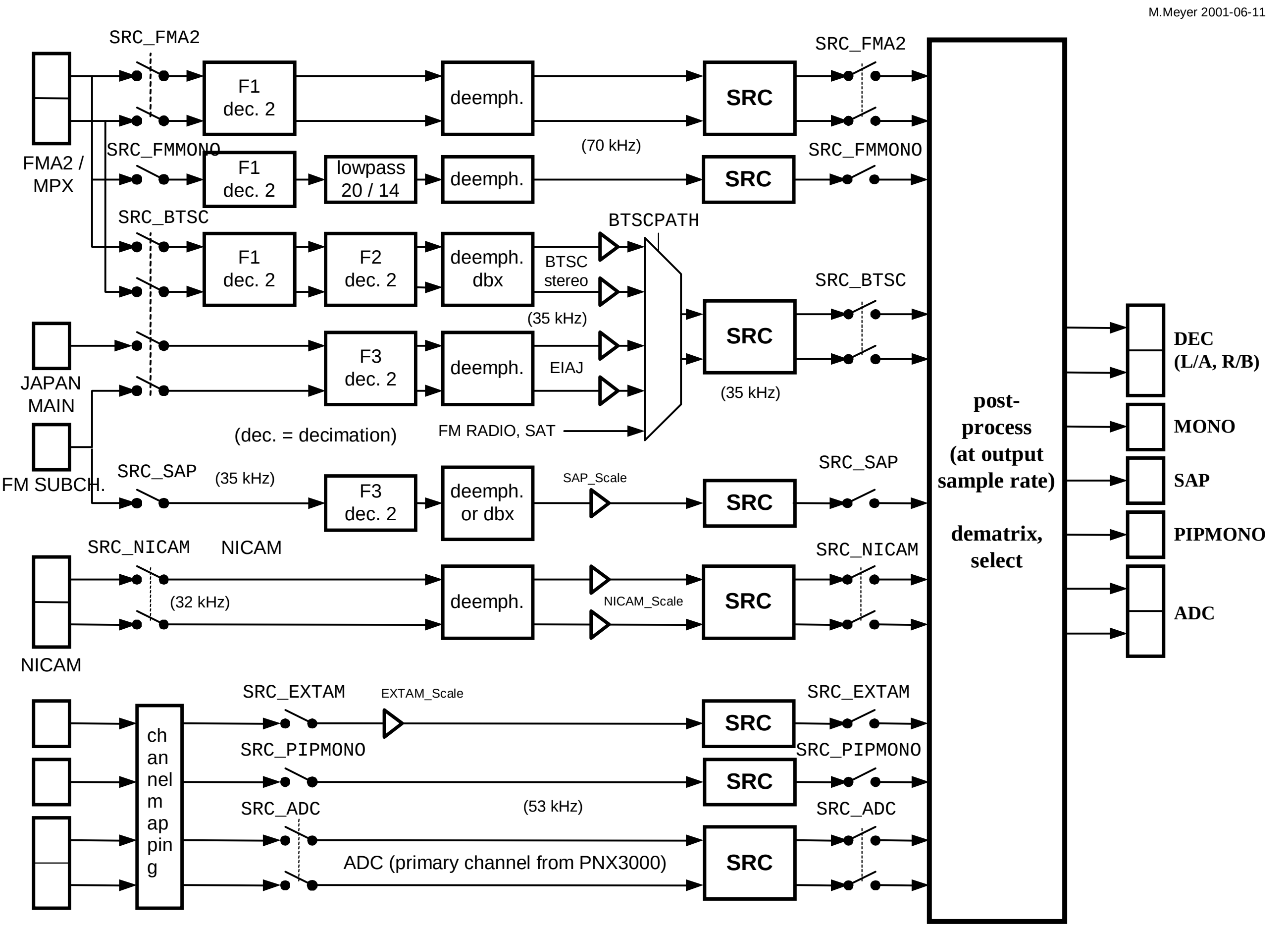


Fig 73.Signal processing modules and SRC in DEMDEC DSP (simplified)

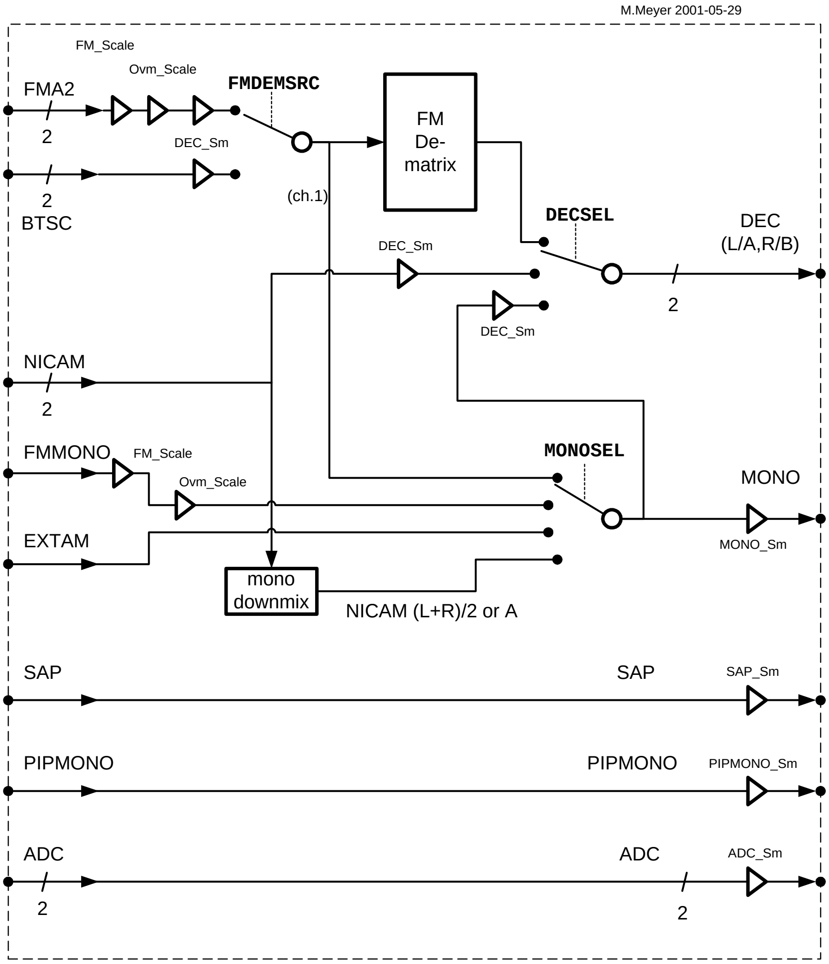


Fig 74.Switching and matrixing in post-processing block

SRC constraints: A high-quality flexible sample rate conversion, i.e. a re-sampling with a fractional and varying frequency ratio, is very costly in terms of computing power due to a high amount of filtering needs. The DEMDEC DSP, if running at the currently specified clock of 140 MHz (= 140 MIPS), has a capacity to process five SRC channels plus the required TV sound-related processing (pre-/postprocessing) and administration overhead.

The TV sound demodulators / decoders and audio ADCs can produce up to six independent channels: two channels for stereo or bilingual TV sound, a third language (SAP in the BTSC standard, or the analog sound carrier in NICAM standards), PIPMONO, and a two-channel ADC. These outputs are indicated in Figure 72 and further explained in this section. However, only up to five channels can be processed by the SRC due to the limited DSP capacity.

Therefore, it is necessary that the controller selects which channels are to be converted by SRC, depending on which sources for the audio backend are desired by the set's user. How this is accomplished is explained below.

This limitation to five channels is not an actual restriction because it can be assumed that not more than three independent audio output facilities are present in the sound subsystem, as it is usually the case in a TV set (speakers, headphone, and a SCART output).

The system controller has to select a certain “preferred” SRC channel configuration by the 2-bit variable SRCPREF (SRC PREFerence) in the DDEP control register, see Section 3.9.8.5 “The DDEP control register”. Although the SRC constraints are not much related to the actual DEMDEC functionality, this variable has been placed in the DDEP control register in order to maintain the advantage of a single-register control facility.

The generalized meanings of the values the SRCPREF variable may assume are listed in the table below. It is meaningful mostly in sound standards where an independent third channel may be present, these are the BTSC and NICAM standards. For the FM A2 and EIAJ standards, no restrictions apply, but SRCPREF can be used consistently to suppress stereo decoding like in the other cases, by choosing SRCPREF:= 3. Likewise, if no stereo / dual mode is detected or auto-mute is active, a configuration equivalent to SRCPREF = 3 is chosen automatically (i.e. other settings are internally overruled), such that all the other signals can be converted. Table 115 gives a detailed overview of all these possible SRC configurations.

The controller will rarely choose configurations with SRCPREF = 3, but it is required for example if NICAM stereo is received, and the analog sound carrier, PIPMONO, and ADC sources are selected in the audio backend. It can also serve to disable stereo decoding (“forced mono”) in case of unstable reception conditions to avoid flickering.

This case the GST and GDU bits (which always reflect the signal type on the DEC output) will also be zero, such that the actual reception conditions have to be determined from the other status bits.

Table 114: Output signal restriction depending on SRCPREF

SRCPREF(dec.)	UNAVAILABLE DEMDEC output signal, if standard is BTSC or NICAM
0	PIPMONO
1	third language (SAP or analog carrier @NICAM)
2	ADC
3	DEC stereo / dual (output forced to mono, i.e. DEC has same contents as MONO also for FM A2 and EIAJ)

The SRCPREF variable may be changed at any time and is always active in DDEP mode.

To avoid pop or click noises, the SRC channels are switched with automatic internal softmute (raised cosine ramp of 32 ms length), that means a short delay may occur before the change becomes effective.

Table 115: Active output signals depending on standard and SRCPREF selection

Standard group	Ref. No.	variable SRCPREF (dec.)	Output (active yes/no, contains …)				
			DEC ("Decoded")	MONO	PIPMONO	SAP	ADC
FMA2	1	0, 1, 2	M / St / Du	M	yes	-	yes
	1m	3	M	M	yes	-	yes
EIAJ, FM Radio [4]	2	0, 1, 2	M / St / Du	main ch.	yes	-	yes
	2m	3	M	M	yes	-	yes
BTSC	3	0	M / St	main ch.	NO	yes	yes
	4	1			yes	NO	yes
	5	2			yes	yes	NO
	6	3 / X [1]	FMMONO	FMMONO	yes	yes	yes
NICAM B/G,I,D/K	7	0	NICAM (St / Du / M)	FMMONO	NO	-	yes
	8	1 [2]		NICAM to MONO	yes	-	yes
	9	2		FMMONO	yes	-	NO
	10	3 / X [3]	M	M	yes	-	yes
NICAM L	11	0	NICAM (St / Du / M)	EXTAM	NO	-	yes
	12	1 [2]		NICAM to MONO	yes	-	yes
	13	2		EXTAM	yes	-	NO
	14	3 / X [3]	EXTAM	EXTAM	yes	-	yes

[1] Configuration 6 is chosen automatically if no stereo pilot is detected, or if the dbx decompressor is used for the SAP channel (bit SAPDBX = 1, see Table 118). SRCPREF is not effective in this case.

[2] In configurations 8 and 12, the MONO output is derived from the NICAM signal (instead of the analog sound carrier) for compatibility reasons, i.e. mix stereo to mono or use language A, respectively.

[3] Configurations 10 or 14 are chosen automatically in case of NICAM automute.

[4] Configuration 2/2m is chosen automatically for FM Radio.

Table 116 shows which SRC channels are active per configuration, and the number of active channels.

According to current data, the worst case with regard to the cycle budget is BTSC stereo with SAP.

Table 116: Active SRC Channels per Configuration

Std. Group	Ref. No.	Ctrl. Bits SRCPREF (dec)	Active SRC channels and preprocessing routines								Active Channels
			FMA2	FMMONO	BTSC	NICAM	SAP	EXTAM	PIPMONO	ADC	
FM A2	1	0,1,2	2						1	2	5
	1m	3	2						1	2	5

Table 116: Active SRC Channels per Configuration *…continued*

Std. Group	Ref. No.	Ctrl. Bits SRCPREF (dec)	Active SRC channels and preprocessing routines								Active Channels
EIAJ, FM Radio	2	0,1,2			2				1	2	5
	2m	3			2				1	2	5
BTSC	3	0			2		1			2	5
	4	1			2				1	2	5
	5	2			2		1		1		4
	6	3/X		1			1		1	2	5
NICAM	7	0		1		2				2	5
B/G,I,D/K	8	1				2			1	2	5
	9	2		1		2			1		4
	10	3/X	2						1	2	5
NICAM L	11	0				2		1		2	5
	12	1				2			1	2	5
	13	2				2		1	1		4
	14	3/X						1	1	2	4

Table 117 shows how the SRCPREF setting can be derived from the desired channels, i.e. the channels that the Audio DSP wants to use. (3 out of 5 = 10 combinations).

Table 117: SRCPREF selection depending on desired sources

No	DEC	MONO	SAP	PIPMONO	ADC	SRCPREF
1	X	X	X			0
2	X	X		X		2
3	X	X			X	0
4	X		X	X		2
5	X		X		X	0
6	X			X	X	1
7		X	X	X		2
8		X	X		X	0
9		X		X	X	3
10			X	X	X	3

3.9.8.5 The DDEP control register

All control and status registers of the DEMDEC DSP are listed inTable 118 . This is also the source for generating the URT file for semiautomatic generation of the "QuickView" software and control / demonstration software development. A brief overview of all registers is given the DDEP control register (short DDEPR).

The DDMUTE bit mutes all DEMDEC outputs (DEC, MONO, SAP) with a 32 ms raised-cosine ramp, such that no extra action is required for the audio backend. It is recommended to be used in Section "Channel switch procedure". For the PIPMONO and ADC paths, separate mute bits are provided in the DEM_ADC_SEL_REG register. The SAP detection bits are independent from the selected SRC configuration, i.e. SAPDET can be 1 while the SAP SRC path is not currently active. All mute bits do not affect the SRC configuration, see Section "SRC constraints". Most important are the EPMODE (Easy Programming mode), STDSEL (standard selection and ASD control), REST (restart), and SRCPREF (SRC preference).

Mode selection: As explained previously, there are two basic operating modes, ASD and SSS, plus the possibility to disable DDEP entirely (means "expert mode" DDXM). This is determined by the EPMODE variable: 0 chooses ASD mode, 1 means SSS mode, and 3 enables the expert mode and its separate register set (not in the scope of this document).

Due to space restrictions, the STDSEL variable (five bit wide) is used for different purposes depending on the operating mode.

With ASD, each bit represents one of the five standard groups: if it is set to 1, the following carrier search will perform an amplitude detection at the corresponding carrier frequency, otherwise it will not detect a standard of this group. This approach helps to minimize possible ambiguities and to reduce the risk of wrong standard detection (see Section "Search procedures (ASD mode)").

The DDMUTE bit mutes all DEMDEC outputs (DEC, MONO, SAP) with a 32 ms raised-cosine ramp, such that no extra action is required for the audio backend. It is recommended to be used in the channel switch procedure (see Section 7.8.13.4). For the PIPMONO and ADC paths, separate mute bits are provided in the DEM_ADC_SEL_REG register. The SAP detection bits are independent from the selected SRC configuration, i.e. SAPDET can be 1 while the SAP SRC path is not currently active. All mute bits do not affect the SRC configuration.

Table 119 shows the relation between STDSEL and standard groups. Between one and four bits may be set to 1. The result of a standard search can be found in Section "DEMDEC status register".

In SSS mode, STDSEL is considered an integer number and represents a code for a certain stereo standard code listed in The DDMUTE bit mutes all DEMDEC outputs (DEC, MONO, SAP) with a 32 ms raised-cosine ramp, such that no extra action is required for the audio backend. It is recommended to be used in the channel switch procedure (see Section 7.8.13.4). For the PIPMONO and ADC paths, separate mute bits are provided in the DEM_ADC_SEL_REG register. The SAP detection bits are independent from the selected SRC configuration, i.e. SAPDET can be 1 while the SAP SRC path is not currently active. All mute bits do not affect the SRC configuration Table 120. For example, if STDSEL equals 5, "B/G NICAM" is selected, without any searching

Note that a correct signal handling with FM dematrix control and automute is always applied due to the switching task, but only if the selected and the received standards match, a stereo or bilingual output can be reproduced. For above example (select "B/G NICAM"), if B/G A2 is received, only the first sound carrier is reproduced as mono.

Starting and restarting: Many settings in the DDEPR become effective only at a "restart" condition, triggered via the REST bit. This is needed to start the standard detection process at a defined moment, and also to reset a number of hardware and software states in order to adapt as quickly as possible to a different TV channel. A restart should be performed at every channel switch, or for initialization after power up. The REST bit is the only "edge sensitive" control variable in the DEMDEC part, i.e. a change from 0 to 1 triggers the restart.[2]

A change from 1 to 0 may be done any time later and has no effect.

DDEP control variables: The DDMUTE bit mutes all DEMDEC outputs (DEC, MONO, SAP) with a 32 ms raised-cosine ramp, such that no extra action is required for the audio backend. It is recommended to be used in the channel switching (see Section "Channel switch procedure"). For the PIPMONO and ADC paths, separate mute bits are provided in the DEM_ADC_SEL_REG register. The SAP detection bits are independent from the selected SRC configuration, i.e. SAPDET can be 1 while the SAP SRC path is not currently active. All mute bits do not affect the SRC configuration. (SeeTable 127).

The DDEPR alone is already sufficient for a typical terrestrial TV sound application. Additional registers (explained later) allow changes of the standard behavior or adjustments of several important parameters like detection thresholds which by default have values considered "reasonable."

Most variables in the DDEPR become effective at every write access, independent from a restart condition: EPMODE, OVMADAPT, DDMUTE, FILTBW (only if OVMADAPT=0), IDMOD (only in SSS mode), OVMTHR (only if OVMADAPT=1), SAPDBX, SRCPREF.

In contrast, the STDSEL and FHPAL variables are evaluated only at a restart, i.e. in the same write access when REST changes from 0 to 1.

The DDEPR should always be written and refreshed, also in expert mode, where all other bits except EPMODE are not active. For more information, see Section 3.9.8.12 "Other details".

When using SSS mode for FM A2 or EIAJ standards, the IDENT speed selection via IDMOD should be "fast" mode until stereo or dual is detected, and then switched to "slow" mode (except for EIAJ where the "medium" setting is recommended; see Section "Using the SSS mode"). This is done automatically in ASD mode.

The DDMUTE bit mutes all DEMDEC outputs (DEC, MONO, SAP) with a 32 ms raised-cosine ramp, such that no extra action is required for the audio backend. It is recommended to be used in Section "Channel switch procedure". For the PIPMONO and ADC paths, separate mute bits are provided in the DEM_ADC_SEL_REG register. The SAP detection bits are independent from the selected SRC configuration, i.e. SAPDET can be 1 while the SAP SRC path is not currently active. All mute bits do not affect the SRC configuration.

[2] The REST bit was introduced because with the original EPICS7A I^{2}C interface (the MPI unit) there was no means of finding which I^{2}C register (= XRAM address) was accessed, i.e. written or read. The address indication hardware ("JTA" register) was added later and allowed a restart with a simple write to the DDEPR, but the REST bit was kept in order to allow changing of other parts of the register, and for repeated refreshing which some control architectures may require.

Table 118: Contents of DDEP Control Register (DDEPR)2 (DD22)

VARIABLE NAME	# of Bits	BIT INDEX	DETAILLED DESCRIPTION
EPMODE	2	[1..0]	DEMDEC Easy Programming (DDEP) mode $0 = 'AUTO STANDARD DET.' (ASD), STDSEL[4:0] defines the set of detectable standards. $1 = 'STATIC STANDARD SELECT' (SSS). STDSEL[4:0] contains standard code. $2 = reserved $3 = DEMDEC expert mode (fully manual mode, DDEP disabled)
STDSEL	5	[6..2]	Bits multiplexed for ASD or SSS. In ASD mode (EPMODE=0): flags for allowed standards B/G | D/K | L/L' | I | M (LSB to MSB). In SSS mode (EPMODE=1): standard code as defined in status register STDRES, e.g. code 4 selects B/G A2.
REST	1	[7]	RESTART decoder and initialize Easy Programming after channel switch if changed from 0 to 1 (see Section "Starting and restarting")
OVMADAPT	1	[8]	FM over modulation adaptation (avoids distortion, filter bandwidth and gain is chosen adaptively) $0 = disabled $1 = enabled (recommended)
DDMUTE	1	[9]	mute DEMDEC output signals DEC, MONO, and SAP (softmute) $0 = no mute $1 = mute
FILTBW	2	[11..10]	sound carrier filter bandwidth (like FILTBW_M). NOT effective if BTSC, EIAJ, FMRADIO active, or if OVMADAPT=1. $0 = narrow (recommended) $1 = extra wide $2 = medium $3 = wide
IDMOD	2	[13..12]	FM ident speed in SSS mode (otherwise not effective) $0 = slow $1 = medium $2 = fast $3 = off (reset)
RSD_Bit14	1	[14]	reserved, should be written as 0
RSD_Bit15	1	[15]	reserved, should be written as 0
SAPDBX	1	[16]	SAP decompression mode $0 = dbx used for BTSC stereo decoding, fixed compromise de-emphasis for SAP (recommended) $1 = dbx used for SAP, BTSC stereo forced to mono
FHPAL	1	[17]	line frequency for BTSC decoding (see Section "Dependencies between variables in the DDEPR") $0 = NTSC line frequency (15.734 kHz) used in SSS, or preferred in ASD mode $1 = PAL line frequency (15.625 kHz) used in SSS, or preferred in ASD mode

Table 118: Contents of DDEP Control Register (DDEPR)2 (DD22) *…continued*

VARIABLE NAME	# of Bits	BIT INDEX	DETAILLED DESCRIPTION
OVMTHR	2	[19..18]	over modulation level threshold relative to nominal (applies if OVMADAPT=1) $0 = +3 dB = -12 dBFS $1 = +6 dB = - 9 dBFS (recommended) $2 = +9 dB = -6 dBFS $3 = +12 dB= -3 dBFS
BGA2PREF		[20]	BG A2 preference selection 0: V0; start with NICAM search 1: V1; start with A2 search, NICAM found only after timeout
RSD_Bit21		[21]	reserved, should be written as 0
SRCPREF	2	[23..22]	select channels to convert via SRC (see Section “SRC constraints”) $0 = DECoder, 3. language and ADC (not PIPMONO) $1 = DECoder, PIPMONO and ADC (not 3. language) $2 = DECoder. PIPMONO and 3. language (not ADC) $3 = MONO, PIPMONO, 3. language and ADC (DEC only MONO)

Table 119: Standard detection control bits in ASD mode

STDSEL Bit	if set, include standard group in search:	Carrier test frequency [MHz]
STDSEL[0]	B/G/H (2 possible stereo standards)	5.5
STDSEL[1]	D/K/K' (4 possible stereo standards)	6.5 (FM)
STDSEL[2]	L/L'	6.5 (AM)
STDSEL[3]	I	6.0
STDSEL[4]	M (Korea A2, BTSC, or EIAJ)	4.5

Table 120: Static standard selection codes in SSS mode

STDSEL (decimal)	Description		
	Standard	Ch. 1 freq. MHz	Ch. 2 freq. MHz
0…3	Reserved		
4	B/G A2	5.5	5.742
5	B/G NICAM	5.5	5.850
6	D/K A2 (1)	6.5	6.258
7	D/K A2 (2)	6.5	6.742
8	D/K A2 (3)	6.5	5.742
9	D/K NICAM	6.5	5.850
10	L NICAM	6.5	5.850
11	I NICAM	6.0	6.552
12	M Korea	4.5	4.724
13	M BTSC	4.5	- (MPX demod.)
14	M EIAJ	4.5	- (MPX demod.)
15	FM radio, Europe (50 µs de-emphasis)	10.7	- (MPX demod.)

Table 120: Static standard selection codes in SSS mode *…continued*

STDSEL (decimal)	Description		
16	FM radio, USA (75 µs de-emphasis)	10.7	- (MPX demod.)
17	FM radio, Europe (50 µs de-emphasis)	selected via CARRIER1	- (MPX demod.)
18	FM radio, USA (75 µs de-emphasis)	selected via CARRIER1	- (MPX demod.)
19…30	Reserved for future extensions		

Dependencies between variables in the DDEPR: IDMOD: only effective in SSS mode, in ASD mode overruled internally (fast until detection, then slow mode). Note: for the expert mode, the variable IDMOD_M, located in the hardware configuration register (DEM_HWCF_REG) should be used.

FILTBW: overruled internally if OVMADAPT = 1 or if an MPX-type standard is active ("wide" needed for BTSC, FMRADIO and EIAJ standards). Note: for the expert mode, the variable FILTBW_M, located in the hardware configuration register (DEM_HWCF_REG) should be used.

OVMTHR: only relevant if OVMADAPT = 1.

FHPAL: determines lines frequency for BTSC in SSS mode. It must match the currently received pilot frequency, or the PLL cannot lock to the pilot and stereo indication is not possible. In ASD mode, it selects the frequency to be tested first during M standard detection, therefore it should be set to 1 in Argentina (or other areas with PAL + BTSC sound) for fastest detection.

SAPDBX: enforces SRC configuration as if SRCPREF = 3 (forced mono) if standard is BTSC and reserves the dbx decoder for the SAP channel and not for the stereo sub channel. Recommended: set to 0, only set to 1 if SAP is selected for output (i.e. by an audio backend path) but the DEC output is not.

Automute function: In this context, automute refers to an automatic mute of the stereo or bilingual if the signal reception is below a certain (adjustable) level of quality, although a stereo or bilingual identification is still present. The term “automute” is still used although a more precise description would be “auto-fallback to mono”.

The automute function is active in both the ASD and SSS modes. It should be distinguished from the case that the DEMDEC outputs are muted after an ASD search (first carrier search as explained in Section “Search procedures (ASD mode)”) has failed due to poor reception conditions. An automute function is available for all sound standards, means FM A2, NICAM, and multiplex standards (BTSC, EIAJ, FM Radio).

The original FM sound carrier of all sound standards (except L) is rather robust against noise and is still well audible if the picture is hardly visible, i.e. at tuner input levels below 20 dBµV. In contrast, the additional stereo information is more sensitive due to the technical parameters (second carrier with lower level, or subcarrier above audio baseband) and requires better signal quality (higher C/N). It has always been common practise for NICAM standards to mute the NICAM sound above a certain bit error rate threshold to avoid an unpleasant crackling sound. For A2 standards, the situation is similar: since the second sound carrier (SC2) is nominally 7 dB lower than the first, the C/N threshold at which the crackling (caused by signal phase wraparounds due to noise) starts is reached at a ~7dB higher C/N than for the first carrier. However the FM identification block of Philips Semiconductors is very sensitive and detects the stereo or dual information even at C/N levels several dB below the crackling threshold. To avoid this crackling, a noise detector (corresponding to the bit error rate estimation from the NICAM decoder) is used to auto-mute the SC2 signal if a certain noise level is exceeded. For the multiplex standards BTSC (and FM Radio) and EIAJ, automute works accordingly. (Although here the FM crackling threshold is the same for the baseband signal and the sub carrier, it makes sense to disable the sub carrier above a certain noise level.) For each standard type, a separate noise threshold detector is used internally.

In case of an A2, BTSC, or EIAJ stereo reception, muting the (sub) carrier with the stereo information already causes a mono reproduction. For bilingual or NICAM reception, automute also means a replacement of the muted signal by the monaural analog sound carrier.

If the automute function has suppressed the stereo / dual signal, the DEC output is switched to the same source as the MONO output. This may coincide with a change of the SRC channel configuration as shown in Table 115.

DDEP by itself never mutes the mono sound no matter how bad the signal may be. It is however common practise to mute the decoder outputs if no video signal (sync) is detected, as recommended in Section “Channel switch procedure”.

In case of a switch from stereo to mono, it is advisable to adjust audio features in a subsequent audio backend processing path (e.g. change from I-Stereo to I-Mono, or switch off DPL etc.).

Similarly, the separate SAP output is silent when no SAP carrier is detected, as described below.

NICAM configuration: If the NICAM configuration register is not changed, following default settings are active:

- ONLY_RELATED=0 (see below)
- J17 de-emphasis on (NIC_DEEM=0)
- automute enabled (NIC_AMUTE=0)
- bit error thresholds NICLOERRLIM=100 (dec.)
- NICUPERRLIM=200 (dec.).

In expert mode, only NDEEM is active.

The EXTAM bit is set to 1 if the AM demodulation for standard L/L’ can be provided by the IF stage. This bit is active with standard L and only evaluated at a restart.

It may not be required to have automatic switching from FM/AM to NICAM if contents are different (conveyed by the reserve sound switching flag RSSF being 0). For this reason, the ONLY_RELATED control variable has been introduced. While ONLY_RELATED is 1, NICAM is only reproduced on the DEC output in case of “related” sound (RSSF=1), otherwise the analog sound carrier. In case of “unrelated” NICAM, and if the user knowingly selects NICAM (via remote control or the like), this bit is set to 0 so that the NICAM output is allowed.

In total, the conditions required for a NICAM output are the following: (VDSP =1) AND (SRCPREF <>3) AND (RSSF=1 OR ONLY_RELATED=0) AND (bit error threshold detect=false).

The NICAM automute (auto fallback to mono) function uses a threshold detector for the bit error count ERR_OUT, the thresholds of which are selectable by NICLOERRLIM (lower error limit) and NICLOERRLIM (upper error limit) . Automute can be disabled by NIC_AMUTE=1 (not recommended). The bit errors are by default counted by the hardware NICAM decoder over 128 ms. However if NICERRDETLO is set to 1, the bit errors are counted by the DSP code in a selectable interval and used as the lower threshold, instead of the hardware counter value ERR_OUT.

This modified automute function is provided to minimize NICAM on/off transitions during marginal or unstable reception. The length of the counting interval is selected by NICAM_ERRDETECTTIME in the DDEP_OPTIONS1_REG register, see Section 3.9.8.6. This means, NICAM is only un-muted if the number of bit errors in the expired variable interval is less than or equal NICLOERRLIM. To prevent increased detection times after a channel switch, NICAM_ERRDETECTTIME is set initially to a small value, and after NICAM detection the desired setting for a longer interval is applied (for example 1 second). The timer and counter for the variable interval are reset to 0 if the upper threshold is exceeded or if sync state is lost.

Table 121: NICAM configuration register (NICAM_CFG_REG, DD21)

Variable Name	No. of Bits	Bit Index	Description
ONLY_RELATED	01	[0]	reproduce only related NICAM on DEC output (DDEP only) $0 = false (NICAM whenever possible) $1 = true (NICAM suppressed if RSSF=0)
RSD_Bit1		[1]	reserved, written as 0
EXTAM	01	[2]	fall back source in case of automute in standard L (DDEP only) $0 = channel 1 output (AM) $1 = ADC output (external AM demodulator)"
NICDEEM	01	[3]	NICAM de-emphasis (J17) $0 = ON $1 = OFF (only for test purposes)
NIC_AMUTE	01	[4]	NICAM auto mute function depending on bit error rate (DDEP only) $0 = ON (recommended) $1 = OFF
NICLOERRLIM	08	[12..5]	NICAM lower error limit for automute (DDEP only)
NICUPERRLIM	08	[20..13]	NICAM upper error limit for automute (DDEP only)
NICERRDELTO	0	[21]	select NICAM bit error detector type for lower threshold0 = hardware, 128 ms fixed1 = software, adjustable time
RSD_23To22		[23:22]	reserved, written as 0

Amplitude and noise threshold registers: These four registers (MAGDET_THR_REG, NMUTE_FMA2_SAP_REG, NMUTE_MPX_REG, NMUTE_EIAJ_REG; DD17 to DD20) are, like the NICAM configuration register, initialized with reasonable default values only at start-up and not later on. The system controller may change the contents of these registers, but it is recommended to keep the defaults, because a number of measurements and tests are needed to ensure a correct operation.

Magnitude detection register

The variables in the MAGDET_THR_REG (DD17) register allow adjustments of the detection thresholds (lower and upper thresholds to form a hysteresis) for the firsts sound carrier detection (ASD step one), the multiplex pilot (BTSC, FM Radio), and the SAP carrier (BTSC), in dB relative to the nominal values.

Reasonable hysteresis sizes, i.e. the difference between upper and lower threshold, for the pilot and SAP detection are 3 to 6 [dB].

ASD_SC1_THR sets the internal threshold used for indicating a "failure" of the first ASD step (described in section Section “Search procedures (ASD mode)”) relative (in dB) to the internal constant reference value of -30 dBFS (active if ASD_SC1_THR is 0 = initial value of this variable). If ASD_SC1_THR is set to -16, the threshold is set to 0 such that a failure cannot occur, but it is doubtful whether this makes sense.

Table 122: Magnitude Detection Register (MAGDET_THR_REG, DD17)

Variable	No of	Bit	Valid	Desc.
Name	Bits	Index	Range	
MPX_PILOT_THR_UP	04	[3..0]	15..0	upper threshold for MPX pilot detection (BTSC, FM RADIO) in dB below nominal level
MPX_PILOT_THR_LO	04	[7..4]	15..0	lower threshold for MPX pilot detection (BTSC, FM RADIO) in dB below nominal level
SAP_CAR_THR_UP	04	[11..8]	15..0	upper threshold for SAP carrier detection in dB below nominal level
SAP_CAR_THR_LO	04	[15..12]	15..0	lower threshold for SAP carrier detection in dB below nominal level
RSD_17To16		[17:16]		reserved, written as 0
ASD_SC1_THR	05	[22..18]	15..-15	threshold for detection of first sound carrier (SC1) during ASD first step, relative to -30 dBFS
RSD_Bit23		[23]		reserved, written as 0

Noise threshold registers

These registers (NMUTE_FMA2_SAP_REG (DD18), NMUTE_MPX_REG (DD19), NMUTE_EIAJ_REG (DD20)) allow adjustments of the automute behavior with respect to noise.

For each carrier type (SAP, SC2 in FMA2 standards, BTSC , FM Radio, EIAJ), a relative threshold (-15 to +15 dB) and a hysteresis size (0 to 15 dB), both given in dB, are available. Each of these pairs are parameters for threshold detectors which cause an automute of the corresponding stereo (sub-) carrier in case of strong noise. As an example, register NMUTE_FMA2_SAP_REG (DD18) is listed below, the other registers have the same functionality applied to the corresponding standards.

A relative threshold setting of 0 (=default) means that the reference threshold (an internal constant) is used as the effective upper threshold, a setting of -6 means a 6 dB lower threshold etc. The threshold is selected subjectively at a degree of noise where some crackling is just audible. For SAP, a higher threshold is selected (i.e. more noise is allowed before mute) because this carrier often suffers from noise interference. A threshold setting of -16 deactivates the noise threshold detector; this is strongly discouraged at least for the SAP and SC2 cases. In the latter case (FMA2 standards), the noise threshold detector prevents both crackling and a (very unlikely) wrong stereo / dual identification (e.g. if SC2 is not present and catches crosstalk from SC1 modulated with the “European stereo” identification frequency of 117.5 Hz).

The lower threshold is the upper threshold reduced by the hysteresis size. Reasonable hysteresis size settings are 3 to 6 [dB].

Table 123: Noise Automute Control Register, FMA2/SAP (NMUTE_FMA2_SAP_REG, DD18

Variable	No of	Bit	Valid	Desc.
Name	Bits	Index	Range	
NMUTE_SAP_THR	05	[4..0]	15..-16	noise threshold adjustment for automute of SAP (-16 means automute off)
NMUTE_SAP_HYST	04	[8..5]	15..0	hysteresis size [dB] for automute of SAP
NMUTE_SC2_THR	05	[13..9]	15..-16	noise threshold adjustment for automute of SC2 in FM A2 standards (-16 means automute off)
NMUTE_SC2_HYST	04	[17..14]	15..0	hysteresis size [dB] for automute of SC2 in FM A2 standards
RSD_23To18		[23..18]		reserved, written as 0

SAP detection: The SAP detection routine is active while "M MONO" or "M BTSC" is detected or selected. This routine performs two separate hysteresis detections, one for the SAP carrier magnitude, and another for the noise level. Following truth table shows the effects of the detector outputs and the SAP detection status.

Remark: The SAP detection bits are independent from the selected SRC configuration, i.e. SAPDET can be 1 while the SAP SRC path is not currently active.

Table 124: SAP Detection by magnitude and noise detection

SAP magnitude detector	noise detector hysteresis for SAP	DEMDEC Status Register	
		SAPDET (1 means signal is present)	SAPMUT (1 means muted)
0	0	0	0
0	1	0	0
1	0	1	0
1	1	0	1

EIAJ subcarrier detection: The presence and contents (L-R, language B) of a subcarrier in the Japanese multichannel sound standard (EIAJ) is indicated by a pilot tone at 3.5 * fH. However, if the pilot tone is switched off at the same time as the subcarrier, a short noise becomes audible due to the delayed pilot detection. To prevent this noise, a detector for the subcarrier at 2* fH can be enabled by the EIAJ_CAR_DETECT bit (introduced in code version 3.2.0). If enabled, the logical output of this detector will be 1 (=carrier detected) for stereo detection. If it is 0 (=no carrier detected), any stereo detection from the hardware identification unit is immediately forced to zero. The stereo status is internally derived by the expression GST := (AST=1) AND (noise detector output = 0) AND ((EIAJ_CAR_DETECT= 0) OR (subcarrier detector output=1)).

This subcarrier detector works in the same way as the SAP carrier detector, and its thresholds can be adjusted in the same way by the control variables EIAJ_CAR_THR_UP and EIAJ_CAR_THR_LO. Due to lack of space in the DEMDEC status register, there is no status bit showing the output of the EIAJ subcarrier detector.

The new control variables are placed in the NMUTE_EIAJ_REG register.

Table 125: Control variables for EIAJ subcarrier detection

Variable name	No of bits	Bit index	Valid range	Description
EIAJ_CAR_THR_UP	04	[12..9]	15..0	upper threshold for EIAJ SUB carrier detection in dB below nominal level
EIAJ_CAR_THR_LO	04	[16..13]	15..0	lower threshold for EIAJ SUB carrier detection in dB below nominal level
EIAJ_CAR_DETECT	01	[17]		enable EIAJ SUB carrier detector 0 = sub carrier detector disabled 1 = sub carrier detector enabled

3.9.8.6 Other DEMDEC control options

The DDEP_OPTIONS1_REG register contains some additional control bits.

Table 126: DDEP_OPTIONS1_REG register

Variable name	No of bits	Bit index	Description
SRC_CFG_TABLE	03	[2..0]	SRC configuration table 0 = 5 channels max. 1 = 6 channels max. 2 = 4 channels max. 3 = 3 channels max., no PIPMONO and ADC 4 = table in YRAM (default identical with 5 channel table)
RSD_Bit3		[3]	reserved, written as 0
IDMOD_SLOW_EUR	2	[5..4]	in ASD mode, IDMOD setting when European A2 standards (B/G, D/K) are detected 0 = slow 1 = medium 2 = fast
IDMOD_SLOW_KOR	2	[7..6]	in ASD mode, IDMOD setting when M Korea standard detected 0 = slow 1 = medium 2 = fast
IDMOD_SLOW_JAP	2	[9..8]	in ASD mode, IDMOD setting when EIAJ standard detected 0 = slow 1 = medium 2 = fast
NICAMCPLL_ACQHELP_OFF	1	[10]	acquisition help for NICAM carrier loop 0 = active 1 = disabled
NICAM_ERRDETECTTIME	8	[18..11]	detection time interval for software bit error detector, in multiples of 32 ms
RSD_23To19	1	[23..19]	reserved, written as 0

SRC_CFG_TABLE selects a table of SRC channel configurations. Section "SRC constraints" explained the restriction to a maximum of five SRC channels, which is valid if SRC_CFG_TABLE is set to 0 (default). Due to design and process optimizations it is however possible to process up to six channels simultaneously, if a DSP clock of 126 MHz or more can be achieved and guaranteed under worst-case conditions. Table 127 shows the SRC configuration table for this mode, enabled by setting SRC_CFG_TABLE to 1.

Although no longer required, but for compatibility reasons, EPMODE=3 still enforces mono decoding, by overriding the identification and possibly skipping the stereo processing channel. The other defined settings of SRC_CFG_TABLE are intended for internal test purposes.

The CPU load of the DSP, means the fraction of CPU time spent on processing and not waiting for input/output tasks, can be read out from the INF_CPULOAD_REG register, as a signed 24 bit integer (averaging time is approx. 223 DSP clock cycles). If the load is more than approx. 0.95 * full scale, the sound output may be distorted and the DSP clock should be increased.

Table 127: SRC Configuration - up to six channels

Std. Group	**Ref. No.**	**Ctrl. Bits** **SRCPREF (dec)**	**active SRC channels and preprocessing routines**								**Active Channels**
			FMA2	FMMONO	BTSC	NICAM	SAP	EXTAM	PIPMONO	ADC	
FM A2	1	0,1,2	2						1	2	5
	1m	3	2						1	2	5
EIAJ, FM Radio	2	0,1,2			2				1	2	5
	2m	3			2				1	2	5
BTSC	3	0			2		1			2	5
	4	1			2				1	2	5
	5	2			2		1		1		4
	6	3/X		1			1		1	2	5
NICAM	7	0		1		2			1	2	5
B/G,I,D/K	8	1				2			1	2	5
	9	2		1		2			1		4
	10	3/X	2						1	2	5
NICAM L	11	0				2		1		2	5
	12	1				2			1	2	5
	13	2				2		1	1		4
	14	3/X						1	1	2	4

NICAM_ERRDETECTTIME selects the interval of counting bit errors to un-mute NICAM sound. The interval length is 32 ms * NICAM_ERRDETECTTIME.

The IDMOD_SLOW_xyz variables select the IDMOD setting for the identification unit which are used in ASD mode after a stereo or dual ident has been found. The ASD procedure always applies the "fast" mode (IDMOD=2) first to minimize the detection time, until stereo or dual has been detected, and then switches to the setting given in this register for the current standard. Separate settings are available for the European (B/G, D/K), Korean and Japanese standards. (For Japan, the corresponding variable IDMOD_SLOW_JAP is set to 1 (default) to avoid problems with certain TV test generators.)

The NICAMCPLL_ACQHELP_OFF bit can be used to disable an acquisition help for the NICAM carrier loop (for test purposes), which has been added as a countermeasure against locking problems of this PLL under unwanted n* fH phase modulation (see PR no. 631 and related documentation). It is recommended to keep it active.

SAP_BW selects the bandwidth of the bandpass filter of the FM subchannel demodulator for SAP, that is either “narrow” (SAP_BW =0) or “wide” (1). Latter results in a more flat frequency response (with dbx) and lower harmonic distortion and is therefore the recommended setting, although the default value is currently still 0.

The DD_OPTIONS2_REG register provides control bits related to a known (and not yet solved) problem in the timing PLL the sample rate conversion: after a audio clock interruption (e.g. no I^{2}S input clock, or power down mode of WS PLL) there is a chance of approx. 1:60 that one or more SRC output signals are permanently distorted. To prevent this, a “watchdog” routine has been added in the DSP code which enforces a new initialization of the SRC PLLs after a clock interruption of more than 6000 DSP clock cycles. This watchdog can be switched off by WATCHOUTCLK_OFF=1.

However it was found that in rare cases the distortion might still occur. A “sanity check” of the internal buffers can remove the distortion, but this test cannot be active all the time because it causes click noise when the SRC configuration changes. Therefore it is recommended to enable the sanity check (by CHECK_SRC_BUFFERS=1) only once for a short time (few ms) after a possible interruption or disturbance of the audio clock.

Table 128: DD_OPTIONS2_REG register

Variable name	No of bits	Bit index	Description
WATCHOUTCLK_OFF	01	[0]	watchdog for failure of output audio clock 0 = active 1 = disabled
SRCPLL_OUT_INIT	01	[1]	initialize SRC PLL for output thread 0 = no action 1 = enforce initialization
SRCPLL_32K_INIT	01	[2]	initialize SRC PLL for 32 kHz input thread 0 = no action 1 = enforce initialization
SRCPLL_35K_INIT	01	[3]	initialize SRC PLL for 35 kHz input thread 0 = no action 1 = enforce initialization
SRCPLL_53K_INIT	01	[4]	initialize SRC PLL for 53 kHz input thread 0 = no action 1 = enforce initialization
RSD_23To5	01	[23..5]	reserved, written as 0

ADC channel control: For the sake of additional flexibility, the ADC channels from the I^{2}D link input can be mapped to any of the DEMDEC internal signals and channels by means of the register DEM_ADC_SEL_REG. It also allows to mute / unmute the

PIPMONO and ADC channels. A mute bit for the EXTAM signal is not present here as this is managed by the standard dependent signal routing of DDEP. All control bits in this register are active in DDEP or expert mode.

Table 129: ADC channel control register (DEM_ADC_SEL_REG, DD24)

Variable Name	No of Bits	Bit Index	Desc.
MAP_EXTAM	02	[1..0]	ADC channel from DLINK to EXTAM $0 = primary channel 1 $1 = primary channel 2 $2 = secondary channel 1 $3 = secondary channel 2
MAP_PIPMONO	02	[3..2]	ADC channel from DLINK to PIPMONO $0 = primary channel 1 $1 = primary channel 2 $2 = secondary channel 1 $3 = secondary channel 2
MAP_ADCST_L	02	[5..4]	ADC channel from DLINK to ADC stereo ch. L $0 = primary channel 1 $1 = primary channel 2 $2 = secondary channel 1 $3 = secondary channel 2
MAP_ADCST_R	02	[7..6]	ADC channel from DLINK to ADC stereo ch. R $0 = primary channel 1 $1 = primary channel 2 $2 = secondary channel 1 $3 = secondary channel 2
MUTE_PIPMONO	01	[8]	soft mute PIPMONO channel $0 = active $1 = muted
MUTE_ADCST_LR	01	[9]	soft mute ADC channels $0 = active $1 = muted
RSD_23To10	14	[23..10]	reserved, written as 0

3.9.8.7 Status registers

DEMDEC status register: Table 130 contains the most important status information. In a typical application, mainly the generalized stereo and dual flags (GST and GDU), the SAP detection flag (SAPDET) and (partly) the standard detection result (STDRES) are of interest to the user, as explained in .

In order to simplify the control software, this register is also available as a low latency register.

The GST and GDU bits indicate the type of signal present on the DEC output: both GST=0 and GDU=0 means a monophonic signal, GST=1 indicates a stereophonic and GDU=1 a bilingual (dual) signal. GST and GDU are never both 1 at the same time. By evaluating these flags, the user is relieved from checking the type of standard (i.e. A2, NICAM, BTSC) and from selecting the corresponding identification status bits, which are also present in the device status register as additional information. Table 131 explains in terms of logical equations the dependence of the GST and GDU bits on the other status flags.

Remark: If automute is active, or if "forced mono" is applied (by SRCPREF=3), the “primary” identification may indicate a stereo reception (e.g. AST = 1) while GST is zero.

Furthermore, in the audio backend source selectors, GST and GDU are used for an automatic language selection in case of dual transmission at the input of each two-channel audio processing path (e.g. MAIN, DAC): By selecting a special matrix code, for example "automatrix language A", the system switches from an identity matrix (output=input, while GDU is 0) to language A (first of the two channels) if the GDU flag is 1. Language B can be "pre-selected" accordingly.

Bits VDSP_C, NICST_C, and NICDU_C are copies of the identical information in the NICAM status register, see below. NAMUT indicates an automute condition, that is, the DEC output does not carry the NICAM signal, but the same signal as the mono channel, because of an exceeded bit error count or synchronization loss. In this case, GST and GDU are always zero although NICST_C or NICDU_C may still be 1 (depending on the NICAM decoder status: if still in sync (VDSP=1), NICST_C or NICDU_C can be 1; if not in sync or no audio (VDSP=0), NICST_C and NICDU_C are 0).

In expert mode, STDSEL, GST, GDU, NAMUT, BPILOT, SAPDET, BAMUT, SAPMUT and AAMUT are 0

Table 130: DEMDEC status register (INF_MAIN_STATUS_REG, DD01)

Variable Name	# of Bits	Bit Index	Detailed Description ($ = HEX values)
STDRES	5	[4..0	standard detection result (ASD mode), or selected standard in SSS mode 00 = failed to find any standard or not supported by device 01 = B/G (still searching, SC2 not (yet) found) 02 = D/K (still searching, SC2 not (yet) found) 03 = M (still searching, no ident or pilot found) 04 = B/G A2 05 = B/G NICAM 06 = D/K A2 (1) 07 = D/K A2 (2) 08 = D/K A2 (3) 09 = D/K NICAM 10 = L NICAM 11 = I NICAM 12 = M Korea 13 = M BTSC 14 = M EIAJ 15 = FM Radio, 10.7 MHz IF (50 us de-emphasis) 16 = FM Radio, 10.7 MHz IF (75 us de-emphasis) 17 = FM radio, variable IF (50 µs de-emphasis) 18 = FM radio, variable IF (75 µs de-emphasis) 31 = still searching for a standard (can occur a short time after RESTART)
GST	1	[5]	general stereo flag (ident source determined by currently detected or selected standard) 0 = No stereo mode 1 = Stereo mode detected
GDU	1	[6]	general dual flag 0 = No dual mode 1 = Dual mode detected
APILOT	1	[7]	A2 or EIAJ pilot tone detected 0 = False 1 = True
ADU	1	[8]	A2 or EIAJ ident dual flag 0 = False 1 = True

Table 130: DEMDEC status register (INF_MAIN_STATUS_REG, DD01) *…continued*

Variable Name	# of Bits	Bit Index	Detailed Description ($ = HEX values)
AST	1	[9]	A2 or EIAJ ident stereo flag 0 = False 1 = True
AAMUT	1	[10]	SC2 (if A2 mode) or EIAJ subchannel muted due to noise 0 = False 1 = True
BPILOT	1	[11]	BTSC or FM radio pilot tone detected (stereo indicator) 0 = False 1 = True
SAPDET	1	[12]	SAP carrier detected 0 = False 1 = True
BAMUT	1	[13]	BTSC stereo muted due to noise (if noise detector enabled) 0 = False 1 = True
SAPMUT	1	[14]	SAP muted due to noise (if noise detector enabled) 0 = False 1 = True
VDSP_C	1	[15]	NICAM decoder VDSP flag 0 = DATA or undefined format 1 = SOUND
NICST_C	1	[16]	NICAM decoder stereo flag 0 = False 1 = True
NICDU_C	1	[17]	NICAM decoder dual flag 0 = False 1 = True
NAMUT	1	[18]	NICAM automute flag 0 = not muted 1 = muted (fallback source)
RSSF	1	[19]	NICAM reserve sound switching flag (=C4), see NICAM specification 0 = analogue sound carrier conveys different contents than NICAM carrier 1 = analogue sound carrier conveys same contents as the NICAM carrier (M1 if DUAL)

Table 130: DEMDEC status register (INF_MAIN_STATUS_REG, DD01) *…continued*

Variable Name	# of Bits	Bit Index	Detailed Description ($ = HEX values)
INITSTAT	1	[20]	initialisation status (set to 0 upon read access) 0 = no reset performed 1 = reset has been applied to DSP and init routine has been executed
SRC_UNLOCK	1	[21]	SRC out of lock flag (output or 32 kHz PLL) 0 = SRC in normal operation 1 = SRC out of lock, outputs are muted or distorted
SRD_STATUS	2	[23..22]	sample rate detector 0 = 32 kHz 1 = 44.1 kHz 2 = 48 kHz 3 = less than 6 kHz / invalid I2S input clock

[1] The output PLL of the SRC may be temporarily unlocked in case of a sample rate switch or an invalid audio clock in I^{2}S slave mode, or if the DSP is overloaded (too low DSP clock selected).

Table 131: Generalized Stereo / Dual Flags

Standard type	GST equals	GDU equals
FM A2, EIAJ	AST and APILOT and NOT(AAMUTE) and NOT(forced mono)	ADU and APILOT and NOT(AAMUTE) and NOT(forced mono)
NICAM	NICST_C and NOT(NAMUT) and NOT(forced mono)	NICDU_C and NOT(NAMUT) and NOT (forced mono)
BTSC, FM Radio	BPILOT and NOT(BAMUT) and NOT(forced mono)	0

[1] “and” means a logical (not bit-wise) “and” operation, “NOT” means logical negation.

Status evaluation during forced mono mode

Table 126 suggests how the standard-dependent status flags can be used to derive the current sound mode while the controller has selected "forced mono": by skipping the "and NOT(forced mono)" term, the equations for the actually received sound mode are obtained.

Table 132: Derive stereo / dual information - forced mono

standard type	stereo received if	dual received if
FM A2, EIAJ	AST and APILOT and NOT(AAMUTE)	ADU and APILOT and NOT(AAMUTE)
NICAM	NICST and NOT(NAMUT)	NICDU and NOT(NAMUT)
BTSC, FM Radio	BPILOT and NOT(BAMUT)	0

NICAM status registers: Due to the automatic switching provided by DDEP and the flags in the device status register, the NICAM status register (listed inSection 3.9.8.8 “Noise detection”) is normally not needed by the system controller. This applies even more to the NICAM additional data register as long as no application of this data stream is known.

As mentioned above, the VDSP, NICST and NICDU status bits are also available in the device status register. The bit error counter is used by the NICAM automute function of DDEP. CO_LOCKED (formerly called OSB in TDA9875A and TDA9874A) is used by the standard detection procedures to identify a NICAM standard.

Table 133: NICAM status register (INF_NICAM_STATUS_REG, DD02)

Variable name	No of Bits	Bit Index	Description
ERR_OUT	08	[7..0]	NICAM error counter: number of parity errors found in the last 128ms period
CFC	01	[8]	NICAM Configuration Change 0 = No configuration change 1 = Configuration change at the 16 frame (CO) boundary
CO_LOCKED	01	[9]	NICAM frame and CO synchronization 0 = Audio output from NICAM part is digital silence 1 = Device has both frame and CO (16 frames) synchronization
NACB	04	[13..10]	NICAM application control bits (see C1..C4 in NICAM transmission)
VDSP	01	[14]	Identification of NICAM sound 0 = DATA or undefined format 1 = SOUND
NICST	01	[15]	NICAM stereo flag 0 = No NICAM stereo mode (= Mono mode if NICDU = $0) 1 = NICAM stereo mode
NICDU	01	[16]	NICAM dual mono mode 0 = No NICAM dual mono mode (= Mono mode if NICST = 0) 1 = NICAM dual mono mode
RSD_23To17	07	[23..17]	reserved

3.9.8.8 Noise detection

DDEP uses a noise detector for the automute function in non-NICAM standards. The noise detector consists of a fourth-order bandpass connected to one of the two FM demodulators, followed by a rectifier (absolute value operation) and a lowpass. The bandpass center frequency can be switched either to a low band (center at 2.5 * line frequency fh) or a high band (7.5 * fh). The configuration that is currently set by DDEP and the output level can be read from the Noise detector status register. NOISELEVEL is a 22 bit positive signed integer; it can be converted to dBFS (decibel relative to full scale (=221)) by 20 * log10(NOISELEVEL / 221).

Table 134: Noise detector status register (INF_NOISELEVEL_REG, DD08)

Variable Name	No of Bits	Bit Index	Description
NDETCH_STAT	1	[0]	status noise detector channel 0= channel 1, 1= channel 2
NDETPB_STAT	1	[1]	status noise detector passband 0= low (2.5fh), 1= high (7.5fh)
NOISELEVEL	22	[23..2]	noise detector output

3.9.8.9 Muting all DEMDEC outputs

DDEP does not provide a function to mute all sound outputs in case of very bad reception, nonexistent sound carriers, or unused TV channel. This decision should be taken by the system controller. Next to a video signal detection, a further criterion is the noise level measured by the noise detector. However, only one noise detector is available, and it should be switched to the first sound carrier (a second FM carrier may not exist). This is the case in MPX (BTSC, EIAJ, FM Radio) and NICAM standards and in one of the “mono” detections (status STDRES is 1, 2, or 3), but not while an A2 standard is detected (ASD mode) or selected (SSS mode), means while STDRES is 4 (=B/G A2), 6, 7, 8 (D/K), or 12 (Korea).

Therefore, detecting “unacceptable” sound generally only works in ASD mode, since only with ASD the (detected) standard code is always a NICAM standard (I or L) or a “mono standard” (STDRES = 1, 2, or 3) at very high noise levels. In SSS mode it is possible only with standards other than FM A2.

The table below lists the noise levels at which it seems reasonable to consider the sound unacceptable, means above these levels the DEMDEC outputs may be muted (set DDMUTE bit to 1).

Table 135: Noise levels for muting per standard

Standard	Noise Detector Band and Channel	FM Filter Bandwidth (internal)	Max. Noise Detector Level (dBFS)	Detector Level ‘unacceptable’ threshold (dBFS)
FM A2	low /ch.2	irrelevant	-8	-11
NICAM	low /ch.1	irrelevant	-8	-11
B/G or D/K Mono (std, code 1 or 2)	low /ch.1	irrelevant	-8	-11
MPX (BTSC,EIAJ,FM radio)	high /ch.1	wide/extra wide	-9	-11
M Mono (std, code 3)	high /ch.1	wide/extra wide	-9	-11

3.9.8.10 Using DDEP in a set design

As explained later in section Section “Search procedures (ASD mode)”, the ASD feature must be used with great care to avoid malfunctions of the set, such as a distorted or noisy audio output, due to a wrongly detected sound standard.

This section proposes ways to handle the problems involved with the multitude of sound standards and to minimize the risks.

Application related constant settings: Registers which are usually written only at startup and not changed later :

- optional control variables: EXTAM, OVMADAPT, OVMTHR, FILTBW (if OVMADAPT not used)
- additional scaling for NICAM and EXTAM
- noise and magnitude thresholds if default settings are not wanted.

Prerequisites and user interface: Every application or set should provide some kind of setup facility. A TV or VCR will make use of an On-Screen Display (OSD), while the application software for a PC TV card will probably include a setup or configure dialog. Often these setup facilities do not satisfy the end users demands, which can be divided in two groups:

- Users with little or no technical background do not want to be bothered with questions of video or sound standard settings.
- Advanced users like to have more influence on settings.

Therefore it is recommend to build a user interface which allows two alternative selections:

1. Country / area selection : everyone should know where the set is currently located. Not every country or state needs to be listed, they may be grouped in areas with common TV standards.
2. Direct standard selection : the user can select a video standard (PAL, NTSC, SECAM; not needed if the video decoder is capable of auto-detection), and one of the five sound standard groups like "B/G", or even the precise stereo / multichannel standard like "B/G NICAM".

The control software should contain a table of countries and areas to map the user-supplied info from alternative (1) to the STDSEL variable for the ASD function. Optionally this table may contain a default standard for each area, which is selected (via SSS) in case the ASD returns a "failed" code(Table 136 gives an example), but it may be modified according to the setmaker's preferences. One may consider to add standards present in border regions of some countries and make the table more detailed, for example allow both L/L' and B/G at the French/German border.

A special case is PAL-N (line frequency 15.625 kHz with BTSC sound) in Argentina and some other South-American countries, where the FHPAL bit is set to 1.

If an unused (empty) TV channel is selected, it may occur that ASD cannot find any sufficiently strong sound carrier, reports a failure (STDRES=0) and mutes the outputs. However, in most cases, the noise is strong enough that a sound carrier is (erroneously) detected randomly at one of the selected test frequencies. Therefore, the system controller should check whether the video decoder is detecting a valid picture. If no video is present, it mutes the audio outputs (via the DDMUTE bit) and blanks the screen.

An alternative approach to the “default standard” is to prevent the “failure” condition by setting the detection threshold to 0 and set only one bit in STDSEL, such that it is ensured that corresponding standard is always “detected” (known as “forced standard detection [FSD]”).

It is important that the standard information from the video decoder is not considered for the sound standard detection (e.g. NTSC video -> select only M standard sound), because almost every combination of video and sound standards may occur in the field.

Of course, hardware-related aspects may also have an influence on the control software design: it may be required to select a different filter for the SIF signal depending on the country, even before ASD can be used. Typically, a conventional intercarrier configuration is used for M standards with a SIF passband up to 5 MHz, while QSS with a separate SAW filter and a passband of 5 to 7 MHz is applied for the other standards.

Table 136: Areas and ASD Settings

Country/ Area	ASD Control (STDSEL variable)					
	B/G	D/K	L/L'	I	M	recommended fallback standard (set by system controller via SSS)
Germany, Netherlands, Italy, Austria, Switzerland, Malaysia, Australia, Israel, Saudi Arabia	1	1[1]				B/G A2
Scandinavia, Spain, Belgium, New Zealand, Singapore	1					B/G NICAM
Great Britain, Hong Kong				1		I NICAM

Table 136: Areas and ASD Settings

Country/ Area	ASD Control (STDSEL variable)					
	B/G	D/K	L/L'	I	M	recommended fallback standard (set by system controller via SSS)
France	1[1]		1			L NICAM
Eastern Europe, GUS, China	1[1]	1				D/K NICAM
USA, Canada, Mexico, Brazil, Taiwan (NTSC, set FHPAL=0)					1	M BTSC
Argentina, … (PAL, set FHPAL=1)					1	M BTSC
Korea					1	M A2 (Korea)
Japan					1	M EIAJ (TESONIC: M MONO)

[1] May be useful in areas near border.

Auto-tune process: At the auto-tune or scan run, the system controller steps through all possible TV channels, checks if a signal is present and builds a program list. This is performed at the installation of the set or device, but rarely later on. I recommend to store only the information whether a video signal was found or not, but not to store the detected sound standard for later use, because there is no advantage in doing so:

1. Detecting a standard by means of ASD practically does not take any longer (only some milliseconds) than the identification of a multichannel sound standard needs anyway.
2. Storing a wrongly identified standard multiplies the seriousness.
3. The scan can be faster if only the video standard detection is required. Therefore it is advantageous to do the sound standard detection only after changing the channel, by using the ASD mode.

Channel switch procedure: With the information from the setup, a channel switch is carried out as follows:

1. Write the DDEPR, set EPMODE=0 (select ASD mode), DDMUTE = 1 (soft-mute outputs within ~32 ms) and REST = 0.
2. Wait at least 32 ms until softmute is complete.
3. Program tuner and (possibly) IF demodulator with settings for the new channel. Wait until the SIF signal is stable.
4. Write the DDEPR with STDSEL taken from the area table or the direct standard selection, DDMUTE = 0 (means un-mute after ASD first step) and REST = 1. A standard search is now started, and all DDEP-internal states are reset (automute hysteresis, pilot detection, FM identification, overmodulation adaptation, etc.).
5. At least 70 ms later, read the DEMDEC status register. After the minimal allowed time, read the video detection info from the video decoder.
6. If no video has been detected, blank screen (if possible) and mute audio.
7. If video is present, but ASD has failed with STDRES=0, select the default standard via SSS mode (includes another "restart"). A warning may be issued on the OSD.

8. During normal operation, reflect changes of the GST and GDU bits on the OSD together with the sound standard. If a NICAM standard is found, also display the RSSF information. Likewise, with M BTSC reception, a SAP carrier detection should be indicated. Some audio backend features like I-Stereo, Dolby® Pro Logic® etc.may have to configured according to the sound mode (M / St. / Du).

3.9.8.11 Details of operation

This section explains the DDEP functionality in more detail, with focus on the Automatic Standard Detection (ASD).

Search procedures (ASD mode): The standard detection works in two steps. The first step is to find the first sound carrier and determine the standard group. In a second step, it is attempted to identify the received stereo (or multi-channel) standard.

The first sound carrier search is configured via the five bits of the STDSEL variable and delivers a result code in less than 50 ms. The result code may indicate a failure if no sufficient signal strength is found. The second search step can be skipped if the first step already yields an "unambiguous" stereo standard, i.e. I NICAM and L NICAM.

For the three other cases (B/G, D/K, and M standards), the "step 2" search procedures attempt to identify the stereo standard as soon as possible. If a stereo standard is detected, the procedure is held as long as the standard can be identified. Otherwise the search continues after a timeout, this means the procedure remains in an infinite loop until the next channel switch or it is otherwise stopped. This serves three purposes:

- A temporary mis-identification of the stereo system (due to special signal conditions) does not make the search hang.
- If an identification is not found immediately due to temporary bad reception (e.g. airplane flutter), or is switched on by the broadcasting station some time after the channel switch, it is detected.
- The decoder can adapt if the stereo system changes (e.g. from B/G A2 to NICAM), although this should not occur in the field, but is common practise in the laboratory.

The current state of the search "engine" is not completely visible from the outside, but the code of the currently active standard is available via STDRES in the device status register.

Standard search first step

Figure 75 show the spectral conditions of four different TV standards. Amplitude axis and sound carrier width not to scale, frequency is relative to picture carrier in RF domain.

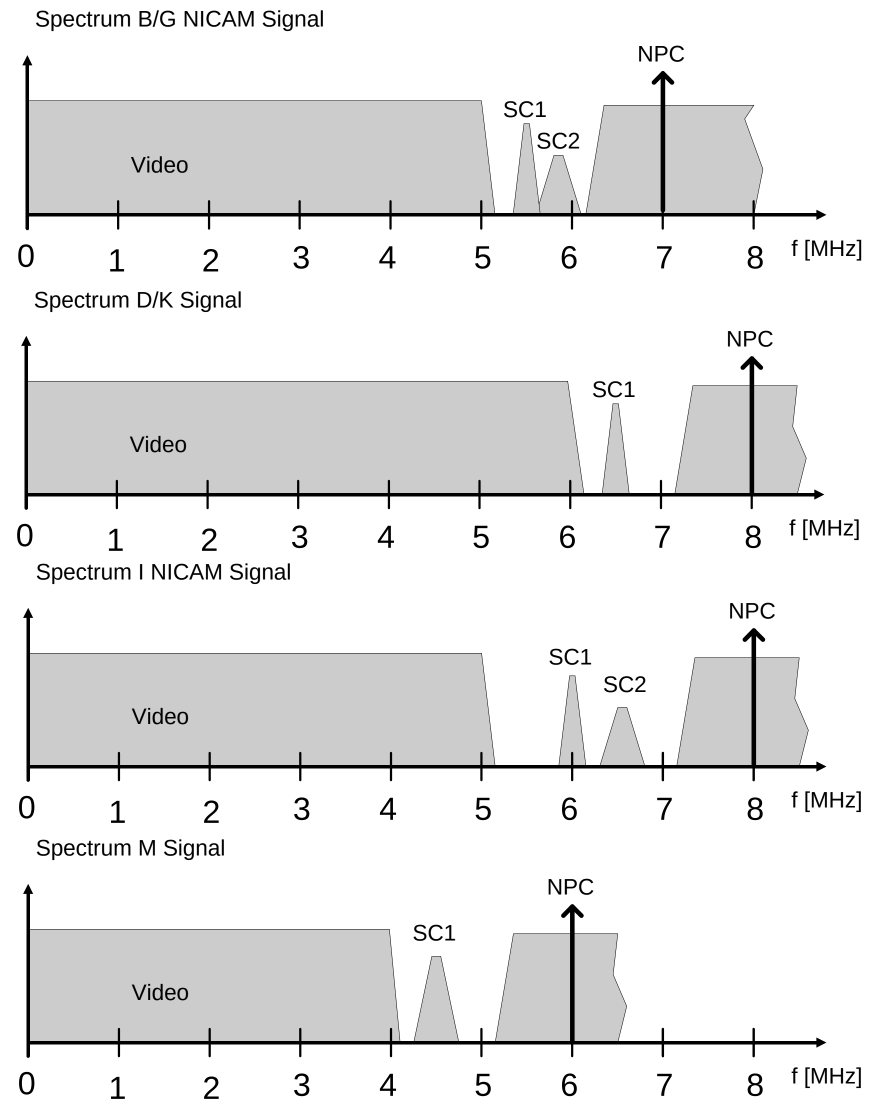


Fig 75.Spectra of TV HF signals

The neighbor picture carrier (NPC) distance is assumed worst-case (e.g. 7 MHz for B/G, which only occurs in CATV but not in terrestrial broadcasts). The spectrum seen at the SIF input depends on the passband of the sound filter in the IF stage. For PAL and SECAM QSS applications, the equivalent 2. SIF passband is usually from 5 to 7 MHz, while for NTSC (M standards), conventional intercarrier is most common, such that the NPC is suppressed. Of course the video power spectrum of a live picture is not constant, therefore a certain amplitude at a certain location is not guaranteed. Only in SECAM the FM color subcarriers have a constant amplitude. This means that high frequent video components might be detected as sound carrier.

A second sound carrier (SC2) is not always present.

Furthermore, a 6.5 MHz sound carrier may be an AM carrier in L/L´ or a FM carrier in D/K, they cannot be reliably distinguished since a not modulated FM sound carrier is identical to a not modulated AM carrier.[3]

These facts show that it is impossible to identify a sound standard with 100% certainty by means of simple (and fast) amplitude detection and without further knowledge. However, in practise and without any special test signals, a correct standard detection is possible in probably more than 99% of all cases.

In order to minimize the risk of wrong detection and to solve the D/K vs. L/L´ ambiguity, the five standard detection control bits STDSEL have been introduced. By setting only those bits for "expected" or "supported" standards, any other standard will not be detected. It depends on the application, i.e. IF filter types, supported standards etc., and the location (country / area) of the set, which flags are set. For further recommendations, please refer to Section 3.9.8.10 “Using DDEP in a set design”.

After a restart, the standard search performs the following:

1. mute all DEMDEC outputs immediately.
2. measure the amplitude at the frequency 6.5, 6.0, 5.5 and 4.5 MHz. If the corresponding standard is not selected, skip the measurement and set the measured amplitude to 0.
3. find the largest of all measured amplitudes.
4. if the maximum amplitude is below a selectable threshold (currently -30 dBFS +/- 15 dB, via control variable ASD_SC1_THR; -30 dbFS is about 20 dB below typical values), indicate “detection failed” and STOP (enter idle state).
5. determine standard group according to following table:
6. start demodulating SC1, unmute outputs, i.e. start reproducing mono sound (normally FM, except standard L: internal AM demodulation or input via ADC if bit EXTAM=1‘.
7. in case of B/G, D/K, or M, proceed with search for stereo system. Otherwise apply a static demodulator setup (standards I or L NICAM).

Remark: If both STDSEL[1] and STDSEL[2] are set to 1 (which does not make sense), D/K is given priority.

The above procedure is only started explicitly via the REST bit, not by any other event.

The second sound carrier cannot be considered in this first search step, as it may be non-existent or too close to the noise floor, and detecting an identification such as A2 stereo takes too long, the user wants to hear the sound immediately.

Table 137: Deciding for a standard group

Maximum found at	Condition (order is relevant)	Decided standard	Possible source(s) of misidentification
6.5 MHz	STDSEL[1]=1	D/K	L/L´ SC1; NPC VSB B/G, M
	STDSEL[2]=1	L/L´	D/K SC1; NPC VSB B/G, M
6.0 MHz	STDSEL[3]=1	I NICAM	M NPC; D/K, L high frequency video (unlikely)
5.5 MHz	STDSEL[0]=1	B/G	M NPC VSB; D/K or L high frequency video (unlikely)
4.5 MHz	STDSEL[4]=1	M	video components from other standards

[3] Additionally, a FM carrier also yields a (distorted) output if AM demodulation is applied due to the filter shape.

Search procedure for B/G standards

The B/G search routine is started when ASD step 1 has found a maximum at 5.5 MHz. A simplified flowchart of this procedure is given Figure 76.

The timeout for NICAM or A2 identification is 2 seconds.

The DC offset detection is a method to speed up detection.

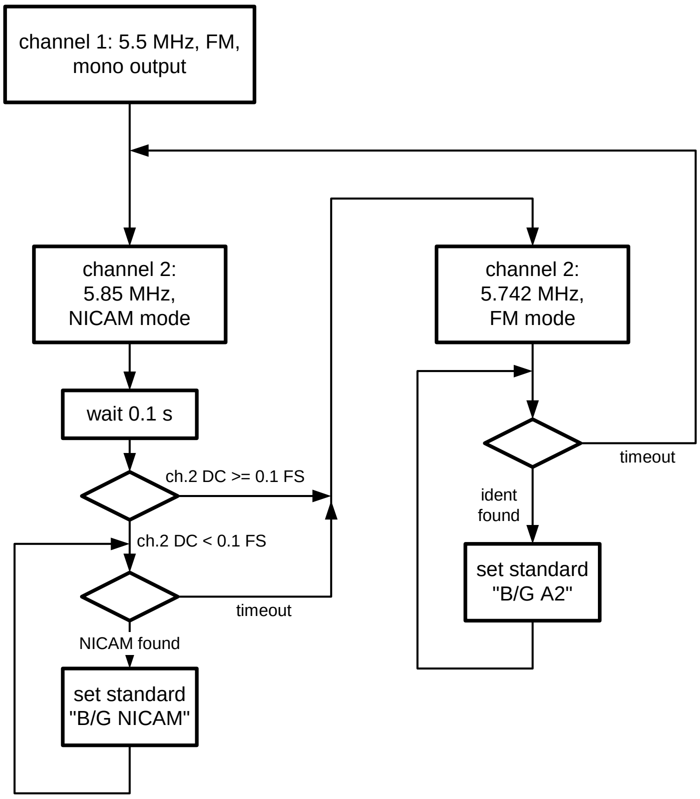


Fig 76.B/G Search Procedure

Search procedure for D/K standards

Here, another amplitude detection is performed at three of the possible SC2 frequencies in order to start the search at the most likely location. 5.742 MHz is not measured as there are often video components, and the pilot detector is not used at that frequency. As in the B/G search procedure, the timeout for NICAM or A2 identification is 2 seconds.

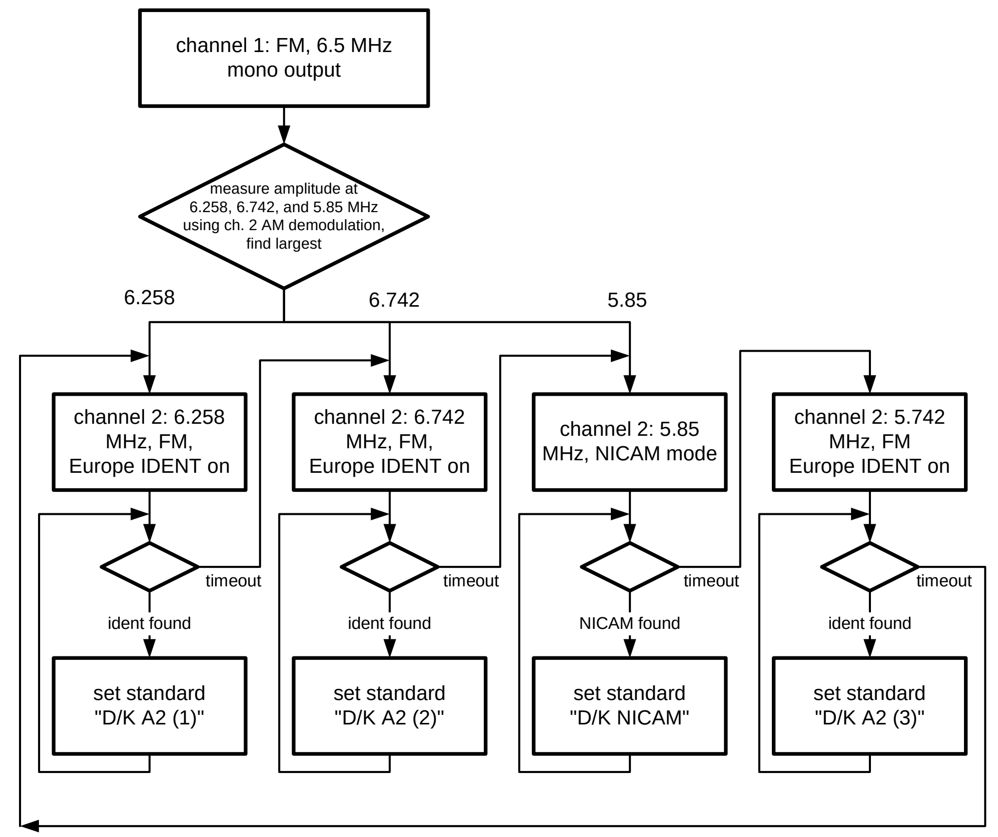

channel 1: FM, 6.5 MHz mono output
measure amplitude at 6.258, 6.742, and 5.85 MHz using ch. 2 AM demodulation, find largest
6.258
6.742
5.85
channel 2: 6.258 MHz, FM, Europe IDENT on
channel 2: 6.742 MHz, FM, Europe IDENT on
channel 2: 5.85 MHz, NICAM mode
channel 2: 5.742 MHz, FM Europe IDENT on
timeout
timeout
timeout
timeout
ident found
ident found
NICAM found
ident found
set standard "D/K A2 (1)"
set standard "D/K A2 (2)"
set standard "D/K NICAM"
set standard "D/K A2 (3)"

Fig 77.D/K search procedure

Search procedure for M standards

Pilot lock indication

The MPX demodulator provides the lowpass filtered in-phase product of the received and the PLL- regenerated pilot carrier via an internal hardware status register. This information (combined with a hysteresis detector in software) acts as a reliable lock indication . The upper / lower thresholds are selectable via register MAGDET_THR_REG.

If "M BTSC" is detected, the "wide" filter is always active for the FM demodulator and overrules any other setting from the DDEPR or determined by the OVMADAPT routine.

Search procedure approach

As the MPX demodulator is connected to the FM demodulator channel 1, it is possible to search for a BTSC pilot and a FM ident in parallel. However, the ident can only operate in one of the Europe, Korea and EIAJ modes and is connected to channel 2, thus only either M Korea or M EIAJ can be checked at a time. Therefore the search procedure measures the FM subchannel magnitude and starts the ident in the "more promising" configuration, while also checking the BTSC pilot detection in parallel.

A separate SAP detection routine is active whenever the current standard is "M MONO" or "M BTSC", as explained in sectionSection “SAP detection” .

Line frequency handling

Two different line frequencies are used with BTSC, and as the pilot PLL of the MPX demodulator is very narrow (10 or 20 Hz), the current line frequency shall be provided by the ASD routine (or by the system controller in SSS mode via the FHPAL variable).

The search procedure starts with trying to get a pilot detection at the "preferred" pilot frequency f_{H1}, this is the NTSC line frequency (f_H) of 15734 Hz if bit FHPAL in the DDEPR equals 0, or the PAL f_H of 15625 Hz otherwise. If this detection is not successful within a timeout period of 400 ms (about four times the typical detection time), the search proceeds with testing the other possible line frequency f_{H2}.

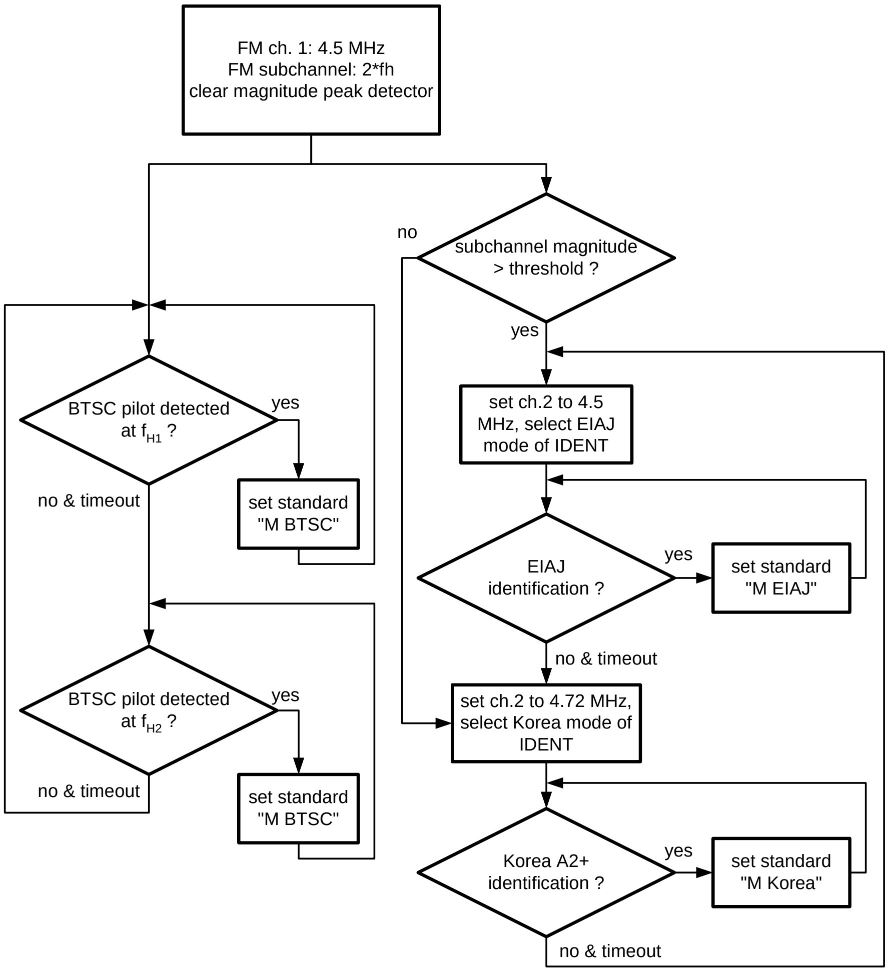

FM ch. 1: 4.5 MHz
FM subchannel: 2*fh
clear magnitude peak detector
no
subchannel magnitude > threshold ?
yes
BTSC pilot detected at f_H1 ?
yes
no & timeout
set standard "M BTSC"
set ch.2 to 4.5 MHz, select EIAJ mode of IDENT
EIAJ identification ?
yes
set standard "M EIAJ"
no & timeout
set ch.2 to 4.72 MHz, select Korea mode of IDENT
BTSC pilot detected at f_H2 ?
yes
no & timeout
set standard "M BTSC"
Korea A2+ identification ?
yes
set standard "M Korea"
no & timeout

Fig 78.Search procedure for M standards

Using the SSS mode: Unlike the ASD mode, the SSS mode requires the controller to set an appropriate value for the FM ident time constants via the IDMOD bits (11 means "off/reset", thus no stereo or dual detection possible) for the FM A2 and EIAJ standards (don't care for all other standards). The ident area code is known from the standard code.

The STDSEL variable must only have values of 4 to 18, others will result in a "failure" code (STDRES = 0 in DEMDEC status register) and mute all outputs.

Remark: STDSEL is only evaluated at "restart".

FM Radio

FM Radio can only be activated via SSS. It does not make sense to include it into the carrier search procedure in ASD mode, because FM Radio requires special settings for entire receiver (tuner, IF demodulator, filters, screen switched off etc.) and is thus known beforehand.

The FM Radio feature of the current DEMDEC is merely a "by-product" of the BTSC decoder because the same hardware MPX demodulator can be used. It cannot compete with a dedicated high-performance FM Radio receiver solution. The digital filter selectivity by itself is not sufficient, at least one narrow bandpass before the SIF input (to suppress neighbor channels) is required. It depends on the tuner and IF application how FM Radio is handled.

FM filter bandwidth is "wide" as in BTSC mode. Pilot frequency is 19 kHz, nominal pilot level is 7.5 kHz, hysteresis thresholds are selectable via register MAGDET_THR_REG.

Four different FM Radio settings can be selected see Table 120. Choices are 50 µs (Europe) or 75 µs (USA) de-emphasis, and either a fixed 10.7 MHz IF or a freely selectable IF determined by the contents of the carrier1 variable (DEM_CA1_REG). A carrier frequency of 10.7 MHz is outside the specified (~4 to 10 MHz) range of the SIF input, but works fine as long as no interferers at IF +/- 6.75 MHz are present.

As the frequency resolution of the tuner is typically only 62.5 kHz, it is recommended to always use the variable IF option and fine-tune the mixer frequency via carrier1 within a range of +/-31.25 kHz, to avoid this offset which may lead to earlier clipping or distortion.

The 24 bit value Ω_1 of the `CARRIER1` control variable can be computed according to:

$$\Omega_1 = round\left(\frac{f_1}{f_{mixer}} 2^{24}\right)$$

(f_1 = desired mixer frequency, fmixer = 13.5 MHz).

For example, f_1 = 4.5 MHz yields Ω1 = 5592405 (dec) = 555555 (hex). Values for common mixer frequency settings are listed in the description of CARRIER1 and CARRIER2.

There are two options to implement fine-tuning:

- Since today's tuners use a crystal-based oscillator with high precision, the optimal mixer frequency for a certain RF frequency (e.g. 87.60 MHz) can be precisely computed. In other words, the frequency offset due to the limited frequency resolution of the tuner can be compensated via carrier1. For example, if the offset is +25 kHz (i.e. FM carrier is located 25 kHz above the nominal mixer frequency like 4.5 MHz), CARRIER1 is increased by 31069 (dec).
- The current frequency offset can be measured via the SC1_DC status variable (DC output of the DC notch filter, 24 bit signed integer) in INF_DC1_REG (DD06):

$$\Delta f = \frac{SC1\_DC}{2^{24}} f_{S,FMDEM}$$

with $f_{S,FMDEM}$ = 27 MHz / 64 ≈ 422 kHz.

Df > 0 means that the mixer frequency is larger than the carrier frequency. For example, SC1_DC = +390777 means Df = 10 kHz (mixer frequency is 10 kHz larger than carrier frequency).
A settling time for the DC filter of at least 200 ms is recommended. Ensure the controller allows this time after restart before reading the DC value and adjusting the mixer frequency.

See Section 3.9.8.8 "Noise detection"and Section 3.9.8.9 "Muting all DEMDEC outputs" for how to detect a valid sound carrier.

Automatic signal switching and routing: The switching task takes care of FM dematrix controlling, level adjusts, (un-)mute, automute, and routing the correct signals to the outputs, including management of the SRC configurations and SAP detection. Internally, different switch routines are active depending on the standard type.

Depending on the current standard, the switching task has to evaluate different status information and handles the internal signal switches (see Figure 73 and Figure 74 accordingly).

Table 138 lists all possible signals on the DEC and MONO outputs depending on the type of the current standard and ident / detection conditions, but without considering the SRC limitations explained in Section "SRC constraints". The SAP detection is described in section Section "SAP detection".

Table 138: DEMDEC output signals (not considering SRC restriction)

standard type	signal on "DEC" output		signal on "MONO" output
	stereo / dual detected and no automute	no stereo / dual detection or automute	
mono (no stereo standard found)	SC1 output	SC1 output	SC1 output
A2	L/A, R/B (output of FM dematrix)	SC1 output	SC1 output
BTSC, FM RADIO	L, R (output of FM dematrix)	mono (baseband)	mono (baseband)
EIAJ	L/A, R/B (output of FM dematrix)	mono (baseband)	mono (baseband)
NICAM	L/A, R/B (output of NICAM decoder)	mono (analog sound carrier; internal FM or external AM demodulator)	mono (analog sound carrier; internal FM or external AM demodulator)

Overmodulation Adaptation

The overmodulation adaptation (short OVMADAPT) feature is active while the corresponding variable OVMADAPT in the DDEPR is set to 1 in ASD or SSS mode. It measures the peak deviation of the (first) sound carrier before the DC notch filter and de-emphasis and feeds this values through a kind of "quasi peak detector" with an infinite decay time, this means that the detected value does not decay at all. The FILTBW variable in the DDEPR is not effective, but depending on the detected quasi-peak value, the FM filter bandwidth is chosen, starting with "narrow", then "medium", "wide" and "extra wide" (high deviation mode). However, if M BTSC or FM Radio is active, the filter bandwidth is overruled with "wide" because a constant setting is required to obtain an constant level of the subcarrier. The high deviation mode is not used while an A2 standard is active as only a single demodulator channel is available in this mode.

Secondly, OVMADAPT reduces the internal scaling coefficient for the analog sound channels if the output signal level (derived from the detected peak level without considering the de-emphasis) exceeds the threshold selected by the OVMTHR variable. This can be 3, 6, 9, or 12 dB above the nominal output level of -15 dBFS. +6 dBr = -9 dBFS is recommended.

Both the filter selection and the level reduction are not changed if the FM deviation remains below the thresholds or tends towards zero (i.e. no decay, "one-way road") in order to avoid small clicks which occur when switching the filter bandwidth, and to avoid varying output levels.

All internal variables of the OVMADAPT task are reset if the OVMADAPT bit is set to 0 and also at a "restart".

A critical parameter is the attack time constant of the quasi peak detector filter. It has been aligned by listening tests with overmodulated sound carriers.

The current scaling coefficient and filter settings can be read from the INF_OVMADAPT_REG register.

Table 139: Overmodulation adaptation status register (INF_OVMADAPT_REG, DD28)

Variable Name	No of bits	Bit index	Description
FILTBW_STAT	02	[1..0]	indicates internal FM/AM filter bandwidth (FILTBW) 0 = narrow 1 = extra wide 2 = medium 3 = wide
RSD_11To2	10	[11..2]	reserved
OVM_SCALE_STAT	12	[23..12]	current scaling factor for OVMADAPT (off: 1024 = 0 dB)

3.9.8.12 Other details

Power-up state: After power-up or reset, the DDEP software is in ASD mode (EPMODE=0) but in the same state as after a failed standard search, i.e. outputs are muted and all background routines are idle. The DDEPR is written and a "restart" performed as explained above, or EPMODE can be set to 3 to use the expert mode. In this case, the expert mode registers are written afterwards.

Control Interface: The hardware of the control interface is chip dependent. From the DSP's view, the type of hardware interface (PI bus, PCI bus, I^{2}C,...) makes (almost) no difference: all control accesses by the control interface to the DSP (X or Y) memories are only granted at the execution of a dedicated instruction called "MPI". This is necessary because the DSP needs the full bandwidth of the memories, and the DSP is not forced to wait for an external memory access to complete, as this may not allow the DSP to meet the real-time signal processing requirements.

At the MPI instruction, a read or write memory access by the control interface may be performed. If this has happened, the address which was accessed is available to the DSP software via a special hardware register, i.e. the software knows which register was accessed. This information, plus the actual register contents from XRAM, are put into a FIFO buffer. The issuing of MPI instructions and filling of the FIFO takes place within the 53 kHz thread; if the FIFO buffer is full (currently max. 12 entries), no MPI instruction is executed such that control accesses are disabled until at least one FIFO location becomes vacant. At every second 32 kHz cycle, if at least one item is in the FIFO, the oldest item is read out and the corresponding register handling routine is invoked. This routine, in case of a control register, decodes the register contents (which was stored in the FIFO and may differ from the actual value in the XRAM register!) and performs the required actions, or (in case of a status register) implements the defined behavior of a few status variables (after read, clear INITSTAT bit in DD01; reset simple peak detector in DD04; special protocol for NICAM additional data in DD03).

Remark: Every other 32 kHz time slot is occupied by a "background" control routine, which are to be executed at a constant rate.

By storing both register address and contents in a FIFO, all control words sent by the controller are decoded and executed in the correct order. However the control software should consider that status data is read a certain time after changing settings. With respect to DDEP, this is not a critical issue and this report should contain sufficient information about such delays, for example see the channel switch procedure (for expert mode, things are a lot more complex). Likewise, it should be considered that the average control data bandwidth is clearly smaller than the peak rate (53 kWords/s) because the DSP needs time to process the control data as explained.

If the PI bus control interface is to be used together with an interrupt-driven scheme, i.e. an interrupt is triggered each time a control access has completed, a resulting peak interrupt rate of 53 kHz (≈ 1/19µs) is considered too high for a typical microcontroller. Therefore, the MPI instruction execution rate can be reduced by means of the MPI_DIV control variable in the MPI_CONTROL_REG register (DD25). The actual MPI rate is 53 kHz divided by MPI_DIV (a 0 behaves like a 1) with a maximum value of 255, i.e. the smallest possible MPI rate is approx. 207 Hz ≈ 1 / 5 ms.

System initialization: Do not initialize the DEMDEC DSP until the clock system of the IC is functioning, i.e. the DSP clock and the audio clocks are present. The SSIF signal input (e.g. from I^{2}D link) does not need to be a valid sound IF for the initialization, but it should be at the DDEP restart as described in section Section “Channel switch procedure”.

For resuming after a power-saving mode, perform a DSP hardware reset and a new initialization. These control registers are chip dependent and are not in the scope of this manual.

The DSP needs up to 100 µs after reset before it can process control accesses.

Refreshing control registers: Control registers can be refreshed in any order. Note that some changes only take effect when the DDEPR is written, or if a DDEP restart is triggered. However, due to the dynamic properties of the DDEPR, refreshing cannot prevent the consequences of certain bit errors on the I^{2}C (e.g. if a bit error causes a wrong standard selection at the restart point of time). Therefore, refreshing does not yield a 100% certainty against I^{2}C bit errors. Higher safety can be achieved by verifying each write access by reading back the data, but this is not common practice.

Mode transitions

1. ASD to SSS mode: the search routine is disabled, everything else remains unchanged. This may be used to stop the search procedures and hold the currently active standard.
2. SSS to ASD mode: the last selected standard remains active, search is not active until "restart".
3. DDEP mode to expert mode: all DDEP-internal variables are reset, but the expert mode register contents are undefined and require writing afterwards to initialize the hardware and software.
4. Expert to DDEP mode: settings from expert mode remain effective. Initialize DDEP via a "restart".

Hardware versioning: Some products may contain special versions of the DDEP software in which certain features can be disabled by means of hardware input signals of the DSP, which are typically generated by bond options, flash / OTP memory contents or similar. For example, for commercial reasons a certain sound standard might not be enabled. In such a case, the underlying DEMDEC software usually mutes the stereo or dual signal such that only the mono sound is available (even in expert mode), and DDEP likewise indicates a mono reception via the GST and GDU flags, although the stereo/dual identification information is still available. For example, if all FM A2 standards are disabled by versioning, these signals are only reproduced as mono, GST and GDU are always 0, even if the AST (FM A2 stereo ident) or ADU (FM A2 dual) flag is 1. The standard detection is not limited by versioning.

In AVIP, only the dbx noise reduction is controlled by a versioning. If dbx is not enabled, the dbx decoder is replaced by a simplified algorithm, resulting in a limited stereo channel separation.

3.9.8.13 Register map of DEMDEC DSP

Table 140: Overview of DEMDEC DSP (high latency registers)

W/R	XMEM ADDR. decimal	Register Name	Cluster Coarse	Detailed Description
R	1	INF_MAIN_STATUS_REG	DEMDEC STATUS	DEMDEC main status register. Some variables are active only in DDEP mode.
R	2	INF_NICAM_STATUS_REG	DEMDEC STATUS	NICAM status register. status signals from decoder hardware. active in DDEP and expert mode.
R	3	INF_NICAM_ADD_REG	DEMDEC STATUS	NICAM additional data register. updated every NICAM frame (1Khz rate).
R	4	INF_MONLEVEL_REG	DEMDEC STATUS	monitor level register. updated every 35 Khz period.

Table 140: Overview of DEMDEC DSP (high latency registers) *…continued*

W/R	XMEM ADDR. decimal	Register Name	Cluster Coarse	Detailed Description
R	5	INF_MPX_LEVEL_REG	DEMDEC STATUS	MPX (BTSC / FM radio) pilot level register. from MPX demodulator hardware. close to 0 if not locked.
R	6	INF_DC1_REG	DEMDEC STATUS	DC offset of sound carrier 1 after FM demodulation
R	7	INF_SUBMAGN_REG	DEMDEC STATUS	FM subchannel magnitude register. lowpass filtered AM output of subchannel FM/AM demodulator.
R	8	INF_NOISELEVEL_REG	DEMDEC STATUS	noise detector output and status register
R	9	INF_SRCSTATUS_REG	DEMDEC STATUS	SRC status information
W/R	10	DEM_HWCFG_REG	DEMDEC EXPERT MODE	main hardware configuration register (FM/AM demodulator, IDENT)
W/R	11	DEM_CA1_REG	DEMDEC EXPERT MODE	sound carrier 1 (mixer 1) frequency register
W/R	12	DEM_CA2_REG	DEMDEC EXPERT MODE	sound carrier 2 (mixer 2) frequency register
W/R	13	DEM_MPXCFG_REG	DEMDEC EXPERT MODE	MPX demodulator configuration register
W/R	14	DEM_FMSUBCFG_REG	DEMDEC EXPERT MODE	FM subchannel demodulator and noise detector configuration register
W/R	15	DEM_SWCFG_REG	DEMDEC EXPERT MODE	software signal processing before SRC (filter, de-emphasis, decompression etc.) configuration and SRC control
W/R	16	DEM_OUT_CFG_REG	DEMDEC EXPERT MODE	output processing (SRC postprocessing) and monitor control register
W/R	17	MAGDET_THR_REG	DEMDEC EASY PROGRAMMING	magnitude detection thresholds for DDEP mode

Table 140: Overview of DEMDEC DSP (high latency registers) *…continued*

W/R	XMEM ADDR. decimal	Register Name	Cluster Coarse	Detailed Description
W/R	18	NMUTE_FMA2_SAP_REG	DEMDEC EASY PROGRAMMING	noise detector configuration for automute of SAP and SC2 in FM A2 standards (DDEP only)
W/R	19	NMUTE_MPX_REG	DEMDEC EASY PROGRAMMING	noise detector hysteresis configuration for subchannel automute in BTSC and FM RADIO standards (DDEP only)
W/R	20	NMUTE_EIAJ_REG	DEMDEC EASY PROGRAMMING	noise detector hysteresis configuration for subchannel automute in the EIAJ standard (DDEP only)
W/R	21	NICAM_CFG_REG	DEMDEC EASY PROGRAMMING	NICAM configuration register
W/R	22	DDEP_CONTROL_REG	DEMDEC EASY PROGRAMMING	DEMDEC Easy Programming register (DDEPR)
W/R	23	DEM_LEVELADJUST_REG	DEMDEC EASY PROGRAMMING	DEMDEC level adjust and scaling register
W/R	24	DEM_ADC_SEL_REG	DEMDEC EASY PROGRAMMING	ADC channel selection and mute
W/R	25	MPI_CONTROL_REG	DEMDEC EASY PROGRAMMING	MPI (microprocessor interface) control
R	26	INF_REVID_DD_REG	DEMDEC STATUS	Revision ID of Embedded DSP Code
R	27	INF_CPULOAD_REG	DEMDEC STATUS	CPU load indicator register
R	28	INF_OVMADAPT_REG	DEMDEC STATUS	Overmodulation status register
W/R	29	DDEP_OPTIONS1_REG	DEMDEC EASY PROGRAMMING	DDEP options register no.1
W/R	30	DD_OPTIONS2_REG	DEMDEC EASY PROGRAMMING	DEMDEC options register no. 2

3.9.9 AUDIO-DSP

3.9.9.1 Functional overview

Functional overview of Audio-DSP for AVIP is given in Figure 79. The processing of the loudspeaker channels (MAIN, SUB, C, Ls, Rs), the headphone channel (AUX1/HP) and channels AUX2 to AUX6 is nested between the Digital Input Crossbar and the Digital Output Crossbar.

The following sources are inputs to the Digital Input Crossbar:

- DEC, left and right inputs from the demodulator / decoder (DEMDEC),
- Mono, mono input from the demodulator / decoder (DEMDEC),
- SAP, Secondary Audio Program input from the demodulator / decoder (DEMDEC),
- ADC (L, R), two channel input from the demodulator / decoder (DEMDEC),
- PIPMONO, mono input from the demodulator / decoder (DEMDEC),
- I^{2}S 1 to 6 IN, from the I^{2}S inputs at AVIP.

All these signals pass the Level Adjust function before entering the crossbar. An adjustment is needed to level the source signals if they deviate from nominal setting.

A special source is the feedback from the Digital Output Crossbar. This OUTCOPY 1, 2 path offers the possibility e.g. to connect the Audio Monitor to one of the inputs of the Digital Output Crossbar.

The Noise/Silence Generator is another selectable source, needed for Dolby® Pro Logic® II and Dolby® Digital speaker trim. This noise source is compliant to the Dolby requirements for noise sequencer.

The Digital Input Crossbar provides source select and also matrixing for the channels MAIN (L, R), AUX1 to AUX6 and the Audio Monitor (AUDMON). But only source select for Center (C), Left surround (Ls), Right surround (Rs) and low-frequency effects (LFE), because these are already decoded multi-channel signals.

Although the selectors are all of the same type not all facilities are used in normal applications of the AVIP e.g. the center and surround selectors can be permanently connected to the Noise/Silence Generator. On the other hand there is normally no need to switch the AUX channels to Noise/Silence.

The setting of the Digital Matrix depends on the type of input signal. The different signal types: stereo, dual language or mono, may be detected by the identification circuit located in the demodulator/decoder block. So the switching can be done dependent on the identification. For dual language the preference for language A or B can be set if automatrix is selected. In this case the matrix provides the language according to the preference selected by the end-user.

The signal type is unknown, if an external audio source (ADC, I^{2}S) is chosen. In this case the end-user needs to have a selection facility. It should include the choice between stereo (AB, being L,R for stereo or both languages e.g. passed to SCART/cinch for recording), mono (from stereo by (A+B)/2, also called forced mono), sound A or B and a swap (BA) for stereo if the source has interchanged L and R.

The processing channels are dedicated to loudspeakers (MAIN, SUB, C, Ls, Rs), to headphones (AUX1/HP), or to I2S1 to I2S6 out, DAC1 and DAC2 (AUX2 to AUX6). AUX2 to AUX6 offer only volume plus trim control (Vol/Trim) and Soft Mute (SM), whereas AUX1/HP has additional bass / treble control (Ba/Tr) and a mixer to feed in a beeper signal (e.g. as prompt from remote control commands).

The loudspeaker channels of the AVIP can process mono, normal stereo, already decoded multi-channel or Dolby® Pro Logic® II encoded signals. The provided functions can be used according to these signal types. Some of them are dedicated to specific modes leading to constraints. E.g., Pseudo Hall/Matrix ((L+R)/2; (L-R)/2) can only be used with stereo or mono signals, VDS only with DPL II decoded signals, and VDD applies to 5 externally decoded Dolby® Digital channels. IMono or IStereo can be selected, but both have to be switched off to meet the Dolby requirements if DPL II is active.

Other selections depend on the speaker system, whether the set is equipped with 6 speakers (L, R, SUB, C, Ls, Rs) or subsets of this. For example, the surround speakers may be disconnected, or only the 2 main speakers (L, R) are in use. Another important issue is the size and bandwidth of the used loudspeakers.

Some of the functions are set by 'SNDMODE'-register control according to the Sound Mode Table (see Table 141). The remaining functions need to be controlled by individual settings.

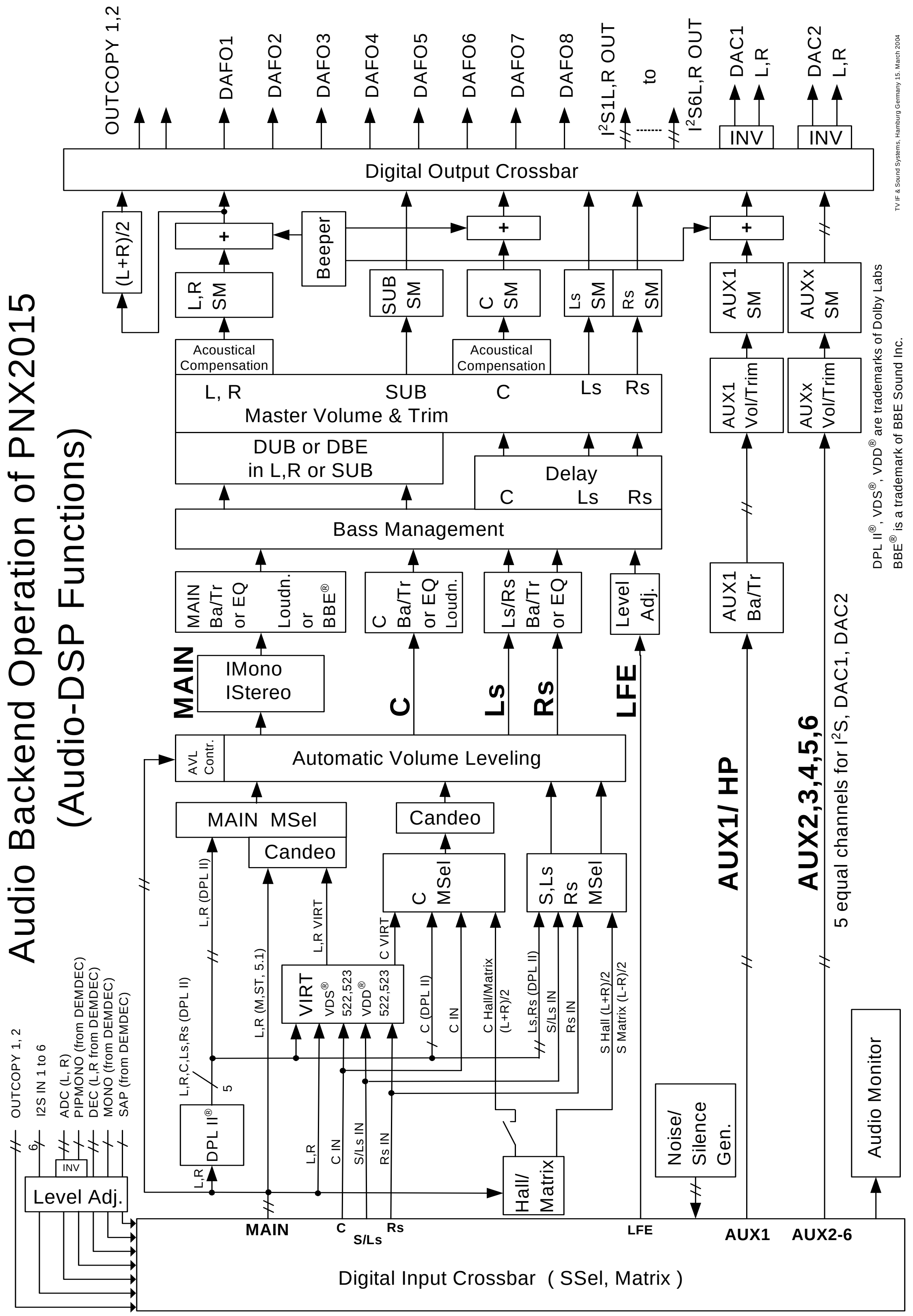


Fig 79. Audio Backend Operation of AVIP

3.9.9.2 Sound modes of the loudspeaker channels

Sound modes are defined in the following table:

Table 141: Sound mode

Mode	Description
M/ST	Mono/Stereo in case of mono or stereo source signals
M/ST Hall	Mono/Stereo with pseudo Hall in case of mono or stereo source signals[1]
M/ST Matrix	Mono/Stereo with pseudo Matrix in case of mono or stereo source signals[1]
DPLII Wide Center	DPLII Wide Center in case of DPL encoded source signals (large center speaker)
DPLII Phantom Center	DPL II Phantom Center in case of DPL encoded source signals (small center speaker)
DPLII 3 Stereo	DPLII 3 Stereo in case of DPL encoded source signals (no surround speakers available)
VDS523	DPLII+VDS(523) Virtual Dolby Surround 523 in case of DPL encoded source signals
VDS522	DPLII+VDS(522) Virtual Dolby Surround 522 in case of DPL encoded source signals
VDD523	VDD(523) Virtual Dolby Digital 523 in case of externally decoded AC3 source signals
VDD522	VDD(522) Virtual Dolby Digital 522 in case of externally decoded AC3 source signals
Multi-Ch. Post Mode	Multi-channel post processing mode[2]
NSEQ	Multi-channel speaker level Trim (noise sequencing)

[1] In case that Hall or Matrix mode is active the extra 'CENTERMUTE-bit' gives the possibility to switch off the center signal. This is needed because it is not sufficient to mute the center signal by use of the Soft Mute (located directly in front of the Output Crossbar). The Bass Management would simply redirect the bass signals of the center, and therefore give an extra bass boost in left, right or subwoofer channel.

[2] The Multi-channel Post Processing mode gives the option to use the PNX2015 as sound processor IC for multi channel input signals.

The Sound Mode sets explicitly the functions DPLII, VDS, VDD, Main MSel, C MSel, S MSel, Pseudo Hall/Matrix and provides a specific setting for noise sequencing.

Two steps are needed to change from one sound mode to another. Select the new sound mode at first, and then toggle the execution bit 'EXEMODTAB' from '0' to '1' to activate the new mode. To make this happen two write accesses are needed.

1. Set EXEMODTAB-bit and set new SNDMOD-bits.
2. Clear EXEMODTAB-bit.

Therefore it is always necessary to write twice to change the sound mode once! This control mechanism is built in to make the system robust against regular register updates, which are processed every few seconds.

Table 142 shows the setting for the loudspeaker channels dependent on the selected "Sound Mode".

Table 142: Sound Modes

Sound mode	DPLII	Noise/silence gen.	VDSVDD	MSEL.main	Pseudo hall matrix	MSEL. centre	MSEL. surround
M/ST	not active	active 1)	not active	connected to L,R M/ST	not active	connected to CIN	connected to LsIN, RsIN
M/ST Hall	not active	not active	not active	connected to L,R M/ST	Pseudo Hall active	connected to (L+R)/2	connected to (L+R)/2
M/ST MATRIX	not active	not active	not active	connected to L,R M/ST	Pseudo Matrix active	connected to (L+R)/2	connected to (L-R)/2

Table 142: Sound Modes *…continued*

Sound mode	DPLII	Noise/silence gen.	VDSVDD	MSEL.main	Pseudo hall matrix	MSEL. centre	MSEL. surround
DPLII N/W	DPLII N/W	not active	not active	connected to L,R DPLII	not active	connected to C (DPLII)	connected to Ls, Rs (DPLII)
DPLII PH	DPLII PH	not active	not active	connected to L,R DPLII	not active	connected to silence 2)	connected to Ls, Rs (DPLII)
DPLII 3ST	DPLII 3ST	not active	not active	connected to L,R DPLII	not active	connected to C (DPLII)	connected to silence 2)
VDS523	DPLII N/W	not active	VDS523	connected to L,R VIRT	not active	connected to C VIR	connected to silence 2)
VDS522	DPLII N/W	not active	VDS522	connected to L,R VIRT	not active	connected to silence 2)	connected to silence 2)
VDD523	Not active	Not active	VDD523	connected to L,R VIRT	not active	connected toC VIR	connected to silence 2)
VDD522	Not active	Not active	VDD522	connected to L,R VIRT	not active	connected to silence 2)	connected to silence 2)
M-CH PP	Not active	active 1)	not active	L,R M/ST	not active	connected to CIN	connected to LsIN, RsIN
DPLII NSEQ	not active	active 1)	not active	connected to L,R M/ST	not active	connected to CIN	connected to LsIN, RsIN

[1] The Noise/Silence Generator is active, MSel Center is connected to CIN and MSel Surround is connected to LsIN, RsIN. This is done to give the setmaker the facility to build a noise sequencer application of his choice with the M/ST sound mode.

[2] (silence) means that the signal carries digital silence, no audio signal nor noise.

3.9.9.3 Remarks to function control

Automatic Volume Levelling (AVL): The AVL reduces the audio input signal in the MAIN channels (L, R, C, Ls, Rs) to a selectable maximum output level, if it exceeds this level at the input of the stage.

A detector creates a control signal from Lt and Rt (input of the DPL II decoder) or from the L, R output of the virtualizer. The AVL provides a short attack time of 4ms and selectable decay times of 20ms, 2s, 4s, 8s and 16s. A weighting filter can be chosen in the control signal generation. The advantage is that bass signals and high frequency components have less impact on control.

Also the AVL level (maximum output level) can be set and the AVL can be switched OFF.

Virtual Dolby® Surround (VDS): Virtual Dolby® Surround gives a surround sound impression with use of only two speakers (VDS522) or three speakers (VDS523). Inputs to VDS are the L, R, C, Ls and Rs outputs of the DPL II decoder. The surround signals Ls and Rs are virtualized and redirected in both cases to the left and right channel, whereas the center signal is only redirected to L, R when VDS522 is selected. In VDS523 mode a separate center channel is provided. But digital silence is connected to the C output of the virtualizer in VDS522 mode. Ls and Rs carry digital silence in both modes VDS522 and VDS523.

Figure 80 shows the left and right output signal of the VDS module. The signals C (center), Rs (right surround) and Ls (left surround) are connected to digital silence. The main left and right inputs are connected to the source signal, which in this case is: L = -R with 20dB headroom. VDS522 is selected as sound mode, with a mix level set to 80% and DPLII set to Movie mode.

Remark: In case of VDS522 is selected, then DPLII has to be set to Movie mode to produce a de-correlation of mono surround signals.

Refer to Figure 80 for characteristics of the VDS module.

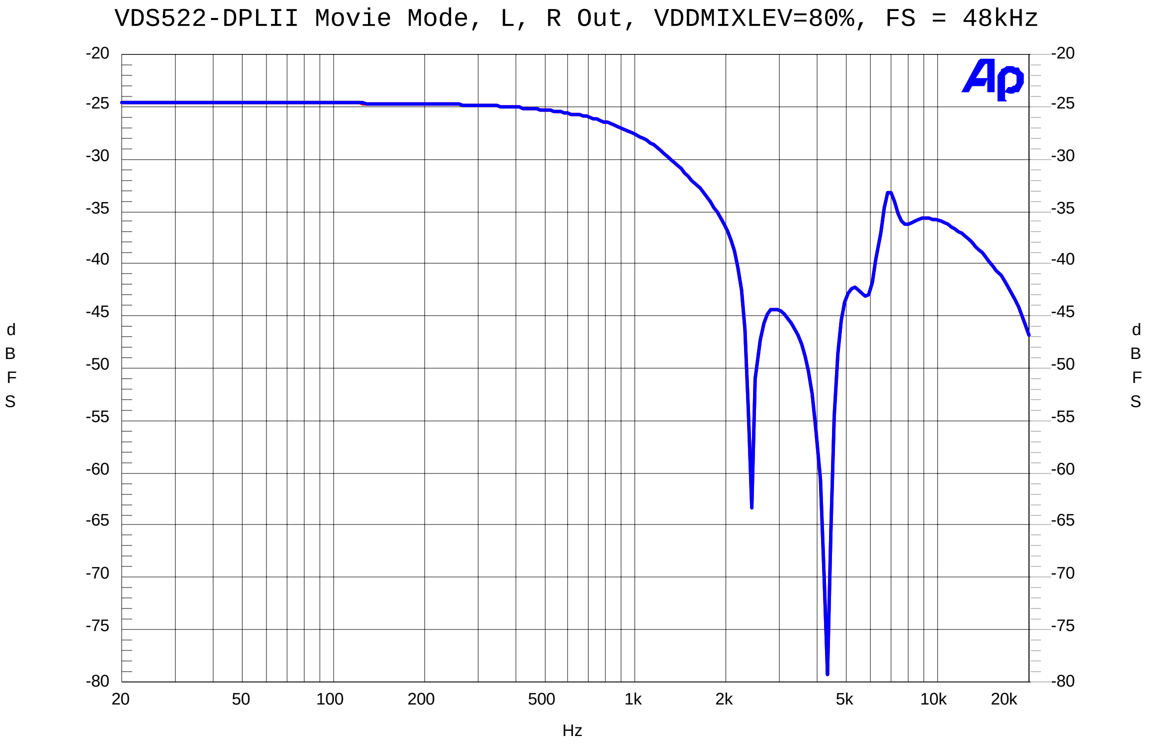


Fig 80.Virtual Dolby® surround left and right output signal

Virtual Dolby® Digital (VDD): Virtual Dolby Digital gives a surround sound impression with use of only two speakers (VDD522) or three speakers (VDD523). Inputs to VDD are the L, R, C, Ls and Rs inputs of the Digital Input Crossbar. The surround signals Ls and Rs are virtualized and redirected in both cases to the left and right channel, whereas the center signal is redirected to L, R only in case of VDD522 is selected. In VDD523 mode a separate center channel is provided. But digital silence is connected to the C output of the virtualizer in VDD522 mode. Ls and Rs carry digital silence in 522 and 523 mode.

Activate the 'VDDMONSUR' switch to get some de-correlation via phase inversion, this is necessary for mono surround signals.

Figure 81 shows the left and right output signal of the VDD module. The input signals MAIN (L and R), as well as Center (CIN) and Right Surround (Rs) are connected to digital silence. The left surround (Ls) input is connected to a source signal with 15dB headroom. The diagram is scaled to -15dBFS = 0dBr. VDD522 is selected as sound mode and the mix level is set to 80%.

Refer to Figure 81 for characteristics of the VDD module.

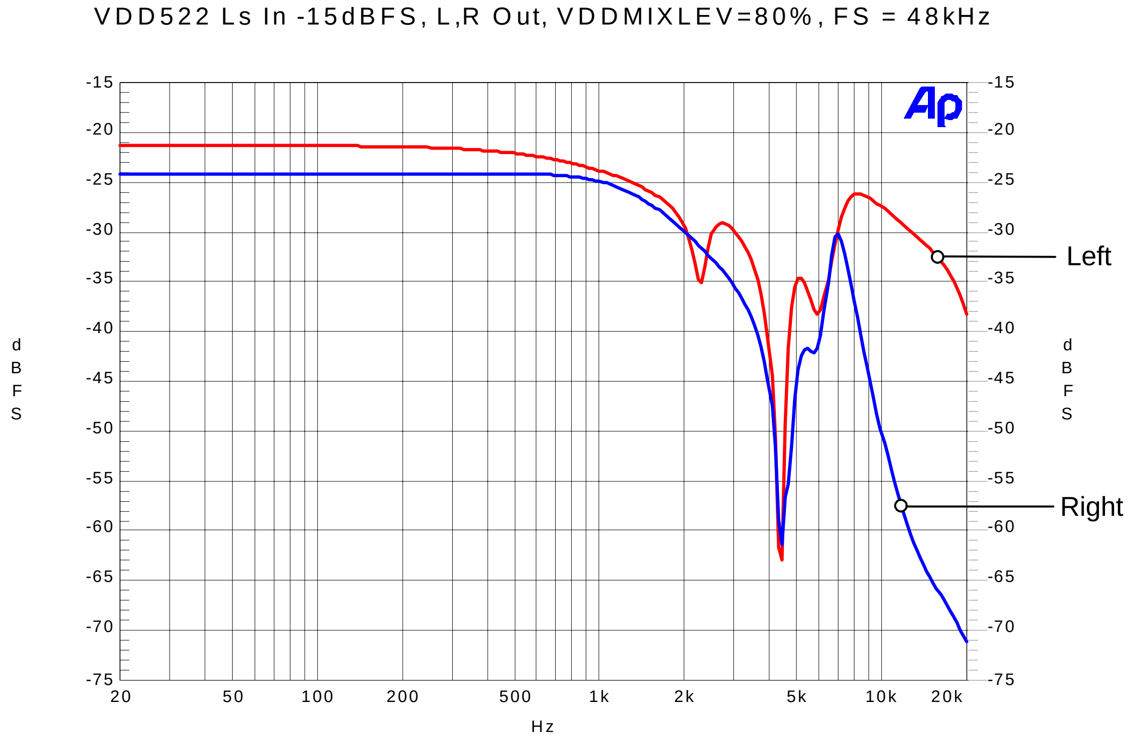


Fig 81.Virtual Dolby® digital left and right output signal

Noise Sequencer for DPLII: The set maker has to provide a program to build a noise sequencer with the components available in the AVIP. The noise sequencer is to meet the requirements as specified in the Dolby Licensee Information Manual: Dolby® Digital Consumer Decoder. All channels L, C, R, Ls, Rs can be connected to the noise source, then the noise can be cycled by use of the Soft Mute (SM). It is recommended to mute the subwoofer output, while the noise sequence is active.

Dolby® Pro Logic® II Function (DPLII): The DPLII decodes the incoming Dolby encoded Left (Lt) and Right (Rt) channels into the 5 output channels: Left (L), Right (R), Center (C), Left surround (Ls) and Right surround (Rs).

This implemented DPLII decoder is compliant to all requirements mentioned within the 'Dolby Licensee Information Manual: Dolby Pro Logic II '.

The selection of DPLII normal/wide, Phantom Center, or 3 Stereo mode delivers output signals according to the Dolby requirements. Outputs which are not used in a specific mode carry Digital Silence '-'.

The Surround delay is adjustable between 0ms and 25ms, depending on the selected DPLII decoder mode. The DPLII function provides 5 different decoder modes: Movie, Music, Virtual , Pro Logic Emulation and Matrix.

Movie Mode

The Movie mode is very similar to the Pro Logic mode. The main difference is, that Pro Logic has a 7 kHz surround filter and a mono surround output, while the Movie mode has no surround filter and stereo surround outputs.

Music Mode

The Music mode offers users some flexibility to control the end results to their own taste. Music mode is recommended by Dolby Labs. as the standard mode for auto-sound music systems (without video). The Music mode has to be identified as the "Music" version of Pro Logic II, to distinguish it from the Movie mode.

Only the Music mode offers additional control of the following features:

Dimension Control:

Allows the user to gradually adjust the sound field either towards the front or towards the rear. This can be useful to help achieve a more suitable balance from all the speakers with certain recordings.

Center Width Control:

Allows variable adjustment of the center image so it may be heard only from the center speaker, only from left/right speakers as a "Phantom" center image, or various combinations of all three front speakers. This control (used in a car application) allows to create a balanced left-center-right stage presentation for both the driver and the front passenger.

On the other hand it allows improved blending of the center and main speakers, or to control the sense of image width, or "weight".

Panorama:

Extends the front stereo image to include the surround speakers for an exciting "wraparound" effect with side wall imaging.

Virtual Mode

The Virtual mode is usually used when Pro Logic II is connected to a virtual surround process. However, there might be some virtualizer for which this mode does not produce the intended result. The Movie mode may give a better surround effect for those virtualizers.

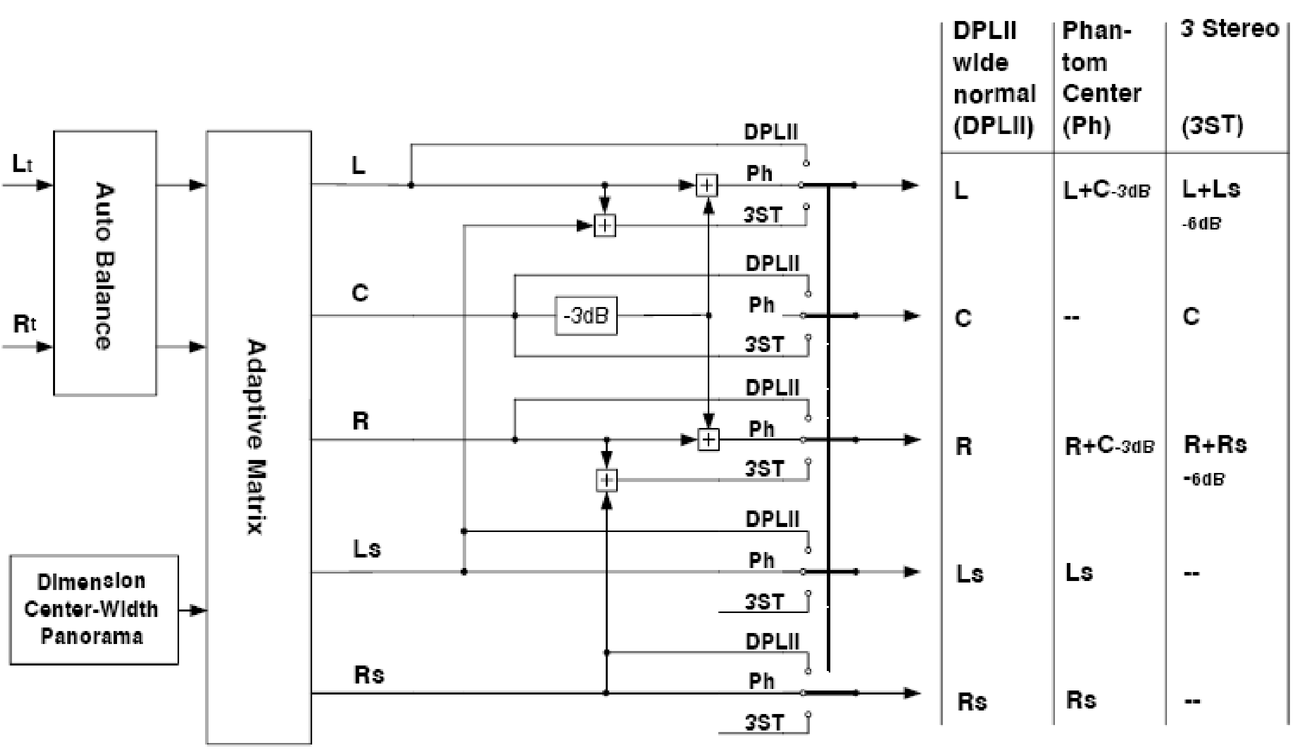
Pro Logic® Emulation Mode

The Pro Logic Emulation mode included in the DPLII technology package is as robust as the original Pro Logic decoding function without having to provide separate decoding circuitry in the product.

Matrix Mode

The Matrix mode is the same as the Music mode except that the directional enhancement is turned off. It may be used to enhance mono signals by making them seem "larger". The Matrix mode may also find use in auto systems, where the fluctuations from poor FM stereo reception can otherwise cause disturbing surround signals from a logic decoder.

Listening tests done at Dolby Laboratories during the Dolby Approval of the AVIP showed that for VDS522 and VDS523 sound mode (DPLII combined with the internal virtualizer) DPLII Movie mode is selected, with the "VDDMONSUR" switch set to ON.



DPLII wide normal (DPLII)	Phan- tom Center (Ph)	3 Stereo (3ST)
L	$L+C_{-3dB}$	$L+Ls_{-6dB}$
C	--	C
R	$R+C_{-3dB}$	$R+Rs_{-6dB}$
Ls	Ls	--
Rs	Rs	--

Fig 82.Dolby® Pro Logic® II Function

Bass / Treble: Bass and treble functions are implemented in all 5 main signal paths (L, R, C, Ls, Rs) and in the AUX1/HP. The user is able to attenuate or boost the bass and high frequency signals independently within a range of -16dB to +15dB. The external resolution (under user control) is defined to 1dB steps, whereas the internal resolution (not under user control, 1/32dB steps) is used to avoid 'pop noise'. The internally used 1/32dB per step leads to a maximum speed of amplitude change, which is defined to 15.625dB/s. The corner frequency of the bass function is fixed to 40 Hz and for the treble function fixed to 14 kHz.

In the three plots the following is shown:

- Figure 83 shows the bass and treble function with equal settings and steps of 5dB.
- Figure 84 shows the treble function within steps of 2dB. The bass control is set to flat.
- Figure 85 shows the bass function within steps of 2dB. The treble control is set to flat.

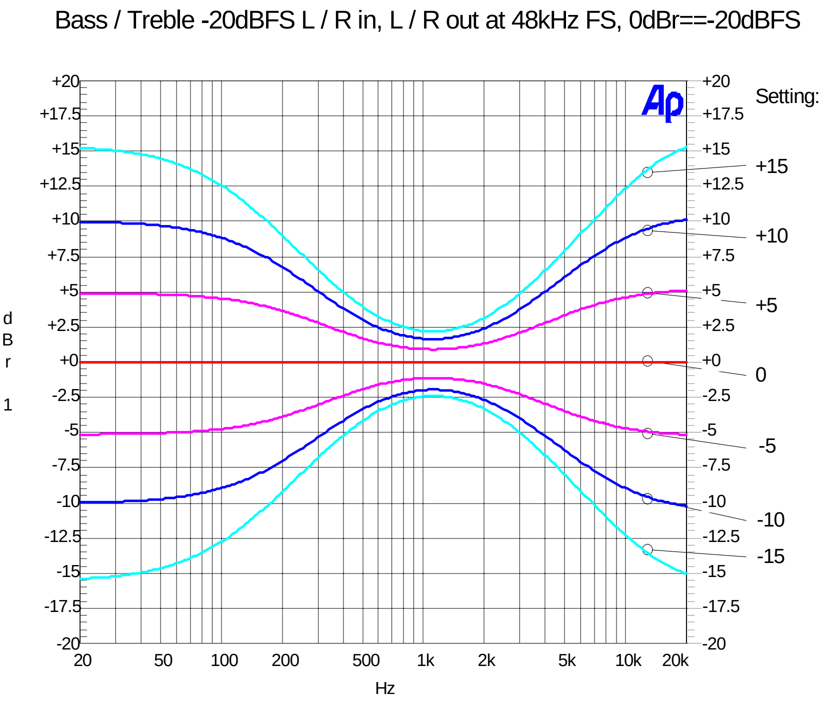


Fig 83.Bass / Treble Function with equal Settings and Steps of 5dB

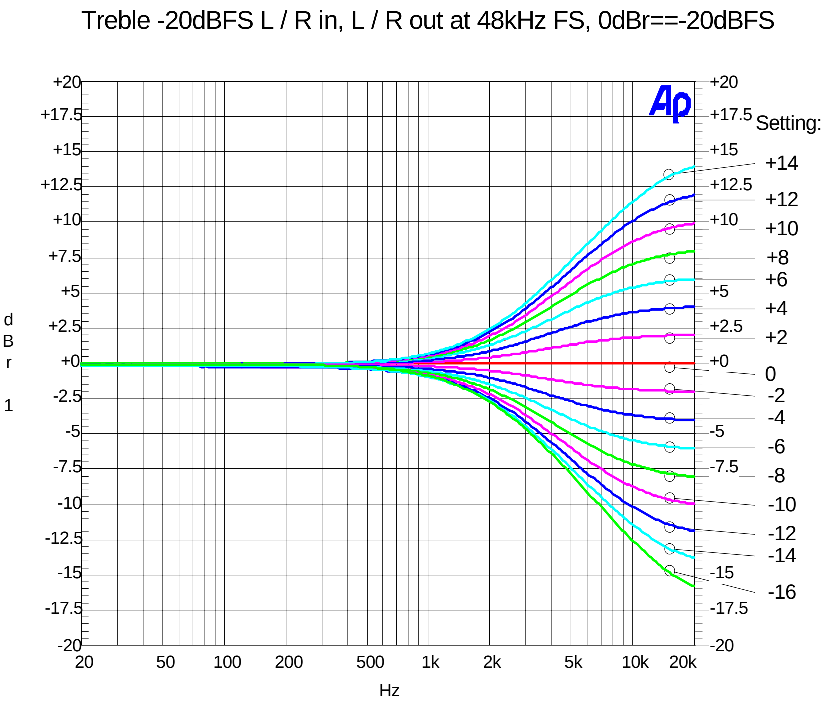


Fig 84.Treble Function within Steps of 2dB. The Bass Control is set to flat

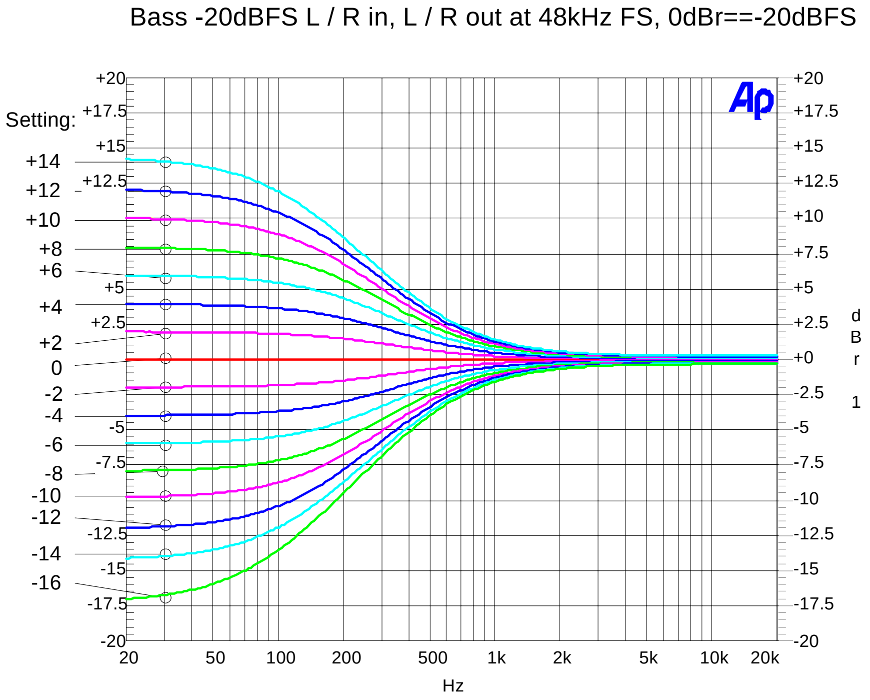


Fig 85.Bass Function within Steps of 2dB. The Treble Control is set to flat

Loudness: The human ear listening curves (Fletcher-Munson loudness contours) show, that the ears of a human are less sensitive for low and high frequencies at low sound pressure level (volume level). In general a loudness function can be used to compensate the human ear sensitivity loss at low volume levels.

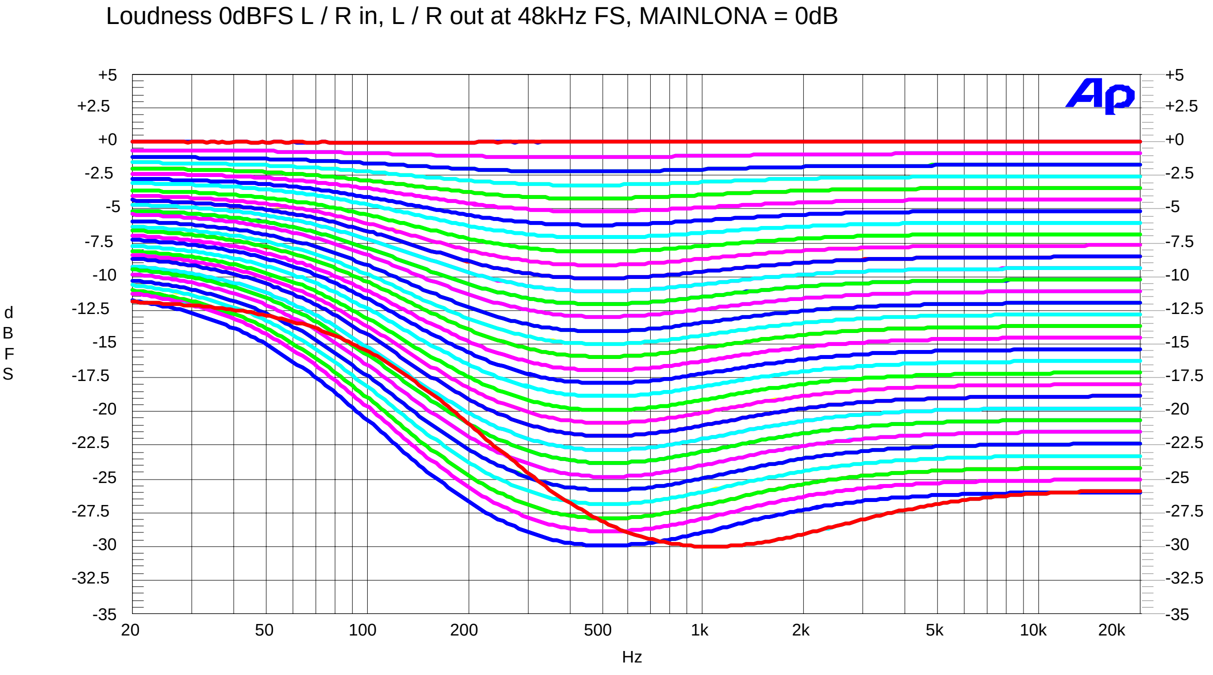
The supported sample rates are 32 kHz and 48 kHz. Sample rate 48 kHz is used in case of 44.1 kHz is required. This results in a 10% shift of the loudness curves.

Within a volume range of 30dB the loudness gain varies with the total gain value of the volume stage. The loudness curves are automatically adjusted to the input volume level, where the allowed input volume steps can be 1/8dB or even smaller to avoid step-noise.

At high volume the resulting loudness curve is flat, because there is no need to boost high and low frequencies at a high sound pressure level. The loudness boost becomes active, if the input volume is reduced below the (adjustable) no attack threshold. The resulting gain depends on the actual input volume level. Within a range of 30dB below the no attack threshold the loudness function gives an increasing boost of low and high frequencies. If the input volume is reduced more than 30dB below the no attack threshold level then the maximum loudness gain is reached and the loudness curve for 30dB remains active. The maximum loudness gain is +18dB at 20 Hz and + 4.5dB at 16 kHz.

The frequency where the gain is not affected by the loudness function is called no attack frequency. This no attack frequency can be adjusted to 500 Hz or 1 kHz. This results in different loudness curves, but the maximum gain at 20 Hz and 16 kHz remains the same. Figure 86 shows loudness curve measurements with a no attack frequency set to 500Hz, as well as one example curve where the no attack frequency is set to 1kHz.

As already mentioned, the no attack threshold of the input volume level (which gives a flat loudness curve) can be adjusted, to allow a flexible selection of the active loudness working range



For the MASTVOL range of 0 - -30dB (upper to lower) the no attack frequency (MAINLOCH) was set to 500Hz

For the last curve (red) at MASTVOL = -30dB the no attack frequency was set to 1kHz

Fig 86.Loudness Curves for a MASTVOL Range of 0 - -30dB (Step Width 1dB)

Incredible Stereo (IStereo): The IStereo module, also known as Robust Incredible Sound (RIS), is a Stereo Expander. The listener gets the impression of a sound reproduced by 2 virtual speakers, positioned at a larger distance (angle) than the actual speakers. So, the stereo image is expanded by this widening sound effect.

The only parameter controlled by the user is the Stereo Widening Parameter Alfa (a). A register named 'INSOEF' is reserved within the AVIP to allow user controlled access. Additionally this feature can be switched ON or OFF controlled by use of the 'INSOMO' register.

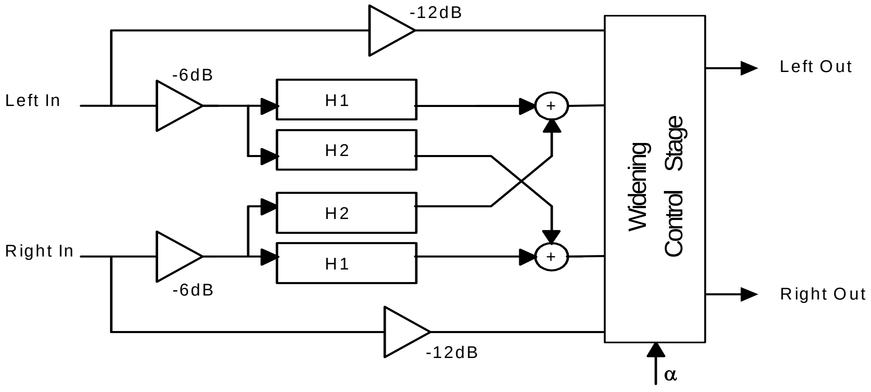


Fig 87. IStereo (RIS) Module

Figure 88 shows the IStereo output for a stereo signal with 15dB headroom and left, right with 0° phase [1)]. The second curve is measured under the same conditions, but with -180° phase between left and right [2)].

Remark: All curves are measured with a sample rate of 48kHz. The frequency responses shift with the sample rate.

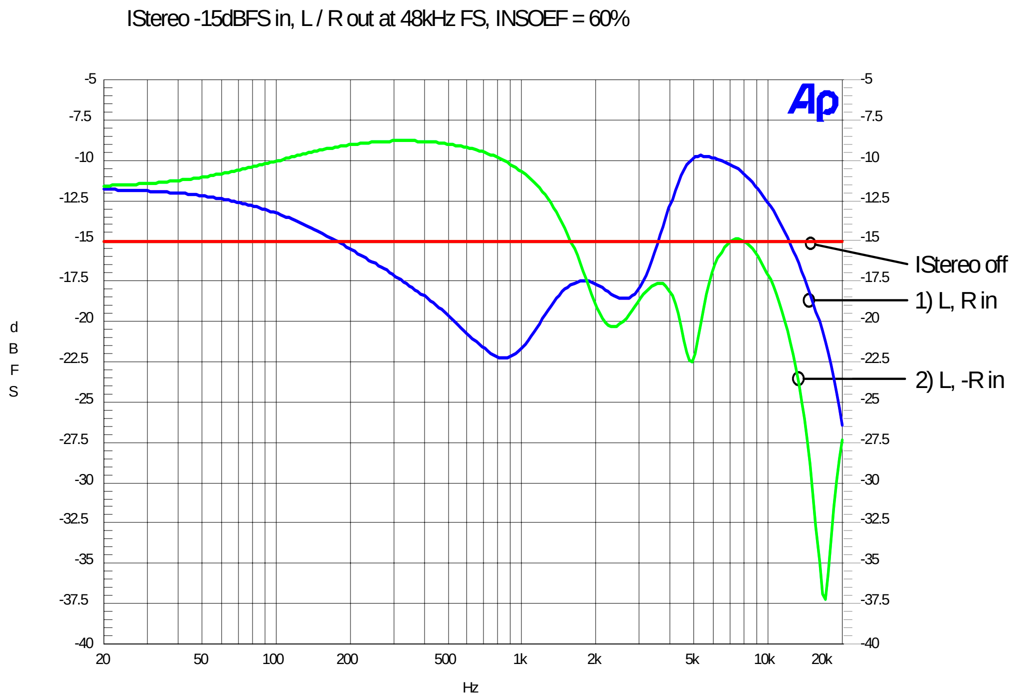


Fig 88. Left/Right Output Signal of IStereo Module (conditions 1 and 2)

Incredible Mono (IMono): The Incredible Mono module (IMono) generates two channels (a stereo signal: Left and Right) from one mono input signal. When the sound of the mono input signal is processed, the listener gets the impression that the sound is reproduced as pseudo stereo signal with incredible sound.

This module is implemented by a combination of two cascaded stages: the IMONO Decorrelator and the RIS module (IStereo). The RIS module is exactly the same module already described in the previous chapter [IStereo]. All data handling is done internally within AVIP. The user has the possibility to control the effect strength via the 'INSOEF' variable being shared between the IStereo and the IMono module. It can be switched ON or OFF, by use of the 'INSOMO' register. There is no user control for ScaleML/MR.

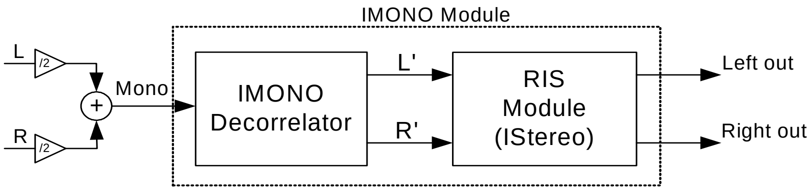


Fig 89. IMono Module

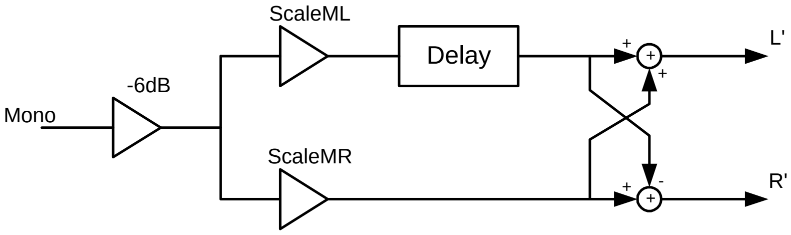


Fig 90. IMONO Decorrelator Module

Remark: Delay is a delay line of 4 samples.

In Figure 91 the output signal of the IMono module is shown. The input is connected to a source with 15dB headroom and the left, right output signal is measured at a sample rate of 48kHz.

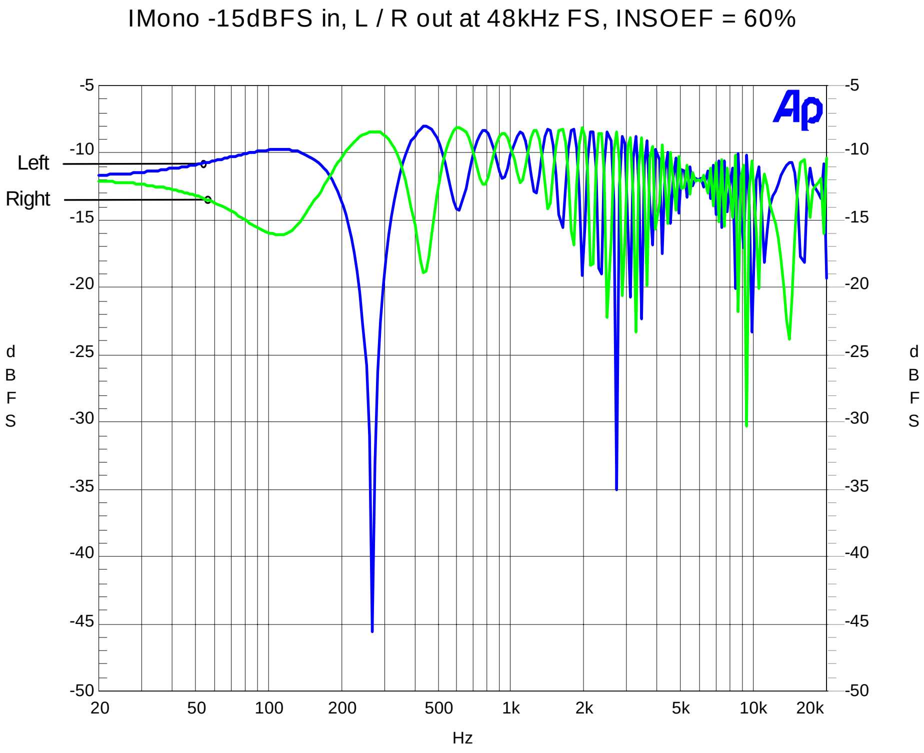


Fig 91. Left and Right Output Signal of the IMono Module

Dynamic Ultra Bass (DUB): In general the DUB function is used in small TV-Sets without full range speakers that cannot reproduce deep bass signals. The effect is caused by producing harmonics of the low frequency content. It gives the impression of deep bass reproduction although the fundamentals are missing. The level of harmonics added to the original signal is made dependent from the total signal level at the output. The dynamic behavior allows a strong amplification of the harmonics for small volume signals, but only small amplification for high volume signals. Maximum gain and the target output level can be set. The acoustical behavior of this feature has to be tuned to the TV internal loudspeaker set. Therefore a certain set of filter coefficients has to be found for each used TV-set. This is done by use of the loudspeaker characteristics as well as by listening tests. This coefficient set has to be loaded into the AVIP once after power on reset. During the coefficient download turn the DUB off, returning to the on condition when complete.

DUB is applied to the left and right speakers, or alternatively with different coefficients to the subwoofer signal to enhance the bass reproduction. When using DUB it is not possible to apply DBE.

The DUB coefficient RAM layout can be found in Section 3.9.9.5 “DUB/DBE Coefficients”

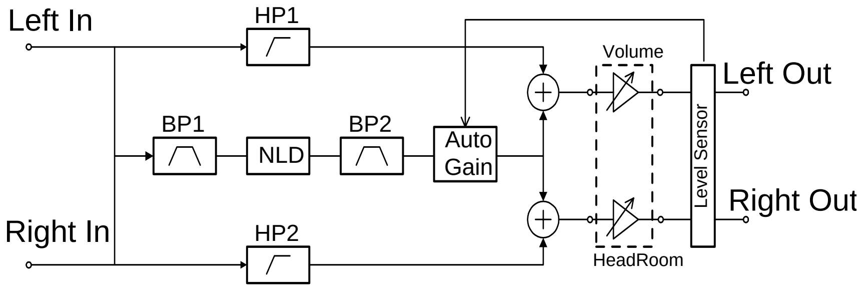


Fig 92. DUB Function

In the following the download procedure is shown how to write the coefficients into the DUB and DBE module.

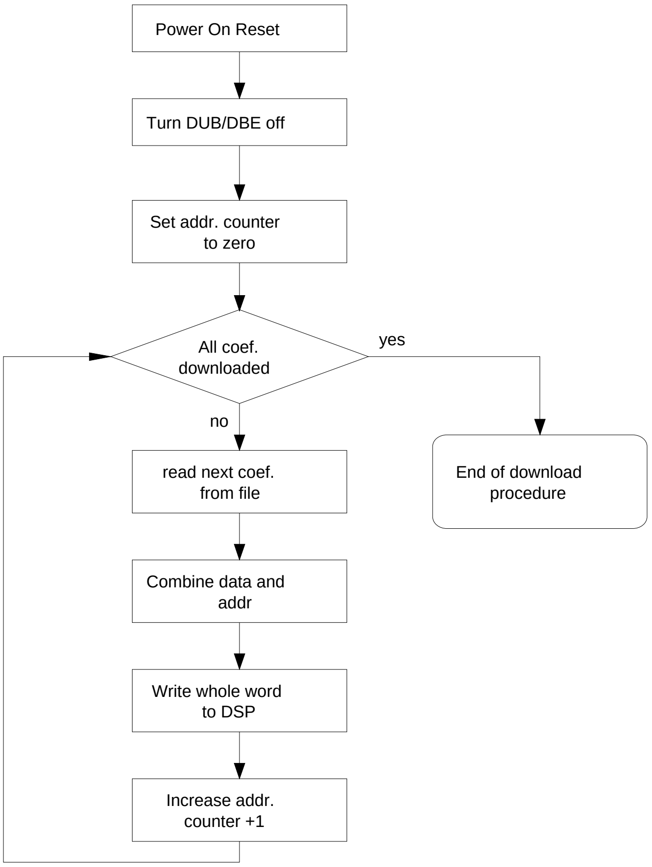

Power On Reset
Turn DUB/DBE off
Set addr. counter to zero
All coef. downloaded
yes
no
End of download procedure
read next coef. from file
Combine data and addr
Write whole word to DSP
Increase addr. counter +1

Fig 93. Download for DUB and DBE Coefficients

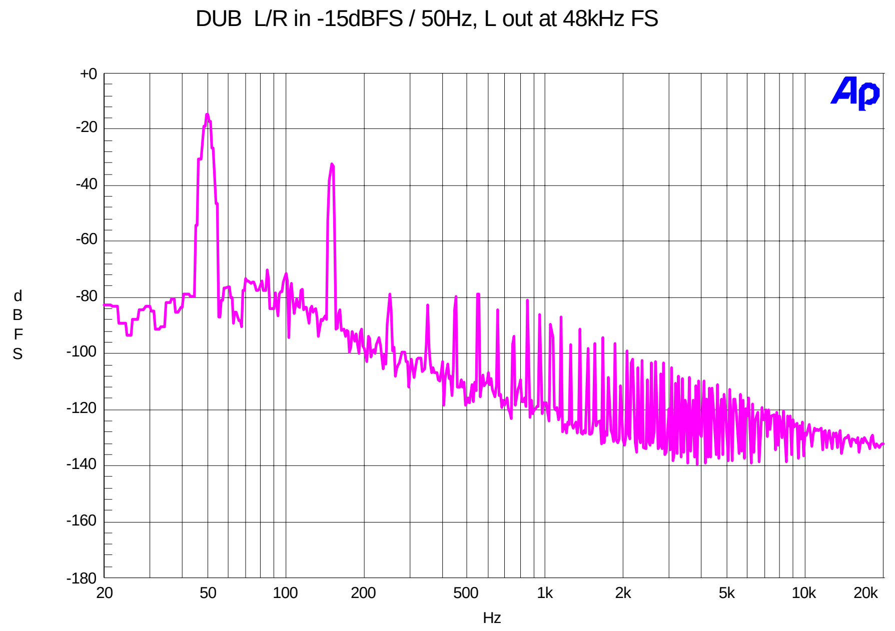


Fig 94. DUB Spectrum Plot

Dynamic Bass Enhancement (DBE): The DBE function is used in TV-sets equipped with large speakers or a subwoofer system. This feature produces a level depending bass boost. The dynamic behavior allows a strong bass amplification for small volume signals, but only small bass amplification for high volume signals. This function has to be tuned to the speakers used within a certain TV-set. The coefficients for the filters as well as the parameters for the bass boost control have to be stored into the AVIP once after power on reset. A method to find the best coefficients and parameters for a certain TV-set is available. During the coefficient download set DBE to off and returned to on when complete.

DBE is applied to the left and right speakers or alternatively with different coefficients to the subwoofer. When using DBE it is not possible to provide DUB.

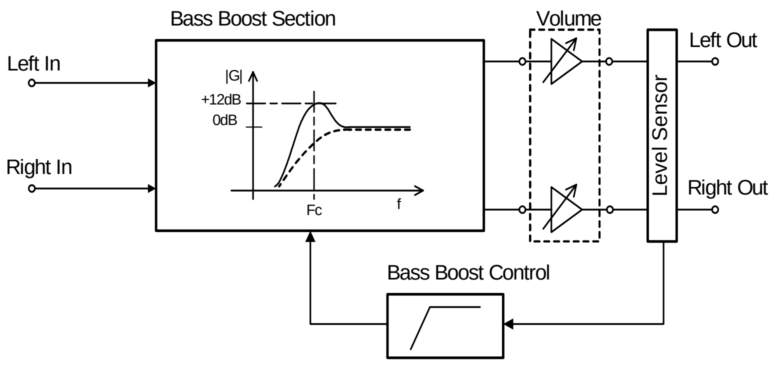


Fig 95. Dynamic Bass Enhancement (DBE)

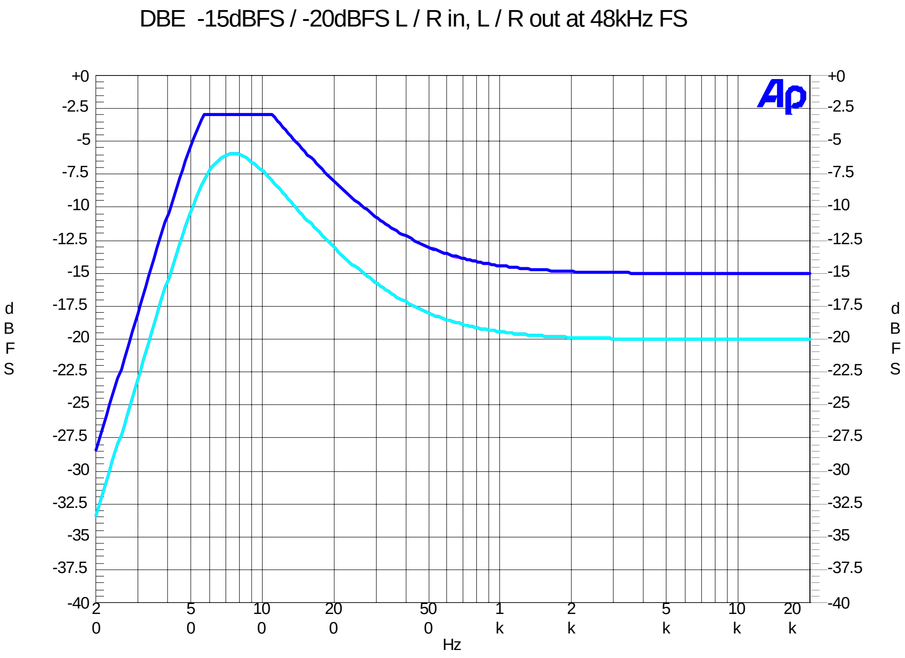


Fig 96. DBE Response Curves, two Input Levels

For the coefficient download procedure see Section "Equalizers".

The DBE coefficient RAM layout can be found in Section 3.9.9.5 "DUB/DBE Coefficients".

Candeo: Candeo is a sound feature invented by and under licence of the Philips GmbH. The Candeo feature is intended to give clear speech reproduction. This is achieved by special enhancement of the treble frequency range, as well as by use of a compressor curve. The Candeo function of PNX2015 is used for Center signals only, because most of the spoken signal content is carried by the Center.

The following curve should be used for characterization of the Candeo function.

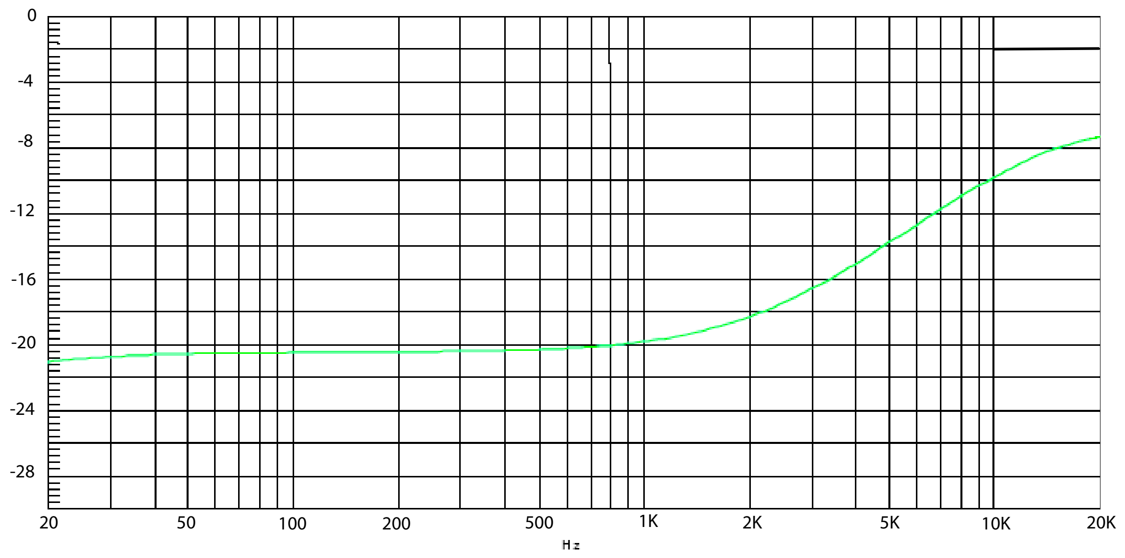


Fig 97. Candeo Response Curve @ -20dBFS input level; fs=48kHz

BBE®: BBE is a sound feature which is invented by and under licence of BBE Sound Incorporated. BBE is intended to enhance the bass and treble of the original sound, as well as to do a frequency dependent phase shift from about -180 degrees at 20 Hz to +180 degrees at 20 kHz. As announced by BBE Sound, Inc., BBE is told to lead to a "clear sound with a much higher speech/voice transparency".

The BBE function used within the AVIP has to be tuned to different TV-sets to produce the desired clear and full sound. The two control coefficients BBECONTOUR (bass) and BBEPROCESS (treble) have to be adjusted depending on the taste of the TV-set maker in combination with the used speakers of a certain TV-set. These control parameters have to be stored into the AVIP once after power on reset. During the coefficient download set BBE to off and return to on when complete.

BBE is applied to the left and right speakers. When using BBE it is not possible to use loudness in parallel.

Remark: The recommendations by BBE Sound Inc. for min. signal frequency boost have to be achieved. Contact BBE Sound Inc. for specification.

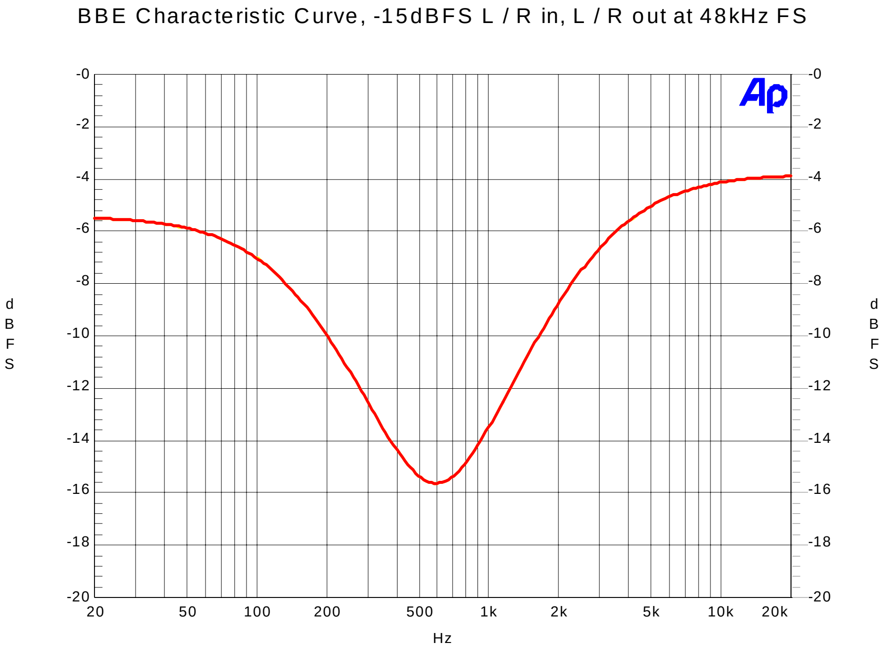


Fig 98. Response Curve - Contour +9dB @ 100Hz and Process +12dB @ 10kHz

Bass Management: Every multi-channel reproducing sound IC licensed by Dolby Laboratories includes a Bass ManagemenT BMT (also called bass redirection). The AVIP bass redirection fulfils the different configuration modes requested by Dolby Laboratories.

In general the bass redirection is used to redirect the low frequency components of the audio signal to loudspeakers (large speakers), being capable to reproduce them. In audio equipment all speakers may be large, but in TV-sets either, the L and R speakers are large or a subwoofer is applied. Thus a bass redirection can be done to the L and R large speakers or to the subwoofer. The low frequency components are cut out of the audio signals, which are directed to satellite loudspeakers (small speakers); on the other hand, the high frequency components are cut out of the audio signals, which are redirected to the subwoofer.

The corner frequency of the high and complementary low pass filters can be selected, to allow specific adjustments with respect to the loudspeakers in use. The corner frequency of the LP / HP- filters is adjustable within a range from 50 Hz to 400 Hz. There are 16 different corner frequencies to choose from: 50Hz, 60Hz, 70Hz, 80Hz, 90Hz, 100Hz, 110Hz, 120Hz, 130Hz, 140Hz, 150Hz, 200Hz, 250Hz, 300Hz, 350Hz and 400Hz. This implemented bass redirection covers sample rates of 32 kHz and 48 kHz. The 48 kHz sample rate is also used, if 44.1 kHz is active. This results in a 10% shift of the low and high pass filter curves.

The BMT in this device has been implemented on the basis of the BMT for Dolby Digital Implementations. The bass redirection (BMT) covers four different configuration modes:

1. BMT1 covers the bass management described as configuration 1 (see Dolby Licensee Information Manual for Dolby Digital Consumer Decoder). The BMT1 mode is used to redirect the low frequency components of all five channels (L, R, C, Ls and Rs) together with the LFE to a separate subwoofer.
2. BMT2 is equivalent to the bass management described as configuration 2 (see Dolby Licensee Information Manual for Dolby Digital Consumer Decoder). The BMT2 mode is used to redirect the low frequency components of the center and the surround channels to the full-range main loudspeakers (large left and right speakers). Additionally a separate subwoofer can be used in this configuration.

   Remark: The low frequency components of the surround channels must not be redirected if DPLII (or DPL) is active.
3. BMT3 represents the normal center mode described as configuration 3 (see Dolby Licensee Information Manual for Dolby Digital Consumer Decoder). The BMT3 mode is used to redirect the low frequency components of the center to the full-range main loudspeakers (large left and right speakers). If an external subwoofer is used, then another variation of configuration 3 can be used (see Figure 99)
4. BMT4 is equivalent to the bass management described as car configuration2 (see Dolby Additional Bass Management Configurations). The BMT4 mode is used to redirect the low frequency components of the left, right, center and LFE to the full-range surround loudspeakers (large Ls and Rs speakers). The external subwoofer is always disabled.

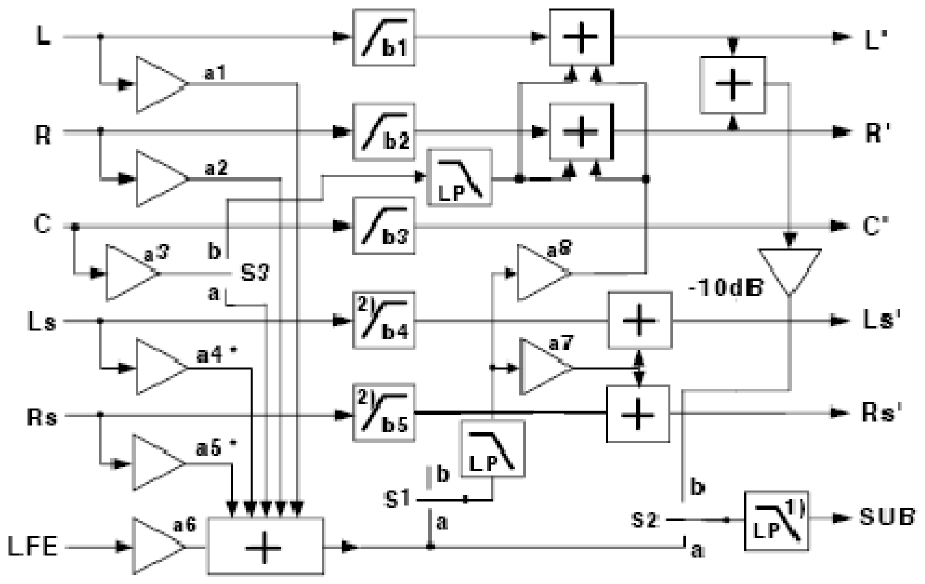
   Remark: The low frequency components of the surround channels must not be redirected if DPLII (or DPL) is active.

The BMTOFF mode is used if no redirection of the low frequency components is needed, in case of all five loudspeakers (left, right, center, left surround and right surround) are large loudspeakers and a subwoofer (used for LFE) is applied.

Remark: There is an option to switch off the low pass filter, which is located in the subwoofer output path. This non-processed sub-woofer mode can be used for BMT1 and BMT2, and gives the possibility to use an external subwoofer filter.

In BMT3 this switch is used to select between the two different configuration 3 variants.

Figure 99 to Figure 104 provide a general overview of the AVIP bass redirection.



1) LP filter can be switched flat to allow the use of external sub-woofer filtering.

*) Low frequency content of Ls and Rs is not to be redirected in DPL or DPLII mode, only in Dolby Digital mode.

Fig 99. AVIP Bass Redirection

Table 143: AVIP Bass Redirection

	BMT1[1]	BMT2[2]	BMT3[3]	BMT3a	BMT4[4]
Gain	**[dB]**	**[dB]**	**[dB]**	**[dB]**	**[dB]**
a1	-10	-100	-100	-100	-4.5
a2	-10	-100	-100	-100	-4.5
a3	-10	-4.5	-4.5	-4.5	-4.5
a4	-10	-4.5	-100	-100	-100
a5	-10	-4.5	-100	-100	-100
a6	0	+5.5	0	+1	+5.5
a7	-100	-100	-100	0	0
a8	-100	0	-100	0	-100
Switch					
s1	b	a	b	a	a
s2	a	b	a	open	open
s3	a	a	b	b	a
Filter					
b1	HP	HP	flat	flat	HP
b2	HP	flat	flat	flat	HP
b3	HP	HP	flat	HP	HP
b4	HP	HP	flat	flat	flat/HP[5]
b5	HP	HP	flat	flat	flat/HP[5]

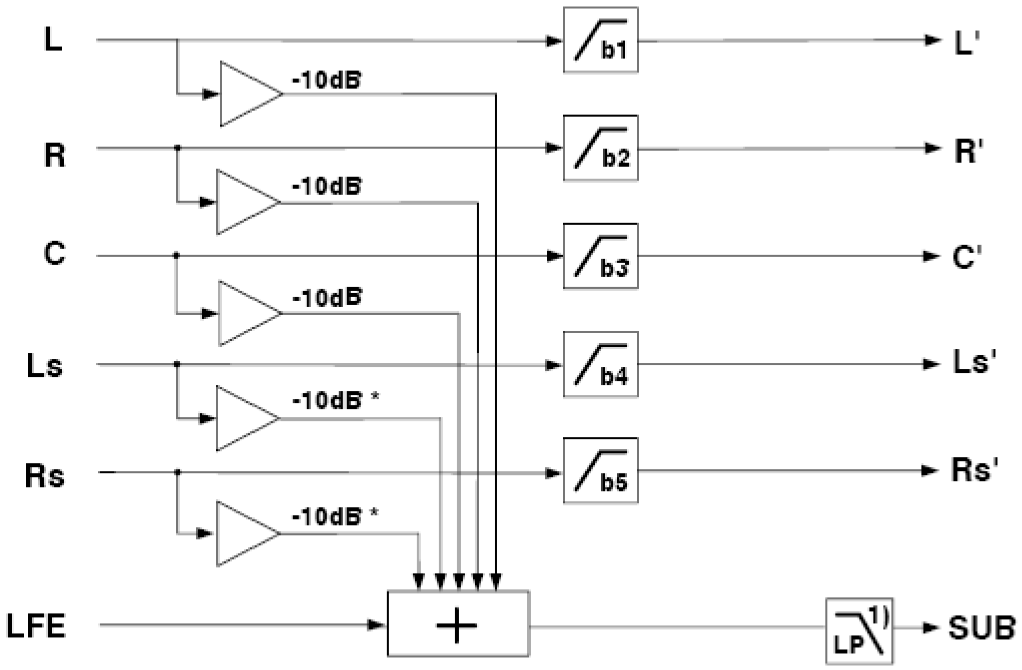
[1] BMT1, single woofer system with improved filtering: small speaker for L, R, C, Ls, Rs: extra SUB (subwoofer)

[2] BMT2, normal centre mode with bass splitter: large speaker for L, R: small speaker C, Ls, Rs : SUB optional

[3] BMT3, normal centre mode: large speaker for L, R, Ls, Rs: small speaker C: SUB optional

[4] BMT4, car configuation: large speakers for Ls, Rs: small speakers L, R, C: no subwoofer

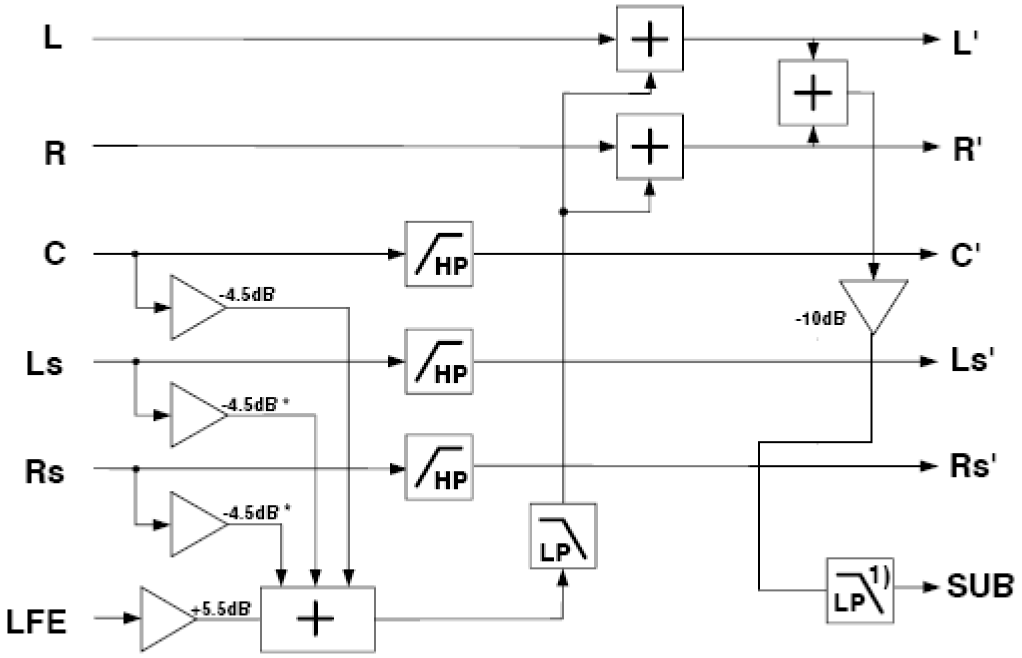
[5] BMT4 only: HP filter is active if DPL or DPLII mode is selected; otherwise HP is switched to flat.



1) LP filter can be switched flat to allow the use of external sub-woofer filtering.

*) Low frequency content of Ls and Rs is not to be redirected in DPL or DPLII mode, only in Dolby Digital mode.

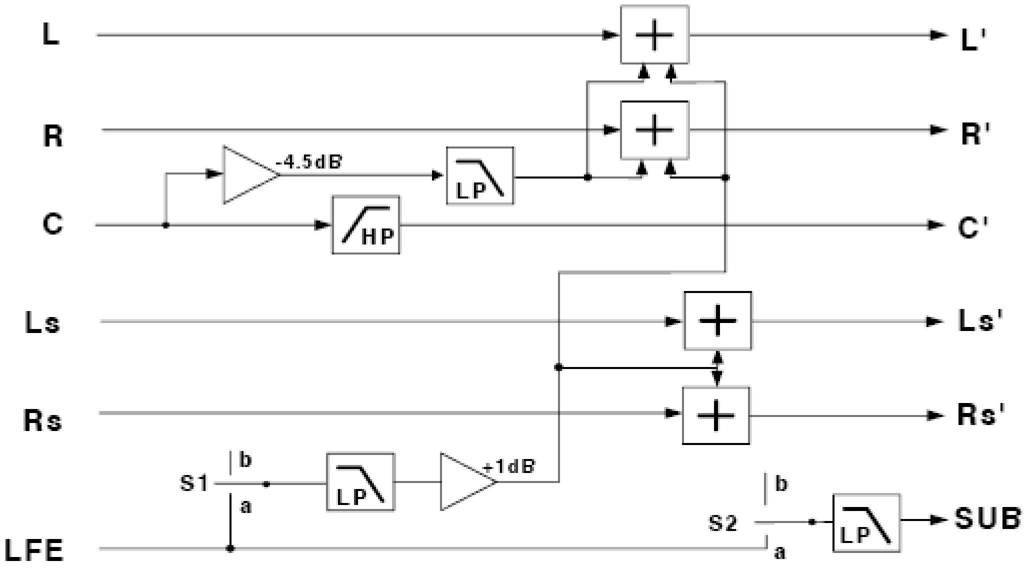
Fig 100. AVIP Bass Redirection in Configuration 1



1) LP filter can be switched flat to allow the use of external sub-woofer filtering.

*) Low frequency content of Ls and Rs is not to be redirected in DPL or DPLII mode, only in Dolby Digital mode.

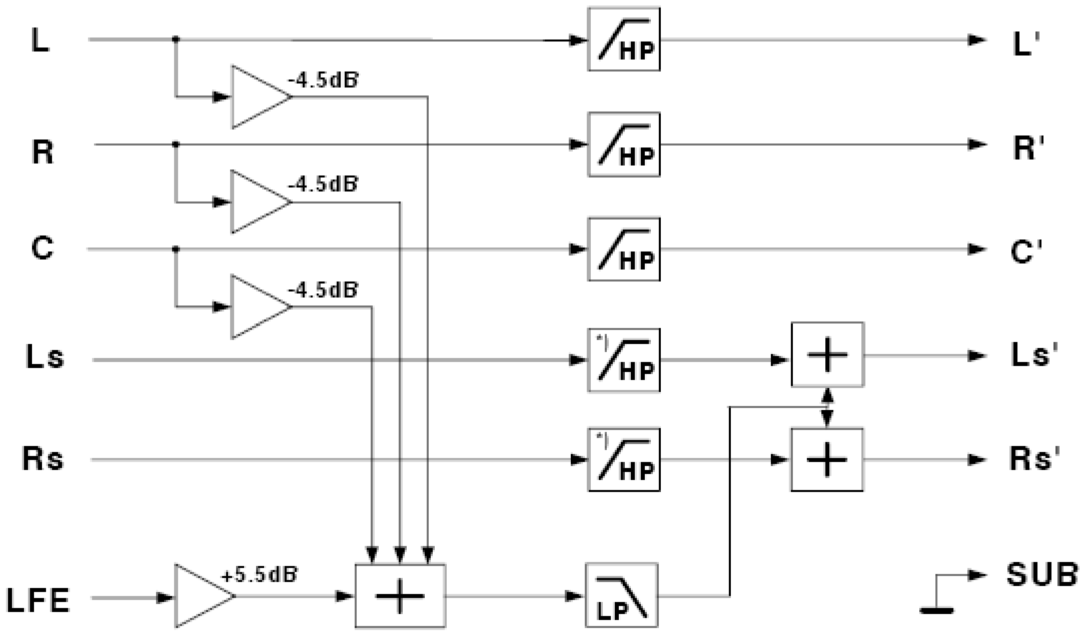
Fig 101. AVIP Bass Redirection in Configuration 2



1) LP filter can be switched flat to allow the use of external sub-woofer filtering.

*) Low frequency content of Ls and Rs is not to be redirected in DPL or DPLII mode, only in Dolby Digital mode.

Fig 102. AVIP Bass Redirection in Configuration 3



*) HP filter is active in DPL or DPLII modes because low frequency content of Ls and Rs is not to be redirected. HP filter is set flat in Dolby Digital and other modes.

Fig 103. AVIP Bass Redirection in Configuration 4 (car application)

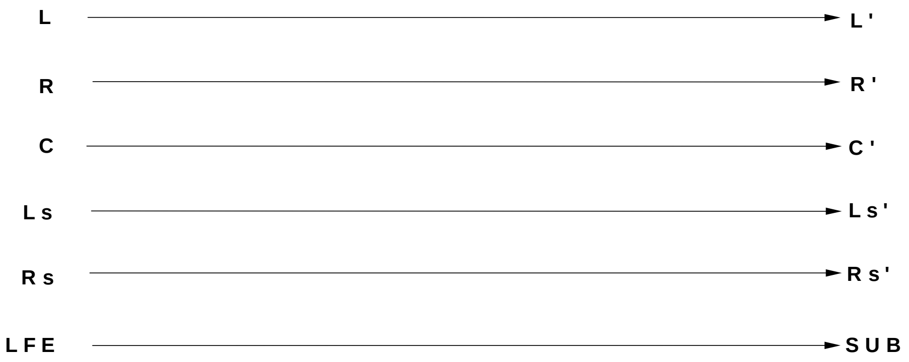


Fig 104. AVIP Bass Redirection switched off

Acoustical Compensation: For acoustical compensation issues the AVIP provides two cascaded second order IIR filter in the front channels (Main L/R, and Center).

The filters can be used for all kind of filter purposes, e.g. notches. All kinds of different filter shapes can be implemented. It is only limited to the filter order and the resolution of the 12bit wide coefficients. The coefficients (a2, b2, c2 and d2) are in a range from +1 to -1 and the coefficients (a0, a1, b1, c0, c1 and d1) in a range from +2 to -2.

The output of the filter can be gained by up to G=+24dB in case an output boost is required. The coefficient for G should be in a range of 0dB (128 default) to +24dB (2047).

Remark: Volume/trim control should be used if attenuation or a more accurate gain setting is required.

Figure 105 shows a block diagram for one filter cascade as an example.

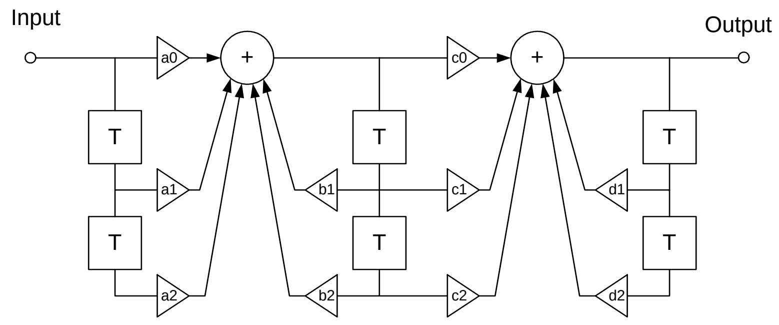


Fig 105. Acoustical Compensation Filter

The coefficients can be loaded via two control registers. One register for main left and right and one register for center channel. So it is not possible to implement two different filter shapes in main left and right.

Equalizers: The AVIP provides 3 independent 5-band equalizers for Main L/R, Center C, and Surround Ls/Rs channels. The fixed frequency bands (250, 500, 2000, 5000 and 9500Hz) used for the graphic equalizer can be easily controlled by gain settings, where the coefficients for the parametric equalizer have to be downloaded. The gain of each band is adjustable within the range of -12dB to +12dB in 1dB steps.

The equalizers are built out of 4 cascaded cross filter sections, which provide 5 filter bands. Each of the 4 sections allows the choice between graphic (fixed) and parametric mode. Measurements of the equalizer in graphic mode can be found on the next pages (see Figure 107 to Figure 110). The 3 equalizers can only be used alternatively to the ‘bass / treble’ sections. The 3 provided ‘bypass / enable’ switches (MAINEQENABLE, CENTEREGENABLE, SURROUNDEQENABLE) allow to disable (bypass) or enable each of the 3 independent equalizers.

As already mentioned, each equalizer consists of 4 Cross-Filter sections, which provide 5 filter bands. Therefore the parametric equalizer consists of 4 variable coefficient sections.

Figure 106 shows the coefficient download procedure for the parametric equalizer:

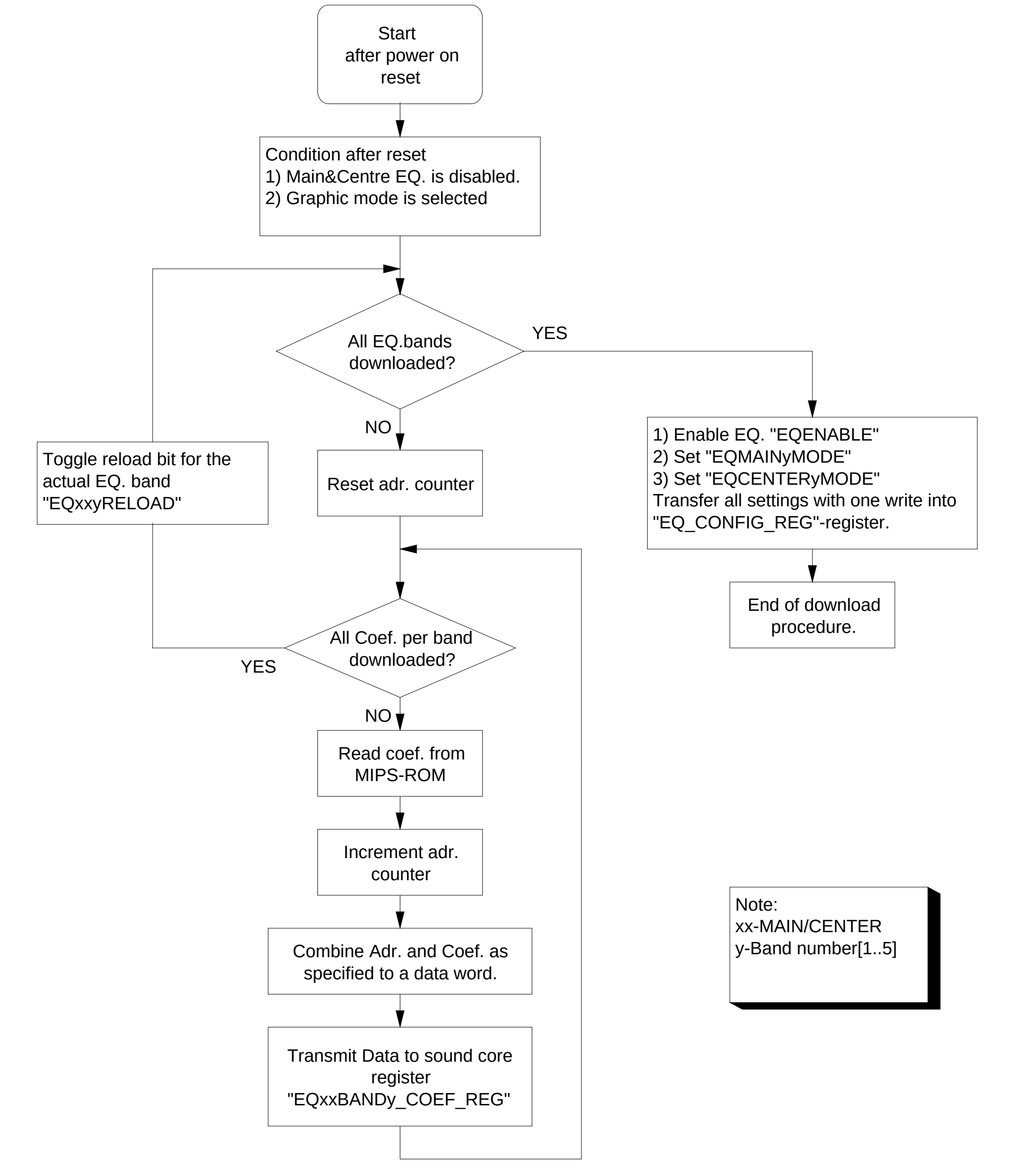


Fig 106. Download Procedure - Equalizer Coefficients after Power On Reset

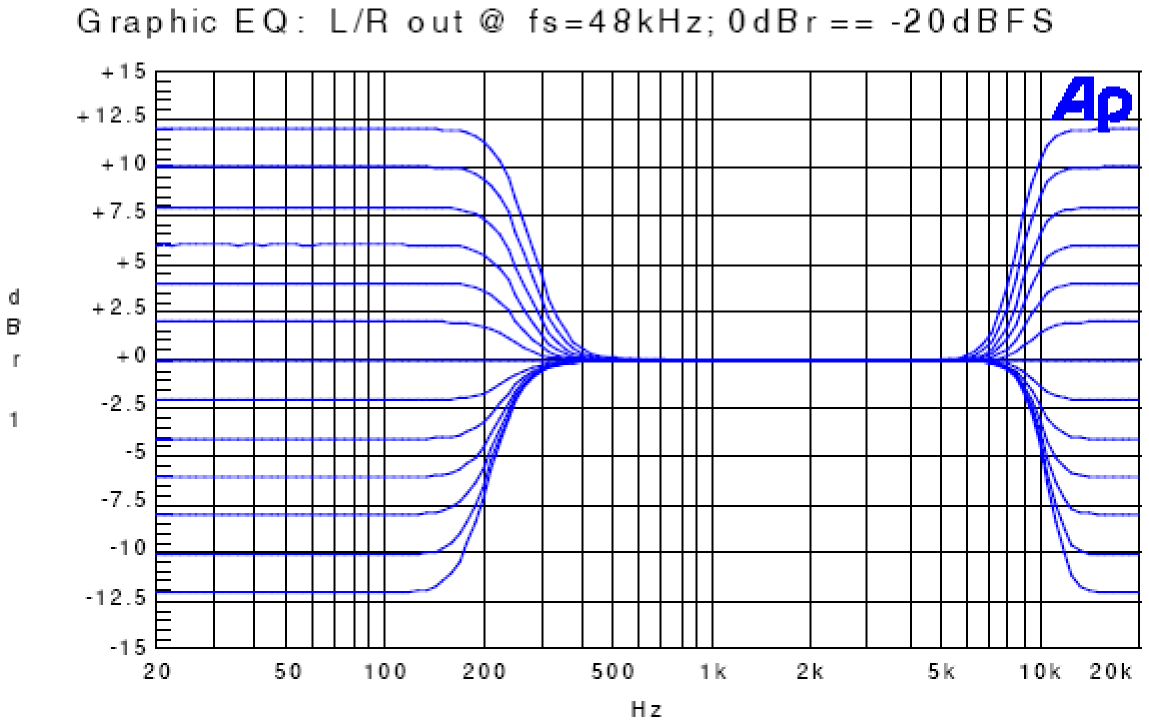


Fig 107. Graphic Equalizer: 250Hz and 9.5kHz Band (2dB Steps)

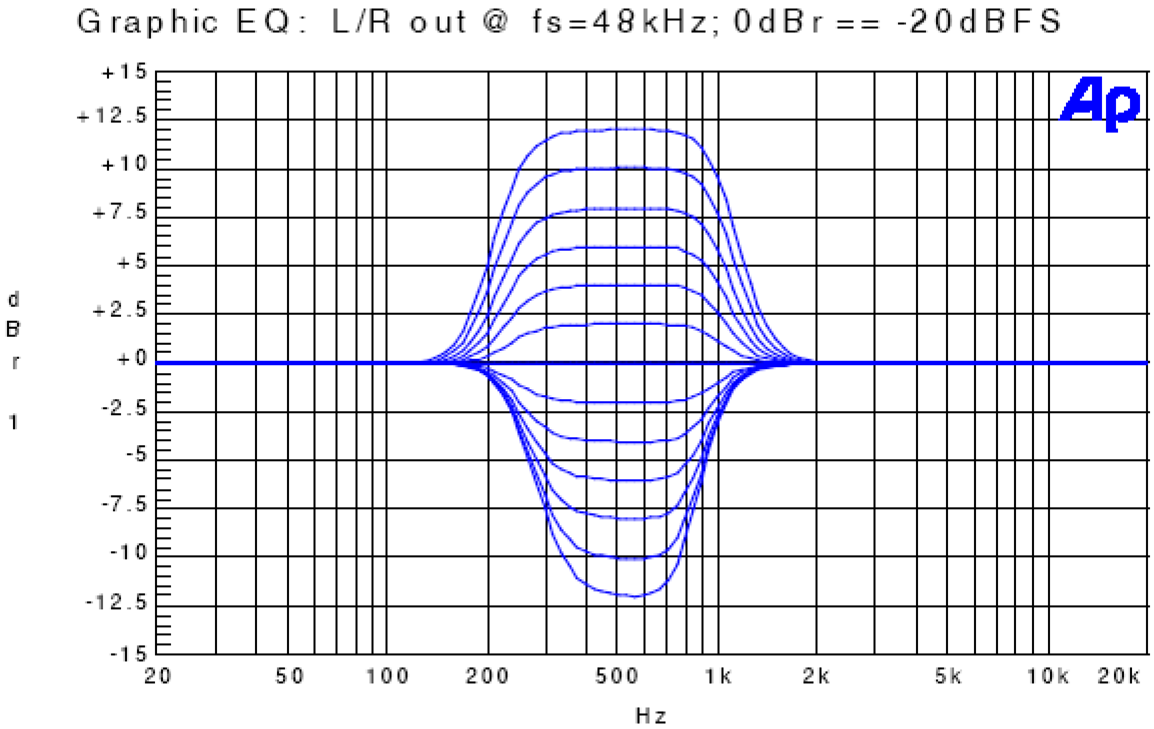


Fig 108. Graphic Equalizer: 500Hz Band (2dB Steps)

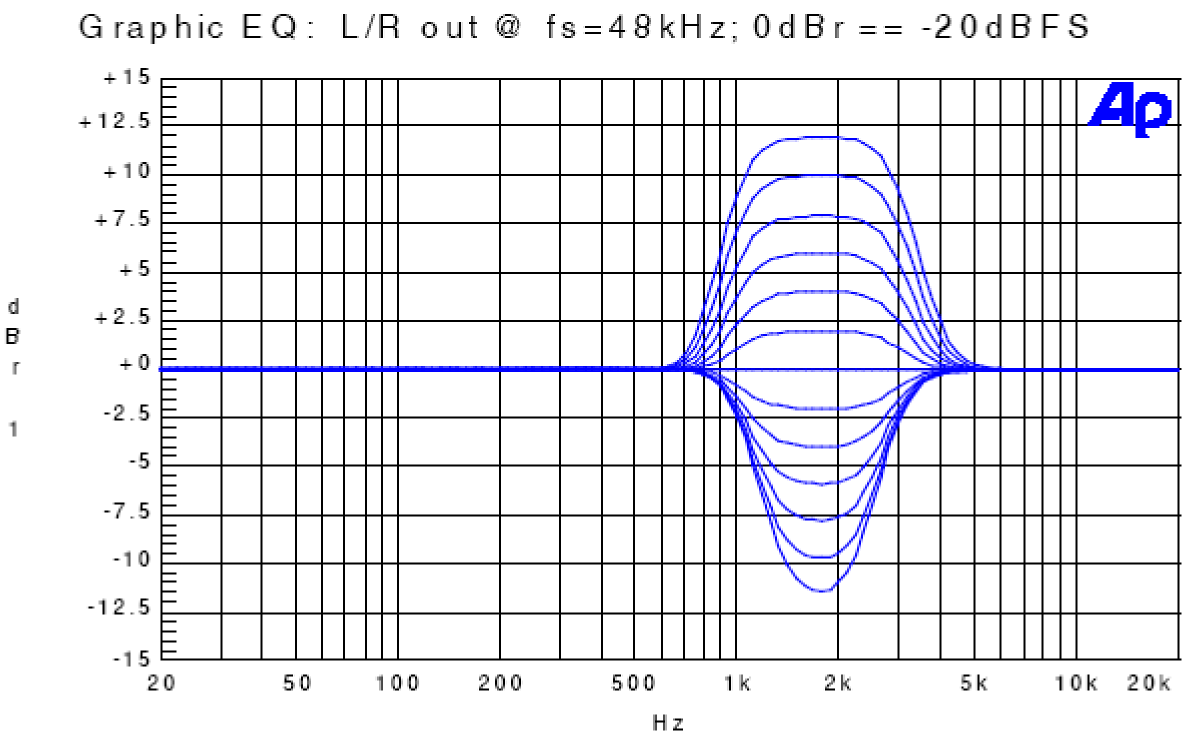


Fig 109. Graphic Equalizer: 2kHz Band (2dB Steps)

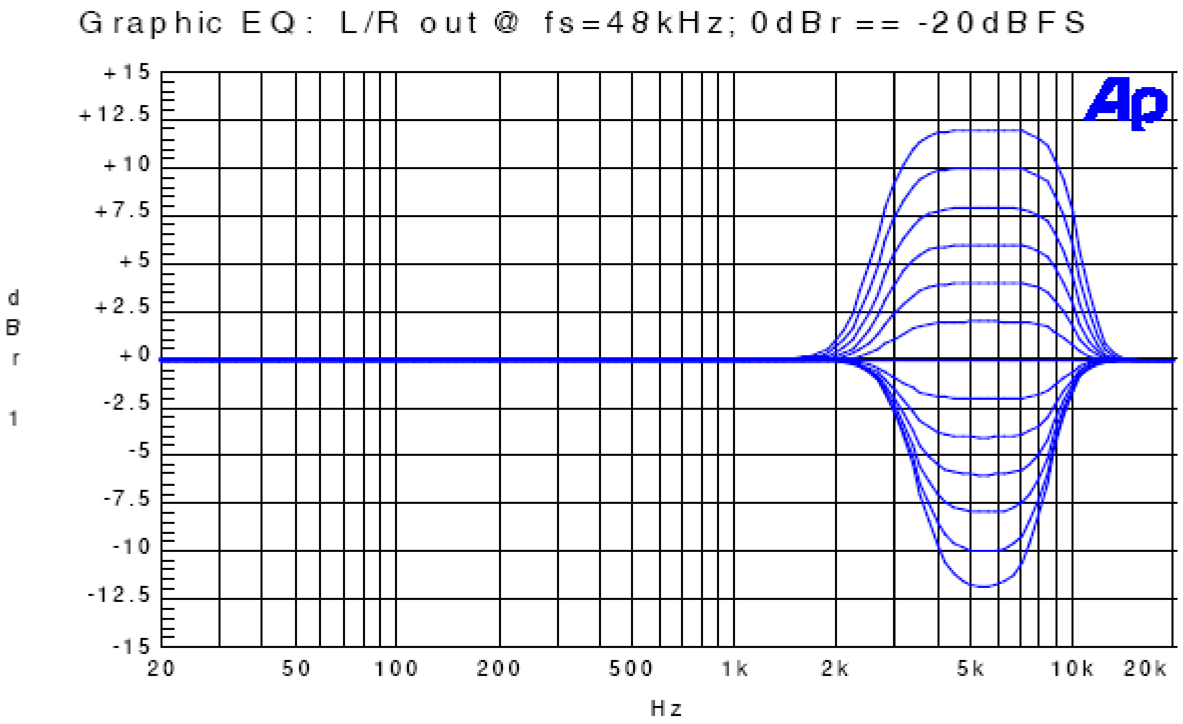


Fig 110. Graphic Equalizer: 5kHz Band (2dB Steps)

Volume and Trim: Stereo channels have separate gain elements in the left and right branch. In the MAIN channel the L,R Trims are used. Shift to the right is done by attenuation of the L Trim, shift to the left by attenuation of the R Trim. The same control is provided within the AUX channels by use of Volume/Trim Left and Right. This needs to be programmed by the set maker.

Beeper: The Beeper is a sine wave generator for frequencies from 200Hz to 12.5kHz at a sample rate of 32kHz and 48kHz. For a sample rate of 44.1kHz the coefficient set of 48kHz is used. This will end up with a 10% shift of the selectable frequencies. The level can be set between 0dBFS and -83dBFS. A Mute/Off step is available. The signal is mixed into the MAIN Left, Right, Center and the AUX1/HP channel.

If the Beeper is not used it needs to be set into the Mute/Off stage.

Mono Signal for Cancellation in the Voice Control IC: The L and R output signals of the Main channel are added before entering the Digital Output Crossbar giving (L+R)/2. This signal can be provided to the I^{2}S output and passed to a voice control IC.

Audio Monitor: The audio monitor is able to monitor the level of the sum (A+B)/2, left or right signal of all input channels of the Digital Input Crossbar. Via the OUTCOPY feedback also any input to the Output Crossbar is accessible. A special setting is the (A-B)/2 mode in the Digital Matrix that offers the possibility to identify a signal as mono or stereo.

The audio monitor provides three different modes:

- Last sample: in this mode the level of the last sample from the selected input is stored in the monitor register.
- Peak detection: in this mode the peak level after the last read command is stored in the monitor register.
- Quasi peak detection: a quasi peak detector with an attack time of 4ms and a decay time of 1s is applied.

If the monitor is used for mono/stereo detection the quasi peak mode should be selected. The transfer rate via control bus is limited to about 15kHz.

Digital Output Crossbar: The Digital Output Crossbar provides 24 selectors 'one out of 20'. That means each of the outputs e.g. DAFO1 or I2S1L can be connected to each of the inputs e.g. MAIN L or C etc.

By this the set maker is free to assign the outputs DAFO1 to DAFO8 to any of the L, R, SUB, C, Ls, Rs speaker or headphones. It gives the possibility to the set maker to avoid crossovers in the chassis layout and to place the audio power amplifiers at an optimal location. Connect the headphone outputs to AUX1.

Clip Management: Great care has to be taken when applying gain to large input signals in order not to exceed the clipping level. For nominal modulation ratios and levels of the various input signals the level at the digital input crossbar is -15dBFS.

Internal clipping in the chain of bass, treble and equalizer stages with a total maximum gain of 27dB can be avoided by the implementation of level shifts of -12dB in front of the bass treble stage and -6dB in front of the equalizer. Thus the headroom is still 6dB in front of master volume. If the loudness function is switched on another level shift of -18dB is done in front of it. The level shift (in total -18dB or -36dB) is compensated behind the master volume/trim stage. Clipping behind all gain elements can be avoided by limiting the maximum volume/trim gain and/or the bass, treble and equalizer gain.

To prevent clipping different strategies in handling the gain settings of bass, treble, equalizer and master volume/trim are possible. The set maker can implement such strategies by his own microcontroller programs. Or he can use the integrated clip management, which offers four different modes.

- Static Volume Mode" - In this mode the master volume setting is limited to a maximum gain of -30dB. The trim and the volume/balance in the Aux1-6 are not effected.
- Static Control Mode" - In this mode the bass, treble and equalizer setting is limited to a maximum of +8dB. The Volume plus Trim setting is limited to -1dB.
- Dynamic Control Mode" - In this mode the master volume and trim setting is limited to +3dB. If the master volume plus trim setting exceeds +3dB the bass and treble are reduced until the amplification is less then +3dB. Every 1dB more master volume plus trim results in 1dB less bass and treble.
- Dynamic Volume Mode" - In the dynamic volume mode the main left and right signal is measured. If the internal signal exceeds a limit of -3dBFS for a longer time, the master volume is reduced automatically until the measured signal is lower -3dBFS.

Table 144: Clip Management

Clip management	**Settings**					
	Master volume	**Bass**	**Treble**	**Loudness**	**EQb1, EQb2**	**EQb4, EQb5**
Static volume mode	limited to -30dB	not effected	not effected	not effected	not effected	not effected
Static control mode	limited to -1dB	limited to +8dB	limited to +8dB	none attack level 0dB	limited to +8dB	limited to +8dB
Dynamic control mode	limited to +3dB	-	-	none attack level 0dB	not effected	not effected
Dynamic volume mode	reduced/limited until the signal is smaller than -3dBFS	not effected	not effected	none attack level 0dB	not effected	not effected

Power On / Reset Specification: After a power on or reset condition the whole DSP-RAM is cleared. Afterwards all module defined memory cells are initialized and the Control-Registers are set to their default values. These values are defined as 'Default@INIT' within the control table.

3.9.9.4 Audio Feature Specification

The sound features of the AVIP sound processor are described in the following tables with inputs / outputs, functionality, sample rate and the control bits. 'Control Mechanism' shows the control bits with the 'variable name' from the register map. The control bits are split into 'Set once during start up' and 'Change during operation'.

'Set once during start up' means that these bits are not under end-user control, they are only used in procedures written by the setmaker for the start-up configuration of a set.

'Change during operation' means that these bits are under end-user control, they are used in procedures for the operation of a set.

The row 'Sample Rate' shows the sample rates, which are provided by this module. If a sample rate is missing, e.g. 44.1kHz the sample rate depending coefficients of the best fitting sample rate is used - for this example 48kHz.

Automatic Volume Levelling (AVL)

Table 145: AVL Feature Description

Functional description	The AVL reduces the audio input signal to a settable maximum output signal. The processing can be done linear over the frequency range or weighted by a CCIR468 filter characteristic.
Application	The AVL is processed on the front channels (L,R,C,Ls,Rs). Depending on the bass redirection mode, also the subwoofer channel is effected.
Control mechanism (set during start-up)	1) The reference level ('AVLLEV') is settable in 2dB steps. A control range of 32dB is available. ('AVLLEV': '0000' -> -6dB, '0001' -> -8dB, '1111' -> -36dB 2) Switchable weighting filter ('AVLWEIGHT': '0' -> Off, '1' -> On) Note: During On/Off switching a short 'pop' noise is audible.
Control mechanism (change during operation)	3) Switchable decay time ('AVLMOD': '000'-> Off, '001'-> 20ms, '010'-> 2s, '011'-> 4s, '100'-> 8s '101'-> 16s '110'...'111' ->Reserved)
Sample rate	32kHz and 48kHz

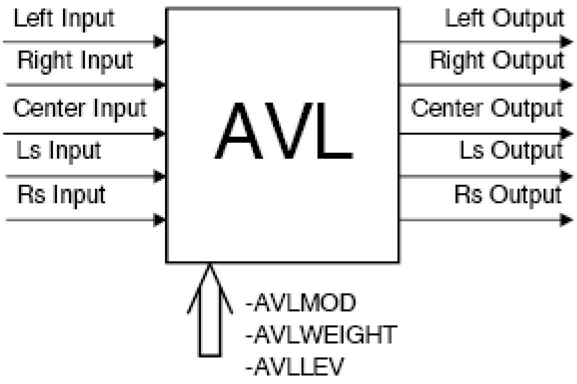


Fig 111. AVL Block Diagram

Dolby® Pro Logic® II (DPLII)

Table 146: DPLII Feature Description

Functional description	See Dolby Licensee Information Manual: Dolby Pro Logic II'.
Application	Processed on Main channels (L/R). The Main, Centre, Left Surround and Right Surround channels are reproduced by the Dolby Pro Logic II decoder from an DPL encoded input signal. Note: 1) During the different operation modes digital silence is available on the unused outputs (e.g. in 3Stereo mode the surround channel is carrying digital silence) 2) The input and output levels are specified by Dolby Labs. 3) During On/Off switching a short 'pop' noise is audible.
Control mechanism (change during operation)	1) The Dolby Mode Control field: 'SNDMOD': $3 -> Dolby Wide Centre. $4 -> Dolby 3Stereo. $5 -> Dolby Phantom Centre.
	2) The Dolby Decoder Mode Control field: 'DPL2MOD': $0 -> Movie mode $1 -> Matrix m ode $2 -> Music mode $3 -> Pro Logic Emulation mode $4 -> Virtual mode
Sample rate	32kHz, 44.1kHz and 48kHz

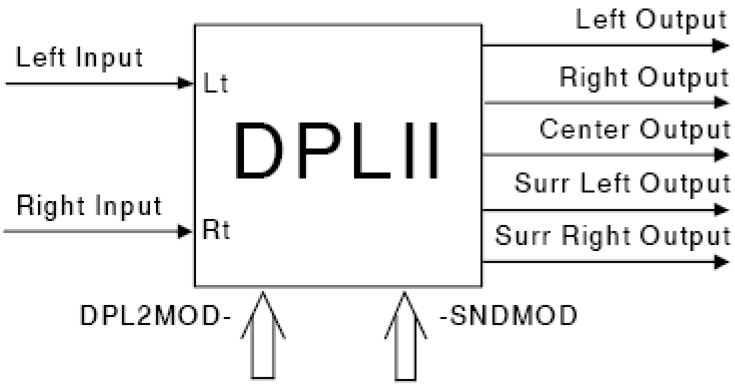


Fig 112. DPLII Block Diagram

Virtual Dolby® Surround (VDS)

Table 147: VDS Feature Description

Functional description	1) VDS(522) re-directs the centre and surround channel information to the main channels (front speakers). This feature is used in two speaker applications. 2) VDS(523) re-directs the surround channel information to the main channel. The centre channel information is reproduced by a separate centre speaker, therefore this feature is used in three speaker applications.
Application	The VDS needs a 5 channel input source (L,R,C,Ls,Rs). Note: 1) The DPLII decoder is set in Wide Centre mode. 2) in VDS522 the centre channel is muted. 3) During On/Off switching a short 'pop' noise is audible.
Control mechanism (set during start-up)	1) Switchable via VDS Mode between VDS522 and VDS523 'SNDMOD': $6 ->VDS522 $7 ->VDS523
Control mechanism (change during operation)	2) Selectable intensity in percentage 'VDDMIXLEV': '000'-> 0% (minimum effect) '001'-> 20% '101'-> 100% (maximum effect) '110'&'111' ->Reserved
Sample rate	32kHz, 44.1kHz and 48kHz

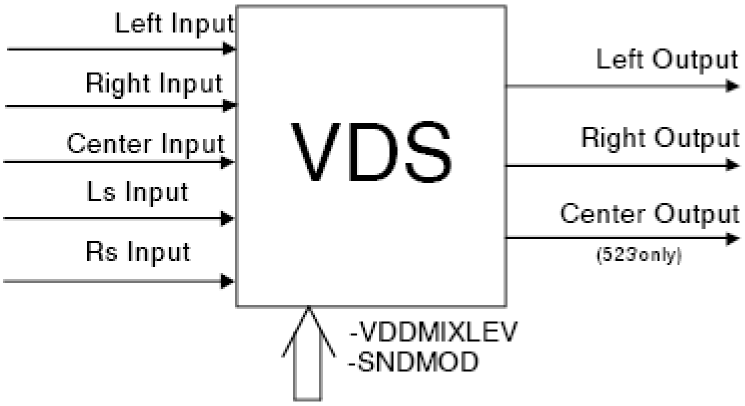


Fig 113. VDS Block Diagram

Virtual Dolby® Digital (VDD)

Table 148: VDD Feature Description

Functional description	1) VDD(522) re-directs the centre and surround channel information to the main channels (front speakers). This feature is used in two speaker applications. 2) VDD(523) re-directs the surround channel information to the main channel. The centre channel information is reproduced by a separate centre speaker, therefore this feature is used in three speaker applications.
Application	The VDD needs a 5 channel input source (L,R,C,Ls,Rs). Note: 1) An external Dolby Digital Decoder is used in combination with VDD. 2) in VDD522 the centre channel is muted. 3) During On/Off switching a short 'pop' noise is audible.
Control mechanism (set during start-up)	1) Switchable via VDS Mode between VDD522 and VDD523 'SNDMOD': $8 ->VDD522 $9 ->VDD523
Control mechanism (change during operation)	2) Selectable intensity in percentage 'VDDMIXLEV': '000'-> 0% (minimum effect) '001'-> 20% '101'-> 100% (maximum effect) '110'&'111' ->Reserved
Sample rate	32kHz, 44.1kHz and 48kHz

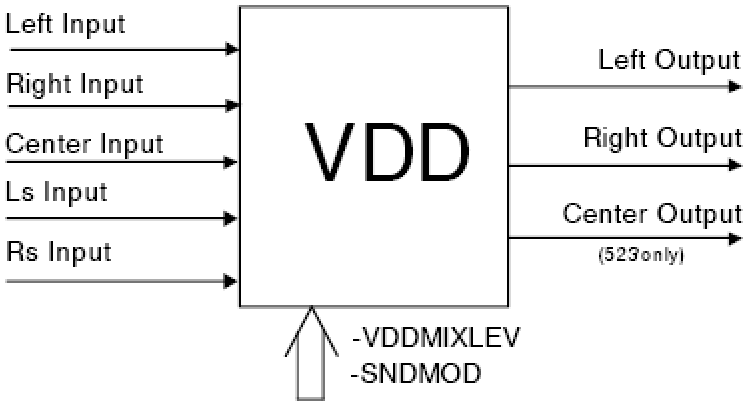


Fig 114. VDD Block Diagram

Incredible Stereo (IStereo)

Table 149: IStereo Feature Description

Functional description	When the main signal (stereo signal) is processed by the IStereo module, the listener gets the impression that the signal is reproduced by 2 virtual speakers. It gives the impression that the main channel speakers are positioned at a greater distance (angle) than the actual speakers are. The stereo image is expanded.
Application	The IStereo module can be used for a main stereo signal. This is useful, if the distance of the speakers is low compared to the distance between speakers and listener. Note: 1) On/Off switching with an intensity of 100% will cause no signal level shift. 2) During On/Off switching a short 'pop' noise is audible.
Control mechanism (set during start-up)	1) The intensity can be changed in percentage 'INSOEF': '000'-> 0%, '001'-> 20%, '101'-> 100%, '110' & '111' ->Reserved. Note: This bit field is used for both Incredible sound features (IMono and IStereo).
Control mechanism (change during operation)	2) Turn effect On/Off 'INSOMO': '00' ->OFF, '01' ->IStereo On Note: This bit field is used for both Incredible sound features (IMono and IStereo).
Sample rate	32kHz and 48kHz

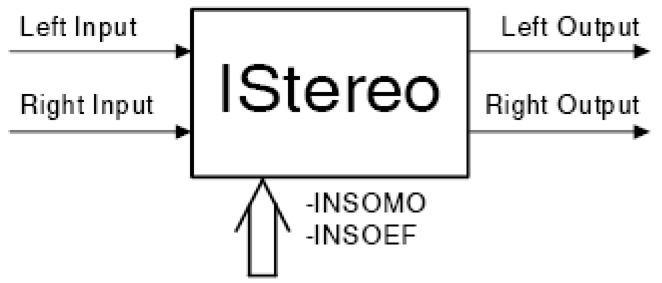


Fig 115. IStereo Block Diagram

Incredible Mono (IMono)

Table 150: IMono Feature Description

Functional description	The Incredible Mono (IMONO) module reproduces a pseudo stereo signal (Left and Right) from a mono input signal. The mono input signal is split into frequency bands and afterwards re-directed to the left and right speaker.
Application	The IMono module is used for the main channel signal, if the received signal is mono and a pseudo stereo impression is preferred. Note: 1) On/Off switching with an intensity of 100% will cause no signal level shifts. 2) During On/Off switching a short 'pop' noise is audible.
Control mechanism (set during start-up)	1) The intensity can be changed in percentage 'INSOEF': '000'-> 0%, '001'-> 20%, '101'-> 100%, '110' & '111' ->Reserved. Note: This bit field is used for both Incredible sound features (IMono and IStereo).
Control mechanism (change during operation)	1) Switchable On/Off 'INSOMO': '00' ->OFF, '10' ->IMono ON Note: This bit field is used for both Incredible sound features (IMono and IStereo).
Sample rate	32kHz and 48kHz

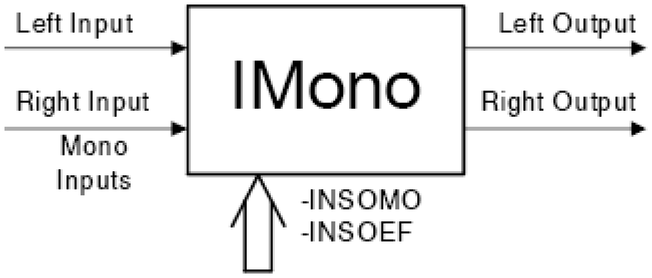


Fig 116. IMono Block Diagram

Dynamic Ultra Bass (DUB)

Table 151: DUB Feature Description

Functional description	In principle the DUB algorithm shifts the bass signal band (30Hz-70Hz) to higher frequencies (70Hz-140Hz). The original bass spectra is removed from the signal. The filter characteristics are optimized for the TV set. Only with this optimization it is possible to reach the best or maximum effect.This feature makes it possible to reproduce deep bass image with smaller speakers.
Application	The DUB module is used in the main or subwoofer channel for application with smaller speakers, which are not able to reproduce the frequency range of 30Hz to 70Hz. Note: During On/Off switching a short 'pop' noise is audible.
Control mechanism (set during start-up)	1) Use default setting or overwrite the coefficients via DUB coef. Download register. 2) Switch On/Off and select the active channel Main/Subwoofer.
Control mechanism (change during operation)	1) Switchable On/Off and Main/Subwoofer channel (Combined control field for DBE and DUB) '000' -> DBE and DUB OFF, '001' -> DBE main channel ON '010' -> DUB main channel ON '011' -> DBE subwoofer channel ON '100' -> DUB subwoofer channel ON)
Sample rate	32kHz, 44.1kHz and 48kHz

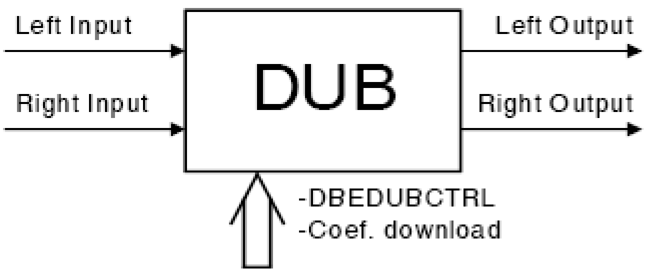


Fig 117. DUB Block Diagram

Dynamic Bass Enhancement (DBE)

Table 152: DBE Feature Description

Functional description	In principle the DBE algorithm boost the bass signal (60Hz) with a maximum gain of +12dB. This boost depends on the signal output level of the DBE module. The frequency (60Hz) and the maximum gain should be adjusted and optimized for every TV set.
Application	Is used in the main or subwoofer channel to generate a bass boost depending on the signal output level. Note: During On/Off switching a short 'pop' noise is audible.
Control mechanism (set during start-up)	1) Use default setting or overwrite the coefficients via DBE coef. Download register. 2) Switch On/Off and select the active channel Main/Subwoofer.
Control mechanism (change during operation)	1) Switchable On/Off and Main/Subwoofer channel (Combined control field for DBE and DUB) '000' -> DBE and DUB OFF, '001' -> DBE main channel ON '010' -> DUB main channel ON '011' -> DBE subwoofer channel ON '100' -> DUB subwoofer channel ON)
Sample rate	32kHz, 44.1kHz and 48kHz

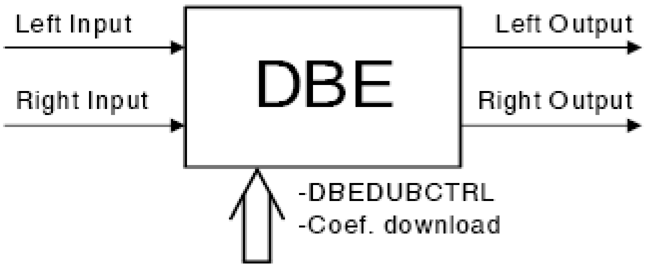


Fig 118. DBE Block Diagram

Loudness

Table 153: Loudness Feature Description

Functional description	The Loudness function boosts the low and high frequency parts of the main signal. The low frequency parts with a maximum gain of +18dB and the high frequency parts with a maximum gain of +4,5dB. The gain for the low and high frequency parts depends on the volume level. This characteristic is used for signal correction if the output volume is low.This depends on the hearing characteristic of the human ear.
Application	Is built in the main and centre channel to correct the signal characteristic when the reproduced signal amplitude is low. Note: During On/Off switching a short 'pop' noise is audible.
Control mechanism (set during start-up)	1) Settable none attack level in 3dB steps ('MAINLONA': '000' ->Volume -15dB, '001' ->-12dB, '010' ->-9dB, '111' ->+6dB). 2) Settable none attack frequency ('MAINLOCH': '00' ->Standard, '01' -> extra bass, '10', '11' -> Reserved)
Control mechanism (change during operation)	3) Switchable On/Off ('MAINLOUD': '0' ->OFF, '1' ->ON)
Sample rate	32kHz and 48kHz

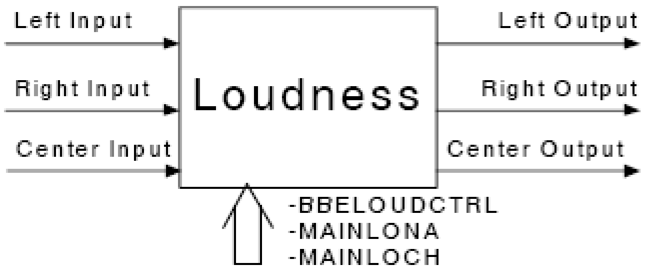


Fig 119. Loudness Block Diagram

BBE

Table 154: BBE Feature Description

Functional description	
Application	The BBE function is built in the main channels to achieve a better signal and speech clarity. The settings of BBECONTOUR and BBEPROCESS have to be adjusted depending on the different loudspeakers used in the TV application. Note: During On/Off switching a short 'pop' noise is audible.
Control mechanism (set during start-up)	1) Settable low frequency boost ('BBECONTOUR': '0000' -> 0dB, '1111' -> + 14dB). 2) Settable high frequency boost ('BBEPROCESS': '0000' -> 0dB, '1111' -> + 13dB)
Control mechanism (change during operation)	3) Switchable On/Off (BBELOUDCTRL': '00' ->OFF, '10' ->ON)
Sample rate	32kHz, 44.1kHz and 48kHz

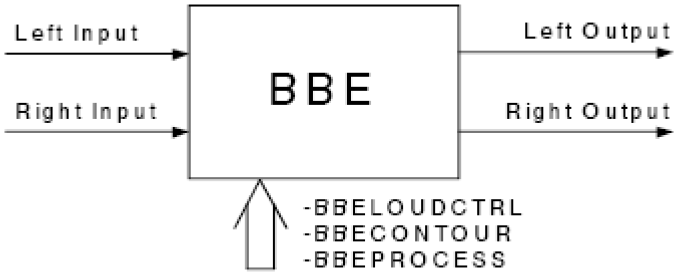


Fig 120. BBE Block Diagram

Base and Treble

Table 155: Base and Treble Feature Description

Functional description	Independent low and high frequency cut and boost controlled by the user. The external resolution is defined in 1dB steps. The internal resolution 1/32dB steps are used to avoid DC plops. The maximum of change is 15.625dB/s.
Application	Implemented in Main, Center, Left and Right Surround and AUX1 (Headphone) channels.
Control mechanism (change during operation)	1) A five bit wide control field. The bit field is interpreted as a 2's complement value: xx -> 'MAIN', 'CENTRE', 'SURROUND', 'AUX1' ('xxBASS': '01111' ->+15dB, '00000' ->0dB, '11111' ->-1dB) '10001' ->-15dB) '10000' ->-16dB) ('xxTREBLE': '01111' ->+15dB, '00000' ->0dB, '11111' ->-1dB) '10001' ->-15dB) '10000' ->-16dB)
Sample rate	32kHz and 48kHz

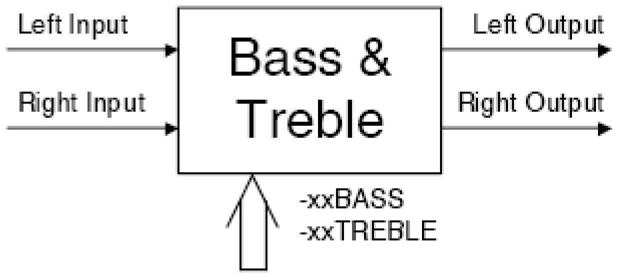


Fig 121. Base and Treble Block Diagram

Base Management

Table 156: Base Management Feature Description

Functional description	The task of the Bass Management is the re-direction of the bass signal. This is necessary if the loudspeakers are no full range speakers. The structure of the BMT module is defined by Dolby specifications: 1) 'Licensee Information Manual: Dolby Digital Consumer Decoder 2) Dolby Pro Logic II - 'Dolby Licensee Information Manual: Dolby Pro Logic II'.
Application	Main, Centre, Left surround and Right surround channels are fed through the BMT module to re-direct the bass information depending on the speaker set used. The subwoofer signal is generated by the BMT module. Note: During On/Off switching a short 'pop' noise is audible.
Control mechanism (set during start-up)	1) Mode field ('BAMAMO': '00'-> Off, '01'-> Type1, '10'-> Type2, '11'-> Type3) 2) Control field for the cut-off frequency ('BAMAFC': '0000'->50Hz, '0001'->60Hz, '1010'->150Hz, '1011'->200Hz, '1111'->400Hz) 3) Subwoofer filter control ('BAMASUB': '0-> Off (flat), '1-> On)
Sample rate	32kHz, 44.1kHz and 48kHz

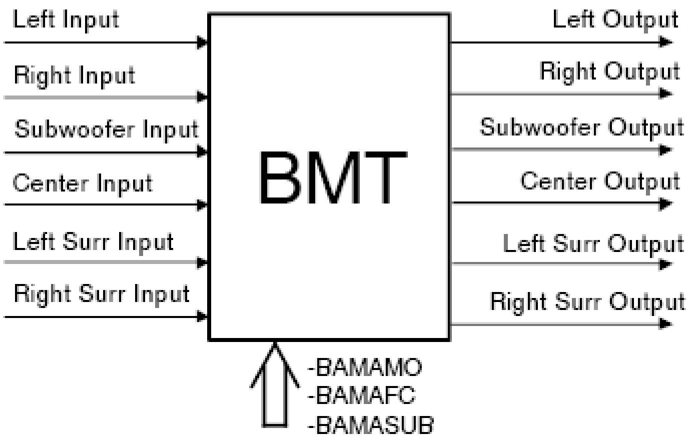


Fig 122. Base Management Block Diagram

Delay Line Unit

Table 157: Delay Line Unit Feature Description

Functional description	Delays a mono signal between 0ms and 50ms depending on the 'DPLDEL' value. This can be used for Hall or Matrix effect. Delays a stereo signal between 0ms and 25ms depending on the 'DPLDEL' value. This can be used for DPLII or Dolby Digital. Mono or stereo delay line depends on the selected sound mode.
Application	Is required by the Dolby specification for the surround channels. Note: During On/Off switching a short 'pop' noise is audible.
Control mechanism (change during operation)	1) Delay line length from 0ms up to 25ms for stereo or 0ms up to 50ms for mono. ('DPLDEL': Stereo / Mono '00000' ->OFF / OFF, '00001' ->1ms / 2ms, '00010' ->2ms / 4ms, '11001' ->25ms / 50ms '11010'...'11111' -> Reserved)
Sample rate	32kHz, 44.1kHz and 48kHz

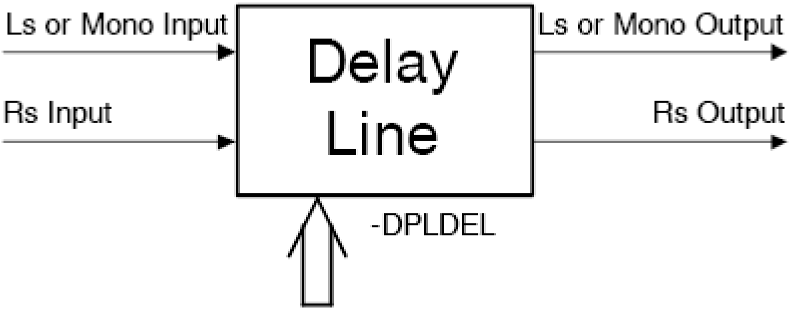


Fig 123. Delay Line Unit Block Diagram

Pseudo Hall / Matrix

Table 158: Pseudo Hall / Matrix Feature Description

Functional description	This module produces always mono signal (L+R)/2 for the center channel. For the surround channels in 'Pseudo Hall Mode' the mono signal is used or in 'Pseudo Matrix Mode' the difference signal (L-R)/2. The center output can be switched off independent from the mode (Hall or Matrix).
Application	Is built in the main channel to generate output signals for the center and surround channels when no multi channel signal is available. Note: During On/Off switching a short 'pop' noise is audible.
Control mechanism (change during operation)	This module can be switched between 'Pseudo Hall', 'Pseudo Matrix' or 'Off' via the Sound Application Mode Table. 1) Via "Sound Application Mode" $0 = Mono/Stereo (Off) $1 = Mono/Stereo (Pseudo Hall) $2 = Mono/Stereo (Pseudo Matrix) 2) 'CENTERMUTE': $0 = Center On $1 = Center Off
Sample rate	32kHz, 44.1kHz and 48kHz

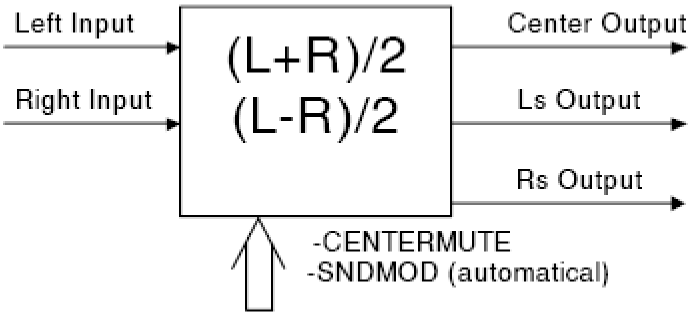


Fig 124. Pseudo Hall / Matrix Block Diagram

Master Volume & Trim

Table 159: Master Volume & Trim Feature Description

Functional description	A three-stage gain element (Master Volume/Trim control) with a dynamic range from +24dB down to -83dB is implemented. Master Volume can be set in steps of 1/8dB. Trim can be set in steps of 1dB - internal 0.125dB steps are processed to avoid DC plops. The maximum speed of change is 62,5dB/s. Therefore a complete transition from mute to +24dB will take 1.7sec. This gain element is controlled via two control registers - Master Volume and Trim register. The values of these both registers are added, clipped to the max. range from +24dB down to -83dB and afterwards transferred to the three-stage gain element per channel.
Application	Master volume and trim is built in the Main, Subwoofer, Center, Left and Right Surround channels.
Control mechanism (change during operation)	1) An eleven bit wide Master Volume control field ('MASTVOL': '192dec' ->+24dB, '191dec' ->23.875dB, '0dec' ->0dB, '-671dec' ->-83.875dB, '-672dec' ->Mute) 2) An eight bit wide Trim control value for every channel (2's complement) xx -> MAIN, SW, C, S ('xxVOLL', 'xxVOLR', 'xxVOL': '24dec' ->+24dB, '23dec' ->+23dB, '0dec' ->0dB, '-84dec' ->Mute)
Sample rate	32kHz, 44.1kHz and 48kHz

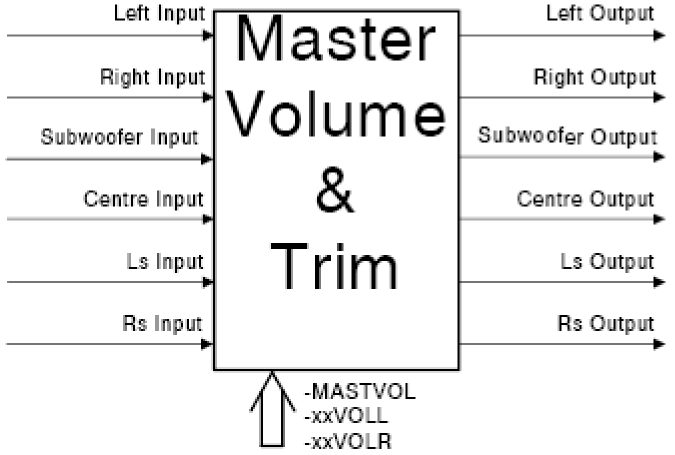


Fig 125. Master Volume & Trim Block Diagram

Volume and Balance for Auxiliary Channels

Table 160: Volume and Balance for Auxiliary Channels Feature Description

Functional description	A three-stage gain element (Volume control) with a dynamic range from +24dB down to -83dB is implemented. Volume left and right are settable in steps of 1dB - internal 0.125db steps are processed to avoid DC plops. The maximum of change is 62,5dB/s. Balance can be done by independent control of volume left and volume right (by microcontroller program).
Application	Volume is built in the AUX-Channels.
Control mechanism (change during operation)	1) An eight bit wide Volume control value for every channel (2's complement) xx -> AUX1, AUX2, AUX3, AUX4, AUX5, AUX6 ('xxVOLL', 'xxVOLR': '24dec' ->+24dB, '23dec' ->+23dB, '0dec' ->0dB, '-84dec' ->Mute)
Sample rate	32kHz, 44.1kHz and 48kHz

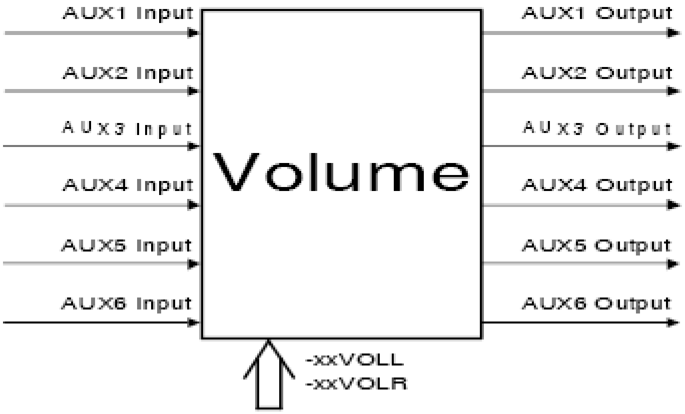


Fig 126. Volume and Balance for Auxiliary Channels Block Diagram

Level Adjust

Table 161: Level Adjust Feature Description

Functional description	A single stage gain element with a maximum gain of +15dB and a cut-down of -15dB. The mute position is defined at -16.
Application	Is built in all signal sources - DemDec, PIPMONO, IIS1..6 and ADC.
Control Mechanism [Set during start up]	1) A five bit wide control field. The value is interpreted as a 2's complement value. xx ->'DEC', 'MONO', 'SAP', 'PIPMONO', 'ADCL/A', ADCR/B' and 'IIS' ('xxLEV': '01111' ->+15dB, '00000' ->0dB, '10001' ->-15dB, '10000' ->Mute)
Sample rate	32kHz, 44.1kHz and 48kHz

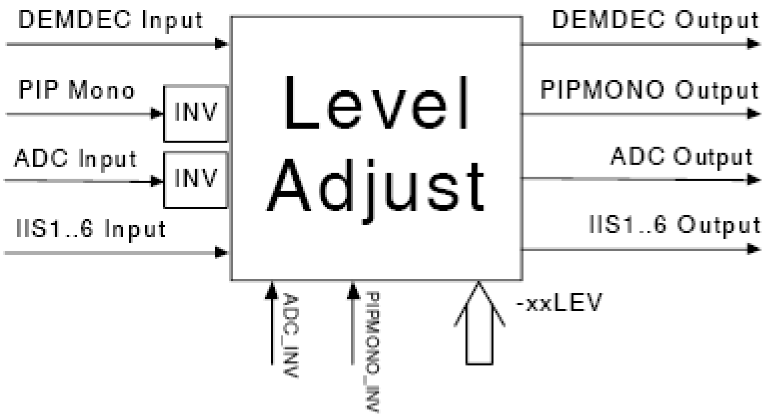


Fig 127. Level Adjust Block Diagram

Main Equalizer

Table 162: Main Equalizer Feature Description

Functional description	A five band parametric stereo or graphic stereo equalizer with pre-defined bands.
Application	Built in the Main channels. Can be used in two different modes: 1) Parametric equalizer with two bands(1&2) useable in the whole frequency range and three bands(3,4&5) useable in a range from 500Hz to maximum. 2) Graphic equalizer with pre-defined bands (100Hz, 300Hz, 1kHz, 3kHz, 8kHz).
Control Mechanism [Set during start up]	1) Switchable On/Off (EQENABLE': '0' -> Off, '1' ->On) 2) Setable as parametric or graphic equalizer per band[x] ('EQMAINxMODE': '0' ->Graphic Mode, '1' ->Parametric Mode)
Control Mechanism [Change during operation]	3) In graphic mode a five bit wide control field is defined for every band. The field is interpreted as a 2's complement value. Values larger then +12dB and lower then -12dB are clipped to +/-12dB. ('EQCHM1': '01100' ->+12dB, '00000' ->0dB, '10100' ->-12dB) for 100Hz Band1 ('EQCHM2': '01100' ->+12dB, '00000' ->0dB, '10100' ->-12dB) for 300Hz Band2 ('EQCHM3': '01100' ->+12dB, '00000' ->0dB, '10100' ->-12dB) for 1kHz Band3 ('EQCHM4': '01100' ->+12dB, '00000' ->0dB, '10100' ->-12dB) for 3kHz Band4 ('EQCHM5': '01100' ->+12dB, '00000' ->0dB, '10100' ->-12dB) for 8kHz Band5 4) In parametric mode one control register is defined for every band. The higher 12 bits are interpreted as the coefficient and the lowest three bits are interpreted as the address of the coefficient. Coefficient: ('EQMAINCOEFB1': '12 bit 2's complement') Band1 ('EQMAINCOEFB2': '12 bit 2's complement') Band2 ('EQMAINCOEFB3': '12 bit 2's complement') Band3 ('EQMAINCOEFB4': '12 bit 2's complement') Band4 ('EQMAINCOEFB5': '12 bit 2's complement') Band5 Address: ('EQMAINADRB1': '5 bit coefficient address') Band1 ('EQMAINADRB2': '5 bit coefficient address') Band2 ('EQMAINADRB3': '5 bit coefficient address') Band3 ('EQMAINADRB4': '5 bit coefficient address') Band4 ('EQMAINADRB5': '5 bit coefficient address') Band5 5) In parametric mode a copy bit is defined to enable the new coefficients per band[x]. This enable process is started by the transition from '0' to '1'. Afterwards it should be set to zero again by the micro controller. ('EQMAINxRELOAD': '0' -> '1' =>activate new coefficients)
Sample rate	32kHz and 48kHz

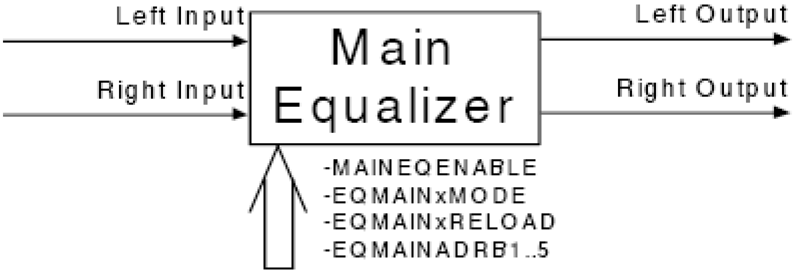


Fig 128. Main Equalizer Block Diagram

Central Equalizer

Table 163: Central Equalizer Feature Description

Functional description	A five band parametric mono or graphic mono equalizer with pre-defined bands.
Application	Built in the Center channel. Can be used in two different modes: 1) Parametric equalizer with two bands(1&2) useable in the whole frequency range and three bands(3,4&5) useable in a range from 500Hz to maximum. 2) Graphic equalizer with pre-defined bands (100Hz, 300Hz, 1kHz, 3kHz, 8kHz).
Control Mechanism [Set during start up]	1) Switchable On/Off (EQENABLE': '0' -> Off, '1' ->On) 2) Setable as parametric or graphic equalizer per band[x] ('EQCENTERxMODE': '0' ->Graphic Mode, '1' ->Parametric Mode)
Control Mechanism [Change during operation]	3) In graphic mode a five bit wide control field is defined for every band. The field is interpreted as a 2's complement value. Values larger then +12dB and lower then -12dB are clipped to +/-12dB. ('EQCHC1': '01100' ->+12dB, '00000' ->0dB, '10100' ->-12dB) for 100Hz Band1 ('EQCHC2': '01100' ->+12dB, '00000' ->0dB, '10100' ->-12dB) for 300Hz Band2 ('EQCHC3': '01100' ->+12dB, '00000' ->0dB, '10100' ->-12dB) for 1kHz Band3 ('EQCHC4': '01100' ->+12dB, '00000' ->0dB, '10100' ->-12dB) for 3kHz Band4 ('EQCHC5': '01100' ->+12dB, '00000' ->0dB, '10100' ->-12dB) for 8kHz Band5 4) In parametric mode one control register is defined for every band. The higher 12 bits are interpreted as the coefficient and the lowest three bits are interpreted as the address of the coefficient. Coefficient: ('EQCENTERCOEFB1': '12 bit 2's complement') Band1 ('EQCENTERCOEFB2': '12 bit 2's complement') Band2 ('EQCENTERCOEFB3': '12 bit 2's complement') Band3 ('EQCENTERCOEFB4': '12 bit 2's complement') Band4 ('EQCENTERCOEFB5': '12 bit 2's complement') Band5 Address: ('EQCENTERADRB1': '5 bit coefficient address') Band1 ('EQCENTERADRB2': '5 bit coefficient address) Band2 ('EQCENTERADRB3': '5 bit coefficient address s) Band3 ('EQCENTERADRB4': '5 bit coefficient address s) Band4 ('EQCENTERADRB5': '5 bit coefficient address s) Band5 5) In parametric mode a copy bit is defined to enable the new coefficients per band[x]. This enable process is started by the transition from '0' to '1'. Afterwards it should be set to zero again. ('EQCENTERxRELOAD': '0' -> '1' =>activate new coefficients)
Sample rate	32kHz and 48kHz

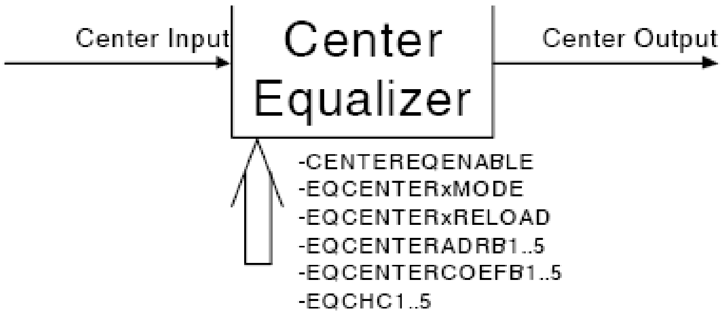


Fig 129. Central Equalizer Block Diagram

Soft Mute

Table 164: Soft Mute Feature Description

Functional description	A single stage gain element with a 32 step wide cosine curve from full scale to zero. After starting a mute or un-mute every 2 ms the gain is increased or decreased, therefore a whole mute or un-mute takes 64ms.
Application	Built in Main, Center, Subwoofer, Left and Right Surround and Aux1..6 channels. All softmutes can be controlled independently.
Control Mechanism [Change during operation]	1) Switchable On/Off ('MAINMUT': '0' ->Off - signal available, '1' ->On - no signal) *1) ('MAINLMUT': '0' ->Off - signal available, '1' ->On - no signal) *1) ('MAINRMUT': '0' ->Off - signal available, '1' ->On - no signal) *1) ('SWMUT': '0' ->Off - signal available, '1' ->On - no signal) ('CMUT': '0' ->Off - signal available, '1' ->On - no signal) ('LSMUT': '0' ->Off - signal available, '1' ->On - no signal) ('RSMUT': '0' ->Off - signal available, '1' ->On - no signal) ('AUX1MUT': '0' ->Off - signal available, '1' ->On - no signal) ('AUX2MUT': '0' ->Off - signal available, '1' ->On - no signal) ('AUX3MUT': '0' ->Off - signal available, '1' ->On - no signal) ('AUX4MUT': '0' ->Off - signal available, '1' ->On - no signal) ('AUX5MUT': '0' ->Off - signal available, '1' ->On - no signal) ('AUX6MUT': '0' ->Off - signal available, '1' ->On - no signal)
Sample rate	32kHz, 44.1kHz and 48kHz

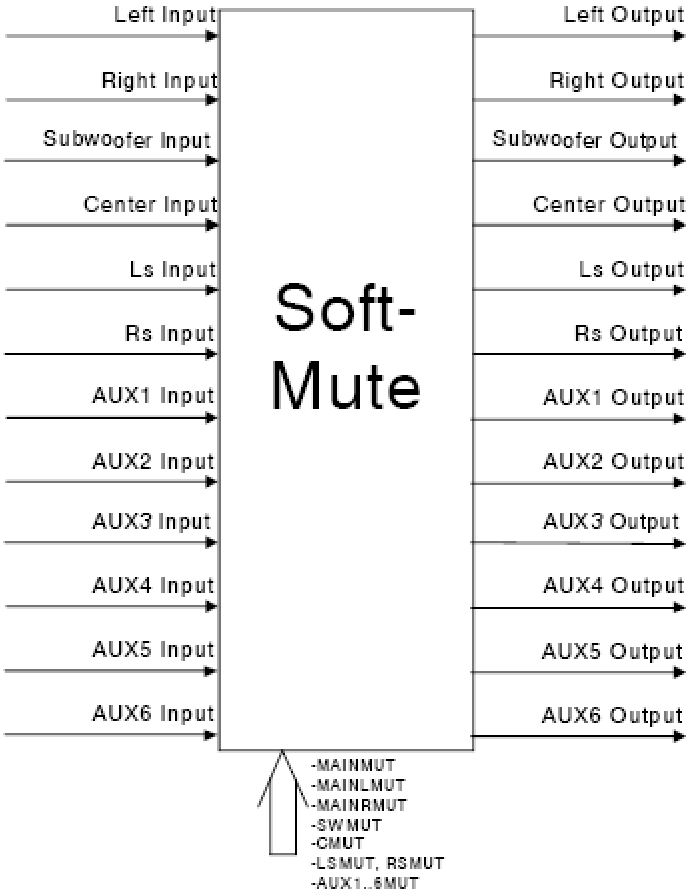


Fig 130. Soft Mute Block Diagram

Digital Input Crossbar

Table 165: Digital Input Crossbar Feature Description

Functional description	A general cross bar where all output channels can be connected to every input source. For main, Aux1..6 and monitor channels. Also a digital matrix is implemented. The digital matrix provides a special mode - the (A-B)/2 mode. This mode should be used for the audio monitor, to have an identification help for the decision if the incoming signal is mono or stereo. This may be used, if the signal is coming from an external source, which doesn't provide an identification.
Application	Built in Main, Center, Left and Right Surround, Aux1..6 and Monitor channel.
Control Mechanism [Change during operation]	'SRIN', 'MAIN', 'AUX1..6', 'AUDIOMON' ('xxSS': '00000' ->Dec) ('xxSS': '00001' ->Mono) ('xxSS': '00010' ->SAP) ('xxSS': '00011' ->PIPMONO) ('xxSS': '00100' ->ADC) ('xxSS': '00101' ->Noise/Silence) ('xxSS': '00110' ->OUTCOPY) ('xxSS': '00111'...'11111' ->IIS1..6) yy -> 'MAIN', 'AUX1..6', 'AUDIOMON' ('yyDM': '000' ->AB [Stereo]) ('yyDM': '001' ->(A+B)/2 [Mono]) ('yyDM': '010' ->AA [Lang.A]) ('yyDM': '011' ->BB [Lang.B]) ('yyDM': '100' ->BA [Swap]) ('yyDM': '101' ->(A-B)/2) ('yyDM': '110' ->Auto Lang.A) ('yyDM': '111' ->Auto Lang.B)
Sample rate	32kHz, 44.1kHz and 48kHz

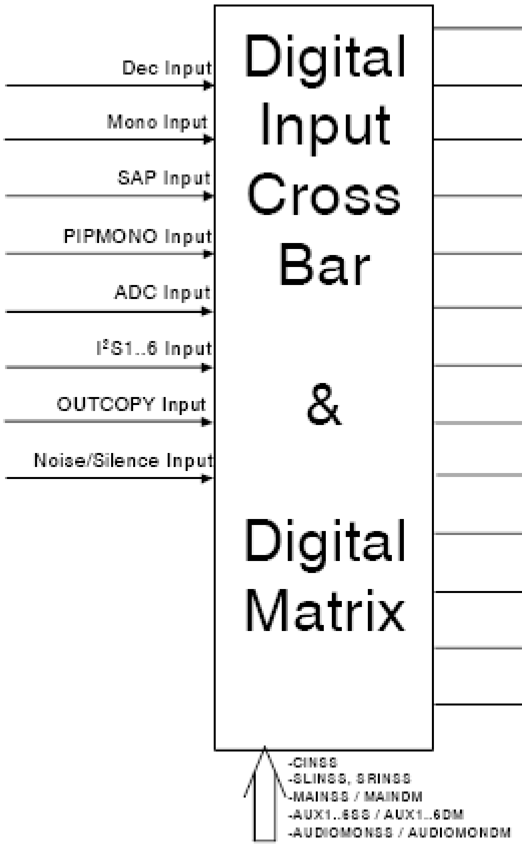


Fig 131. Digital Input Crossbar Block Diagram

Digital Output Crossbar

Table 166: Digital Output Crossbar Feature Description

Functional description	A general cross bar were all output channels can be connected to every input source.
Application	Built in for all DAFO1..8, OUTCOPY, I2S1..6 and DAC1..2 channels.
Control Mechanism [Change during operation]	xx -> 'DAFO1..8', 'DAC1L', 'DAC1R', 'DAC2L', 'DAC2R', 'IIS1..6L', 'IIS1..6R', 'ASMOCP1', 'ASMOCP2' ('xx': '00000' ->MAIN/L) ('xx': '00001' ->MAIN/R) ('xx': '00010' ->SW) ('xx': '00011' ->C) ('xx': '00100' ->Ls) ('xx': '00101' ->Rs) ('xx': '00110' ->AUX1L) ('xx': '00111' ->AUX1R) ('xx': '01000' ->AUX2L) ('xx': '01001' ->AUX2R) ('xx': '01010' ->AUX3L) ('xx': '01011' ->AUX3R) ('xx': '01100' ->AUX4L) ('xx': '01101' ->AUX4R) ('xx': '01110' ->AUX5L) ('xx': '01111' ->AUX5R) ('xx': '10000' ->AUX6L) ('xx': '10001' ->AUX6R) ('xx': '10010' ->Main Sum)
Sample rate	32kHz, 44.1kHz and 48kHz

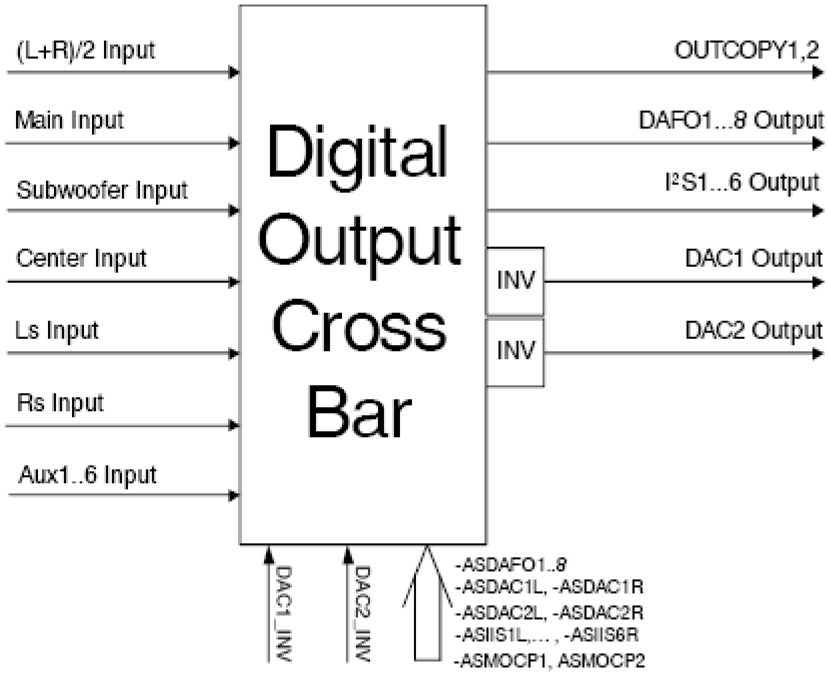


Fig 132. Digital Output Crossbar Block Diagram

Audio Monitor

Table 167: Audio Monitor Feature Description

Functional description	The audio monitor is able to monitor the sum (A+B)/2, left or right signal of all input channels of the digital cross bar. Via the "OUTCOPY" loop also every output channel of the output cross bar can be monitored. A special setting is the (A-B)/2 mode in the digital matrix. With this setting it is possible to identify a signal as a mono or stereo signal. If the audio monitor is used as a mono/stereo detector the quasi peak detection mode should be selected. The audio monitor provides three different modes: 1) Last sample, in this mode the last signal sample from the selected input is read out from the monitor register. 2) Peak detection, in this mode the highest signal sample since the last read command is read from the monitor register. 3) Quasi peak detection, in this mode a quasi peak detector is combined with the monitor. The quasi peak detector has an attack time of 4ms and a decay time of 1sec.
Application	The audio monitor is located in a separate monitor channel, the output of the measurement can be read from the register 'INF_AUD_LEV_MON_REG' as a 24 bit wide 2's complement value.
Control Mechanism [Change during operation]	AUDIOMONMODE': '00' ->Last sample, '01' ->Peak detection, '10' ->Quasi peak detection '11' ->Reserved)
Sample rate	32kHz, 48kHz

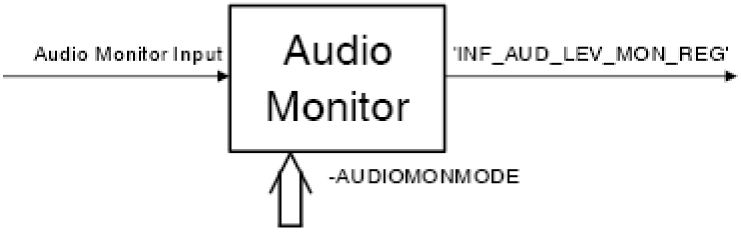


Fig 133. Audio Monitor Block Diagram

Beeper

Table 168: Beeper Feature Description

Functional description	Sine wave generator with adjustable amplitude in 1dB steps. The frequency is selectable in a range from 200Hz up to 12.5kHz.
Application	The beeper signal is added to the Main, Center and AUX1 channel.
Control Mechanism [Change during operation]	1) An eight bit wide control field for the volume setting. The value is interpreted as a 2's complement number. ('BEEPVOL': '0dec' ->0dBFS, '-83dec' ->-83dBFS, '-84dec' ->Mute/Off) 2) For the selectable frequencies a three bit wide control field is defined. ('BEEPFRE': '000' ->200Hz, '001' ->400Hz, '010' ->1000 Hz, '011' ->2000 Hz, '100' ->3000 Hz, '101' ->5000 Hz, '110' ->8000 Hz, '111' ->12.5kHz)
Sample rate	32kHz, 48kHz

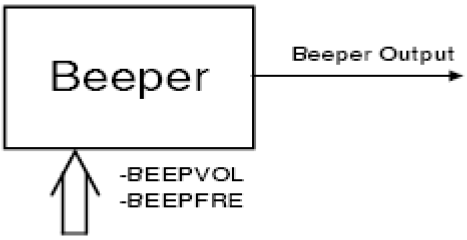


Fig 134. Beeper Block Diagram

Noise Generator

Table 169: Noise generator Description

Functional description	The Generator is producing bandpass weighted noise, with a center frequency of 500Hz (Fs=32kHz). This generator is defined by Dolby Labs. It is used for the level trim for Dolby.
Application	It is built in as a independent signal source, which can be selected by every processing channel via the digital cross bar. The trim procedure has to be provided by the micro controller sw.
Control Mechanism [Change during operation]	1) Controlled by Sound Mode
Sample rate	32kHz

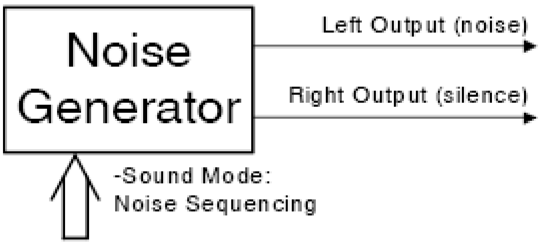


Fig 135. Noise generator Block Diagram

3.9.9.5 DUB/DBE Coefficients

Table 170: DUB - Coefficient

Address	Content	Remark
1-20	LP filter coef. 1-20	Filter Coefficient for 32kHz
21-40	BP filter coef. 1-20	
41-60	HP filter coef. 1-10	
61-80	LP filter coef. 1-20	Filter Coefficient for 44.1kHz
81-120	BP filter coef. 1-20	
121-130	HP filter coef. 1-10	
131-151	LP filter coef. 1-20	Filter Coefficient for 48kHz
151-170	BP filter coef. 1-20	
171-180	HP filter coef. 1-10	
181	AUB2_Headroom	read only
182	AUB2_KMLsb	
183	AUB2_KMMsb	
184	AUB2_KPLsb	
185	AUB2_KPMsb	
186	AUB2_ClippingLevel	
187	AUB2_AGCGainLsp	
188	AUB2_AGCGainMsp	
189-255	Reserved	

Table 171: DBE - Coefficient

Addr.	Content	Remark	Addr.	Content	Remark
1	GainDecrement Low	Control Variables	33	Boost filter coef. 1	Filter Coefficient for 44.1kHz
2	GainDecrement High		34	Boost filter coef. 2	
3	GainIncrement Low		35	Boost filter coef. 3	
4	GainIncrement High		36	Boost filter coef. 4	
5	Headroom (read only)		37	Boost filter coef. 5	
6	Boost Gain		38	Boost filter coef. 6	
7	HP filter coef. 1	Filter Coefficient for 48kHz	39	HP filter coef. 1	Filter Coefficient for 32kHz
8	HP filter coef. 2		40	HP filter coef. 2	
9	HP filter coef. 3		41	HP filter coef. 3	
10	HP filter coef. 4		42	HP filter coef. 4	
11	HP filter coef. 5		43	HP filter coef. 5	
12	HP filter coef. 6		44	HP filter coef. 6	
13	HP filter coef. 7		45	HP filter coef. 7	
14	HP filter coef. 8		46	HP filter coef. 8	
15	HP filter coef. 9		47	HP filter coef. 9	
16	HP filter coef. 10		48	HP filter coef. 10	
17	Boost filter coef. 1		49	Boost filter coef. 1	
18	Boost filter coef. 2		50	Boost filter coef. 2	
19	Boost filter coef. 3		51	Boost filter coef. 3	
20	Boost filter coef. 4		52	Boost filter coef. 4	
21	Boost filter coef. 5		53	Boost filter coef. 5	
22	Boost filter coef. 6		54	Boost filter coef. 6	
23	HP filter coef. 1	Filter Coefficient for 44.1kHz	55	DBE2_Front_Gain	Read Only
24	HP filter coef. 2		56	DBE2_AGCGainMax	Control Variable
25	HP filter coef. 3		57	Reserved	Not Used
26	HP filter coef. 4		58	Reserved	
27	HP filter coef. 5		59	Reserved	
28	HP filter coef. 6		60	Reserved	
29	HP filter coef. 7		61	Reserved	
30	HP filter coef. 8		62	Reserved	
31	HP filter coef. 9		63	Reserved	
32	HP filter coef. 10		64	Reserved	

Table 172: DBE - Coefficient Table for Maximum Boost

DBE2_AGC GainMax (dB)	Hex value
14	2CF
13	26C
12	21A
11	1D2
10	190
9	156

Table 172: DBE - Coefficient Table for Maximum Boost *…continued*

DBE2_AGC GainMax (dB)	Hex value
8	122
7	0F4
6	0CA
5	0A5
4	084
3	067
2	04C
1	034

Table 173: Acoustical Compensation - Coefficient

Addr.	Content	Remark
1	ACC_Main_a0	Main Channel (left & right)
2	ACC_Main_a1	
3	ACC_Main_b1	
4	ACC_Main_b2	
5	ACC_Main_a2	
6	ACC_Main_c2	
7	ACC_Main_c1	
8	ACC_Main_c0	
9	ACC_Main_d1	
10	ACC_Main_d2	
11-255	Reserved	
1	ACC_Center_a2	Center Channel
2	ACC_Center_a1	
3	ACC_Center_b1	
4	ACC_Center_b2	
5	ACC_Center_a2	
6	ACC_Center_c2	
7	ACC_Center_c1	
8	ACC_Center_c0	
9	ACC_Center_d1	
10	ACC_Center_d2	
11-255	Reserved	

3.9.9.6 Audio DSP registers

See Ref. 8 for detailed register descriptions.

3.10 Clock Generation Unit (CGU)

3.10.1 Introduction

All clocks for both AVIPs are generated by AVIP1. Clock generation registers in AVIP2 are reserved.

In the AVIP design, a central Clock Generation Unit (CGU) is used to generate all the required clock signals. This approach satisfies the clock requirements of the video, sound, and control subsystems, while minimizing the number of PLLs required. The primary clock input is an external clock input. PLLs and frequency dividers are used to generate the various clock signals required by the AVIP blocks.

3.10.2 AVIP Clock requirements

Table 174 to Table 177 show the clock signals used in the AVIP grouped by major subsystem.

Table 174: GTU Clock

Module	Clock name	Frequency MHz	Source	Comments
CGU	xin ll_clk i2s_sck_sys	27/13.5	external	xtal1 I/P clock line locked external input clock external I2S bit clock
PI2DTL	pi_clk	27/13.5/6.75	xtaldiv	PI_bus clock
GPR	gpr_clk	27/13.5/6.75	xtaldiv	general purpose register clock equal to pi_clk
PMRU	gpr_clk dcu_clk_pmru master_clk0_pmru i2d_clk_pmru itu_input_clk_pmru pi_clk_pmru snd_clk27_pmru snd_clk135_pmru snd_clk675_pmru	27/13.5/6.75 13.5 54 54 32.4/64.8/ll_pll/ll_clk 27/13.5/6.75 27 13.5 6.75	xtaldiv syspll syspll syspll syspll/llpll/ext xtaldiv syspll syspll syspll	pi_rst_hrd_n clock dcu_resetn clock vid_resetn clock i2d_resetn clock itu_resetn clock snd_clk_por_n clock snd_reset27 clock snd_reset135 clock snd_reset675 clock

Table 175: Control Subsystem Clock

Module	Clock name	Frequency MHz	Source	Comments
BS2CON	tck	12-15	JTAG port	boundary scan logic clock
PLU	tcb_tck	12-15	JTAG port	TCB and TCB scan register clock
VCB	gpr_clk clk_testshell	27/13.5/6.75 12-15	xtaldiv JTAG port	general purpose register clock equal to pi_clk CTAG isolation logic clock
I2C	pi_clk	27/13.5/6.75	xtaldiv	PI_bus clock
BCU	pi_clk	27/13.5/6.75	xtaldiv	PI_bus clock
AVIP	DCLK MPIFCLK ADAC_CLK	all frequencies 27/13.5 256*fs/128*fs	all sources syspll wspll	AVIP debug output clock output clock to PNX3000 output clock to external DACs

Table 176: Video Subsystem Clock

Module	Clock name	Frequency MHz	Source	Comments
DLINK	pi_clk	27/13.5/6.75	xtaldiv	PI_bus clock
	i2d_clk	54	syspll	module clock
	clk_testshell	12-15	JTAG port	CTAG isolation logic clock
	tst_rail_clk	all frequencies	all sources	test rail and bypass clock
	tcb_tck	12-15	JTAG port	TCB clock
Viddec	pi_clk	27/13.5/6.75	xtaldiv	PI_bus clock
	rsclk	27/13.5/6.75	xtaldiv	redundancy register clock only runs when pi_reset_n is active
	dto_clk	54/27	syspll	viddec system clock
	master_clk0	54	syspll	gated clock based on gen_llclk_y
	gated_llclk0	54	syspll	1FH: gated_llclk0 divided by 2
	gated_llclk1	54/27	syspll	2fh:gated clock based on gen_llclk_y
	gated_llclk2	27/13.5	syspll	gated clock based on gen_llclk_uv
	tst_rail_clk	all frequencies	all sources	CTAG isolation, test rail and bypass clock
	tcb_tck	12-15	JTAG port	TCB's clock
DCU	pi_clk	27/13.5/6.75	xtaldiv	PI_bus clock
	dcu_clk	13.5	syspll	data capture unit acquisition clock
	dcu_vid_clk	54	syspll	clock used to clock data in from DLINK
	tst_rail_clk	all frequencies	all sources	CTAG isolation, test rail and bypass clock
	tcb_tck	12-15	JTAG port	TCB clock
itu656	pi_clk	27/13.5/6.75	xtaldiv	PI_bus clock
	itu_input_clk	32.4/64.8/ll_pll/ll_clk	syspll/llpll/ext	CTAG isolation, test rail and bypass clock
	itu_viddec_clk	54	syspll	TCB clock
	tst_rail_clk	all frequencies	all sources	
	tcb_tck	12-15	JTAG port	
mbftime-base	vcr_clk	27	syspll	
	master_clk0	54	syspll	

Table 177: Sound Subsystem Clock

Module	Clock name	Frequency MHz	Source	Comments
Sound	pi_clk	27/13.5/6.75	xtaldiv	PI_bus clock
System	rsclk	27/13.5/6.75	xtaldiv	redundancy register clock only runs when pi_reset_n is active.
	snd_clk27	27	syspll	DEMDEC receive data clock
	snd_clk135	13.5	syspll	DEMDEC module
	snd_clk675	6.75	syspll	DECI module
	snd_clk128fsa	4.096-6.144	wspll	sample frequency = 32/44.1/48 KHz
	snd_clk128fsd	4.096-6.144	wspll	inverted snd_clk128fsa
	snd_clkep	100-170	turbopll	clock to Sound Core
	i2s_out_sck_mch	32fs/64fs	wspll	multi channel bitclock (32fs/64fs)
	snd_sck_sys	64fs	wspll	system bit clock
	clk_testshell	12-15	JTAG port	CTAG isolation logic clock
	tst_rail_clk	all frequencies	all sources	CTAG isolation, test rail and bypass clock
	tcb_tck	12-15	TAG port	TCB clock

3.10.3 Crystal Oscillator Specification

The AVIP contains a crystal oscillator which should be used with an external clock signal as shown in Figure 136.

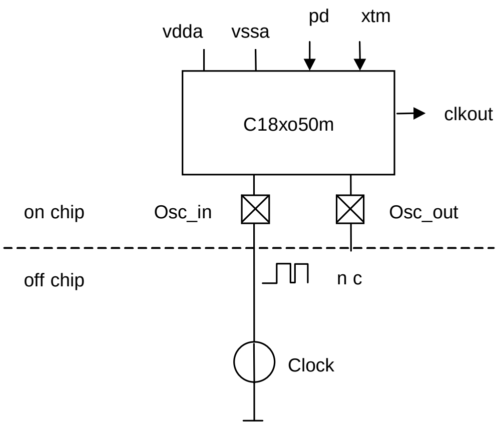


Fig 136.Application diagram - Slave/Test mode

The supported crystal/external clock frequencies are 27 MHz and 13.5 Mhz. The crystal oscillator is followed by a selectable divide-by-two frequency divider, giving three available clock frequencies as shown in Table 178.

Table 178: Primary Clock settings

Clock/Crystal Input	Divider Setting	Clock Frequency
27 MHz	x/1	27 MHz
27 MHz	x/2	13.5 MHz
13.5 MHz	x/1	13.5 MHz
13.5 MHz	x/2	6.75 MHz

Table 179 lists the clock signals that are generated directly from the crystal oscillator.

Table 179: Clocks Derived From Crystal

Output clock	Frequency MHz	Comment
gpr_clk	27/13.5/6.75	only used inside GTU
pi_clk	27/13.5/6.75	
pi_clk_pmru	27/13.5/6.75	only used inside GTU.
rsclk	27/13.5/6.75	

The crystal oscillator output is also used as the clock input to the WSPLL, TURBOPLL, and optionally, the SYSPLL and LLPLL.

3.10.4 Phase Locked Loops (PLL)

The AVIP contains four programmable PLLs. The output frequency of each PLL is controlled by three programmable frequency dividers, as shown in Figure 137.

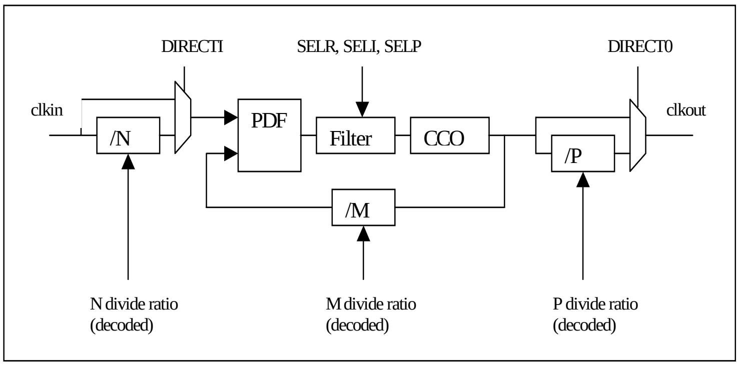


Fig 137.Simplified schematic of PLL

The input divider (N), output divider (P), and feedback divider (M) have programmable divide ratios. The N and P dividers can optionally be bypassed.

The operation of the PLL is controlled by software-programmable configuration bits in the general-purpose registers. The register locations and recommended settings for each PLL are given in “Clock Configuration and Status Registers”.

Table 180 gives the function of the PLL control signals.

Remark: The divider ratio signals are decoded values, not the direct binary representation of the desired divide ratio.

Table 180: PLL Control Signals

Signal	Description
GP_NDEC	decoded divide ratio for the input frequency divider
GP_PDEC	decoded divide ratio for the output frequency divider
GP_MDEC	decoded divide ratio for the feedback frequency divider
GP_WSPLLDIRECTI	bypass the input frequency divider

Table 180: PLL Control Signals *…continued*

Signal	Description
GP_WSPLLDIRECTO	bypass the output frequency divider
GP_WSPLLPD	Power down the PLL. The PD bit should be toggled to make changes to the GP_NDEC, GP_PDEC, GP_MDEC, GP_WSPLLDIRECTI, and GP_WSPLLDIRECTO bits take effect.
GP_SELR, GP_SELI, GP_SELP	select filter bandwidth

Table 181 presents a summary of the PLLs used in the AVIP design.

Table 181: Summary of PLLs

PLL	Description
GP_TURBOPLL	Generates clock for Sound Core DSPs. Frequency 100 - 170 MHz.
GP_WSPLL	Generates the BCLK and WS signals for the sound core I2S interface master mode. In slave mode, the external BCLK signal is used.
GP_LLPLL	Can optionally be used to generate the ITU output clock.
GP_SYSPLL	Generates a 162 MHz clock, which is synchronously divided down to produce most of the clock signals used in the AVIP.

The configuration of the PLLs and frequency dividers is illustrated in Figure 138.

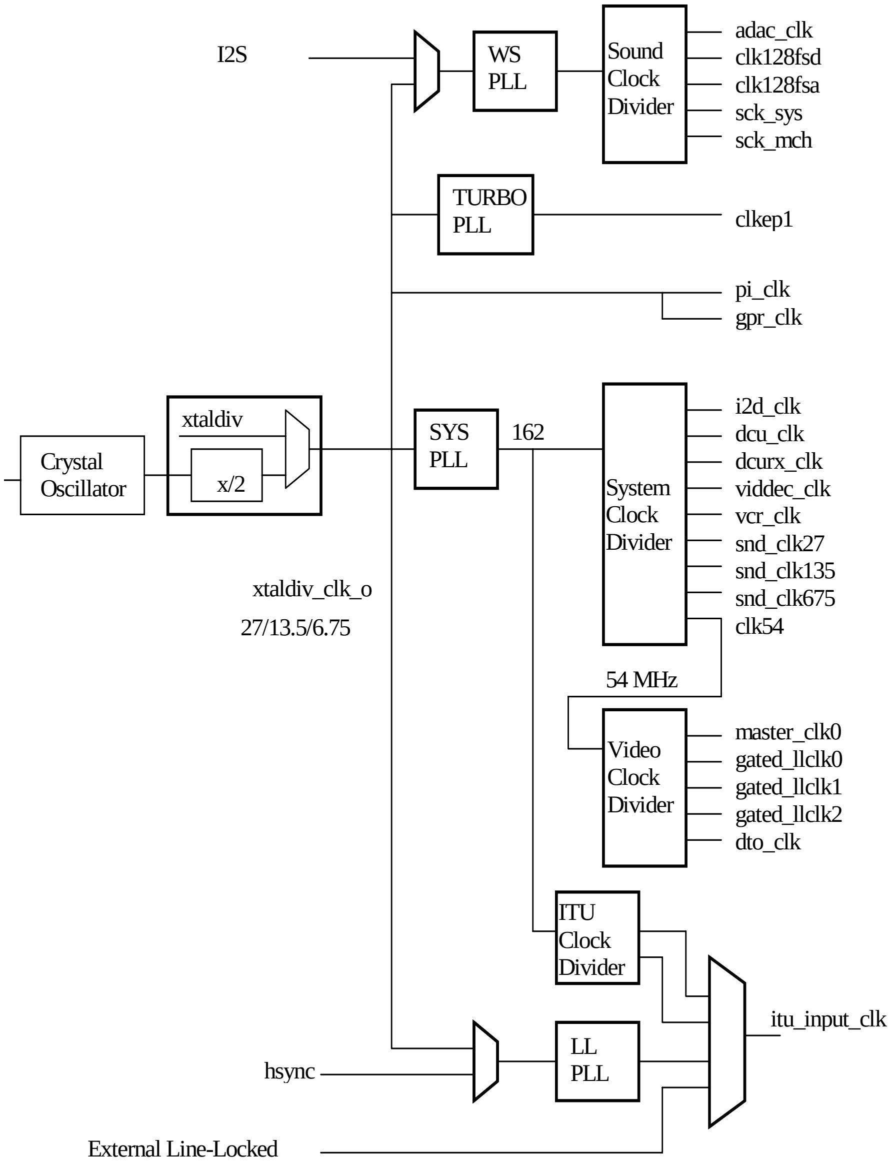


Fig 138.Block diagram - AVIP CGU

3.10.4.1 Power saving mode

The AVIP blocks can be put into a low-power standby mode by disabling all PLLs. In standby mode, the general purpose registers (located in the GTU) are still accessible via the PI bus and I^{2}C interface. Thus, the AVIP can be switched from standby mode to full-power mode by deasserting the PLL power-down bits in the control registers.

3.10.5 ITU output clock generation

The CGU provides several methods of generating the clock for the ITU656 digital video output interface. The ITU output clock generation is illustrated in Figure 139.

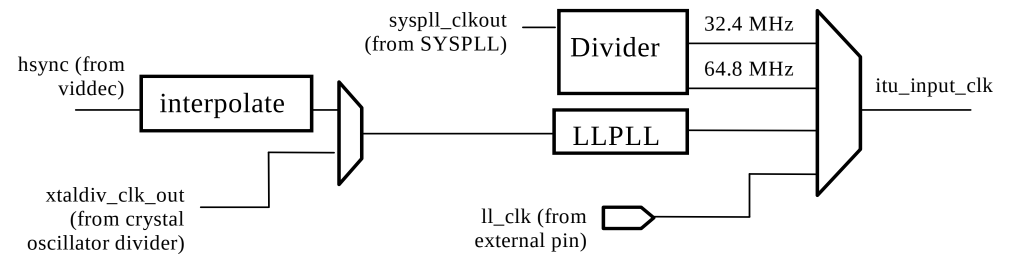


Fig 139.ITU Output Clock Generation

In normal operation, the ITU clock is selected from the 32.4 MHz clock and the 64.8 MHz clock signals from the SYSPLL. Alternatively, an externally generated line-locked clock or the clock generated by the LLPLL may be used. The LLPLL is an experimental feature and not recommended for normal use.

3.10.6 I^{2}S Word Select (WS) Clock Generation

The WS PLL and associated logic are used to generate the clock signals for the I^{2}S audio interface. The I^{2}S interface can operate in either master mode (I^{2}S clocks generated internally) or slave mode (locked to external I^{2}S clock). The I^{2}S Clock generator (WS PLL and associated clock dividers) support automatic sample rate detection.

The following table summarizes the available WS clock generation modes:

Table 182: WS clock generation modes

Mode	Control Bit Settings	Source for Divider Settings

Master	MASTERENABLE=1
	AUTOMASTERENABLE=0
	SLAVETYPE=don’t care
		GP_WSPLLCONTROL
		GP_WSPLLMASTERSEL

Slave	MASTERENABLE=0
	AUTOMASTERENABLE=0
	SLAVETYPE=0
		GP_WSSLAVEPLLCONTROL
		GP_WSPLLSLAVESEL

When in Master mode, the AVIP drives the I^{2}S clock signals, which are generated by the WS PLL, using the crystal oscillator as the reference clock.

In slave mode, the I^{2}S clock signals are generated externally, and act as inputs to the AVIP. The I^{2}S clock generator automatically detects the incoming WS frequency, which may be 32kHz, 44.1kHz, 48kHz, or out of range. The WS PLL divider settings depend on both the WS frequency and the bit clock frequency, because the bit clock is used as the WS PLL reference clock.

The following tables show the WS PLL configuration and status registers.

Table 183: GP_WSSLAVEPLLCONTROL

Bits	Name	Access Type	Description
31..17	RSD_31To17	Read Only	Reset value: 0x0 non-existent bits. Reads zero
16..7	GP_WSNDEC	Read/Write	Reset value: 0x2 n: Pre-divider value for Slave Mode
6..0	GP_WSPDEC	Read/Write	Reset value: 0x17 p: Post divider value for Slave Mode

Table 184: GP_WSPLLMASTERSEL

Bits	Name	Access Type	Description
31..30	RSD_31To30	Read Only	Reset value: 0x0 non-existent bits. Reads zero
29..26	GP_SELR	Read/Write	Reset value: 0x0 Pins to select bandwidth for Master mode
25..22	GP_SELI	Read/Write	Reset value: 0x2 Pins to select bandwidth for Master mode
21..17	GP_SELP	Read/Write	Reset value: 0x1f Pins to select bandwidth for Master mode
16..0	GP_MDEC	Read/Write	Reset value: 0x4022 m: Feedback divider for Master mode

Table 185: GP_WSPLLCONTROL

Bits	Name	Access Type	Description
31..27	RSD_31To27	Read Only	Reset value: 0x0 non-existent bits. Reads zero
26	GP_TESTMODEENABLE	Read/Write	Reset value: 0x0 Enables a testmode for testing the WS PLL with low frequencies
25	Reserved	-	Reserved
24	Reserved	-	Reserved
23	GP_MASTERENABLE	Read/Write	Reset value: 0x1 1: Master mode; 0: slave mode
22	GP_BYPASS256FS	Read/Write	Reset value: 0x0 Bypass incoming 256fs SYSCLKIN (0: PLL; 1: Bypass)
21	GP_BYPASS512FS	Read/Write	Reset value: 0x0 Bypass incoming 512fs SYSCLKIN (0: PLL; 1: Bypass)
20..11	GP_NDEC	Read/Write	Reset value: 0x2d n: Pre-divider for Master mode
10..4	GP_PDEC	Read/Write	Reset value: 0x17 p: Post divider for Master mode
3	GP_WSPLLDIRECTI	Read/Write	Reset value: 0x0 Direct to pin, 1: bypass of pre-divider
2	GP_WSPLLDIRECTO	Read/Write	Reset value: 0x0 Direct to pin, 1: bypass of post-divider
1	GP_WSPLLBYPASS	Read/Write	Reset value: 0x0 Bypass pin
0	GP_WSPLLPD	Read/Write	Reset value: 0x1 pd pin

Table 186: GP_WSPLLSTATUS

Bits	Name	Access Type	Description
31..5	RSD_31To5	Read Only	Reset value: 0x0 non-existent bits. Reads zero
4	GP_WSPLLFR	Read Only	Reset value: 0x0 fr (freeRunning) status (active high)
3	GP_WSPLLLOCK	Read Only	Reset value: 0x0 lock status (active high)

Table 186: GP_WSPLLSTATUS

Bits	Name	Access Type	Description
2	GP_WSPLLNACK	Read Only	Reset value: 0x0 nack pre divide acknowledge status (active high)
1	GP_WSPLLMACK	Read Only	Reset value: 0x0 mack feedback divide acknowledge status (active high)
0	GP_WSPLLPACK	Read Only	Reset value: 0x0 pack post divide acknowledge status (active high)

Table 187: GP_WS_FSCOUNTER

Bits	Name	Access Type	Description
31..9	RSD_31To9	Read Only	Reset value: 0x0 non-existent bits. Reads zero
8..0	GP_WSPLLFSCOUNT	Read Only	Reset value: 0x0 Counter to determine WSPLL divider ratio in slave mode

Table 188: GP_WS_SAMPLERATE

Bits	Name	Access Type	Description
31..2	RSD_31To2	Read Only	Reset value: 0x0 non-existent bits. Reads zero
1..0	GP_SAMPLERATE	Read Only	Reset value: 0x0 Detected WS sample rate 00: 32kHz, 01: 44.1kHz, 10:48kHz, 11: out of range

3.10.7 Clock Selection for 1fh and 2fh Video Modes

The AVIP video datapath supports both 1fh and 2fh video modes. Several clock signals are selected based on the video mode, as shown in the following table. These settings are controlled by registers in the GTU.

Table 189: Clock Selection

Clock Signal	1fh setting	2fh setting
ITU clock	32.4 MHz	64.8 MHz
DTO clock	27 MHz	54 MHz
PNX3000 clock	13.5 MHz	27 MHz

3.10.8 Clock Configuration and Status Registers

The following registers (located in the GPR section of the GTU) control the operation of the CGU.

Refer to Ref. 8 for full details of GPR registers.

Table 190: GP_CLKEN

Bits	Name	Access Type	Description
31..9	RSD_31To9	Read Only	Reset value: 0x0 non-existent bits. Reads zero
8	GP_MPIFCLKEN	Read/Write	Controls PNX3000 Clock. Reset value: 0x0 Clock enable '0' off, '1' on
7	GP_DCUCLKEN	Read/Write	Reset value: 0x0 Clock enable '0' off, '1' on
6	GP_DCURXCLKEN	Read/Write	Reset value: 0x0 Clock enable '0' off, '1' on
5	GP_ITU_CLKEN	Read/Write	Reset value: 0x0 Clock enable '0' off, '1' on
4	GP_I2DCLKEN	Read/Write	Reset value: 0x0 Clock enable '0' off, '1' on

Table 190: GP_CLKEN

Bits	Name	Access Type	Description
3	GP_VIDDECCLK54_1EN	Read/Write	Reset value: 0x0 Clock enable '0' off, '1' on
2	GP_DECI6P75CLKEN	Read/Write	Reset value: 0x0 Clock enable '0' off, '1' on
1	GP_DEMDEC13P5CLKEN	Read/Write	Reset value: 0x0 Clock enable '0' off, '1' on
0	GP_DEMDEC27CLKEN	Read/Write	Reset value: 0x0 Clock enable '0' off, '1' on

Table 191: GP_CLKSEL

Bits	Name	Access Type	Description
31..7	RSD_31To7	Read Only	Reset value: 0x0 non-existent bits. Reads zero
6	GP_LLPLLREFCLKSEL	Read/Write	Reset value: 0x0 0':hsync, '1':xtal
5	GP_ADACCLKSEL	Read/Write	Reset value: 0x0 0':128*fs, '1':256*fs
4	GP_MPIFCLKSEL	Read/Write	Controls PNX3000 Clock. Reset value: 0x0 '0':13.5MHz, '1':27MHz
3..2	GP_ITUCLKSEL	Read/Write	Reset value: 0x0 00: 64.8 MHz, 01: 32.4 MHz, 10: Line locked PLL, 11 line locked external input
1	GP_DTOFREQSEL_VID	Read/Write	Reset value: 0x0 '0': 27 MHz, '1' 54 MHz
0	GP_XTALCLKSEL	Read/Write	Reset value: 0x0 0' x_in, '1' x_in/2

Table 192: GP_DISTRICONTROL

Bits	Name	Access Type	Description
31..8	RSD_31To8	Read Only	Reset value: 0x0 non-existent bits. Reads zero
7	GP_SOFTRESETEPICS	Read/Write	Reset value: 0x0 Active high soft reset (min. two 6.75 MHz periods)
6	GP_SOFTRESET128FS	Read/Write	Reset value: 0x0 Active high soft reset 128fs (min. two 6.75 MHz periods)
5	GP_ADAC_CLKEN	Read/Write	Reset value: 0x0
4	GP_HALF_MCH	Read/Write	Reset value: 0x0 HALF_MCH (0: Full MCH speed; 1: Half MCH speed)
3	GP_INV128FSA	Read/Write	Reset value: 0x0 '1' Inverts CLK128FSA
2	GP_ENBCLK	Read/Write	Reset value: 0x0 '1' Enable for local bitclock
1	GP_EN128FS	Read/Write	Reset value: 0x0 '1' Enable for CLK128FSD & CLK128FSA
0	GP_ENEP	Read/Write	Reset value: 0x0 '1' Enable for CLKEP2

Table 193: GP_TURBOPLLSEL

Bits	Name	Access Type	Description
31..30	RSD_31To30	Read Only	Reset value: 0x0 non-existent bits. Reads zero
29..26	GP_TURBOSELR	Read/Write	Reset value: 0x0 Pins to select bandwidth
25..22	GP_TURBOSELI	Read/Write	Reset value: 0xd Pins to select bandwidth
21..17	GP_TURBOSELP	Read/Write	Reset value: 0x1f Pins to select bandwidth
16..0	GP_TURBOMDEC	Read/Write	Reset value: 0x1006 m: Feedback divider

Table 194: GP_TURBOPLLCONTROL

Bits	Name	Access Type	Description
31..21	RSD_31To21	Read Only	Reset value: 0x0 non-existent bits. Reads zero
20..11	GP_TURBONDEC	Read/Write	Reset value: 0x2 n: Pre-divider
10..4	GP_TURBOPDEC	Read/Write	Reset value: 0x42 p: Post divider
3	GP_TURBOPLLDIRECTI	Read/Write	Reset value: 0x0 TURBOPLL direct to pin. 1: bypass of pre-divider
2	GP_TURBOPLLDIRECTO	Read/Write	Reset value: 0x0 TURBOPLL direct to pin. 1: bypass of post-divider
1	GP_TURBOPLLBYPASS	Read/Write	Reset value: 0x0 TURBOPLL bypass pin
0	GP_TURBOPLLPD	Read/Write	Reset value: 0x1 TURBOPLL pd pin

Table 195: GP_TURBOPLLSTATUS

Bits	Name	Access Type	Description
31..5	RSD_31To5	Read Only	Reset value: 0x0 non-existent bits. Reads zero
4	GP_TUBOPLLFR	Read Only	Reset value: 0x0 fr (freeRunning) status (active high)
3	GP_TURBOPLLLOCK	Read Only	Reset value: 0x0 lock status (active high)
2	GP_TURBOPLLNACK	Read Only	Reset value: 0x0 nack pre divide acknowledge status (active high)
1	GP_TURBOPLLMACK	Read Only	Reset value: 0x0 mack feedback divide acknowledge status (active high)
0	GP_TURBOPLLPACK	Read Only	Reset value: 0x0 pack post divide acknowledge status (active high)

Table 196: GP_SYSPLLSEL

Bits	Name	Access Type	Description
31..30	RSD_31To30	Read Only	Reset value: 0x0 non-existent bits. Reads zero
29..26	GP_SYSSELR	Read/Write	Reset value: 0x0 Pins to select bandwidth
25..22	GP_SYSSELI	Read/Write	Reset value: 0x4 Pins to select bandwidth
21..17	GP_SYSSELP	Read/Write	Reset value: 0x7 Pins to select bandwidth
16..0	GP_SYSMDEC	Read/Write	Reset value: 0x200 m: Feedback divider

Table 197: GP_SYSPLLCONTROL

Bits	Name	Access Type	Description
31..21	RSD_31To21	Read Only	Reset value: 0x0 non-existent bits. Reads zero
20..11	GP_SYSNDEC	Read/Write	Reset value: 0x302 n: Pre-divider
10..4	GP_SYSPDEC	Read/Write	Reset value: 0x62 p: Post divider
3	GP_SYSPLLDIRECTI	Read/Write	Reset value: 0x1 SYSPLL direct to pin. 1: bypass of pre-divider
2	GP_SYSPLLDIRECTO	Read/Write	Reset value: 0x1 SYSPLL direct to pin. 1: bypass of post-divider
1	GP_SYSPLLBYPASS	Read/Write	Reset value: 0x0 SYSPLL bypass pin
0	GP_SYSPLLPD	Read/Write	Reset value: 0x1 SYSPLL pd pin

Table 198: GP_SYSPLLSTATUS

Bits	Name	Access Type	Description
31..5	RSD_31To5	Read Only	Reset value: 0x0 non-existent bits. Reads zero
4	GP_SYSPLLFR	Read Only	Reset value: 0x0 fr (freeRunning) status (active high)
3	GP_SYSPLLLOCK	Read Only	Reset value: 0x0 lock status (active high)
2	GP_SYSPLLNACK	Read Only	Reset value: 0x0 nack pre divide acknowledge status (active high)
1	GP_SYSPLLMACK	Read Only	Reset value: 0x0 mack feedback divide acknowledge status (active high)
0	GP_SYSPLLPACK	Read Only	Reset value: 0x0 pack post divide acknowledge status (active high)

3.10.9 Power-on and Reset

This section describes the reset mechanism of the AVIP blocks. The Power Management and Reset Unit (PMRU) provides selectable internally and externally generated power-on resets (POR). The PMRU filters out glitches of up to 240 ns and extends reset for 1.2 ms to allow the crystal oscillator and PLLs to stabilize. It also produces individual soft reset signals for each block in the AVIP blocks.

3.10.9.1 Reset Selection

The operation of the PMRU is controlled by two external pins: select and reset_n. Three modes of operation can be selected, as shown in the following tables. For normal operation, Internal POR Mode or External POR mode are used.

Table 199: Internal POR Mode (select=1, reset_n=0)

Reset_n	select	Vdd	Output
0	0	0	0 - [From POR with input from reset pin (=0)]
0	1	0	0 - [From internal POR]
0	1	1	1 - [From internal POR]

Table 200: External POR (select=0)

Reset_n	select	Vdd	Output
0	0	0	0 - [From POR with input from reset pin]
0	0	1	0 - [From POR with input from reset pin
1	0	1	1 - [From POR with input from reset pin

Table 201: POR Bypass Mode (select=reset_n)

Reset_n	select	Vdd	Output
0	0	0	0 - [From POR with input from reset pin]
0	0	1	0 - [From POR with input from reset pin]
1	1	1	1 - [Forced high via logic]

3.10.9.2 Reset Operation and Power Management

This section explains the sequence of power-up from initial power on, power down to standby and power up from standby modes.

Power on/External Hard Reset: The AVIP blocks enter a power-on/hard reset state when the internally or externally generated POR is asserted. The reset pulse then passes through the Glitch Filter. The filtered reset signal is extended by 1.2 ms during which the device is in standby mode. In this state all registers are assigned their reset values and all PLLs are switched OFF. The only clock running is the crystal clock. This allows the I^{2}C, GPR and BCU blocks to function.

The sequence to bring the device from standby to full power mode is as follows:

1. Configure and switch ON all PLLs
2. Enable and select clock frequencies
3. Remove software resets

Soft Resets: In addition to the power-on/hard reset, the PMRU also generates soft resets under the control of GPR settings. The following table shows the soft reset control bits:

Table 202: Soft Reset Control Bits

Register	Bit	Function
GP_DISTRICONTROL	GP_SOFTRESETEPICS	Active high soft reset (min. two 6.75 MHz periods) for EPICS DSPs in Sound Core
GP_DISTRICONTROL	GP_SOFTRESET128FS	Active high soft reset 128fs (min. two 6.75 MHz periods)
GP_RESETS	GP_DCU_RESET_N	Reset for DCU. Active low.
GP_RESETS	GP_VID_RESET_VD1_N	Viddec Core Reset. Active low.
GP_RESETS	GP_SND_CLK_POR_N	Sound Core master reset. Only release when clocks stable. Active low.
GP_RESETS	GP_I2D_RESET_N	Active Low DLINK reset
GP_RESETS	GP_SND_RESET27_N	Sound Core reset for 27MHz clock domain. Active low.
GP_RESETS	GP_SND_RESET135_N	Sound Core reset for 13.5MHz clock domain. Active low.
GP_RESETS	GP_SND_RESET675_N	Sound Core reset for 6.75MHz clock domain. Active low.
GP_RESETS	GP_ITU656_RESET_N	ITU656 Formatter Active low.

3.10.10 Interrupts

The AVIP blocks have a single external interrupt line. Interrupt conditions can be generated by the following modules:

- VIDDEC
- DLINK
- DCU
- BCU
- DEMDEC
- AUDIO

The external interrupt signal is the logical OR of the interrupt signals from all of the cores. A two-level interrupt status and control register scheme is used. The GTU contains top-level interrupt status and control registers that affect all interrupts generated by a given module. Each module that can signal interrupts contains a second level of status and control registers that deal with all the interrupt conditions generated by that module. The following procedure is used for interrupt handling:

1. Read top-level interrupt status register to determine which module or modules are signaling an interrupt.
2. Select highest priority module (if multiple interrupts are pending).
3. Read module-level interrupt status register to determine which interrupt condition is being signaled.
4. Handle interrupt.

3.10.10.1 Top-Level Interrupt Status and Control Registers

Table 203: GP_IRQ_STAT.
Interrupt status flag for each module

Bits	Name	Access Type	Description
31..7	RSD_31To7	Read Only	Reset value: 0x0 non-existent bits. Reads zero
6	GP_ITU656_INT	Read Only	Reset value: 0x0 interrupt status for FORMATTER
5	GP_VIDDEC_INT	Read Only	Reset value: 0x0 interrupt status for VIDDEC
4	GP_I2D_INT	Read Only	Reset value: 0x0 interrupt status for I^2D
3	GP_DCU_INT	Read Only	Reset value: 0x0 interrupt status for DCU
2	GP_BCU_INT	Read Only	Reset value: 0x0 interrupt status for BCU
1	GP_AUDIO_INT	Read Only	Reset value: 0x0 interrupt status for AUDIO DSP
0	GP_DEMDEC_INT	Read Only	Reset value: 0x0 interrupt status for DEMDEC DSP

Table 204: GP_IRQ_ENAB
Interrupt enable bits for each module

Bits	Name	Access Type	Description
31..7	RSD_31To7	Read Only	Reset value: 0x0 non-existent bits. Reads zero
6..0	Gp_irq_enab_b	Read/Write	Reset value: 0x0 enable one or more interrupt requests '0' off, '1' on

Table 205: GP_IRQ_CLR
Top-level interrupt clear bits

Bits	Name	Access Type	Description
31..7	RSD_31To7	Write Only	Reset value: 0x0 non-existent bits. Read returns error
6..0	Gp_irq_clr_b	Write Only	Reset value: 0x0 clear one or more interrupt requests '0' off, '1' on

[1] Setting the control bits in this register will not clear the external interrupt signal if the interrupt condition is still being signaled at the module level. Module interrupts are cleared at the interrupt source i.e. the CLEAR register in the module that generated the interrupt.

Table 206: GP_IRQ_SET
Top-level interrupt set bits.

Bits	Name	Access Type	Description
31..7	RSD_31To7	Write Only	Reset value: 0x0 non-existent bits. Read returns error
6..0	Gp_irq_set_t	Write Only	Reset value: 0x0 set one or more interrupt requests '0' off, '1' on

3.10.11 Miscellaneous Registers

The GP_MODULE_ID register contains the module ID of the GTU.

The following registers in the GTU are related to debug and/or experimental functions and are not used in normal operation. These registers are left at their power-up default values.

GP_DEBUGCFG

GP_VCBFUNC_OUT_H

GP_VCBFUNC_OUT_L

GP_VCBVERSION_OUT

GP_VCBFUNCTIONS_H

GP_VCBFUNCTIONS_L

GP_VCBCONTROL

GP_MTD_M_STAB

GP_DTM_M_STAB

GP_TIMEBASE_2

GP_TIMEBASE_1

RFU_28

RFU_22

GP_NCOUNTVAL

GP_LLPLLSEL

GP_LLPLLCONTROL

GP_LLPLLSTATUS

3.11 Standards, Modes and Settings

3.11.1 Video Standards

Table 207: PAL Standard

Standard	FH	AVIP Output Format	Specification	Jaguar Feature list V8.1	Display Mode
PAL-D/K	1	720x576@50i 625 lines	ITU-R BT.470-6	Video 165 + 160	625i
PAL-I	1	720x576@50i 625 lines	ITU-R BT.470-6	Video 166 + 160	625i
PAL-M	1	720x480@60i 525 lines	ITU-R BT.470-6	Video 167 + 160	525i
PAL-N	1	720x576@50i 625 lines	ITU-R BT.470-6	Video 168 + 160	625i
PAL-B/G	1	720x576@50i 625 lines	ITU-R BT.470-6	Video 174 + 160	625i
PAL-H	1	720x576@50i 625 lines	ITU-R BT.470-6	Video 160	625i
PAL443-60	1	720x480@60i 525 lines		Video 160	525i

Table 208: SECAM Standards

Standard	FH	AVIP Output Format	Specification	Jaguar Feature list V8.1	Display Mode
SECAM-D/K	1	720x576@50i 625 lines	ITU-R BT.470-6	Video 169 + 160	625i
SECAM-K1	1	720x576@50i 625 lines	ITU-R BT.470-6	Video 170 + 160	625i
SECAM-L	1	720x576@50i 625 lines	ITU-R BT.470-6	Video 171 + 160	625i

Table 208: SECAM Standards

Standard	FH	AVIP Output Format	Specification	Jaguar Feature list V8.1	Display Mode
SECAM-L1	1	720x576@50i 625 lines		Video 172 + 160	625i
SECAM-B/G	1	720x576@50i 625 lines	ITU-R BT.470-6	Video 163 + 160	625i
SECAM-H	1	720x576@50i 625 lines	ITU-R BT.470-6	Video 160	625i

Table 209: NTSC Standards

Standard	FH	AVIP Output Format	Specification	Jaguar Feature list V8.1	Display Mode
NTSC-M	1	720x480@60i 525 lines	ITU-R BT.470-6	Video 164 + 159	525i
NTSC-N	1	720x576@50i 625 lines		Video 159	625i
NTSC-Japan	1	720x480@60i 525 lines		Video 159	525i
NTSC443-60	1	720x480@60i 525 lines		Video 159	525i
NTSC443-50	1	720x576@50i 625 lines		Video 159	625i

Table 210: ATSC Standards

Standard	FH	AVIP Output Format	Specification	Jaguar Feature list V8.1	Display Mode
1080i@50Hz	2	720x1080@50i 1125 lines	SMPTE274M(6)		1080i@50Hz
1080i@60Hz	2	720x1080@60i 1125 lines	SMPTE274M(4)		1080i@60Hz
1152i@50Hz	2	720x1152@50i 1250 lines	AS4933.1-200x		1152i@50Hz
Note: ATSC input source (component video) can be YPrPb (sync on Y) or RGB (sync on external H, V)					

Table 211: NI Standards

Standard	FH	AVIP Output Format	Specification	Jaguar Feature list V8.1	Display Mode
NI@50Hz	1	720x576@50i 624 lines			624ni
NI@60Hz	1	720x480@60i 524 lines			524ni

Table 212: Component Video Standards

Standard	FH	AVIP Output Format	Specification	Jaguar Feature list V8.1	Display Mode	Synchronisation
YPrPb@50Hz	1	720x576@50i 625 lines			625i	Y or CVBS
YPrPb@60Hz	1	720x480@60i 525 lines	EIA-770.2-A(4)		525i	Y or CVBS
RGB@50Hz	1	720x576@50i 625 lines			625i	G or CVBS
YPrPb@50Hz	2	720x576@50p 625 lines			576p	Y
YPrPb@60Hz	2	720x480@60p 525 lines	EIA-770.2-A(8)		480p	Y
RGB@50Hz	2	720x576@50p 625 lines			576p	External H, V
RGB@60Hz	2	720x480@60p 525 lines	EIA-770.2-A(8)		480p	External H, V

3.11.2 Data Capture Standards

Table 213: Data Capture Standards

Data Type	FH	Line Captured	Specification	Jaguar Feature list V8.1	Display Mode
WST625	1	6-22, 318-335	ITU-R BT.653, ETS 300 706	UI 124	625i
WST525	1	10-21, 273-284	ITU-R BT.653	UI 124	525i
WSS625	1	23	ETS 300 294 V1.4.1, ITU-R BT.1119-2	UI 218	625i

Table 213: Data Capture Standards

Data Type	FH	Line Captured	Specification	Jaguar Feature list V8.1	Display Mode
WSS525	1	20, 283	EIAJ CPX-1204	UI 218	525i
CC625	1	22, 335	EIA 608B		625i
CC525	1	21, 284	EIA 608	UI 209	525i
VPS625	1	16	ETS 300 231	UI 213	625i
VITC525	1	14, 277			525i
VITC625	1	19, 332			625i
NABTS	1	10-21, 273-284			525i

3.11.3 Audio Standards

Table 214: Audio Standards

Standard	Ch.1 Freq. MHz	Ch.2 Freq. MHz	Country / Area	DemDec Main O/P	DemDec Mono O/P	DemDec SAP O/P	De-emphasis/ Noise Red.	Stereo / Dual Ident.
B/G A2	5.5 (FM)	5.742 (FM)	Germany, Netherlands, Italy, Austria, Switzerland, Malaysia, Australia, Israel, Saudi Arabia	FM1 / FM2 Dematrix	FM1			Y
B/G NICAM	5.5 (FM)	5.850 (NICAM)	Scandinavia, Spain, Belgium, New Zealand, Singapore	NICAM	FM1 (analogue)			Y
D/K A2 (1)	6.5 (FM)	6.258 (FM)		FM1 / FM2 Dematrix	FM1			Y
D/K A2 (2)	6.5 (FM)	6.742 (FM)		FM1 / FM2 Dematrix	FM1			Y
D/K A2 (3)	6.5 (FM)	5.742 (FM)		FM1 / FM2 Dematrix	FM1			Y
D/K NICAM	6.5 (FM)	5.850 (NICAM)	Eastern Europe, GUS, China	NICAM	FM1 (analogue)			Y
L NICAM	6.5 (AM)	5.850 (NICAM)	France	NICAM	AM1 (analogue)			Y
I NICAM	6.0 (FM)	6.552 (NICAM)	Great Britain, Hong Kong	NICAM	FM1 (analogue)			Y
M Korea	4.5 (FM)	4.742 (FM)	Korea	FM1 / FM2 Dematrix	FM1			Y
M BTSC (NTSC)	4.5 (FM)	MPX demod.	USA, Canada, Mexico, Brazil, Taiwan	FM1 / AM subcarrier dematrix	FM1 (baseband)	FM subcarrier	DBX on Sub or SAP	N/A
M BTSC (PAL)	4.5 (FM)	MPX demod.	Argentina	FM1 / AM subcarrier dematrix	FM1 (baseband)	FM subcarrier	DBX on Sub or SAP	N/A

Table 214: Audio Standards

Standard	Ch.1 Freq. MHz	Ch.2 Freq. MHz	Country / Area	DemDec Main O/P	DemDec Mono O/P	DemDec SAP O/P	De-emphasis/ Noise Red.	Stereo / Dual Ident.
M EIAJ	4.5 (FM)	MPX demod.	Japan	FM1 / FM subcarrier dematrix	FM1 (baseband)			Y
FM Radio (Europe)	10.7 or programmable	MPX demod.	Europe	FM1 / AM subcarrier dematrix	FM1 (baseband)		50us	N/A
FM Radio (USA)	10.7 or programmable	MPX demod.	USA	FM1 / AM subcarrier dematrix	FM1 (baseband)		75us	N/A

3.11.4 Display Modes

Table 215: Display Modes

Display Mode	625i	525i	576p	480p	1080i/50Hz	1080i/60Hz	1152i/50Hz	624ni	524ni
FH	1	1	2	2	2	2	2	1	1
Field Freq (Hz)	50	59.94	N/A	N/A	50	60	50	50 (typical)	60 (typical)
Frame Freq (Hz)	25	29.97	50	59.94	25	30	25	25 (typical)	30 (typical)
Line Freq (Hz)	15625	15734	31250	31468	28125	33750	31250	15625 (typical)	15734 (typical)
Total Video Lines/Frame	625	525	625	525	1125	1125	1250	624	524
Active Video Lines/Frame	576	480	576	480	1080	1080	1152	576	480
Total Pixels/Line	864	858	864	858	2640	2200	1536	864	858
Active Pixels/Line	720	720	720	720	1920	1920	1280	720	720
Pixel Clock (MHz)	13.5	13.5	27	27	74.25	74.25	48	13.5	13.5
Total Pixels/Line (From VIDDEC)	864	858	864	858	990	826	864	864	858
Active Pixels/Line (From VIDDEC)	720	720	720	720	720	720	720	720	720
VBI Lines - inclusive (From AVIP)	1-22 311-335 624-625	1-22 263-285	1-44 621-625	1-45	1-20 561-583 1124-1125	1-20 561-583 1124-1125	1-44 621-669 1246-1250	1-22 311-334 623-624	1-22 263-284

Table 215: Display Modes

Display Mode	625i	525i	576p	480p	1080i/50Hz	1080i/60Hz	1152i/50Hz	624ni	524ni
Field Indicator = 0	1-312	4-265	1-625	1-525	1-563	1-563	1-625	1-312	4-265
Field Indicator = 1	313-625	266-525, 1-3	N/A	N/A	564-1125	564-1125	626-1250	313-624	266-524, 1-3
Vsync Line (From VIDDEC)	3, 316	6, 269	3	9	3, 566	3, 566	3, 628	3, 315	6, 268

[1] 525i and 524ni use the NTSC line numbering system

3.11.5 ITU656 Formatter Settings

Table 216: ITU656 Formatter Settings

Display Mode	625i	525i	576p	480p	1080i/50Hz	1080i/60Hz	1152i/50Hz	624ni	524ni
config (0x000)	0x00008000	0x00008001	0x00008000	0x00008001	0x00008003	0x00008002	0x00008000	0x00008000	0x00008001
data ID: VBI (0x004)	0x15491154	0x15491154	0x15491154	0x15491154	0x15491154	0x15491154	0x15491154	0x15491154	0x15491154
data ID:HBI (0x008)	0x2AAAAAA	0x2AAAAAA	0x2AAAAAA	0x2AAAAAA	0x2AAAAAA	0x2AAAAAA	0x2AAAAAA	0x2AAAAAA	0x2AAAAAA
capture (0x00C)	0x0000526E	0x0000526A	0x00100000	0x00100000	0x00100000	0x00100000	0x00100000	0x0000526E	0x0000526A
vf_control (0x014)	0x00001271	0x0000120D	0x00003271	0x0000320D	0x00003465	0x00003465	0x000034E2	0x00001270	0x0000120C
vf_sync (0x018)	0x0013C003	0x0010D006	0x00003003	0x00009009	0x00236003	0x00236003	0x00274003	0x0013B003	0x0010C006
field_1 (0x01C)	0x00271138	0x00003109	0x00FFFFFF	0x00FFFFFF	0x00465233	0x00465233	0x004E2271	0x00270138	0x00003109
field_2 (0x020)	0x0000313C	0x0000610D	0x00003FFF	0x00009FFF	0x00003236	0x00003236	0x00003274	0x0000313B	0x0000610C
vbi_1 (0x024)	0x0014F136	0x0001620D	0x0002C26C	0x0002D20D	0x00014463	0x00014463	0x0002C4DD	0x0014E136	0x0001620C
vbi_2 (0x028)	0x0001626F	0x0011D106	0x00FFFFFF	0x00FFFFFF	0x00247230	0x00247230	0x0029D26C	0x0001626E	0x0011C106
vbi_3 (0x02C)	0x00FFF003	0x00FFF006	0x00FFF003	0x00FFF009	0x00FFF003	0x00FFF003	0x00FFF003	0x00FFF003	0x00FFF006
vbi_4 (0x030)	0x00FFF13C	0x00FFF10D	0x00FFFFFF	0x00FFFFFF	0x00FFF236	0x00FFF236	0x00FFF274	0x00FFF13B	0x00FFF10C

[1] The "data ID: VBI" values are set for the Jaguar system (PNX8550)

[2] The "data ID: HBI" values are nominal to ensure a timing code is not generated, 3D comb filter doesn't use the ANC header

Table 217: Settings - interfacing to 3D comb filter

Display Mode	625i	525i	1080i/50Hz	1080i/60Hz	624ni	524ni
config (0x000)	0x00008044	0x00008045	0x00008000	0x00008001	0x00008044	0x00008045
capture (0x00C)			0x0003BF60	0x1F100000		

Table 218: Settings for External syncs in Display Mode 1152i/50Hz

Display Mode	1152i/50Hz
vf_sync (0x018)	0x00275002
field_1 (0x01C)	0x004E2270
field_2 (0x020)	0x00002275
vbi_3 (0x02C)	0x00FFF002
vbi_4 (0x030)	0x00FFF275

3.11.6 Viddec Settings

Table 219: Viddec Settings

Display Mode	625i + 624ni	525i + 524ni	625i YPrPb	525i YPrPb	625i YPrPb SynconCVBS	525i YPrPb SynconCVBS	625i YC	525i YC	625i RGB	525i RGB	625i SOG RGB	525i SOG RGB	576p YPrPb	480p YPrPb	1080i/ 50Hz YPrPb	1080i/ 60Hz YPrPb	1152i/ 50Hz YPrPb
MUX0 (0x040)	0x00000100	0x00000100	0x00000104	0x00000104	0x00000100	0x00000100	0x00000108	0x00000108	0x00000100	0x00000100	0x00000104	0x00000104	0x00000104	0x00000104	0x00000104	0x00000104	0x00000104
AGC_Y_CYC_AMP (0x088)	0x81380080	0x81380080	0x61380080	0x61380080	0x61380080	0x61380080	0x81c00000	0x81c00000	0x61380080	0x61380080	0x61380080	0x61380080	0x61380080	0x61380080	0x61380080	0x61380080	0x61380080
AGC_CVBS_YYC_CONTROL (0x094)	0x03E101CD	0x03E101CD	0x03E101CD	0x03E101CD	0x03E101CD	0x03E101CD	0x03E101CD	0x03E101CD	0x03E101CD	0x03E101CD	0x03E101CD	0x03E101CD	0x03E101CD	0x03E101CD	0x03E101CD	0x03E101CD	0x03E101CD
AGC_Y_CYC_CONTROL (0x098)	0x036100E8	0x036100E8	0x0361015D	0x0361015D	0x0361015D	0x0361015D	0x00610500	0x00610500	0x0161015D	0x0161015D	0x0161015D	0x0161015D	0x0361015D	0x0361015D	0x0361015D	0x0361015D	0x0361015D
AGC_Y_CYC_TARGETS (0x09C)	0x010001FF	0x010001FF	0x0100013B	0x0100013B	0x0100013B	0x0100013B	0x0100013B	0x0100013B	0x0100013B	0x0100013B	0x0100013B	0x0100013B	0x0100013B	0x0100013B	0x0100013B	0x0100013B	0x00FF01FF
AGC_LOWER_GAIN_LIMITS (0x0A0)	0x11F508AB	0x11F508AB	0x0F5508AB	0x0F5508AB	0x0F5508AB	0x0F5508AB	0x0F5508AB	0x0F5508AB	0x0F5508AB	0x0F5508AB	0x0F5508AB	0x0F5508AB	0x0F5508AB	0x0F5508AB	0x0F5508AB	0x0F5508AB	0x11F508AB
AGC_UPPER_GAIN_LIMITS (0x0A4)	0x332E6ADB	0x332E6ADB	0x2BAE6ADB	0x2BAE6ADB	0x2BAE6ADB	0x2BAE6ADB	0x2BAE6ADB	0x2BAE6ADB	0x15DE6ADB	0x15DE6ADB	0x15DE6ADB	0x15DE6ADB	0x2BAE6ADB	0x2BAE6ADB	0x2BAE6ADB	0x2BAE6ADB	0x2BAE6ADB
FSTBLNK (0x0C0)	0x00001400	0x00001400	0x00001800	0x00001800	0x00001800	0x00001800	0x00001400	0x00001400	0x00001800	0x00001800	0x00001800	0x00001800	0x00001800	0x00001800	0x00001800	0x00001800	0x00001800
HV_INFO_1 (0x100)	0x00000060	0x00000060	0x00000060	0x00000060	0x00000060	0x00000060	0x00000060	0x00000060	0x00000060	0x00000060	0x00000060	0x00000060	0x0000002B	0x0000002C	0x00000004	0x0000004B	0x0000002A
HV_INFO_2 (0x104)	0x00000048	0x00000048	0x00000048	0x00000048	0x00000048	0x00000048	0x00000048	0x00000048	0x00000048	0x00000048	0x00000048	0x00000048	0x00000024	0x00000024	0x00000014	0x00000014	0x00000024
HV_INFO_3 (0x108)	0x01000080	0x01000080	0x01000080	0x01000080	0x01000080	0x01000080	0x01000080	0x01000080	0x01000080	0x01000080	0x01000080	0x01000080	0x01000040	0x01000040	0x0100002C	0x0100002C	0x0100002C
HV_INFO_4 (0x10C)	0x00000120	0x00000120	0x00000120	0x00000120	0x00000120	0x00000120	0x00000120	0x00000120	0x00000120	0x00000120	0x00000120	0x00000120	0x00000090	0x00000090	0x00000090	0x00000090	0x00000090

Table 219: Viddec Settings *…continued*

Display Mode	625i + 624ni	525i + 524ni	625i YPrPb	525i YPrPb	625i YPrPb SynconCVBS	525i YPrPb SynconCVBS	625i YC	525i YC	625i RGB	525i RGB	625i SOG RGB	525i SOG RGB	576p YPrPb	480p YPrPb	1080i/ 50Hz YPrPb	1080i/ 60Hz YPrPb	1152i/ 50Hz YPrPb
SUBPIX_PLL_SYNC0 (0x140)	x	x	x	x	x	x	x	x	x	x	x	x	0x00000200	0x00000200	0x006C0600	0x000C0600	0x004C0200
SUBPIX_PLL_SYNC1 (0x144)	x	x	x	x	x	x	x	x	x	x	x	x	0x00000407	0x00020407	0x00060407	0x00060407	0x00060005
SUBPIX_PLL_SYNC2 (0x148)	x	x	x	x	x	x	x	x	x	x	x	x	0x0009B403	0x00082404	0x0008BC04	0x0008BC04	0x0008BC04
SUBPIX_PLL_SYNC3 (0x14C)	x	x	x	x	x	x	x	x	x	x	x	x	0x0000C7A1	0x0000C7A7	0x000F33BE	0x0000A3BE	0x0000AF9F
DMSD_V_SYNC (0x184)	0x012F0150	0x00FE01D0	0x012F0150	0x00FE01D0	0x012F0150	0x00FE01D0	0x012F0150	0x00FE01D0	0x012F0150	0x00FE01D0	0x012F0150	0x00FE01D0	0x012F0150	0x00FE01D0	0x012F0150	0x00FE01D0	0x012F0150

Table 220: Settings when decoding SECAM

Display Mode	625i	525i	624ni	524ni
DMSD_FILTERS (0x190)	0x016A2DEB	0x016A2DEB	0x016A2DEB	0x016A2DEB

Table 221: Settings when decoding YC

Display Mode	625i	525i	624ni	524ni
DMSD_FILTERS (0x190)	0x016A8006	0x016A8006	0x016A8006	0x016A8006

Table 222: Additional settings - interfacing to 3D comb filter

Display Mode	625i	525i	624ni	524ni
DMSD_FILTERS (0x190)	0x016A0006	0x016A0006	0x016A0006	0x016A0006
DMSD_COL_DEC (0x18C)	0x00780200	0x00780200	0x00780200	0x00780200

Table 223: Additional settings using external HV syncs

Display Mode	576p RGB	480p RGB	1080i/50Hz RGB	1080i/60Hz RGB	1152i/50Hz RGB
MUX0 (0x040)	0x00000004	0x00000004	0x00000004	0x00000004	0x00000004
FSTBLNK (0x0C0)	0x00001800	0x00001800	0x00001800	0x00001800	0x00001800
SUBPIX_PLL_SYNC0 (0x140)	0x00000A00	0x00000A00	0x006C0E00	0x000C0E00	0x004C0A00
SUBPIX_PLL_SYNC3 (0x14C)	0x00001775	0x00001F7D	0x000E7793	0x000FF793	0x00000771

[1] Assumes V connected to vsync1, inverted V connected to vsync2 and H connected to hsync1

[2] Positive syncs settings: Vertical lock achieved on vsync1

Table 224: Negative syncs settings

Display Mode	576p RGB	480p RGB	1080i/50Hz RGB	1080i/60Hz RGB	1152i/50Hz RGB
MUX0 (0x040)	0x00000084	0x00000084	0x00000084	0x00000084	0x00000084
FSTBLNK (0x0C0)	0x00001810	0x00001810	0x00001810	0x00001810	0x00001810
SUBPIX_PLL_SYNC0 (0x140)	0x00000A00	0x00000A00	0x006C0E00	0x000C0E00	0x004C0A00
SUBPIX_PLL_SYNC3 (0x14C)	0x000117B5	0x0000EFB9	0x000EE7B1	0x000077B3	0x0000FFAF

[1] Vertical lock achieved on vsync2

Sync Position

The sync signals to the DCU come from the VidDec, but the VidDec introduces different delays depending on the VidDec processing mode as follows:

VidDec video delay:

- 0 line delay - monochrome or Columbus (3D comb) mode
- 1 line delay - NTSC mode
- 2 line delay - PAL mode

The VidDec sync output is timed to match the video output. Therefore the sync vertical position is dependent on the VidDec video mode.

3.11.7 DCU Register Settings

Table 225: DCU Register Settings

Data Type	WST625	WST525	WSS625	WSS525	VPS625	CC525	CC625	VITC625	VITC525	NABTS
DCR1 (0x000)	0x86402701	0x86402709	0x86402701	0x86402709	0x86402701	0x86402709	0x86402701	0x86402701	0x86402709	0x86402709
LCR2_5 (0x004)	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
LCR6_9 (0x008)	0x00000000	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
LCR10_13 (0x00C)	0x00000000	0x44444444	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xCCCCCCCC
LCR14_17 (0x010)	0x00000000	0x44444444	0xFFFFFFFF	0xFFFFFFFF	0xFF22FFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFAA	0xCCCCCCCC
LCR18_21 (0x014)	0x00000000	0x44444444	0xFFFFFFFF	0xFF77FFFF	0xFFFFFFFF	0x55FFFFFF	0xFFFFFFFF	0xFFFF99FF	0xFFFFFFFF	0xCCCCCCCC
LCR22_24 (0x018)	0xFFFFFF00	0xFFFFFFFF	0xFFFF33FF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFF11	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
DCR2 (0x02C) color (not YC) Monochrome + YC	0x00800444 0x00800244	0x00800644 0x00800544	0x00800444 0x00800244	0x008006440x00800544	0x00800444 0x00800244	0x00800644 0x00800544	0x00800444 0x00800244	0x00800444 0x00800244	0x00800644 0x00800544	0x00800644 0x00800544

Table 226: Settings - Interfacing to 3D comb filter

Data Type	WST625	WST525	WSS625	WSS525	VPS625	CC525	CC625	VITC625	VITC525	NABTS
DCR2 (0x02C) color	0x00800244	0x00800544	0x00800244	0x00800544	0x00800244	0x00800544	0x00800244	0x00800244	0x00800544	0x00800544

3.11.8 DLINK Settings

Table 227: DLINK Settings

Display Mode	625i (Except YC)	525i (Except YC)	576p	480p	1080i/50Hz	1080i/60Hz	1152i/50Hz	624ni (Except YC)	524ni (Except YC)
RX_CTRL (0x000)	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
REC DEMUX MODE (0x018)	0x0001FFF9	0x0001FFF9	0x0001FFFA	0x0001FFFA	0x0001FFFA	0x0001FFFA	0x0001FFFA	0x0001FFF9	0x0001FFF9

Table 228: I^{2}D Settings (YC)

Display Mode	625i YC	525i YC	624ni YC	524ni YC
RX_CTRL (0x000)	0x00000000	0x00000000	0x00000000	0x00000000
REC DEMUX MODE (0x018)	0x0001FFF8	0x0001FFF8	0x0001FFF8	0x0001FFF8

3.11.9 GTU Settings

Table 229: Crystal Divider Settings

	GP_XTALCLKSEL	Input Crystal/Clock frequency	Xtal_clk frequency
GP_CLKSEL[0] (0x004)	0	27 MHz	27 MHz
GP_CLKSEL[0] (0x004)	1	27 MHz	13.5 MHz
GP_CLKSEL[0] (0x004)	0	13.5 MHz	13.5 MHz
GP_CLKSEL[0] (0x004)	1	13.5 MHz	6.75 MHz

[1] For initialisation procedure to set-up and enable clocks refer to INITIALISATION worksheet.

Table 230: SYSPLL and TURBOPLL Divider Settings

Xtal clk	27MHz	13.5MHz (default)	6.75MHz
GP_SYSPLLCONTROL (0x038)	0x0018162C	0x0018162C	0x0018162C
GP_SYSPLLSEL (0x034)	0x00880008	0x010E0200	0x01DA00C0
GP_TURBOPLLCONTROL (0x02C)	0x00016420	0x00001420	0x00101420
GP_TURBOPLLSEL (0x028)	0x037E1006	0x037E1006	0x037E1006

Table 231: WSPLL Divider Settings - Auto Master Mode

Xtal clk	27MHz	13.5MHz (default)	6.75MHz
Audio sample rate = 48KHz			
GP_WSPLLCONTROL (0x038)	0x030659D0	0x030149D0	0x030509D0
GP_WSPLLMASTERSEL (0x010)	0x00BE5B69	0x00BE5B69	0x00BE5B69
Audio sample rate = 44.1kHz			
GP_WSPLLCONTROL (0x038)	0x03077030	0x030550E0	0x030550E0
GP_WSPLLMASTERSEL (0x010)	0x00BE29ED	0x00BE29ED	0x00BE6EB5

Table 231: WSPLL Divider Settings - Auto Master Mode

Xtal clk	27MHz	13.5MHz (default)	6.75MHz
Audio sample rate = 32kHz			
GP_WSPLLCONTROL (0x038)	0x030149D0	0x030509D0	0x030519D0
GP_WSPLLMASTERSEL (0x010)	0x013E1200	0x013E1200	0x013E1200

Table 232: WSPLL Divider Settings - Master Mode

Xtal clk	27MHz	13.5MHz (default)	6.75MHz
Audio sample rate = 48KHz			
GP_WSPLLCONTROL (0x038)	0x008659D0	0x008149D0	0x008509D0
GP_WSPLLMASTERSEL (0x010)	0x00BE5B69	0x00BE5B69	0x00BE5B69
Audio sample rate = 44.1kHz			
GP_WSPLLCONTROL (0x038)	0x00877030	0x008550E0	0x008550E0
GP_WSPLLMASTERSEL (0x010)	0x00BE29ED	0x00BE29ED	0x00BE6EB5
Audio sample rate = 32kHz			
GP_WSPLLCONTROL (0x038)	0x008149D0	0x008509D0	0x008519D0
GP_WSPLLMASTERSEL (0x010)	0x013E1200	0x013E1200	0x013E1200

Table 233: Clock Settings for 1fh/2fh video modes

	Control Bit	1FH	2FH
GP_CLKSEL[1] (0x004)	GP_DTOFREQSEL_VID	0	1
GP_CLKSEL[3:2] (0x004)	GP_ITUCLKSEL	01	00
GP_CLKSEL[4] (0x004)	GP_MPIFCLKSEL	0	1

Table 234: Enables/Resets

GP_CLKEN (0x000)	0x000001FF
GP_DISTRICONTROL (0x00C)	0x00000027

3.11.10 BCU Settings

Table 235: BCU Settings

TOUT (0x018)	0x00000800

[1] Timeout after 2048 PI bus cycles. This is slightly longer than two audio samples at the lowest sample rate and highest PI bus clock frequency.

3.11.11 PNX3000 settings

Table 236: Recommended PNX3000 to AVIP video settings

Display Mode	1FH CVBS	1FH RGB	1FH YC	1FH YPrPb	480p 576p 1152i RGB	480p 576p 1152i YPrPb	1080i RGB	1080i YPrPb
Data link mode (0x07)	0x00	0x00	0x00	0x00	0x01	0x01	0x11	0x11
Video switches 0 (0x08)	0x01	0x02	0x0B	0x02	0x02	0x02	0x02	0x02
Video switches 2 (0x0A)	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
RGB switches (0x0B)	0x22	0x62	0x22	0x52	0x62	0x52	0x62	0x52

[1] Setting shown were used during AVIP M1 validation using a JBS or AMB/ADB

[2] CVBS source uses Scart 1 connector (Bottom)

[3] RGB/YPrPb source uses Scart 2 connector (Top)

[4] YC source uses SVHS in connector

3.11.12 Analogue domain to AVIP 656 interface

The mapping for VBI data from the analogue domain to the ITU656 stream for AVIP is shown in Table 237.

The PNX2015 DCU always delays the VBI data by 1 line.

Table 237: VBI data mapping

AVIP in PAL Mode				
	PNX8550 (Color) mode		Columbus / PNX8550 (Mono) mode	
	Analog Line	Digital Line	Analog Line	Digital Line
Teletext	6 to 22 318 to 335	5 to 21 317 to 334	6 to 22 318 to 335	7 to 22, 311 319 to 335, 624
VPS	16	15	16	17
WSS	23	22	23	311 or 312 See Note 1
CC	22 and 335	21 and 334	22 and 335	311 and 624
AVIP in NTSC Mode				
	PNX8550 (Color) mode		Columbus / PNX8550 (Mono) mode	
	Analog Line	Digital Line	Analog Line	Digital Line
Teletext	10 to 21 273 to 284	10 to 21 273 to 284	10 to 21 273 to 284	11 to 22 274 to 285
WSS	20 and 283	20 and 283	20 and 283	21 and 284
CC	21 and 284	21 and 284	21 and 284	22 and 285
GEMSTAR	10 to 25	10 to 22, 263 to 265	10 to 25	11 to 22, 263 to 266

[1] If both Teletext and WSS are being transmitted WSS will occur in line 312, else it will occur in line 311

[2] AVIP decoding PAL colour = 2 line delay in VIDDEC

[3] AVIP decoding NTSC colour = 1 line delay in VIDDEC

[4] AVIP decoding PAL/NTSC monochrome or Columbus mode = 0 line delay in VIDDEC

[5] PNX2015/AVIP 656 formatter config reg Bit2 Columbus

4. Columbus (3D comb filter and noise reduction)

4.1 Functional overview

The Columbus comb filter is a combination of 3D comb filter for PAL and NTSC and spatial/temporal noise reduction for both color and luminance signals.

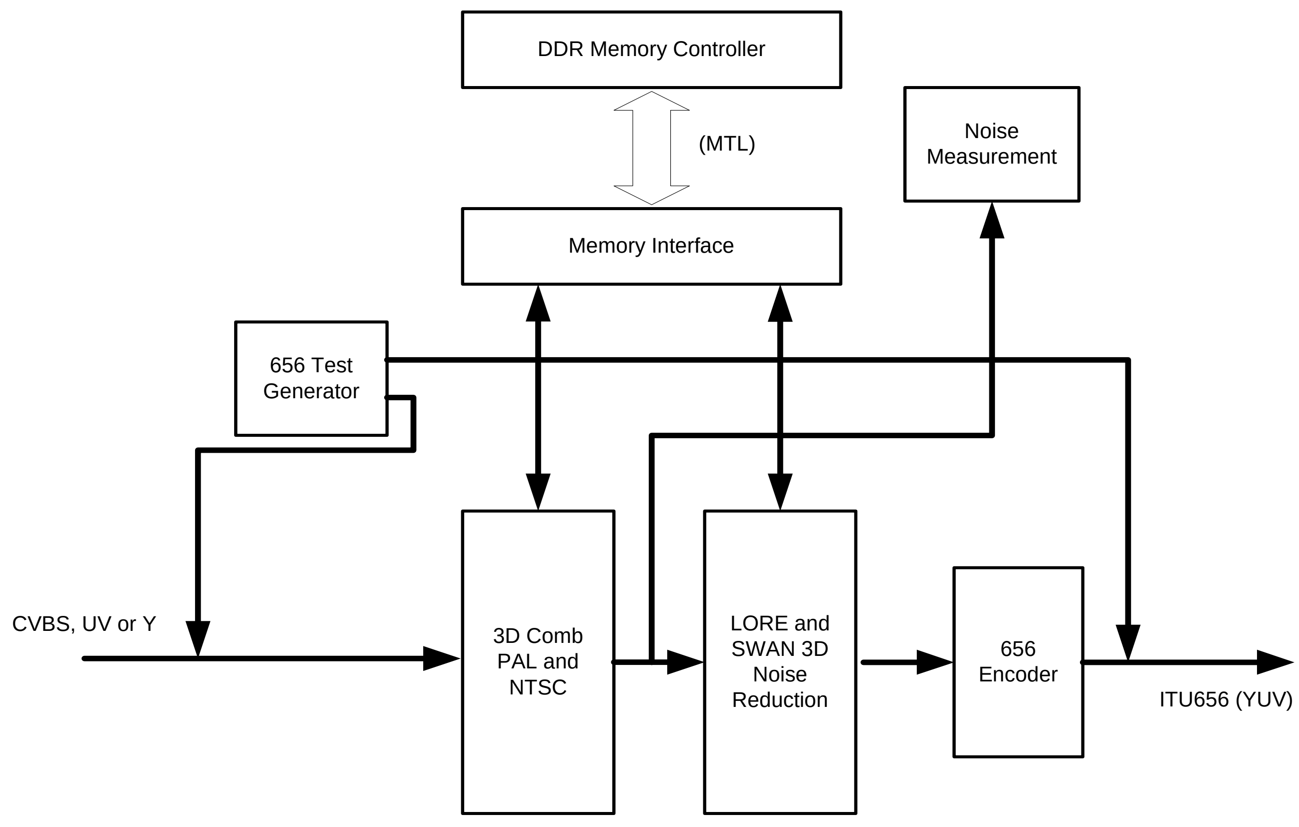


Fig 140. Columbus comb filter block diagram

Figure 140 shows a block diagram of the Columbus comb filter in the PNX2015 device. An input video signal is supplied by the AVIP and fed to the Columbus block. The signal is supplied in digitised components of:

- CVBS or Y
- Uncombed U
- Uncombed V

The CVBS signal is comb filtered, extracting the luminance components and rejecting the chroma components. The UV signals are comb filtered rejecting the left over luminance components, from a previous filtering (normally band pass filtered).

The outputs from the 3D comb filter are:

- Combed luminance signal (Y)

- Combed U-signal
- Combed V-signal

The output from the 3D comb filter feeds the SWAN and LORE noise reduction block, which performs spatial/temporal noise reduction, for both luminance and chrominance components.

4.1.1 3D comb filter

The 3D comb filter features several programmable modes of operation:

- Bypass mode

  In this mode, the CVBS, U and V signals are just bypassed to the output[4]
- Band-pass-Notch mode

  Used only for non standard signals, e.g. VCR signals.
- 2D comb filter modes
- 3D comb filter mode

Figure 141 shows a detailed block diagram of the 3D comb filter in Columbus:

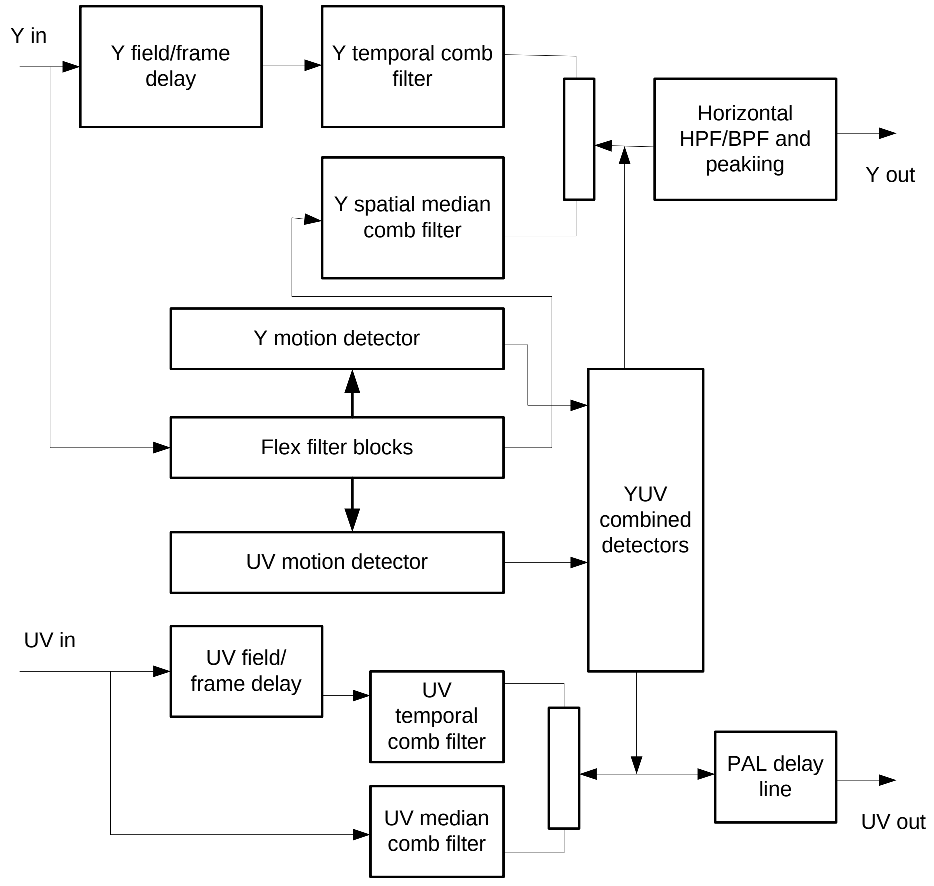


Fig 141. 3D comb filter block diagram

[4] The bypass mode has to be used whereas RGB, YprPb and SECAM signals are used at the front end of the PNX2015.

Columbus has two inputs:

- CVBS or Y - from which a clean luminance (Y) is extracted.
- UV components - from which a clean U and V signal is extracted.

The frame delay is used for the frame comb filter mode. An external DDR SDRAM memory externally connected to the PNX2015 will provide this delay. Frame comb filtering is impossible in case the external memory is not connected.

Phase correction at the inputs of the spatial/temporal comb filter is needed to restore the phase relation between the line frequency and the colour sub-carrier.

The CVBS input signal is also fed to a horizontal flex filter block, where horizontal band pass filtering can be set. The characteristic of the filter is adaptable and a number of possible filters can be set (see Section 4.1.2). The setting of this filter block greatly influences the performance of the comb filter. The output is used as input for the several detectors, and as input for the Y spatial comb filter. The flex filter will provide the band pass characteristic and the complementary notch characteristic.

Motion detectors are used in the frame mode to control the fader position as described above. There are two motion detectors:

- The Y motion detector (luminance motion detector) detects motion in the luminance signal.
- The C motion detector (chroma motion detector) detects motion in the chroma signal.

The output of the detectors are combined and filtered and result in the fader control signals Fader Y and Fader C (identical signals). Depending on the Fader Y and Fader C signal, the output of the 3D comb filter is a mix of temporally/spatially filtered signals.

A fully adaptable flex filter, a second time, horizontally filters the output of the luminance fader.

The delta and offset block receives as its input the motion dependant 3D comb filtered signal that is an ‘as clean as possible’ chroma signal. That signal is, in the delta and offset block, subtracted from the CVBS signal which produces the ‘as clean as possible’ luminance signal. The black level of the luminance signal is restored and the result is output. The black level restoration is corrected continuously. However on VCR signals, this restoration can become unstable. Therefore, on VCR signals, a fixed black level restoration value must be used.

A horizontal dynamic peaking can be carried out on the luminance signal. This peaking is adaptive in order not to amplify any cross luminance distortion. It detects where there may be leftover sub carrier in the luminance signal and reduces the peaking at that point.

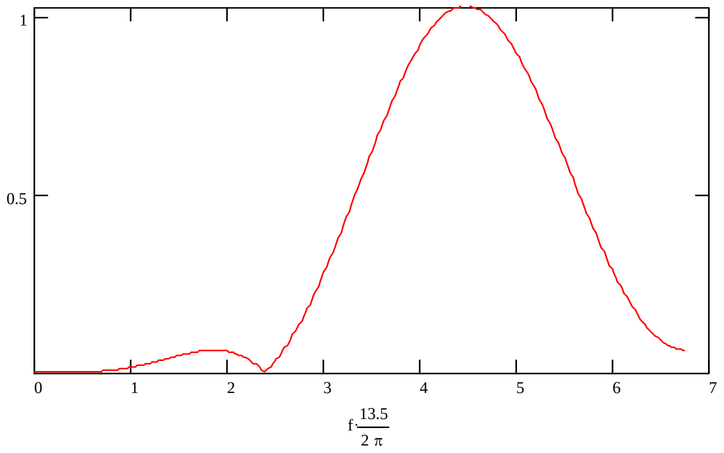
There are also two bypass blocks with the obvious purpose of bypassing the input signal, in case no comb filtering is wanted. Note that when the luminance path in the comb filter is bypassed, there is no black level restoration. Amplitude of the output signal level are set to a value of 205 (COMB_S00 register) in Normal mode. In Bypass mode it is set to 255.

The UV path of the 3D comb filter directly receives the UV signals from the input. There are no additional horizontal filters in this path. The C-fader signal is in fact the same signal as the Y-fader signal. The output of the fader is an ‘as clean as possible’ chroma signal.

4.1.2 Examples of spatial/temporal filter curves

13.5MHz sample rate

For a sample frequency of 13.5 MHz, the following filters are possible:



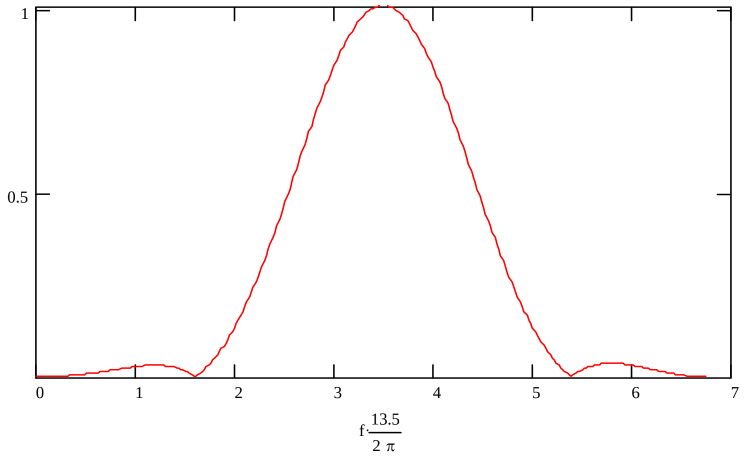
Filter 1 - [1,2,-4,-4,7,-4,-4,4,1]

Filter 2 - bypass

Filter 3 - [-1,2,-1]

Output - direct

Fig 142.Band-pass filter 4.43 MHz, normal width



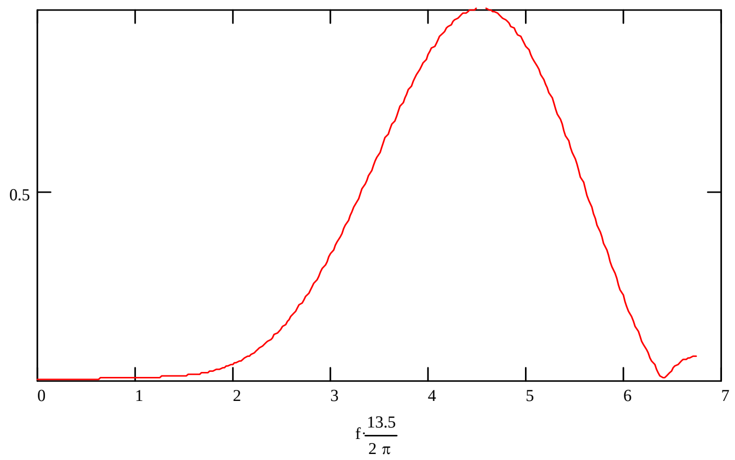
Filter 1 - [1,1,-5,0,4,0,-5,1,1]

Filter 2 - bypass

Filter 3 - [-1,0,2,0,-1]

Output - direct

Fig 143.Band-pass filter 3.58MHz, normal width



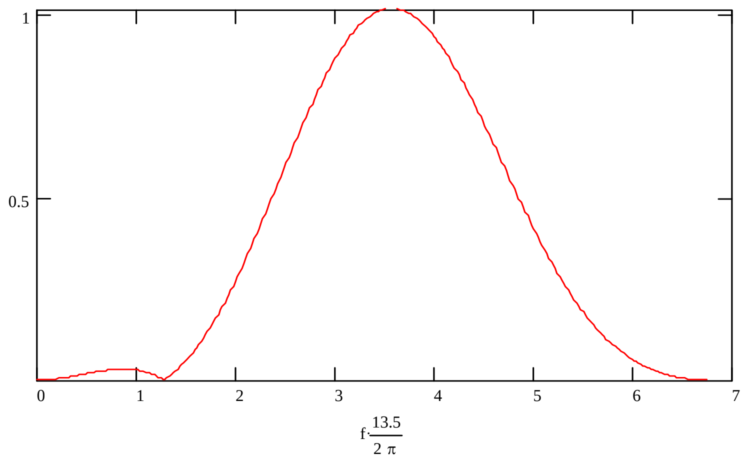
Filter 1 - [0,3,-4,-2,9,-2,-4,3,0]

Filter 2 - bypass

Filter 3 - [-1,2,-1]

Output - direct

Fig 144.Band-pass filter 4.43 MHz, wide



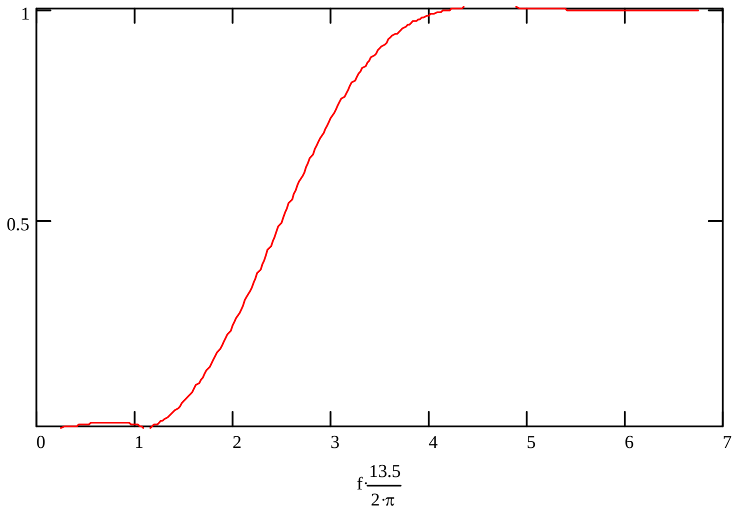
Filter 1 - [0,0,-4,-3,8,-3,-4,0,0]

Filter 2 - bypass

Filter 3 - [-1,0,2,0,-1]

Output - direct

Fig 145.Band-pass filter 3.58MHz, wide



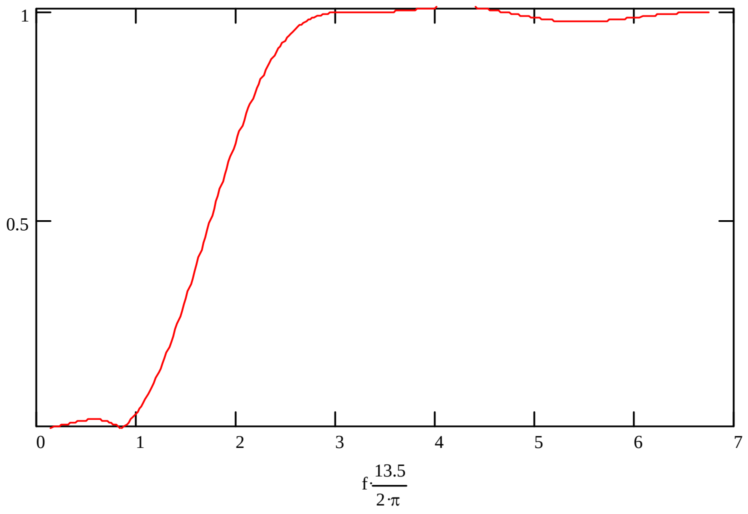
Filter 1 - [0,-2,0,6,8,6,0,-2,0]

Filter 2 - [1,2,1]

Filter 3 - [1,2,1]

Output - 1-filter

Fig 146.High-pass filter for 4.43MHz



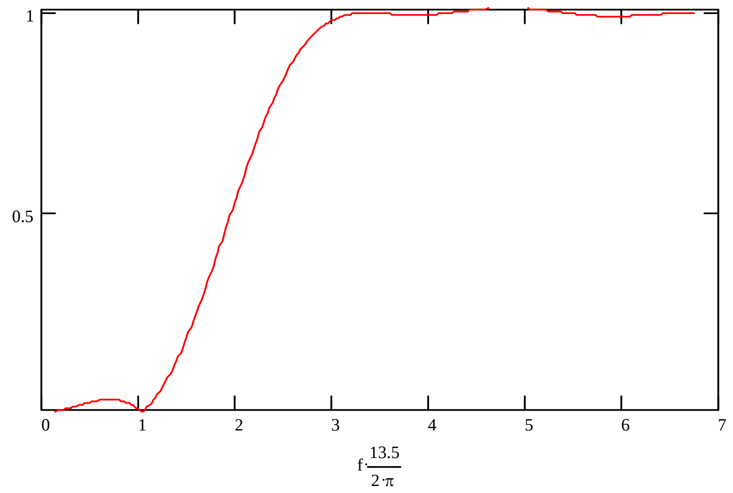
Filter 1 - [-2,-2,4,4,8,4,4,-2,-2]

Filter 2 - [1,2,1]

Filter 3 - [1,0,2,0,1]

Output - 1-filter

Fig 147.High-pass filter for 3.58 MHz



Filter 1 - [0,-4,0,6,12,6,0,-4,0]

Filter 2 - [1,2,1]

Filter 3 - [1,0,2,0,1]

Output - 1-filter

Fig 148.High pass filter for 3.58 MHz and 4.43 MHz

4.2 Noise reduction

The noise reduction system comprises noise reduction for both luminance and color difference signals. The noise reduction can work as a standalone noise reduction system or in conjunction with a Temporal Noise Reduction (TNR) system, e.g. PNX8550.

The noise reduction for luminance is divided into two parts:

- Spatial temporal Weighted Averaging Noise reduction (SWAN) filter.
- LOcal REgression approximation noise reduction (LORE) filter.

Refer to Figure 149 for a block diagram of the noise reduction system.

The SWAN block performs well in noisy pictures, but with noise breakthrough in areas with a soft gradient. This can lead to unnatural looking “plastic” faces. The LORE block performs well on low noise pictures, not showing mentioned artefacts but with reduced sharpness impression in detailed picture content.

The piecewise linear function block detects how valuable the local regression approximation is. When there is much detail in the pixel vicinity, the output of the control block indicates a bad match, forcing towards the edge-preserving SWAN algorithm. In the other case, if the local regression approximation results in a valuable output is forced towards the LORE algorithm. This is done with the fader. This setup results in a well performing noise reduction system for luminance over a wide range of noise levels.

The SWAN block performs noise reduction on both luminance and color difference signals. The LORE block is only used for luminance. The LORE block may be switched off.

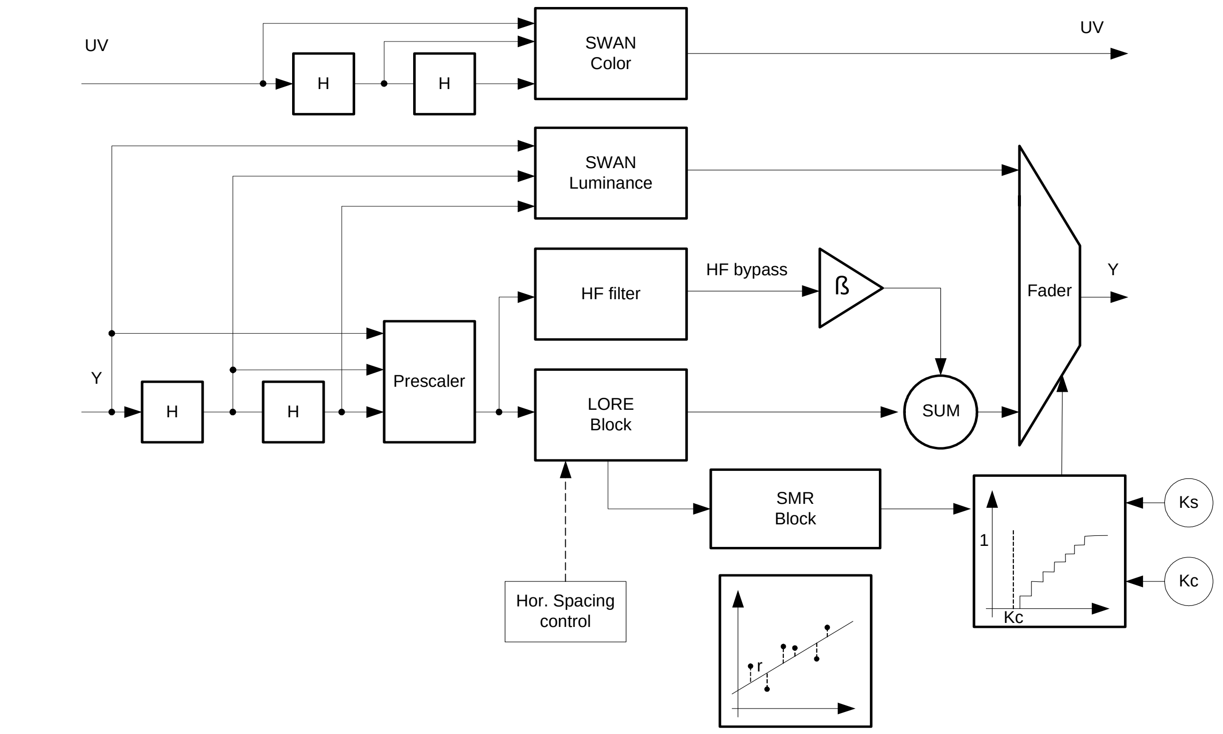


Fig 149.Block diagram - noise reduction system

4.3 SWAN filter

Spatial temporal Weighted Averaging Noise reduction is a combination of a spatial filter and a temporal recursive filter, which is used to reduce white Gaussian noise, added to luma and chroma components of the video data. The algorithm is based on a sigma filter. Around each pixel from the video data that has to be filtered, a 3-D vicinity is defined. To each selected pixel one weighting coefficient is assigned, based on the magnitude of the difference between the selected pixel and the pixel to be filtered. At the end, the sum of the weighted pixels is calculated and normalized. The filter is called discriminating averaging filter (DAF).

Averaging noise filters tend to blur image detail. To improve the subjective impression of the noise reduction a 2-D 3x3 high pass filter is used to bypass the high frequency component of the signal. The bypass is done by controlled adding of high frequencies to the filtered signal. To increase the perceived sharpness of the image, additional peaking is applied on the high frequency component. Since the high frequency noise can be amplified by peaking operation, coring is applied to reduce this effect. Peaking and coring are used only on luminance. See Figure 150 for peaking and coring characteristics.

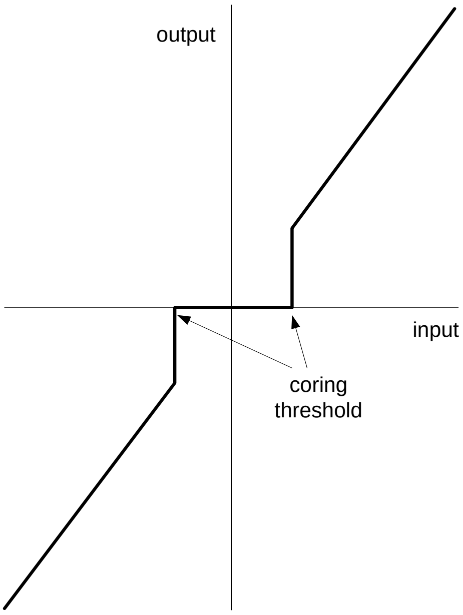


Fig 150.Peaking and coring characteristics

The noise reduction block diagram is given in Figure 151 for both luminance and color difference signals.

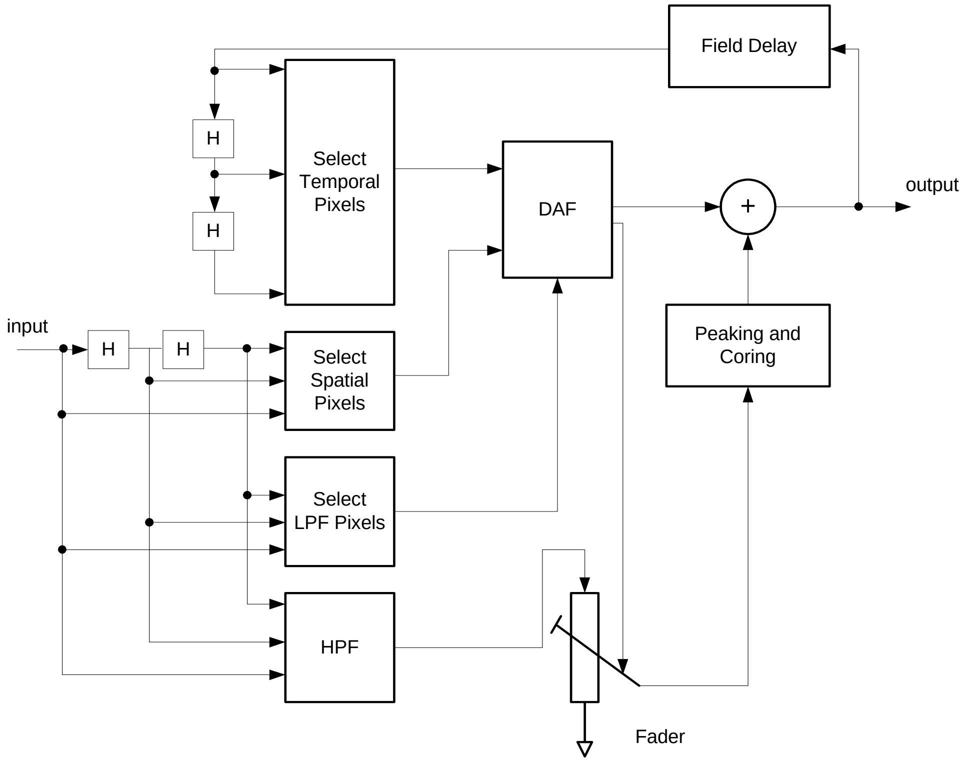


Fig 151.Block diagram SWAN for luminance and color difference signals

4.3.1 Spatial/temporal pixel selection

The Discriminating Averaging Filter (DAF) gets 15 pixels from the present field and 9 pixels from the previous field from the pixel selection blocks. The location of the pixels (called “constellation”) can be set using the different parameters. In Table 238 the parameters for setting the constellation in the DAF filter are given for both luminance and color difference signals.

Table 238: Parameters for setting the constellation in the DAF Luminance filter

Register luminance[1]	Register color[2]	Parameter	Distance wrt reference pixel	Remark
SWAN_L1A[7:6] SWAN_L1B[7:6]	SWAN_C1[7:6]	cn_dist	±1,±2,±3,±4	Distance between the current pixel and the two nearby pixels on the current line
SWAN_L1A[5:4] SWAN_L1B[5:4]	SWAN_C1[5:4]	ce_dist	±2,±4,±6±8	Distance between the current pixel and the two end pixels on the current line
SWAN_L1A[3:2] SWAN_L1B[3:2]	SWAN_C1[3:2]	nn_dist	±1,±2,±3,±4	Distance between the centre pixel and the two nearby pixels on the neighboring lines
SWAN_L1A[1:0] SWAN_L1B[1:0]	SWAN_C1[1:0]	ne_dist	±2,±4,±6,±8	Distance between the centre pixel and the two end pixels on the neighboring lines
SWAN_L0A[6:4] SWAN_L0B[6:4]	SWAN_C0[6:4]	n_sh	-4,-3,-2-1,1,0,1,2,3	Right/left shift of the pixels on the neighboring line wrt current pixel
SWAN_L0A[1:0] SWAN_L0B[1:0]	SWAN_C0[1:0]	rec_dist	1, 2, 3, 4	Distance between the centre pixel and the two nearby pixels on the neighboring lines from the previous recursive field

[1] At the moment of the vertical timing pulse, alternating data is selected from the A and B registers. The result is that for consecutive fields the settings as programmed in registers A, B, A, B… etc. is used (luminance only).

[2] Only the even values are allowed for the color difference signals due to the 4:2:2 format.

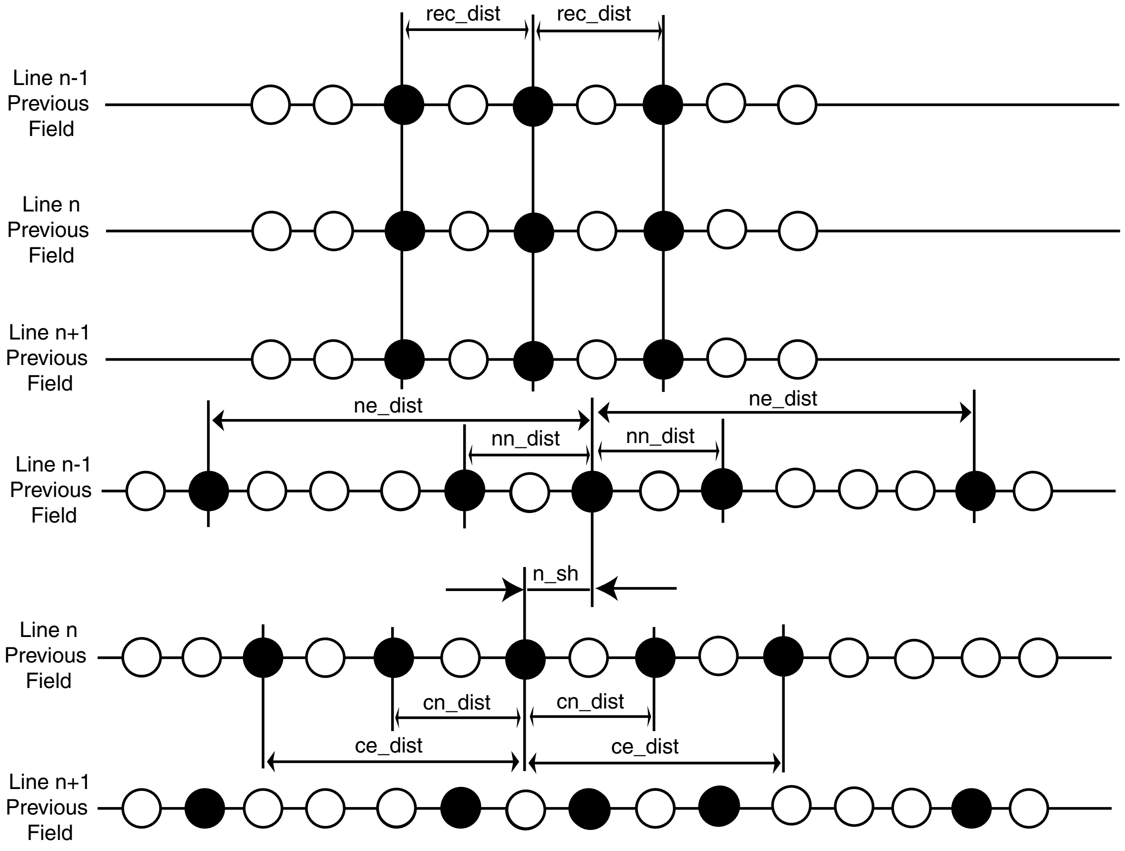


Fig 152.Pixel constellation present and previous field DAF-filter

Referring to Figure 152, the absolute differences between the reference pixel (dashed pixel) and the selected pixels (black pixels) are calculated for both the present (spatial) and previous (temporal) field.

The absolute differences are compared to a spatial and temporal noise threshold level (Thr1, register: SWAN_L3[7:0] for luminance and SWAN_C3[7:0] for color). Depending on the outcome different weighting coefficients are assigned (refer to Figure 153). Hence, a higher setting of Thr1 will result in a higher level of noise reduction. User definable threshold levels Thr2 and Thr3 result in applying negative weighting coefficients C3. As a rule Thr2 > 4 * Thr1 and Thr3 > Thr2, furthermore C3 is set to zero for the color difference signals.

All different threshold levels can be assigned to both the spatial and temporal part of the filter.

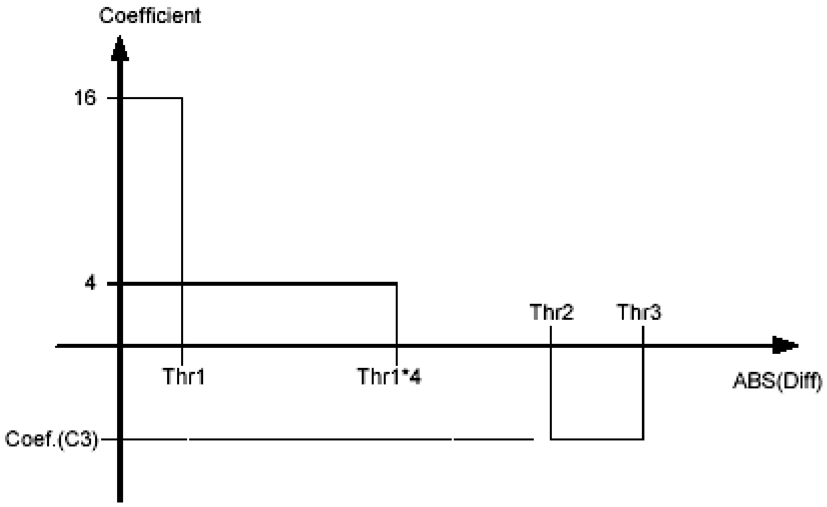


Fig 153.Relation between weighting coefficient and absolute difference

4.3.2 High pass filter, fader, peaking and coring block

The current pixel is filtered with a high pass filter. Depending on the amount of pixels with positive weight coefficients (information coming from the DAF block), a part (fader) of the high frequency spectrum from the high pass filter is after peaking and coring added to the filtered signal. If the amount of pixels with positive weight coefficients is bigger more high frequency spectrum is added to the filtered signal.

4.4 LORE filter

LORE is the abbreviation of “LOcal REgression” which is the heart of the algorithm. The noise in a picture is reduced by making a new estimate for every pixel based on a linear local regression in the vicinity of the pixels. This comes down to finding the “best fitting surface” in the pixel vicinity. By calculating the sum of residues of every pixel in the vicinity, a figure of merit is calculated that indicates how good this surface actually fits. (The residue is the magnitude of the difference of the noisy pixel value and its estimate on the regression surface.) In the area of edges, this sum of residues is high. In this case, the local regression estimate may be inaccurate and an estimate based on an edge preserving noise reduction algorithm (such as SWAN) is taken. In the actual implementation, a fader is used which chooses between the LORE and the SWAN output depending on the Sum of Magnitudes of the Residual (SMR) value.

4.4.1 Pixel selection

The Local Regression Approximation Noise Reduction block gets 15 pixels from the present field from the Pixel Selection blocks. The constellation can be set to narrow or wide (register: LORE_F0[6], See Figure 154). This selection is used as filter input to calculate a new value for the current pixel. The same selection of pixels will also be used to calculate the quality of the new estimate.

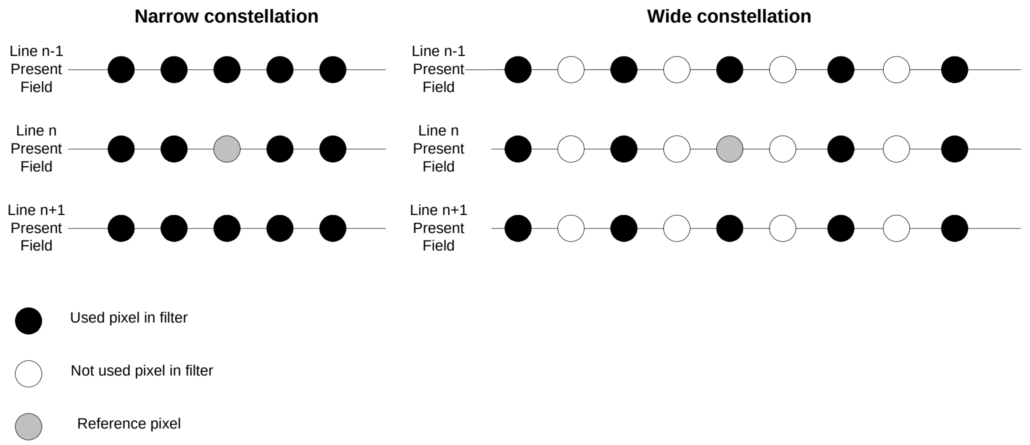


Fig 154.Constellation for local regression approximation noise reduction

4.4.2 The SMR block

The SMR block calculates the ‘Sum of Magnitudes of Residues’. This is a figure that gives an indication of how well the pixel vicinity is approximated by a linear surface. If the figure is low, the new estimate of the current pixel can be trusted as a good noise reduced value. If the figure is high, e.g. in the vicinity of an edge, the estimate may be regarded as dubious and the final output better achieved with an edge preserving noise reduction algorithm such as SWAN.

4.4.3 The fader

The fader block provides the right mix between the LORE and the SWAN algorithm. The fading is adapted for every pixel based on the output of the transform block. This transform block makes a transformation from the output of the SMR block to a signal that drives the fader. The transformation function is piecewise linear. Through the user parameters Kc (“corner”) and Ks (“slope” registers: LORE_F1[7:4] and LORE_F1[3:0] respectively), one can adjust the function (see Figure 155). Set these parameters on a global noise measurement. The higher Kc and Ks are, the higher the noise reduction will be, but expect loss of detail and sharpness.

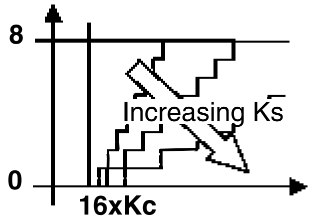


Fig 155.Transform block as function of Kc and Ks

The following settings of Kc and Ks are special:

Table 239: Special cases for Ks and Kc

ks	kc	Function
0	-	step function
0	0	Constant 8 => always SWAN
15	don't care	Constant 0 => always LORE

4.5 Noise estimator

Two noise estimators are present in the Columbus; the auxiliary noise estimator and a second noise estimator that builds on the LORE noise reduction.

To estimate the noise in the video signal the correlation in flat areas is used. For each video field a large number of estimates of the noise is calculated. Such an estimate, called SAD (Sum of Absolute Differences) is obtained by summing absolute differences between current pixel values and delayed pixel values over an area of 4 pixels. In order to get as many as possible noise estimates in a picture, the groups of pixels for which SAD is calculated have 3 pixels in common i.e. they partially overlap. Within the lower part of the total range of possible estimates 15 intervals are defined. Each interval is defined by a low boundary and an upper boundary. The low and the upper boundary can be varied, while the upper boundary has a fixed relation to the low boundary. The low boundary is increased or decreased with 1 each field until an interval is found which contains at least predefined number of estimates, and is at the same time lowest in the range. The value of the low boundary of this interval determines the current output noise figure. The predefined number of estimates is called “wanted value” (register NOISE_E1[7:0]) and good results are obtained with a value which is about 0.27% of the total number of blocks.

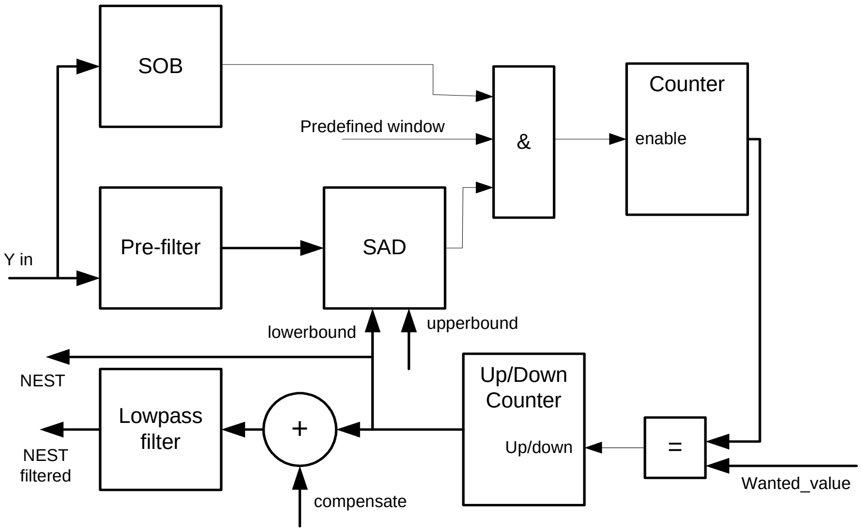


Fig 156.Block diagram auxiliary noise measurement

For video fields with a lot of noise the number of small differences is very low, respectively the number of noise estimates in the lower intervals is close to 0. On the contrary, for clean sequences this number is very high. This means that for clean sequences enough number of estimates will be found in the lower intervals and the noise estimate figure will be close to 0, and for sequences with a lot of noise there will be not enough noise estimates and the noise estimate figure will reach 15.

Noise measurement is performed within a predefined video “window”. The predefined window can be set using registers NOISE_E5, NOISE_E6, NOISE_E7 and NOISE_E8. The parameters are all 8 bits and they are given in multiples of 4 samples or lines. The start value of the window is included, while the stop value is excluded. When the start and the stop values of the window are equal, the window is not active.

The input (Yin) can cope with both 8 bit and 9 bit formats (register: NOISE_E0[3]). To increase the sensitivity of the noise measurement circuit for video sequences with low noise level a pre-filter is added at the input of the Yin signal. In Table 240 recommendations are given for the filter use cases. In Figure 157 to Figure 159 the transfer curves are drawn for this pre-filter. A pre-filter scaling factor of 1, ½, ¼ or off can be set in register NOISE_E0[5:4].

Table 240: Filter use case and related settings for the NOISE_E0 register

F	NOISE_E0	10 tap filter	13.5 MHz	13.5 MHz
	[1:0]	coefficients	NTSC	NTSC
1	0	0 1 1 0 0 -1 0 0 1 0 0 -1	yes	yes
2	1	1 2 1 0 0 -3 0 0 3 0 0 -1	yes	yes
3	2,3	2, 3 3 2 0 -2 -1 0 0 1 2 0 -2	no	no

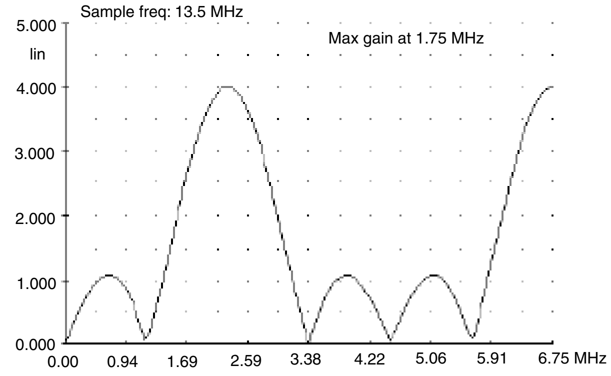


Fig 157.Transfer curve pre-filter F1 at 13.5 MHz

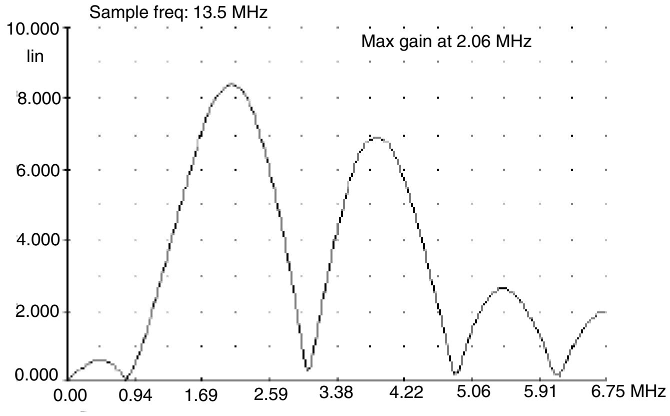


Fig 158.Transfer curve pre-filter F2 at 13.5 MHz

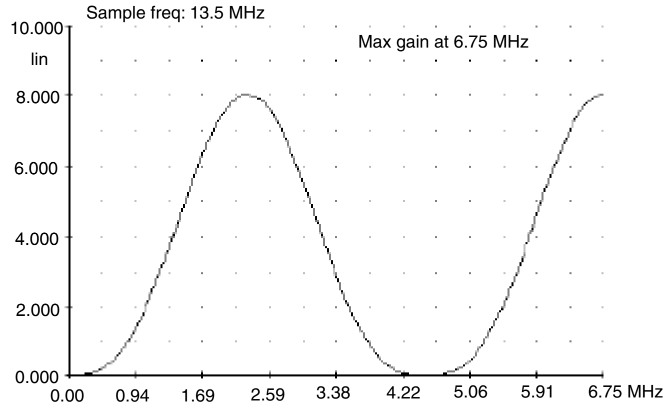


Fig 159.Transfer curve pre-filter F3 at 13.5 MHz

The noise estimate value is available reading register NOISE_E10[3:0]. A low pass value of the noise estimate is available in register NOISE_E11[7:0]. The filter is a low pass recursive filter with system function:

$$H(z) = \frac{16}{16 - 15z} - 1$$

4.5.1 Noise estimator based on LORE noise reduction

In the LORE algorithm, an intermediate value called the "sum of magnitudes of residues" (SMR) is calculated in every pixel. It is found that this SMR of a pixel is very closely related to the sigma of the noise in the filter aperture of that pixel when this pixel lies in a flat area: the SMR then has a precise linear relation with the noise amplitude. So, to obtain a good measure of the noise power in a picture, search an area in that picture without fine detail (it is assumed that in every field, there is at least one such area). Note that for this noise estimator, such a flat area may have a gradient.

To find a flat area in a field, the pixel with the lowest SMR is searched. This lowest SMR may already be a good measure of the noise power of the field. However, there would be some correlation with the size and number of flat areas in the field. A better measure is to take the SMR of one or more pixels close to the pixel with the lowest SMR: the chance is very high that these pixels also belong to the same flat surface. However, carry out a check to be sure that there is no picture detail in the aperture of those neighboring pixels. This is achieved in the block ‘Check Conditions’.

In the heart of the block diagram are two registers: the Minimum register and the Intermediate NoiseEst register. These registers are reset every field. When a certain pixel is found around which the SMR is lower than the one kept in the Minimum register and when some other conditions are met, both registers are updated: the Minimum register with the value CurrNoiseEst and the intermediate NoiseEst register with the value CandNoiseEst. At the end of a field, the intermediate NoiseEst register is transferred to the output filter.

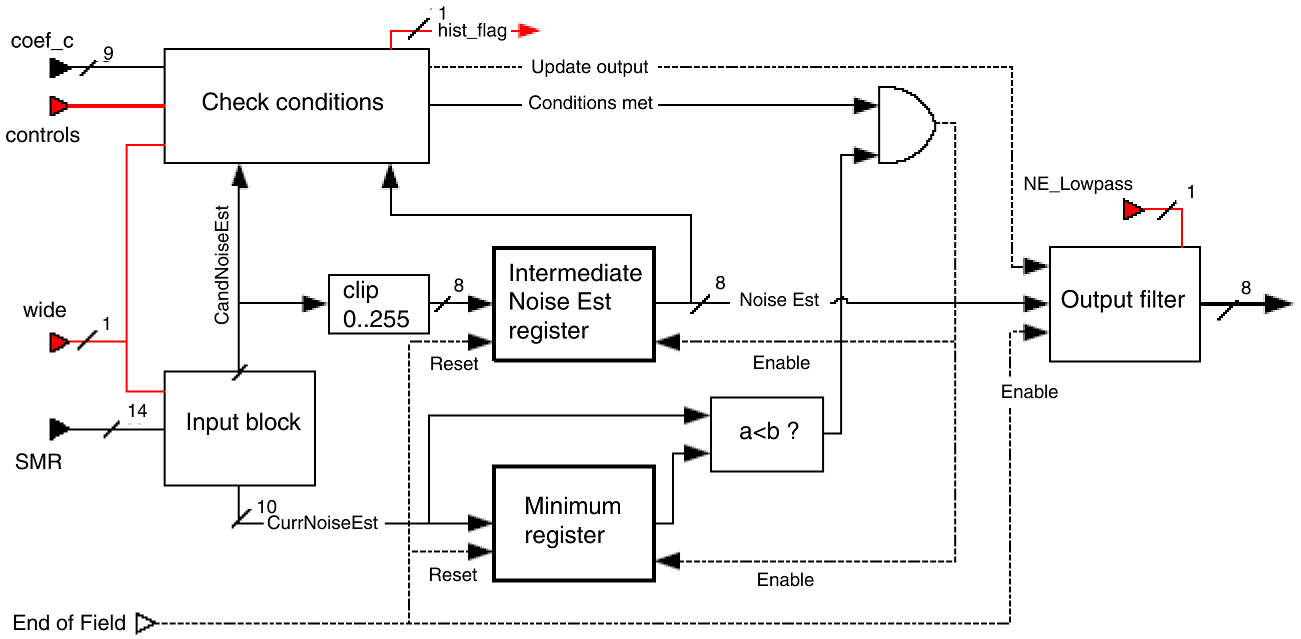


Fig 160.Block diagram LORE based noise measurement

The input block scales and clips the SMR signal. It also derives four different noise estimations in the vicinity of the current pixel: CurrNoiseEst, which is the SMR around the current pixel and the new candidate for the intermediate NoiseEst, register CandNoiseEst, which is the minimum of the SMR of a left and a right pixel. Note that the delay D is one pixel for wide=1 and four pixels for wide=0: To avoid correlation between CurrNoiseEst and CandNoiseEst, the delay is rather high for wide=0 because their apertures overlap but it can be small for wide=1 because the apertures are interleaved in this case.

In the output filter, a NoiseEstFilt register is updated every field and to the actual output this register is scaled and rounded. When the control signal NE_lowpass (register: LORE_N0[1]) is low, this register is updated by the intermediate NoiseEst register. The output is then equal to the intermediate register at the end of a field. When NE_lowpass is high, a recursive low pass filter with this system function averages the values from the intermediate register:

$$H(z) = \frac{16}{16 - 15z} - 1$$

When a pixel has a very low SMR, so that it can be assumed that it lies in a flat area, the pixels close to it have a high chance to lie in that same flat area. To be sure, carry out a check in this block. This is done by comparing the value of the intermediate NoiseEst register with the new candidate to update it CandNoiseEst. When the latter is twice as high than the former, it is assumed that the CandNoiseEst is calculated in an aperture with picture detail.

When the feature value of the pixel is too close to the value range (almost black or almost white), then it can happen that the original pixel values plus the noise power results in a clipped value. The resulting SMR in such clipped flat areas would be on average lower than the SMR of non-clipped flat areas. The clipping control avoids that Intermediate

NoiseEst register and the Minimum register is updated when the current pixel is in a flat area. The check is done on the signal coef_c of some pixels around the current pixel which is after the local regression approximation. The clip range can be defined by the control parameter clip_offset (register: LORE_N0[7:6]. This functionality can also be turned-off by setting the clip_neglect control parameter (register LORE_N0[5]).

By resetting the control parameter win_neglect (register: LORE_N0[0], it is defined that the noise measurement is only performed in a specified window. The window is defined by four 8-bit parameters win_H_start, win_H_stop, win_V_start and win_V_start. To calculate the actual pixel and line values, the window parameters are multiplied by four. The start pixel and lines are included, the stop pixel and lines are excluded (registers: LORE_N1[7:0], LORE_N2[7:0], LORE_N3[7:0] and LORE_N4[7:0]).

The histogram functionality monitors the number of valid measurements over a field. This number is influenced by the windowing control and by the clipping control. The control parameter hist set defines a minimum number of blocks. In every field where the minimum number of blocks is not reached, the output flag hist_flag is set. Some active control can be coupled to this flag when the control parameter hist_neglect is reset: when the hist_flag is set, the noise estimate in the output filter is not updated at the end of the field. When hist_neglect is set, the noise estimate is updated every field, no matter the status of the histogram flag.

4.6 I^{2}C interfaces

All control registers can be accessed through I^{2}C. Most signals that can be written via I^{2}C are double buffered.

The SWAN distance registers are switched field frequently. Two sets of registers are available (SWAN_L0A, SWAN_L0B and SWAN_L1A, SWAM_L1B). The output register is copied during the V-Sync pulse from the A or B registers and toggles field frequently in the hardware.

The fast I^{2}C interface implemented on the Columbus is a 5V tolerant, 400 kHz slave receiver/transmitter. The I^{2}C will not be blocked during voltage shorts or opens. The allocated I^{2}C address of the Columbus is:

Table 241: Address of the 3D comb filter [1]

1	0	1	1	0	0	0	RWn[2]

[1] Hex: 0xB0

[2] RWn is the Read Write (not) bit of the I^{2}C protocol.

For the system dependent parameters of the 3D Comb filter (I^{2}C address 0x10 .. 0x18) 5 register banks are present. Data can be written in one the bank via I^{2}C by programming bits[2:0] of the SYSTEM_SELECT register. The bits [6:4] of the SYSTEM_SELECT register select, which register bank, is used by the 3D comb filter to define the filter settings.

Table 242: Relation between bank selected and TV system

Bank number	System
0	PAL B, G, H, I, D, K
1	PAL M

Table 242: Relation between bank selected and TV system *…continued*

Bank number	System
2	PAL N
3	NTSC
4	Bypass

Each internal (8-bits) device register has its own register address. The registers are grouped in two groups, write/read and read-only registers. This grouping allows reading or writing a sequence of registers without extra I^{2}C overhead. For reading or writing of several registers in a sequential order the bit “aif” in the register address byte is set to ‘0’, which means auto increment of the register address. Figure 161 shows flow diagrams of the register write/read protocol.

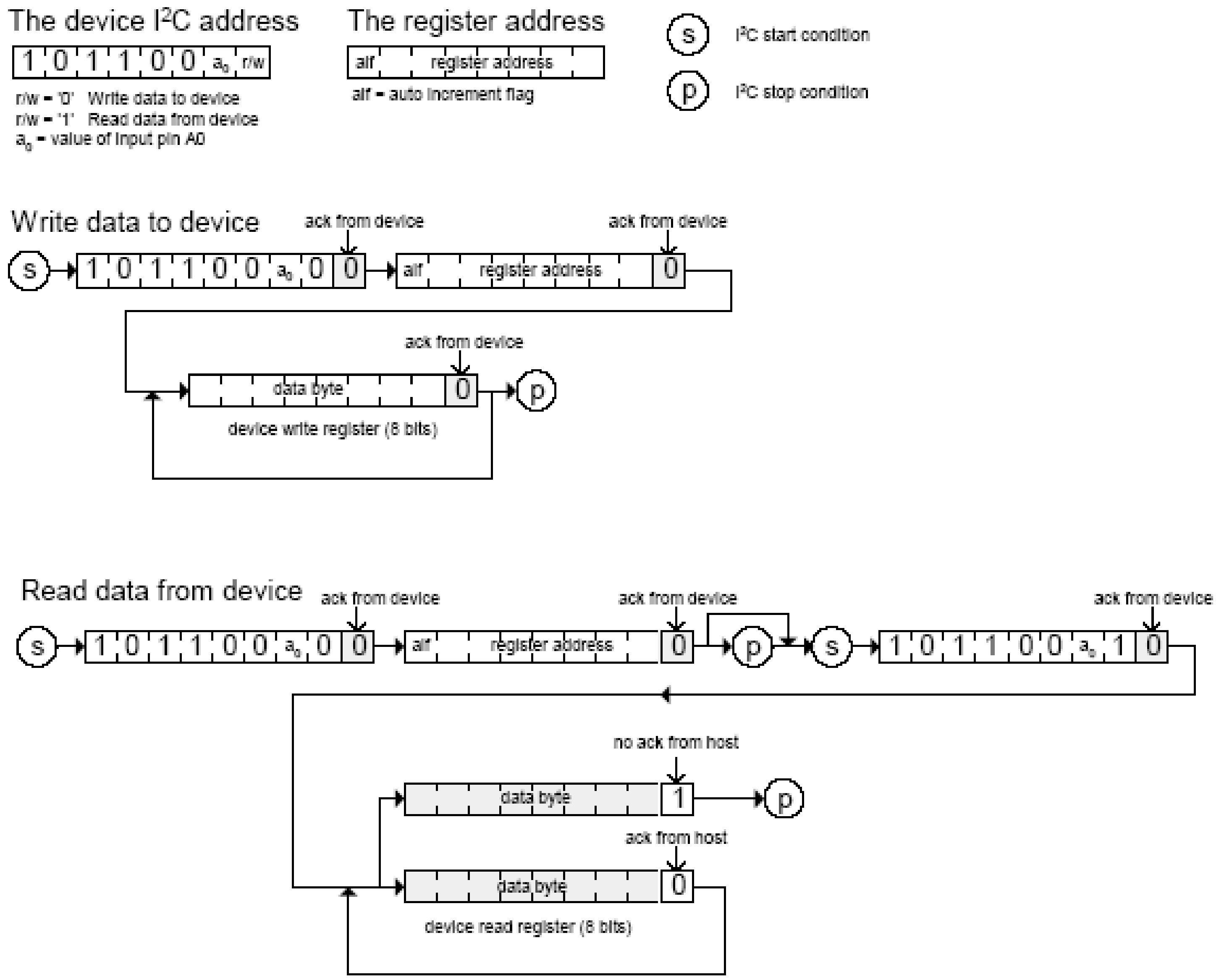


Fig 161.Higher-level I^{2}C register interface protocol

4.7 Control registers description

All internal registers in the Columbus have their own register address. The registers for the 3D Comb filter that program the system dependent parameters also have a sub-register address, which is set by the SYSTEM_SELECT register (I^2C address 0x03). In this way five different banks are present, one for each system.

The internal registers are grouped, which allows writing or reading of a sequence of register in a sequential order without extra I^2C overhead. In this mode the “auto increment flag” (‘aif’ bit which is the msb of the register address byte) is set to ‘0’, which means auto increment of the register address after a write or read access.

Table 243: Register summary

Offset	Name	Description
0x0	RESET0	
0x1	GENERAL0	
0x2	RESERVED0	
0x3	SYSTEM_SELECT0	
0x5	TEST_GEN0	
0x6	DEMO_SCREEN_MODE0	
0x7	DEMO_SCREEN_H0	
0x8	DEMO_SCREEN_V0	
0x9	Reserved0	
0xA	Reserved0	
0xB	Reserved0	
0xC	Reserved0	
0xD	Reserved0	
0xE	Reserved0	
0xF	Reserved0	
0x10	COMB_S00	Data is written in one of the five banks, selected by the system select bits [3:0] on I2C addr 0x03
0x11	COMB_S10	
0x12	COMB_S20	
0x13	COMB_S30	
0x14	COMB_S40	
0x15	COMB_S50	
0x16	COMB_S60	
0x17	COMB_S70	
0x18	COMB_S80	
0x1A	COMB_N00	
0x1B	COMB_N10	
0x1C	COMB_N20	
0x1D	COMB_N30	
0x1E	COMB_N40	
0x20	COMB_P00	
0x21	COMB_P10	

Table 243: Register summary *…continued*

Offset	Name	Description
0x22	COMB_P20	
0x23	COMB_P30	
0x24	COMB_P40	
0x25	COMB_P50	
0x26	COMB_P60	
0x27	COMB_P70	
0x28	COMB_P80	
0x2C	COMB_M00	
0x2D	COMB_M10	
0x2E	COMB_M20	
0x2F	COMB_M30	
0x30	SWAN_L20	
0x31	SWAN_L30	
0x32	SWAN_L40	
0x33	SWAN_L50	
0x34	SWAN_L60	
0x35	SWAN_L70	
0x36	SWAN_L80	
0x37	SWAN_L90	
0x38	SWAN_L0A0	
0x39	SWAN_L0B0	
0x3A	SWAN_L1A0	
0x3B	SWAN_L1B0	
0x40	SWAN_C20	
0x41	SWAN_C30	
0x42	SWAN_C40	
0x43	SWAN_C50	
0x44	SWAN_C60	
0x45	SWAN_C70	
0x46	SWAN_C80	
0x47	SWAN_C90	
0x48	SWAN_C00	
0x49	SWAN_C10	
0x50	NOISE_E00	
0x51	NOISE_E10	
0x52	NOISE_E20	
0x53	NOISE_E30	
0x54	NOISE_E40	
0x55	NOISE_E50	
0x56	NOISE_E60	
0x57	NOISE_E70	

Table 243: Register summary *…continued*

Offset	Name	Description
0x58	NOISE_E80	
0x5B	PROT_00	
0x5B	NOISE_E90	
0x5C	PROT_10	
0x5C	NOISE_E100	
0x5D	CLOCK_CONTROL0	
0x5D	NOISE_E110	
0x5E	PROT_30	
0x5E	NOISE_E120	
0x5F	PROT_40	
0x5F	NOISE_E130	
0x60	LORE_F00	
0x61	LORE_F10	
0x64	LORE_N00	
0x65	LORE_N10	
0x66	LORE_N20	
0x67	LORE_N30	
0x68	LORE_N40	
0x6A	TGITU_W00	
0x6B	TGITU_W10	
0x6C	TGITU_W20	
0x6D	TGITU_W30	
0x6E	TGITU_W40	
0x6F	TGITU_W50	
0x70	INOUT_I00	
0x71	INOUT_I10	
0x72	INOUT_O00	
0x73	INOUT_O10	
0x74	INOUT_O20	
0x75	INOUT_O30	
0x78	RESERVED0	
0x79	RESERVED0	
0x7A	RESERVED0	
0x7B	RESERVED0	
0x7C	INOUT_I60	
0x7D	INOUT_I70	
0x7E	RESERVED0	
0x7F	RESERVED0	

Table 244: Columbus registers

Bit	Symbol	Access	Reset Value	Description
Offset 0x0 - RESET0				
3:0	RES[3:0]	W	0xX	'0101' - reset Columbus, without changing the programmed register values. '1010' - set all writable I2C registers to their default value. Note: These are ‘sticky bits’, i.e. value is removed when write access to this address is completed.
Offset 0x1 - GENERAL0				
5	SDSEL	W	0x1	'0' - No SDRAM present. '1' - SDRAM present
4	COLSEL	W	0x1	'0' - 3D data outputs LowZ. '1' - 3D data outputs HiZ
3	IO_2FH1FH	W	0x1	'0' - input and output is 2 fh (progressive). '1' - input and output is 1 fh
0	RSD	W	0x1	Reserved set to '0'
Offset 0x2 - RESERVED0				
7:0	RSD[7:0]	R	0xC3	Resreved
Offset 0x3 - SYSTEM_SELECT0				
5:3	RDSYS[2:0]	W	0x4	Select register bank to read out data to the 3 D Comb filter for system dependent parameters. '000' - PAL B, G, H, I, D, K, '001' - PAL M, '010' - PAL N, '011' - NTSC, '100' - Bypass, '101 .. '111' - Reserved
2:0	WRSYS[2:0]	W	0x0	Select register bank to write data via I2C for system dependent parameters of 3D Comb filter. '000' - PAL B, G, H, I, D, K, '001' - PAL M, '010' - PAL N, '011' - NTSC, '100' - Bypass, '101 .. '111' - Reserved
Offset 0x5 - TEST_GEN0				
7:6	RSD[1:0]	W	0x3	Reserved
1	TSGIO	W	0x0	'0' - Test screen generator on input, '1' - Test screen generator on output
0	TSGOO	W	0x0	'0' - Test screen generators off. '1' - Test screen generators on
Offset 0x6 - DEMO_SCREEN_MODE0				
3	DEM3D	W	0x0	Demo mode 3D_Comb. '0' - mode is notch, '1' - mode is simple median
2	DEM3D_SWUV	W	0x0	'0' - On demo screen 3D_Swan UV off, '1' - On demo screen 3D_Swan UV on
1	DEM3D_SWLOY	W	0x0	'0' - On demo screen 3D_Swan & Lore Y off, '1' - On demo screen 3D_Swan & Lore Y on
0	DEM3DOO	W	0x1	'0' - On demo screen 3D_Comb off, '1' - On demo screen 3D_Comb on
Offset 0x7 - DEMO_SCREEN_H0				
7:0	HOR[7:0]	R/W	0x00	N: Horizontal demo screen point = 4 x N. Note:
Offset 0x8 - DEMO_SCREEN_V0				
7:0	VER[7:0]	W	0x00	N: Vertical demo screen point = 4 x N.
Offset 0x9 - Reserved0				
7:0	COM_P7RSD[7:0]	R	0xFF	Reserved
Offset 0xA - Reserved0				
7:0	RSD[7:0]	W	0xFF	Reserved

Table 244: Columbus registers *…continued*

Bit	Symbol	Access	Reset Value	Description
Offset 0xB - Reserved0				
7:0	RSD[7:0]	W	0xFF	Reserved
Offset 0xC - Reserved0				
7:0	RSD[7:0]	W	0xC0	Reserved
Offset 0xD - Reserved0				
7:0	RSD[7:0]	W	0xFF	Reserved
Offset 0xE - Reserved0				
7:0	RSD[7:0]	W	0xC1	Reserved
Offset 0xF - Reserved0				
7:0	RSD[7:0]	W	0xFF	Reserved
Offset 0x10 - COMB_S00				
7:0	AMPCON[7:0]	W	0xXX	Amplitude control of the output signal. Double Buffered. System Dependent Reset Values as follows: PALB2K 205, PALM 205, PALN 205, NTSC 205, BYPASS 255
Offset 0x11 - COMB_S10				
7:4	SF3TAPC[3:0]	W	0xX	Spatial Filter 3 TapC (range 0 ..15). Double Buffered. System Dependent Reset
3:0	SF3TAP1[3:0]	W	0xX	Spatial Filter 3 Tap1 (range -8 .. 7). Double Buffered. System Dependent Reset Values
Offset 0x12 - COMB_S20				
7	MCOLV	W	0xX	Motion Colour Version (range 0, 1). Double Buffered. System Dependent Reset Values
6:4	SF3TAP3[2:0]	W	0xX	Spatial Filter 3 Tap3 (range -4 .. 3). Double Buffered. System Dependent Reset Values
3:0	SF3TAP2[3:0]	W	0xX	Spatial Filter 3 Tap2 (range -8 .. 7). Double Buffered. System Dependent Reset Values
Offset 0x13 - COMB_S30				
7:6	SF3TAP4[1:0]	W	0xX	Spatial Filter 3 Tap4 (range -2 .. 1). Double Buffered. System Dependent Reset Values
5	SF1WIDTH	W	0xX	Spatial Filter 1 Width (range 0, 1). Double Buffered. System Dependent Reset Values
4	SF1TAP1	W	0xX	Spatial Filter 1 Tap1 (range 0, 1). Double Buffered. System Dependent Reset Values
3	SF1BYPASS	W	0xX	Spatial Filter 1 Bypass (range 0, 1). Double Buffered. System Dependent Reset Values
2	SF2WIDTH	W	0xX	Spatial Filter 2 Width (range 0, 1). Double Buffered. System Dependent Reset Values
1	SF2TAP1	W	0xX	Spatial Filter 2 Tap1 (range 0, 1). Double Buffered. System Dependent Reset Values
0	SPATDEL	W	0xX	Spatial Delta (range 0, 1). Double Buffered. System Dependent Reset Values

Table 244: Columbus registers *…continued*

Bit	Symbol	Access	Reset Value	Description
Offset 0x14 - COMB_S40				
7:4	TF3TAPC[3:0]	W	0xX	Temp. Filter 3 TapC (range 0 .. 15). Double Buffered. System Dependent Reset Values
3:0	TF3TAP1[3:0]	W	0xX	Temp. Filter 3 Tap1 (range -8 .. 7). Double Buffered. System Dependent Reset Values
Offset 0x15 - COMB_S50				
7	VSTOP_LSB	W	0xX	Vstop, LSB (range 0 .. 1). Double Buffered. Note: Valid for Comb filter, Swan & Lore Y, Swan UV. System Dependent Reset Values
6:4	TF3TAP3[2:0]	W	0xX	Temp. Filter 3 Tap3 (range -4 .. 3). Double Buffered. Note: Valid for Comb filter, Swan & Lore Y, Swan UV. System Dependent Reset Values
3:0	TF3TAP2[3:0]	W	0xX	Temp. Filter 3 Tap2 (range -8 .. 7). Double Buffered. Note: Valid for Comb filter, Swan & Lore Y, Swan UV. System Dependent Reset Values
Offset 0x16 - COMB_S60				
7:6	TF3TAP4[1:0]	W	0xX	Temp. Filter 3 Tap4 (range -2 .. 1) . Double Buffered. System Dependent Reset Values
5	TF1TAP1_1109	W	0xX	Temp. Filter 1 Width (range 0, 1). Double Buffered. System Dependent Reset Values
4	TF1TAP1	W	0xX	Temp. Filter 1 Tap1 (range 0, 1). Double Buffered. System Dependent Reset Values
3	TF1BYPASS	W	0xX	Temp. Filter 1 Bypass (range 0, 1). Double Buffered. System Dependent Reset Values
2	TF2WIDTH	W	0xX	Temp. Filter 2 Width (range 0, 1). Double Buffered. System Dependent Reset Values
1	TF2TAP1	W	0xX	Temp. Filter 2 Tap1 (range 0, 1). Double Buffered. System Dependent Reset Values
0	TEMPDELTA	W	0xX	Temp. Delta (range 0, 1). Double Buffered. System Dependent Reset Values
Offset 0x17 - COMB_S70				
7	COUPFAD	W	0xX	CoupleFaders (range 0,1). Double Buffered. System Dependent Reset Values
6:0	VSTA[6:0]	W	0xXX	Vstart (range 0 .. 127). Double Buffered. Note: The default value is 3+22=25. System Dependent Reset Values
Offset 0x18 - COMB_S80				
7:0	VSTO[7:0]	W	0xXX	Vstop, 8 MSB's (range 0 .. 255). Double Buffered. Note: Vstop(8:0) = COMB_S8(7:0) & COMB_S5(7). Range 0 .. 511. Note: The default value for Vstop is 288+22=310 (PAL) and 240+22=262 (NTSC). System Dependent Reset Values
Offset 0x1A - COMB_N00				

Table 244: Columbus registers *…continued*

Bit	Symbol	Access	Reset Value	Description
7:0	BURSTSTART[7:0]	W	0x00	BurstStart (range 0 ..255). Double Buffered. Note: The BurstStart value specified the number of samples between the internal H pulse and the first burst sample.
Offset 0x1B - COMB_N10				
7:0	ACTSTART[7:0]	W	0x40	ActiveStart (range 1 ..255). Double Buffered. Note: The ActiveStart value specifies the number of samples between the internal H pulse and the first active sample. ActiveStart and ActiveStop are also used in other processing units than 3D COMB.
Offset 0x1C - COMB_N20				
7:0	OFFSET[7:0]	W	0xFF	Offset (range 0 .. 244 for black level correction output of 3D comb filter). Double Buffered : Note: 255 = AUTOMATIC correction based on clamp meter.
Offset 0x1D - COMB_N30				
7:0	ACTSTOP_8M SB[7:0]	W	0xE2	ActiveStop (8 MSB's). Double Buffered. Note: ActiveStop(9:0) = COMB_N3(7:0) & COMB_N4(1:0). Range 2 .. 1023. For proper operation ActiveStop > ActiveStart. ActiveStart and ActiveStop are also used in other processing units than 3D COMB
Offset 0x1E - COMB_N40				
7:0	ACTSTOP_2LS B[7:0]	W	0x00	ActiveStop (2 LSB's). Double Buffered
Offset 0x20 - COMB_P00				
6	LOCMAX	W	0x0	LocalMax (range 0, 1). Double Buffered
5	SWA	W	0x0	Switch A. Double Buffered set to '1' for - Frame processing
4	RSD	W	0x0	Reserved
3	SWD	W	0x0	Switch D (vertical delay of the median filter). Double Buffered. '0' - PAL, 2 lines, '1' - NTSC, 1 line
2	SWI	W	0x0	Switch I. Double Buffered. '0' - UV comb filters parallel, '1' - UV comb filters in series
1:0	NFI[1:0]	W	0x0	Double Buffered. Number of Fields index: '00' - Spatial, '01' - Field, '10' - Frame NTSC (2 fields), '11' - Frame PAL (4 fields).
Offset 0x21 - COMB_P10				
4	RSD	W	0x1	Reserved
3	SWE	W	0x1	Switch E. Double Buffered. '0' -normal operation, '1' - by-pass the Y comb filter
2	SWF	W	0x1	Switch F. Double Buffered. '0' - normal operation, '1' - switch to Spatial only
1	SWG	W	0x1	Switch G. Double Buffered. '0' - normal operation, '1' - by-pass the UV comb filter
0	SWH	W	0x0	Switch H. Double Buffered. '0' - normal operation, '1' - force the spatial filter to notch
Offset 0x22 - COMB_P20				
7:2	CORING[5:0]	W	0x00	Coring (range 0 .. 63). Double Buffered

Table 244: Columbus registers *…continued*

Bit	Symbol	Access	Reset Value	Description
1	TOTH	W	0x1	Special filter preference (1 = Chroma: 0 = LUMA). Double Buffered
0	XLUMA	W	0x1	Xlumaprot (range (0, 1). Double Buffered
Offset 0x23 - COMB_P30				
6:4	PEAK[2:0]	W	0x0	Peaking (range 0 .. 7). Double Buffered
3:1	MOTH[2:0]	W	0x4	Motion Threshold (range 0 .. 7). Double Buffered
0	SWC	W	0x1	Switch C. Double Buffered. '1' - switch to pre filtering, '0' - switch to post filtering
Offset 0x24 - COMB_P40				
5:3	RSD[2:0]	W	0x4	Reserved
2:0	SPAT[2:0]	W	0x4	Spat Threshold (range (0 .. 7). Double Buffered
Offset 0x25 - COMB_P50				
6	ULEX	W	0x0	UpperLowerEx (range 0, 1). Double Buffered
5:3	COLTH[2:0]	W	0x4	Colour Threshold (range 0 .. 7). Double Buffered
2:0	CLT[2:0]	W	0x2	Clamp Time (range 0 .. 7). Double Buffered
Offset 0x26 - COMB_P60				
7	SPPH	W	0x1	Spatial Phase (range 0, 1). Double Buffered
6	TEPH	W	0x1	Temporal Phase (range 0, 1). Double Buffered
5:3	STC[2:0]	W	0x2	SpatialTimeConst (range 0 .. 5). Double Buffered
2:0	TTC[2:0]	W	0x1	TemporalTimeConst (range 0 .. 5). Double Buffered
Offset 0x27 - COMB_P70				
7	AOO	W	0x1	AutoOnOff (range 0, 1). Double Buffered
6:0	ATH[6:0]	W	0x14	AutoThreshold (range 0 .. 127). Double Buffered
Offset 0x28 - COMB_P80				
5	APR	W	0x0	Auto Protect. Switches/resets automatic system. (range off, on). Double Buffered
4	RSD	W	0x0	Reserved
3:1	PTH[2:0]	W	0x2	Protection Threshold (range 0 .. 7). Double Buffered
0	VMO	W	0x0	View Motion (range 0, 1). Double Buffered. '0' = Off '1' = Colour Motion
Offset 0x2C - COMB_M00				
7:0	SPHIAV[7:0]	R	0xXX	SpatialPhiAv (range -128 .. 127)
Offset 0x2D - COMB_M10				
7	APO	R	0xX	Auto Protect Out (range 0, 1)
6:0	SPDEL[6:0]	R	0xXX	SpatialPhiDelta (range 0 .. 127)
Offset 0x2E - COMB_M20				
7:0	TEMPHIAV[7:0]	R	0xXX	TemporalPhiAv (range -128 .. 127)
Offset 0x2F - COMB_M30				
6:0	TEMPPHID[6:0]	R	0xXX	TemporalPhiDelta (range 0 .. 127)
Offset 0x30 - SWAN_L20				
7	LPF	W	0x0	LPF on ref. pixel (off, on). Double Buffered

Table 244: Columbus registers *…continued*

Bit	Symbol	Access	Reset Value	Description
6:4	C3[2:0]	W	0x0	Coef. C3 (0, -1/16, -2/16, -3/16, -4/16, -5/16, -6/16, -7/16). Double Buffered
2:0	C3REC[2:0]	W	0x0	Coef. C3 rec. (0, -1/16, -2/16, -3/16, -4/16, -5/16, -6/16, -7/16). Double Buffered
Offset 0x31 - SWAN_L30				
7:0	THR1S[7:0]	W	0x05	Thr1 for spatial filter. Double Buffered
Offset 0x32 - SWAN_L40				
7:0	THR1T[7:0]	W	0x05	Thr1 for temporal filter. Double Buffered
Offset 0x33 - SWAN_L50				
7:0	THR2S[7:0]	W	0x32	0.5 * Thr2 for spatial filter. Double Buffered
Offset 0x34 - SWAN_L60				
7:0	THR2T[7:0]	W	0x32	0.5 * Thr2 for temporal filter. Double Buffered
Offset 0x35 - SWAN_L70				
7:0	THR3S[7:0]	W	0x64	0.5 * Thr3 for spatial filter. Double Buffered
Offset 0x36 - SWAN_L80				
7:0	THR3T[7:0]	W	0x64	0.5 * Thr3 for temporal filter. Double Buffered
Offset 0x37 - SWAN_L90				
6:4	HFGAIN[2:0]	W	0x4	Gain coefficient of HF (0, ¬º, 2/4, ¬æ, 4/4, 5/4, 6/4, 7/4). Double Buffered
3:0	COTH[3:0]	W	0x2	Coring threshold. Double Buffered
Offset 0x38 - SWAN_L0A0				
6:4	L0A_N[2:0]	W	0x0	Select n_sh (0, 1, 2, 3, -4, -3, -2, -1). Double Buffered. Note: For SWAN_L0A [6:4], [1:0]: During the copying of data in the output buffer at the moment of the vertical timing pulse (double buffering), alternating data is selected from the A and B registers. The result is that for consecutive fields the settings as programmed in registers A, B, A, B, .. etc. is used.
1:0	L0A_REC[1:0]	W	0x0	Select rec_dist (1, 2, 3, 4). Double Buffered
Offset 0x39 - SWAN_L0B0				
6:4	L0B_N[2:0]	W	0x0	Select n_sh (range 0, 1, 2, 3, -4, -3, -2, -1). Double Buffered. Note: For SWAN_L0B [6:4], [1:0]: During the copying of data in the output buffer at the moment of the vertical timing pulse (double buffering), alternating data is selected from the A and B registers. The result is that for consecutive fields the settings as programmed in registers A, B, A, B, .. etc. is used
1:0	L0B_REC[1:0]	W	0x0	Select rec_dist (1, 2, 3, 4). Double Buffered
Offset 0x3A - SWAN_L1A0				
7:6	L1A_CN_1210[1:0]	W	0x0	Select cn_dist (1, 2, 3, 4). Double Buffered. Note: For SWAN_L1A [7:0]: During the copying of data in the output buffer at the moment of the vertical timing pulse (double buffering), alternating data is selected from the A and B registers. The result is that for consecutive fields the settings as programmed in registers A, B, A, B, .. etc. is used.

Table 244: Columbus registers *…continued*

Bit	Symbol	Access	Reset Value	Description
5:4	L1A_CE[1:0]	W	0x0	Select ce_dist (2, 4, 6, 8). Double Buffered
3:2	L1A_NN[1:0]	W	0x0	Select nn_dist (1, 2, 3, 4). Double Buffered
1:0	L1A_CN[1:0]	W	0x0	Select ne_dist (2, 4, 6, 8). Double Buffered
Offset 0x3B - SWAN_L1B0				
7:6	L1B_CN[1:0]	W	0x0	Select cn_dist (1, 2, 3, 4). Double Buffered. Note: For SWAN_L1B [7:0]: During the copying of data in the output buffer at the moment of the vertical timing pulse (double buffering), alternating data is selected from the A and B registers. The result is that for consecutive fields the settings as programmed in registers A, B, A, B, .. etc. is used.
5:4	L1B_CE[1:0]	W	0x0	Select ce_dist (2, 4, 6, 8). Double Buffered
3:2	L1B_NN[1:0]	W	0x0	Select nn_dist (1, 2, 3, 4). Double Buffered
1:0	L1B_NE[1:0]	W	0x0	Select ne_dist (2, 4, 6, 8). Double Buffered
Offset 0x40 - SWAN_C20				
7	C2LPF_1221	W	0x0	LPF on ref. pixel (off, on). Double Buffered
6:4	C2LPF[2:0]	W	0x0	Coef. C3 (0, -1/16, -2/16, -3/16, -4/16, -5/16, -6/16, -7/16). Double Buffered
2:0	C2COEF[2:0]	W	0x0	Coef. C3 rec. (0, -1/16, -2/16, -3/16, -4/16, -5/16, -6/16, -7/16). Double Buffered
Offset 0x41 - SWAN_C30				
7:0	C3T1[7:0]	W	0x0A	Thr1 for spatial filter. Double Buffered
Offset 0x42 - SWAN_C40				
7:0	C4T1[7:0]	W	0x0A	Thr1 for temporal filter. Double Buffered
Offset 0x43 - SWAN_C50				
7:0	C5T2[7:0]	W	0xFF	0.5 * Thr2 for spatial filter. Double Buffered
Offset 0x44 - SWAN_C60				
7:0	C6T2[7:0]	W	0xFF	0.5 * Thr2 for temporal filter. Double Buffered
Offset 0x45 - SWAN_C70				
7:0	C7T3[7:0]	W	0xFF	0.5 * Thr3 for spatial filter. Double Buffered
Offset 0x46 - SWAN_C80				
7:0	C8T3[7:0]	W	0xFF	0.5 * Thr3 for temporal filter. Double Buffered
Offset 0x47 - SWAN_C90				
6:4	C9HFGAIN[2:0]	W	0x0	Gain coefficient of HF (0, ¬º, 2/4, ¬æ, 4/4, 5/4, 6/4, 7/4). Double Buffered
3:0	C9CORE[3:0]	W	0x0	Coring threshold(0 to 15). Double Buffered
Offset 0x48 - SWAN_C00				
6:4	C0SELN[2:0]	W	0x0	Select n_sh (0, 1, 2, 3, -4, -3, -2, -1). Double Buffered. Only the even values are allowed, as the odd values result in a mixture of U and V samples.
1:0	C0SELR[1:0]	W	0x1	Select rec_dist (1, 2, 3, 4). Double Buffered. Only the even values are allowed, as the odd values result in a mixture of U and V samples.

Table 244: Columbus registers *…continued*

Bit	Symbol	Access	Reset Value	Description
Offset 0x49 - SWAN_C10				
7:6	C1CN[1:0]	W	0x1	Select cn_dist (1, 2, 3, 4). Double Buffered. Only the even values are allowed, as the odd values result in a mixture of U and V samples.
5:4	C1CE[1:0]	W	0x1	Select ce_dist (2, 4, 6, 8). Double Buffered
3:2	C1NN[1:0]	W	0x1	Select nn_dist (1, 2, 3, 4). Double Buffered. Only the even values are allowed, as the odd values result in a mixture of U and V samples.
1:0	C1NE[1:0]	W	0x1	Select ne_dist (2, 4, 6, 8). Double Buffered
Offset 0x50 - NOISE_E00				
5:4	E0PS[1:0]	W	0x3	Prefilter scaling (1, ¬¾, ¬º, off). Double Buffered
3	E0NOB	W	0x0	Number of bits (9, 8). Double Buffered
2	E0CC	W	0x0	Copy lsb1 to lsb0, convert 9 bits to 8 bits. Double Buffered
1:0	E0SEL[1:0]	W	0x0	Select filter (filter1, filter2, filter3, filter3). Double Buffered
Offset 0x51 - NOISE_E10				
7:0	E1NO[7:0]	W	0x46	Number of estimates (wanted value). Double Buffered Note: internally multiplied by 8.
Offset 0x52 - NOISE_E20				
7:4	E2COMP[3:0]	W	0x0	Compensation value (range 0, 1, 2, 3, 4, 5, 6, 7, -8, -7, -6, -5, -4, -3, -2, -1). Double Buffered
3	E2SOB	W	0x0	SOB (use, neglect). Double Buffered
2:0	E2GAIN[2:0]	W	0x0	Gain to calculate upper boundary of interval. Double Buffered
Offset 0x53 - NOISE_E30				
7:0	E3BOT[7:0]	W	0x32	Bottom limit of detail counter. Double Buffered
Offset 0x54 - NOISE_E40				
7:0	E4TOP[7:0]	W	0xBE	Top limit of detail counter. Double Buffered
Offset 0x55 - NOISE_E50				
7:0	E5STA[7:0]	W	0x05	0.25 * Start of horizontal window. Double Buffered
Offset 0x56 - NOISE_E60				
7:0	E6STO[7:0]	W	0xAA	0.25 * Stop of horizontal window. Double Buffered
Offset 0x57 - NOISE_E70				
7:0	E7STA[7:0]	W	0x05	0.25 * Start of vertical window. Double Buffered
Offset 0x58 - NOISE_E80				
7:0	E8STO[7:0]	W	0x46	0.25 * Stop of vertical window. Double Buffered
Offset 0x5B - NOISE_E90				
7:0	E9NE[7:0]	R	0xXX	Lore: NoiseEstimate
Offset 0x5B - PROT_00				
7:0	PT0[7:0]	W	0x00	Prototype Register 0
Offset 0x5C - NOISE_E100				
7	E10HIS	R	0xX	Lore: Hist_flag. Indication that the number of non-clipped measurements falls below MinNrBlocks

Table 244: Columbus registers *…continued*

Bit	Symbol	Access	Reset Value	Description
3:0	E10NEV[3:0]	R	0xX	Noise estimator value
Offset 0x5C - PROT_10				
7:0	PT1[7:0]	W	0x00	Prototype Register 1
Offset 0x5D - CLOCK_CONTROL0				
0	CLKCON	W	0x0	Invert the output clock: Clkasb. '0' : Do not invert Clkasb, '1' : Invert Clkasb
Offset 0x5D - NOISE_E110				
7:0	E11FNEV[7:0]	R	0xXX	Filtered noise estimator value
Offset 0x5E - NOISE_E120				
7:0	E12NDH[7:0]	R	0xXX	Number of details in field (high byte)
Offset 0x5E - PROT_30				
7:0	PT3[7:0]	W	0x00	Prototype Register 3
Offset 0x5F - NOISE_E130				
7:0	E13NDL[7:0]	R	0xXX	Number of details in field (low byte)
Offset 0x5F - PROT_40				
7:0	PT4[7:0]	W	0x00	Prototype Register 4
Offset 0x60 - LORE_F00				
7	F0TM	W	0x0	Test mode (Test Fade). Double Buffered. '0' : Off, '1' : On
6	F0WID	W	0x1	Width pixel constellation. Double Buffered. '0' : Narrow, '1' : Wide
2:0	F0BY[2:0]	W	0x6	Hf_bypass (Beta): (range 0 .. 7). Double Buffered
Offset 0x61 - LORE_F10				
7:4	F1KC[3:0]	W	0x1	Kc. Double Buffered
3:0	F1KS[3:0]	W	0x2	Ks. Double Buffered
Offset 0x64 - LORE_N00				
7:6	N0CL[1:0]	W	0x2	Clip_offset. Sets clipping upper and lower bound. Double Buffered. '00' : lower = 17, upper = 238, '01' : lower = 18, upper = 237, '10' : lower = 20, upper = 235, '11' : lower = 24, upper = 231
5	N0CN	W	0x0	Clip_neglect. Double Buffered. '0' : Clipping range is defined, '1' : No clipping range is defined
4	N0NEL	W	0x1	NE_lowpass. Double Buffered. '0' : Output is not filtered, '1' : Output is low pass filtered with the following transfer function: H = 16 / (16-15 (1/z))
3	N0HN	W	0x0	Hist_neglect. Double Buffered. '0' : Conditional update noise lore estimator, '1' : Update lore noise estimator every field
2:1	N0HS[1:0]	W	0x2	Hist_set: definition of MinNrBlocks parameter. Double Buffered. '00' : MinNrBlocks = 256, '01' : MinNrBlocks = 1024, '10' : MinNrBlocks = 4096, '11' : MinNrBlocks = 16384,
0	N0WN	W	0x1	Win_neglect. Double Buffered. '0' : Noise measurement is done over window, '1' : Noise measurement is done over the entire field.
Offset 0x65 - LORE_N10				

Table 244: Columbus registers *…continued*

Bit	Symbol	Access	Reset Value	Description
7:0	N1WSTA[7:0]	W	0x00	0.25 x Window start pixel. Double Buffered
Offset 0x66 - LORE_N20				
7:0	N2WSTO[7:0]	W	0xD2	0.25 x Window stop pixel. Double Buffered
Offset 0x67 - LORE_N30				
7:0	N3WSTA[7:0]	W	0x00	0.25 x Window start line. Double Buffered
Offset 0x68 - LORE_N40				
7:0	N4WSTO[7:0]	W	0x48	0.25 x Window stop line. Double Buffered
Offset 0x6A - TGITU_W00				
7:0	W0VSTA[7:0]	W	0x02	Position of V start 1 (8 MSB's). Note: Vstart position(10:0) = TGITU_W0(7:0) & TGITU_W2(5:3)
Offset 0x6B - TGITU_W10				
7:0	W1VSTA[7:0]	W	0x2A	Position of V start 2 (8 MSB's). Note: Vstop position(10:0) = TGITU_W1(7:0) & TGITU_W2(2:0)
Offset 0x6C - TGITU_W20				
5:3	W2VS1[2:0]	W	0x7	Position of V start 1 (3 LSB's)
2:0	W2VS2[2:0]	W	0x0	Position of V start 2 (3 LSB's)
Offset 0x6D - TGITU_W30				
7:0	W3FT[7:0]	W	0x00	Position of F toggle to '0' (8 MSB's). Note: Ftoggle0 position(10:0) = TGITU_W3(7:0) & TGITU_W5(5:3)
Offset 0x6E - TGITU_W40				
7:0	W4FT[7:0]	W	0x27	Position of F toggle to '1' (8 MSB's). Note: Ftoggle1 position(10:0) = TGITU_W4(7:0) & TGITU_W5(2:0)
Offset 0x6F - TGITU_W50				
5:3	W5FT0[2:0]	W	0x1	Position of F toggle to '0' (3 LSB's)
2:0	W5FT1[2:0]	W	0x1	Position of F toggle to '1' (3 LSB's)
Offset 0x70 - INOUT_I00				
4	I0DMSD	W	0x1	'0' - DMSD (like) signals: Clkasa = 27 MHz. Test generator generates continuous ITU656 stream. '1' - AVIP (like) signals: Clkasa = 32.4 MHz (1fh) or 64.8 MHz (2fh). Test generator generates ITU656 stream in which data is valid during each 5 out of 6 clkasa periods. (32.4 MHz * 5/6 = 27 MHz).
3	I0ITU8	W	0x0	ITU8bit. '0' - ITU656 sync detection/generation on 10 bits. '1' - ITU656 sync detection/generation on 8 bits .
2	I0VBI	W	0x0	VBI_only. '0' - input is ITU656 with video data. '1' - input is ITU656 with only VBI lines.
1:0	I0ITU[1:0]	W	0x0	ITU mode: '00' - PAL, '01' - NTSC, '10' - ATSC, '11' - Reserved (test).
Offset 0x71 - INOUT_I10				
7:0	I1CL[7:0]	W	0x78	Clamp luminance level during ITU656
Offset 0x72 - INOUT_O00				

Table 244: Columbus registers *…continued*

Bit	Symbol	Access	Reset Value	Description
7:0	O0AVS[7:0]	W	0x80	ANC VBI SDID value (8 MSB's). Note: The default value is chosen unequal to zero to prevent false sync occurrences in ITU656 streams (8 bits and 10 bits). Programming of DID and SDID both on zero is not allowed for this reason. ANC VBI SDID (9:0) = INOUT_O0(7:0) & INOUT_O3(5:4)
Offset 0x73 - INOUT_O10				
7:0	O1AVD[7:0]	W	0x80	ANC VBI DID value (8 MSB's) for field2 (even field). Note: ActiveStop(9:0) = COMB_N3(7:0) & COMB_N4(1:0). Range 2 .. 1023. For proper operation ActiveStop > ActiveStart. ANC VBI DID field2(9:0) = INOUT_O1(7:0) & INOUT_O3(3:2).
Offset 0x74 - INOUT_O20				
7:0	O2AVD[7:0]	W	0x80	ANC VBI DID value (8 MSB's) for field1 (odd field). Note: ActiveStop(9:0) = COMB_N3(7:0) & COMB_N4(1:0). Range 2 .. 1023. For proper operation ActiveStop > ActiveStart. ANC VBI DID field1(9:0) = INOUT_O2(7:0) & INOUT_O3(1:0).
Offset 0x75 - INOUT_O30				
7	O3AV	W	0x1	'0' - disable Columbus ANC VBI insertion. '1' - enable Columbus ANC VBI insertion
5:4	O3AVS[1:0]	W	0x0	ANC VBI SDID value (2 LSB's). Note: During the copying of data in the output buffer at the moment of the vertical timing pulse (double buffering), alternating data is selected from the A and B registers. The result is that for consecutive fields the settings as programmed in registers A, B, A, B, .. etc. is used.
3:2	O3AVDF2[1:0]	W	0x0	ANC VBI DID value (2 LSB's) for field2 (even field). Note: Vstart position(10:0) = TGITU_W0(7:0) & TGITU_W2(5:3)
1:0	O3AVDF1[1:0]	W	0x0	ANC VBI DID value (2 LSB's) for field1 (oddfield). Note: Vstop position(10:0) = TGITU_W1(7:0) & TGITU_W2(2:0)
Offset 0x78 - RESERVED0				
7:0	RSD[7:0]	W	0x0B	Reserved
Offset 0x79 - RESERVED0				
7:0	RSDL[7:0]	W	0x9A	Reserved
Offset 0x7A - RESERVED0				
4	RSD	W	0x0	Reserved
Offset 0x7B - RESERVED0				
7:4	RSD[3:0]	W	0x8	Reserved
Offset 0x7C - INOUT_I60				
7:0	I6LF[7:0]	W	0x0B	First active line of a field (8 MSB's). Note: First active line of a field(8:0) = INOUT_I6(7:0) & INOUT_I4(4)
Offset 0x7D - INOUT_I70				
7:0	I7LVB[7:0]	W	0x9B	First line of the vertical blanking (8 MSB's). Note: First line of vertical blanking(8:0) = INOUT_I7(7:0) & INOUT_I4(3)
Offset 0x7E - RESERVED0				

Table 244: Columbus registers *…continued*

Bit	Symbol	Access	Reset Value	Description
7:0	RSD[7:0]	R	0xXX	Reserved
Offset 0x7F - RESERVED0				
7:0	RSD[7:0]	R	0xXX	Reserved

Table 245: Default setting for the system dependent parameters of Columbus

Register	Bits	Description	Default Value				
			PAL B2K	PAL M	PAL N	NTSC	Bypass
AMPCON	[7:0]	Amplitude control output signal	205	205	205	205	255
SF3TAPC	[7:4]	Spatial Filter 3 TapC	9	8	8	8	9
SF3TAP1	[3:0]	Spatial Filter 3 Tap1	-2	-3	-3	-3	-2
MCOLV	[7]	Color Motion Version	0	0	0	0	0
SF3TAP3	[6:4]	Spatial Filter 3 Tap3	3	0	0	0	3
SF3TAP2	[3:0]	Spatial Filter 3 Tap2	-4	-4	-4	-4	-4
SF3TAP4	[7:6]	Spatial Filter 3 Tap4	0	0	0	0	0
SF1WIDTH	[5]	Spatial Filter 1 Width	0	0	0	0	0
SF1TAP1	[4]	Spatial Filter 1 Tap1	1	1	1	1	1
SF1BYPASS	[3]	Spatial Filter 1 Bypass	1	1	1	1	1
SF2WIDTH	[2]	Spatial Filter 2 Width	0	1	1	1	0
SF2TAP1	[1]	Spatial Filter 2 Tap1	0	0	0	0	0
SPATDEL	[0]	Spatial Delta	0	0	0	0	0
TF3TAPC	[7:4]	Temp. Filter 3 TapC	6	4	4	4	6
TF3TAP1	[3:0]	Temp. Filter 3 Tap1	6	2	2	2	6
VSTOP_LSB	[7]	Vstop LSB	0	0	0	0	0
TF3TAP3	[6:4]	Temp. Filter 3 Tap3	-1	1	1	1	-1
TF3TAP2	[3:0]	Temp. Filter 3 Tap2	1	2	2	2	1
TF3TAP4	[7:6]	Temp. Filter 3 Tap4	-1	1	1	1	-1
TF1TAP1_1109	[5]	Temp. Filter 1 Width	0	1	1	1	0
TF1TAP1	[4]	Temp. Filter 1 Tap1	1	1	1	1	1
TF1BYPASS	[3]	Temp. Filter 1 Bypass	0	0	0	0	0
TF2WIDTH	[2]	Temp. Filter 2 Width	0	1	1	1	0
TF2TAP1	[1]	Temp. Filter 2 Tap1	1	1	1	1	1
TEMPDELTA	[0]	Temp. Delta	1	1	1	1	1
COUPFAD	[7]	CoupleFaders	0	0	0	0	0
VSTA	[6:0]	Vstart[1]	25	25	25	25	25
VSTO	[7:0]	Vstop, 8 MSB's[2]	155	131	155	131	155

[1] The default value is 3+22=25

[2] The default value for Vstop is 288+22=310 (PAL) and 240+22=262 (NTSC)

The following (18) register values are inserted in the ITU656 stream during VBI, in order to release the timing demands on the control I^{2}C bus:

COMB_M0, COMB_M1, COMB_M2, COMB_M3, SYSTEM_SELECT, COMB_P0, COMB_P1, NOISE_E9, NOISE_E10, NOISE_E11, NOISE_E12, NOISE_E13, NOISE_E0, NOISE_E1, NOISE_E2, NOISE_E3, NOISE_E4, LORE_N0.

4.8 Testing

4.8.1 Internal test generator

4.8.1.1 656 test generator

The 656 test generator generates a 656 compliant stream and is used for testing the functionality of the 656 encoder and decoder. The 656 stream can be injected at the front-end or the backend of the chip (see Figure 162).

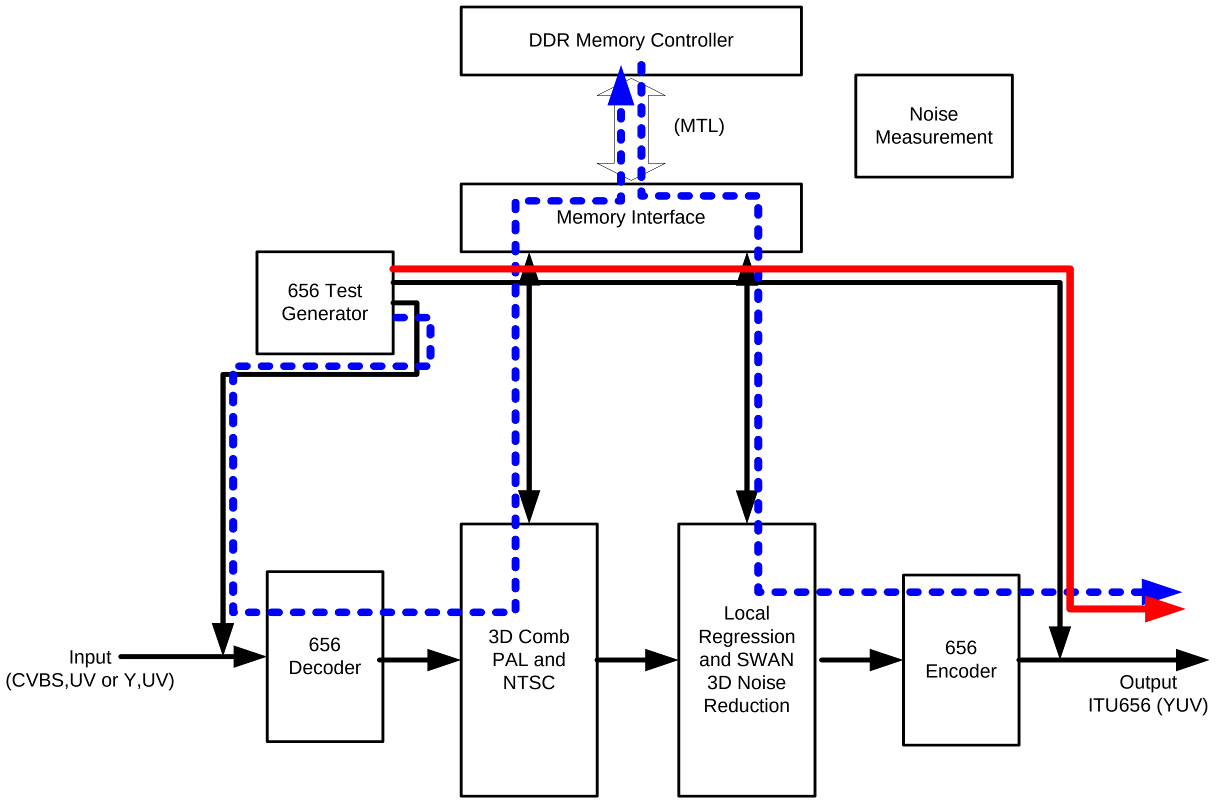


Fig 162.Possible data flow from the 656 pattern generator

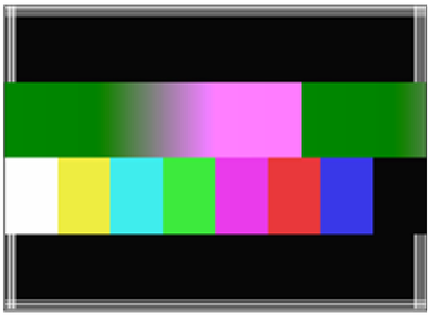


Fig 163.Picture 656 pattern generator

5. HD (High Definition) subsystem

5.1 HD block overview

The HD subsystem performs MPEG video decoding for up to MP@HL streams. It interfaces with PNX8550 and video coprocessor via tunnel interfaces, HD/SD using DV4 and DV5 inputs, and PNX8550 using DV1, DV2 and DV3 outputs.

The HD subsystem can also perform horizontal and vertical scaling of video images, and perform a range of video measurements on a stream.

A block diagram is shown in Figure 164.

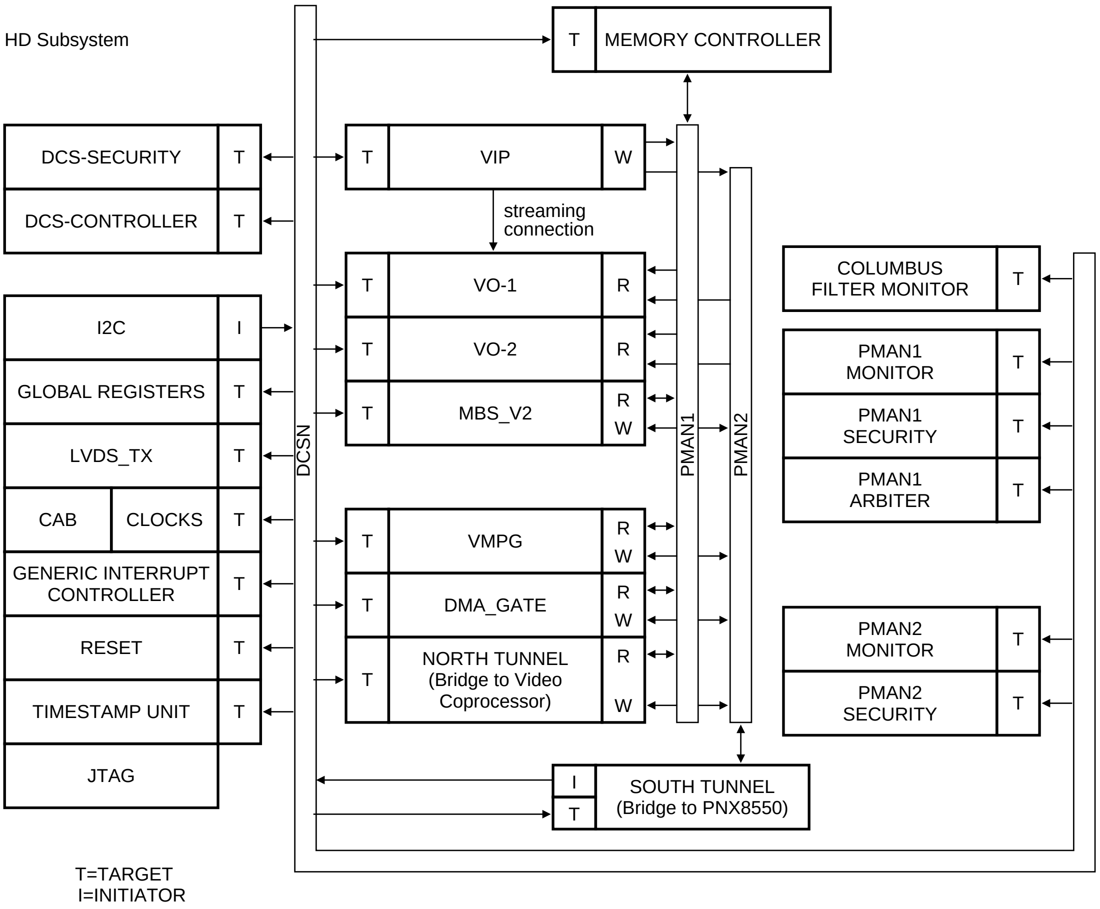


Fig 164.HD subsystem and interfaces

5.1.1 Bus architecture

Two different types of bus are used within the HD subsystem. The Device, Control and Status Network (DCSN) is used for control of agents within the HD subsystem. The Pipelined Memory Access Network (PMAN) is used for agents that require access to external memory. The combination of the two PMAN buses, the arbiter and the security blocks is referred to as the HUB.

DCS-Network (DCSN)

The DCSN consists of the interconnection bus between the agents and the control and security blocks. The bus has two initiators that can issue read and write transaction on the bus, and 23 targets that respond to transactions.

The initiator can be either the I^2C slave block, or the south tunnel. The south tunnel is connected to the PNX8550, so either the MIPS™ or TriMedia™ on the PNX8550 can access registers within the targets. The south tunnel is the principal interface for control of the PNX2015 HD subsystem as it operates faster than the I^2C.

The target agents all have defined Memory Mapped Input/Output (MMIO) Registers; all agents occupy at least a 4 kB MMIO address range.

The DCS-Controller block performs all control functions for the DCSN. It provides for programmable timeout generation, and handling of error conditions, together with arbitration on the bus.

The DCS-Security block is used for access rights for each agent. Each agent can be configured to allow a specified initiator (I^2C or south tunnel) access to the block. If an initiator tries to access an agent that it does not have rights to then an interrupt is generated.

HUB

The HUB consists of the PMAN1 and PMAN2 networks together with the security and arbiter blocks for each network. The PMAN1 network allows agents to access the local memory of the PNX2015 via the memory controller. All agents have access to this memory. The PMAN2 network allows agents to access PNX8550 device via the south tunnel. All agents have access to the south tunnel, with the exception of the 2D read/write transaction of the MPEG Video Decoder (VMPG) block.

An agent can access either the PMAN1 or PMAN2 based on the address of the transaction. If the address is below a programmable value (defined in the Global Registers) then PMAN2 is accessed, if it is above or equal to it, then PMAN1 is accessed. It is not possible for an agent to access both PMAN1 and PMAN2 simultaneously. The programmable value is set to the value of DDR-SDRAM attached to the PNX8550.

5.1.2 Memory map

The HD subsystem has two address spaces, one for the MMIO Registers on the DCSN, and one for the HUB.

MMIO registers

Each of the agents requires registers to be read and/or written to control the functionality.

The address range of the south tunnel limits the MMIO address range for the PNX2015 HD subsystem. The south tunnel can only supply a 19-bit address value; this limits the MMIO range for devices connected to the south tunnel to a maximum of 512 kB. The PNX2015 requires a contiguous address range of 128 kB for control of the agents.

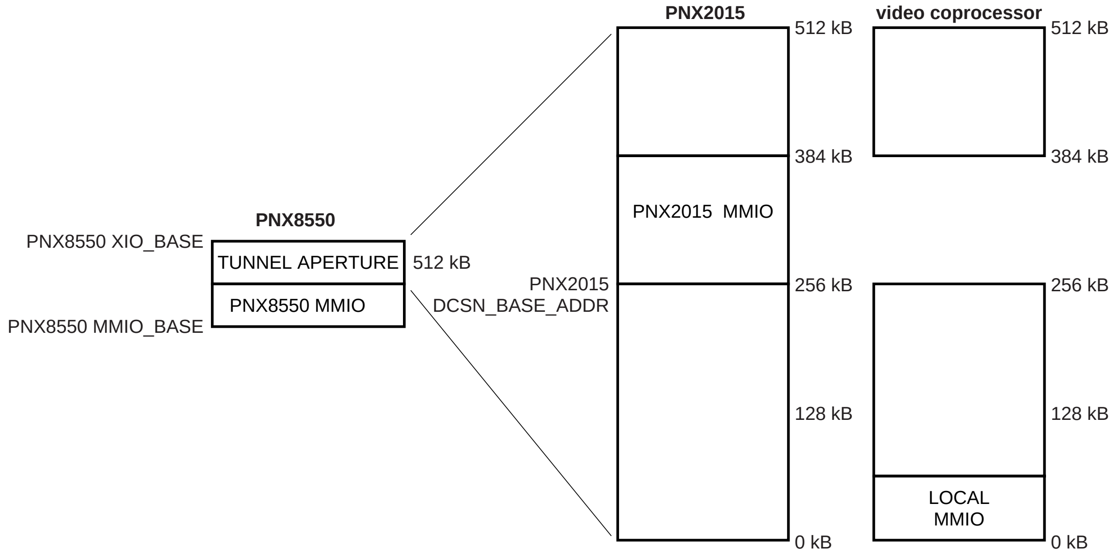


Fig 165.MMIO addressing (default values)

The location of the PNX2015 MMIO region within the 512 kB address range is configurable via registers in the Global Register block. The base address (DCSN_BASE_ADDR) for the PNX2015 MMIO can be set at either 0,128, 256 or 384 kB see Figure 165. The default value for the base address is 256 kB. Any MMIO transactions on the DCSN that are not within the defined 128 kB range are routed towards the north tunnel (i.e. video coprocessor).

The MMIO offset for the agents is shown in Table 246. All agents have a 4 kB aperture.

Table 246: MMIO register map

Cell name	Aperture start (offset from DCSN base address)
SOUTH_Tunnel	0x00016000
NORTH_Tunnel	0x00015000
DDRControl	0x00014000
COLUMBUS_Monitor	0x00013000
PMAN2_Security	0x00012000
PMAN2_Monitor	0x00011000
VMPG	0x0000F000
TSU	0x0000E000
MBS2	0x0000D000
VIP1	0x0000C000
NHP_VO2	0x0000B000
NHP_VO1	0x0000A000
PMAN1_Arbiter	0x00009000
PMAN1_Security	0x00008000
PMAN1_Monitor	0x00007000

Table 246: MMIO register map *…continued*

Cell name	Aperture start (offset from DCSN base address)
Reset	0x00006000
GIC	0x00005000
HDClocks	0x00004000
GLOBALREG	0x00002000
DCS_SECURITY	0x00001000
DCSControl	0x00000000

System memory

The HUB allows access to either local memory attached to the PNX2015, or memory attached to the PNX8550. The PNX8550 DRAM is always located in the address range 0 to DRAM_SIZE. For PNX8550 DRAM_SIZE can be 32, 64 or 128 MB.

The PNX2015 local memory sits directly on top of the PNX8550 DRAM address space. A global register LOCAL_BASE that defines the start address of the local memory defines this. For example, if PNX8550 has 128 MB of DRAM attached and the PNX2015 32 MB, the register LOCAL_BASE = 128 MB. Address on the HUB less than 128 MB would be routed towards the south tunnel on PMAN2. Address on the HUB equal to, or above 128 MB, would be routed towards the local memory on PMAN1.

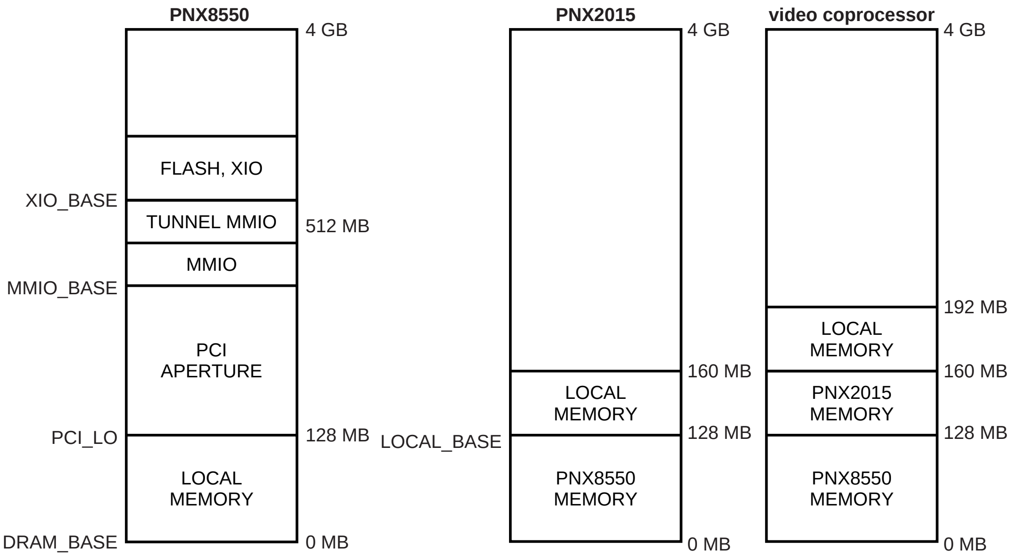


Fig 166.System DRAM memory addressing

5.1.3 South tunnel

The PNX2015 south tunnel is connected to the north tunnel of the PNX8550. The tunnel is used for MMIO configuration and DMA (Direct Memory Access) data transactions. MMIO transactions to and from the PNX2015 are initiated by the PNX8550. In consequence, the HD Subsystem is configured by PNX8550. DMA data transactions to and from the PNX8550 are initiated by the PNX2015.

The tunnel receive channel and the transmit channel both operate up to a maximum frequency of 225 MHz clock rate. (Note that the PNX8550 receive channel is limited to 192 MHz). This allows a peak data rate of 300 MB for both transmit and receive paths. The PNX8550 north and PNX2015 south tunnels are initialised before tunnel transactions can commence. For the south tunnel this is achieved via the I^2C.

5.1.4 North tunnel

The PNX2015 north tunnel is connected to the south tunnel of the video coprocessor (referred to as spid in the register descriptions). The tunnel is used for MMIO configuration and DMA data transactions. Configuration of the video coprocessor is via MMIO transactions initiated by the PNX8550. The video coprocessor initiates DMA data transactions to and from the PNX8550 and/or the PNX2015.

The tunnel receive channel and the transmit channel both operate up to a maximum frequency of 225 MHz clock rate. This allows a peak data rate of 300 Mb/s for both transmit and receive paths. The PNX2015 north and video coprocessor south tunnels are initialised before tunnel transactions can commence. For the north side this can be achieved via either MMIO accesses across the PNX8550 to PNX2015 tunnel or I^2C accesses. Initialisation for the video coprocessor south tunnel is done via the I^2C.

5.1.5 Tunnel interface efficiency

Reading 720 × 480 at 8-bit resolution from PNX8550 memory into video coprocessor, scaling to 1366 × 768 at 10-bit resolution and writing back, can be performed within less than 15 ms.

The video coprocessor can issue four consecutive 128 byte write requests and five consecutive 128 byte read requests.

5.1.6 Global register block

Global register block contains MMIO registers that are shared by various agents and buses. The registers are either general-purpose (read/write) and can be used as temporary storage, or dedicated to specific module functions.

5.1.6.1 Register summary

Table 247: Register Summary

Address	Name	Description
TransactionTarget_2 address space		
0x0	SYS_ENDIANNESS	System endianness register
0x200	DRAM_LO	Internal bus DRAM low address register
0x204	DRAM_HI	Internal bus DRAM high address register
0x208	APERTURE_WE	Enable DRAM_LO and DRAM_HI registers to be writable
0x300	COLUMBUS_SRC_SEL	Selection of Columbus clock and data source
0x304	COLUMBUS_DDR_OFF	DDS memory offset value for Columbus
0x400	DVO1_SELECT	DV output 1 source selection
0x404	DV02_SELECT	DV output 2 source selection
0x408	DVO3_SELECT	DV output 3 source selection

Table 247: Register Summary *…continued*

Address	Name	Description
0x500	SCRATCH0	32 bit writable and readable register
0x504	SCRATCH1	32 bit writable and readable register
0x508	SCRATCH2	32 bit writable and readable register
0x50C	SCRATCH3	32 bit writable and readable register
0x510	SCRATCH4	32 bit writable and readable register
0x514	SCRATCH5	32 bit writable and readable register
0x518	SCRATCH6	32 bit writable and readable register
0x51C	SCRATCH7	32 bit writable and readable register
0x600	MM_CTRL	Memory pad control register
0x604	TUN_V2_CTRL	Tunnel pad control register
0x608	TUN_SPID_CTRL	Tunnel pad control register
0x60C	GLB_SPARE_CTRL	Spare registers control register
0x700	DCSN_BASE_ADDR	Offset for DCSN base address
0x704	DMA_GATE_OFF	DMA gateway 4k DCSN offset
0x800	CMD_LIMIT_PMAN1	Limit on total number of pipeline transactions
0x804	READ_LIMIT_PMAN2	Limit on pipeline read transactions
0x808	WRITE_LIMIT_PMAN2	Limit on pipeline write transactions
0x80C	PMAN_ROUTE_VALUE	Address routing value for PMAN
0x810	PMAN_ROUTE_MASK	Address routing mask for PMAN
0x814	MTL_PMAN_EN	MTL to PMAN address bit enable mask
0x818	STREAM_CTL	VIP to VO stream control
0xFF4	GLB_REG_PWR_DWN	Powerdown bit for the global registers
0xFFC	GLB_REG_MOD_ID	Module Identification and revision information

5.1.6.2 Register tables

Table 248: GLOBALREG Registers

Bit	Symbol	Access	Reset Value	Description
Offset 0x0 - SYS_ENDIANNESS				
31:1	RSD[31:1]	R	0xXXXX XXXX	Ignore upon read. Write as zeros
0	BIGENDIAN	R/W	0x0	System endianess 0: little endian 1: big endian
Offset 0x200 - DRAM_LO				
31:16	DRAMLO[15:0]	R/W	0x0000	Lowest internal bus address mapped to DRAM, granularity f 64k. Internal bus DRAM aperture is reset to start at 0.
15:0	RSD[15:0]	R	0xXXXX	Ignore during writes and read as zeroes
Offset 0x204 - DRAM_HI				

Table 248: GLOBALREG Registers *…continued*

Bit	Symbol	Access	Reset Value	Description
31:16	DRAMHI[15:0]	R/W	0x0400	First address outside internal bus DRAM space, granularity of 64k. Internal bus DRAM aperture is reset to 64Mb
15:0	RSD[15:0]	R	0x0000	Ignore during writes and read as zeroes
Offset 0x208 - APERTURE_WE				
31:1	RSD[31:1]	R	0xXXXX XXXX	Ignore during writes and read as zeroes
0	DRAM_WE	R/W	0x0	0 = writing to DRAM_LO or DRAM_HI is disabled; 1: When writing to either DRAM_LO or DRAM_HI occurs, this bit is automatically cleared.
Offset 0x300 - COLUMBUS_SRC_SEL				
31:2	RSD[31:2]	R	0x00000 000	Reserved
1:0	COLUMBUS_SRC[1:0]	R/W	0x0	Memory pad reference voltage generation 0 0 = No data clock set to xtal 0 1 = Data and clock from AVIP1 1 0 = Data and clock from AVIP2 1 1 = Reserved
Offset 0x304 - COLUMBUS_DDR_OFF				
31:20	COLUMBUS_OFF[11:0]	R	0x000	MSB's of the address used by columbus for DDR access
19:0	RSD[19:0]	R	0x00000	Ignore during writes and read as zeroes
Offset 0x400 - DVO1_SELECT				
31:3	RSD[31:3]	R	0x00000 000	Reserved
2:0	DVO_1_SEL[2:0]	R/W	0x0	DV output 1 source selection 0 0 0 = DVO1 from AVIP1 0 0 1 = DVO1 from AVIP2 0 1 0 = DVO1 from COLUMBUS 0 1 1 = DVO1 from VO1 lower 1 0 0 = DVO1 from VO2 lower 1 0 1 = Reserved. 1 1 1 = Reserved
Offset 0x404 - DV02_SELECT				
31:3	RSD[31:3]	R	0x00000 000	Reserved
2:0	DVO_2_SEL[2:0]	R/W	0x1	DV output 2 source selection 0 0 = DVO2 from VO1 lower 0 1 = DVO2 from VO2 lower 1 0 = DVO2 from VO1 upper 1 1 = DVO2 from VO2 upper
Offset 0x408 - DVO3_SELECT				
31:3	RSD[31:3]	R	0x00000 000	Reserved
2:0	DVO_3_SEL[2:0]	R/W	0x1	DV output 3 source selection 0 0 = DVO3 from AVIP1 0 1 = DVO3 from AVIP2 1 0 = DVO3 from COLUMBUS 1 1 = DVO3 from VO1 lower
Offset 0x500 - SCRATCH0				
31:0	SCRATCH_0[31:0]	R/W	0x00000 000	32 bit writeable and readable register
Offset 0x504 - SCRATCH1				

Table 248: GLOBALREG Registers *...continued*

Bit	Symbol	Access	Reset Value	Description
31:0	SCRATCH_1[31:0]	R/W	0x00000000	32 bit writeable and readable register
Offset 0x508 - SCRATCH2				
31:0	SCRATCH_2[31:0]	R/W	0x00000000	32 bit writeable and readable register
Offset 0x50C - SCRATCH3				
31:0	SCRATCH_3[31:0]	R/W	0x00000000	32 bit writeable and readable register
Offset 0x510 - SCRATCH4				
31:0	SCRATCH_4[31:0]	R/W	0x00000000	32 bit writeable and readable register
Offset 0x514 - SCRATCH5				
31:0	SCRATCH_5[31:0]	R/W	0x00000000	32 bit writeable and readable register
Offset 0x518 - SCRATCH6				
31:0	SCRATCH_6[31:0]	R/W	0x00000000	32 bit writeable and readable register
Offset 0x51C - SCRATCH7				
31:0	SCRATCH_7[31:0]	R/W	0x00000000	32 bit writeable and readable register
Offset 0x600 - MM_CTRL				
31:2	RSD[31:2]	R	0xXXXXXXXX	Ignore upon read. Write as zeros
1	MM_REFEN	R/W	0x0	Memory pad reference voltage generation 0: Disable internal reference voltage generation. An external voltage reference can now be used. 1: Enable internal reference voltage generation. No external voltage reference is required.
0	MM_PON	R/W	0x0	Memory pad power on 0: Power disabled to memory pads, pads have TTL levels 1: Power is enabled to memory pads. Pads work at SSTL levels
Offset 0x604 - TUN_V2_CTRL				
31:2	RSD[31:2]	R	0xXXXXXXXX	Ignore upon read. Write as zeros
1	RSD_Bit1	R/W	0x0	Reserved for future use
0	LVDS_PON	R/W	0x0	LVDS pad bandgap reference power on. 0: Bandgap reference for LVDS pads enabled. 1: Bandgap reference for LVDS pads disabled.
Offset 0x608 - TUN_SPID_CTRL				

Table 248: GLOBALREG Registers *…continued*

Bit	Symbol	Access	Reset Value	Description
31:2	RSD[31:2]	R	0xXXXX XXXX	Ignore upon read. Write as zeros
1	TUN_SPID_REFEN	R	0x0	Tunnel pad reference voltage generation 0: An external voltage reference can now be used. 1: No external voltage reference is required.
0	TUN_SPID_PON	R	0x0	Tunnel pad power on. 0: pads have TTL levels. 1: Pads work at SSTL levels.
Offset 0x60C - GLB_SPARE_CTRL				
31:8	RSD[31:8]	R	0xXXXX XX	Ignore during writes and read as zeroes
7:5	GLBREG_SPARE[2:0]	R/W	0x0	Spare control bits
4	AVIP2_VSYNC2	R/W	0x0	Avip2 Vsync2 input invert control 0: Input non inverting 1: Inout Inverting
3	AVIP2_VSYNC1	R/W	0x0	Avip2 Vsync1 input invert control 0: Input non inverting 1: Input inverting
2	AVIP1_VSYNC2	R/W	0x0	Avip1 Vsync2 input invert control 0: Input non inverting 1: Input inverting
1	AVIP1_VSYNC1	R/W	0x0	Avip1 Vsync1 input invert control 0: Input non inverting 1: Input inverting
0	AVIP2_I2D_SEL	R/W	0x0	Avip 2 only 0: AVIP2 Video 1: AVIP1 CVBS (AvPip)
Offset 0x700 - DCSN_BASE_ADDR				
31:2	RSD[31:2]	R	0xXXXX XXXX	Ignore upon read. Write as zeros
1:0	DCSN_BASE[1:0]	R/W	0x2	DCSN Base address offset 0 0 = 0 0 1 = 128 1 0 = 256 1 1 = 384
Offset 0x704 - DMA_GATE_OFF				
31:12	DMA_BASE[19:0]	R/W	0x00000	Selects the base 4k address offset for DMA
11:0	RSD[11:0]	R	0xXXX	Ignore upon read. Write as zeros
Offset 0x800 - CMD_LIMIT_PMAN1				
31:4	RSD[31:4]	R	0x00000 00	Always return zeroes when read, not changed when written

Table 248: GLOBALREG Registers *…continued*

Bit	Symbol	Access	Reset Value	Description
3:0	WR_LIM_PMAN1[3:0]	R/W	0x3	Limits the total number of pipelined transactions (after top-level arbitration) in PMAN2 0 - No limit 1-15 = Limit of number of pipelined transactions
Offset 0x804 - READ_LIMIT_PMAN2				
31:4	RSD[31:4]	R	0x0000000	Always return zeroes when read, not changed when written
3:0	RD_LIM_PMAN2[3:0]	R/W	0x3	Limits the number of pipelined read transactions (after top-level arbitration) in PMAN2 0 = No Limit; 1-15 = Limit of number of pipelined transactions; 0: No Limit;
Offset 0x808 - WRITE_LIMIT_PMAN2				
31:4	RSD[31:4]	R	0x0000000	Always return zeroes when read, not changed when written
3:0	WR_LIM_PMAN2[3:0]	R/W	0x2	Limits the number of pipelined write transactions (after top-level arbitration) in PMAN2 0 = No Limit; 1-15 = Limit of number of pipelined transactions; 0: No Limit;
Offset 0x80C - PMAN_ROUTE_VALUE				
31:24	ROUTE_VALUE[7:0]	R/W	0x80	Address route value for PMAN applied to address bits [31:24] < ROUTE_VALUE routed to V2 Tunnel > ROUTE_VALUE routed to local DDR See route_mask for more info.
23:0	RSD[23:0]	R	0x000000	Always return zeroes when read, not changed when written
Offset 0x810 - PMAN_ROUTE_MASK				

Table 248: GLOBALREG Registers *…continued*

Bit	Symbol	Access	Reset Value	Description
31:24	ROUTE_MASK[7:0]	R/W	0x80	Address route mask for PMAN applied to address bits [31:24] The HUB in the PNX2015 contains two PMAN networks. The target PMAN network is decided based on an address decoding of the upper 8 bits. The upper bits are compared with the value set in PMAN_ROUTE_VALUE after application with the mask set by PMAN_ROUTE_MASK. All matching addresses are sent to PMAN2, routed to PNX8550 in the JAguar system via the South TUnnel. If the address match fails, then the transaction is routed to PMAN1 (PNX2015 DDR). This address decoding is done in the Gizmos and the output of the Hub is on the appropriate MTL Port Mode PMAN_ROUTE_MASK PMAN_ROUTE_VALUE PMAN2 only (always pnx8550) 00 00 PMAN1 only (always pnx2015) 00 FF PMAN2 if address < 128 Mbytes F8 00 PMAN1 if address > 128 Mbytes PMAN2 if address < 256 Mbytes F0 00 PMAN1 if address > 256 Mbytes
23:0	RSD[23:0]	R	0x000000	Always return zeroes when read, not changed when written
Offset 0x814 - MTL_PMAN_EN				
31:0	MTL_PMAN_ADDR_EN[31:0]	R/W	0x00000000	MTL to PMAN Address bit enable mask
Offset 0x818 - STREAM_CTL				
31:2	RSD[31:2]	R	0xXXXXXXXX	

Table 248: GLOBALREG Registers *…continued*

Bit	Symbol	Access	Reset Value	Description
1	VIVO_STRM_ENA	R/W	0x0	Enables direct streaming from VIP to VO 0: Disable Streaming 1: Enable Streaming
0	VIVO_STRM_RST	W	0x0	Resets direct streaming from VIP to VO 0: no effect 1: flush streaming connection
Offset 0xFF4 - GLB_REG_PWR_DWN				
31	POWER_DOWN	R/W	0x0	Powerdown register for the module 0 = Normal operation of the peripheral. This is the reset value. 1 = Module is powered down and module clock can be removed. At power down, module responds to all reads with DEADABBA (except for reads of powerdown bit) and all writes with ERR ack (except for writes to power down bit). 0: Normal operation of the peripheral. This is the reset value. 1: Module is powered down and module clock can be removed.
30:0	RSD[30:0]	R	0xXXXX XXXX	
Offset 0xFFC - GLB_REG_MOD_ID				
31:16	MODULE_ID[15:0]	R	0xA08B	Unique 16-bit code. Module ID 0 and -1 are reserved for future use.
15:12	MAJOR_REV[3:0]	R	0x0	Major Revision ID.
11:8	MINOR_REV[3:0]	R	0x0	Minor Revision ID.
7:0	MODULE_APERTURE_ SIZE[7:0]	R	0x00	Encoded as: Aperture size = 4K*(bit_value+1). The bit value is reset to 0 meaning a 4K aperture for the Global register 2 module according to the formula above.

5.1.7 Memory bandwidth monitor

There are two memory bandwidth monitors in the system. The first monitors transactions between PMAN1 and the memory controller, the second monitors transactions between PMAN2 and the south tunnel.

The monitors can be programmed to monitor either latency of a transaction, or number of transaction for a single agent. The monitor can work on read only, write only or total transactions for a given agent.

The memory monitor is used for performance analysis, and system optimization during system development.

5.1.8 Time Stamp Unit (TSU)

To ensure synchronization, video streams are timestamped as they enter and exit the PNX2015.

To allow the TV system synchronisation of timestamps, the timestamp clock must originate from the same source; this is the reason for using PNX8550 as the clock source for PNX2015. The timestamp unit supports the following functionality:

- Monitoring of rising, falling, or both edge of events
- 31-bit timestamp plus direction bit
- Resolution of timestamp 13.5 MHz/ > 75 ns

Refer to Ref. 8 for details of TSU registers.

5.1.9 I^{2}C

The I^{2}C slave receiver/transmitter in the HD subsystem is primarily used for initialization of the south tunnel. Once the south tunnel is initialized, all control of the HD subsystem is via MMIO transactions. The following features are available:

- I^{2}C slave receiver/transmitter.
- Up to 400 kHz operation.
- 7-bit I^{2}C slave address.
- DSCN initiator.
- 32-bit sub-address (DSCN address).
- I^{2}C data must be transmitted big endian i.e. msb first.

5.1.10 Video Input Processor (VIP)

The VIP allows for either single stream ITU-656, or dual stream ITU-1120 video to enter the HD subsystem.

The following functions are available:

- 8/10-bit YUV Interface with embedded (ITU-656) or explicit syncs.
- 16/20-bit Y+UV Interface with embedded or explicit syncs.
- Horizontal downscaling of video stream.
- Video stream capture to memory, or direct stream to VO-1.
- Ancillary data capture to memory.

5.1.11 Video Output (VO)

The HD subsystem has two Video Outputs, labelled VO-1 and VO-2. They have identical functionality, except that VO-1 can additionally be directly connected to the VIP for streaming video. Both VO-1 and VO-2 can accept data from system memory.

The following functions are available:

- Conversion format YUV 4:2:0 to YUV 4:2:2.
- No video processing improvements.
- Output formats 8 to 10-bit, or 16 to 10-bit YUV 4:2:2.

- Maximum pixel processing rate 43 Mpix/s for 8 to 10-bit output.
- Maximum pixel processing rate 86 Mpix/s for 16 to 20-bit output.
- Maximum clock frequency 86 MHz.

5.1.12 Memory Based Scaler (MBS_V2)

The Memory Based Scaler (MBS) performs processing on images in main memory. The MBS can either be controlled task by task with a TM3260, or it can be given a list of de-interlacing and scaling tasks.

The following functionality is available:

- De-interlacing using either a median 2-field majority select, or 3-field majority select algorithm with an edge detect/correct post-pass.
- Edge detect/correct on an input frame that has been software de-interlaced.
- Horizontal and vertical scaling (on the input image or on the result of edge detect/correct stage).
- Linear and non-linear aspect ratio conversion.
- Anti-flicker filtering.
- Conversions from any input pixel format to any non-indexed pixel format, including conversions between 4:2:0, 4:2:2 and 4:4:4.

5.1.13 Video MPEG decoder (VMPG)

VMPG parses and decodes a given number of macroblocks or entire field/frame. One HD MPEG2 stream (MP@HL) in any of the 18 ATSC formats can be decoded. Alternately, two SD MPEG2 streams (MP@ML) may be processed simultaneously.

VMPG provides a VLD only mode. In this mode, it outputs Run Length Pairs and Motion Vectors of a slice to system memory, but the slice is not decoded to pixels. This mode can be used for software processing on MPEG2 tokens, such as transrating to a lower bit rate MPEG2 stream, or partial decoding and transcoding to other video formats.

5.1.14 Memory controller

The Memory controller is a multi-port memory controller for DDR-SDRAM devices.

The following features are available:

- JEDEC compliant
- Support for single 16-bit DDR-SDRAM memory devices.
- 200 MHz clock speed, 400 MHz data rate.
- Supports 64, 128, 256 Mbits DDR-SDRAM memories.
- Maximum of four open pages.
- Maximum address range 32 MB.
- Programmable latency.

5.2 Reset module

The chip input RESET_IN is an active low signal that provides the global "hard" system reset to the PNX2015 chip (with exception of the TV Microprocessor). This input is received by the Reset Module which provides the reset sequencing and reset management to all other modules.

Remark: All external devices on the tunnel interface need to be reset following system reset.

5.2.1 Features

This Reset Module generates a reset to all DCS bus peripherals, including the AVIPs and 3D comb filter.

This reset is triggered or released in a number of ways:

- External reset input to the PNX2015.
- Programmable "do software system reset", which also asserts resets.

5.2.2 Functional description

Reset is a slave module on the DCS bus, as shown in Figure 167.

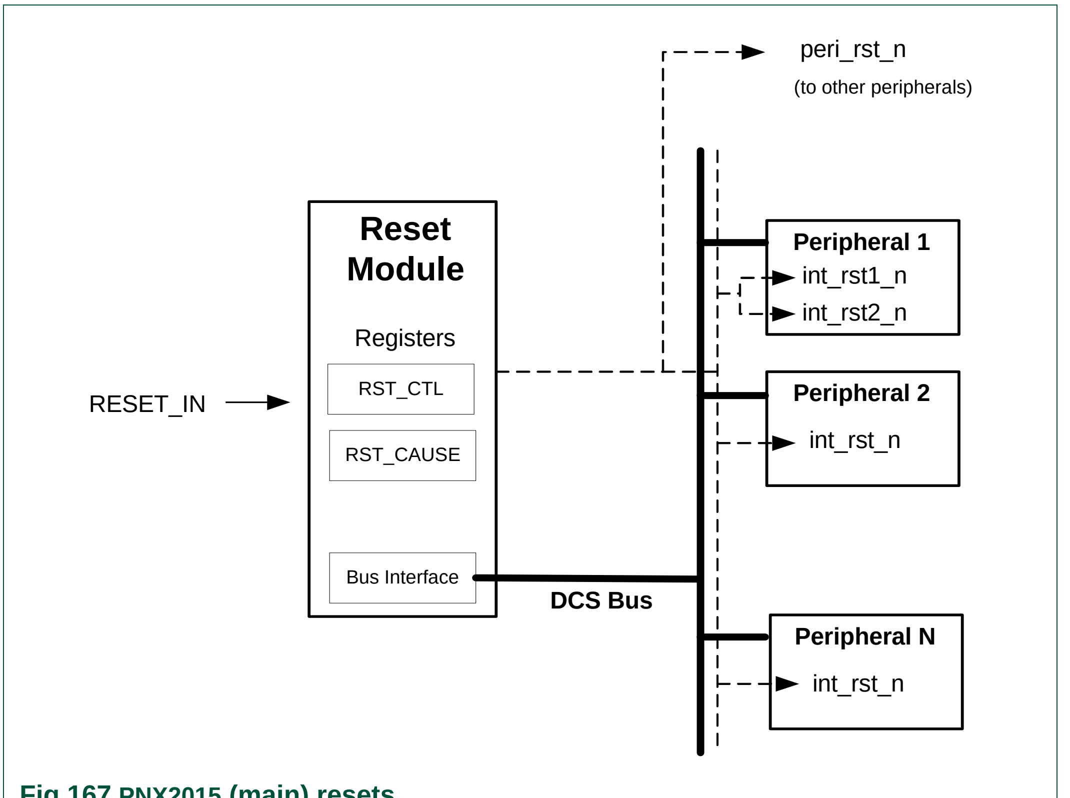


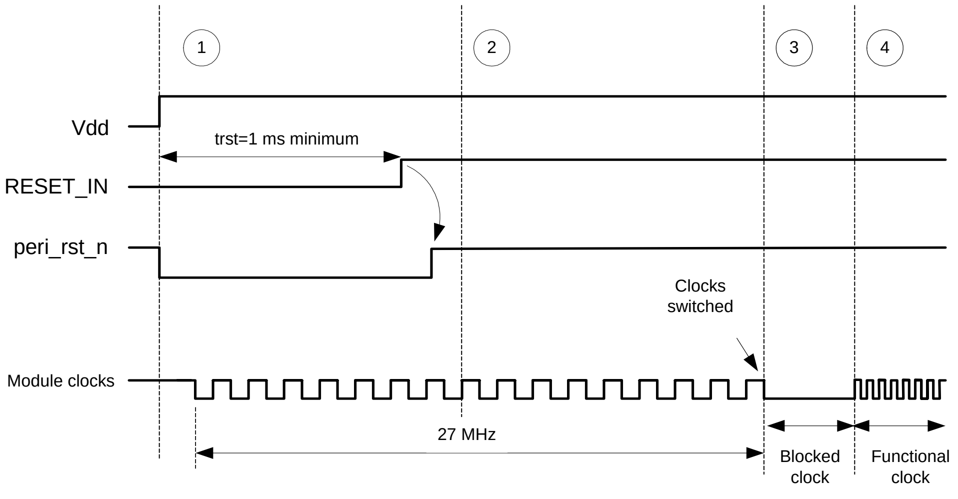
Fig 167.PNX2015 (main) resets

5.2.2.1 General operation

During normal boot of the system, the external reset input to PNX2015, RESET_IN, are asserted following power up of the system. This causes the assertion of the reset signal as shown in Figure 168. When the Clock Module receives the peri_rst_n, it ensures that all modules receive the 27 MHz clk_ip input. The 27 MHz clock to all modules remains until programming of registers in the Clock Module to switch from 27 MHz to the functional module clocks. This allows all modules to be reset synchronously with the 27 MHz clock.

After de-asserting the RESET_IN signal, the peri_rst_n is also de-asserted and all peripherals release their internal resets synchronously.

Following Reset, the Clock Module registers are programmed, resulting in a switch from the 27 MHz clock to the separate module functional clocks. This is shown in Figure 168.



1 - RESET_IN asserted for 1 msmin after power-good. peri_rst_n follows the release of RESET_IN. Clock Module kicks off 27MHz clock to all modules.

2 - All module resets sync to 27 MHz and are reset at the same time.

3 - All module clocks are blocked after programming registers in the Clock Block to ensure safe, glitchless switch over from 27MHz to module clocks.

4 - All modules are now receiving their required clocks for normal function.

Fig 168.Reset Timing Diagram

During power up, RESET_IN remains asserted for a minimum of 1 ms (Trst). At all other times RESET_IN is asserted for a minimum of 100 µs.

Remark: The entire sequence shown in Figure 168 is repeated in the event of a software reset. In this case the clock module switches all module clocks to 27 MHz; peri_rst_n is asserted for 100 µs before being released.

The Reset Module has two major blocks:

- PIO interface which allows DCS bus read/write access to the Reset Module configuration registers.
- PERI_RST State Machine generates reset to peripherals, including MMI, AVIPs and
- 3D comb filter.

The following sections describe the operation of the reset state machines. All reset signals are generated “glitch-free.”

PERI_RST State Machine

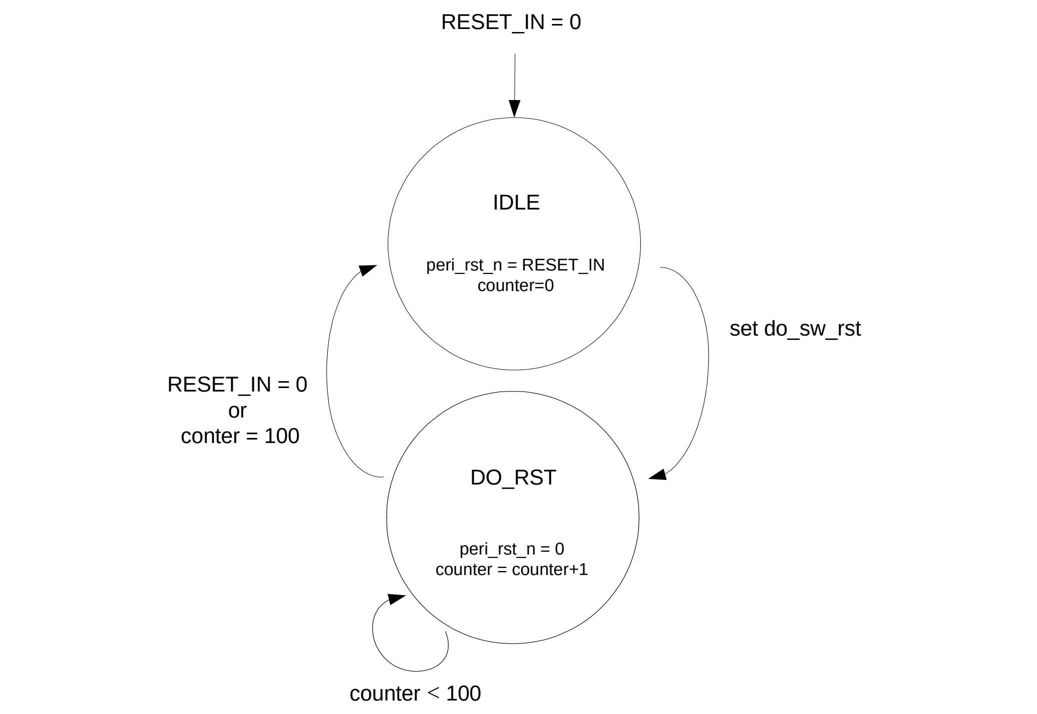


Fig 169.PERI_RST state machine

The peri_rst_n signal is asserted if a software reset is programmed or an external RESET_IN is asserted. Software reset cause peri_rst_n to be asserted for 100 µs. A counter is triggered to count 100 µs and runs at a frequency of 27 MHz during reset.

The peri_rst_n signal is synchronized in each DCS bus peripheral.

5.3 Clock generation

5.3.1 Overview

The purpose of the Clock module is to generate and control all HD Subsystem module and bus clocks in PNX2015. The clock system architecture includes the following features:

- Generation of HD Subsystem module clocks.
- Generation of DCSN and memory bus clocks.
- Register settings are software-controlled in the Clock module allowing module clocks to be stopped or changed to a lower frequency. This allows a net saving of power when a module is not used.
- Clock control (frequency transitions and switching) occurs without generation of glitches that may lead to potential malfunction of the module.

5.3.2 Functional description

The Clock module has two main interfaces (as shown in Figure 170).

- An interface to a Custom Analog Block (CAB). The CAB module includes all PLLs, several high speed clock dividers, and several DDS (Direct Digital Synthesizer) blocks which provide the main source of clocks to the Clock module.
- An interface to the DTL-MMIO bus. Clock controls are programmed via this interface.

The clk_in clock provides the source clock for the 1.7 GHz PLL in the CAB block, and for the low jitter DDR PLL. The PLLs are programmable from the Clock module registers to generate a range of possible frequencies. The DDS blocks are required to make slight adjustments to each video clock to track sources. Software controls this tracking by programming the DDSs to within 6.6 ns (p-p).

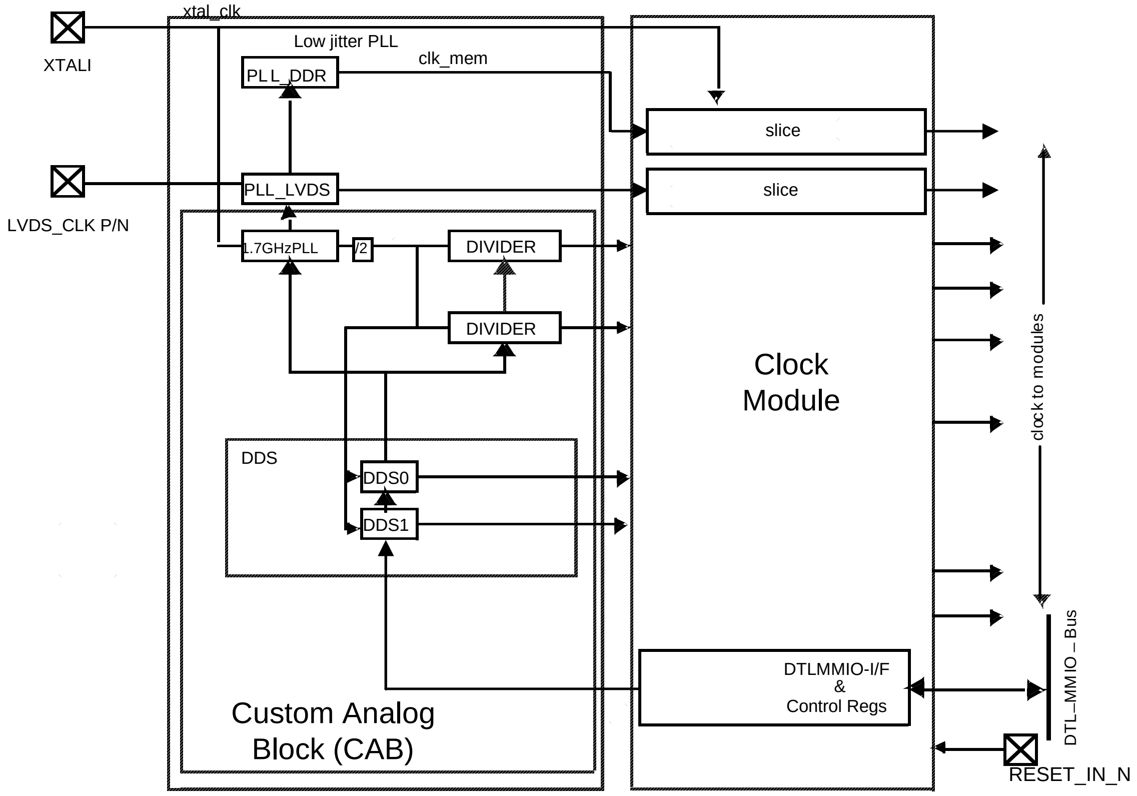


Fig 170.DVI clock block diagram

The Clock module consists of a DTL-MMIO interface with programmable Clock and PLL control registers, and control logic for every clock generated.

The clock control logic consists of:

- Programmable dividers, controlled by configuration registers
- Switching of clocks. Clocks are typically switched when:

PLLs or dividers are reprogrammed.

Clocks are switched on/off for power saving during Standby.

Following reset of the chip when clocks are switched from 27 MHz to their programmed functional frequencies.

Remark: All clocks to the modules are generated in the Clock module. Clocks coming from external sources into the PNX2015 are fed through the Clock module, so that they may undergo the controls required during reset and power down.

5.3.2.1 General operation

System level clocks: Clocks available to the PNX2015 modules are summarized in Table 249 .

Table 249: System level and internal clocks

Bus or Module	Signal Name	Description	Available Frequencies	Clock Module Control
Memory Bus, PMAN Hub, MMON	clk_mem	clock to memory interface and PMAN hub	low frequencies, 100MHz, 125MHz, 133MHz, 143MHz, 183MHz up to 225MHz [1]	PLL_DDR_CTL & CLK_MEM_CTL register
DCS Bus	clk_dcs	clock to DCS bus	72MHz, 157MHz up to 192MHz [1]	CLK_DCS_CTL register
DTL Bus, EJTAG, RESET, GLBREG,	clk_dtl	clock to DTL bus	66MHz, 133MHz up to 157MHz [1]	CLK_DTL_CTL register
Tunnel	clk_tunnel_spid clk_tunnel_spid_2	clocks to tunnel block and interface	108MHz up to 225MHz [1]	CLK_TUNNEL_SPID_CTL register
Tunnel	clk_tunnel_v2 clk_tunnel_v2_|2	clocks to tunnel block and interface	108MHz up to 192MHz [1]	CLK_TUNNEL_V2_CTL register
VMPG	clk_vmpg	video MPEG decoder clock	86MHz, 133MHz [1] up to 157MHz	CLK_VMPG_CTL register
I^{2}C	clk_iic	I^{2}C module clock	24,48MHz [1]	CLK_IIC_CTL register
MBS	clk_mbs	MBS module clock	66MHz, 144MHz [1] to 192MHz	CLK_MBS_CTL register
VIP	clk_vip1	VIP1 module clock	low freq...81MHz	CLK_VIP1_CTL register
VO1	clk_vo1_ out	VO1 output clock from DDS0	Up to 100MHz Typical: 27MHz, 54MHz, 81MHz	DDS0_VO1_CTL register
	clk_vo1_pix	VO1 pixel clock	Up to 50MHz	CLK_VO1_CTL: Can be programmed to 1/2, 1/3, 1/4 or 1/6 of the clk_vo1_out
	clk_vo1_ proc	VO1 processing/ layer clock		CLK_VO1_PROC_CTL register
VO2	clk_vo2 _out	VO2 output clock from DDS1	Up to 100MHz Typical: 27MHz, 54MHz, 81MHz	DDS1_VO2_CTL register
	clk_vo2_pix	VO2 pixel clock	Up to 50MHz	CLK_VO2_CTL: Can be programmed to 1/2, 1/3, 1/4 or 1/6 of the clk_vo2_out
	clk_vo2 _proc	VO2 processing/ layer clock		CLK_VO2_PROC_CTL register

Table 249: System level and internal clocks *...continued*

Bus or Module	Signal Name	Description	Available Frequencies	Clock Module Control
LVDS	clk_lvds	LVDS data stream clock	35 to 86 MHz	CLK_LVDS_CTL register
	clk_lvdsx7	LVDS transmit clock	245 to 595 Mhz	PLL_LVDS_CTL and CLK_LVDSX7_CTL registers
TSU	clk_tstamp	Timestamp clock	13.5 to 108MHz	CLK_TSTAMP_CTL

[1] Nominal frequency.

Sources of the PNX2015 clocks: All clocks in the PNX2015 clock system are derived from one of six sources:

- Standard PLL for LVDS transmit clock
- Low jitter PLL for DDR-RAM control
- High-frequency dividers from the 1.7 GHz PLL in the CAB
- Derived clocks generated in clock block
- External clock inputs
- Direct Digital Synthesizer (DDS) blocks in the CAB

Clock assignments for each of these sources are summarized in Table 250 to Table 253.

Table 250: LVDS, DDR-RAM PLL transmit clock assignments

Source	Clocks	Notes
PLL LVDS	clk_lvdsx7	Seven times multiplying PLL for clk_lvds
PLL DDR	clk_mem	Low jitter PLL for DDR

Table 251: Clocks generated by HF dividers in CAB

Clock Name	Divider Value	Actual Value
clk_192	9	192 MHz
clk_173	10	172.8 MHz
clk_157	11	157.1 MHz
clk_144	12	144 MHz
clk_133	13	132.9 MHz
clk_123	14	123.4 MHz
clk_116	15	115.2 MHz
clk_108	16	108 MHz
clk_102	17	101.6 MHz

Table 252: Derived clocks generated in clocks block

Clock Name	Based on CAB Clock	Divider Value	Actual Value
clk_96	clk_192	2	96 MHz
clk_86	clk_173	2	86.4 MHz
clk_79	clk_157	2	78.5 MHz
clk_72	clk_144	2	72 MHz
clk_66	clk_133	2	66.5 MHz

Table 252: Derived clocks generated in clocks block *...continued*

Clock Name	Based on CAB Clock	Divider Value	Actual Value
clk_62	clk_123	2	61.7 MHz
clk_58	clk_116	2	57.6 MHz
clk_54	clk_108	2	54 MHz
clk_48	clk_192	4	48 MHz
clk_39	clk_157	4	39.3 MHz
clk_33	clk_133	4	33.2 MHz
clk_24	clk_192	8	24 MHz
clk_17	clk_133	8	16.6 MHz
clk_13_5	clk_27	2	13.5 MHz
clk_1_7	clk_27	16	1.69 MHz

Table 253: External clock inputs

Signal Name	Frequency	Used for Clock
XTAL_CLK	27MHz	Reference clock for PLLs, 13.5 MHz, etc.
DV5_DATA_8 to DV5_DATA_0	as needed	Used for test and debug purposes
DV4_CLK [1]	up to 81MHz (video)	clk_vip1
DV5_CLK [1]	up to 81MHz (video)	clk_vip1
CLK_LVDS_IN	up to 85MHz	clk_lvds
vip_stream_clk [1]	up to 81 MHz	clk_vo1_out, clk_vo2_out

[1] Separate for this module, but all come from DV_CLK in PNX2015. See Table 263 for actual chip I/O.

Table 254: DDS blocks in the CAB

DDS[1:0]_CTL	Byte
DDS[1]	High order
DDS[0]	Low order

LVDS DDR-RAM PLL requirements: Figure 171 shows the LVDS PLL structure.

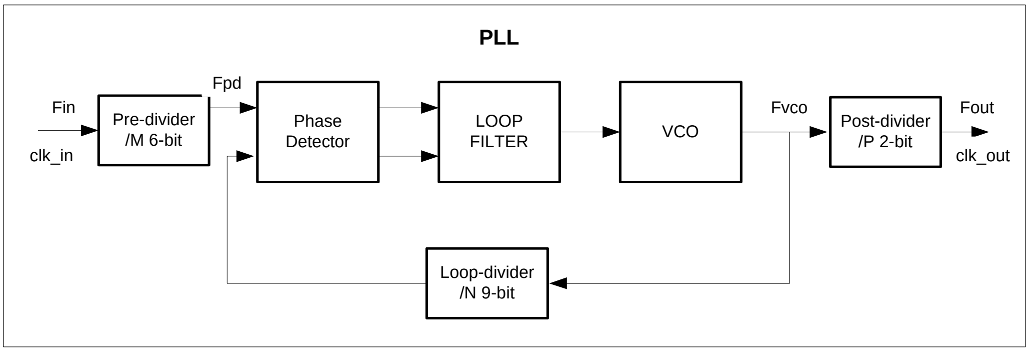


Fig 171.CAB PLL block diagram

DDR P is fixed to divide by two.

Remark: In Figure 171 a 600 MHz PLLs is used for PLL LVDS. The frequency for this PLL is given by:

Frequency = clk_in *N/(M*(2^P))

For normal operation the LVDS PLL is set to multiply by seven (default) clk_out/clk_in.

Where N is a 9-bit vector, M is a 6-bit vector and P is a 2-bit vector. The N, M and P vectors are programmable register bits in the Clock module control register for the LVDS. The PLL also contains a 4-bit charge-pump current adjustment for less jitter.

Note that using a value of 0 for either M or N may lead to undesirable behavior. For that reason, setting either M or N to 0 results in a value of 1 being used for both M and N. Assuming the P value is set to 0, this results in a PLL output frequency equal to the input frequency.

P represents a two-bit divider that can divide by 1, 2, 4, or 8. The setting for the post divider P is shown in Table 255.

Table 255: Post divider values

“P” control pin settings	Post divider
00	1
01	2
10	4
11	8

Table 256 contains several typical PLL programming values, using 27 MHz input clock.

Table 256: Programmable output frequency example

M	Fpd	(N)	Fvco MHz	2^P	Fout MHz	Reg_P
5	5.4	74	399.6	2	199.9	1
9	3	111	333	2	166.5	1
9	3	110	330	2	165	1
9	3	100	300	2	150	1
5	5.4	53	286.2	2	143.1	1
9	3	88	264	2	132	1
9	3	89	267	2	133.5	1
9	3	83	249	2	124.5	1
9	3	84	252	2	126	1
9	3	80	240	2	120	1
5	5.4	37	199.8	2	99.9	1
7	3.857	43	165.9	2	82.93	1
3	9	32	288	4 [1]	72	2
9	3	56	168	4 [1]	42	2

[1] Not possible with DDR PLL

DDR PLL charge pump settings: General recommendations:

- low M values
- run oscillator close to but less than 600 MHz
- 30 ≤ N ≤ 180 but track N with I_{ch} as in table below

Pre-Divider, M:

For minimum PLL input frequency of 2MHz at Phase Detector input

M≤ 27/2 = 13.5, In practice M≤ 14

To keep the input clock frequency to the phase detector as high as possible chose small M values.

Loop-Multiplier, N:

Calculations show that for optimum tracking N/I_{ch} = 8.2. A table of preferred I_{ch} values against N follows.

Table 257: Preferred I_{ch} values

N range	Ich mA	I-bits (3210)	Comments
25-29	4	1111	not recommended
30-37	4	1111	-
38-46	5.067	1110	-
47-54	6.133	1101	-
55-63	7.2	1100	-
64-72	8.267	1011	-
73-82	9.333	1010	-
83-89	10.4	1001	-
90-97	11.467	1000	-
98-107	12.53	0111	-
108-116	13.6	0110	-
117-125	14.66	0101	-
126-133	15.73	0100	-
134-142	16.8	0011	-
143-151	17.86	0010	-
152-160	18.93	0001	-
161-180	20	0000	-
181-200	20	0000	not recommended

If the oscillator is run close to its maximum frequency of 600MHz it will be less susceptible to jitter caused by other on-chip activity. But maintain 100MHz ≤ f_{osc} ≤ 600MHz.

The output divider, for the DDR PLL, has a hardwired divide by two post divider, to produce a 50% duty cycle. This divider limits the output frequency to 300MHZ.

Clock control and selection logic: All clocks generated in the clocks block use similar logic to select the input clock without producing glitches in the output clock. The clock control logic for each clock typically consists of an input mux, a “testing” clock mux, then output drivers and inverters. When the control register for a clock is written, the logic automatically blocks the output clock to a low level, selects the new clock, then releases the output clock to operate at the new frequency.

Figure 172 to Figure 175 are block diagrams of the logic used to produce specific clocks. In general:

- the ‘clk_observe’ output is used by test logic and is of no interest to the user.
- the clock select input muxes are controlled by the output clock control register.
- the ‘BLOCKER’ logic holds the output clock in a low state while the clock is being reprogrammed.

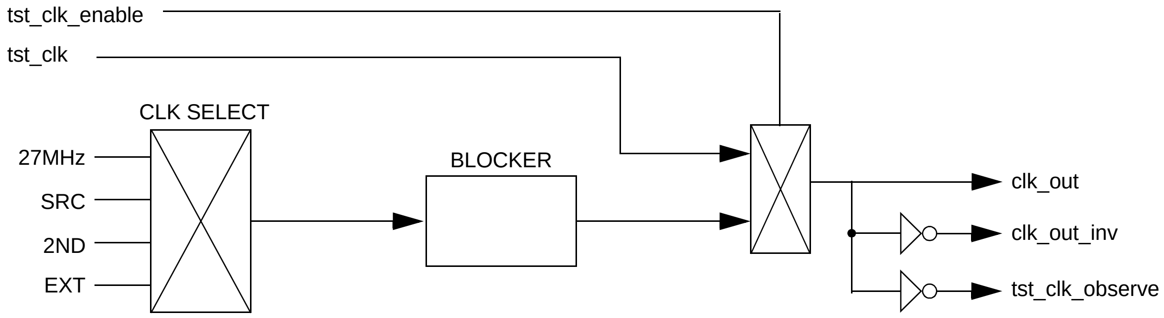


Fig 172.DVI_CLOCKS_LOGIC

Table 258: DVI_CLOCKS_LOGIC clock inputs

clk_out	Sources		
	SRC	2ND	EXT
clk_mem	PLL DDR	1.7 MHz	DV5_DATA_0
clk_lvds	clk_lvds_in	vo_out1	DV5_DATA_1
clk_lvdsx7	PLL LVDS	1.7MHz	DV5_DATA_0
clk_vo1_pix	DDS0/N	DDS0/N	DV5_DATA_7
clk_vo2_pix	DDS1/N	DDS1/N	DV5_DATA_8
clk_tstamp	108 MHz	13.5 MHz	27 MHZ

The logic shown in Figure 172 generates the clocks listed in Table 258. For these clocks, one of four input clocks can be selected by programming the appropriate control register. Although not listed in the table, 27 MHz is one possible source option for each of these clocks.

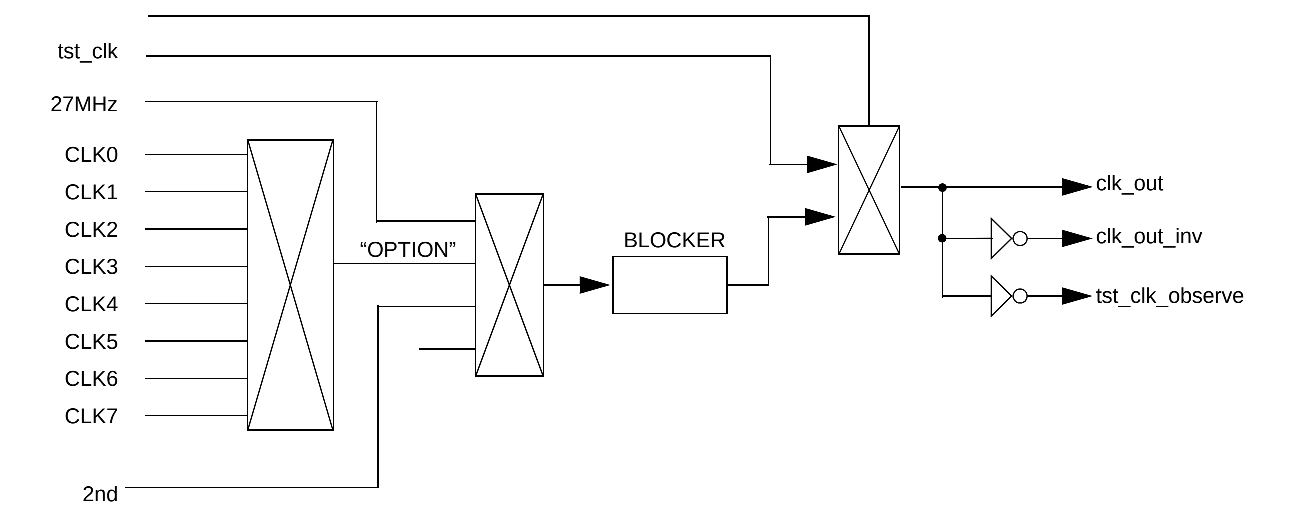


Fig 173.DVI_CLOCKS_8CLK

Table 259: DVI_CLOCKS_8CLK clock inputs

clk_out	Sources									
	2ND	0	1	2	3	4	5	6	7	EXT
clk_dcs	option	192MHz	173MHz	157MHz	144MHz	123MHz	108MHz	86MHz	77MHz	DV5_DATA_0
clk_dtl	option	157MHz	144MHz	133MHz	123MHz	115MHz	108MHz	96MHz	66MHz	DV5_DATA_0
clk_iic	13.5Mhz	157MHz	144MHz	133MHz	123MHz	115MHz	108MHz	96MHz	66MHz	54MHz
clk_mbs	vo1_out	192MHz	173MHz	157MHz	144MHz	123MHz	108MHz	96MHz	66MHz	DV5_DATA_2
clk_vip1	option	27MHz	dv4_clk	dv5_clk	vo_out1	vo_out2	27MHz	27MHz	27MHz	DV5_DATA_4
clk_vmpg	vo1_out	157MHz	144MHz	133MHz	123MHz	115MHz	102MHz	96MHz	86MHz	DV5_DATA_3
clk_vo1_out	option	DDS0	vip_st	resrvd	resrvd	27MHz	27MHz	27MHz	27MHz	DV5_DATA_7
clk_vo2_out	option	DDS1	vip_st	resrvd	resrvd	27MHz	27MHz	27MHz	27MHz	DV5_DATA_8
clk_vo1_proc	PLL LVDS	108MHz	96MHz	86MHz	78MHz	58MHz	39MHz	33MHz	17MHz	DV5_DATA_5
clk_vo2_proc	PLL LVDS	108MHz	96MHz	86MHz	78MHz	58MHz	39MHz	33MHz	17MHz	DV5_DATA_6

The logic shown in Figure 173 generates the clocks listed in Table 259. In addition to 27 MHz, these clocks allow 9 other clock sources to be selected. In general, the "EXT" option is only used for testing or special applications. Although not listed in the table, 27 MHz is one possible source option for each of these clocks.

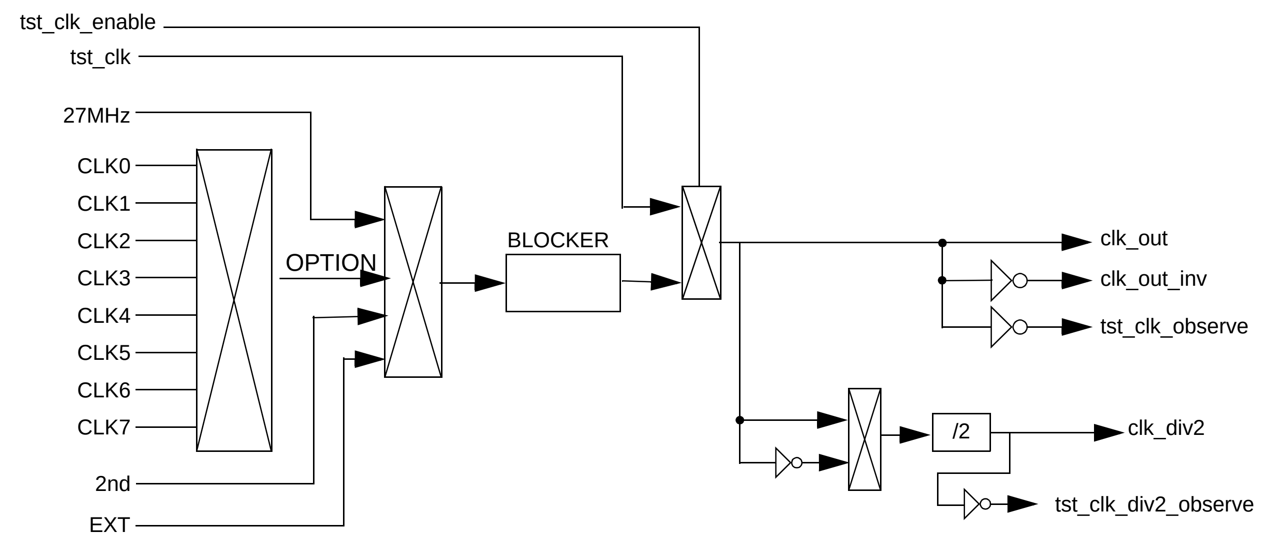


Fig 174.DVI_CLOCK_VMSP

Table 260: DVI_CLOCK_VMSP clock inputs

clk_out	**Sources**									
clk_div2	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**2ND**	**EXT**
clk_tunnel_spid clk_tunnel_spid_2	192MHz	173MHz	157MHz	144MHz	133MHz	123MHz	115MHz	108MHz	PLL DDR	vo2_out
clk_tunnel_v2 clk_tunnel_v2_2	192MHz	173MHz	157MHz	144MHz	133MHz	123MHz	115MHz	108MHz	PLL DDR	vo2_out

The logic shown in Figure 174 generates the clocks listed in Table 260. For these clocks, one of 10 clock sources can be selected to drive the output clock. These clock blocks also produce a “divide by 2” clock output. This “clk_div2” is routed to the clock shown in italics in the table. Although not listed in the table, 27 MHz is one possible source option for each of these clocks.

Power-up and reset sequence: On power-up, the Clock Module outputs all clocks at a default 27 MHz. Thus reset of modules always run on a 27 MHz clock.

Powerdown: All clocks generated in the Clock module may be disabled by programming the relevant clock enable bit of the appropriate clock control register.

Clock detection: Clock detection is required in the case of an external clock being removed or connected e.g if the video cable to a set top box is suddenly removed and an external video clock thereby stopped. This event is detected by the Clock module. Also the Clock module detects when the cable is re-connected and a clock is present again.

These events are flagged by an interrupt.

The clock detection is done on the following clocks inputs to the PNX2015 :

- DV_CLK
- RGB_CLK_IN

Clock detection is done based on a 5-bit counter running at the crystal clock frequency. The implementation detects clocks between 1 MHz and 200 MHz. It takes up to 2.5us from when the clock is removed until the interrupt condition is generated.

Remark: dv4_clk and dv5_clk are treated as separate clocks entering the clock module however they are both sourced from the same device pin DV_CLK.

See Table 263 for details of Chip I/O.

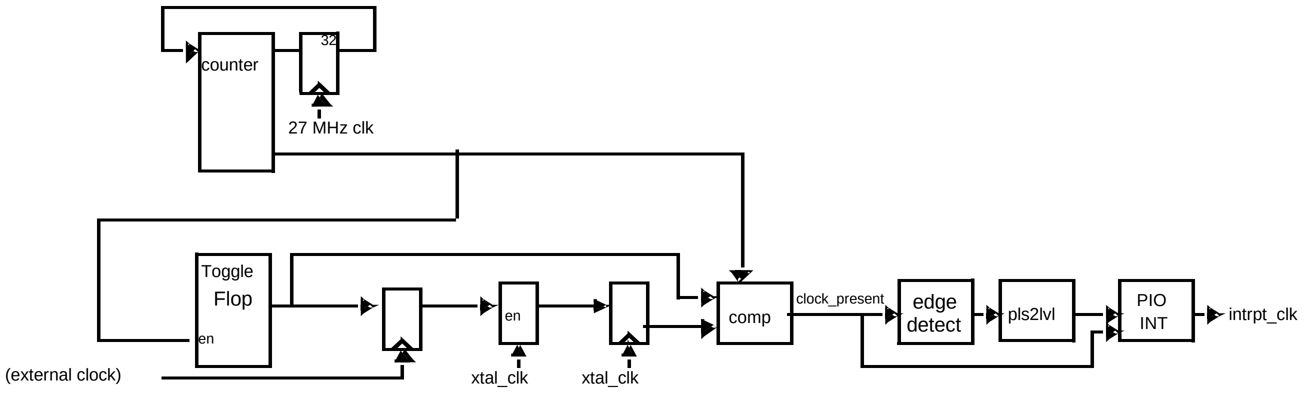


Fig 175.Clock detection circuitry

Interrupt generation: Interrupts are generated whenever 'clock present' changes status. Therefore an interrupt is generated if a clock changes from “present” to “non-present” OR from 'non-present' to “present.” It takes up to 2.5us from when the clock is removed until the interrupt condition is generated.

The interrupt registers are implemented using the standard peripheral interrupt module and can thus be enabled/cleared/set by software.

5.3.3 Register descriptions

5.3.3.1 Register summary

Table 261: Register summary

Address	Name	Description
0xC	PLL_LVDS_CTL	LVDS PLL Control Register
0x14	PLL_DDR_CTL	DDR PLL Control Register
0x18	PLL1_7_CTL	PLL_1.7GHz Control Register
0x1C	DDS0_VO1_CTL	DDS0:clk_vo1_mux frequency control
0x20	DDS1_VO2_CTL	DDS1:clk_vo2_mux frequency control
0x44	POWER_DOWN_CTL	CAB divider powerdown control
0x100	CLK_MEM_CTL	clk_mem_control
0x104	DFT_FREQ_CTR_CTL	DFT frequency counter control
0x20C	CLK_DCS_CTL	DCS bus clock control
0x210	CLK_DTL_CTL	DTL bus clock control
0x214	CLK_TUNNEL_V2_CTL	Tunnel clock control

Table 261: Register summary *…continued*

Address	Name	Description
0x21C	CLK_TUNNEL_SPID_CTL	Tunnel clock control
0x400	CLK_VMPG_CTL	clk_vmpg control
0x500	CLK_MBS_CTL	clk_mbs control
0x600	CLK_VIP1_CTL	clk_vip1 control
0x704	CLK_LVDSX7_CTL	clk_lvdsx7 control
0x708	CLK_LVDS_CTL	clk_lvds control
0x70C	CLK_IIC_CTL	clk_iic control
0xA00	CLK_VO1_OUT_CTL	clk_vo1_out control
0xA04	CLK_VO1_PIX_CTL	clk_vo1_pix control
0xA08	CLK_VO2_OUT_CTL	clk_vo2_out_control
0xA0C	CLK_VO2_PIX_CTL	clk_vo2_pix_control
0xA18	CLK_VO1_PROC_CTL	clk_vo1_proc control
0xA1C	CLK_VO2_PROC_CTL	clk_vo2_proc control
0xB30	CLK_TSTAMP_CTL	clk_tstamp control
0xFE0	INTERRUPT_STATUS	Status of Clock Detection interrupts
0xFE4	INTERRUPT_ENABLE	Enable Clock Detection interrupts
0xFE8	INTERRUPT_CLEAR	Clear clock detection interrupts
0xFEC	INTERRUPT_SET	Set clock detection interrupts
0xFFC	CLOCKS_MODULE_ID	Module identification and revision information

5.3.3.2 Register table

Table 262: HDCLOCKS registers

Bit	Symbol	Access	Reset value	Description
Offset 0xC - PLL_LVDS_CTL				
31	Pll_lvds_blocked	R	0xX	1: Blocking of clock from PLL is in progress
30	Pll_lvds_lock	R	0xX	1: LVDS PLL is phase locked
29:26	Current_adj_lvds[3:0]	R/W	0x0	4-bit current adjust
25	RSD_Bit25	R/W	0xX	reserved, read as 0, write nothing
24:16	Pll_lvds_n[8:0]	R/W	0x023	9-bit N parameter to LVDS PLL
15:14	RSD[15:14]	R/W	0xX	reserved, read as 0, write nothing
13:8	Pll_lvds_m[5:0]	R/W	0x05	6-bit M parameter to LVDS PLL
7:4	RSD[7:4]	R/W	0xX	reserved, read as 0, write nothing
3:2	Pll_lvds_p[1:0]	R/W	0x0	2-bit P parameter to LVDS PLL
1	Pll_lvds_pd	R/W	0x0	1: powerdown LVDS PLL also forces clk_lvdsx7 to 1.7MHz
0	RSD_Bit0	R	0x0	reserved, read as 0, write nothing
Offset 0x14 - PLL_DDR_CTL				
31	Pll_ddr_blocked	R	0xX	1: Blocking of clock from PLL is in progress
30	Pll_ddr_lock	R	0xX	1: PLL is phase locked

Table 262: HDCLOCKS registers *...continued*

Bit	Symbol	Access	Reset value	Description
29:26	Current_adj_ddr[3:0]	R/W	0x0	4-bit current adjust
25	RSD_Bit25	R/W	0xX	reserved, read as 0, write nothing
24:16	Pll_ddr_n[8:0]	R/W	0x02C	9-bit N parameter to PLL_DDR
15:14	RSD[15:14]	R/W	0xX	reserved, read as 0, write nothing
13:8	Pll_ddr_m[5:0]	R/W	0x09	6-bit M parameter to PLL_DDR
7:4	RSD[7:4]	R/W	0xX	reserved, read as 0, write nothing
3:2	RSD[3:2]	R/W	0xX	reserved, ignore read value, write nothing
1	Pll_ddr_pd	R/W	0x0	1: powerdown DDR PLL also forces clk_DDR_PLL to 1.7MHz
0	RSD_Bit0	R	0x0	reserved, read as 0, write nothing
Offset 0x18 - PLL1_7_CTL				
31:3	RSD[31:3]	R/W	0xXXXXXXXX	reserved, read as 0, write nothing
2	Pll1_7ghz_pd	R/W	0x0	1: powerdown PLL1_7GHZ
1:0	RSD[1:0]	R/W	0xX	reserved, read as 0, write nothing
Offset 0x1C - DDS0_VO1_CTL				

31	Dds0_vo1_test_select	R/W	0x0	1 = Input clock to DDS is from @clock_dds_test';
				0: Input clock to DDS is from 1.7GHz PLL;

30:0	Dds0_vo1_ctl[30:0]	R/W	0x04000000	DDS input N control value
Offset 0x20 - DDS1_VO2_CTL				

31	Dds1_vo2_test_select	R/W	0x0	1 = Input clock to DDS is from @clock_dds_test';
				0: Input clock to DDS is from 1.7GHz PLL;

30:0	Dds1_vo2_ctl[31:0]	R/W	0x04000000	DDS input N control value
Offset 0x44 - POWER_DOWN_CTL				
31:11	RSD[31:11]	R/W	0xXXXXXX	reserved, read as 0, write nothing
10	Pd_27	R/W	0x1	Power down 27MHz clock to pad 0: Normal mode 1: Powerdown
9	Pd_13_5	R/W	0x1	Power down 13.5MHz clock to pad 0: Normal mode 1: Powerdown
8	Pd_192	R/W	0x0	Power down 192MHz divider in the CAB block 0: Normal mode 1: Powerdown

Table 262: HDCLOCKS registers *...continued*

Bit	Symbol	Access	Reset value	Description
7	Pd_173	R/W	0x0	Power down 173MHz divider in the CAB block 0: Normal mode 1: Powerdown
6	Pd_157	R/W	0x0	Power down 157MHz divider in the CAB block 0: Normal mode 1: Powerdown
5	Pd_144	R/W	0x0	Power down 144MHz divider in the CAB block 0: Normal mode 1: Powerdown
4	Pd_133	R/W	0x0	Power down 133MHz divider in the CAB block 0: Normal mode 1: Powerdown
3	Pd_123	R/W	0x0	Power down 123MHz divider in the CAB block 0: Normal mode 1: Powerdown
2	Pd_115	R/W	0x0	Power down 115MHz divider in the CAB block 0: Normal mode 1: Powerdown
1	Pd_108	R/W	0x0	Power down 108MHz divider in the CAB block 0: Normal mode 1: Powerdown
0	Pd_102	R/W	0x0	Power down 102MHz divider in the CAB block 0: Normal mode 1: Powerdown
Offset 0x100 - CLK_MEM_CTL				
31:3	RSD[31:3]	R/W	0xXXXXXXXX	reserved, read as 0, write nothing
2:1	Sel_clk_mem[1:0]	R/W	0x0	00: clk_mem =27MHz xtal_clk 01: clk_mem = functional clock (PLL DDR) 10: clk_mem = 1.7MHz 11: clk_mem = backup[3]
0	En_clk_mem	R/W	0x1	1: enable clk_mem
Offset 0x104 - DFT_FREQ_CTR_CTL				
31:16	Count[15:0]	R	0x0000	result of a frequency count operation
15:11	RSD[15:11]	R/W	0x00	reserved, read as 0, write nothing
10:7	Clk_sel[3:0]	R/W	0x0	selects 8 clocks for output to the 8 dft_clk_pins. Dft_clk[0] is used as input to the frequency counter.

Table 262: HDCLOCKS registers *…continued*

Bit	Symbol	Access	Reset value	Description
6:4	Clk_rotate[2:0]	R/W	0x0	Rotate control for the 8 dft_clk signals. This allows any of the selected output clocks to be rotated into dft_clk[0] for the frequency counter to use.
3:2	RSD[3:2]	R/W	0x0	read as 0
1	Done	R	0x1	This bit is set to cleared by when the start bit is set and returns to a value of 1 when the frequency counter finishes a 10usec acquisition.
0	Start	R/W	0x0	This is a self clearing bit set to 1 to start frequency counter
Offset 0x20C - CLK_DCS_CTL				
31:7	RSD[31:7]	R/W	0xXXXXXXX	reserved, read as 0, write nothing
6:4	Sel_clk_dcs_src[2:0]	R/W	0x0	000: clk_dcs = clk_192 001: clk_dcs = clk_173 010: clk_dcs = clk_157 011: clk_dcs = clk_144 100: clk_dcs = clk_123 101: clk_dcs = clk_108 110: clk_dcs = clk_86 111: clk_dcs = clk_72
3	RSD_Bit3	R/W	0x0	reserved
2:1	Sel_clk_dcs[1:0]	R/W	0x0	00: clk_dcs = 27MHz xtal_clk 01: clk_dcs = functional clock select from bits [6:4] 10: clk_dcs = NOT functional clock select from bits [6:4] 11: clk_dcs = backup[3]
0	RSD_Bit0	R/W	0x0	Reserved
Offset 0x210 - CLK_DTL_CTL				
31:7	RSD[31:7]	R/W	0xXXXXXXX	reserved, read as 0, write nothing

Table 262: HDCLOCKS registers *...continued*

Bit	Symbol	Access	Reset value	Description
6:4	Sel_clk_dtl_src[2:0]	R/W	0x0	000: clk_dtl = clk_157 001: clk_dtl = clk_144 010: clk_dtl = clk_133 011: clk_dtl = clk_123 100: clk_dtl = clk_115 101: clk_dtl = clk_108 110: clk_dtl = clk_96 111: clk_dtl = clk_66
3	RSD_Bit3	R/W	0x0	reserved
2:1	Sel_clk_dtl[1:0]	R/W	0x0	00: clk_dtl = 27NHz xtal_clk 01: clk_dtl = functional clock select from bits [6:4] 10: clk_dtl = NOT functional clock select from bits [6:4] 11:clk_dtl = backup[3]
0	RSD_Bit0	R/W	0x0	Reserved
Offset 0x214 - CLK_TUNNEL_V2_CTL				
31:7	RSD[31:7]	R/W	0xXXXXXXX	reserved, read as 0, write nothing
6:4	Sel_clk_tunnel_src[2:0]	R/W	0x0	000: clk_v2_tunnel = clk_192 001: clk_v2_tunnel = clk_173 010: clk_v2_tunnel = clk_157 011: clk_v2_tunnel = clk_144 100: clk_v2_tunnel = clk_133 101: clk_v2_tunnel = clk_123 110: clk_v2_tunnel = clk_115 111: clk_v2_tunnel = clk_108
3	RSD_Bit3	R/W	0x0	Reserved

Table 262: HDCLOCKS registers *…continued*

Bit	Symbol	Access	Reset value	Description
2:1	Sel_clk_tunnel[1:0]	R/W	0x0	00: clk_v2_tunnel = 27MHz xtal_clk 01: clk_v2_tunnel = functional clock select from bits [6:4] 10: clk_v2_tunnel = DDR PLL 11: clk_v2_tunnel = vo2_out
0	En_clk_tunnel	R/W	0x1	1: enable clk_v2_tunnel
Offset 0x21C - CLK_TUNNEL_SPID_CTL				
31:7	RSD[31:7]	R/W	0xXXXXXXX	reserved, read as 0, write nothing
6:4	Sel_clk_spid_tunnel_src[2:0]	R/W	0x0	000: clk_spid_tunnel = clk_192 001: clk_spid_tunnel = clk_173 010: clk_spid_tunnel = clk_157 011: clk_spid_tunnel = clk_144 100: clk_spid_tunnel = clk_133 101: clk_spid_tunnel = clk_123 110: clk_spid_tunnel = clk_115 111: clk_spid_tunnel = clk_108
3	RSD_Bit3	R/W	0x0	Reserved
2:1	Sel_clk_spid_tunnel[1:0]	R/W	0x0	00: clk_spid_tunnel = 27MHz xtal_clk 01: clk_spid_tunnel = functional clock select from bits [6:4] 10: clk_spid_tunnel = DDR PLL 11: clk_spid_tunnel = vo2_out
0	En_clk_spid_tunnel	R/W	0x1	1:enable clk_spid_tunnel
Offset 0x400 - CLK_VMPG_CTL				
31:7	RSD[31:7]	R/W	0xXXXXXXX	reserved, read as 0, write nothing

Table 262: HDCLOCKS registers *...continued*

Bit	Symbol	Access	Reset value	Description
6:4	Sel_vmpg_clk_src[2:0]	R/W	0x7	000: clk_vmpg = clk_157 001: clk_vmpg = clk_144 010: clk_vmpg = clk_133 011: clk_vmpg = clk_123 100: clk_vmpg = clk_115 101: clk_vmpg = clk_102 110: clk_vmpg = clk_96 111: clk_vmpg = clk_86
3	RSD_Bit3	R/W	0xX	reserved, read as 0, write nothing
2:1	Sel_clk_vmpg[1:0]	R/W	0x0	00: clk_vmpg = 27MHz xtal_clk 01: clk_vmpg = functional clock select from bits [6:4] 10: clk_cmpg = vo1_out 11: clk_vmpg = backup[6]
0	En_clk_vmpg	R/W	0x1	1: enable clk_vmpg
Offset 0x500 - CLK_MBS_CTL				
31:7	RSD[31:7]	R/W	0xXXXXXXX	reserved, read as 0, write nothing
6:4	Sel_mbs_clk_src[2:0]	R/W	0x7	000: clk_mbs =clk_192 001: clk_mbs = clk_173 010: clk_mbs = clk_157 011: clk_mbs = clk_144 100: clk_mbs = clk_123 101: clk_mbs = clk_108 110: clk_mbs = clk_96 111: clk_mbs = clk_66
3	RSD_Bit3	R/W	0xX	reserved, read as 0, write nothing

Table 262: HDCLOCKS registers *…continued*

Bit	Symbol	Access	Reset value	Description
2:1	Sel_clk_mbs[1:0]	R/W	0x0	00: clk_mbs = 27MHz xtal_clk 01: clk_mbs = functional clock clock select from bits [6:4] 10: clk_mbs = backup[5] 11: clk_mbs = vo1_out
0	En_clk_mbs	R/W	0x1	1: enable clk_mbs
Offset 0x600 - CLK_VIP1_CTL				
31:6	RSD[31:6]	R/W	0xXXXXXXX	reserved, read as 0, write nothing
5:3	Sel_clk_vip1_src[2:0]	R/W	0x0	000: clk_vip1_src = 27MHz xtal_clk 001: clk_vip1_src = DV4_CLK 010: clk_vip1_src = DV5_CLK 011: clk_vip1_src = clk_vo1_out 100: clk_vip1_src = clk_vo2_out 101, to 111: reserved
2:1	Sel_clk_vip1[1:0]	R/W	0x0	00: clk_vip1 = 27MHz xtal_clk 01: clk_vip1 = functional clock select bits [5:3] 10: clk_vip1 = NOT functional clock select from bits [5:3] 11: clk_vip1 = backup[7]
0	En_clk_vip1	R/W	0x1	1: enable clk_vip1
Offset 0x704 - CLK_LVDSX7_CTL				
31:3	RSD[31:3]	R/W	0xXXXXXXXX	reserved, read as 0, write nothing
2:1	Sel_clk_lvdsx7[1:0]	R/W	0x0	00: clk_mem =27MHz xtal_clk 01: clk_mem = functional clock 10: clk_mem = 1.7MHz 11: clk_mem = backup[3]
0	En_clk_lvdsx7	R/W	0x1	1:enable clk_lvdsx7
Offset 0x708 - CLK_LVDS_CTL				
31:3	RSD[31:3]	R/W	0xXXXXXXXX	reserved, read as 0, write nothing

Table 262: HDCLOCKS registers *…continued*

Bit	Symbol	Access	Reset value	Description
2:1	Sel_clk_lvds[1:0]	R/W	0x0	00: clk_lvds = 27MHz xtal_clk 01: clk_lvds = functional clock 10: clk_lvds = clk_vo_out1 11: clk_lvds = backup[4]
0	En_clk_lvds	R/W	0x1	1: enable clk_lvds
Offset 0x70C - CLK_IIC_CTL				
31:7	RSD[31:7]	R/W	0xXXXXXXX	reserved, read as 0, write nothing
6:4	Sel_clk_iic_src[2:0]	R/W	0x7	000: clk_iic = clk_157 001: clk_iic = clk_144 010: clk_iic = clk_133 011: clk_iic = clk_123 100: clk_iic = clk_115 101: clk_iic = clk_108 110: clk_iic = clk_96 111: clk_iic = clk_66
3	RSD_Bit3	R/W	0xX	reserved
2:1	Sel_clk_iic[1:0]	R/W	0x0	00: clk_iic = 27MHz xtal_clk 01: clk_iic = functional clock select from bits [6:4] 10: clk_iic = 13.5MHz clock 11: clk_iic = 54MHz clock
0	En_clk_iic	R/W	0x0	Reserved
Offset 0xA00 - CLK_VO1_OUT_CTL				
31:6	RSD[31:6]	R/W	0xXXXXXXX	reserved, read as 0, write nothing

Table 262: HDCLOCKS registers *…continued*

Bit	Symbol	Access	Reset value	Description
5:3	sel_clk_vo1_out_src[2:0]	R/W	0x0	Functional clock 0: DDSO 1: Vip stream clock 10: Reserved 11: Reserved 100: Reserved 101: Reserved 110: Reserved 111: Reserved
2:1	Sel_clk_vo1_out[1:0]	R/W	0x0	00: clk_vo1_out = 27MHz xtal_clk 01: clk_vo1_out = functional clock select bits [5:3] 10: clk_vo1_out = NOT functional clock select bits [5:3] 11: clk_vo1_out = backup[10]
0	En_clk_vo1_out	R/W	0x1	1: enable clk_VO1_out
Offset 0xA04 - CLK_VO1_PIX_CTL				
31:6	RSD[31:6]	R/W	0xXXXXXXX	reserved, read as 0, write nothing
5:3	Div_clk_vo1_pix[2:0]	R/W	0x1	functional clock: 000: no divide 001: divide DDS0 by 2 010: divide DDS0 by 3 011: divide DDS0 by 4 100: divide DDS0 by 6 101: divide DDS0 by 8
2:1	Sel_clk_vo1_pix[1:0]	R/W	0x0	00: clk_vo1_pix = 27MHz xtal_clk 01: clk_vo1_pix = functional clock select from bits [5:3] 10: clk_vo1_pix = NOT functional clock select from bits[5:3] 11: clk_vo1_pix = backup[10]
0	En_clk_vo1_pix	R/W	0x1	1: enable clk_vo1_pix
Offset 0xA08 - CLK_VO2_OUT_CTL				
31:6	RSD[31:6]	R/W	0xXXXXXXX	reserved, read as 0, write nothing

Table 262: HDCLOCKS registers *…continued*

Bit	Symbol	Access	Reset value	Description
5:3	sel_clk_vo2_out_src[2:0]	R/W	0x0	Functional clock 0: DDS1 1: Vip stream clock 10: Reserved 11: Reserved 100: Reserved 101: Reserved 110: Reserved 111: Reserved
2:1	Sel_clk_vo2_out[1:0]	R/W	0x0	00: clk_vo2_out = 27MHz xtal_clk 01: clk_vo2_out = functional clock select from bits [5:3] 10: clk_vo2_out = NOT functional clock select from bits [5:3] 11: clk_vo2_out = backup[11]
0	En_clk_vo2_out	R/W	0x1	1: enable_clk_vo2_mux
Offset 0xA0C - CLK_VO2_PIX_CTL				
31:6	RSD[31:6]	R/W	0xXXXXXXX	reserved, read as 0, write nothing
5:3	Div_clk_vo2_pix[2:0]	R/W	0x1	functional clock: 000: no divide 001: divide DDS1 by 2 010: divide DDS1 by 3 011: divide DDS1 by 4 100: divide DDS1 by 6 101: divide DDS1 by 8
2:1	Sel_clk_vo2_pix[1:0]	R/W	0x0	00: clk_vo2_pix = 27MHz xtal_clk 01: clk_vo2_pix = functional clock select from bits[5:3] 10: clk_vo2_pix = NOT functional clock select from bits[5:3] 11: clk_vo2_pix = backup[11]
0	En_clk_vo2_pix	R/W	0x1	1: enable clk_vo2_pix
Offset 0xA18 - CLK_VO1_PROC_CTL				
31:6	RSD[31:6]	R/W	0xXXXXXXX	reserved, read as 0, write nothing

Table 262: HDCLOCKS registers *...continued*

Bit	Symbol	Access	Reset value	Description
5:3	Sel_vo1_proc_clk_src[2:0]	R/W	0x7	This is the source of functional clock to vo1_proc_clk 000: clk_108 001: clk_96 010: clk_86 011: clk_76 100: clk_58 101: clk_39 110: clk_33 111: clk_17
2:1	Sel_vo1_proc_clk[1:0]	R/W	0x0	00: clk_vo1_proc = 27MHz xtal_clk 01: clk_vo1_proc = functional clock select from bits [5:3] 10: clk_vo1_proc = PLL LVDS 11: clk_vo1_proc = backup[8]
0	En_clk_vo1_proc	R/W	0x1	1: enable clk_vo1_proc
Offset 0xA1C - CLK_VO2_PROC_CTL				
31:6	RSD[31:6]	R/W	0xXXXXXXX	reserved, read as 0, write nothing
5:3	Sel_vo2_proc_clk_src[2:0]	R/W	0x7	This is the source of functional clock to vo2_proc_clk 000: clk_108 001: clk_96 010: clk_86 011: clk_76 100: clk_58 101: clk_39 110: clk_33 111: clk_17

Table 262: HDCLOCKS registers *...continued*

Bit	Symbol	Access	Reset value	Description
2:1	Sel_vo2_proc_clk[1:0]	R/W	0x0	00: clk_vo2_proc = 27MHz xtal_clk 01: clk_vo2_proc = functional clock select from bits[5:3] 10: clk_vo2_proc = PLL LVDS 11: clk_vo2_proc = backup[9]
0	En_clk_vo2_proc	R/W	0x1	1: enable clk_vo2_proc
Offset 0xB30 - CLK_TSTAMP_CTL				
31:3	RSD[31:3]	R/W	0xXXXXXXXX	reserved, read as 0, write nothing
2:1	Sel_clk_tstamp[1:0]	R/W	0x0	00: clk_tstamp = 27MHz xtal_clk 01: clk_tstamp = 108MHz 10: clk_tstamp = 13.5MHz 11: clk_tstamp = 27MHz xtal_clk
0	En_clk_tstamp	R/W	0x1	1: enable clk_tstamp
Offset 0xFE0 - INTERRUPT_STATUS				
31	lvds_clk_present	R	0x0	0: Clock NOT present 1: clock present
30	dv5_clk_present	R	0x0	0: Clock NOT present 1: Clock present
29	dv4_clk_present	R	0x0	0: clock NOT present 1: Clock present
28:3	RSD[28:3]	R/W	0xXXXXXXX	reserved, read as 0, write nothing
2	Lvds_clk_int	R	0x0	1: clock int
1	Dv5_clk_int	R	0x0	1: clock int
0	Dv4_clk_int	R	0x0	1: clock int
Offset 0xFE4 - INTERRUPT_ENABLE				
31:3	RSD[31:3]	R/W	0xXXXXXXXX	reserved, read as 0, write nothing
2	Lvds_clk_int_enable	R/W	0x0	1: interrupt enabled 0: interrupt NOT enabled
1	Dv5_clk_int_enable	R/W	0x0	1: interrupt enabled 0: interrupt NOT enabled
0	Dv4_clk_int_enable	R/W	0x0	1: interrupt enabled 0: interrupt NOT enabled
Offset 0xFE8 - INTERRUPT_CLEAR				
31:3	RSD[31:3]	R/W	0xXXXXXXXX	reserved, read as 0, write nothing

Table 262: HDCLOCKS registers *…continued*

Bit	Symbol	Access	Reset value	Description
2	Clear_lvds_clk_int	R/W	0x0	1: clear interrupt, 0: no effect
1	Clear_dv5_clk_int	R/W	0x0	1: clear interrupt, 0: no effect
0	Clear_dv4_clk_int	R/W	0x0	1: clear interrupt, 0: no effect
Offset 0xFEC - INTERRUPT_SET				
31:3	RSD[31:3]	R/W	0xXXXXXXXX	reserved, read as 0, write nothing
2	Set_lvds_clk_int	R/W	0x0	1: set interrupt, 0: no effect
1	Set_dv5_clk_int	R/W	0x0	1: set interrupt, 0: no effect
0	Set_dv4_clk_int	R/W	0x0	1: set interrupt
Offset 0xFFC - CLOCKS_MODULE_ID				
31:0	HDClocks_Module_id[31:0]	R	0xA07E0000	Module ID: 0xA07E TBD Major_rev: 0x0 Minor_rev: 0x0 MMIO_SPACE: 0x00

5.3.4 Chip I/Os

The following chip I/Os are related directly to the Clock module:

Table 263: Chip I/Os

Module Port	Chip Port	Type	I/O Buffer	Description
xtal_clk	XTALI_SYS	OSC	apio3v3	27 MHz clock input from oscillator pad
clk_lvds_in	RGB_CLK_IN	LVDS	bpx2t14mchp	LVDS clock source
dv4_clk	DV_CLK	DVI	bpx2t14mchp	DV clock input for port 4
dv5_clk	DV_CLK	DVI	bpx2t14mchp	DV clock input for port 5 (same as for port 4 at device level)
clk_DV5_DATA_8 to DV5_DATA_0	DV5_DATA_8 to DV5_DATA_0	DVI	bpts3chp	Backup clock sources
vip_stream_clk	DV_CLK	DVI	bpx2t14mchp	Clock for VO output clock in VIP streaming mode

5.4 DCS-Network (DCSN)

5.4.1 DCS-Network Target ID returns

If the targets on the DCSN network generate DCS errors, then they will return identifier values to the DCS Control register as shown in Table 264.

Table 264: DCSN target return IDs

Target return ID	Target
0x40	Null Module
0x41	DCS Control
0x42	DCS Security
0x0	Global Register
0x1	LVDS
0x2	HD Clocks
0x3	GIC
0x4	Reset
0x5	Pman1_Monitor

Table 264: DCSN target return IDs

Target return ID	Target
0x6	Pman1_Security
0x7	Pman1_Arbiter
0x8	VO1
0x9	VO2
0xA	VIP
0xB	MBS
0xC	TSU
0xD	VMPG
0xE	Pman2_Monitor
0xF	Pman2_Security
0x10	Columbus_Monitor
0x11	DDR Controller
0x12	North_Tunnel config
0x13	South_Tunnel config
0x14	DMA Gate
0x15	North Tunnel
0x16	Default Slave

5.4.2 Network controller block

5.4.2.1 Features

The DCS Network Controller block includes the following features:

- Implements interconnect between the DCS Network Controller and 32-bit DTL-DCS adapters.
- Provides a configurable number of initiator and target DTL ports and a configurable number of initiator and target DCS ports (used for connecting to devices without DTL interfaces, such as a DCS-DCS bridge.)
- Creates OR-tree and synchronizer flip flops for target response signals and buffer tree for target broadcast signals.
- The resulting network includes all of the features of the DCS Network Controller and the DCS-DTL adapters.

5.4.2.2 Place in the system

Figure 176 shows where the DCS Network would be used in a system. The example shows a number of DTL Initiators and Targets, which communicate through a DCS Network, which consists of the DCS Network Controller, DTL2DCS Adapters, and DCS2DTL Adapters.

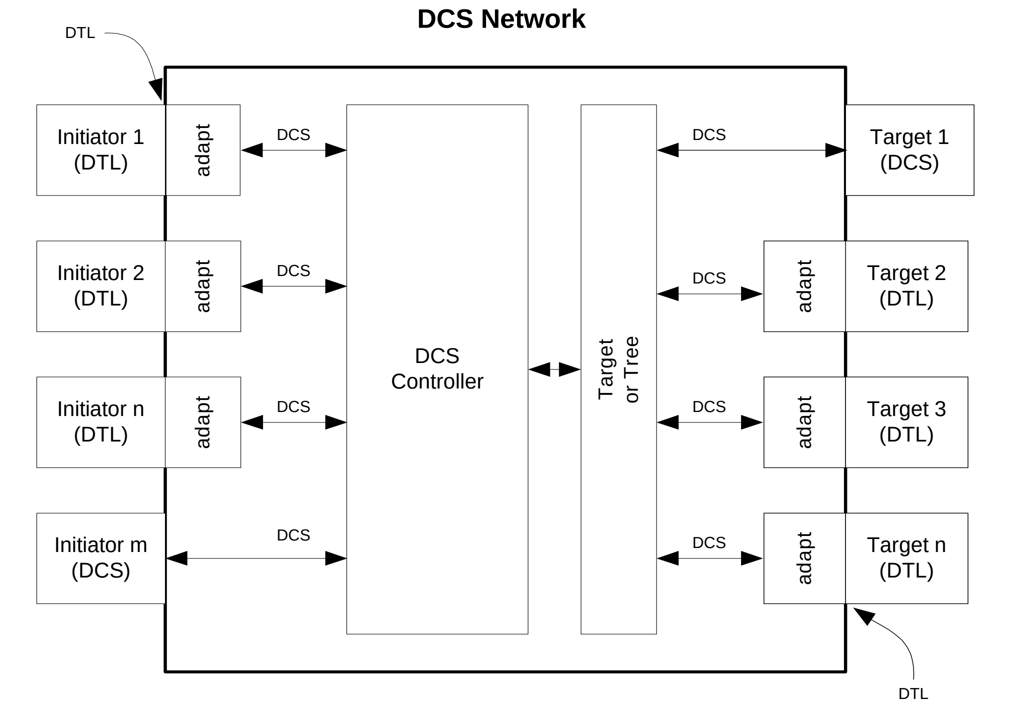


Fig 176.DCS network

5.4.2.3 Architecture

The DCS network builder creates dedicated DTL or DCS interfaces for each of the initiators and dedicated DTL or DCS interfaces for each of the targets. All of the synchronous adapters as well as the DCS controller operate on the dcs_clk clock input. Asynchronous adapters each have an independent clock input. No assumption is made about the frequency of phase relationship between the asynchronous clock inputs.

The targets communicate with the DCS controller through an OR-tree. For the asynchronous targets, a separate OR-tree (called the asynchronous OR-tree) combines signals before synchronizing to the dcs_clk. It is assumed that the asynchronous targets may be in different islands (chiplets) and therefore the DCS network builder allows the user to specify the arrangement of the asynchronous OR-tree.

The targets are specified as synchronous or asynchronous when the block is built by HDLi. Only those specified as asynchronous are connected to the asynchronous OR-tree (and in the case of DTL ports, have a separate clock signal). The initiators each include a configuration signal that determines whether that port is synchronous or asynchronous. All DTL initiator ports include a clock signal, but that clock must be equal in frequency and phase to the dcs_clk signal if the configuration pin indicates a synchronous initiator.

An overall block diagram for the network is shown in Figure 177.

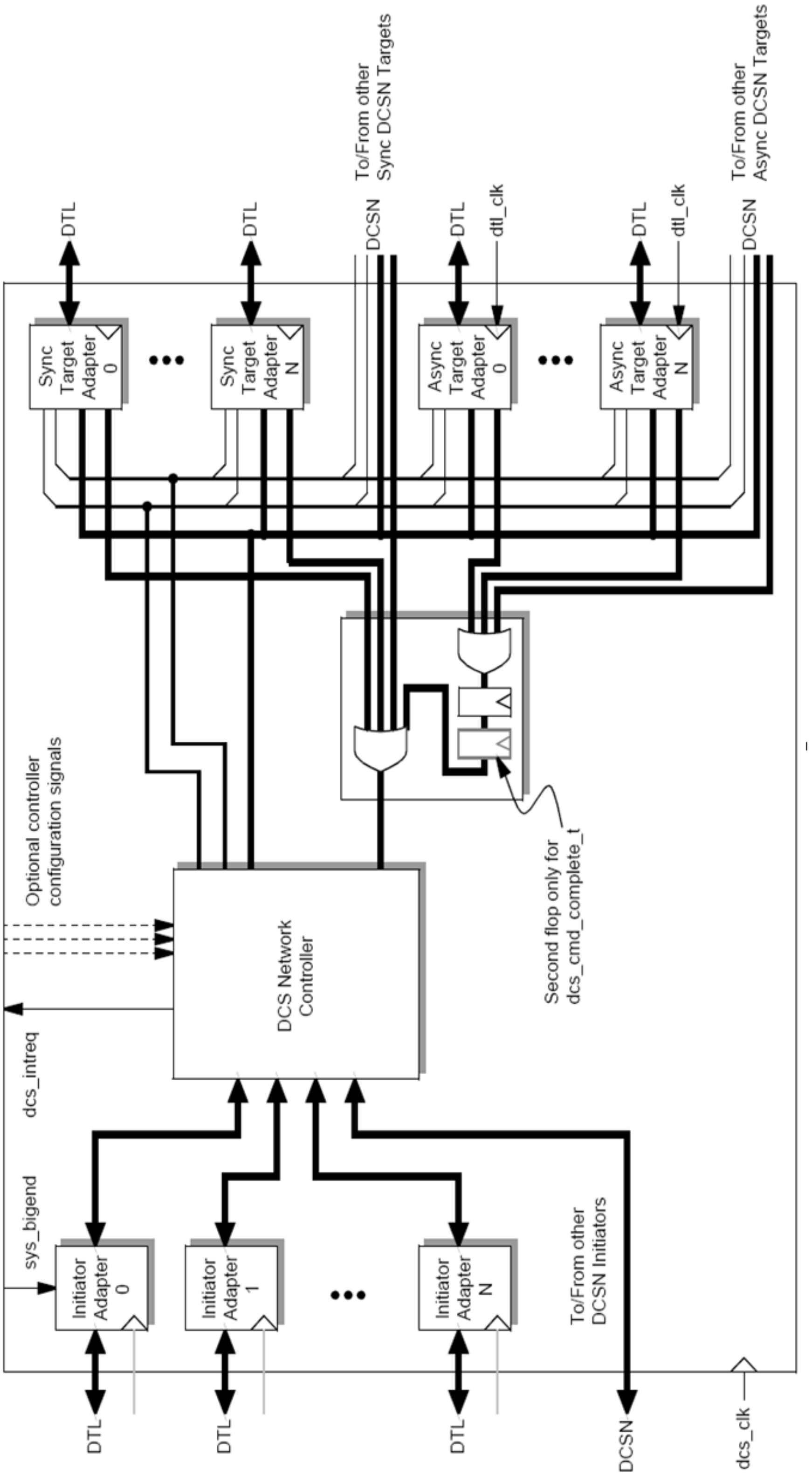


Fig 177.DCS network builder block diagram

5.4.2.4 Interrupt

An active high output signal named dcs_intreq is asserted when the DCS Controller requires attention and issues an interrupt.

5.4.2.5 Clocking

A single dcs_clk input is used as the clock for the central controller and the synchronizer flip flops in the OR-tree block. Each initiator and target adapter can be synchronous or asynchronous. Synchronous target adapters (connecting to DTL targets) are also clocked by dcs_clk, and it is expected that dcs_clk is used on the connected DTL target. Asynchronous target adapters are clocked by an individual dtl_at#_clk signal, which may be faster or slower than dcs_clk.

All initiator adapters (connected to DTL initiators) include a dtl_i#_clk signal, even if they are synchronous. However, if the dcs#_sync_i signal is high, indicating synchronous operation, then the dtl_i#_clk must be identical in frequency and phase to the dcs_clk signal.

5.4.2.6 Reset

A single dcs_rst_an signal is an active low, asynchronous reset input. This is connected to every adapter as well as the DCSN controller. This signal is synchronized before being used to force the D-input to flip flops low. The clock(s) must be running to complete the reset. It is expected that all connected DTL devices also receive this reset signal.

5.4.2.7 The OR-tree

The DCS Network Builder is primarily a convenient way to create all of the connections between DCS-DTL adapters and a DCS Controller. The only logic required is the target ORtree and a synchronizer for the asynchronous targets. The OR-tree is instantiated in a sub-block. Each separate OR function in the asynchronous OR-tree is also instantiated as a sub-block. This hierarchy has been selected to simplify netlist partitioning. The user has complete control over the organization of the asynchronous OR-tree.

5.4.3 DCS-Controller (DCSC)

This section specifies the DCSC that is available through the HDL Integrator. This network controller conforms to the DCS-Network (DCSN) specification Ref. 5. The IP is a generic design and is applicable to most designs where DCS-Network is used.

5.4.3.1 Features

The DCSC includes the following features:

- Supports 1 to 8 initiator interfaces
- Supports 2 to 64 target interfaces
- Programmable initiator ID for each DCSN initiator

  Fixed at compile time or dynamic depending upon the number of initiators in the system.
- Programmable Address Map

  Optionally, address map information may be input signals to the DCSN.

  Each target has a defined aperture of a given size, offset from a defined base address register. A given bus controller may be configured to have up to four different base address registers shared by various targets.

In addition, each target may have additional regions which are steered to it. This is used primarily by bridges.

- Programmable timeout generation
- Error and Timeout information captured

  Initiator ID of the currently granted DCSN initiator.

  32-bit address of the currently granted DCSN transaction.

  Encoded target number for the currently selected DCSN device.

  Additional command Information including cmd_mask and cmd_read.
- Interrupt generation for any non-masked timeouts and errors
- Optional Selective Blocking of initiators via R/W registers with programmable power on defaults
- Optional Blocking of errors on 32-bit reads
- Anti-glitch filter is included for all dcs*_cmd_sel_t and dcs*_cmd_req_i signals
- Conforms to signaling protocol of the DCS Specification Ref. 5.

5.4.3.2 Functional Description

The top level block diagram of the DCSC is shown in Figure 178.

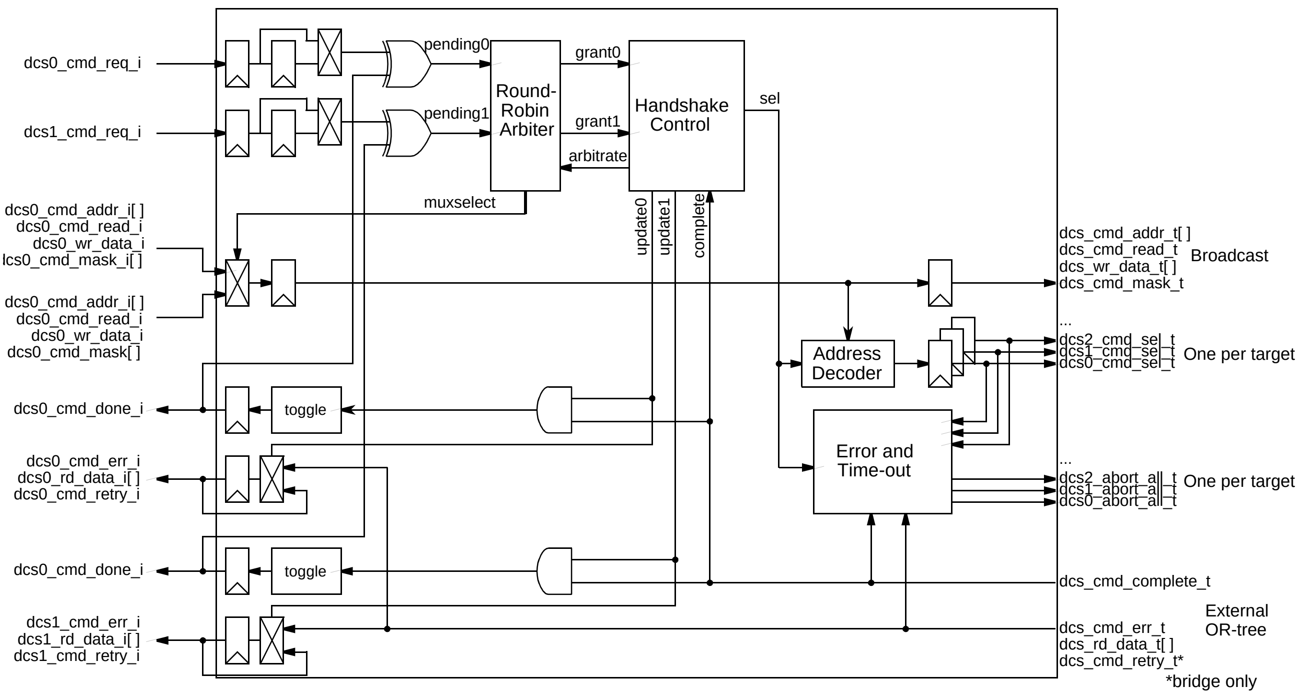


Fig 178. DCSC Block Diagram

Architecture: The bus controller has dedicated interfaces for each of the n initiators and a shared broadcast bus for each of the m targets. The most notable exception is that each target has a dedicated select and abort signal. All configuration information regarding address map, control settings, and error status is contained within the bus controller. This requires that registers inside the bus controller be written to setup the address map and other configuration information. This is done via any of the DCSC initiator interfaces.

For signals returned from the targets an OR-tree is implemented outside the bus controller, which combines all these signals together and synchronizes them to the bus controller clock.

Each of the initiator ports is independently configured as synchronous or asynchronous. When synchronous mode is selected, a synchronizer flip flop is bypassed. The DTL adapter connected to each port is configured to match the bus controller port.

Special features

- Initiator interfaces

  The DCSC template can be configured with 1 to 8 initiator interfaces, which share no common signals. Each interface can be declared synchronous or asynchronous by the state of the dcs#_sync_i inputs. These inputs are expected to be connected to tie-high or tie-low cells, and are therefore static.

  All other signals in the initiator interface come from or go to a DCSN initiator. All inputs contain only a minimum amount of logic and outputs on the initiator interface are output directly from flip flop as shown in Figure 178.
- Initiator IDs

  Each DCSN Initiator in a given system has an associated ID which is unique throughout the entire DCSN subsystem. When creating a bus controller for an DCSN segment the user has the option to choose between a static or dynamic initiator ID input for each initiator interface. Most initiators would only have a single ID, and therefore the static option is sufficient. For bridges and other special DCSN initiators, the dynamic option is chosen, thus creating another initiator signal on that interface. The provided signal is dcs#_initid_i[4:0].
- Target interfaces

  The number of target interfaces is configurable from 2 to 64. Each of these interfaces have dedicated select and abort_all outputs, while the command and write data information is broadcast to all targets. The response and read data signals are collected in an external OR and connected to a single bus controller input.

  All outputs are directly connected to a register. The inputs (command response and read data) go through minimal logic before reaching a flip flop. The OR-tree from synchronous targets, together with any logic inside the bus controller, can be traversed in a single clock cycle. The setup time for these inputs to the bus controller is as small as possible.
- Address mapping

  In order to accommodate many different addressing schemes the template allows the user a number of options for each DCSN target attached to the system. For each target in the user has the ability to specify address decoding using a base/offset notation. In general it is expected that most MMIO targets with be addressed via an offset from a base address. The user may define up to 4 different base address registers to be used by an combination of targets. Under many conditions only a single base address register is required. Further, it is expected that a given target may be an DCSN bridge and therefore require a number of other regions to be mapped to that target.

In order to accommodate these two different addressing needs each DCSN target has several options for address decoding. For each DCSN target the user can specify one or more apertures which start from any of the defined base registers. These offsets may be specified in increments of 4k and further the size of the target aperture can be specified in increments of 4k. For some cases, a given target may not have an addressable region within the MMIO aperture, and this can also be accommodated. For targets such as this, the only addresses that can be mapped to those targets are additional regions that are specified at compile time.

The user has the flexibility when arranging MMIO targets to leave holes between targets by simply specifying non-contiguous offsets. It is up to the user to either leave these holes or place Null targets in each of the holes. If null modules are provided then when an offset of 0xffc within each hole is addressed, the device responds with a module ID and size of the region.

In addition to addressing each DCSN target with a base/offset method, each target may also be addressed by additional regions.

5.4.3.3 Operation

Clock Programming: The target operating frequency for *dcs_clk* is 192 MHz. There is no lower limit on the frequency (the clock can be slowed without adverse affects). The clock frequencies of initiator and target ports can be faster, slower, or the same as the bus controller.

Reset: The *dcs_rst_an* signal is an active low, asynchronous input. Resetting of all flip flops in the bus controller is achieved synchronously (i.e. by affecting the “D” input of flip flops). There is no requirement for immediate (clear-direct) reset of any flip flops. The *dcs_rst_an* signal is synchronized by connecting it to a two-bit shift register clocked by *dcs_clk*. The output of the synchronizer flip flops is distributed to synchronously reset all other flip flops in the bus controller.

Module-Specific Capabilities: Timeout: The timeout block uses a 17-bit counter to count clock cycles of an active transaction. The counter increments when the select signal (sel) is high. The counter synchronous resets to zero when sel is low.

When the timeout counter reaches a certain value, determined by the control register BC_CTRL, the counter stops incrementing and the abort_all signal is sent to the currently selected target. Each timeout limit is a number equal to 2n-1, allowing the timeout detection circuit to be a simple mux which selects one bit from the timeout counter as shown in Figure 179. Note that the three least significant bits of the counter are not monitored, because no transaction can complete in less than four clock cycles.

When a timeout occurs, the dcs#_abort_all signal is asserted to the currently selected target. The timeout condition is also logically ORed with the dcs_cmd_err_t input to force an error indication back to the target. This OR gate is before the MIPS error blocking logic described in Section “Error Handling”.

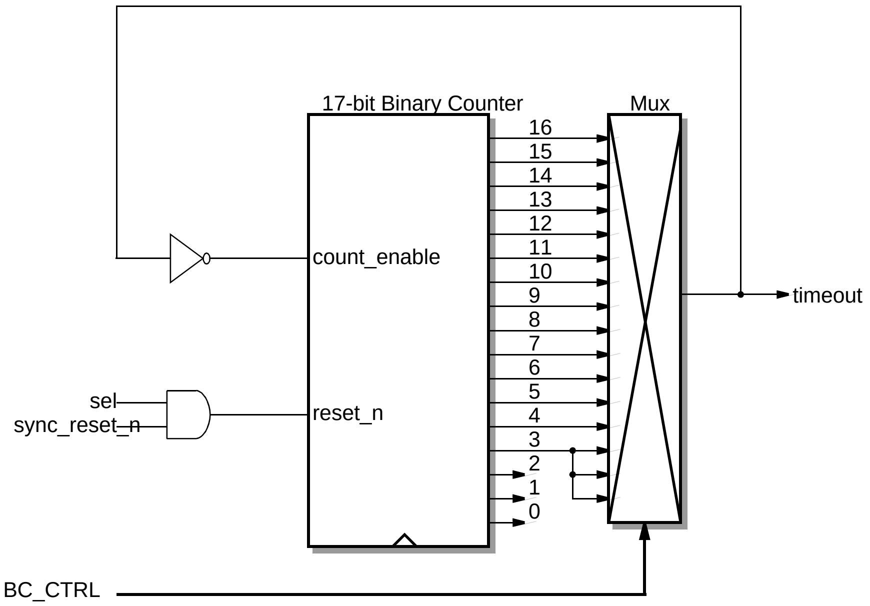


Fig 179.Timeout Circuit

Error Handling: Error capture registers inside the bus controller capture the current address and operation that was in progress when an error is reported or when a timeout occurs. Once an error has been captured, the capture registers (BC_ADDR and BC_STAT) are no longer updated until the interrupt is cleared.

Errors caused by MIPS 32-bit read operations are not captured, and not reported to the MIPS. Therefore, when the currently selected initiator is the MIPS (as determined by the Arbiter), the operation is a read (dcs_cmd_read_t is high), and the mask is all ones (dcs_cmd_mask = 0xF), any errors are blocked, including timeout errors. In this case, the capture registers are not updated and the dcs*_cmd_err_i signals are not asserted.

When a non-blocked error or timeout does occur, it sets an interrupt flag in the BC_INT_STATUS register. Bit 0 is set when dcs_cmd_err_t was asserted, and bit 1 is set when a timeout occurs. When either of these two bits are 1, the error capture registers cannot be updated. When both bits are zero, the error_capture registers update every clock cycle, capturing the current transaction information.

Bit 1 of the BC_STAT register is set high when a non-block error occurs with no timeout. If timeout is detected at the exact same time that an error response is received from a target, bit 1 of BC_STAT is set to indicate a timeout.

The Interrupt enable bits in the BC_INT_EN register affects which interrupt flags can cause the dcs_intreq signal to be asserted. These bits do not affect the updating of the BC_INT_STATUS register.

Selective Blocking of Initiators (System Security): In order to facilitate the implementation of system security features, the DCSC optionally includes selective blocking of initiators. If this option is specified then an additional 4k aperture is created internal to the DCSN controller, which contains the system security registers. A register is created for each target with a width of the number of DCSN initiators in the system. Note that it is not just the number of DCSN initiators attached to the individual bus controller, but the entire number in the system. The initiator ID is used to identify which initiator is requesting access and depending upon the value of the access register for that target, either a grant is given or an error returned. The user has the ability to specify the power on value of the security region access register. This insures that only a known trusted initiator has access to the security aperture, and thus the ability to change any access permissions for targets attached to that local network controller. All other targets allow access to all system initiators until programmed otherwise.

Configuration Interface: All configuration information for a given bus controller is programmed via any of the DCSN Initiator Interfaces for that bus controller. This includes address map information, captured error information, interrupt enabling/disabling, and selective blocking of initiators. Optionally, the address map information may be programmed elsewhere and only appear as inputs to the bus controller. The network controller contains a separate apertures for the selective blocking registers as described in Section "Selective Blocking of Initiators (System Security)". Therefore everything, but the access registers for system security are programmed via one configuration interface. When the address map information is contained within the bus controller aperture, take care when changing the base registers as they effect the aperture map.

Arbitration: A 'round robin' Arbiter (see paragraph below) is used to selectively grant access to each of the requesting initiators. The Arbiter grants by default the last device that was granted. Therefore, when no initiator is requesting access, the default grant is given to the initiator that most recently performed a transaction. The initiator with a default grant can access a target one clock cycle faster than an initiator without the default grant. Assigning the default grant to the initiator that most recently used the "bus" is expected to yield the highest performance, since one initiator is likely to execute several transactions at once. Figure 180 shows a block diagram for the Arbiter.

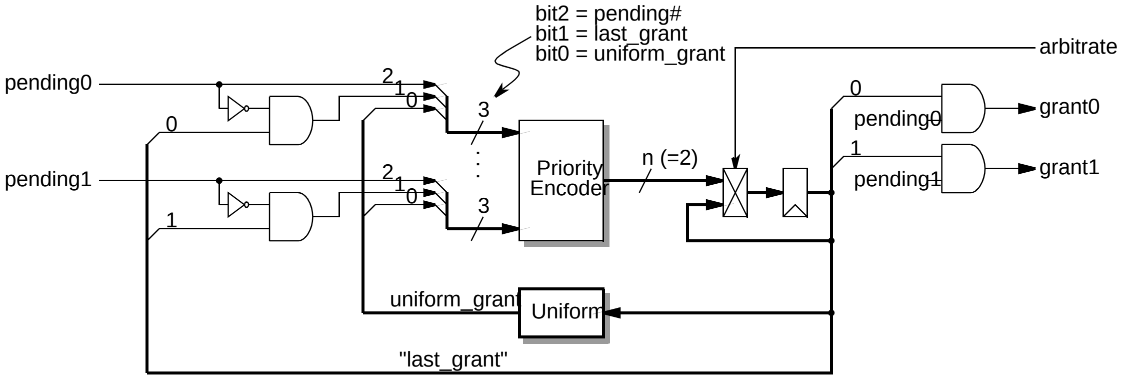
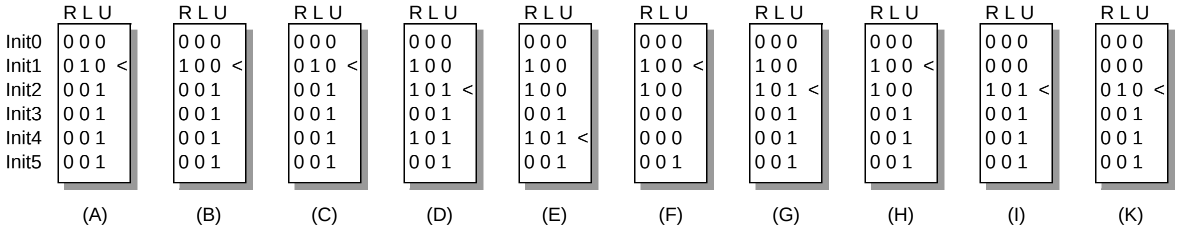


Fig 180. Arbiter Block Diagram

To achieve the 'round robin' feature with a dynamic default grant, the Arbiter uses an internal priority comparator. The comparator selects an initiator to grant by comparing the "priorities" of each device. The priority value consists of three bits. The most significant bit is high when there is a pending request from that initiator. The second most significant bit is generated by "last_grant", which is high for the most recently granted initiator only if that initiator does not have a pending request, and low for all other initiators. The third bit is a “uniform scheduling” value. This is set to one for all bit locations higher than the last granted initiator and zero for all lower ones. The priority block picks the highest value (3-bit) input. In the case of a tie, the lower numbered port wins.

To understand how this works, consider a 6-initiator system. Assume that at time zero, the most recent initiator requesting was initiator 1, but currently no initiators are making a request. In that case, the 3-bit values into the priority block would be as shown in column (A) of Figure 181. The R (request) bits are all low. The L bit for initiator 1 is high (last initiator granted), and the U bits for initiators 2-5 are high (all higher than the last initiator granted).The priority for initiator 0 is 0 (binary 000), for initiator 1 it is 2 (binary 010), and for all other initiators it is 1 (binary 001).



	R L U	R L U	R L U	R L U	R L U	R L U	R L U	R L U	R L U	R L U
Init0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
Init1	0 1 0 <	1 0 0 <	0 1 0 <	1 0 0	1 0 0	1 0 0 <	1 0 0	1 0 0 <	0 0 0	0 0 0
Init2	0 0 1	0 0 1	0 0 1	1 0 1 <	1 0 0	1 0 0	1 0 1 <	1 0 0	1 0 1 <	0 1 0 <
Init3	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 0	0 0 1	0 0 1	0 0 1	0 0 1
Init4	0 0 1	0 0 1	0 0 1	1 0 1	1 0 1 <	0 0 0	0 0 1	0 0 1	0 0 1	0 0 1
Init5	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1
	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)	(I)	(K)

Fig 181. Example Priority Bit Values

Now, at some later time, assume initiator 1 makes a request. That forces the L bit low, but this initiator is still granted because the priority is now 4 (binary 100). This is shown in column B. Later, after completing the transaction, the request from initiator 1 is low, and the state is as shown in column C. If simultaneously three initiators, 1, 2, and 4, all make a request, then initiator 1 is granted first (because it had the default grant). When the Arbiter signal becomes true, the Arbiter moves the grant to initiator 2 because it is the new high-priority initiators, as shown in column D. On the next arbitration cycle, initiator 4 is granted (column E). Assuming initiators 1 and 2 continued to have requests, the Arbiter would next grant initiator 1, and then 2 (columns F and G). Then 1 again (column H), and 2 again (column I). Now, assuming the initiators each dropped their request following the last round of transactions, a high L bit for initiator 2 is left - the last initiator granted. The default grant therefore is given to initiator 2.

Handshake Control: The handshake control circuit is responsible for driving the select line to the address decoder, and for driving the update lines to the initiator response circuits.

5.4.3.4 Register Descriptions

Register Summary: DCSC Configuration Registers: Table 265 summarizes the control and status registers visible inside the DCSC.

Table 265: DCSC Configuration Register Summary

Offset	Name	Description
0x0	DCSC_CTRL	DCS control register
0xC	DCSC_ADDR	DCS error/timeout address capture register
0x10	DCSC_STAT	DCS error/timeout status register
0x40	DCSC_FEATURES	DCSC features register
0x100	BASE_REG_0	DCS base address register 0

Table 265: DCSC Configuration Register Summary *…continued*

Offset	Name	Description
0x104	BASE_REG_1	DCS base address register 1 (optional)
0x108	BASE_REG_2	DCS base address register 2 (optional)
0x10C	BASE_REG_3	DCS base address register 3 (optional)
0xFD8	INT_CLR_ENABLE	Interrupt clear enable
0xFDC	INT_SET_ENABLE	interrupt set enable
0xFE0	INT_STATUS	interrupt status
0xFE4	INT_ENABLE	Interrupt Enable
0xFE8	INT_CLR_STATUS	Interrupt clear status
0xFEC	INT_SET_STATUS	interrupt set status
0xFFC	MODULE_ID	DCSC module ID register (configuration aperture)

Remark: The BC_INT_EN register is R/W for backward compatibility. Software drivers consider BC_INT_EN as read only and use the BC_INT_CLR_ENABLE and BC_INT_SET_ENABLE registers to update the value of BC_INT_EN.

DCSC Security Registers: The DCSN may be optionally compiled to include selective initiator blocking support, which may be used to enforce system security. If this option is selected, the DCSN creates a separate 4 KB aperture that contains a number of registers needing to be programmed.

Since the system security settings are stored within the network controller, it may be necessary to restrict access to the bus controller security aperture. Therefore, at compile time the user specifies the access permissions for the security aperture by indicating which initiator IDs can access this aperture. By default, all initiators can access all targets.

Table 266 summarizes the control and status registers visible inside the DCSC Security aperture.

Table 266: DCSC Security Register Summary

Offset	Name	Description
0x0	ACCESS_DCSC	DCSC Configuration Aperture Access Control Register
0x4	ACCESS_DCSC_SEC	DCSC Security Aperture Access Control Register
0x200	ACCESS_TGT0	DCS Target DCS_Control Access Control Register
0x204	ACCESS_TGT1	DCS Target DCS_Security Access Control Register
0x208	ACCESS_TGT2	DCS Target Global Registers Access Control Register
0x20C	ACCESS_TGT3	DCS Target LVDS_Tx Access Control Register
0x210	ACCESS_TGT4	DCS Target Clocks Access Control Register
0x214	ACCESS_TGT5	DCS Target GIC Access Control Register
0x218	ACCESS_TGT6	DCS Target Reset Access Control Register
0x21C	ACCESS_TGT7	DCS Target PMAN1_monitor Access Control Register
0x220	ACCESS_TGT8	DCS Target PMAN1_Security Access Control Register
0x224	ACCESS_TGT9	DCS Target PMAN1_Arbiter Access Control Register
0x228	ACCESS_TGT10	DCS Target VO-1 Access Control Register
0x22C	ACCESS_TGT11	DCS Target VO-2 Access Control Register
0x230	ACCESS_TGT12	DCS Target VIP Access Control Register

Table 266: DCSC Security Register Summary *…continued*

Offset	Name	Description
0x234	ACCESS_TGT13	DCS Target MBS_V2 Access Control Register
0x238	ACCESS_TGT14	DCS Target TSU Access Control Register
0x23C	ACCESS_TGT15	DCS Target VMPG Access Control Register
0x240	ACCESS_TGT16	DCS Target PMAN2_Monitor Access Control Register
0x244	ACCESS_TGT17	DCS Target PMAN2_Security Access Control Register
0x248	ACCESS_TGT18	DCS Target Columbus_Monitor Access Control Register
0x24C	ACCESS_TGT19	DCS Target DDR Controller Access Control Register
0x250	ACCESS_TGT20	DCS Target North Tunnel Config Access Control Register
0x254	ACCESS_TGT21	DCS Target South Tunnel Config Access Control Register
0x258	ACCESS_TGT22	DCS Target DMA Gate Access Control Register
0x25C	ACCESS_TGT23	DCS Target North tunnel Access Control Register
0xFFC	MODULE_ID_DCS_SEC	DCS Security MODULE ID Register

Register Tables: DCSC Configuration Registers: Table 267 shows detailed bit locations for each register.

Table 267: DCSC Configuration Registers

Bit	Symbol	Access	Value	Description
Offset 0x0 - DCSC_CTRL				
31:5	RSD[31:5]	W	0xXXXXXXX	Ignore upon read. Write as zeros

4:1	TOUT_SEL[3:0]	R/W	0x0	Timeout select 0 = Timeout generated after 7 consecutive wait cycles; 1 = Timeout generated after 7 consecutive wait cycles; 2 = Timeout generated after 7 consecutive wait cycles; 3 = Timeout generated after 15 consecutive wait cycles; 4 = Timeout generated after 31 consecutive wait cycles; 5 = Timeout generated after 63 consecutive wait cycles; 6 = Timeout generated after 127 consecutive wait cycles; 7 = Timeout generated after 255 consecutive wait cycles; 8 = Timeout generated after 511 consecutive wait cycles; 9 = Timeout generated after 1023 consecutive wait cycles; A = Timeout generated after 2047 consecutive wait cycles; B = Timeout generated after 4095 consecutive wait cycles; C = Timeout generated after 8191 consecutive wait cycles; D = Timeout generated after 16383 consecutive wait cycles; E = Timeout generated after 32767 consecutive wait cycles; F = Timeout generated after 65535 consecutive wait cycles;
				0: Timeout generated after 7 consecutive wait cycles;
				1: Timeout generated after 7 consecutive wait cycles;
				10: Timeout generated after 7 consecutive wait cycles;
				11: Timeout generated after 15 consecutive wait cycles;
				100: Timeout generated after 31 consecutive wait cycles;
				101: Timeout generated after 63 consecutive wait cycles;
				110: Timeout generated after 127 consecutive wait cycles;
				111: Timeout generated after 255 consecutive wait cycles;
				1000: Timeout generated after 511 consecutive wait cycles;
				1001: Timeout generated after 1023 consecutive wait cycles;

0	TOUT_OFF	R/W	0x1	Timeout disable
				0: Timeout enabled;
				1: Timeout disabled;

Offset 0xC - DCSC_ADDR				
31:2	ERR_TOUT_ADDR[31:2]	R	0x00000000	Full 30 bits of the address which causes an error or timeout
1:0	RSD[1:0]	R	0xX	Ignore upon read. Write as zeros

Table 267: DCSC Configuration Registers *…continued*

Bit	Symbol	Access	Value	Description
Offset 0x10 - DCSC_STAT				
31:29	RSD[31:29]	R	0xX	Ignore upon read. Write as zeros
28:24	ERR_TOUT_GNT[4:0]	R	0x00	Active initiator causing error or timeout - Initiator ID of the current pending transaction will be captured
23:17	RSD[23:17]	R	0xXX	Ignore upon read. Write as zeros
16:10	ERR_TOUT_SEL[6:0]	R	0x00	Selected agent during error or timeout 7'b 0000000 = Target 0; 7'b 0000001 = Target 1; 7'b 0000010 =Target 2; . . 7'b 0111111 = Target 63; 7'b1000000 = Any Null target or non successful access; 7'b 1000001 = Network Controller configuration Aperture; 7'b 1000010 = Network Controller security Aperture;
9	RSD_Bit9	R	0xX	Ignore upon read. Write as zeros

8	ERR_TOUT_READ	R	0x0	Value of cmd_read signal during error or timeout
				0: write operation;
				1: read operation;

7:4	ERR_TOUT_MASK[3:0]	R	0x0	Value of cmd_mask during error or timeout. Indicates which bytes were to be read or written.
3:2	RSD[3:2]	R	0xX	Ignore upon read. Write as zeros

1	ERR_ACK	R	0x0	Error or Timeout
				0: timeout;
				1: error;

0	RSD_Bit0	R	0xX	Ignore upon read. Write as zeros
Offset 0x40 - DCSC_FEATURES				
31:19	RSD[31:19]	R	0xXXXX	Ignore upon read. Write as zeros
18:16	UNIQUE_ID[2:0]	R	0xX	This field is user provided and optional. This field may be used by software if more than one controller is present with the same number of initiators and targets. This would further allow the integrator to differentiate the two controller, without having to specify different module IDs. When this field is present, it resets to the provided value, otherwise it is 0.
15	RSD_Bit15	R	0xX	Ignore upon read. Write as zeros
14	SECURITY	R	0xX	Indicates whether the DCS controller has been compiled with security features. If so then a security aperture exists, and controls access to all targets, including the DCS configuration and DCS security apertures.
13	RSD_Bit13	R	0xX	Ignore upon read. Write as zeros
12:11	NUM_BASE_REGS[1:0]	R	0xX	Number of base registers defined for the DCS controller (zero relative). Valid values are 0x0 (1 base reg) to 0x3 (4 base regs)

Table 267: DCSC Configuration Registers *…continued*

Bit	Symbol	Access	Value	Description
10:5	NUM_TARGETS[5:0]	R	0xXX	Number of DCS target attached to the DCS controller (zero relative). Valid values are 0x0 (1 target) to 0x3f (64 targets)
4:0	NUM_INITIATORS[4:0]	R	0xXX	Number of DCS initiators attached to the DCS controller (zero relative). Valid values are 0x0 (1 initiator) to 0x1f (32 initiators)
Offset 0x100 - BASE_REG_0				
31:12	BASE_0_REG[19:0]	R/W	0xXXXXX	Base register n - this base register is used in address decoding. This field ranges in width from 1 -19 bits and is used to match the upper bits of address. The user may then program a number of targets which exist at an offset from this base address. The offset will match the lower bits of the address from y-1 down to 12.
11:0	RSD[11:0]	R	0x000	Unused (minimum target size is 4K)
Offset 0x104 - BASE_REG_1				
31:12	BASE_1_REG[19:0]	R/W	0xXXXXX	Base register n - this base register is used in address decoding. This field ranges in width from 1 -19 bits and is used to match the upper bits of address. The user may then program a number of targets which exist at an offset from this base address. The offset will match the lower bits of the address from y-1 down to 12.
11:0	RSD[11:0]	R	0x000	Unused (minimum target size is 4K)
Offset 0x108 - BASE_REG_2				
31:12	BASE_2_REG[19:0]	R/W	0xXXXXX	Base register n - this base register is used in address decoding. This field ranges in width from 1 -19 bits and is used to match the upper bits of address. The user may then program a number of targets which exist at an offset from this base address. The offset will match the lower bits of the address from y-1 down to 12.
11:0	RSD[11:0]	R	0x000	Unused (minimum target size is 4K)
Offset 0x10C - BASE_REG_3				
31:12	BASE_3_REG[19:0]	R/W	0xXXXXX	Base register n - this base register is used in address decoding. This field ranges in width from 1 -19 bits and is used to match the upper bits of address. The user may then program a number of targets which exist at an offset from this base address. The offset will match the lower bits of the address from y-1 down to 12.
11:0	RSD[11:0]	R	0x000	Unused (minimum target size is 4K)
Offset 0xFD8 - INT_CLR_ENABLE				
31:2	RSD[31:2]	W	0xXXXXXXXX	Ignore upon read. Write as zeros

Table 267: DCSC Configuration Registers *…continued*

Bit	Symbol	Access	Value	Description
1	INT_CLR_ENABLE_TOUT	W	0x0	Timeout interrupt enable clear register. This is written by software to clear the interrupt enable (bit 1 of DCSC_INT_EN). 0: Timeout interrupt enable is unchanged; 1: Timeout interrupt enable is cleared;
0	INT_CLR_ENABLE_ERROR	W	0x0	Error interrupt enable clear register. This is written by software to clear the interrupt enable (bit 0 of DCSC_INT_EN). 0: Error interrupt enable is unchanged; 1: Error interrupt enable is cleared;
Offset 0xFDC - INT_SET_ENABLE				
31:2	RSD[31:2]	W	0xXXXXXXXX	Ignore upon read. Write as zeros
1	INT_SET_ENABLE_TOUT	W	0x0	Timeout interrupt enable set register. This is written by software to set the interrupt enable (bit 1 of DCSC_INT_EN). 0: timeout interrupt enable is unchanged; 1: timeout interrupt enable is set;
0	INT_SET_ENABLE_ERROR	W	0x0	Error interrupt enable set register. This is written by software to set the interrupt enable (bit 0 of DCSC_INT_EN). 0: error interrupt enable is unchanged; 1: error interrupt enable is set;
Offset 0xFE0 - INT_STATUS				
31:2	RSD[31:2]	R	0xXXXXXXXX	Ignore upon read. Write as zeros
1	INT_STATUS_TOUT	R	0x0	Timeout interrupt status. Reports any pending timeout interrupts: 0: Timeout interrupt is not pending; 1: DCS controller has generated a timeout because a target has violated the programmable limit for timeout
0	INT_STATUS_ERROR	R	0x0	Error interrupt status. Reports any pending error interrupts: 0: Error interrupt is not pending; 1: meaning network controller has detected an error acknowledge from a target.
Offset 0xFE4 - INT_ENABLE				
31:2	RSD[31:2]	W	0xXXXXXXXX	Ignore upon read. Write as zeros
1	INT_ENABLE_TOUT	R/W	0x0	Timeout interrupt enable register 0: timeout interrupt is disabled; 1: timeout interrupt is enabled;

Table 267: DCSC Configuration Registers *…continued*

Bit	Symbol	Access	Value	Description
0	INT_ENABLE_ERROR	R/W	0x0	Timeout interrupt enable register 0: error interrupt is disabled; 1: error interrupt is enabled;
Offset 0xFE8 - INT_CLR_STATUS				
31:2	RSD[31:2]	W	0xXXXXXXXX	Ignore upon read. Write as zeros
1	INT_CLEAR_TOUT	W	0x0	Timeout interrupt clear register. This is written by software to clear the interrupt 0: timeout interrupt is unchanged; 1: timeout interrupt is cleared;
0	INT_CLEAR_ERROR	W	0x0	Error interrupt clear register. This is written by software to clear the interrupt 0: error interrupt is unchanged; 1: error interrupt is cleared;
Offset 0xFEC - INT_SET_STATUS				
31:2	RSD[31:2]	W	0xXXXXXXXX	Ignore upon read. Write as zeros
1	INT_SET_TOUT	W	0x0	Timeout interrupt set register. Allows software to set interrupts 0: timeout interrupt is unchanged; 1: timeout interrupt is set;
0	INT_SET_ERROR	W	0x0	Error interrupt set register. Allows software to set interrupts 0: error interrupt is unchanged; 1: error interrupt is set;
Offset 0xFFC - MODULE_ID				
31:16	ID[15:0]	R	0xA06D	Identification Number: Unique module identifier. This value defaults to 0xA06D, which is the register module ID for the DCS Controller template. Typically it is expected that this module ID will be used, unless there is a reason for another ID. If an alternative value is specified then that value will be the reset value for this field
15:12	MAJOR_REV[3:0]	R	0x1	Major Revision: Major revision of module implementation. For templates the major and minor revision match the revision of the template
11:8	MINOR_REV[3:0]	R	0x0	Minor revision: Minor revision of module implementation, starting at 0
7:0	APERTURE_SIZE[7:0]	R	0x00	Aperture size: 4KByte address aperture

DCSC Security Registers: Table 268 shows detailed bit locations for each register. In the “Offset” column, n is equal to the number of initiators in the system and m is equal to the number of targets attached to this specific DCSC.

Table 268: DCSC Security Registers

Bit	Symbol	Access	Value	Description
Offset 0x0 - ACCESS_DCSC				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_CONFIG	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_CONFIG	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x4 - ACCESS_DCSC_SEC				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_SECURITY	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.

Table 268: DCSC Security Registers *…continued*

Bit	Symbol	Access	Value	Description
0	INIT_ID_0_SEC_SECURITY	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x200 - ACCESS_TGT0				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT0	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_TGT0	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x204 - ACCESS_TGT1				

Table 268: DCSC Security Registers *…continued*

Bit	Symbol	Access	Value	Description
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT1	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_TGT1	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x208 - ACCESS_TGT2				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT2	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.

Table 268: DCSC Security Registers *…continued*

Bit	Symbol	Access	Value	Description
0	INIT_ID_0_SEC_TGT2	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x20C - ACCESS_TGT3				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT3	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_TGT3	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x210 - ACCESS_TGT4				

Table 268: DCSC Security Registers *…continued*

Bit	Symbol	Access	Value	Description
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT4	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_TGT4	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x214 - ACCESS_TGT5				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT5	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.

Table 268: DCSC Security Registers *...continued*

Bit	Symbol	Access	Value	Description
0	INIT_ID_0_SEC_TGT5	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x218 - ACCESS_TGT6				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT6	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_TGT6	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x21C - ACCESS_TGT7				

Table 268: DCSC Security Registers *…continued*

Bit	Symbol	Access	Value	Description
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT7	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_TGT7	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x220 - ACCESS_TGT8				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT8	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.

Table 268: DCSC Security Registers *...continued*

Bit	Symbol	Access	Value	Description
0	INIT_ID_0_SEC_TGT8	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x224 - ACCESS_TGT9				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT9	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_TGT9	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x228 - ACCESS_TGT10				

Table 268: DCSC Security Registers *...continued*

Bit	Symbol	Access	Value	Description
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT10	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_TGT10	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x22C - ACCESS_TGT11				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT11	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.

Table 268: DCSC Security Registers *…continued*

Bit	Symbol	Access	Value	Description
0	INIT_ID_0_SEC_TGT11	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x230 - ACCESS_TGT12				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT12	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_TGT12	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x234 - ACCESS_TGT13				

Table 268: DCSC Security Registers *…continued*

Bit	Symbol	Access	Value	Description
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT13	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_TGT13	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x238 - ACCESS_TGT14				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT14	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.

Table 268: DCSC Security Registers *…continued*

Bit	Symbol	Access	Value	Description
0	INIT_ID_0_SEC_TGT14	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x23C - ACCESS_TGT15				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT15	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_TGT15	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x240 - ACCESS_TGT16				

Table 268: DCSC Security Registers *...continued*

Bit	Symbol	Access	Value	Description
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT16	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_TGT16	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x244 - ACCESS_TGT17				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT17	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.

Table 268: DCSC Security Registers *...continued*

Bit	Symbol	Access	Value	Description
0	INIT_ID_0_SEC_TGT17	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x248 - ACCESS_TGT18				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT18	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_TGT18	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x24C - ACCESS_TGT19				

Table 268: DCSC Security Registers *...continued*

Bit	Symbol	Access	Value	Description
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT19	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_TGT19	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x250 - ACCESS_TGT20				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT20	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.

Table 268: DCSC Security Registers *...continued*

Bit	Symbol	Access	Value	Description
0	INIT_ID_0_SEC_TGT20	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (e.g. Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x254 - ACCESS_TGT21				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT21	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_TGT21	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (eg Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x258 - ACCESS_TGT22				

Table 268: DCSC Security Registers *…continued*

Bit	Symbol	Access	Value	Description
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT22	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.
0	INIT_ID_0_SEC_TGT22	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (eg Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0x25C - ACCESS_TGT23				
31:2	RSD[31:2]	R/W	0xXXXX XXXX	Unused
1	INIT_ID_1_SEC_TGT23	R/W	0x1	Initiator ID South Tunnel Security Value - Refer to INIT_ID_0_SEC Wherever IIC is used replace with South Tunnel.

Table 268: DCSC Security Registers *...continued*

Bit	Symbol	Access	Value	Description
0	INIT_ID_0_SEC_TGT23	R/W	0x1	Initiator ID IIC Security Value This field controls the permissions of Initiator IIC to access the specified DCS Target. 0 - IIC is not allowed to access the target and a DCS error will be generated on attempts to access the target. No external target select will be generated externally for disallowed initiators. 1 - IIC is allowed to access the target. If the initiator attempting the access is compiled with an error blocking logic enabled (eg Trimedia) and the access is a 32bit read then no error will be generated and the transaction will appear to have completed.
Offset 0xFFC - MODULE_ID_DCS_SEC				
31:16	ID[15:0]	R	0xA06E	Identification number: Unique module identifier. This value defaults to 0xA06E, which is the registered Module ID for the DCS Controller Template (Security Aperture). Typically it is expected that this module ID will be used, unless there is a reason for another ID. If an alternate value is specified then that value will be the Reset value for this field.
15:12	MAJOR_REV[3:0]	R	0x1	Major Revision: Major revision of module implementation. For templates the major and minor revision match the revision template.
11:8	MINOR_REV[3:0]	R	0x0	Minor Revision: Minor revision of module implementation, starting at 0.
7:0	APERTURE_SIZE[7:0]	R	0x00	Aperture size: 4KByte address aperture

5.5 Pipeline Memory Access Network (PMAN) hub

5.5.1 Introduction

This section describes the PMAN (also known as DMA-HUB or “the HUB”) in the PNX2015. The HUB includes a generic Arbiter which is able to control data flow within complex memory systems and a PMA Security block. See Figure 182 and Figure 183 for more details.

The PMAN Hub provides DMA data paths and control, linking peripheral devices to the main memory controller and providing high speed memory access. The Arbiter controls which peripheral gains access to the main memory controller via the Hub. The PMA security block provides separation of memory areas within main memory to limit unintentional interference by a mis-programmed peripheral.

5.5.1.1 Features

The key features of the HUB are:

- Provides a hierarchical memory access network that connects Peripheral DMA ports to one of two targets. Based on a programmable address mask, transactions are either routed to the System Memory Controller or to the South Tunnel (to access System Memory of PNX8550).
- Includes simple round-robin sub-arbitration for lower levels of hierarchy.
- Provides sophisticated intermediate arbitration for upper levels of the network hierarchy (HUB1).
- Utilizes round-robin Arbiter to provide arbitration for upper levels of the network hierarchy (HUB2).
- Includes a security mechanism to limit memory access of Peripherals to programmable regions in system memory (MSEC).
- Provides data synchronization, transaction buffering and partitioning mechanisms
- Provides VIP - V0 data streaming.
- Includes a memory access “gate” (MGATE) for MMIO transactions, for debug applications via EJTAG.

5.5.2 Functional description

5.5.2.1 PNX2015 hub block

The following diagram shows the Hub as it interconnects within the PNX2015 system.

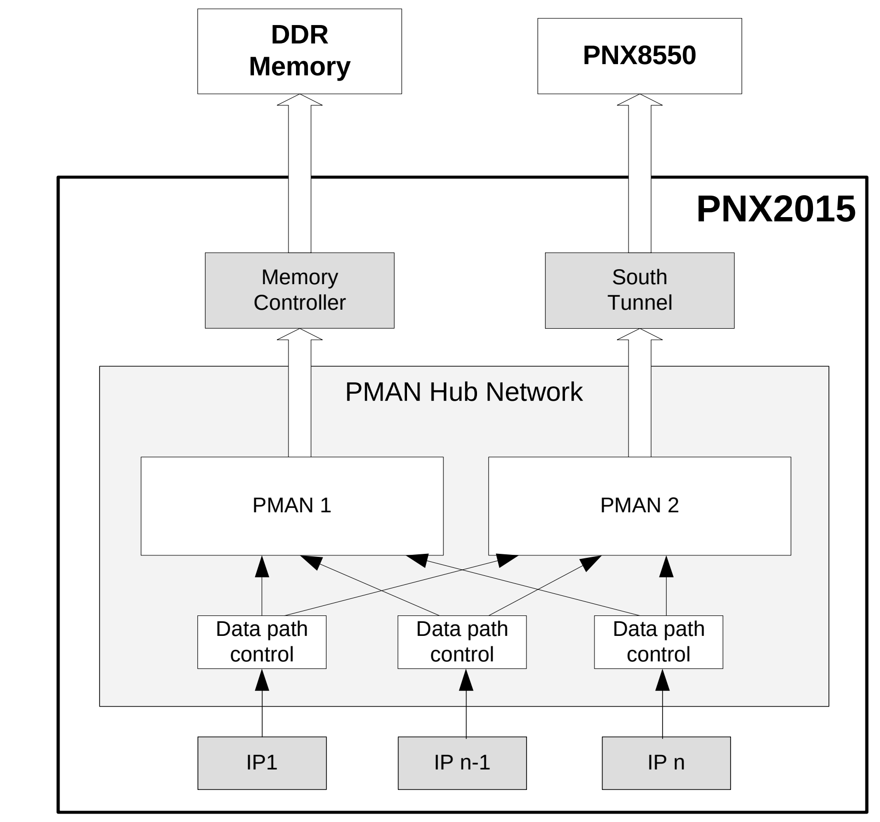


Fig 182.Overall system diagram

Figure 183 shows the hub and peripheral devices that transfer data via the hub.

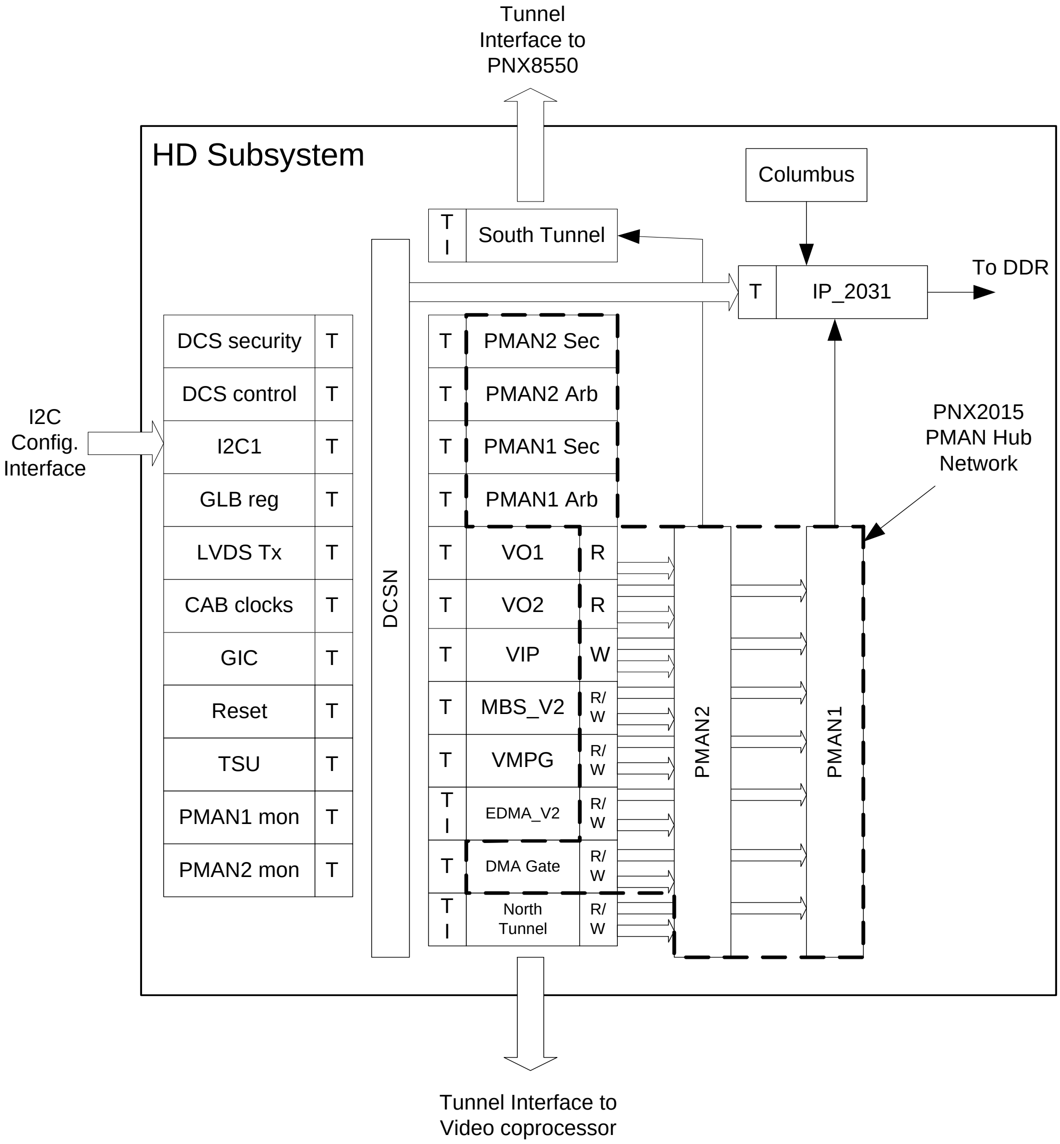


Fig 183.PNX2015 PMAN Hub Network place in the system

5.5.2.2 Architecture

The NHP_HUB_2015 module contains the following blocks:

- PMA1 Arbiter
- PMA1 Security (MSEC2)
- PMA2 Arbiter (round-robin)
- PMA2 Security (MSEC2)
- DMA Adapters (DTL to MTL)
- DMA Gate (MGATE)

PMA arbitration: Multiple levels of arbitration exist in the PNX2015 PMA-network. round-robin arbitration is done between multiple DMA channels of each IP. In some cases traffic of read and write channels for one IP is separated in order to provide a finer control over bandwidth allocation (grouping of read and write transaction in order to improve overall memory bandwidth).

Transaction requests from these “clustered” channels are then forwarded to the next level together with an ID tag (see Table 269 below). Depending on the transactions target, arbitration is done via the programmable Arbiter block (PMA1) or a generic Round-Robin scheme (PMA2).

MTL ID mapping: Table 269 shows the mapping of each peripheral device to unique identification numbers. It also shows the amount of sub arbitration for the given peripherals. Unless otherwise noted, the amount of buffering per DMA channel is 256 bytes.

Table 269: MTL ID mapping and subarbitration

ID	Peripheral	DMA channels	Remarks
0	Video Output 1	3 x R	Ch#1 ip-ip streaming enabled
1	Video Output 2	3 x R	
2	MBS Read	3 x R	
3	MBS Write	3 x W	
4	VIP	3 x W	Ch#1 ip-ip streaming enabled
5	VMPG Regular	2 x R, 3 x W	
6	North Tunnel	1 x RW	two MTL ports, external buffers
7	Memory Gate	1 x RW	single transaction buffer

PMA security: The PMAN Security block (dvi_msec for “memory security”) invalidates memory access to all locations that are not inside any of a peripheral’s assigned sandbox[5]. There are four sandboxes defined via a lower and upper address limit for each sandbox. Each peripheral device can be associated with each sandbox under software control. The sandbox access information is held in a set of registers with a dedicated entry for each peripheral.

Access to these registers can be enabled/disabled for every initiator via the DCSS module.

Implementing 4 sandboxes allows a sandbox for both the MIPS and TM3260-controlled peripherals, plus 2 additional sandboxes for special purposes. The sandboxes adhere to the following rules:

- Sandbox address ranges can be overlapping.
- Granularity of the address range is 64 kB.
- There are 2 registers for each sandbox; sandbox Base Address and sandbox Top Address.
- A peripheral can be assigned to several or no sandboxes.
- There is no support for an inverted sandbox (Base_Address > Top_Address).

[5] Memory access region - A Sandbox is an area outside of the main development tree where developers can develop their code without affecting other users or the development tree.

In the case of an access violation, the address value and ID value of the attempted access is stored in the “Protection Error Address” and Interrupt Status registers within the dvi_msec module. In case of multiple violators, only the address and ID of the first violator are stored. The Protection Error Address register can be ‘re-opened’ to store the new Address and ID values by clearing the “Memory Access protection Error” status bit (STAT_PROT_ERR).

DMA gate: The DMA gate allows single transfer access to both PMA1 & PMA2 Networks. The Gate only supports one outstanding transaction at any given point in time. Write posting, if desired, can be enabled through the associated DCS to DTL adapter (if available).

5.5.2.3 Data flow and control

Arbitration algorithm: See Figure 184.

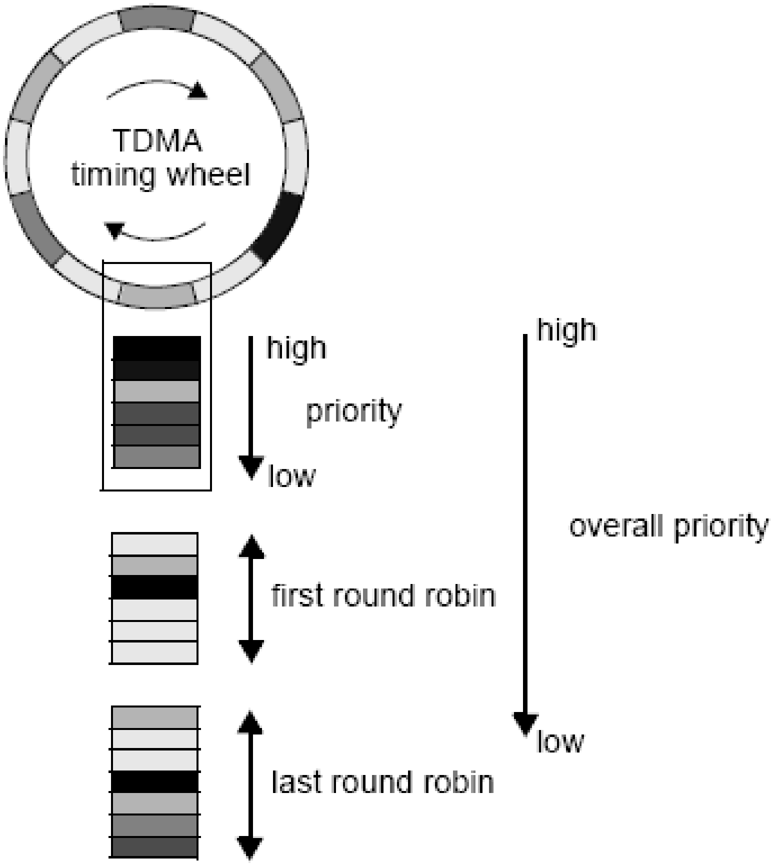


Fig 184.Arbiter for DMA traffic arbitration algorithm

When the arbiter has been configured to include the priority list and both round-robin lists, any arbiter decision is made in the following four steps:

1. First the DMA requests are compared against the current entry in the TDMA timing wheel. If the agent in the current entry is requesting this agent will be granted.

2. If the agent in the current entry is not requesting the DMA requests will be compared against the agents in the priority list and if one or more of the agents in the priority list is requesting the one that has the highest priority will be granted.
3. If none of the DMA requests matches the current entry in the TDMA timing wheel or one or more entries in the priority list, the arbiter will grant the DMA agent that hasn't been served for the longest time from the first round robin list of entries. Every time the arbiter grants any DMA agent, the round robin arbiter checks if this agent is in it's list and makes it the lowest priority entry in the round robin table. If a certain agent is granted because of it's entry in the TDMA timing wheel or priority list and the same agent has also an entry in the round-robin list in the next clockcycle this agent will have the lowest priority in the round-robin list. Also in case there are multiple entries of the same agent in the round-robin list on a hit only the highest entry in the list gets the lowest priority during the next cycle. The other entries of the same agents do not get the lowest priority.
4. If none of the DMA requests matches the current entry in the TDMA timing wheel or one or more entries in the priority list or one or more entries in the first round-robin list the arbiter will grant the DMA agent that hasn't been served for the longest time from the last round robin list of entries. This round-robin list operates the same way as the first round-robin list but all entries in this list have a lower priority than the entries in the first round-robin list.

5.5.2.4 Standard features

Clock programming: The Hub operates with the Memory Controller clock, as well as the clocks of all the peripheral modules that connect to the Hub. There is no separate clock for the Hub.

Reset-related issues: A partial reset of the HUB data transfer buffers is not possible on a global basis. Each peripheral device may use an Abort at the DTL-DMA interface to clear transactions that may be pending within the data transfer buffers for that peripheral.

Register programming guidelines: The PMA Security sandbox register settings must be initialized by software since the default is for all peripherals to use sandbox #1. It is recommended that the sandboxes be organized so that each processor (e.g. MIPS, TriMedia1 and TriMedia2) and the peripherals that are associated with each processor have access to a separate memory region. The memory regions may overlap if there is a need to share I/O buffers with more than one processor. Note that there are separate enable bits for “read” access and “write” access, allowing one processor and its peripherals to fully access a memory region; while another processor may only have “read” access to shared buffer or data space.

The fourth sandbox may be used for special purposes, such as inter-process communication buffers and semaphores, etc.

5.5.3 Register descriptions

5.5.3.1 Global registers

Table 270: HUB external (global) registers

Symbol	Clock domain	Suggested reset value	Description
Transaction Pipelining Control			
par_cmd_limit_pman1[3:0]	clk_mem	0x4	Pipelining Limit for Any Commands in PMA1 0 = no limit 1 = one transaction ...
par_rd_limit_pman2[3:0]	clk_mem	0x3	Pipelining Limit for Read Commands in PMA2 0 = no limit 1 = one transaction ...
par_wr_limit_pman2[3:0]	clk_mem	0x2	Pipelining Limit for Write Commands in PMA2 0 = no limit 1 = one transaction ...
Transaction Routing Control			
par_addr_mask[7:0]	clk_mem	0xC0	Enables monitoring of Address bits for transaction routing. If mask[7] is set, address[31] will be monitored and so on.
par_addr_value[7:0]	clk_mem	0x40	Defines value of monitored address bits. If enabled bits match bits in this register, transactions will be routed to PMA2.
IP-IP streaming control			
par_vivo_stream_enable	clk_mem	0x0	Enables streaming between VIP channel #1 and VO1 channel #1.
par_vivo_stream_rst	clk_mem	0x0	Aborts all outstanding streaming transactions between VIP and VO1. (Signal is single cycle strobe)

5.5.3.2 PMA Arbiter registers

The MTL ID addresses for PMAN1 are shown in Table 269. PMA2 is using a fixed round robin arbitration scheme and has no programmable registers.

PMAN1 Arbiter register summary

Table 271: Register summary

Offset	Name	Description
0x0	TDMA_A_0	Entry 1 of TDMA Timing Wheel (Set A)
0x4	TDMA_A_1	Entry 2 of TDMA Timing Wheel (Set A)
0x8	TDMA_A_2	Entry 3 of TDMA Timing Wheel (Set A)
0xC	TDMA_A_3	Entry 4 of TDMA Timing Wheel (Set A)
0x10	TDMA_A_4	Entry 5 of TDMA Timing Wheel (Set A)
0x14	TDMA_A_5	Entry 6 of TDMA Timing Wheel (Set A)
0x18	TDMA_A_6	Entry 7 of TDMA Timing Wheel (Set A)
0x1C	TDMA_A_7	Entry 8 of TDMA Timing Wheel (Set A)
0x200	PRIORITY_A_0	Entry1 of Priority List Note: Offset 0x200 has the highest priority.
0x204	PRIORITY_A_1	Entry 2 of Priority List (Set A)Note: Offset 0x200 has the highest priority.

Table 271: Register summary *…continued*

Offset	Name	Description
0x208	PRIORITY_A_2	Entry 3 of Priority List (Set A)Note: Offset 0x200 has the highest priority.
0x20C	PRIORITY_A_3	Entry 4 of Priority List (Set A)Note: Offset 0x200 has the highest priority.
0x210	PRIORITY_A_4	Entry 5 of Priority List (Set A)Note: Offset 0x200 has the highest priority.
0x214	PRIORITY_A_5	Entry 6 of Priority List (Set A)Note: Offset 0x200 has the highest priority.
0x218	PRIORITY_A_6	Entry 7 of Priority List (Set A)Note: Offset 0x200 has the highest priority.
0x21C	PRIORITY_A_7	Entry 8 of Priority List (Set A)Note: Offset 0x200 has the highest priority.
0x280	round_robin_1_A_0	Entry 1 of Round Robin List #1 (Set A)
0x284	round_robin_1_A_1	Entry 2 of Round Robin List #1 (Set A)
0x288	round_robin_1_A_2	Entry3 of Round Robin List #1 (Set A)
0x28C	round_robin_1_A_3	Entry 4 of Round Robin List #1 (Set A)
0x290	round_robin_1_A_4	Entry 5 of Round Robin List #1 (Set A)
0x294	round_robin_1_A_5	Entry 6 of Round Robin List #1 (Set A)
0x298	round_robin_1_A_6	Entry 7 of Round Robin List #1 (Set A)
0x29C	round_robin_1_A_7	Entry 8 of Round Robin List #1 (Set A)
0x300	round_robin_2_A_0	Entry 1 of Round Robin List #2 (Set A)
0x300	round_robin_2_A_1	Entry 2 of Round Robin List #2 (Set A)
0x304	round_robin_2_A_2	Entry 3 of Round Robin List #2 (Set A)
0x308	round_robin_2_A_3	Entry 4 of Round Robin List #2 (Set A)
0x30C	round_robin_2_A_4	Entry 5 of Round Robin List #2 (Set A)
0x310	round_robin_2_A_5	Entry 6 of Round Robin List #2 (Set A)
0x314	round_robin_2_A_6	Entry 7 of Round Robin List #2 (Set A)
0x318	round_robin_2_A_7	Entry 8 of Round Robin List #2 (Set A)
0x31C	round_robin_2_A_8	Entry 9 of Round Robin List #2 (Set A)
0x320	round_robin_2_A_9	Entry 10 of Round Robin List #2 (Set A)
0x324	round_robin_2_A_10	Entry 11 of Round Robin List #2 (Set A)
0x328	round_robin_2_A_11	Entry 12 of Round Robin List #2 (Set A)
0x32C	round_robin_2_A_12	Entry 13 of Round Robin List #2 (Set A)
0x330	round_robin_2_A_13	Entry 14 of Round Robin List #2 (Set A)
0x334	round_robin_2_A_14	Entry 15 of Round Robin List #2 (Set A)
0x338	round_robin_2_A_15	Entry 16 of Round Robin List #2 (Set A)
0x400	TDMA_B_0	Entry 1 of TDMA Timing Wheel (Set B)
0x404	TDMA_B_1	Entry 2 of TDMA Timing Wheel (Set B)
0x408	TDMA_B_2	Entry 3 of TDMA Timing Wheel (Set B)
0x40C	TDMA_B_3	Entry 4 of TDMA Timing Wheel (Set B)
0x410	TDMA_B_4	Entry 5 of TDMA Timing Wheel (Set B)
0x414	TDMA_B_5	Entry 6 of TDMA Timing Wheel (Set B)

Table 271: Register summary *…continued*

Offset	Name	Description
0x418	TDMA_B_6	Entry 7 of TDMA Timing Wheel (Set B)
0x41C	TDMA_B_7	Entry 8 of TDMA Timing Wheel (Set B)
0x600	PRIORITY_B_0	Entry 1 of Priority List (Set B)Note: Offset 0x200 has the highest priority.
0x604	PRIORITY_B_1	Entry 2 of Priority List (Set B)Note: Offset 0x200 has the highest priority.
0x608	PRIORITY_B_2	Entry 3 of Priority List (Set B)Note: Offset 0x200 has the highest priority.
0x60C	PRIORITY_B_3	Entry 4 of Priority List (Set B)Note: Offset 0x200 has the highest priority.
0x610	PRIORITY_B_4	Entry 5 of Priority List (Set B)Note: Offset 0x200 has the highest priority.
0x614	PRIORITY_B_5	Entry 6 of Priority List (Set B)Note: Offset 0x200 has the highest priority.
0x618	PRIORITY_B_6	Entry 7 of Priority List (Set B)Note: Offset 0x200 has the highest priority.
0x61C	PRIORITY_B_7	Entry 8 of Priority List (Set B)Note: Offset 0x200 has the highest priority.
0x680	round_robin_1_B_0	Entry 1 of Round Robin List #1 (Set B)
0x684	round_robin_1_B_1	Entry 2 of Round Robin List #1 (Set B)
0x688	round_robin_1_B_2	Entry 3 of Round Robin List #1 (Set B)
0x68C	round_robin_1_B_3	Entry 4 of Round Robin List #1 (Set B)
0x690	round_robin_1_B_4	Entry 5 of Round Robin List #1 (Set B)
0x694	round_robin_1_B_5	Entry 6 of Round Robin List #1 (Set B)
0x698	round_robin_1_B_6	Entry 7 of Round Robin List #1 (Set B)
0x69C	round_robin_1_B_7	Entry 8 of Round Robin List #1 (Set B)
0x700	round_robin_2_B_0	Entry 1 of Round Robin List #2 (Set B)
0x704	round_robin_2_B_1	Entry 2 of Round Robin List #2 (Set B)
0x708	round_robin_2_B_2	Entry 3 of Round Robin List #2 (Set B)
0x70C	round_robin_2_B_3	Entry 4 of Round Robin List #2 (Set B)
0x710	round_robin_2_B_4	Entry 5 of Round Robin List #2 (Set B)
0x714	round_robin_2_B_5	Entry 6 of Round Robin List #2 (Set B)
0x718	round_robin_2_B_6	Entry 7 of Round Robin List #2 (Set B)
0x71C	round_robin_2_B_7	Entry 8 of Round Robin List #2 (Set B)
0x720	round_robin_2_B_8	Entry 9 of Round Robin List #2 (Set B)
0x724	round_robin_2_B_9	Entry 10 of Round Robin List #2 (Set B)
0x728	round_robin_2_B_10	Entry 11 of Round Robin List #2 (Set B)
0x72C	round_robin_2_B_11	Entry 12 of Round Robin List #2 (Set B)
0x730	round_robin_2_B_12	Entry 13 of Round Robin List #2 (Set B)
0x734	round_robin_2_B_13	Entry 14 of Round Robin List #2 (Set B)
0x738	round_robin_2_B_14	Entry 15 of Round Robin List #2 (Set B)
0x73C	round_robin_2_B_15	Entry 16 of Round Robin List #2 (Set B)
0x800	NR_entries_A	NR_ENTRIES_A (Set A)

Table 271: Register summary *…continued*

Offset	Name	Description
0x804	NR_entries_B	NR_ENTRIES_B (Set B)
0x900	Control	
0x904	Status	
0xFFC	ARBITER_Module_ID	

PMAN1 Arbiter registers

Table 272: PMAN1_ARBITER registers

Bit	Symbol	Access	Reset value	Description
Offset 0x0 - TDMA_A_0				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_tdma_a_0[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	Theses bits must be written as zero.
4:0	Agent_id_tdma_a_0[4:0]	R/W	0xXX	ID of the agent that gets granted in this entry.
Offset 0x4 - TDMA_A_1				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_tdma_a_1[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	Theses bits must be written as zero.
4:0	Agent_id_tdma_a_1[4:0]	R/W	0xXX	ID of the agent that gets granted in this entry.
Offset 0x8 - TDMA_A_2				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_tdma_a_2[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	Theses bits must be written as zero.
4:0	Agent_id_tdma_a_2[4:0]	R/W	0xXX	ID of the agent that gets granted in this entry.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
Offset 0xC - TDMA_A_3				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_tdma_a_3[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	Theses bits must be written as zero.
4:0	Agent_id_tdma_a_3[4:0]	R/W	0xXX	ID of the agent that gets granted in this entry.
Offset 0x10 - TDMA_A_4				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_tdma_a_4[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	Theses bits must be written as zero.
4:0	Agent_id_tdma_a_4[4:0]	R/W	0xXX	ID of the agent that gets granted in this entry.
Offset 0x14 - TDMA_A_5				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_tdma_a_5[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	Theses bits must be written as zero.
4:0	Agent_id_tdma_a_5[4:0]	R/W	0xXX	ID of the agent that gets granted in this entry.
Offset 0x18 - TDMA_A_6				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
9:8	read_tdma_a_6[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	Theses bits must be written as zero.
4:0	Agent_id_tdma_a_6[4:0]	R/W	0xXX	ID of the agent that gets granted in this entry.
Offset 0x1C - TDMA_A_7				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_tdma_a_7[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	Theses bits must be written as zero.
4:0	Agent_id_tdma_a_7[4:0]	R/W	0xXX	ID of the agent that gets granted in this entry.
Offset 0x200 - PRIORITY_A_0				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_priority_a_0[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_priority_a_0[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x204 - PRIORITY_A_1				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_priority_a_1[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
4:0	Agent_id_priority_a_1[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x208 - PRIORITY_A_2				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_priority_a_2[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_priority_a_2[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x20C - PRIORITY_A_3				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_priority_a_3[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_priority_a_3[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x210 - PRIORITY_A_4				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_priority_a_4[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_priority_a_4[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x214 - PRIORITY_A_5				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
9:8	read_priority_a_5[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_priority_a_5[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x218 - PRIORITY_A_6				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_priority_a_6[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_priority_a_6[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x21C - PRIORITY_A_7				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_priority_a_7[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_priority_a_7[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x280 - round_robin_1_A_0				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_first_1_a_0[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
4:0	Agent_id_first_1_a_0[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x284 - round_robin_1_A_1				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_first_1_a_1[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_first_1_a_1[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x288 - round_robin_1_A_2				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_first_1_a_2[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_first_1_a_2[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x28C - round_robin_1_A_3				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_first_1_a_3[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_first_1_a_3[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x290 - round_robin_1_A_4				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
9:8	read_first_1_a_4[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_first_1_a_4[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x294 - round_robin_1_A_5				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_first_1_a_5[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_first_1_a_5[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x298 - round_robin_1_A_6				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_first_1_a_6[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_first_1_a_6[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x29C - round_robin_1_A_7				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_first_1_a_7[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
4:0	Agent_id_first_1_a_7[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x300 - round_robin_2_A_0				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_a_0[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_last_2_a_0[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x304 - round_robin_2_A_1				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_a_1[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_last_2_a_1[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x308 - round_robin_2_A_2				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_a_2[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_last_2_a_2[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x30C - round_robin_2_A_3				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
9:8	read_last_2_a_3[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_last_2_a_3[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x310 - round_robin_2_A_4				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_a_4[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_last_2_a_4[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x314 - round_robin_2_A_5				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_a_5[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_last_2_a_5[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x318 - round_robin_2_A_6				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_a_6[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
4:0	Agent_id_last_2_a_6[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x31C - round_robin_2_A_7				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_a_7[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_last_2_a_7[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x320 - round_robin_2_A_8				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_a_8[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_last_2_a_8[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x324 - round_robin_2_A_9				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_a_9[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_last_2_a_9[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x328 - round_robin_2_A_10				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
9:8	read_last_2_a_10[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_last_2_a_10[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x32C - round_robin_2_A_11				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_a_11[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_last_2_a_11[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x330 - round_robin_2_A_12				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_a_12[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_last_2_a_12[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x334 - round_robin_2_A_13				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *…continued*

Bit	Symbol	Access	Reset value	Description
9:8	read_last_2_a_13[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_last_2_a_13[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x338 - round_robin_2_A_14				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_a_14[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_last_2_a_14[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x33C - round_robin_2_A_15				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_a_15[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_last_2_a_15[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x400 - TDMA_B_0				
31:10	Reserved[21:0]	R	0xXXXXXX	these bits must be written as zero.

Table 272: PMAN1_ARBITER registers *…continued*

Bit	Symbol	Access	Reset value	Description
9:8	read_tdma_b_0[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_tdma_b_0[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x404 - TDMA_B_1				
31:10	Reserved[21:0]	R	0xXXXXXX	these bits must be written as zero.
9:8	read_tdma_b_1[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_tdma_b_1[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x408 - TDMA_B_2				
31:10	Reserved[21:0]	R	0xXXXXXX	these bits must be written as zero.
9:8	read_tdma_b_2[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_tdma_b_2[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x40C - TDMA_B_3				
31:10	Reserved[21:0]	R	0xXXXXXX	these bits must be written as zero.
9:8	read_tdma_b_3[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
4:0	Agent_id_tdma_b_3[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x410 - TDMA_B_4				
31:10	Reserved[21:0]	R	0xXXXXXX	these bits must be written as zero.
9:8	read_tdma_b_4[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_tdma_b_4[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x414 - TDMA_B_5				
31:10	Reserved[21:0]	R	0xXXXXXX	these bits must be written as zero.
9:8	read_tdma_b_5[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_tdma_b_5[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x418 - TDMA_B_6				
31:10	Reserved[21:0]	R	0xXXXXXX	these bits must be written as zero.
9:8	read_tdma_b_6[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_tdma_b_6[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x41C - TDMA_B_7				
31:10	Reserved[21:0]	R	0xXXXXXX	these bits must be written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
9:8	read_tdma_b_7[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_tdma_b_7[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x600 - PRIORITY_B_0				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_priority_b_0[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_priority_b_0[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x604 - PRIORITY_B_1				
31:10	Reserved2[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_priority_b_1[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_priority_b_1[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x608 - PRIORITY_B_2				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_priority_b_2[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *…continued*

Bit	Symbol	Access	Reset value	Description
4:0	Agent_id_priority_b_2[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x60C - PRIORITY_B_3				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_priority_b_3[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_priority_b_3[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x610 - PRIORITY_B_4				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_priority_b_4[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_priority_b_4[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x614 - PRIORITY_B_5				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_priority_b_5[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_priority_b_5[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x618 - PRIORITY_B_6				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
9:8	read_priority_b_6[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_priority_b_6[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x61C - PRIORITY_B_7				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_priority_b_7[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_priority_b_7[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x680 - round_robin_1_B_0				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_first_1_b_0[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_first_1_b_0[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x684 - round_robin_1_B_1				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_first_1_b_1[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
4:0	Agent_id_first_1_b_1[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x688 - round_robin_1_B_2				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_first_1_b_2[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_first_1_b_2[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x68C - round_robin_1_B_3				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_first_1_b_3[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_first_1_b_3[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x690 - round_robin_1_B_4				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_first_1_b_4[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_first_1_b_4[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x694 - round_robin_1_B_5				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
9:8	read_first_1_b_5[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_first_1_b_5[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x698 - round_robin_1_B_6				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_first_1_b_6[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_first_1_b_6[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x69C - round_robin_1_B_7				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_first_1_b_7[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits must be written as zero.
4:0	Agent_id_first_1_b_7[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x700 - round_robin_2_B_0				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_b_0[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits are written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
4:0	Agent_id_last_2_b_0[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x704 - round_robin_2_B_1				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_b_1[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits are written as zero.
4:0	Agent_id_last_2_b_1[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x708 - round_robin_2_B_2				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_b_2[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits are written as zero.
4:0	Agent_id_last_2_b_2[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x70C - round_robin_2_B_3				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_b_3[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits are written as zero.
4:0	Agent_id_last_2_b_3[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x710 - round_robin_2_B_4				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
9:8	read_last_2_b_4[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits are written as zero.
4:0	Agent_id_last_2_b_4[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x714 - round_robin_2_B_5				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_b_5[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits are written as zero.
4:0	Agent_id_last_2_b_5[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x718 - round_robin_2_B_6				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_b_6[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits are written as zero.
4:0	Agent_id_last_2_b_6[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x71C - round_robin_2_B_7				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_b_7[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits are written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
4:0	Agent_id_last_2_b_7[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x720 - round_robin_2_B_8				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_b_8[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits are written as zero.
4:0	Agent_id_last_2_b_8[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x724 - round_robin_2_B_9				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_b_9[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits are written as zero.
4:0	Agent_id_last_2_b_9[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x728 - round_robin_2_B_10				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_b_10[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits are written as zero.
4:0	Agent_id_last_2_b_10[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x72C - round_robin_2_B_11				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *…continued*

Bit	Symbol	Access	Reset value	Description
9:8	read_last_2_b_11[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits are written as zero.
4:0	Agent_id_last_2_b_11[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x730 - round_robin_2_B_12				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_b_12[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits are written as zero.
4:0	Agent_id_last_2_b_12[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x734 - round_robin_2_B_13				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_b_13[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits are written as zero.
4:0	Agent_id_last_2_b_13[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x738 - round_robin_2_B_14				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.

Table 272: PMAN1_ARBITER registers *...continued*

Bit	Symbol	Access	Reset value	Description
9:8	read_last_2_b_14[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits are written as zero.
4:0	Agent_id_last_2_b_14[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x73C - round_robin_2_B_15				
31:10	Reserved[21:0]	R	0xXXXXXX	These bits must be written as zero.
9:8	read_last_2_b_15[1:0]	R/W	0xX	Grant on read, write or both. 0x0 : grant independent whether it is a read or write 0x1 : grant only if it is a read 0x2 : grant only if it is a write 0x3 : reserved. 0: grant independent whether it is a read or write 1: grant only if it is a read 10: grant only if it is a write 11: reserved
7:5	Reserved2[2:0]	R	0xX	These bits are written as zero.
4:0	Agent_id_last_2_b_15[4:0]	R/W	0xXX	ID of the agent that is identified by this entry.
Offset 0x800 - NR_entries_A				
31:28	Reserved[3:0]	R	0x0	Must be written as zero.
27:24	last_round_robin_entries_a[3:0]	R/W	0x0	Number of valid entries in last round robin list Programming any value > SIZE_LAST_RR_LIST will result in use of the full round-robin list.
23:21	Reserved2[2:0]	R	0x0	Must be written as zero.
20:16	first_round_robin_entries_a[4:0]	R/W	0x00	Number of valid entries in first round robin list Programming any value > SIZE_FIRST_RR_LIST will result in use of the full round-robin list.
15:13	Reserved3[2:0]	R	0x0	Must be written as zero.
12:8	Priority_entries_a[4:0]	R/W	0x00	Number of valid entries in priority list Programming any value > SIZE_PRIO_LIST will result in use of full priority list.
7:0	Tdma_entries_a[7:0]	R/W	0x00	Number of valid entries in TDMA wheel Programming any value > 0x080 will result in use of all 128 entries.
Offset 0x804 - NR_entries_B				
31:28	Reserved[3:0]	R	0x0	Must be written as zero.
27:24	last_round_robin_entries_b[3:0]	R/W	0x0	Number of valid entries in last round robin list Programming any value > SIZE_LAST_RR_LIST will result in use of the full round-robin list.

Table 272: PMAN1_ARBITER registers *…continued*

Bit	Symbol	Access	Reset value	Description
23:21	Reserved2[2:0]	R	0x0	Must be written as zero.
20:16	first_round_robin_entries_b[4:0]	R/W	0x00	Number of valid entries in first round robin list Programming any value > SIZE_FIRST_RR_LIST will result in use of the full round-robin list.
15:13	Reserved3[2:0]	R	0x0	Must be written as zero.
12:8	Priority_entries_b[4:0]	R/W	0x00	Number of valid entries in priority list Programming any value > SIZE_PRIO_LIST will result in use of full priority list.
7:0	Tdma_entries_b[7:0]	R/W	0x00	Number of valid entries in TDMA wheel Programming any value > 0x080 will result in use of all 128 entries.
Offset 0x900 - Control				
31:2	Reserved[29:0]	R	0x00000000	Must be written as zero.
1:0	Request_Mode[1:0]	R/W	0x0	Indicates request to put arbiter in a certain mode 0x0 : boot mode 0x1 : normL operation mode using set A 0x2 : operation mode using set B 0x3: reserved. 0: Boot mode 1: Normal operation mode uses register set A. 10: Operation mode using set B. 11: Reserved
Offset 0x904 - Status				
31:2	Reserved[29:0]	R	0xXXXXXXXX	Unused
1:0	Functional_Mode[1:0]	R	0x0	Indicates functional mode of the arbiter 0x0 : boot mode 0x1: normal operation mode using set A 0x2 : operation mode using set B 0x3 : reserved. 0: Boot mode 1: Normal operation mode uses register set A. 10: Operation mode using set B. 11: Reserved
Offset 0xFFC - ARBITER_Module_ID				
31:16	Module_id_PMAN_ARBITER[15:0]	R	0x1010	Arbiter module ID number
15:12	Major_revision[3:0]	R	0x0	
11:8	Minor_revision[3:0]	R	0x0	
7:0	Aperture[7:0]	R	0x00	4 kB aperture size

5.5.3.3 PMA Monitor registers

The MTL ID addresses for PMAN1 are shown in Table 269. The PMAN2 MTL ID addresses are unreliable, for further information see Section 8.3.2.

Table 273 shows the Columbus Monitor MTL ID Addresses

Table 273: Columbus monitor MTL ID addresses

MTL ID	R/W	Columbus filter
1	write	3D comb filter
2	write	3D comb filter
3	write	SWAN/LORE filter
4	write	SWAN/LORE filter
5	read	3D comb filter
6	read	3D comb filter
7	read	SWAN/LORE filter
8	read	SWAN/LORE filter

PMAN1 Monitor register summary

Table 274: Register summary

Offset	Name	Description
0x0	CTL	control Register
0x4	CTRL	Current TimestampRegister - Low/High. Contains the total number of MTL clock cycles in the current or previous sample. The upper 8 bits of the value are stopped in the H version of the register
0x8	CTRH	Current TimestampRegister - Low/High. Contains the total number of MTL clock cycles in the current or previous sample. The upper 8 bits of the value are stopped in the H version of the register
0xC	TCRL	Transaction CountRegister - Low/High. Contains the total number of read or write transactions during the current or previous sample.
0x10	TCRH	Transaction CountRegister - Low/High. Contains the total number of read or write transactions during the current or previous sample.
0x14	SUML	Measurement Sum Register - Low/High. Contains either latency or bandwidth accumulations. Given this value, bandwidth can be calculated by dividing SUM by CTR. Average latency can be determined by dividing SUM by TCR.
0x18	SUMH	Measurement Sum Register - Low/High. Contains either latency or bandwidth accumulations. Given this value, bandwidth can be calculated by dividing SUM by CTR. Average latency can be determined by dividing SUM by TCR.
0x1C	MAX	Maximum Latency Value of current or previous sample
0xFD8	int_clr_enable	Interrupt clear enable
0xFDC	int_set_enable	Interrupt set enable
0xFE0	int_status	Interrupts status
0xFE4	int_enable	Interrupt enable
0xFE8	int_clr_status	Interrupt clear status
0xFEC	int_set_status	Interrupt set status
0xFF0	soft_rst	Software reset register

Table 274: Register summary *…continued*

Offset	Name	Description
0xFF4	disable_mmio	Software reset register
0xFF8	ext_module_id	Reserved
0xFFC	MONITOR_Module_ID	

PMAN1 Monitor registers

Table 275: PMAN1_MONITOR registers

Bit	Symbol	Access	Reset value	Description
Offset 0x0 - CTL				
31	Enable	R/W	0x0	Measurements taken when active.
30	Reset	R/W	0x0	Soft reset for monitor. Does not reset interrupt registers. Must be set then cleared.
29:16	LTV[13:0]	R/W	0x0000	Latency Threshool Value. If a latency sample surpasses this value, an interrupt is set. (A value of zero will not trigger an interrupt).
15:8	ID[7:0]	R/W	0x00	MTL ID field. Indicates the MTL node in the downstream MTL hierarchy to analyse if bit3 (ALL) is inactive. The width of this value depends on the ID width parameter.
7:6	Reserved[1:0]	R/W	0x0	Ignore upon read. Write as zeroes.
5:4	LMC[1:0]	R/W	0x0	Latency Measurement Criteria. All latency measurements start at cmd_VALID and complete based on this value. 00: start of first valid field data. 01: completion of data payload indicated by RD_LAST or WR_LAST. 10 and 11: completion of data payload indicated by RD_LAST (for read payloads) and WS_DONE (for write payloads)
3	ALL	R/W	0x0	If active, measure the entire MTL network from the monitor's perspective. Else, monitor a specific node of the network given the subsequent ID field.
2:1	TRANS[1:0]	R/W	0x0	Transaction type: 00: none 10: write only 01: read only 11: both
0	CRITERIA	R/W	0x0	Monitoring mode 0 : bandwidth 1: latency.

Table 275: PMAN1_MONITOR registers *…continued*

Bit	Symbol	Access	Reset value	Description
Offset 0x4 - CTRL				
31:0	CTRL_32[31:0]	R/W	0x00000000	Contains the lower 32 bits of the total number of MTL clock cycles in the current or previous sample. Writing to this register is not recommended.
Offset 0x8 - CTRH				
31:8	Reserved[23:0]	R/W	0x000000	Ignore upon read. Write as zeroes.
7:0	CTRH_32[7:0]	R/W	0x00	Contains the upper 8 bits of the total number of MTL clock cycles in the current or previous sample. Writing to this register is not recommended
Offset 0xC - TCRL				
31:0	TCRL_32[31:0]	R	0x00000000	Contains the lower 32 bits of the total number of read or write transactions during the current or previous sample. They register only updates in latency mode.
Offset 0x10 - TCRH				
31:8	Reserved[23:0]	R	0x000000	Ignore upon reads
7:0	TCRH_32[7:0]	R	0x00	Contains the upper 8 bits of the total number of read or write transactions during the current or previous sample. This register only updates in latency mode.
Offset 0x14 - SUML				
31:0	SUML_32[31:0]	R/W	0x00000000	Contains the lower 32 bits of either latency or bandwidth accumulations. Writing to this register is not recommended
Offset 0x18 - SUMH				
31:8	Reserved[23:0]	R/W	0x000000	Ignore upon reads. Write as zeroes.
7:0	SUMH_32[7:0]	R/W	0x00	Contains the upper 8 bits of either latency or bandwidth accumulations. Writing to this register is not recommended.
Offset 0x1C - MAX				
31:14	Reserved[17:0]	R	0x00000	Ignore upon reads
13:0	MAX_latency[13:0]	R	0x0000	Contains the maximum latency value of the current or previous sample. The register only updates in latency mode.
Offset 0xFD8 - int_clr_enable				
31:4	Reserved[27:0]	W	0x0000000	Write as zeroes.
3	sum_ovrflw_clr_en	W	0x0	SUM overflow clear enable register 1: SUM overflow interrupt enable is cleared 0: SUM overflow interrupt enable is unchanged.

Table 275: PMAN1_MONITOR registers *…continued*

Bit	Symbol	Access	Reset value	Description
2	ltv_excded_clr_en	W	0x0	Latency threshold value exceeded clear enable register 1: Latency threshold value exceeded interrupt enable is cleared 0: Latency threshold value exceeded interrupt enable is unchanged.
1	fifo_ovrflw_clr_en	W	0x0	FIFO overflow clear enable register 1: FIFO overflow interrupt enable is cleared 0: FIFO overflow interrupt enable is unchanged.
0	ts_ovrflw_clr_en	W	0x0	Timestamp overflow clear enable register 1: Timestamp overflow interrupt enable is cleared 0: Timestamp overflow interrupt enable is unchanged.
Offset 0xFDC - int_set_enable				
31:4	Reserved[27:0]	W	0x0000000	Write as zeroes
3	sum_ovrflw_set_en	W	0x0	SUM overflow set enable register 1: SUM overflow interrupt enable is set 0: SUM overflow interrupt enable is unchanged.
2	ltv_excded_set_en	W	0x0	Latency threshold value exceeded set enable register 1: Latency threshold value exceeded interrupt enable is set 0: Latency threshold value exceeded interrupt enable is unchanged.
1	fifo_ovrflw_set_en	W	0x0	FIFO overflow set enable register 1: FIFO overflow interrupt enable is set 0: FIFO overflow interrupt enable is unchanged.
0	ts_ovrflw_set_en	W	0x0	Timestamp overflow set enable register 1: Timestamp overflow interrupt enable is set 0: Timestamp overflow interrupt enable is unchanged.
Offset 0xFE0 - int_status				
31:4	Reserved[27:0]	R	0x0000000	Ignore upon reads.
3	sum_ovrflw_stat	R	0x0	SUM overflow interrupt status 1: Interrupt pending 0: Interrupt not pending.
2	ltv_excded_stat	R	0x0	Latency threshold value exceeded interrupt status 1: Interrupt pending 0: Interrupt not pending.
1	fifo_ovrflw_stat	R	0x0	FIFO overflow interrupt status 1: Interrupt pending 0: Interrupt not pending.
0	ts_ovrflw_stat	R	0x0	Timestamp overflow interrupt status 1: Interrupt pending 0: Interrupt not pending.
Offset 0xFE4 - int_enable				
31:4	Reserved[27:0]	R	0x0000000	Ignore upon reads.
3	sum_ovrflw_en	R	0x0	SUM overflow enable register 1: SUM overflow interrupt is enabled 0: SUM overflow interrupt is disabled
2	ltv_excded_en	R	0x0	Latency threshold value exceeded enable register 1: Latency threshold value exceeded interrupt is enabled 0: Latency threshold value exceeded interrupt is disabled
1	fifo_ovrflw_en	R	0x0	FIFO overflow enable register 1: FIFO overflow interrupt is enabled 0: FIFO overflow interrupt is disabled

Table 275: PMAN1_MONITOR registers *…continued*

Bit	Symbol	Access	Reset value	Description
0	ts_ovrflw_en	R	0x0	Timestamp overflow enable register 1: Timestamp overflow interrupt is enabled 0: Timestamp overflow interrupt is disabled
Offset 0xFE8 - int_clr_status				
31:4	Reserved[27:0]	W	0x0000000	Write as zeroes.
3	sum_ovrflw_clr_stat	W	0x0	SUM overflow clear status register 1: SUM overflow interrupt is cleared 0: SUM overflow interrupt is unchanged.
2	ltv_excded_clr_stat	W	0x0	Latency threshold value exceeded clear status register 1: Latency threshold value exceeded interrupt is cleared 0: Latency threshold value exceeded interrupt is unchanged.
1	fifo_ovrflw_clr_stat	W	0x0	FIFO overflow clear status register 1: FIFO overflow interrupt is cleared 0: FIFO overflow interrupt is unchanged.
0	ts_ovrflw_clr_stat	W	0x0	Timestamp overflow clear status register 1: Timestamp overflow interrupt is cleared 0: Timestamp overflow interrupt is unchanged.
Offset 0xFEC - int_set_status				
31:4	Reserved[27:0]	W	0x0000000	Write as zeroes.
3	sum_ovrflw_set_stat	W	0x0	SUM overflow set status register 1: SUM overflow interrupt is set 0: SUM overflow interrupt is unchanged.
2	ltv_excded_set_stat	W	0x0	Latency threshold value exceeded set status register 1: Latency threshold value exceeded interrupt is set 0: Latency threshold value exceeded interrupt is unchanged.
1	fifo_ovrflw_set_stat	W	0x0	FIFO overflow set status register 1: FIFO overflow interrupt is set 0: FIFO overflow interrupt is unchanged.
0	ts_ovrflw_set_stat	W	0x0	Timestamp overflow set status register 1: Timestamp overflow interrupt is set 0: Timestamp overflow interrupt is unchanged.
Offset 0xFF0 - soft_rst				
31:1	Reserved[30:0]	R/W	0x00000000	Write as zeroes.
0	soft_reset	R/W	0x0	Software reset register. This bit is self clearing. 1: software reset is active 0: software reset inactive.
Offset 0xFF4 - disable_mmio				
31:1	Reserved[30:0]	R/W	0x00000000	Write as zeroes.
0	disable_mmio_register	R/W	0x0	Disable MMIO register. While the interface is disabled, this is the only valid register. 1: MMIO interface is disabled 0: MMIO interface is enabled

Table 275: PMAN1_MONITOR registers *…continued*

Bit	Symbol	Access	Reset value	Description
Offset 0xFF8 - ext_module_id				
31:0	Reserved[31:0]	R/W	0x00000000	Unused
Offset 0xFFC - MONITOR_Module_ID				
31:16	Module_id_PMAN_ MONITOR[15:0]	R	0xA077	
15:12	Major_revision[3:0]	R	0x1	
11:8	Minor_revision[3:0]	R	0x0	
7:0	Module_aperture_ size[7:0]	R	0x00	

5.5.3.4 PMA1 & PMA2 security registers

Each Memory Security Block has its own 4k MMIO aperture. Assignment of the system address is done outside of the Hub; addresses are therefor shown as a relative offset within the assigned memory aperture.

Register Summary

Table 276: PMA1 & PMA2 security register summary

Offset	Symbol	Description
0x0	Protection_Error_Address	Protection error address
0x4	Register_Write_Protect	Register write protect
0x80	Sandbox_1_lower	Sandbox 1 lower
0x84	Sandbox_1_upper	Sandbox 1 upper
0x88	Sandbox_2_lower	Sandbox 2 lower
0x8C	Sandbox_2_upper	Sandbox 2 upper
0x90	Sandbox_3_lower	Sandbox 3 lower
0x94	Sandbox_3_upper	Sandbox 3 upper
0x98	Sandbox_4_lower	Sandbox 4 lower
0x9C	Sandbox_4_upper	Sandbox 4 upper
0x100	Sandbox_Enable_VO1	Sandbox assignment ID0 [1]
0x104	Sandbox_Enable_VO2	Sandbox assignment ID1 [1]
0x108	Sandbox_Enable_MBS_read	Sandbox assignment ID2 [1]
0x10C	Sandbox_Enable_MBS_write	Sandbox assignment ID3 [1]
0x110	Sandbox_Enable_VIP	Sandbox assignment ID4 [1]
0x114	Sandbox_Enable_VMPG	Sandbox assignment ID5 [1]
0x118	Sandbox_Enable_North_tunnel	Sandbox assignment ID6 [1]
0x11C	Sandbox_Enable_Memory_gatel	Sandbox assignment ID7 [1]
0xFE0	Interrupt_Status	MSEC interrupt status
0xFE4	Interrupt_Enable	MSEC interrupt enable
0xFE8	Interrupt_Clear	MSEC interrupt clear
0xFEC	Interrupt_Set	MSEC interrupt set
0xFFC	Module_ID	

[1] Refer to Table 269 for ID assignment.

Register table

Table 277: PMAN security registers

Bit	Symbol	Access	Value	Description
Offset 0x0 - Protection_Error_Address				
31:3	Stat_addr[31:3]	R	0x00000000	address of transaction causing protection error
2:0	Stat_addr[2:0]	R	0x0	fixed
Offset 0x4 - Register_Write_Protect				
31:1	Reserved[30:0]	R	0x00000000	
0	Write_protect	R/W	0x0	writing a one into this register disables write access to the PSEC module (except IRQ registers). Once set, this bit can only be cleared by a hardware reset.
Offset 0x80 - Sandbox_1_lower				
31:16	Sb1_base[27:16]	R/W	0x0000	lower limit of accessible space (this address is included)
15:0	Sb1_base[15:0]	R	0x0000	fixed
Offset 0x84 - Sandbox_1_upper				
31:16	Sb1_top[27:16]	R/W	0xFFFF	upper limit of accessible space (this address is included)
15:0	Sb1_top[15:0]	R	0xFFFF	fixed
Offset 0x88 - Sandbox_2_lower				
31:16	Sb2_base[27:16]	R/W	0x0000	lower limit of accessible space (this address is included)
15:0	Sb2_base[15:0]	R	0x0000	fixed
Offset 0x8C - Sandbox_2_upper				
31:16	Sb2_top[27:16]	R/W	0xFFFF	upper limit of accessible space (this address is included)
15:0	Sb2_top[15:0]	R	0xFFFF	fixed
Offset 0x90 - Sandbox_3_lower				
31:16	Sb3_base[27:16]	R/W	0x0000	lower limit of accessible space (this address is included)
15:0	Sb3_base[15:0]	R	0x0000	fixed
Offset 0x94 - Sandbox_3_upper				
31:16	Sb3_top[27:16]	R/W	0xFFFF	upper limit of accessible space (this address is included)
15:0	Sb3_top[15:0]	R	0xFFFF	fixed
Offset 0x98 - Sandbox_4_lower				
31:16	Sb4_base[27:16]	R/W	0x0000	lower limit of accessible space (this address is included)
15:0	Sb4_base[15:0]	R	0x0000	fixed
Offset 0x9C - Sandbox_4_upper				
31:16	Sb4_top[27:16]	R/W	0xFFFF	upper limit of accessible space (this address is included)
15:0	Sb4_top[15:0]	R	0xFFFF	fixed
Offset 0x100 - Sandbox_Enable_VO1				

Table 277: PMAN security registers *…continued*

Bit	Symbol	Access	Value	Description
31:8	Reserved[23:0]	R	0x00000 0	RESERVED
7	Sb4_vo1_wen	R/W	0x0	Sandbox 4 Write enable
6	Sb4_vo1_ren	R/W	0x0	Sandbox 4 Read enable
5	Sb3_vo1_wen	R/W	0x0	Sandbox 3 Write enable
4	Sb3_vo1_ren	R/W	0x0	Sandbox 3 Read enable
3	Sb2_vo1_wen	R/W	0x0	Sandbox 2 Write enable
2	Sb2_vo1_ren	R/W	0x0	Sandbox 2 Read enable
1	Sb1_vo1_wen	R/W	0x1	Sandbox 1 Write enable
0	Sb1_vo1_ren	R/W	0x1	Sandbox 1 Read enable
Offset 0x104 - Sandbox_Enable_VO2				
31:8	Reserved[23:0]	R	0x00000 0	RESERVED
7	Sb4_vo2_wen	R/W	0x0	Sandbox 4 Write enable
6	Sb4_vo2_ren	R/W	0x0	Sandbox 4 Read enable
5	Sb3_vo2_wen	R/W	0x0	Sandbox 3 Write enable
4	Sb3_vo2_ren	R/W	0x0	Sandbox 3 Read enable
3	Sb2_vo2_wen	R/W	0x0	Sandbox 2 Write enable
2	Sb2_vo2_ren	R/W	0x0	Sandbox 2 Read enable

Table 277: PMAN security registers *…continued*

Bit	Symbol	Access	Value	Description
1	Sb1_vo2_wen	R/W	0x1	Sandbox 1 Write enable
0	Sb1_vo2_ren	R/W	0x1	Sandbox 1 Read enable
Offset 0x108 - Sandbox_Enable_MBS_read				
31:8	Reserved[23:0]	R	0x000000	RESERVED
7	Sb4_mbs_read_wen	R/W	0x0	Sandbox 4 Write enable
6	Sb4_mbs_read_ren	R/W	0x0	Sandbox 4 Read enable
5	Sb3_mbs_read_wen	R/W	0x0	Sandbox 3 Write enable
4	Sb3_mbs_read_ren	R/W	0x0	Sandbox 3 Read enable
3	Sb2_mbs_read_wen	R/W	0x0	Sandbox 2 Write enable
2	Sb2_mbs_read_ren	R/W	0x0	Sandbox 2 Read enable
1	Sb1_mbs_read_wen	R/W	0x1	Sandbox 1 Write enable
0	Sb1_mbs_read_ren	R/W	0x1	Sandbox 1 Read enable
Offset 0x10C - Sandbox_Enable_MBS_write				
31:8	Reserved[23:0]	R	0x000000	Reserved
7	Sb4_mbs_write_wen	R/W	0x0	Sandbox 4 Write enable
6	Sb4_mbs_write_ren	R/W	0x0	Sandbox 4 Read enable
5	Sb3_mbs_write_wen	R/W	0x0	Sandbox 3 Write enable

Table 277: PMAN security registers *...continued*

Bit	Symbol	Access	Value	Description
4	Sb3_mbs_write_ren	R/W	0x0	Sandbox 3 Read enable
3	Sb2_mbs_write_wen	R/W	0x0	Sandbox 2 Write enable
2	Sb2_mbs_write_ren	R/W	0x0	Sandbox 2 Read enable
1	Sb1_mbs_write_wen	R/W	0x1	Sandbox 1 Write enable
0	Sb1_mbs_write_ren	R/W	0x1	Sandbox 1 Read enable
Offset 0x110 - Sandbox_Enable_VIP				
31:8	Reserved[23:0]	R	0x000000	Reserved
7	Sb4_vip_wen	R/W	0x0	Sandbox 4 Write enable
6	Sb4_vip_ren	R/W	0x0	Sandbox 4 Read enable
5	Sb3_vip_wen	R/W	0x0	Sandbox 3 Write enable
4	Sb3_vip_ren	R/W	0x0	Sandbox 3 Read enable
3	Sb2_vip_wen	R/W	0x0	Sandbox 2 Write enable
2	Sb2_vip_ren	R/W	0x0	Sandbox 2 Read enable
1	Sb1_vip_wen	R/W	0x1	Sandbox 1 Write enable
0	Sb1_vip_ren	R/W	0x1	Sandbox 1 Read enable
Offset 0x114 - Sandbox_Enable_VMPG				
31:8	Reserved[23:0]	R	0x000000	Sandbox 4 Write enable

Table 277: PMAN security registers *…continued*

Bit	Symbol	Access	Value	Description
7	Sb4_vmpg_wen	R/W	0x0	Sandbox 4 Write enable
6	Sb4_vmpg_ren	R/W	0x0	Sandbox 4 Read enable
5	Sb3_vmpg_wen	R/W	0x0	Sandbox 3 Write enable
4	Sb3_vmpg_ren	R/W	0x0	Sandbox 3 Read enable
3	Sb2_vmpg_wen	R/W	0x0	Sandbox 2 Write enable
2	Sb2_vmpg_ren	R/W	0x0	Sandbox 2 Read enable
1	Sb1_vmpg_wen	R/W	0x1	Sandbox 1 Write enable
0	Sb1_vmpg_ren	R/W	0x1	Sandbox 1 Read enable
Offset 0x118 - Sandbox_Enable_North_tunnel				
31:8	Reserved[23:0]	R	0x000000	Reserved
7	Sb4_north_tunnel_wen	R/W	0x0	Sandbox 4 Write enable
6	Sb4_north_tunnel_ren	R/W	0x0	Sandbox 4 Read enable
5	Sb3_north_tunnel_wen	R/W	0x0	Sandbox 3 Write enable
4	Sb3_north_tunnel_ren	R/W	0x0	Sandbox 3 Read enable
3	Sb2_north_tunnel_wen	R/W	0x0	Sandbox 2 Write enable
2	Sb2_north_tunnel_ren	R/W	0x0	Sandbox 2 Read enable
1	Sb1_north_tunnel_wen	R/W	0x1	Sandbox 1 Write enable

Table 277: PMAN security registers *...continued*

Bit	Symbol	Access	Value	Description
0	Sb1_north_tunnel_ren	R/W	0x1	Sandbox 1 Read enable
Offset 0x11C - Sandbox_Enable_Memory_gatel				
31:8	Reserved[23:0]	R	0x000000	Reserved
7	Sb4_memory_gatel_wen	R/W	0x0	Sandbox 4 Write enable
6	Sb4_memory_gatel_ren	R/W	0x0	Sandbox 4 Read enable
5	Sb3_memory_gatel_wen	R/W	0x0	Sandbox 3 Write enable
4	Sb3_memory_gatel_ren	R/W	0x0	Sandbox 3 Read enable
3	Sb2_memory_gatel_wen	R/W	0x0	Sandbox 2 Write enable
2	Sb2_memory_gatel_ren	R/W	0x0	Sandbox 2 Read enable
1	Sb1_memory_gatel_wen	R/W	0x1	Sandbox 1 Write enable
0	Sb1_memory_gatel_ren	R/W	0x1	Sandbox 1 Read enable
Offset 0xFE0 - Interrupt_Status				
31:9	Reserved[22:0]	R	0x000000	Reserved
8	Stat_read	R	0x0	protection error caused by read
7:3	Stat_id[4:0]	R	0x00	ID of initiator causing protection error
2:1	Stat_sid[1:0]	R	0x0	sub-ID of initiator causing protection error
0	Stat_prot_err	R	0x0	Memory Access Protection Error
Offset 0xFE4 - Interrupt_Enable				
31:1	Reserved[30:0]	R/W	0x00000000	
0	Ien_prot_err	R/W	0x0	Memory Access Protection Error
Offset 0xFE8 - Interrupt_Clear				
31:1	Reserved[30:0]	R/W	0x00000000	
0	Clr_prot_err	W	0x0	Memory Access Protection Error

Table 277: PMAN security registers *…continued*

Bit	Symbol	Access	Value	Description
Offset 0xFEC - Interrupt_Set				
31:1	Reserved[30:0]	R/W	0x00000000	
0	Set_prot_err	W	0x0	Memory Access Protection Error
Offset 0xFFC - Module_ID				
31:16	Mod_id[15:0]	R	0xA056	Module ID unique 16-bit code
15:12	Rev_major[3:0]	R	0x0	Major revision counter
11:8	Rev_minor[3:0]	R	0x0	Minor revision counter
7:0	App_size[7:0]	R	0x00	Aperture Size

5.5.3.5 Reset registers

Reset register summary

Table 278: Register summary

Offset	Name	Description
0x0	RST_CTL	
0x4	RST_CAUSE	Note: This register is set on every write to RST_CTL register or watchdog timeout or RESET_IN.
0x8	EN_WATCHDOG_RST	
0xFFC	Reset_MODULE_ID	

Reset registers

Table 279: Reset registers

Bit	Symbol	Access	Reset value	Description
Offset 0x0 - RST_CTL				
31:4	Unused[27:0]	R/W	0x0000000	
3	Rel_mips_rst_n	W	0x0	0: No Action 1: Release MIPS reset 0: No action 1: Release MIPS Reset.
2	Do_sw_rst	W	0x0	0: No action 1: Do software reset 0: No action 1: Do Software Reset.
1	Rel_sys_rst_out	W	0x0	0: 0: No action 1: Release System Reset of External Peripherals.

Table 279: Reset registers *…continued*

Bit	Symbol	Access	Reset value	Description
0	Assert_sys_rst_out	W	0x0	0: No action 1: Do system reset of external peripherals. 0: No action 1: Do System Reset of External Peripherals.
Offset 0x4 - RST_CAUSE				
31:2	Unused[29:0]	R/W	0x00000000	
1:0	Rst_cause_register[1:0]	R	0x0	Reset Cause register: 00: Cause is external system reset, RESET_IN. 01: Cause is software system reset. 11: Cuase is watchdog timeout. 0: Cause is External System Reset, RESET_IN. 1: Cause is Software System Reset. 11: Cause is watchdog timeout.
Offset 0x8 - EN_WATCHDOG_RST				
31:1	Unused[30:0]	R/W	0x00000000	
0	En_watchdog_rst_register	R/W	0x1	Enable Watchdog Reset register:0: Disable reset due to watchdog timeout 1: Enable reset upon watchdog timeout. 0: Disable reset due to watchdog timeout. 1: Enable reset upon watchdog timeout.
Offset 0xFFC - Reset_MODULE_ID				
31:16	Module_id_reset_block[15:0]	R	0x0123	Reset Module ID: 0x0123
15:12	Majrev[3:0]	R	0x0	Major revision = 0
11:8	Minrev[3:0]	R	0x0	Minor revision = 0
7:0	Apersize[7:0]	R	0x00	Aperture: multiple of 4k bytes

5.6 Pixel formats

5.6.1 Introduction

Several hardware subsystems in the PNX2015 deal directly with images. Such hardware subsystems follow the same memory representation and interpretation of images in order to successfully pass images between subsystems.

The PNX2015 uses the following pixel format strategy:

- A limited number of native pixel formats are supported by all image subsystems, as appropriate.

- The Memory Based Scaler supports conversion from arbitrary pixel formats to any native format during the anti-flicker filtering operation. This is a standard operation and no extra passes are introduced.
- Hardware subsystems support all native pixel formats in both little-endian and big-endian system operation.

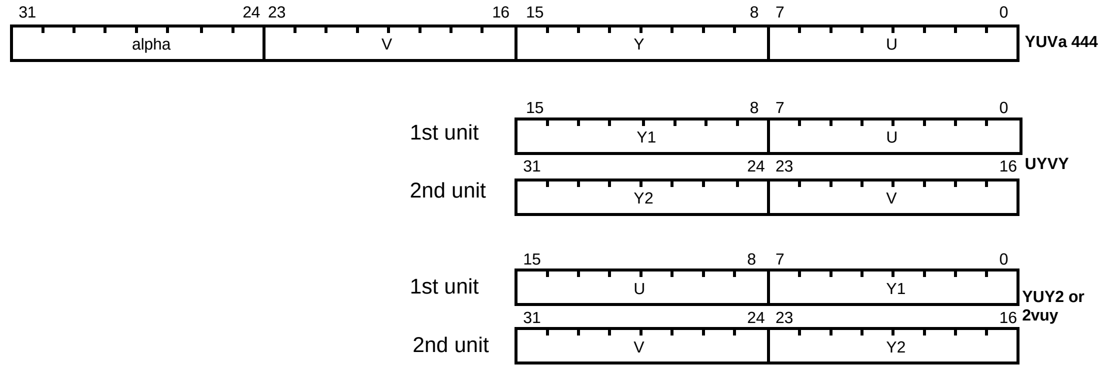
5.6.2 Summary of native pixel formats

Table 280 and Figure 185 summarize the native pixel formats and image hardware subsystems that support them.

Table 280: Native Pixel Format Summary

Name	Note	VIP Out	MBS In Out	VO In	VMPG Out
packed YUV 4:2:2 (UYVY)	16-bit unit, two successive units contain two horizontally adjacent pixels, no alpha	x	x	x	
packed YUV 4:2:2 (YUY2, 2vuy)		x	x	x	
planar YUV 4:2:2	Three arrays, one for each component	x	x		
semi-planar YUV 4:2:2	Two arrays, one with all Ys, one with U and Vs	x	x	x	
planar YUV 4:2:0	Three arrays, one for each component		x		
semi-planar YUV 4:2:0	Two arrays, one with all Ys, one with U and Vs		x	x	x

The layout shown in Figure 185 is the way that a unit ends up in a CPU register given a unit size (8, 16 or 32-bit) load operation, regardless of the PNX2015 endian mode of operation.



For planar YUV 4:2:0 formats, refer to Figure 190 and Figure 191

Fig 185.Native Pixel Format Unit Layout

5.6.3 Native pixel format representation

5.6.3.1 Packed YUV 4:2:2 formats

Packed YUV 4:2:2 formats store two horizontally adjacent pixels into two 16-bit units. Each pixel has an individual luminance (Y1 for the left of the pair, Y2 for the right of the pair). There is a single U and V value associated with the pair. The U and V values are taken from the same spatial position as the Y1 sample (Figure 189).

There are two variants of packed YUV 4:2:2: the Microsoft ‘UYVY’ format and the Microsoft ‘YUY2’ format. The big-endian view of the YUY2 format is identical to the Power MacIntosh ‘2vuy’ format.

Figure 186 and Figure 187 show the software view of UYVY and YUY2/2vuy. Two successive 16-bit units contain a pair of pixels. This view is independent of system endian mode.

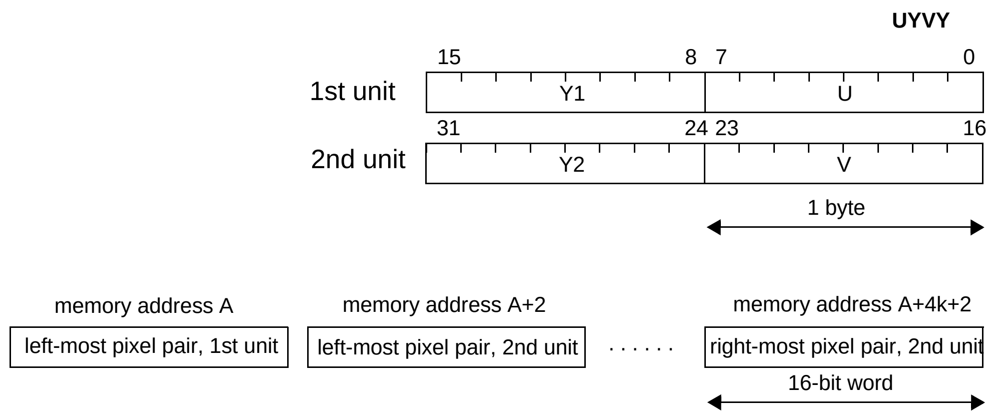


Fig 186.UYVY packed YUV 4:2:2 format

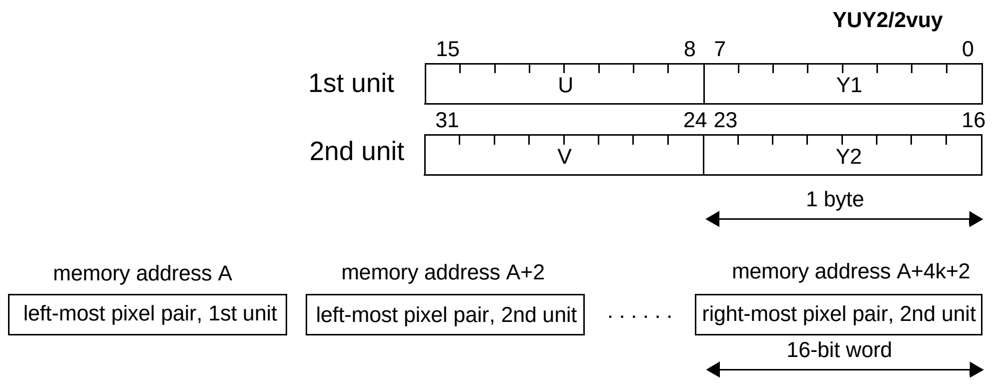


Fig 187.YUY2/2vuy packed YUV 4:2:2 format

5.6.3.2 Planar YUV 4:2:0 and YUV 4:2:2 formats

The spatial sampling structure of planar YUV 4:2:0 data is shown in Figure 188. The planar YUV 4:2:2 data has the spatial sampling structure as shown in Figure 189.

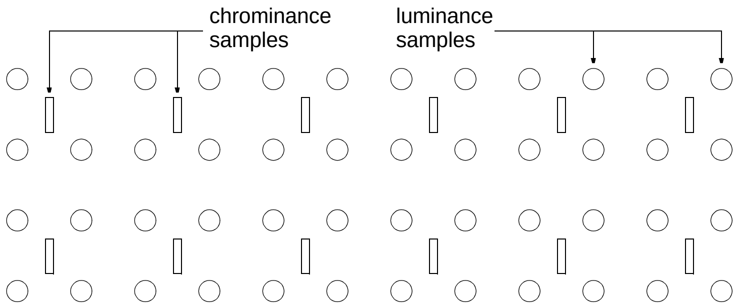


Fig 188.Spatial sampling structure of YUV 4:2:0 data

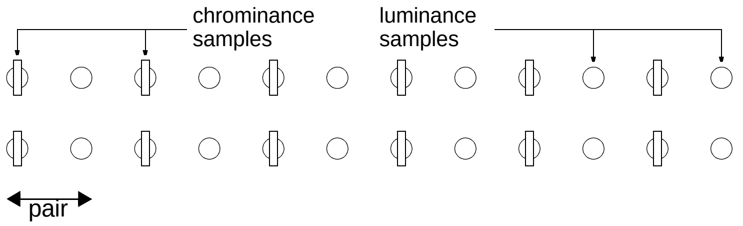


Fig 189.Spatial sampling structure of packed and planar YUV 4:2:2 data

Planar variants: There are two variants of planar YUV 4:2:0 and YUV 4:2:2 formats.

Planar or 3-plane format: An image is described by 3 pointer values (Py, Pu, Pv). Each pointer points to a 2D array of Y, U and V values as shown in Figure 190.

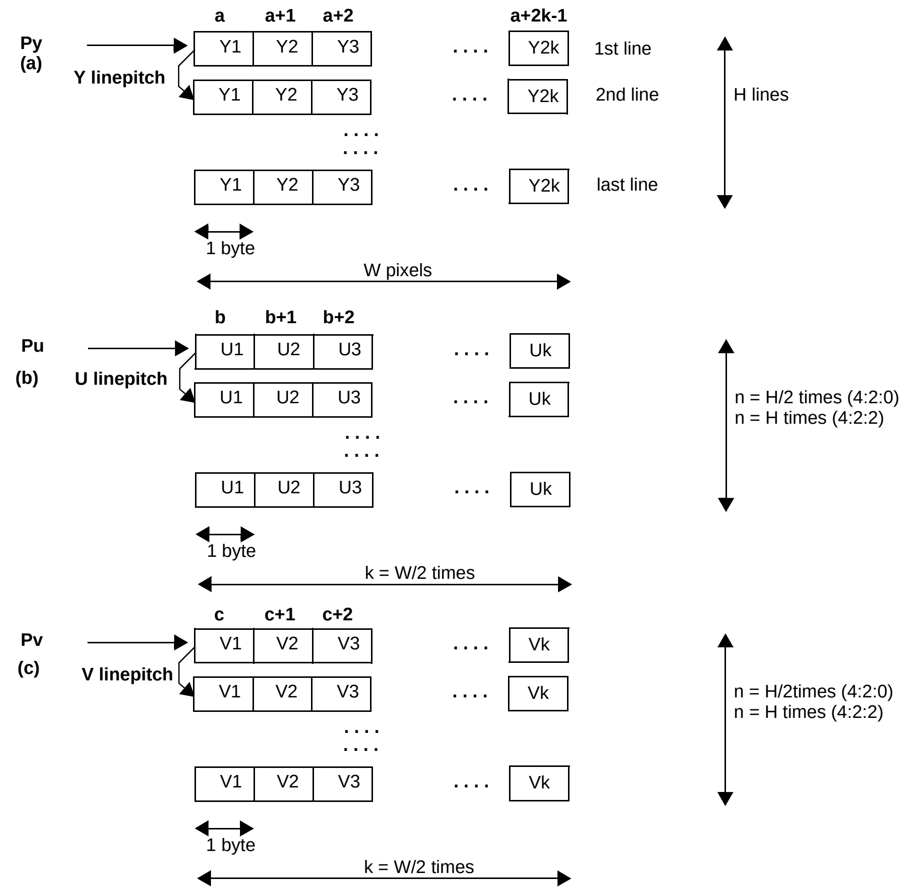


Fig 190.Planar YUV 4:2:0 and 4:2:2 Formats

Semi-planar or 2-plane format: An image is described by 2 pointer values (Py, Puv). The Y pointer points to a 2D array of Y values. The Puv pointer points to a 2D array of UV pair values. Note that the U value of a UV pair always has the lower byte address. See Figure 191.

The MBS supports all planar formats on input and output. The VIP can produce the planar and semi-planar YUV 4:2:2 planar formats. The semi-planar YUV 4:2:0 format is the only format produced by the VMPG.

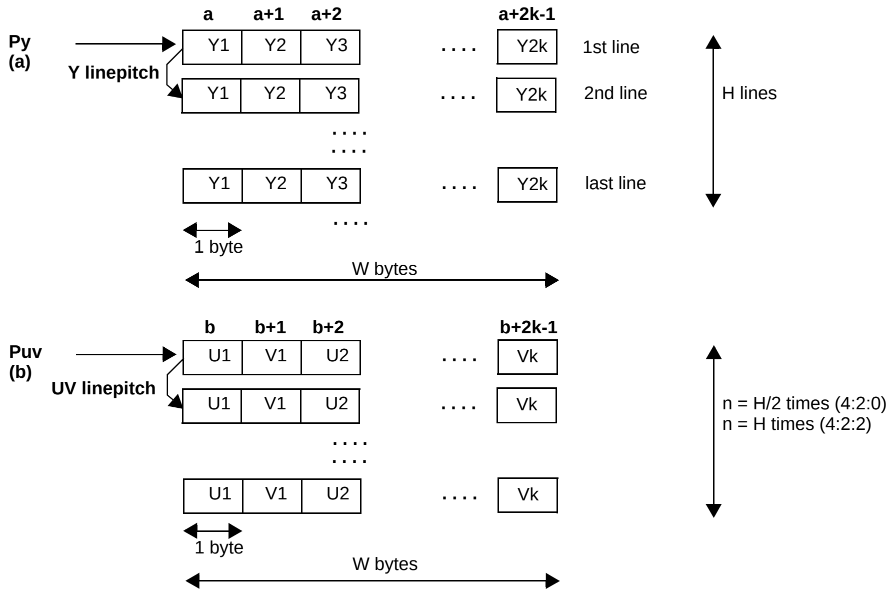


Fig 191.Semi-planar YUV 4:2:0 and YUV 4:2:2 formats

5.6.4 YUV values

5.6.4.1 8-Bit Data

For 8-bit data, the full range of values is allowed i.e., [0~255]. As an option, the data range may be clipped in the MBS or VIP according to the ITU-R BT. 601-4 specification.

- Y range [16~235]
- U range [16~240]
- V range [16~240]

5.6.5 Image storage format

With the exception of the planar formats, described in Section 5.6.3.2 , the layout of an image in memory is determined by the following elements:

- pixel format, which implies the unit size(s)
- origin pointer– the (byte) address of the first unit of the image
- line pitch– the address difference between a pixel on a line and a pixel directly below it
- width W, in number of pixels
- height H, in number of lines

Note that for indexed formats, each unit contains one or more pixels. For the packed formats, a unit is a pixel, with the exception of packed YUV 4:2:2 where two units are needed to describe a pixel pair.

Figure 192 shows how images are stored in memory.

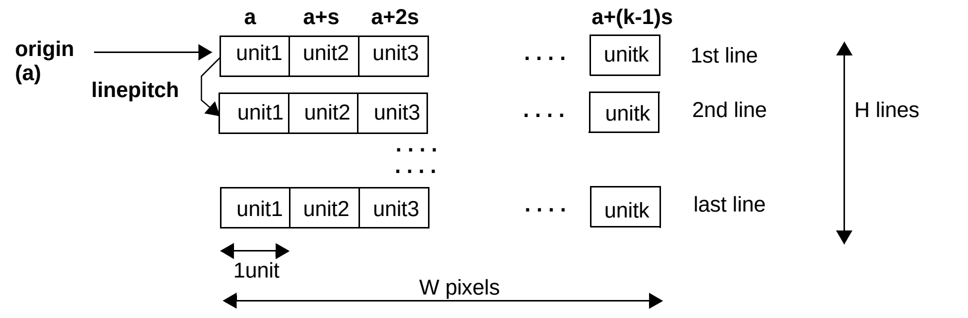


Fig 192.Image storage format

5.6.6 System endian mode

The PNX2015 is designed to run with either little-endian or big-endian software. This is determined by a global endianness flag.

The endian mode determines how a multi-byte value is stored to, or loaded from, memory byte addresses.

For the native pixel formats, Section 5.6.3 two elements are shown in the figures: the layout of a ‘unit’, which is always 8, 16 or 32 bits, and the mapping of adjacent units to memory byte addresses. These two elements are always maintained, independent of system endian mode.

What this implies is that each hardware subsystem needs to map a unit to memory byte addresses in an endian mode-dependent manner. The rules are as follows:

- Storing a 16-bit unit to address ‘A’ results in modifying memory bytes ‘A’ and ‘A+1’.
- Storing a 32-bit unit to memory address ‘A’ results in modifying memory bytes ‘A’ to ‘A+3’ inclusive.
- In little-endian mode, the least significant bits of a unit go to the lowest byte address.
- In big-endian mode, the most significant bits of a unit go to the lowest byte address.
- Reading from left to right, adjacent units go to increasing memory addresses.

5.7 DDR SDRAM controller (also known as IP_2031)

5.7.1 Introduction

This section describes the DDR SDRAM Controller core. The DDR Controller is a multi-port memory controller for DDR SDRAM devices. The DDR core is part of a family that can be used to interface to off-chip DDR memory that suits the platform or application requirements.

For frequencies equal to or below 200Mhz use DDR, for frequencies above 200MHz use GDDR.

5.7.1.1 DDR configuration for PNX20155

The DDR SDRAM controller is a generic IP block which is configured for the PNX2015. The controller has several functions for which there are control registers that are not supported by the PNX2015 IC.

The PNX2015 configuration supports only one physical external DDR memory as shown in Fig 193, thus only one rank is supported. Only 16 bit external DDR memory is supported not 32 bit.

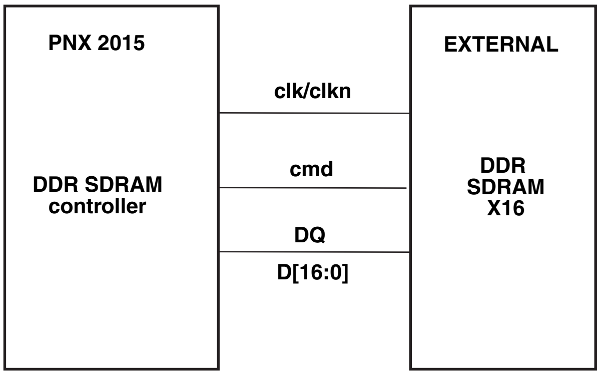


Fig 193.External DDR memory

5.7.1.2 Features

The primary features of the DDR SDRAM Controller include:

- 16-bit data bus width on DDR SDRAM memory side
- Four MTL ports
- Supports x8 and x16 memory devices
- Supports 64-Mbit, 128-Mbit and 256-Mbit DDR SDRAM memory devices
- Data width ratio between on-chip MTL data bus and off-chip DDR data bus is 2:1
- Supports 1 rank (physical bank) of memory
- Support for linear and two-dimensional MTL transactions
- Support for wrapping (linear address wrapping)
- Maximum of 8 open pages
- Maximum address range of 256 MB
- Halt modes to allow for power consumption reduction
- Programmable via a 32-bit DTL-MMIO port
- Independent DTL port and MTL port operating frequencies
- Programmable DDR SDRAM timing parameters that support DDR SDRAM memory devices up to 200 MHz

- Programmable bank mapping scheme
- Programmable arbitration

5.7.2 Functional description

5.7.2.1 DDR controller block level diagram

A functional description of the DDR SDRAM Controller is shown in Figure 194

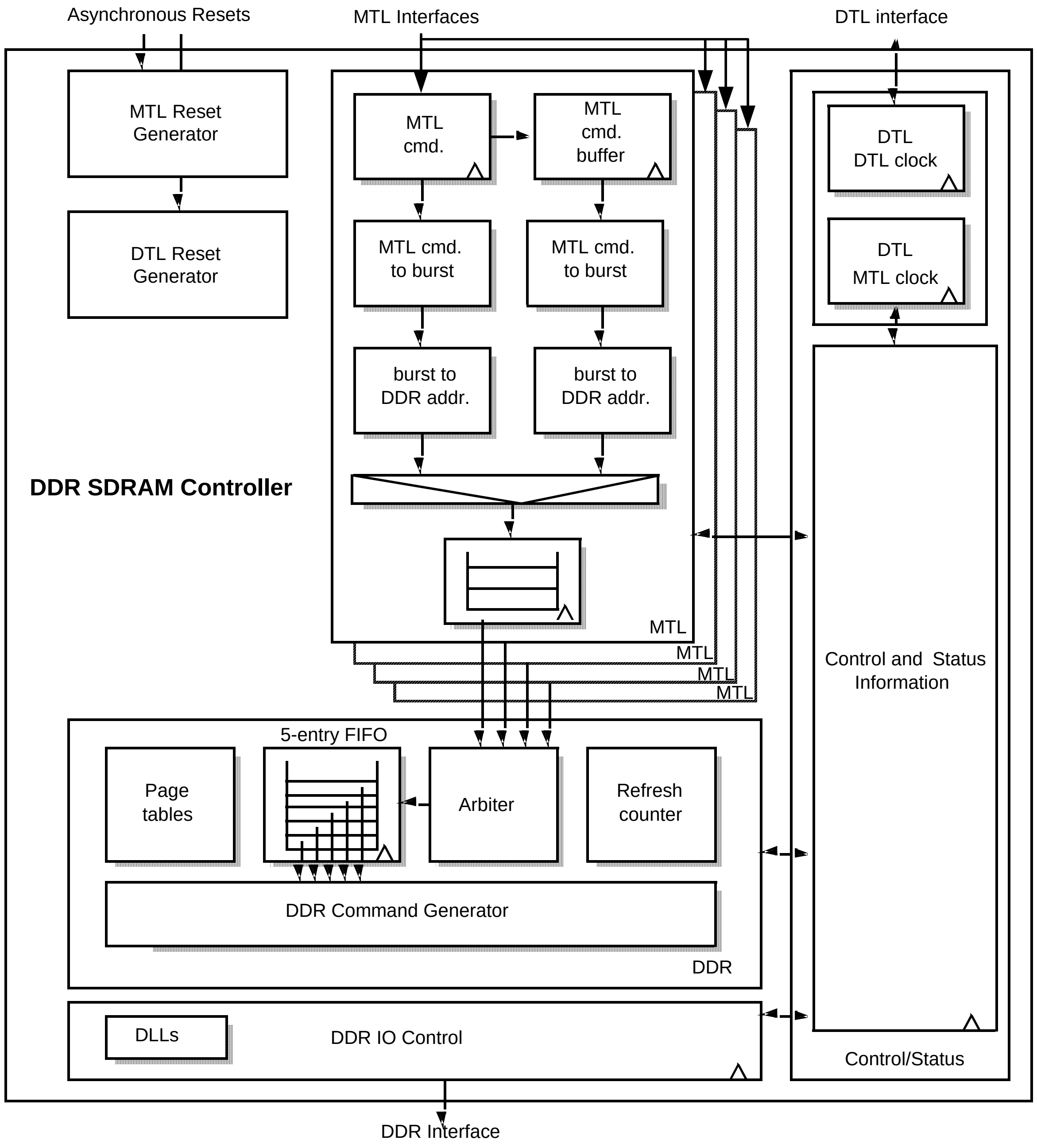


Fig 194.Block diagram of the DDR SDRAM controller

5.7.2.2 Architecture

The main functional blocks of the DDR controller are the MTL block, DDR block, DDR IO block, control/status block, and reset generator blocks.

The MTL block is instantiated once for each MTL port. It receives the MTL commands and is responsible for chopping up a MTL command into one or multiple DDR burst commands (taking into account the DDR burst size). For every DDR burst command an address mapping to DDR addresses (rank, bank, row, and column) is performed based on the address mapping information in the control/status block.

The DDR block includes an Arbiter which arbitrates between DDR burst commands from up to four different MTL blocks. After arbitration, the DDR burst command selected by the Arbiter is put in a 5 entry FIFO. The DDR block has a refresh counter to track refresh timing. The DDR block tracks the open pages in the DDR memories. Up to two DDR ranks (with 4 banks each) are supported resulting in a total of eight pages. The DDR command generator decides upon which command (refresh, precharge, activate, read, or write) to generate based on the information in the 5-entry FIFO, the state of the refresh counter, and the state of the DDR memories as indicated by the open page table.

The DDR IO block provides the interface to the DDR interface pads (the pads themselves are not part of the DDR SDRAM Controller). Dedicated sub-blocks provide read or write functionality for the interface signals.

The control/status block holds the MMIO registers, which are programmable through a DTL interface. The DDR SDRAM Controller DTL interface runs at clock “clk_dtl”, the MMIO registers run at clock “clk_mtl”. To transfer between the two clock domains, an asynchronous clock domain transfer block is included in the control/status block.

The reset generator blocks provide asynchronous resets for the two clock domains of the DDR controller, and ensure a synchronous de-assert of the reset signals.

5.7.2.3 Input processing

The data ports of the DDR SDRAM Controller comply with the MTL bus specification. This is a standard point-to-point bus that originates from DVP and is specifically designed to support SDRAM memory interfacing for high-performance systems. However, the memory bus specification is a bus protocol specification. It does not define the memory bus transactions that a memory bus target supports. Bus transactions might be one or two-dimensional. Sequential address wrapping is supported, XOR-style wrapping is not supported. Two-dimensional bus transactions with wrapping, perform sequential address wrapping on the individual lines of the transaction.

The MTL write accept signal “MTL_WR_ACCEPT” may come as soon as two cycles after the associated MTL command accept signal “MTL_CMD_ACCEPT” has been given.

5.7.2.4 Start

MMIO register IP_2031_CTL provides the interface to start the DDR controller. Typically, the DDR controller is not initialized or in halt mode when a DDR controller start action is triggered.

The START field of MMIO register IP_2031_CTL is used to start the DDR SDRAM Controller. Typically, the DDR controller is not yet initialized when this action is taken. MMIO registers that determine the characteristics of the DDR memories are programmed prior to the start action, since these register values may be used to configure the external

DDR memories. The normal sequence of actions to start the DDR controller (and to configure the DDR memories): apply a hard reset to the DDR controller, program the MMIO registers using the DTL MMIO interface, set the START field of MMIO register IP_2031_CTL to ‘1’. When the DDR controller finished initialization (of itself and the DDR memories), the output signal “ip_2031_initialized” are set to ‘1’. Configurable reset values are provided for certain MMIO register values. These can be used to reduce the amount of DTL MMIO transactions to program MMIO registers.

Sequence of actions: During start the DDR SDRAM Controller performs the following sequence of actions, in compliance with JEDEC DDR protocol.

- Apply a NOP command
- Precharge all command
- Load extended mode register
- Load mode register, with DLL reset
- 256 cycles delay for DDL
- Precharge all command
- Auto refresh command
- Load mode register, with DLL reset deactivated
- 256 cycles delay

5.7.2.5 Arbitration

The DDR SDRAM Controller provides up to four different MTL ports. The functionality provided is the same for all four MTL ports. However, for arbitration purposes a distinction between the different MTL ports is made. MTL port 0 is treated as a port for DMA traffic, MTL ports 1, 2, and 3 are treated as ports for CPU traffic.

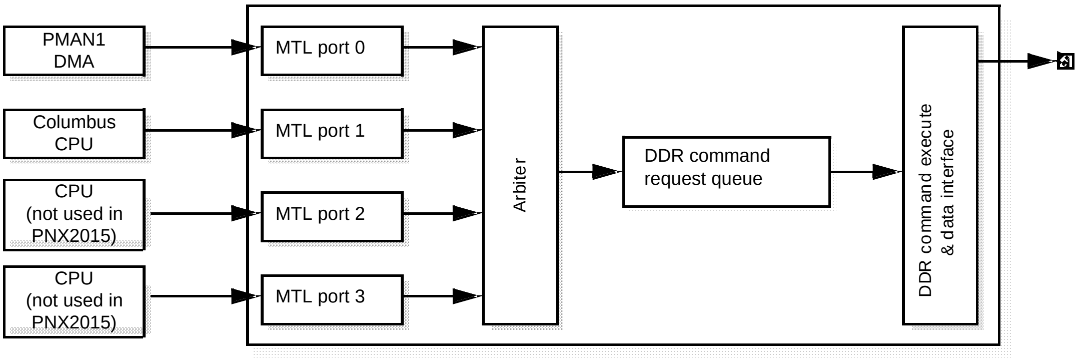


Fig 195. MTL ports of the DDR SDRAM controller

The DDR Controller Arbiter is responsible for scheduling between MTL transaction requests from the different MTL ports. Arbitration has been optimized for CPU latency and high DDR efficiency, but may be programmed to emphasize either.

The DDR Controller arbitration scheme is hierarchical. The first level arbitration selects either the DMA port or a CPU port. If a CPU port wins the first level arbitration, the second level arbitration selects which CPU port wins.

The following text describes the features available.

First Level Arbitration: Between DMA and CPUs: For the first level arbitration there are two mechanisms available: windows and account budgets.

Windows provide the basic means to allocate DDR bandwidth. A window is defined in terms of DDR controller clock cycles. Windows are defined for DMA traffic (HRT_WINDOW) and CPU traffic (CPU_WINDOW), and they alternate with each other in time. During an HRT_WINDOW, the DMA traffic is given priority by the arbitration scheme. During a CPU_WINDOW, the CPU traffic is given priority by the arbitration scheme.

As implied by the names CPU_WINDOW and HRT_WINDOW, windows have been introduced to divide DDR bandwidth between CPU traffic and Hard Real-Time (HRT) DMA traffic. Typically, in an SoC a third type of traffic is present as well: Soft Real_time (SRT) DMA traffic. This type of traffic usually has less hard real-time constraints than HRT DMA traffic i.e., the bandwidth requirements can be averaged over a much larger time period (several windows) than with HRT. However, it is still necessary to ensure that this type of traffic receives DDR memory bandwidth. To this end, a CPU account is introduced.

The CPU account limits (budgets) the memory bandwidth consumption by the CPU traffic to ensure that SRT DMA traffic receives enough memory bandwidth. The CPU account is shared by all CPU traffic (from MTL ports 1, 2, and 3) and is defined by CPU_RATIO, CPU_LIMIT, CPU_CLIP and CPU_DECR.

The value CPU_RATIO controls how much bandwidth the CPUs can get. The value CPU_LIMIT controls how many DDR bursts CPUs can take back-to-back before the CPUs are out of budget. The value CPU_CLIP controls how much debt the CPUs are allowed to build up. CPU_DECR is made programmable so that the accuracy of the accounting can be increased. This is especially needed when using dynamic ratios (see Section “Dynamic Ratios”).

When the internal account exceeds CPU_LIMIT, DMA traffic is given higher priority than CPU traffic, independent of which window is active. The internal account is a saturated counter, that is, it does not wrap around on an underflow or overflow. For every DDR controller memory clock cycle, the internal counter is decremented by CPU_DECR. Whenever a CPU DDR burst is started, the internal counter is incremented by an amount equal to the amount of data transfer cycles, plus the value of CPU_RATIO, except when a CPU MTL transaction is ‘for free’.

A CPU MTL transaction is for free if it starts while the account value is above the CPU_CLIP value [6]. If a DDR burst is for free, then the account gets incremented by an amount equal to the amount of data transfer cycles, without the CPU_RATIO. The CPU_CLIP value is always set equal, or higher than CPU_LIMIT, otherwise CPU_LIMIT would never be reached.

[6] If pre-empting of the MTL transaction is not allowed, then all DDR bursts from one MTL transaction are treated the same. So if the first DDR burst is (not) for free then the other DDR bursts for the same MTL transactions are also (not) for free. If pre-emption of the MTL transaction is allowed, then the ‘for free’ decision is made separately for each DDR burst

By means of the accounting mechanism, the CPU bandwidth can be budgeted. In the CPU_WINDOW a CPU normally has priority over DMA. For every clock cycle, the CPU account gets funded with CPU_DECR. For every CPU DDR burst, the costs of that burst, defined as CPU_RATIO plus data transfer cycles, are accounted for. When the CPU account runs out of budget (account value above CPU_LIMIT), then DMA gets priority over CPU.

If there is no DMA, then the CPUs can still get the bandwidth which they pay for by allowing the CPU account to borrow from its future budget. If there is a longer time period where there is no DMA traffic, the CPU account could potentially build up a large deficit. As soon as DMA traffic restarts, the CPUs could conceivably have an extended period of time where they have a lower priority than DMA (while paying off the debt). The CPU_CLIP value controls how much debt the CPU account is allowed to build up. After that value has been reached and there is still no DMA traffic the CPUs get the bandwidth for free. The number of data transfer cycles is accounted for to approximately (excluding overhead) get the same account value before and after the free transaction.

Second Level Arbitration: Among CPUs: After the CPUs have won the first level arbitration the second level arbitration decides which of the CPUs wins. For every CPU MTL port a separate CPU account is present, each with a CPUn_RATIO, CPUn_LIMIT, CPUn_CLIP and CPUn_DECR. These accounts function exactly the same as the generic CPU account described above. Except, of course, that each register set only accounts the transactions for one specific CPU port instead of any CPU port.

Whenever a CPU account is out of budget the associated CPU request has lower priority with respect to the CPU request whose accounts are not yet out of budget. When there is more than one CPU request pending and the budget situation does not force an arbitration decision (i.e. they are both within budget or out of budget) then the DDR is programmed to favor one over the other. To this end each CPU port has a HIGH priority bit, which makes that CPU higher priority than the CPUs which have their HIGH priority bit not set.

When there is more than one CPU request pending, and neither their budget situation nor their priority setting forces a decision (for example just after reset), the Arbiter decides based on an LRU (Least Recently Used) list which is maintained by the Arbiter.

The above mechanisms are implemented to make sure that the right CPU wins the arbitration in case there is more then one CPU request pending. In systems where there is relatively little CPU traffic and a lot of DMA traffic, it rarely happens i.e., most of the time there is none or only one CPU request pending at any given time.

If a CPU wins the first level arbitration, that CPU wins the arbitration even if it already out of budget. This means an out-of-budget CPU can use up the budget from the generic CPU account. This may prevent another CPU, which is not out of budget, from getting bandwidth. To solve this particular problem a VSDMA (versus DMA) bit has been added for each CPU port. With this bit set an out-of-budget CPU loses the arbitration to DMA (provided there is a DMA request).

Dynamic Ratios: The accounting mechanism described earlier is the static ratio variant. The problem with this approach is that the statically programmed CPU_RATIO that is used, per DDR burst, can not account for significantly different amounts of overhead by a DDR burst that can occur. To fix that problem dynamic ratios have been introduced, which can be enabled through the IP_2031_ARB_CTL register.

Whenever a CPU DDR burst is started with dynamic ratios, the internal account is incremented by an amount equal to “the number of clock cycles spent on the previous CPU DDR burst” multiplied by the CPU_RATIO. This way the real overhead is measured and accounted for and therefore the accounting mechanism is much more accurate.

With dynamic ratios enabled, the free bandwidth is also handled differently. When the bandwidth is for free i.e., the account is above the CPU_CLIP value, the internal account is not incremented. To ensure the internal account has the same value before and after the free bandwidth DDR burst, the account never decrements whenever a clock cycle is spent on a CPU DDR burst, even if the burst is not for free. To account for this the CPU_RATIO is set by the amount CPU_DECR lower as compared to the static ratios approach.

Pre-Emption: The arbitration scheme can be further fine tuned by specifying when arbitration is done. An MTL transaction is chopped up into one or more DDR bursts, as the Arbiter operates on DDR bursts. Typically, the arbitration is done on an MTL transaction basis; i.e., once an MTL transaction has been selected by the arbitration scheme, all of its DDR bursts are processed before a new MTL transaction is selected. This approach tries to maximize bandwidth efficiency by exploiting locality assumed to be present within an MTL transaction. However, it might increase the expected latency of some MTL transactions.

Assume there is a CPU MTL transaction present while doing arbitration in an HRT window (and no DMA MTL transaction present). The CPU MTL transaction is selected by the Arbiter. While the CPU transaction is being processed, a DMA MTL transaction becomes present (in the HRT window). The CPU MTL transaction is consuming HRT window bandwidth, while a DMA MTL transaction is waiting to be selected by the Arbiter. From an overall bandwidth point of view, finishing the CPU MTL transaction to completion will improve the CPU performance, but the programmed bandwidth partitioning is not fully applied. To address this issue, the concept of MTL transaction pre-emption is introduced.

MTL transaction pre-emption is programmable (via the MMIO register IP_2031_ARB_CTL) and can be used to interrupt an ongoing MTL transaction– before it is completed– to favor another MTL transaction. Pre-emption allows ongoing CPU MTL transactions to be interrupted by a DMA MTL transaction while in the HRT_WINDOW, and allows ongoing DMA MTL transactions to be interrupted by a CPU MTL transaction while in the CPU_WINDOW. Interruption of an MTL transaction of the same type never happens. Any interruption reduces the overall efficiency of the DDR Controller as it disallows exploiting locality assumed to be present within a MTL transaction.

The pre-emption field supports three different pre-emption settings. Table 281 describes the CPU pre-emption field.

Table 281: CPU Preemption Field

Preemption Field Value	Description
0b00	No preemption (once a CPU MTL command has started to enter the DDR arbitration buffer, it will go completely into the DDR arbitration buffer, uninterrupted by other (CPU or DMA) MTL commands).

Table 281: CPU Preemption Field *…continued*

Preemption Field Value	Description
0b01	Preempt a CPU MTL command as it starts to enter the DDR arbitration buffer while currently active in the DMA window. The CPU MTL command will only be interrupted by a DMA MTL command, not by another CPU MTL command.
0b10	Undefined
0b11	Preempt a CPU MTL command that is currently active in the DMA window (independent of when it started to enter the DDR arbitration buffer).The CPU MTL command will only be interrupted by a DMA MTL command, not by another CPU MTL command.

Back Log Buffer (BLB): The request for a DDR burst that wins the arbitration is always put in a FIFO queue. This FIFO is 5 levels deep to allow the DDR to look ahead and open and close pages in memory banks in order to increase DDR efficiency. Unfortunately this also means that a new high priority request that has immediately won the arbitration could possibly wait 5 full DDR bursts before it gets serviced. In a system in which almost all the available bandwidth is used (the FIFO is almost always full) this can significantly increase the latency.

Usually CPU traffic requires low latency and DMA traffic requires high bandwidth. In order to reduce latency for the CPUs, the back log buffer (BLB) has been implemented. When the BLB is enabled (through the IP_2031_ARB_CTL register), DMA DDR bursts that are in the FIFO can be temporarily moved to the BLB.

This is done under the following conditions:

- The FIFO entries hold a DMA DDR burst.
- No DDR burst of the same DMA MTL transaction has reached the top of the FIFO yet.
- The BLB is empty
- A CPU DDR burst request wins the arbitration.
- CPU traffic has higher priority than DMA traffic. (This is important in case the CPU wins arbitration, despite being lower priority than DMA, due only to the absence of DMA traffic.)

The BLB therefore allows the CPU transaction to overtake the DMA transaction already in the FIFO. Since the DDR Controller may have already opened/closed pages for the DMA DDR bursts, this feature reduces the DDR efficiency.

As soon as DMA requests start winning the arbitration again, the DMA DDR bursts from the BLB are read and put back in the FIFO until the BLB is empty.

PMAN (Hub) versus DDR Controller Interaction: An additional factor to consider is the interaction of the Hub and the DDR Controller. The DDR Controller command FIFO (pipeline) is 5 entries, however the PMAN only allows 3 transactions to be outstanding. This means that the other two FIFO stages can (and are) occupied by transactions from one of the CPUs. This can result in unexpected CPU bandwidth of up to 50%. This value is an extreme worst-case; a more realistic number (assuming some kind of video decoding) is around 15% of the gross DDR memory cycles.

Under the condition that the total required CPU budget is more than the maximum "leakage" of bandwidth it is possible to reduce the additional "leakage" (above and beyond budget) to zero by setting the value for CLIP = LIMIT + 2 * RATIO * <avg. transaction latency>.

The net result of this setting is that although "leakage" still occurs, it is charged against the budget and compensated for immediately after occurrence.

5.7.2.6 Addressing

The DDR SDRAM Controller performs address mapping of MTL addresses onto DDR memory rank, bank, row and column addresses. The 32-bit MTL addresses provided to the DDR controller cover a 4-GB address range. Of these 32-bit addresses, the upper four bits are ignored by the DDR controller, reducing the addressable range to 256 MB. Note that the DDR controller only supports up to 256 MB of DDR memory (either implemented by a single rank or two ranks of size 128 MB).

The following describes how the mapping of MTL addresses onto DDR addresses is performed.

There are three different parts to address mapping:

- Memory regions
- Memory region mapping scheme
- DDR memory rank locations

Although these three parts are treated independent of each other, their configuration (by means of programming MMIO registers) needs to be consistent to ensure correct operation of the DDR controller.

Memory Regions: The DDR SDRAM Controller supports two different regions in the addressable range of 256 MB. Region 1 is defined by means of MMIO registers IP_2031_DDR_REGION1_BASE and IP_2031_DDR_REGION1_MASK. Given an MTL address X, it is said to be located within region 1 when (X & IP_2031_DDR_REGION1_MASK) equals (IP_2031_DDR_REGION1_BASE & IP_2031_DDR_REGION1_MASK).

An MTL address located outside region 1 is said to be located in the default region. As a result, the default region and region 1 are mutually exclusive and partition the addressable range of 256 MB. Note that regions are defined in the MTL address domain, and are independent of DDR memory characteristics such as DDR memory row or column sizes or the amount of DDR memory ranks present. For region determination, the upper four bits of the MMIO region registers are ignored.

Memory Region Mapping Scheme: The default region and region 1 support their own mapping of MTL addresses onto DDR addresses. The type of MTL transactions and the mapping of MTL addresses onto DDR addresses heavily influences the DDR SDRAM Controller bandwidth efficiency. When possible, different types of MTL transactions are directed to different regions, whose mapping schemes are set to optimize DDR memory bandwidth efficiency for the MTL transaction types used to address the region. The default and region 1 mappings are defined by MMIO registers IP_2031_DDR_DEF_BANK_SWITCH and IP_2031_DDR_REGION1_BANK_SWITCH respectively. These MMIO registers allow the programmer to support different DRR bank, row, and column interleaving schemes. The MTL address to DDR address mapping can be illustrated by means of some examples. In all of these examples, a 32-bit DDR interface (a 32-bit/4-byte column) and a DDR burst length of 8 32-bit/4-byte elements (a full DDR burst transfers 8 * 4 bytes= 32 bytes) are assumed.

Example 1: 32-Byte Interleaving

In 32-byte interleaving mode, the mapping scheme changes the DDR bank every other 2^3 = 8 columns. The BANK_SWITCH field is programmed to 3.

Table 282: 32-Byte Interleaving, 256 Columns

MTL Address Range	Row Address	Bank Address	Column Address
0x000:0000-0x000:001f	0x0000	0b00	0x0000-0x0007
0x000:0020-0x000:003f	0x0000	0b01	0x0000-0x0007
0x000:0040-0x000:005f	0x0000	0b10	0x0000-0x0007
0x000:0060-0x000:007f	0x0000	0b11	0x0000-0x0007
0x000:0080-0x000:009f	0x0000	0b00	0x0008-0x000f
0x000:0fe0-0x000:0fff	0x0000	0b11	0x00f8-0x00ff
0x000:1000-0x000:101f	0x0001	0b00	0x0000-0x0007
0x000:1020-0x000:103f	0x0001	0b01	0x0000-0x0007
0x000:1fe0-0x000:1fff	0x0001	0b11	0x00f8-0x00ff
0x000:2000-0x000:201f	0x0002	0b00	0x0000-0x0007
0x000:2020-0x000:203f	0x0002	0b01	0x0000-0x0007

Table 283: 32-Byte Interleaving, 512 Columns

MTL Address Range	Row Address	Bank Address	Column Address
0x000:0000-0x000:001f	0x0000	0b00	0x0000-0x0007
0x000:0020-0x000:003f	0x0000	0b01	0x0000-0x0007
0x000:0040-0x000:005f	0x0000	0b10	0x0000-0x0007
0x000:0060-0x000:007f	0x0000	0b11	0x0000-0x0007
0x000:0080-0x000:009f	0x0000	0b00	0x0008-0x000f
0x000:0feo-0x000:0fff	0x0000	0b11	0x00f8-0x00ff
0x000:1000-0x000:101f	0x0000	0b00	0x0100-0x0107
0x000:1020-0x000:103f	0x0000	0b01	0x0100-0x0107
0x000:1fe0-0x000:1fff	0x0000	0b11	0x01f8-0x01ff
0x000:2000-0x000:201f	0x0001	0b00	0x0000-0x0007
0x000:2020-0x000:203f	0x0001	0b01	0x0000-0x0007

Example 2: 1024-Byte Interleaving

In 1024-byte interleaving mode, the mapping scheme changes the DDR bank every other 2^8 = 256 columns. The BANK_SWITCH field is programmed to 8.

Table 284: Mapping scheme: 1024-Byte Interleaving, 256 Columns

MTL Address Range	Row Address	Bank Address	Column Address
0x000:0000-0x000:001f	0x0000	0b00	0x0000-0x0007
0x000:0020-0x000:003f	0x0000	0b00	0x0008-0x000f
0x000:0040-0x000:005f	0x0000	0b00	0x0010-0x0017

Table 284: Mapping scheme: 1024-Byte Interleaving, 256 Columns *…continued*

MTL Address Range	Row Address	Bank Address	Column Address
0x000:0060-0x000:007f	0x0000	0b00	0x0018-0x001f
0x000:0400-0x000:041f	0x0000	0b01	0x0000-0x0007
0x000:0800-0x000:081f	0x0000	0b10	0x0000-0x0007
0x000:0c00-0x000:0c1f	0x0000	0b11	0x0000-0x0007
0x000:1000-0x000:101f	0x0001	0b00	0x0000-0x0007
0x000:1400-0x000:141f	0x0001	0b01	0x0000-0x0007
0x000:2000-0x000:201f	0x0002	0b00	0x0000-0x0007
0x000:2400-0x000:241f	0x0002	0b01	0x0000-0x0007

Table 285: 1024-Byte Interleaving, 512 Columns

MTL Address Range	Row Address	Bank Address	Column Address
0x000:0000-0x000:001f	0x0000	0b00	0x0000-0x0007
0x000:0020-0x000:003f	0x0000	0b00	0x0008-0x000f
0x000:0040-0x000:005f	0x0000	0b00	0x0010-0x0017
0x000:0060-0x000:007f	0x0000	0b00	0x0018-0x001f
0x000:0400-0x000:041f	0x0000	0b00	0x0100-0x0107
0x000:0800-0x000:081f	0x0000	0b01	0x0000-0x0007
0x000:0c00-0x000:0c1f	0x0000	0b01	0x0100-0x0107
0x000:1000-0x000:101f	0x0000	0b10	0x0000-0x0007
0x000:1400-0x000:141f	0x0000	0b10	0x0100-0x0107
0x000:2000-0x000:201f	0x0001	0b00	0x0000-0x0007
0x000:2400-0x000:241f	0x0001	0b00	0x0100-0x0107

Example 3: Interleaving for 1024-Byte Strided Information

In 1024-byte strided mode, the mapping scheme puts burst-sized packets of 32 bytes that are 1024 bytes apart in the same DDR memory bank. The 1024_MODE field is programmed to 1.

Table 286: 1024-Byte Strided, 256 Columns

MTL Address Range	Row Address	Bank Address	Column Address
0x000:0000-0x000:001f	0x0000	0b00	0x0000-0x0007
0x000:0020-0x000:003f	0x0000	0b01	0x0000-0x0007
0x000:0040-0x000:005f	0x0000	0b10	0x0000-0x0007
0x000:0060-0x000:007f	0x0000	0b11	0x0000-0x0007
0x000:0080-0x000:009f	0x0001	0b00	0x0000-0x0007
0x000:00a0-0x000:00bf	0x0001	0b01	0x0000-0x0007
0x000:0100-0x000:011f	0x0002	0b00	0x0000-0x0007
0x000:0400-0x000:041f	0x0000	0b00	0x0008-0x000f
0x000:0420-0x000:043f	0x0001	0b01	0x0008-0x000f
0x000:0800-0x000:081f	0x0000	0b00	0x0010-0x0017

Table 286: 1024-Byte Strided, 256 Columns *…continued*

MTL Address Range	Row Address	Bank Address	Column Address
0x000:0c00-0x000:0c1f	0x0000	0b00	0x0018-0x001f
0x000:7fe0-0x000:7fff	0x0000	0b11	0x00f8-0x00ff
0x000:8000-0x000:801f	0x0008	0b00	0x0000-0x0007

5.7.2.7 DDR Memory Rank Locations

The DDR SDRAM Controller supports two DDR memory ranks. The location of these two memory ranks in the MTL address space is defined by means of MMIO registers IP_2031_RANK0_ADDR_LO, IP_2031_RANK0_ADDR_HI, and IP_2031_RANK1_ADDR_HI.

Rank 1 starts where rank0 leaves off in the MTL address space i.e., the ranks are successive. Programming these MMIO registers should be consistent with the memory size. An attempt to address an address outside of the two DDR memory ranks results in an error, which is registered by MMIO registers. Erroneous addressing still results in DDR read or write operations being performed.

The start addresses of the ranks should be a multiple of the respective rank sizes.

5.7.3 Operation

5.7.3.1 Clock programming

The operating frequency of the DTL interface is independent of the operating frequency of the rest of the DDR controller. A DDR controller internal asynchronous clock domain crossing communicates information from the DTL clock domain (“clk_dtl”) to the DDR SDRAM Controller memory clock domain (“clk_mtl”). If the clock of either of the two clock domains (“clk_dtl” or “clk_mtl”) is turned off, the DTL interface is non functional.

5.7.3.2 Asynchronous reset synchronization

The DDR SDRAM Controller has two reset generators, one for each clock domain: the MTL clock domain and the DTL clock domain. The reset generators use the “async_disable” signal to disable asynchronous resets. The “async_disable” signal is typically generated by a DDR controller external Test Control Block (TCB). The reset generator assumes this signal to be stable (it is typically only changed when changing between test modes, and normal operation mode). The generated resets, used by the DDR controller design are asserted (made ‘0’) asynchronously, and de-asserted (made ‘1’) synchronously. The generated reset is stretched for three cycles. The reset generator logic is described Figure 196.

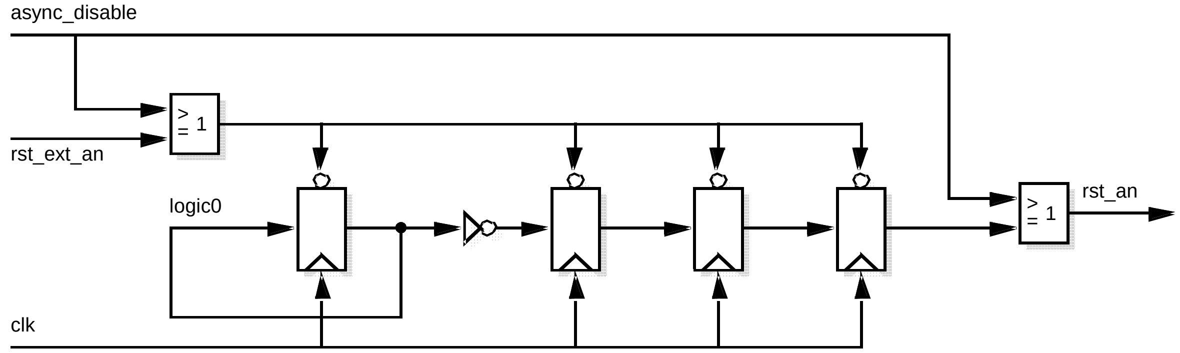


Fig 196.Reset generator logic

5.7.3.3 Programming via the DTL MMIO port

Overview: The DDR SDRAM Controller features a DTL MMIO port via which the DDR controller operation can be controlled and observed. The interface consists of a read/write port with a 32 bit wide data interface. It follows a simple non-pipelined protocol, which means it does not accept any command while it is processing a previous command. The interface supports full 32-bit read or write operations only, partial word read or write operations are not supported.

5.7.3.4 Power management

In order to reduce power consumption, the DDR SDRAM Controller can be placed into halt mode. During halt mode, the clock inputs to the DDR controller can be turned off to reduce dynamic power consumption. When the clock inputs to the DDR controller are turned off, it is non-functional. The DDR controller assumes that during halt mode the clock inputs to the DLLs may be turned off as well. As a result, the DDR controller power up sequence includes resetting the DLLs.

Note that when the clock inputs to the DDR controller are turned off, no access to the DDR controller MMIO registers is possible.

Putting the DDR SDRAM Controller in halt mode, and keeping the clock inputs to the DDR controller turned on, allows for safe programming of the MMIO registers using the DTL MMIO interface. When MMIO registers IP_2031_MR and IP_2031_EMR are re-programmed, a start action has to be performed (after the DDR controller is unhalted), for the new DDR values to take effect.

Halting and resuming: There are three different ways in which halting can be achieved:

1. Using a handshaking protocol using a dedicated pair of input/output signals to the DDR controller core.
2. Writing the halt register-field of a software programmable MMIO register.
3. Instructing the DDR SDRAM Controller to go into halt mode automatically after a certain period of inactivity.

The following paragraphs describe the halt state:

Handshaking protocol: The DDR SDRAM Controller has an “ip_2031_halt_req” input signal, and a “ip_2031_halt_ack” output signal, which implement a handshaking protocol to transfer the DDR controller into halt mode, and to get it out of halt mode.

The halting sequence works as follows. The “ip_2031_halt_req” input signal is made from ‘0’ into ‘1’ to indicate a request for halting. The DDR controller transfers itself to the halt mode state, and acknowledges the request for halting by changing the “ip_2031_halt_ack” output signal from ‘0’ into ‘1’. If so desired, power may be reduced by changing the DDR SDRAM Controller clock to a lower frequency.

The resuming sequence works as follows. The “ip_2031_halt_req” input signal is switched from ‘1’ into ‘0’ to indicate a request for resuming. The DDR controller transfers itself out of the halt mode state, and acknowledges the request for resuming by changing the “ip_2031_halt_ack” output signal from ‘1’ into ‘0’.

MMIO directed halt: MMIO register IP_2031_CTL, field HALT can be written with a ‘1’ to indicate a request for halting. Write a ‘0’ to this field to indicate a request for taking the DDR controller out of halt mode. Software should then wait for a time period equal to a minimum of 256 DDR SDRAM Controller clocks before clocks are changed or the PNX8550 is powered off.

Auto halt: The DDR SDRAM Controller can turn itself in halt mode when it has observed a certain period of inactivity. By programming the MMIO registers IP_2031_HALT_COUNT and IP_2031_CTL a period can be defined and automatic halting can be activated. The DDR controller automatically unhalts when a new MTL memory request is presented to one of its input ports. To ensure the IP_2031 can detect these MTL memory requests, the DDR controller clock inputs need to be turned on during auto halt (or at least have to be turned on before the MTL memory request is presented to the DDR controller).

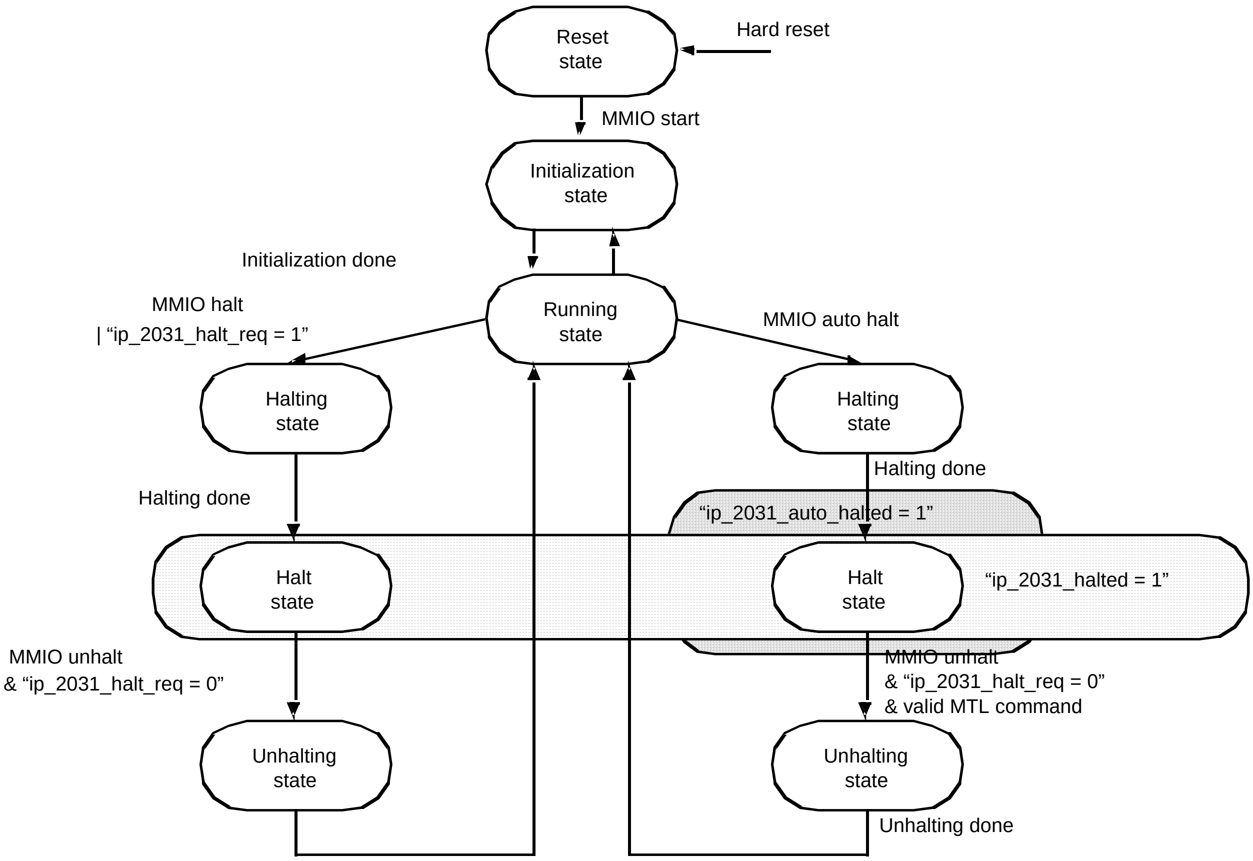


Fig 197.DDR SDRAM controller start and halt state machine

Observing halt mode: The DDR SDRAM Controller has two output signals to indicate when it is in halt mode, and why it went into halt mode. The "ip_2031_halted" output signal indicates when the DDR controller is in halt mode. The "ip_2031_auto_halted" output signal indicates when the DDR controller is in halt mode due to auto halt.

When the DDR SDRAM Controller enters halt mode due to a auto halt, it only unhalts when a MTL memory request is presented to one of its input ports. To ensure the DDR controller can detect these MTL memory requests, the DDR controller clock inputs need to be turned on during auto halt (or at least have to be turned on before the MTL memory request is presented to the DDR controller). Therefore it is advised not to turn the clock to the DDR controller off when "ip_2031_auto_halted" is '1'.

An alternative method to determine if the DDR SDRAM Controller is in halt mode, is by reading the HALT_STATUS bit in the MMIO register IP_2031_CTL ("clk_dtl" and "clk_mtl" need to be on to do this).

Sequence of actions: To enter halt mode, the DDR SDRAM Controller performs the following sequence of actions:

1. Precharge all banks (of all ranks)
2. Apply a NOP command
3. Enter self refresh mode, with CKE low, deactivate internal DLL

To leave halt mode, the DDR controller performs the following action:

256 DDR SDRAM Controller memory cycles with CKE high, NOP commands, to activate DLL.

5.7.4 Application notes

5.7.4.1 Memory configurations

The DDR SDRAM Controller supports a wide range of DDR SDRAM memory configurations. Some examples of memory configurations that are supported for an external data bus of 32 bits are shown in Figure 198.

On the left side a single physical bank of DDR devices is connected to the DDR controller. Throughout this document the term rank is used for a physical bank in order to prevent any confusion with the logical banks inside the DDR devices.

On the right hand side two ranks of DDR devices are connected to the DDR controller. In single rank configurations, there is no need to drive the chip select inputs on the DDR devices from the DDR controller. In a multi-rank configuration, each rank receives its own chip select signal from the DDR controller.

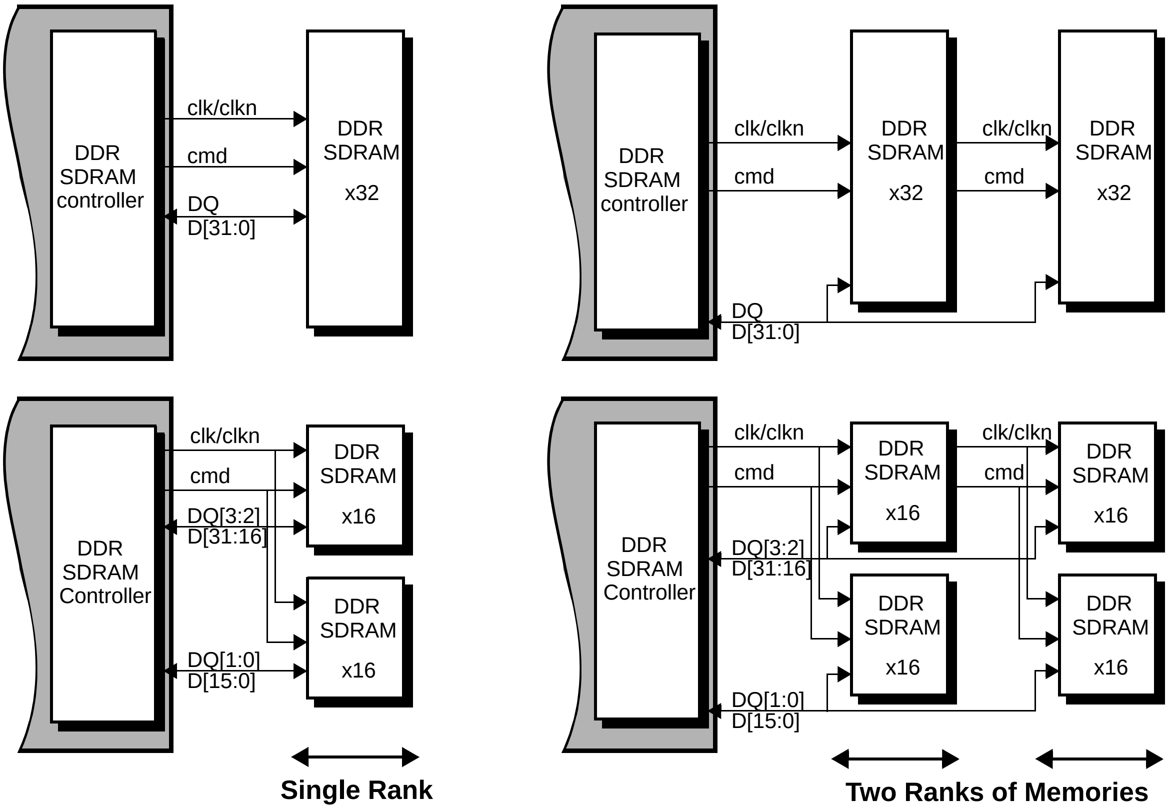


Fig 198.External memory configurations

5.7.4.2 Error signaling

The DTL port does not support error signaling. Reads from invalid addresses return the value “0”, writes to invalid addresses are ignored. The DTL port error signals “dtl_err_rd” and “dtl_err_wr” always have the value ‘0’.

5.7.4.3 Latency

The DDR SDRAM controller uses two pipeline stages to calculate the command(s) that is issued to DDR memories after a MTL command is accepted by the controller.

The latency of an MTL read command is described as follows:- Assume there is a MTL read command on one of the MTL ports in cycle 0, which is accepted by the DDR controller. In cycle 1, the DDR controller determines the first DDR burst for the MTL read command. In cycle 2, the controller determines the DDR commands that need to be sent out on the DDR interface (assuming there are no other MTL transactions pending in the DDR controller).

When the read is to an already activated row in cycle 3, a DDR read command appears on the DDR interface. Given a CAS latency of n cycles (typically the CAS latency is 2, 2.5, 3, 3.5, or 4 cycles), the first read data element is presented by the memory device in cycle 3 + n. To allow for safe clock domain transfer (from the “dqs” clock domain to the “clk_mtl” clock domain) and to combine two DDR read data elements into a single MTL read data element, the DDR controller takes two extra cycles before presenting the read data on the MTL interface in cycle 3 + n + 2.

As a result, the lowest latency from an MTL read command to accept the first valid MTL read data element on the MTL interface is 3 + n + 2 cycles. The latency increases in cases of pending MTL transactions in the DDR controller and required DDR precharge and activate commands.

5.7.4.4 Endianness

The DDR SDRAM Controller is not endianness aware. The MTL bus is byte-oriented and byte address 0 is transferred on data lines data[7:0]. Consider the DDR controller to be little endian.

5.7.4.5 Booting the DDR SDRAM controller

The following C-program performs initialization for the following configuration:

- DDR memory is covering region 0x00000000 up to 0x02000000 of the 32-bit address space.
- 200 MHz DDR memory characteristics (from the DDR memory data sheet)
- Full page mode address mapping
- One rank configuration

Remark: The following configuration does not apply to all memories listed in Section 5.7.4.8. The user should program the correct configuration based on the type of DDR memory component that is actually used in the system.

```
  /* Initialization for micron 128 Mbits x 16 @ 166 MHz (MT46V8M16-6)

void InitializeIP2031 ()

  /* DDR memory starts at address 0x00000000. */

  *((unsigned int *) IP_2031_RANK0_ADDR_LO) = 0x00000000;

  /* DDR memory ends at address 0x01ffffff. */

  *((unsigned int *) IP_2031_RANK0_ADDR_HI) = 0x01ffffff;
```

```
/* Use only rank 0. */
*((unsigned int *) IP_2031_RANK1_ADDR_HI) = 0x01ffffff;
/* Full page mode, switch bank every 2^10 elements, 10 is the amount of
   columns for the DDR memory. */
*((unsigned int *) IP_2031_DDR_DEF_BANK_SWITCH) = 0x0000000a;
/* Turn "region1" off. */
*((unsigned int *) IP_2031_DDR_REGION1_BANK_SWITCH) = 0x00000000;
/* DDR memory has 2^12 rows. */
*((unsigned int *) IP_2031_DDR_ROW_WIDTH) = 0x0000000c;
/* DDR memory has 2^9 columns. */
*((unsigned int *) IP_2031_DDR_COLUMN_WIDTH) = 0x00000009;
/* DDR memory precharge bit is at bit location 10. */
*((unsigned int *) IP_2031_DDR_PRECHARGE_BIT) = 0x0000000a;
/* DDR memory mode register, for CAS latency 2.5, burst type sequential,
   burst length 8. */
*((unsigned int *) IP_2031_DDR_MR) = 0x0000003;
/* DDR extended mode register. */
/* Enable the memory DLL */
*((unsigned int *) IP_2031_DDR_EMR) = 0x00000000;
/* Trcd_wr = 3, Trcd_rd = 3 */
*((unsigned int *) IP_2031_DDR_TRCD) = 0x00030003;
/* Trc = 10 */
*((unsigned int *) IP_2031_DDR_TRC) = 0x0000000a;
/* Twtr = 1 */
*((unsigned int *) IP_2031_DDR_TWTR) = 0x00000001;
/* Twr = 3 */
*((unsigned int *) IP_2031_DDR_TWR) = 0x00000003;
/* Trp = 3 */
*((unsigned int *) IP_2031_DDR_TRP) = 0x00000003;
```

```
    /* Tras = 7 */

    *((unsigned int *) IP_2031_DDR_TRAS) = 0x00000007;

    /* Trrd = 2 */

    *((unsigned int *) IP_2031_DDR_TRRD) = 0x00000002;

    /* Trfc = 12 */

    *((unsigned int *) IP_2031_DDR_TRFC) = 0x0000000c;

    /* Tmrd = 2 */

    *((unsigned int *) IP_2031_DDR_TMRD) = 0x00000002;

    /*Need to add one to data sheet TCAS*/

    /* TCAS = 2.5+1, program 7 to indicate 7 half cycles. */

    *((unsigned int *) IP_2031_DDR_TCAS) = 0x00000007;

    /* Refresh period (2600) calculated using DDR frequency of 166 MHz. */

    *((unsigned int *) IP_2031_DDR_RF_PERIOD) = 0x00000A28;

    /* Start the controller, with speculative auto precharge option on. */

    *((unsigned int *) IP_2031_CTL) = 0x00000003;

    /*******************************************************
     * For arbitration, MMIO register reset values are used. *
     *******************************************************/
}
```

5.7.4.6 Data coherency

Memory requests at an MTL port of the DDR controller are processed and executed in the order that they are received. The DDR controller does not re-order the commands on a MTL interface. From this point of view data coherency between memory bus agents that connect to a single port on the DDR controller is guaranteed.

However, the memory requests that are made to different MTL interfaces on the DDR SDRAM Controller in general are not serviced in the order that they appeared. The order in which these requests are serviced depends on the state of the SDRAM device(s) and how the internal Arbiter is programmed. The user needs to take care of data coherency between memory agents that connect to different MTL ports of the DDR controller.

5.7.4.7 Programming the internal arbiter

To program the parameters for the internal Arbiter follow the steps below:

1. Determine the total available bandwidth (tot_bw) (based on your ddr setup, frequency and expected average ddr efficiency e.g. 75%), determine required peak hard real time bandwidth (hrt_pk), average hard real time bandwidth (hrt_avg), average soft real time (srt) and average total cpu band width (cpu).
2. Select the minimum window size allowed (min), 20 or 40 are good examples. Higher minimum values increase the latency, but can also slightly increase the DDR efficiency (because more requests of one type (DMA or CPU) are handled in sequence). The value 20 is based on a 128 byte transfer that takes 16 data transfer cycles on a 32-bit DDR interface. Assuming an average DDR efficiency of 80% on average a total of 20 cycles would be needed to start and finish this transfer.
3. Use the minimum window value for the port with the least traffic (DMA or CPU) and calculate the other window according to the following formula:

   ```
   if (hrt_pk < cpu)
   hrt_window = min;
   cpu_window = ((tot_bw / hrt_pk) -1)* hrt_window;
   else
   cpu_window = min;
   hrt_window = (hrt_pk / (tot_bw - hrt_pk)) * cpu_window;
   endif
   ```

4. If the selected minimum value is low and the calculated window size is much bigger than that minimum value, then setting ‘always’ preempt (0b11) on the high bandwidth type of traffic (and perhaps ‘never’ preempt (0b00) on the low bandwidth type of traffic) may be needed to ensure that the low bandwidth agents get enough traffic.
5. The next parameter to calculate is cpu_ratio. To do this, first account for the fact that normally not all available bandwidth is used. It is a good idea to distribute the headroom proportionally between the cpu’s and the soft real time DMA. As is done with the following formulas:

   srt2 = (tot_bw - hrt_avg) * srt / (srt + cpu);

   cpu2 = (tot_bw - hrt_avg) * cpu / (srt + cpu);

6. The cpu_ratio and cpu_limit make sure that when the cpus are requesting too much bandwidth that they get blocked out in the cpu window and soft real time DMA is allowed access instead. The cpu_ratio determines for how many cycles the cpus get blocked (in favour of the DMA) for each cycle the ddr spends on cpu data transfer. The cpu_ratio is added to the account for each ddr burst, a ddr burst length is 4 cycles. So the formula for cpu_ratio is:

   cpu_ratio = 4 * (hrt_avg + srt2) / cpu2;

7. Finally the cpu_limit needs to be estimated. cpu_limit basically determines how many consecutive cpu transfers are allowed to finish before the cpus get blocked out. In case of one cpu, one might choose to allow one data cache line replacement (copy back and fetch) and one instruction fetch. Assuming a data and instruction cache line size of 64 bytes that is a total of 3*64 = 192 bytes. For each ddr burst (4 clock cycles) the ddr transfers (for a dual data rate, 32-bit ddr interface) 4*2*4=32 bytes. So 192/32 = 6 bursts are needed. So the cpu_limit needs to be at least 6*cpu_ratio. If this amount is to be allowed for each of the 3 cpus then that value needs to be multiplied by 3 again. Keep in mind however that all requested cpu transfers (when cpu bandwidth is peaking) do not fit in one cpu_window. So all these cpu transfers are not

executed consecutively. In general setting the cpu_limit too low blocks the cpus too much causing a too high latency (execution time). Setting the cpu_limit too high can block the soft real time DMA completely for a long time under the condition that hard real time DMA is peaking and cpu bandwidth is peaking. But the long latency that causes the soft real time may not be a problem.

8. Generally, programming the ratio and limit for the individual cpus is unnecessary. If required, provide each cpu with a fixed high or low priority. That can be achieved by making ratio for the high priority cpu(s) 0 and giving it the maximum limit and for the low priority cpu(s) vice versa, ratio=max and limit=0. If this is not sufficient, use the same approach as for the general cpus ratio and limit. The cpu_dma_decr bit selects whether the individual cpu accounts count down during the hrt_window or not. If not, then the hard real time traffic should not be taken into account when calculating the ratio and limit for the individual cpus. Note that a cpu may get access in an hrt window (therefore getting the account increased) if there is no DMA request pending. This scheme with dividing cpu bandwidth based on these individual cpu ratios and limits only works properly if the cpu accounts remain near their respective limits (otherwise it is fixed priorities). Preventing the cpu accounts from becoming excessively high or low can be very difficult considering the bursty behavior of the cpus and the hard real time DMA.

5.7.4.8 Compatible DDR parts list

The DDR SDRAM Controller is compatible with the following DDR SDRAM parts for supported programmable and configurable features mentioned in this specification. A few of the below mentioned part configurations are checked using denali models and others are verified with configurable verilog DDR model testbench (for all supported features) based on Micron memory model.

Table 287: Compatible DDR parts

Vendor	Configuration	Part number	Speed and CAS latency
Elpida	256 MB(4Mx16bitx4bank)	HM542S161B-10	125 MHz, CL= 2.5
Elpida	256 MB(4Mx16bitx4bank)	HM542S161B-75A	143 MHz, CL= 2.5
Elpida	256 MB(4Mx16bitx4bank)	HM542S161B-75B	133 MHz, CL= 2.5
Elpida	256 MB(8Mx8bitx4bank)	HM542S801B-10	125 MHz, CL= 2.5
Elpida	256 MB(8Mx8bitx4bank)	HM542S801B-75A	143 MHz, CL= 2.5
Elpida	256 MB(8Mx8bitx4bank)	HM542S801B-75B	133 MHz, CL= 2.5
Elpida	128 MB(2Mx16bitx4bank)	UPD45D128164GS-C75	133 MHz, CL= 2.5
Elpida	128 MB(2Mx16bitx4bank)	UPD45D128164GS-C75	125 MHz, CL= 2.5
Elpida	128 MB(4Mx8bitx4bank)	UPD45D128842GS-C75	133 MHz, CL= 2.5
Elpida	128 MB(4Mx8bitx4bank)	UPD45D128842GS-C80	125 MHz, CL= 2.5
Hynix	512 MB(8Mx16bitx4bank)	HY5DU121622T-H	133 MHz, CL= 2.5
Hynix	512 MB(8Mx16bitx4bank)	HY5DU121622T-K	133 MHz, CL= 2,2.5
Hynix	512 MB(8Mx16bitx4bank)	HY5DU121622T-L	100 MHz, CL= 2,2.5
Hynix	512 MB(16Mx8bitx4bank)	HY5DU12822T-H	133 MHz, CL= 2.5
Hynix	512 MB(16Mx8bitx4bank)	HY5DU12822T-K	133 MHz, CL= 2,2.5
Hynix	512 MB(16Mx8bitx4bank)	HY5DU12822T-L	100 MHz, CL= 2,2.5
Hynix	128 MB(2Mx16bitx4bank)	HY5DU281622T-H	133 MHz, CL= 2.5
Hynix	128 MB(2Mx16bitx4bank)	HY5DU281622T-K	133 MHz, CL= 2,2.5

Table 287: Compatible DDR parts *…continued*

Vendor	Configuration	Part number	Speed and CAS latency
Hynix	128 MB(2Mx16bitx4bank)	HY5DU281622T-L	100 MHz, CL= 2,2.5
Hynix	128 MB(4Mx8bitx4bank)	HY5DU28822AT-H	133 MHz, CL= 2.5
Hynix	128 MB(4Mx8bitx4bank)	HY5DU28822AT-K	133 MHz, CL= 2.2.5
Hynix	128 MB(4Mx8bitx4bank)	HY5DU28822AT-L	100 MHz, CL= 2.2.5
Hynix	256 MB(4Mx16bitx4bank)	HY5DU561622-H	133 MHz, CL= 2.5
Hynix	256 MB(4Mx16bitx4bank)	HY5DU561622-K	133 MHz, CL= 2.2.5
Hynix	256 MB(4Mx16bitx4bank)	HY5DU561622-L	100 MHz, CL= 2.2.5
Hynix	256 MB(8Mx8bitx4bank)	HY5DU56822-H	133 MHz, CL= 2.5
Hynix	256 MB(8Mx8bitx4bank)	HY5DU56822-K	133 MHz, CL= 2.2.5
Hynix	256 MB(8Mx8bitx4bank)	HY5DU56822-L	100 MHz, CL= 2.2.5
Hynix	64 MB(1Mx16bitx4bank)	HY5DV641622AT-5	200 MHz, CL=
IBM	256 MB(4Mx16bitx4bank)	IBM0625164GT3-10	125 MHz, CL=2.5
IBM	256 MB(4Mx16bitx4bank)	IBM0625164GT3-75	143 MHz, CL=2.5
IBM	256 MB(4Mx16bitx4bank)	IBM0625164GT3-8	133 MHz, CL=2.5
IBM	256 MB(8Mx8bitx4bank)	IBM0625804GT3-10	125 MHz, CL=2.5
IBM	256 MB(8Mx8bitx4bank)	IBM0625804GT3-75	143 MHz, CL=2.5
IBM	256 MB(8Mx8bitx4bank)	IBM0625804GT3-8	133 MHz, CL=2.5
INFINEON	256 MB(8Mx8bitx4bank)	HYB25D256800T-7	143 MHz, CL=2.5
INFINEON	256 MB(8Mx8bitx4bank)	HYB25D256800T-75	133 MHz, CL=2.5
INFINEON	256 MB(8Mx8bitx4bank)	HYB25D256800T-8	125 MHz, CL=2.5
JEDEC	64 MB(2Mx8bitx4)	JEDEC-64MB-8-PC266	143 MHz, CL=2.5
MICRON	256 MB(4Mx16bitx4bank)	MT46V16M16-75Z	133 MHz, CL=2.2.5
MICRON	256 MB(4Mx16bitx4bank)	MT46V16M16FJ-6	166 MHz, CL=2.5
MICRON	256 MB(4Mx16bitx4bank)	MT46V16M16TG-75	133 MHz, CL=2.5
MICRON	256 MB(4Mx16bitx4bank)	MT46V16M16TG-8	125 MHz, CL=2.5
MICRON	128 MB(4Mx8bitx4bank)	MT46V16M8-6	166 MHz, CL=2.5
MICRON	128 MB(4Mx8bitx4bank)	MT46V16M8-6T	166 MHz, CL=2.5
MICRON	128 MB(4Mx8bitx4bank)	MT46V16M8-75Z	133 MHz, CL=2.2.5
MICRON	128 MB(4Mx8bitx4bank)	MT46V16M8FJ-6	166 MHz, CL=2.5
MICRON	128 MB(4Mx16bitx4bank)	MT46V16M8TG-75	133 MHz, CL=2.5
MICRON	128 MB(4Mx16bitx4bank)	MT46V16M8TG-8	125 MHz, CL=2.5
MICRON	512 MB(8Mx16bitx4bank)	MT46V32M16-75	133 MHz,CL=2.5
MICRON	512 MB(8Mx16bitx4bank)	MT46V32M16-75Z	133 MHz,CL=2,2.5
MICRON	512 MB(8Mx16bitx4bank)	MT46V32M16-8	125 MHz,CL=2.5
MICRON	256 MB(8Mx8bitx4bank)	MT46V32M16-75Z	133 MHz,CL=2,2.5
MICRON	256 MB(8Mx8bitx4bank)	MT46V32M16FJ-6	166 MHz,CL=2.5
MICRON	256 MB(8Mx8bitx4bank)	MT46V32M16TG-75	133 MHz,CL=2.5
MICRON	256 MB(8Mx8bitx4bank)	MT46V32M16TG-8	125 MHz,CL=2.5
MICRON	512 MB(16Mx8bitx4)	MT46V64M8-75	133 MHz,CL=2.5
MICRON	512 MB(16Mx8bitx4)	MT46V64M8-75Z	133 MHz,CL=2.5,2
MICRON	512 MB(16Mx8bitx4)	MT46V64M8-8	125 MHz,CL=2.5

Table 287: Compatible DDR parts *…continued*

Vendor	Configuration	Part number	Speed and CAS latency
MICRON	128 MB(2Mx16bitx4bank)	MT46V8M16-6	166 MHz,CL=2.5
MICRON	128 MB(2Mx16bitx4bank)	MT46V8M16-6T	166 MHz,CL=2.5
MICRON	128 MB(2Mx16bitx4bank)	MT46V8M16-75Z	133 MHz,CL=2.5
MICRON	128 MB(2Mx16bitx4bank)	MT46V8M16TG-75	133 MHz,CL=2.5
MICRON	128 MB(2Mx16bitx4bank)	MT46V8M16TG-75Z	133 MHz,CL=2,2.5
MICRON	128 MB(2Mx16bitx4bank)	MT46V8M16TG-8	125 MHz,CL=2.5
NANYA	128 MB(4Mx8bitx4bank)	NT5DS16M8A-6	166 MHz,CL=2.5
NANYA	128 MB(4Mx8bitx4bank)	NT5DS16M8A-66	151 MHz,CL=2.5
NANYA	128 MB(4Mx8bitx4bank)	NT5DS16M8A-75B	133 MHz,CL=2.5
NANYA	128 MB(4Mx8bitx4bank)	NT5DS16M8A-7K	143 MHz,CL=2.5
NANYA	128 MB(4Mx8bitx4bank)	NT5DS16M8A-8B	125 MHz,CL=2.5
NANYA	256 MB(8Mx8bitx4bank)	NT5D532M8A-6	166 MHz,CL=2.5
NANYA	256 MB(8Mx8bitx4bank)	NT5D532M8A-66	151MHz,CL=2.5
NANYA	256 MB(8Mx8bitx4bank)	NT5D532M8A-75B	133 MHz,CL=2.5
NANYA	256 MB(8Mx8bitx4bank)	NT5D532M8A-7K	143 MHz,CL=2.5
NANYA	256 MB(8Mx8bitx4bank)	NT5D532M8A-8B	125 MHz,CL=2.5
SAMSUNG	128 MB(4Mx8bitx4bank)	K4H280838B-TCA0	100 MHz,CL=2
SAMSUNG	128 MB(4Mx8bitx4bank)	K4H280838B-TCA2	133 MHz,CL=2,2.5
SAMSUNG	128 MB(4Mx8bitx4bank)	K4H280838B-TCB0	133 MHz,CL=2.5
SAMSUNG	128 MB(2Mx16bitx4bank)	K4H281638B-TCA0	100 MHz,CL=2

5.7.5 Register descriptions

The DDR SDRAM Controller contains a number of MMIO registers that are software programmable through the DTL port. These registers can be used to:

- Set generic control and read generic status information
- Set dimensions of the DDR memories
- Set timing characteristics of the DDR memories
- Set arbitration parameters
- Observe the performance of the DDR SDRAM Controller
- Observe specifics about errors

Changing MMIO registers of an initiated DDR SDRAM Controller may cause incorrect behavior. Turning the DDR controller into halt mode, programming MMIO registers while in halt mode, and unhalting the DDR controller when the MMIO registers have been programmed, is the suggested series of actions to change MMIO register values of a started DDR controller.

Some MMIO register fields have a configurable reset value. These values may differ between various deliverables of the DDR controller. For the PNX2015 deliverable, the reset values of the configurable fields are defined in the following tables.

5.7.5.1 Register summary

Table 288: Register summary

Offset	Symbol	Description
0x0	IP_2031_CTL0	DDR GENERAL CONTROL
0x4	IP_2031_DDR_DEF_BANK_SWITCH0	DDR BANK SWITCH ADDRESSING
0x8	IP_2031_AUTO_HALT_LIMIT0	DDR AUTO HALT LIMIT
0x10	IP_2031_RANK0_ADDR_LO0	DDR RANK0 ADDRESS LOW LIMIT
0x14	IP_2031_RANK0_ADDR_HI0	DDR RANK0 ADDRESS HIGH LIMIT
0x18	IP_2031_RANK1_ADDR_HI0	DDR RANK1 ADDRESS HIGH LIMIT
0x40	IP_2031_DDR_REGION1_BASE0	DDR REGION1 BASE ADDRESS
0x44	IP_2031_DDR_REGON1_MASK0	DDR REGION1 ADDRESS MASK
0x48	IP_2031_DDR_REGION1_BANK_SWITCH0	DDR REGION1 BANK SWITCH ADDRESS
0x80	IP_2031_DDR_MR0	DDR MODE REGISTER
0x84	IP_2031_DDR_EMR0	DDR EXTEND MODE REGISTER
0x88	IP_2031_DDR_PRECHARGE_BIT0	DDR PRECHARGE BIT FIELD
0xC0	IP_2031_RANK0_ROW_WIDTH0	DDR RANK0 ROW BIT WIDTH
0xC4	IP_2031_RANK0_COLUMN_WIDTH0	DDR RANK0 COLUMN BIT WIDTH
0xD0	IP_2031_RANK1_ROW_WIDTH0	DDR RANK1 ROW BIT WIDTH
0xD4	IP_2031_RANK1_COLUMN_WIDTH0	DDR RANK1 COLUMN BIT WIDTH
0x100	IP_2031_DDR_TRCD0	DDR ACTIVE to READ or WRITE DELAY
0x104	IP_2031_DDR_TRC0	DDR ACTIVE to ACTIVE/AUTO REFRESH DELAY
0x108	IP_2031_DDR_TWTR0	DDR INTERNAL WRITE to READ COMMAND DELAY
0x10C	IP_2031_DDR_TWR0	DDR WRITE RECOVERY TIME
0x110	IP_2031_DDR_TRP0	DDR PRECHARGE COMMAND PERIOD
0x114	IP_2031_DDR_TRAS0	DDR ACTIVE to PRECHARGE COMMAND PERIOD
0x11C	IP_2031_DDR_TRRD0	DDR ACTIVE BANK A to ACTIVE BANK B COMMAND
0x120	IP_2031_DDR_TRFC0	DDR AUTO REFRESH COMMAND PERIOD
0x124	IP_2031_DDR_TMRD0	DDR LOAD MODE REGISTER COMMAND CYCLE
0x128	IP_2031_DDR_TCAS0	DDR CAS READ LATENCY
0x12C	IP_2031_DDR_RF_PERIOD0	DDR REFRESH PERIOD
0x180	IP_2031_ARB_CTL0	DDR ARBITER CONTROL
0x184	IP_2031_ARB_HRT_WINDOW0	DDR ARBITER HARD REAL TIME WINDOW
0x188	IP_2031_ARB_CPU_WINDOW0	DDR ARBITER CPU WINDOW
0x1C0	IP_2031_ARB_CPU_LIMIT0	DDR ARBITER CPU LIMIT
0x1C4	IP_2031_ARB_CPU_RATIO0	DDR ARBITER CPU RATIO
0x1C8	IP_2031_ARB_CPU_CLIP0	DDR ARBITER CPU CLIP
0x1CC	IP_2031_ARB_CPU_DECR0	DDR ARBITER CPU DECR.

Table 288: Register summary *…continued*

Offset	Symbol	Description
0x1D0	IP_2031_ARB_CPU1_LIMIT0	DDR ARBITER CPU1 LIMIT
0x1D4	IP_2031_ARB_CPU1_RATIO0	DDR ARBITER CPU1 RATIO
0x1D8	IP_2031_ARB_CPU1_CLIP0	DDR ARBITER CPU1 CLIP
0x1DC	IP_2031_ARB_CPU1_DECR0	DDR ARBITER CPU1 DECR.
0x1E0	IP_2031_ARB_CPU2_LIMIT0	DDR ARBITER CPU2 LIMIT
0x1E4	IP_2031_ARB_CPU2_RATIO0	DDR ARBITER CPU2 RATIO
0x1E8	IP_2031_ARB_CPU2_CLIP0	DDR ARBITER CPU2 CLIP
0x1EC	IP_2031_ARB_CPU2_DECR0	DDR ARBITER CPU2 DECR.
0x1F0	IP_2031_ARB_CPU3_LIMIT0	DDR ARBITER CPU3 LIMIT
0x1F4	IP_2031_ARB_CPU3_RATIO0	DDR ARBITER CPU3 RATIO
0x1F8	IP_2031_ARB_CPU3_CLIP0	DDR ARBITER CPU3 CLIP
0x1FC	IP_2031_ARB_CPU3_DECR0	DDR ARBITER CPU3 DECR.
0x200	IP_2031_PF_MTL0_RD_VALID0	DDR PERFORMANCE MTL0 READ VALID
0x204	IP_2031_PF_MTL0_WR_ACCEPT0	DDR PERFORMANCE MTL0 WRITE ACCEPT
0x208	IP_2031_PF_MTL1_RD_VALID0	DDR PERFORMANCE MTL1 READ VALID
0x20C	IP_2031_PF_MTL1_WR_ACCEPT0	DDR PERFORMANCE MTL1 WRITE ACCEPT
0x210	IP_2031_PF_MTL2_RD_VALID0	DDR PERFORMANCE MTL2 READ VALID
0x214	IP_2031_PF_MTL2_WR_ACCEPT0	DDR PERFORMANCE MTL2 WRITE ACCEPT
0x218	IP_2031_PF_MTL3_RD_VALID0	DDR PERFORMANCE MTL3 READ VALID
0x21C	IP_2031_PF_MTL3_WR_ACCEPT0	DDR PERFORMANCE MTL3 WRITE ACCEPT
0x240	IP_2031_PF_IDLE0	DDR PERFORMANCE IDLE
0x280	IP_2031_ERR_VALID0	DDR ERROR VALID
0x284	IP_2031_ERR_MTL_PORT0	DDR ERROR MTL PORT
0x288	IP_2031_ERR_MTL_CMD_ADDR0	DDR ERROR MTL COMMAND ADDRESS
0x28C	IP_2031_ERR_MTL_CMD_READ0	DDR ERROR MTL COMMAND READ
0x290	IP_2031_ERR_MTL_CMD_ID0	DDR ERROR MTL COMMAND ID
0xFFC	IP_2031_MODULE_ID0	DDR MODULE ID

5.7.5.2 Register table

In Table 289 the Register IP_2031_CTL0 bitfield DDR_HALVE_WIDTH should always be set to 0 to ensure that the complete bus of the DDR interface is used.

Table 289: DDR SDRAM controller registers

Bit	Symbol	Access	Value	Description
Offset 0x0 - IP_2031_CTL0				
31	HALT_STATUS	R	0x0	0: Not in halt mode 1: Halt mode
30	AUTO_HALT_STATUS	R	0x0	Indicate if halt is due to auto halt or not 0: Not in auto halt mode 1: Auto halt mode
29:16	RSD[29:16]	R	0xXXXX	These bits should be ignored when read and written as O's
15	HALT	R	0x0	0: Unhalt when in halt mode 1: Halt when not in halt mode;
14	AUTO_HALT	R	0x0	0: No automatic halt allowed 1: allow automatic halt
13	RSD	R/W	0x0	These bits should be ignored when read and written as O's
12:5	RSD[12:5]	R	0xXX	These bits should be ignored when read and written as O's
4	DIS_WRITE_INT	R/W	0x0	This bit must be set for non-JEDEC compliant devices which don't allow write burst to be terminated by read command. 0: DDR write burst would be interrupted by following read command 1: DDR write burst would never be interrupted by following read command
3	DDR_DQS_PER_BYTE	R/W	0x0	0: A single "dqs" signal is provided for all "dq" byte lane. Output signal "ip_2031_dqs_out[0]" should be used for all byte lanes. 1: A separate "dqs" signal is provided for every "dq" byte lane. These strobe signals are used to register "dq" byte lanes.
2	DDR_HALVE_WIDTH	R/W	0x0	0: The complete "dq" bus of the DDR interface is used. 1: Only the lower half of the dq bus of the DDR interface is used. Only DDR data bits dq[15:0} are in use.;
1	SPEC_AUTO_PR	R/W	0x0	0: Speculative auto precharge is off 1: speculative auto precharge is on;

Table 289: DDR SDRAM controller registers *...continued*

Bit	Symbol	Access	Value	Description
0	START	R/W	0x0	 1: Start DDR controller. When the controller is started, this bit will be turned to '0' by the controller itself
Offset 0x4 - IP_2031_DDR_DEF_BANK_SWITCH0				
31:18	RSD[31:18]	R	0xXXXX	These bits should be ignored when read and written as O's
17	DEF_BANK_2048_MODE	R/W	0x0	 1: Sets DDR addressing optimized for 2D transactions with a 2048 byte stride
16	DEF_BANK_1024_MODE	R/W	0x0	 1: Sets DDR addressing optimized for 2D transactions with a 1024 byte stride.
15:4	RSD[15:4]	R	0xXXX	These bits should be ignored when read and written as O's
3:0	BANK_SWITCH[3:0]	R/W	0x3	Switch banks every 2*BANK_SWITCH columns (each column has a width of 4 bytes). For 32 byte interleaving set this value equal to 0x3. For full page/row interleaving set this value equal to the column width value. Only the following values are supported: 0x3, 0x4, 0x5, 0x6, 0x7, 0x8, 0x9, 0xa and oxb
Offset 0x8 - IP_2031_AUTO_HALT_LIMIT0				
31:0	HALT_LIMIT[31:0]	R/W	0xXXXX XXXX	After LIMIT amount of IP_2031 idle cycles, automatic halt kicks in
Offset 0x10 - IP_2031_RANK0_ADDR_LO0				
31:0	ADDR_LO[31:0]	R/W	0x00000 000	Address at which the DDR rank 0 address space starts
Offset 0x14 - IP_2031_RANK0_ADDR_HI0				
31:0	ADDR_HI_RANK0[31:0]	R/W	0xFFFF FFFF	Address at which the DDR rank 0 address space end
Offset 0x18 - IP_2031_RANK1_ADDR_HI0				
31:0	ADDR_HI_RANK1[31:0]	R/W	0xFFFF FFFF	Address at which the DDR rank 1 address space ends
Offset 0x40 - IP_2031_DDR_REGION1_BASE0				
31:16	REGION_BASE[15:0]	R/W	0xXXXX	Region base address offset
15:0	RSD[15:0]	R	0xXXXX	These bits should be ignored when read and written as O's
Offset 0x44 - IP_2031_DDR_REGON1_MASK0				
31:16	REGION_MASK[15:0]	R/W	0xXXXX	Region mask bits
15:0	RSD[15:0]	R	0xXXXX	These bits should be ignored when read and written as O's
Offset 0x48 - IP_2031_DDR_REGION1_BANK_SWITCH0				

Table 289: DDR SDRAM controller registers *...continued*

Bit	Symbol	Access	Value	Description
31	REGION_ON	R/W	0x0	1: Enables region 1
30:18	RSD[30:18]	R	0xXXXX	These bits should be ignored when read and written as O's
17	REGION1_2048_MODE	R/W	0x0	1: Sets DDR addressing optimized for 2D transactions with a 2048 byte
16	REGION1_1024_MODE	R/W	0x0	1: Sets DDR addressing optimized for 2D transactions with a 1024 byte stride
15:4	RSD[15:4]	R	0x000	These bits read zero and can not be altered by write.
3:0	REGION1_BANK_SWITCH[3:0]	R/W	0x3	Switch banks every 2*BANK_SWITCH columns for region 1(each column has a width of 4 bytes). For 32 byte interleaving set this value equal to 0x3. For full page/row interleaving set this value equal to the column width value. Only the following values are supported: 0x3, 0x4, 0x5, 0x6, 0x7, 0x8, 0x9, 0xa and 0xb
Offset 0x80 - IP_2031_DDR_MR0				
31:13	RSD[31:13]	R	0xXXXXX	These bits should be ignored when read and written as O's
12:0	MR[12:0]	R/W	0x0043	Mode register . The assumption is that the DLL reset bit is a bit location 8. Use the data sheet of the DDR memory to determine the value of this register. The following is taken from a DDR datasheet and describes the different bits of the mode register. Bits 0 up to 2: burst length Bit 3: burst type (0:sequential, 1:interleaved) Bits 4 up to 6: CAS latency Bits 7 and up: operating mode (0:normal operation, 2: normal operation/reset DLL
Offset 0x84 - IP_2031_DDR_EMR0				
31:13	RSD[31:13]	R	0xXXXXX	These bits should be ignored when read and written as O's

Table 289: DDR SDRAM controller registers *...continued*

Bit	Symbol	Access	Value	Description
12:0	EMR[12:0]	R/W	0x0000	Extended Mode Register. Use the datasheet of the DDR memory to determine the value of this register. For emulation purposes it may be required to disable the DDL. To this end, make sure that bit 0 of this register contains a 1. In normal mode (non emulation) mode, make sure that bit 0 of this register contains a 0. The following is taken from a DDR datasheet and describes the different bits of the extended mode register. Bit 0: DLL (0:enable, 1:disable) Bit 1: drive strength (0:normal, 1:reduced) Bit 2: QFC mode. Bits 3 and up: operating mode.
Offset 0x88 - IP_2031_DDR_PRECHARGE_BIT0				
31:4	RSD[31:4]	R	0xXXXXXXX	These bits should be ignored when read and written as O's
3:0	PRECHARGE_BIT[3:0]	R/W	0xA	Column bit responsible for precharge. Only the values 0x8 (bit8) and 0xa (bit 10) are supported. The value for this field can be retrieved from the DDR memory data sheet.
Offset 0xC0 - IP_2031_RANK0_ROW_WIDTH0				
31:4	RSD[31:4]	R	0xXXXXXXX	These bits should be ignored when read and written as O's
3:0	RANK0_ROW_WIDTH[3:0]	R/W	0xD	Row dimensions: 2^ROW_WIDTH rows ie. A value of 0xC specifies 2^12 = 4096 rows. Only the following values are supported: 0x8, 0x9, 0xa, 0xb, 0xc, and 0xd (supporting 256 up to 8192 rows)
Offset 0xC4 - IP_2031_RANK0_COLUMN_WIDTH0				
31:4	RSD[31:4]	R	0xXXXXXXX	These bits should be ignored when read and written as O's
3:0	RANK0_COLUMN_WIDTH[3:0]	R/W	0xA	Column dimension: 2^COLUMN_WIDTH columns (each column has a width of 32 bit) ie a value of)xa specifies 2^10 = 1024 columns of 32 bit each. Only the following values are supported: 0x8, 0x9, 0xa and 0xb (supporting 256 up to 2048 columns).
Offset 0xD0 - IP_2031_RANK1_ROW_WIDTH0				
31:4	RSD[31:4]	R	0xXXXXXXX	These bits should be ignored when read and written as O's
3:0	RANK1_ROW_WIDTH[3:0]	R/W	0xD	Row dimensions: 2^ROW_WIDTH rows ie. A value of 0xC specifies 2^12 = 4096 rows. Only the following values are supported: 0x8, 0x9, 0xa, 0xb, 0xc, and 0xd (supporting 256 up to 8192 rows)

Table 289: DDR SDRAM controller registers *...continued*

Bit	Symbol	Access	Value	Description
Offset 0xD4 - IP_2031_RANK1_COLUMN_WIDTH0				
31:4	RSD[31:4]	R	0xXXXX XXX	These bits should be ignored when read and written as O's
3:0	RANK1_COLUMN_WIDTH[3:0]	R/W	0xA	Column dimension: 2^COLUMN_WIDTH columns (each column has a width of 32 bit) ie a value of)xa specifies 2^10 = 1024 columns of 32 bit each. Only the following values are supported: 0x8, 0x9, 0xa and 0xb (supporting 256 up to 2048 columns).
Offset 0x100 - IP_2031_DDR_TRCD0				
31:20	RSD[31:20]	R	0xXXX	These bits should be ignored when read and written as O's
19:16	TRCD_WR[3:0]	R/W	0x2	Minimum time between active and write command (RAS to CAS delay). When the datasheet of the DDR memory does not specify a value for this timing parameter, use the value specified for TRCD. The value for this timing parameter may not be smaller than TRAP of the DDR memory
15:4	RSD[15:4]	R	0xXXX	These bits should be ignored when read and written as O's
3:0	TRCD_RD[3:0]	R/W	0x4	Minimum time between active and read command (RAS to CAS delay). When the datasheet of the DDR memory does not specify a value for this timing parameter, use the value as specified for TRCD. The value for this timing parameter may not be smaller than TRAP of the DDR memory.
Offset 0x104 - IP_2031_DDR_TRC0				
31:5	RSD[31:5]	R	0xXXXX XXX	These bits should be ignored when read and written as O's
4:0	TRC[4:0]	R/W	0xXD	Minimum time between two active commands to the same bank
Offset 0x108 - IP_2031_DDR_TWTR0				
31:4	RSD[31:4]	R	0xXXXX XXX	These bits should be ignored when read and written as O's
3:0	TWTR[3:0]	R/W	0x1	Write to read command delay
Offset 0x10C - IP_2031_DDR_TWR0				
31:4	RSD[31:4]	R	0xXXXX XXX	These bits should be ignored when read and written as O's
3:0	TWR[3:0]	R/W	0x3	Write recovery time
Offset 0x110 - IP_2031_DDR_TRP0				
31:4	RSD[31:4]	R	0xXXXX XXX	These bits should be ignored when read and written as O's
3:0	TRP[3:0]	R/W	0x4	Precharge command period

Table 289: DDR SDRAM controller registers *...continued*

Bit	Symbol	Access	Value	Description
Offset 0x114 - IP_2031_DDR_TRAS0				
31:4	RSD[31:4]	R	0xXXXXXXX	These bits should be ignored when read and written as O's
3:0	TRAS[3:0]	R/W	0x9	Minimum delay from active to precharge
Offset 0x11C - IP_2031_DDR_TRRD0				
31:4	RSD[31:4]	R	0xXXXXXXX	These bits should be ignored when read and written as O's
3:0	TRRD[3:0]	R/W	0x2	Active bank a to active bank b command
Offset 0x120 - IP_2031_DDR_TRFC0				
31:5	RSD[31:5]	R	0xXXXXXXX	These bits should be ignored when read and written as O's
4:0	TRFC[4:0]	R/W	0xF	Auto refresh command period
Offset 0x124 - IP_2031_DDR_TMRD0				
31:4	RSD[31:4]	R	0xXXXXXXX	These bits should be ignored when read and written as O's
3:0	TMRD[3:0]	R/W	0x2	Load mode register command cycle time
Offset 0x128 - IP_2031_DDR_TCAS0				
31:4	RSD[31:4]	R	0xXXXXXXX	These bits should be ignored when read and written as O's
3:0	TCAS[3:0]	R/W	0x8	CAS read latency specified in half cycles ie a value of 0b0111 (7) represents a CAS delay of 3,5 cycles (7 half cycles)
Offset 0x12C - IP_2031_DDR_RF_PERIOD0				
31:16	RSD[31:16]	R	0xXXXX	These bits should be ignored when read and written as O's
15:0	RF_PERIOD[15:0]	R/W	0x0DBB	Refresh period expressed in terms of cycles. Typically a refresh is required at an average interval of 15.625us. For a 100MHz, device this translates into a RF_PERIOD value of 1562. For a 200MHz device this translates into a RF_PERIOD value of 3125.
Offset 0x180 - IP_2031_ARB_CTL0				
31	CPU_DMA_DECR	R/W	0x1	0: Do not decrement CPU counters 1, 2, and 3, when in a DMA_WINDOW. 1: Do decrement CPU counters 1, 2 and 3, when in a DMA_WINDOW.

Table 289: DDR SDRAM controller registers *…continued*

Bit	Symbol	Access	Value	Description
30	CPU_HRT_SRT_ENABLE	R/W	0x0	0: Disable the effect of the signal polarity change on mtl(n)_cmd_hrt or ext_hrt_pending inputs. (in other words, controller would interpret that DMA port contains only Hard Real Time DMA request). 1: Enable the effects of the signal polarity change on mtl(n)_cmd_hrt or ext_hrt_pending inputs. (in other words, controller would interpret that DMA port contains Hard Real Time or soft real time DMA request based on signal polarity on mtl(n)_cmd_hrt or ect_hrt_pending inputs).
29	BLB_ENABLE	R/W	0x0	0: Disable Back Log Buffer 1: Enable Back Log Buffer
28	DYN_RATIOS	R/W	0x0	0: Use static ratios. this means accounts are incremented by the value (RATIO+DDR burst size (in terms of cycles)) wherever a CPU DDR burst is performed 1: Enable dynamic ratios. This means accounts are incremented by the value ratio every clock cycle that is spent on servicing a CPU DDR burst. This feature also causes account not to decrement during clock cycles that are spent on CPU DDR bursts.
27:18	RSD[27:18]	R	0xXXX	These bits should be ignored when read and written as O's
17:16	CPU_PREEMPT[1:0]	R/W	0x1	0: No preemption (once a CPU MTL command has started to enter the DDR arbitration buffer, it will go completely into the DDR arbitration buffer, uninterrupted by other (CPU or DMA) MTL commands. 1: Preempt a CPU MTL command when it started to enter the DDR arbitration buffer while inside the DMA window and is currently active in the DMA window. The CPU MTL command will only be interrupted by a DMA MTL command, not by another CPU MTL command. 10: Undefined 11: Preempt a CPU MTL command that is currently active in the DMA window (independent of when it started to enter the DDR arbitration buffer). The CPU MTL command will only be interrupted by a DMA MTL command, not by another CPU MTL command.
15:2	RSD[15:2]	R	0xXXXX	These bits should be ignored when read and written as O's

Table 289: DDR SDRAM controller registers *...continued*

Bit	Symbol	Access	Value	Description
1:0	DMA_PREEMPT[1:0]	R/W	0x1	0: No preemption (once a DMA MTL command has started to enter the DDR arbitration buffer, it will go completely into the DDR arbitration buffer, uninterrupted by other (CPU or DMA) MTL commands. 1: Preempt a DMA MTL command when it started to enter the DDR arbitration buffer while inside the CPU window and is currently active in the CPU window. The DMA MTL command will only be interrupted by a CPU MTL command, not by another DMA MTL command 10: Undefined 11: Preempt a DMA MTL command that is currently active in the CPU window (independent of when it started to enter the DDR arbitration buffer). The DMA MTL command will only be interrupted by a CPU MTL command, not by another DMA MTL command.
Offset 0x184 - IP_2031_ARB_HRT_WINDOW0				
31:16	RSD[31:16]	R	0xXXXX	These bits should be ignored when read and written as O's
15:0	WINDOW_HRT[15:0]	R/W	0x003F	Window for hard real-time (HRT) MTL requests (in terms of clock cycles). Add 1 for the real effective window size.
Offset 0x188 - IP_2031_ARB_CPU_WINDOW0				
31:16	RSD[31:16]	R	0xXXXX	These bits should be ignored when read and written as O's
15:0	WINDOW[15:0]	R/W	0x003F	Window for Central Processor Unit (CPU) MTL requests (in terms of clock cycles). Add 1 for real effective window size.
Offset 0x1C0 - IP_2031_ARB_CPU_LIMIT0				
31:16	RSD[31:16]	R	0xXXXX	These bits should be ignored when read and written as O's
15:0	CPU_LIMIT[15:0]	R/W	0xFFFF	When the DDR controller internal CPU counter exceeds this value, no CPU DDR burst will be performed when DMA traffic is present (CPU traffic has lower priority than DMA traffic). The internal CPU account is decremented by DECR every clock cycle. For increment information see DYN_RATIOS description.
Offset 0x1C4 - IP_2031_ARB_CPU_RATIO0				
31:8	RSD[31:8]	R	0xXXXXXX	These bits should be ignored when read and written as O's
7:0	RATIO[7:0]	R/W	0x00	If DYN_RATIOS are disabled the value is added to the internal account for each CPU DDR burst. If DYN_RATIOS are enabled then this value is added to the internal account for each clock cycle spent on a CPU DDR burst.
Offset 0x1C8 - IP_2031_ARB_CPU_CLIP0				

Table 289: DDR SDRAM controller registers *...continued*

Bit	Symbol	Access	Value	Description
31:16	RSD[31:16]	R	0xXXXX	These bits should be ignored when read and written as o's
15:0	CPU_CLIP[15:0]	R/W	0xFFFF	CPU account clip. When the internal account goes above this value the CPU DDR bursts are for free. This value should always be equal or higher then LIMIT.
Offset 0x1CC - IP_2031_ARB_CPU_DECR0				
31:8	RSD[31:8]	R	0xXXXXXX	These bits should be ignored when read and written as 0's
7:0	CPU_DECR[7:0]	R/W	0x01	CPU account decrement. This value is used to decrement the internal account each clock cycle (with some exceptions).
Offset 0x1D0 - IP_2031_ARB_CPU1_LIMIT0				
31	CPU1_VSDMA	R/W	0x0	0: Disable Versus DMA 1: Enable Versus DMA. IF the internal account is above LIMIT then DMA gets priority over CPU1
30	CPU1_HIGH	R/W	0x0	0: Low priority 1: High Priority
29:16	RSD[29:16]	R	0xXXXX	These bits should be ignored when read and written as O's
15:0	CPU1_LIMIT[15:0]	R/W	0xFFFF	CPU1 account limit
Offset 0x1D4 - IP_2031_ARB_CPU1_RATIO0				
31:8	RSD[31:8]	R	0xXXXXXX	These bits should be ignored when read and written as O's
7:0	CPU1_RATIO[7:0]	R/W	0x00	CPU1 account ratio
Offset 0x1D8 - IP_2031_ARB_CPU1_CLIP0				
31:16	RSD[31:16]	R	0xXXXX	These bits should be ignored when read and written as 0's
15:0	CPU1_CLIP[15:0]	R/W	0xFFFF	CPU1 account clip
Offset 0x1DC - IP_2031_ARB_CPU1_DECR0				
31:8	RSD[31:8]	R	0xXXXXXX	These bits should be ignored when read and written as 0's
7:0	CPU1_DECR[7:0]	R/W	0x01	CPU1 account decrement.
Offset 0x1E0 - IP_2031_ARB_CPU2_LIMIT0				
31	CPU2_VSDMA	R/W	0x0	0: Disable Versus DMA 1: Enable Versus DMA. IF the internal account is above LIMIT then DMA gets priority over CPU2
30	CPU2_HIGH	R/W	0x0	0: Low priority 1: High Priority
29:16	RSD[29:16]	R	0xXXXX	These bits should be ignored when read and written as O's
15:0	CPU2_LIMIT[15:0]	R/W	0xFFFF	CPU2 account limit
Offset 0x1E4 - IP_2031_ARB_CPU2_RATIO0				

Table 289: DDR SDRAM controller registers *...continued*

Bit	Symbol	Access	Value	Description
31:8	RSD[31:8]	R	0xXXXXXX	These bits should be ignored when read and written as O's
7:0	CPU2_RATIO[7:0]	R/W	0x00	CPU2 account ratio
Offset 0x1E8 - IP_2031_ARB_CPU2_CLIP0				
31:16	RSD[21:16]	R	0xXXXX	These bits should be ignored when read and written as 0's
15:0	CPU2_CLIP[15:0]	R/W	0xFFFF	CPU2 account clip
Offset 0x1EC - IP_2031_ARB_CPU2_DECR0				
31:8	RSD[31:8]	R	0xXXXXXX	These bits should be ignored when read and written as 0's
7:0	CPU2_DECR[7:0]	R/W	0x01	CPU2 account decrement
Offset 0x1F0 - IP_2031_ARB_CPU3_LIMIT0				
31	CPU3_VSDMA	R/W	0x0	0: Disable Versus DMA 1: Enable Versus DMA. If the internal account is above LIMIT then DMA gets priority over CPU 3
30	CPU3_HIGH	R/W	0x0	0: Low priority 1: High priority
29:16	RSD[29:16]	W	0xXXXX	These bits should be ignored when read and written as O's
15:0	CPU3_LIMIT[15:0]	R/W	0xFFFF	CPU3 account limit
Offset 0x1F4 - IP_2031_ARB_CPU3_RATIO0				
31:8	RSD[31:8]	R	0xXXXXXX	These bits should be ignored when read and written as O's
7:0	CPU3_RATIO[7:0]	R/W	0x00	CPU3 account ratio
Offset 0x1F8 - IP_2031_ARB_CPU3_CLIP0				
31:16	RSD[31:16]	R	0xXXXX	These bits should be ignored when read and written as 0's
15:0	CPU3_CLIP[15:0]	R/W	0xFFFF	CPU3 account clip
Offset 0x1FC - IP_2031_ARB_CPU3_DECR0				
31:8	RSD[31:8]	R	0xXXXXXX	These bits should be ignored when read and written as 0's
7:0	CPU3_DECR[7:0]	R/W	0x01	CPU3 account decrement
Offset 0x200 - IP_2031_PF_MTL0_RD_VALID0				
31:0	MTL0_RD_VALID[31:0]	R/W	0xXXXXXXXX	Counter for valid MTL0 read data elements
Offset 0x204 - IP_2031_PF_MTL0_WR_ACCEPT0				
31:0	MTL0_WR_ACCEPT[31:0]	R/W	0xXXXXXXXX	Counter for valid MTL0 write data elements
Offset 0x208 - IP_2031_PF_MTL1_RD_VALID0				
31:0	MTL1_RD_VALID[31:0]	R/W	0xXXXXXXXX	Counter for valid MTL1 read data elements
Offset 0x20C - IP_2031_PF_MTL1_WR_ACCEPT0				

Table 289: DDR SDRAM controller registers *...continued*

Bit	Symbol	Access	Value	Description
31:0	MTL1_WR_ACCEPT[31:0]	R/W	0xXXXXXXXX	Counter for valid MTL1 write data elements
Offset 0x210 - IP_2031_PF_MTL2_RD_VALID0				
31:0	MTL2_RD_VALID[31:0]	R/W	0xXXXXXXXX	Counter for valid MTL2 read data elements
Offset 0x214 - IP_2031_PF_MTL2_WR_ACCEPT0				
31:0	MTL2_WR_ACCEPT[31:0]	R/W	0xXXXXXXXX	Counter for valid MTL2 write data elements
Offset 0x218 - IP_2031_PF_MTL3_RD_VALID0				
31:0	MTL3_RD_VALID[31:0]	R/W	0xXXXXXXXX	Counter for valid MTL3 read data elements
Offset 0x21C - IP_2031_PF_MTL3_WR_ACCEPT0				
31:0	MTL3_WR_ACCEPT[31:0]	R/W	0xXXXXXXXX	Counter for valid MTL3 write data elements
Offset 0x240 - IP_2031_PF_IDLE0				
31:0	IDLE[31:0]	R/W	0xXXXXXXXX	Counts cycles in which DDR memory controller is considered to be idle (not valid entries on the top of the DDR arbitration queue. The same idle information is present on output signal ip_2031_idle
Offset 0x280 - IP_2031_ERR_VALID0				
31:1	RSD[31:1]	R	0xXXXXXXXX	These bits should be ignored when read and written as O's
0	VALID	R/W	0x0	The value of this bIt Is reflected on the ip_2031_ddr_addr_err output pin. Writing a 0 to this bit will make ip_2031_ddr_addr_err '0' as well. This can be used to acknowledge the error indication. 0: no error 1: error
Offset 0x284 - IP_2031_ERR_MTL_PORT0				
31:2	RSD[31:2]	R	0xXXXXXXXX	These bits should be ignored when read and written as O's
1:0	MTL_PORT[1:0]	R	0xX	MTL port that caused the error
Offset 0x288 - IP_2031_ERR_MTL_CMD_ADDR0				
31:0	MTL_CMD_ADDR[31:0]	R	0xXXXXXXXX	MTL command address. If the MTL address is crossed the total addressable range boundary within a transaction, it would store the address of the split MTL command within the controller and not the starting address of MTL command request.
Offset 0x28C - IP_2031_ERR_MTL_CMD_READ0				
31:1	RSD[31:1]	R	0xXXXXXXXX	These bits should be ignored when read and written as O's
0	MTL_CMD_READ	R	0xX	MTL command read
Offset 0x290 - IP_2031_ERR_MTL_CMD_ID0				

Table 289: DDR SDRAM controller registers *…continued*

Bit	Symbol	Access	Value	Description
31:4	RSD[31:4]	R	0xXXXXXXX	These bits should be ignored when read and written as O's
3:0	MTL_CMD_ID[3:0]	R	0xX	MTL command identifier
Offset 0xFFC - IP_2031_MODULE_ID0				
31:16	MODULE_ID[15:0]	R	0x2031	DDR memory controller module ID
15:12	MAJOR_REV[3:0]	R	0x1	Major revision number
11:8	MINOR_REV[3:0]	R	0x1	Minor revision number
7:0	APERTURE[7:0]	R	0x00	Aperture size is 4KByte ((APERTURE+1)*4KByte)

5.8 CTL12 Tunnel

5.8.1 Introduction

The CTL12 tunnel is a high performance, low pin count, point to point connection for interfacing between two devices. It consists of two sides, a north tunnel (SAT_QTUN_IN) which is on the north device and a south tunnel (SAT_QTUN_OUT) on the south device. The north and south tunnels are slightly different in design. Therefore a north tunnel must be connected to a south tunnel.

The tunnel interface consists of two independent transmit and receive connections, each one providing data flow in one direction only. It provides MMIO control of the south device from the north device. Additionally, it allows streaming peripherals in the south device high bandwidth access to the memory hierarchy of the north device. The transmit clocks are source synchronous and allow data to be received without the use of a DLL.

The key features of the tunnel interface are:

- High-speed chip interconnect
- Transparent interconnect
- 256 MB/s peak transmit & 256 MB/s peak receive @ 192 MHz
- 300 MB/s peak transmit & 300 MB/s peak receive @ 225 MHz
- 38 interface pins
- DTL, MTL compatible

5.8.2 Functional description

The CTL12 tunnel supports two independent streams of data. A Memory Transaction Level (MTL) direct memory access is provided from the south device to the north device. This stream is intended for use by DMA devices that transfer data to and from memory. It supports burst mode reads and writes. This requires sending command packets and write data packets from the south to the north, with return status and read data returning from the north to the south.

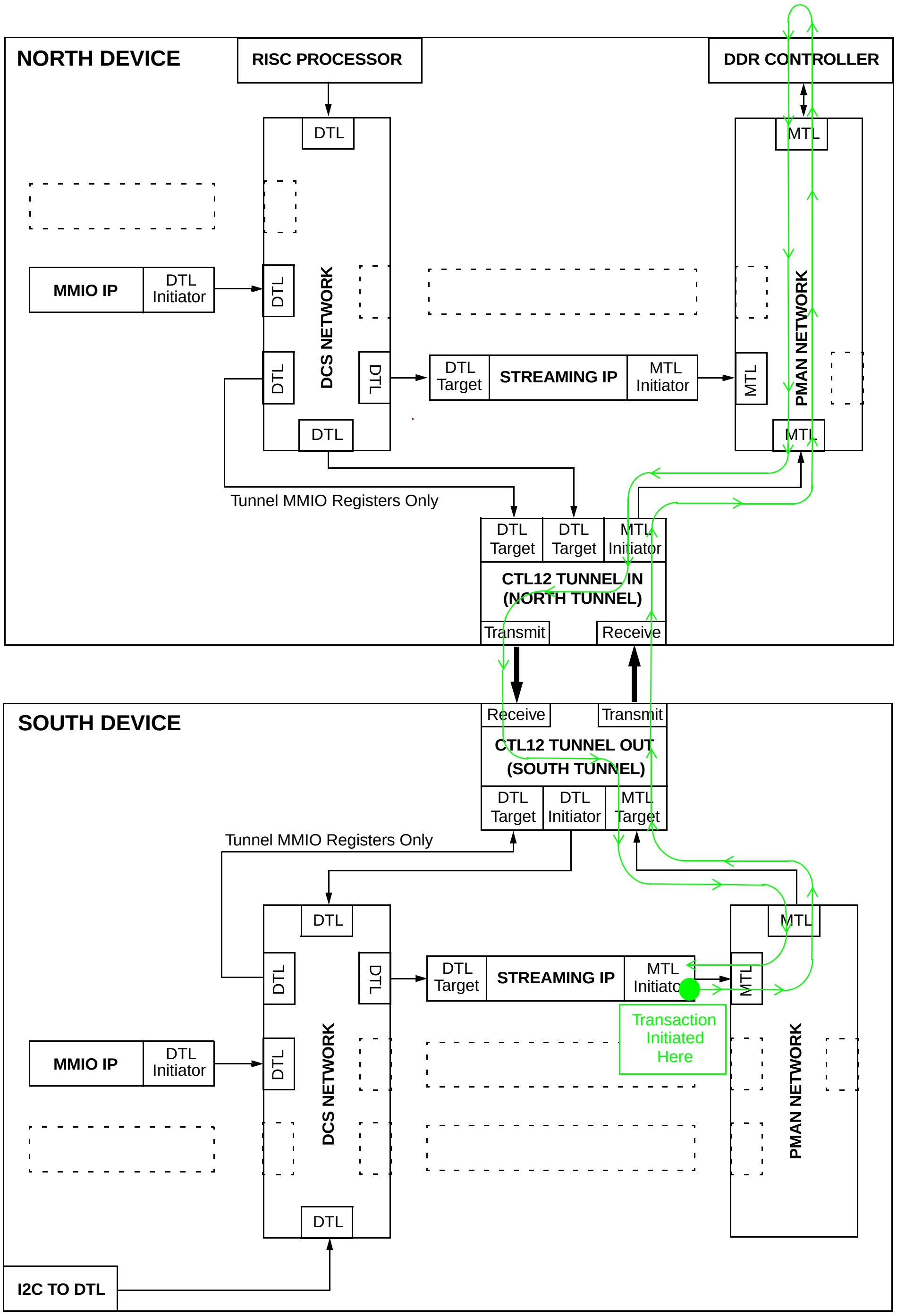


Fig 199.MTL (Direct Memory Access) transaction

An additional stream is supported in the tunnel. It is a Memory Mapped IO (MMIO) stream. This is intended for control, configuration, and status checks for low bandwidth IO ports and is typically used to control the devices that are generating the requests on the MTL. The MMIO interface can also be used by devices that do not require any direct memory access. The MMIO interface transfers commands and write data from the north to the south, and transfers read data and status from the south to the north. These two interfaces, the MTL and the MMIO, appear as completely independent interfaces to users of the CTL12 tunnel.

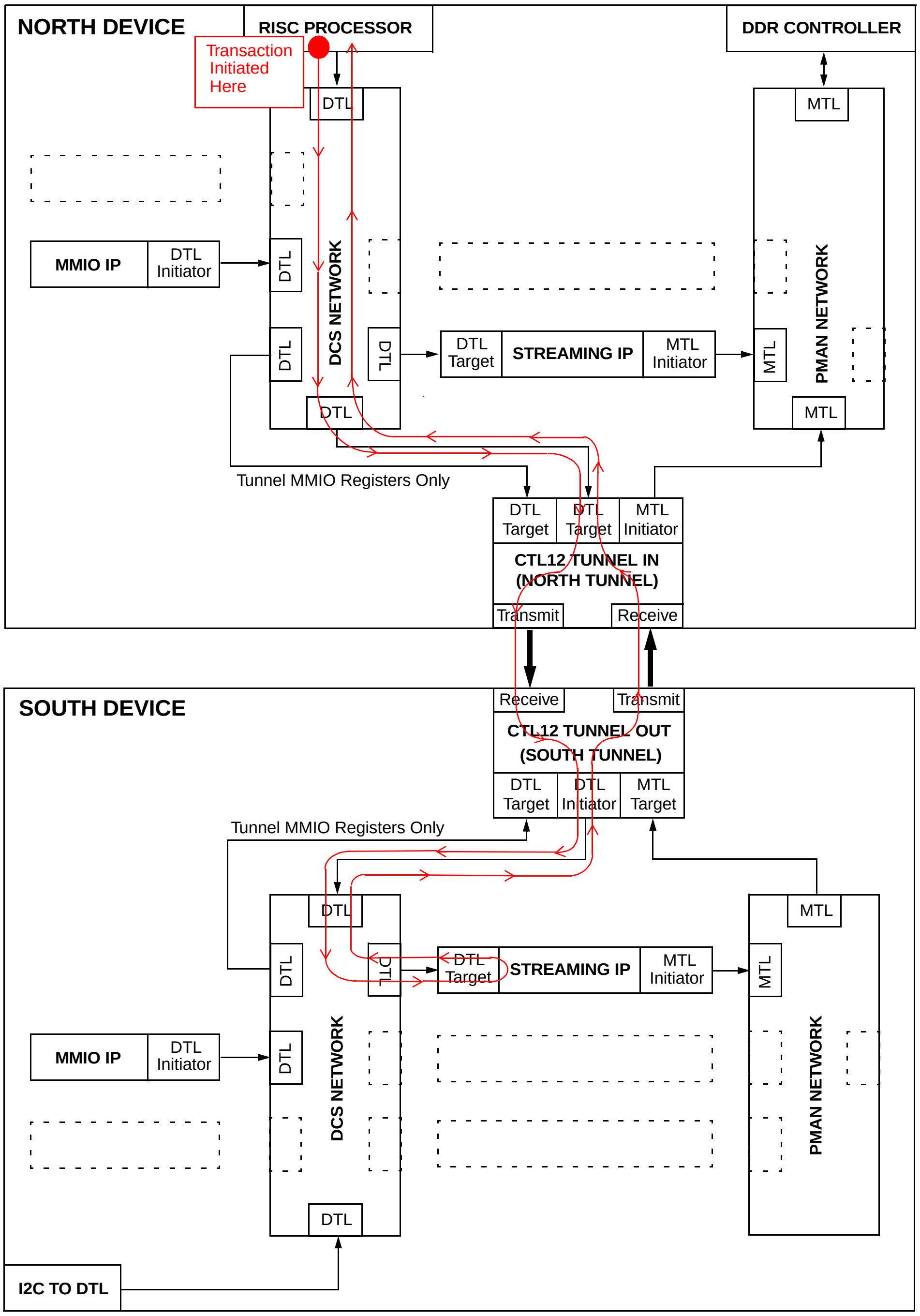


Fig 200.MMIO transaction

5.8.3 Tunnel design

5.8.3.1 CTL12 tunnel overview

The south tunnel (CTL12 tunnel out) block serves the initiators and targets in the south device allowing them to send and receive data from the memory hierarchy and the processor. The north tunnel (CTL12 tunnel in) block serves the initiators and targets in the north device.

Figure 201 shows how the command/data flows over the CTL12 Tunnel.

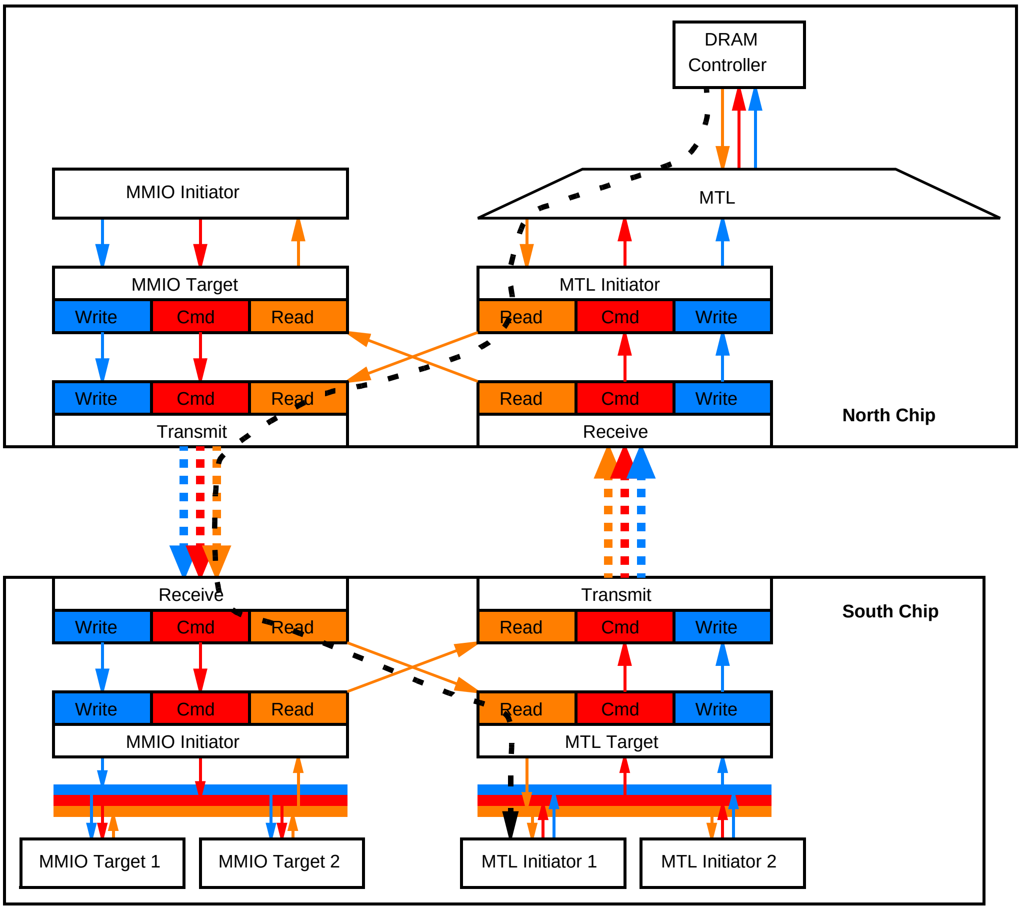


Fig 201.CTL12 tunnel communications paths

The dotted black line in the figure above represents the path for the read data returning to MTL Initiator 1.

5.8.3.2 CTL12 tunnel transmit and receive overview

Figure 202 provides an overview of the transmit and receive sections from the FIFOs, through the muxes and encode, and then through the decode and back into the corresponding FIFOs.

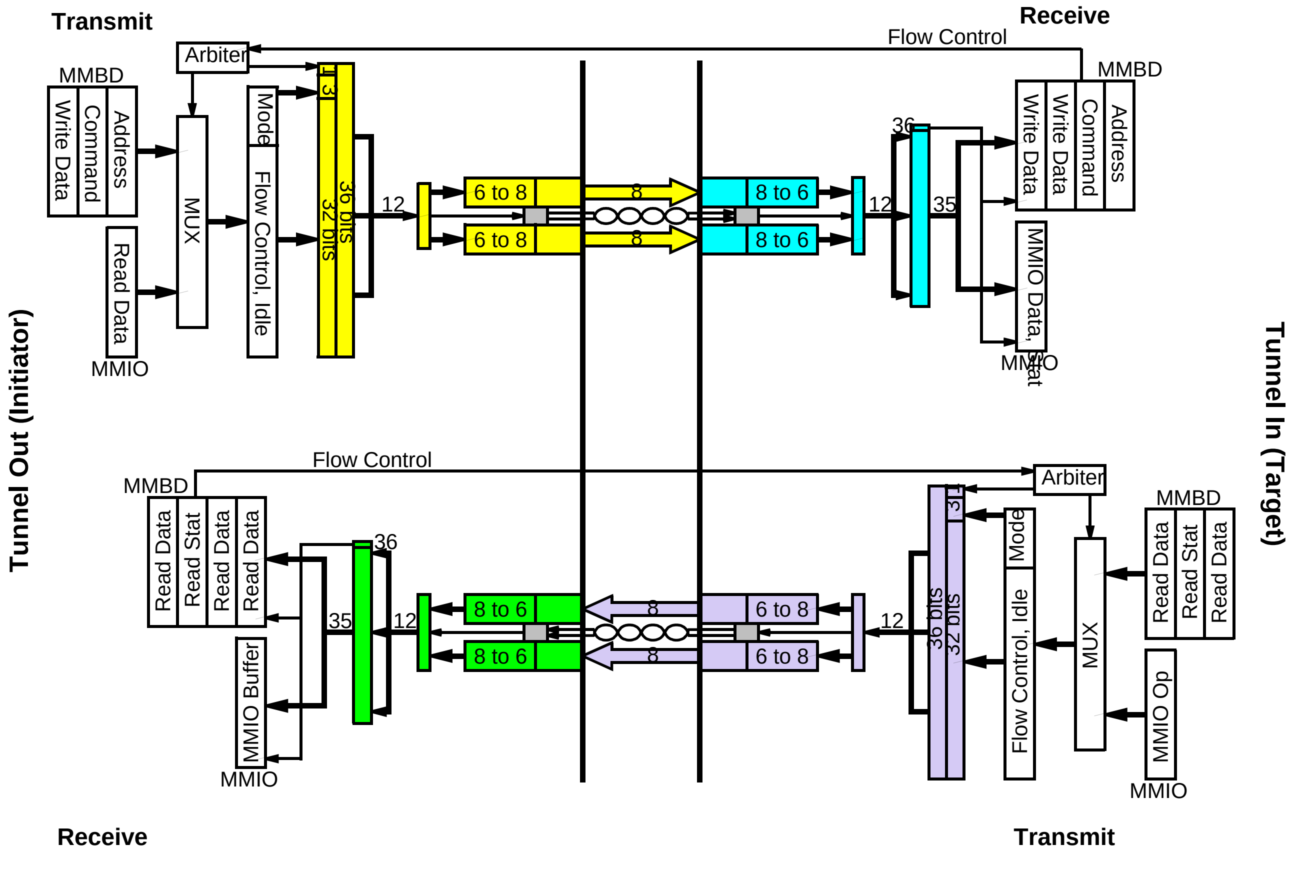


Fig 202.CTL12 tunnel transmit/receive architecture

5.8.3.3 Multiplexing Overview

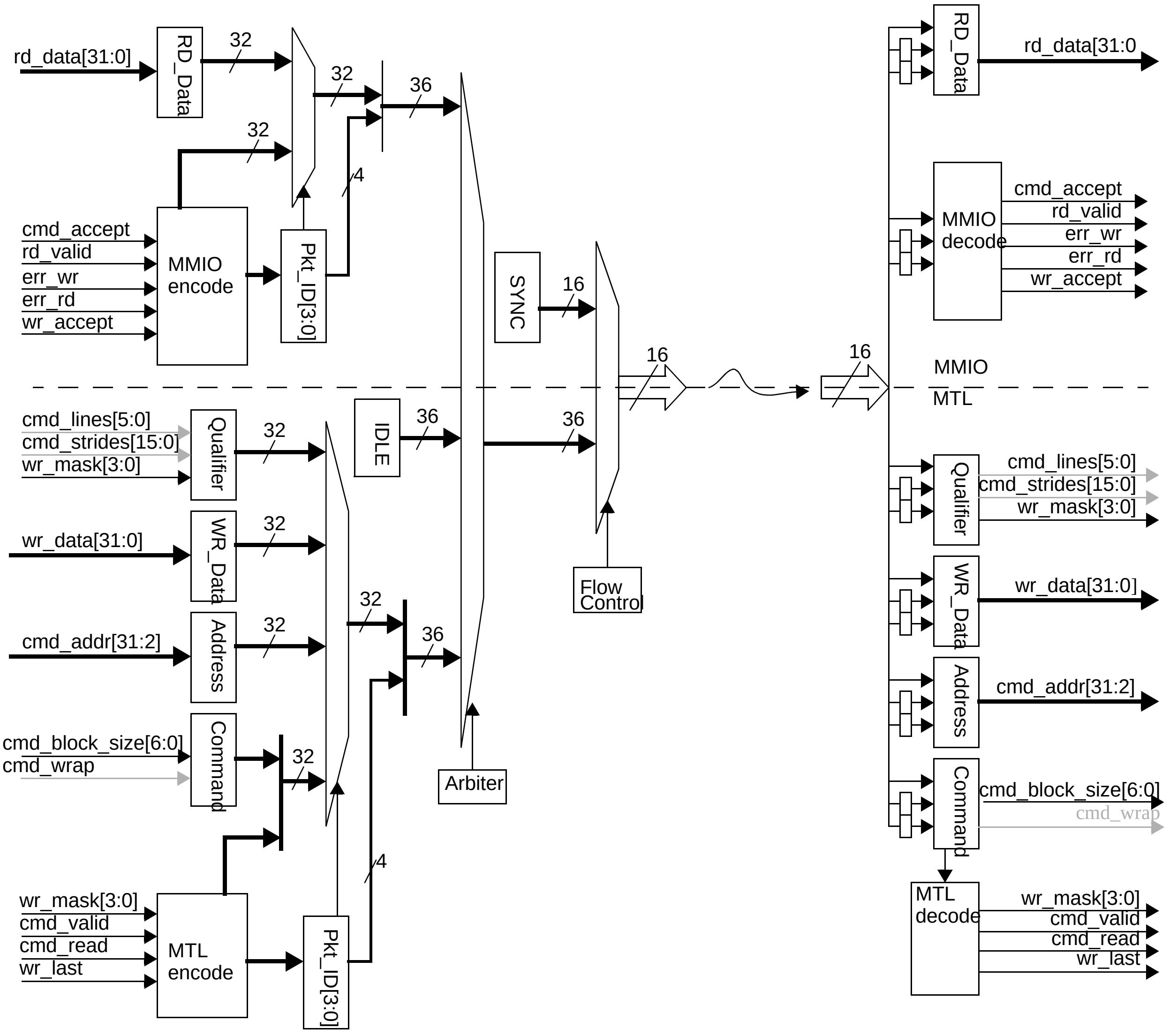


Fig 203.MTL Wr data / MMIO rd data path

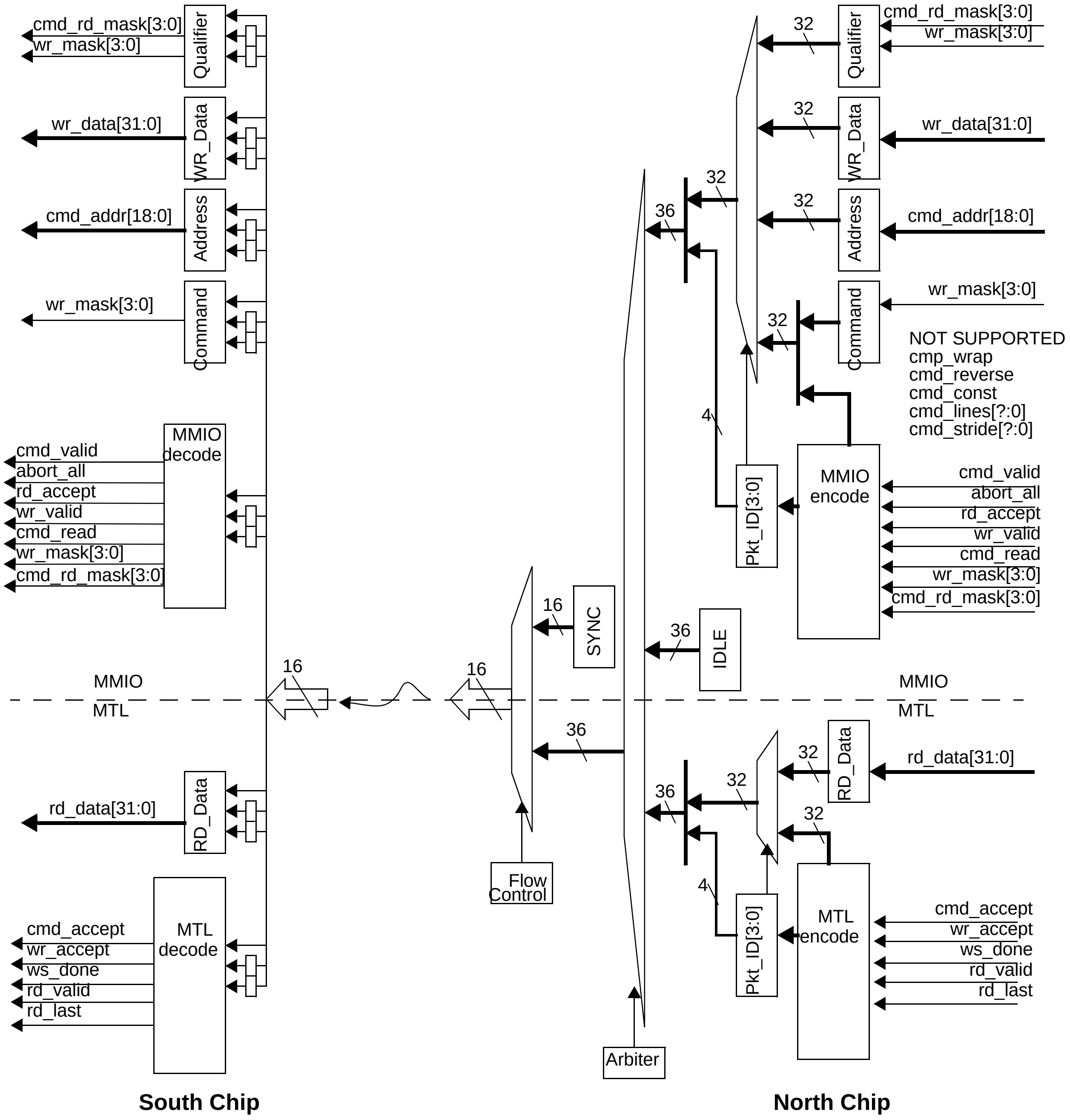


Fig 204.MTL Rd data / MMIO wr data path

5.8.3.4 CTL12 Tunnel Flow Control

The CTL12 tunnel uses separate transmit and receive paths. The receive path contains a FIFO that stores the received packets. For ease of design, the flow control over the tunnel is handled by side band signals. These signals tell the transmit side to stop transmitting because the receiving FIFO is "full." Since the tunnel is a source synchronous interface, the flow control signals are actually signalling almost full. This is needed because there are outstanding packets that have already been transmitted and can’t be stopped due to the source synchronous interface. Figure 205 shows the path of the flow control signals.

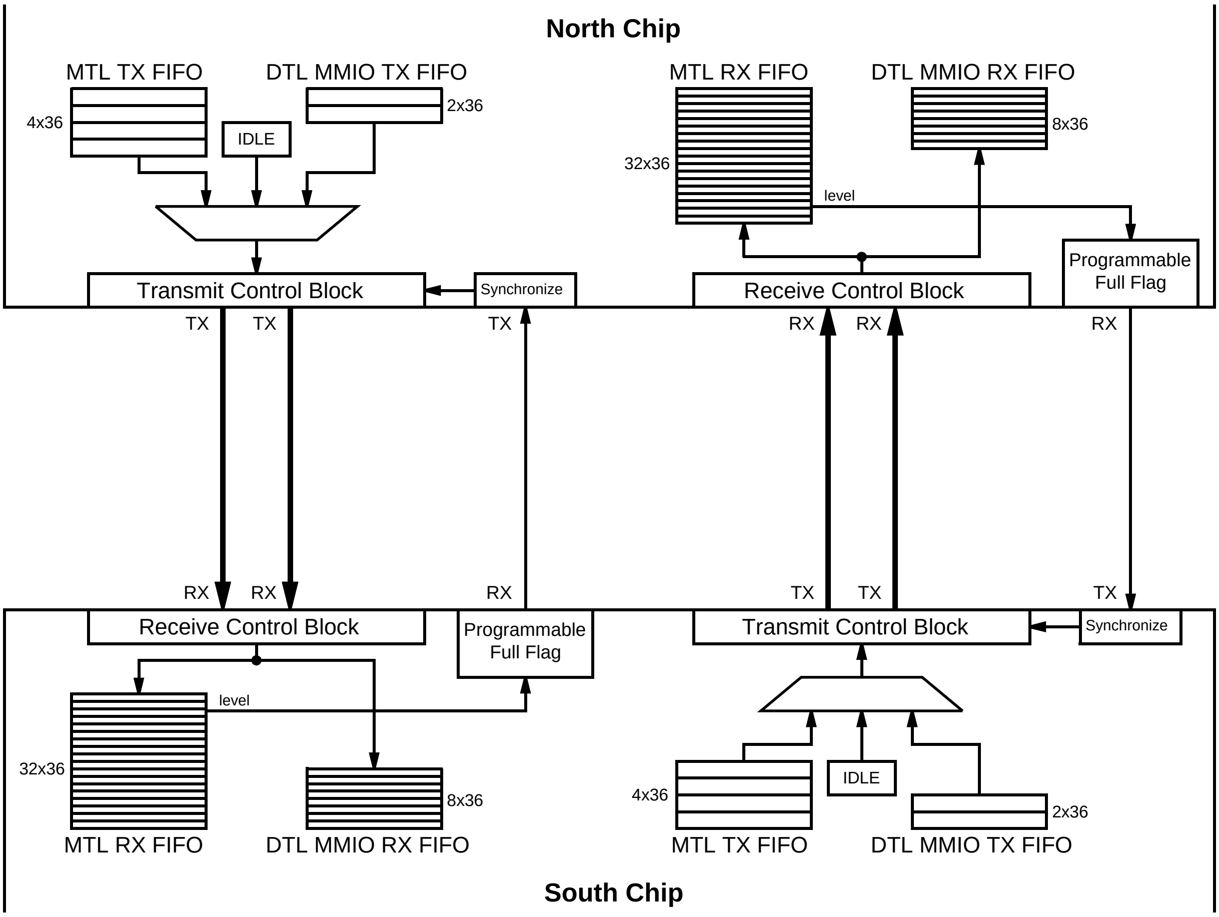


Fig 205.Flow control signaling

The flow control is only used for the MTL traffic because the MMIO traffic never overflows its receive FIFO due to the inherent design of the tunnel blocks. To provide the correct size receive FIFO for the MTL traffic, several things need to be considered. The number of outstanding packets, when "full" is indicated, depends on the following:

- Latency of the level indicator through the programmable full flag block.
- Latency over the PCB.
- Latency through the synchronizer.
- Latency for stopping transmission on a packet boundary.
- Latency of data transmitted from the transmit control block to the receive control block.
- Operating frequency of the tunnel.

To use the tunnel effectively, the receive FIFO needs to be at least 16 deep, preferably 32. (This is an asynchronous FIFO, so the depth has to be powers of 2). This way, if the FIFO is 32 deep, the full flag would be raised when the FIFO is at 75% capacity.

Any of the factors above have an impact on the number of outstanding packets. The programmable full flag block allows the "full" condition to be triggered at different FIFO levels, thus allowing for some system modification without having to change the receiving FIFO.

In a system where the tunnel's MTL port has sufficient bandwidth, the receiving FIFO will never become full, thus the flow control signals will never go active. If the MTL port gets waited/delayed, the receiving FIFO provides a small buffering effect. However, if the MTL port gets waited/delayed too long, the flow control signals for the tunnel will activate.

5.8.3.5 Write transaction

Figure 206 is an example of a write transaction. The write request packets are sent from the south device to the north device and consist of an address, a command packet and all of the data packets. Once that is processed, the north device transmits a status packet indicating success or failure.

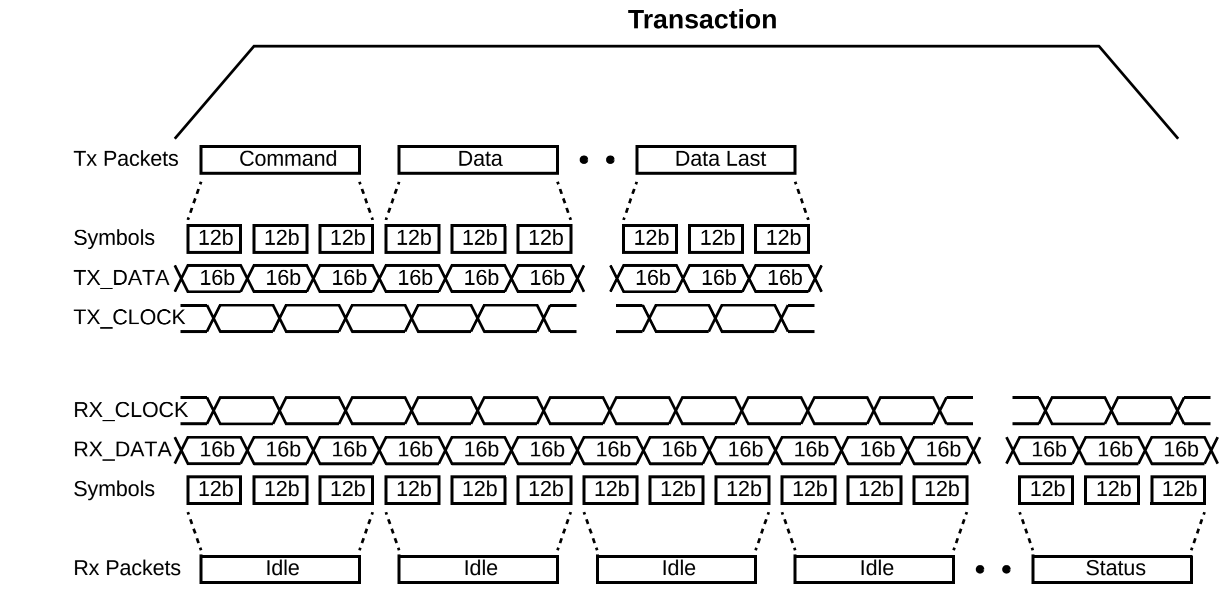


Fig 206.Write transaction example

The CTL12 packets contain three Symbols per packet. The 12-bit Symbols are encoded through two 6b8b encoders to provide 16 bits. These 16 bits contain an equal number of 1's and 0's for DC balancing. They are transmitted source synchronously.

5.8.3.6 Read transaction

Figure 207 is an example of a read transaction. The address and command packets are sent from the south device to the north device. Once that is processed, the read data packets are transmitted back from the north to the south tunnel. A status packet only returns when there is a read error. A read last packet indicates a successful transfer.

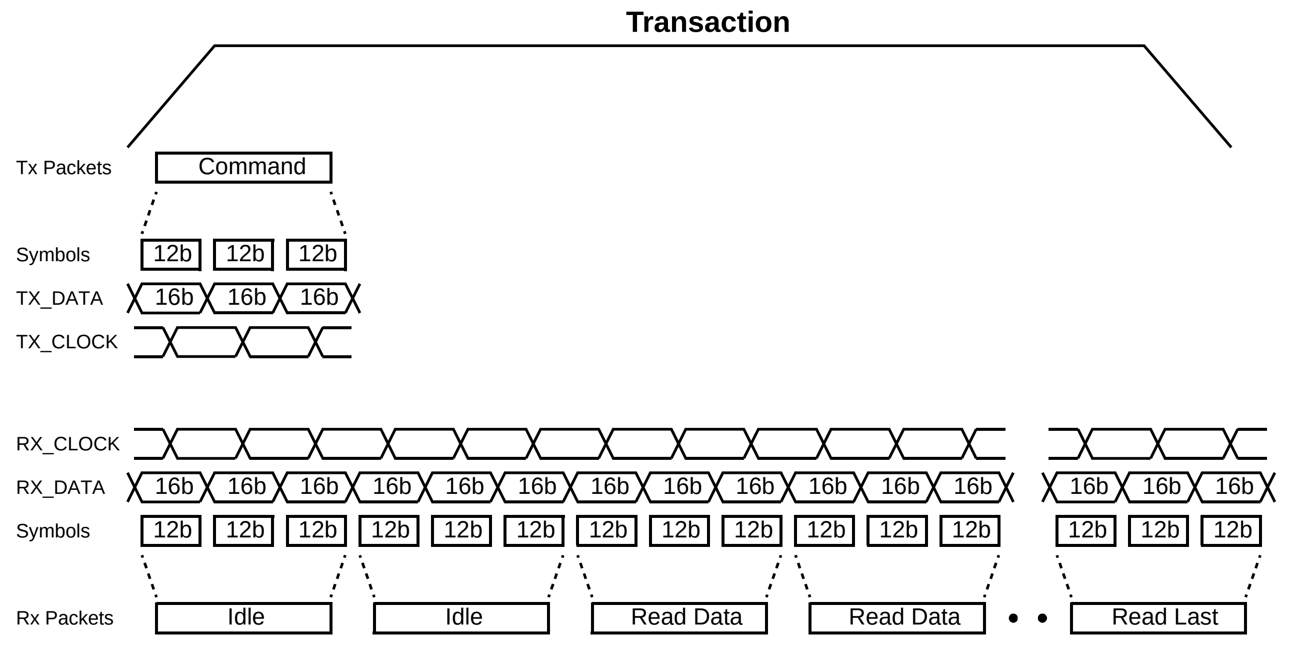


Fig 207.Read transaction example

5.8.3.7 Ordering

The MTL protocol contains no ordering requirements between the read and write data transfers over a given MTL connection. Since the CTL12 tunnel is highly pipelined structure, it is possible for read commands to pass write commands. If there is a write-read combination that is dependent, it is highly recommended that the read command is issued AFTER the write data has been successfully written (via the mtl_wr_done signal).

5.8.3.8 Tunnel traffic

MTL Traffic - All MTL traffic is initiated from the south device. The south tunnel (CTL12 Tunnel Out) block transfers all MTL traffic that arrives at its target interface in the south device, across the tunnel to the MTL initiator interface in the north device, with no address modification. The north bound MTL traffic as well as the resultant data and status is transferred with no filtering or modification.

DCS Traffic - All DCS traffic is initiated from the north device. It is sent through the main tunnel MMIO aperture. The CTL12 Tunnel design makes no assumptions about the size of this aperture or the number of targets on the south side of the tunnel. For the PNX2015, the aperture size within the overall PNX8550 MMIO register map is 512 KB. If the tunnel is initialised, all accesses to this aperture are sent across to the south tunnel. The south device will send back data/command packets from valid accesses or an error. If the CTL12 Tunnel is disabled, it returns errors on writes and 0xDEADABBA on reads. The only exception is the module ID address, where a read returns a "null" module ID and indicates a size of 512 KB.

Tunnel Configuration Registers - Both the north and the south tunnel have a DTL MMIO Target which has a 4K aperture. This target is used for accessing the tunnel configuration registers which are used for initialisation, configuration and test. Unused locations in this aperture return an error for writes and 0xDEADABBA for reads.

5.8.3.9 Bandwidth

Table 290 represents the maximum throughput for the CTL12 Tunnel. It assumes no overhead and no MMIO traffic. There is always overhead in the form of address, command, qualifier, and status packets. The typical burst size determines the number of overhead packets. The smaller the bursts, the more overhead there is. In a real system, the bandwidth is always lower than the maximum.

Table 290: Burst transfer rates

Reference clock (MHz)	Symbol width (bits)	Pin count (balls)	Data strobes (MHz)	Peak read data rate (MB/s)	Peak write data rate (MB/s)	Peak combined data rate (MB/s)
192	12	39	96	256	256	512
225	12	39	112.5	300	300	600

How much lower the bandwidth is depends on several things, such as:

- Burst size
- MMIO traffic
- Write qualifiers

The values represented here have been simplified and are provided as a reference, not as absolute values. The tunnel bandwidth is largely going to depend on the entire system.

Doing write burst transfers of 128 bytes decreases the bandwidth from the south to the north device by about 6% of maximum, with no impact on the transmit bandwidth.

32 bits data = 4 bytes data = 1 tunnel packet

MTL Data pkts: 128 bytes = 32 tunnel packets

MTL Overhead: 2 packets = 1 address packets, 1 command packet (write mtl)

$$\text{Efficiency} = \frac{\text{MTL Data pkts}}{\text{MTL Data pkts + MTL Overhead}} = \frac{32}{32+2} = 94\%$$

Initiating read bursts will decreases the south to north bandwidth further. The read requests (address and command packet) will be dispersed among the write bursts. Logically, doing many small read bursts is less efficient than doing a few large read bursts. Assuming an equal number of read and write burst transfers of 128 bytes, the south to north bandwidth efficiency is around 89%.

MTL Data pkts: 128 bytes = 32 tunnel packets

MTL Overhead: 4 packets = 1 address packet, 1 command packet (read mtl), 1 address packet, 1 command packet (write mtl)

$$\text{Efficiency} = \frac{\text{MTL Data pkts}}{\text{MTL Data pkts + MTL Overhead}} = \frac{32}{32+4} = 89\%$$

The north to south device bandwidth is not impacted by read bursts since all returning packets are read data packets. These efficiency values assume no MMIO traffic or qualifier packets. The following section examines what happens to the bandwidths with MMIO traffic flowing throughout the 128 byte read and write bursts.

At most, the MMIO traffic can send 18 packets for every 180 MTL packets from the south to the north device, thus decreasing the efficiency of the bandwidth to about 81%.

180 MTL pkts = (32 MTL Data pkts x 5 = 160) + (4 MTL Overhead x 5 = 20)

$$\text{Efficiency} = \frac{\text{MTL Data pkts}}{\text{MTL Data pkts + MTL Overhead + MMIO}} = \frac{160}{160 + 20 + 18} = 81\%$$

This is worst case, since it is assuming that there is zero turn around time on the MMIO ports. In most cases, the receiving efficiency is higher.

The north to the south tunnel efficiency can only be impacted by MMIO traffic. This is because all MTL traffic flowing in this direction is read data. The MMIO traffic can send 6 packets for every 16 MTL packets to the south tunnel. In this case, the transmit efficiency is around 73%.

$$\text{Efficiency} = \frac{\text{MTL Data pkts}}{\text{MTL Data pkts + MMIO}} = \frac{16}{16 + 6} = 73\%$$

In almost all cases, the north to south efficiency is higher.

The qualifier packets have no impact on north to south tunnel efficiency. They only impact the south to north tunnel efficiency and are used for write masking, 2D transfers and unaligned transfers. Write qualifiers should be used sparingly. If all traffic is aligned, then the south to north tunnel efficiency is not impacted. However, it is possible to send a write qualifier packet with each write data packet. This decreases the south to north tunnel efficiency to 47%, assuming 128 byte read/write bursts.

MTL Data pkts: 128 bytes = 32 tunnel packets

MTL Overhead: 4 = 1 address packet, 1 command packet (read mtl), 1 address packet, 1 command packet (write mtl)

Qualifiers: 32 Qualifier packets.

$$\text{Efficiency} = \frac{\text{MTL Data pkts}}{\text{MTL Data Pkts + Overhead + Qualifiers}} = \frac{32}{32 + 4 + 32} = 47\%$$

5.8.3.10 Latency

The CTL12 Tunnel is a pipelined structure. This section describes the typical delay through the Tunnel. For MMIO, the delay is measured from the DTL interface to the Tunnel pads. For MMBD, the delay is measured from the internal interface of the MTL to MTL Data Rate Converter to the Tunnel pads.

For MMIO, the receiving latency is 6-7 cycles @ 96MHz (62.5ns - 72.9ns). This is the delay from the receiving pads to the DTL MMIO interface with a clock crossing boundary. This is the required time to decode the packet and make the data available to DTL. The transmitting latency is 5 cycles @ 96MHz (52.1ns). This is the delay from the DTL interface to the transmitting pads. This is the time required to packetize and encode the data to be sent.

For memory accesses, the receiving latency is measured from the receiving pads to the internal interface of the MTL to MTL Data Rate Converter. This latency is 6-7 cycles @ 96MHz (62.5ns - 72.9ns). This is the required time to decode the packet and make the data available to the internal interface. There is additional latency incurred from the MTL to MTL Data Rate Converter. The converter only sends a write command if it has all the necessary write data. This latency depends on the size of the transfer. The transmitting latency is 5 cycles @ 96MHz (52.1ns). This is the delay from the internal interface to the transmitting pads. This is the required time to packetize and encode the data to be sent. The MTL to MTL Data Rate Converter adds around 6 additional cycles for a total of 11 @ 96MHz (114.6ns).

5.8.4 Tunnel PCB connections

5.8.4.1 Both tunnels utilized

Figure 208 shows the physical connections of the tunnel interface. All Tunnel clocks are complementary but not differential. In the PNX2015 the tunnel data and Clock drivers are identical. None of the Tunnel interfaces use the inverted clock input, the PNX8550 does not have an inverted clock input. Both of the complementary clocks however should be routed, as a closely matched pair to reduce cross coupling interference. The Inverted clock to the PNX8550 should be terminated as an open circuit as close to the positive clock as possible.

The source series termination resistors to match the IC output impedance to that of the PCB track and should be mounted as close as possible to the output pins.

The PNX2015 and PNX8550 Tunnel clocks are not differential but complementary type. Clock and data tracks should be source series terminated in the same way, with the termination resistor positioned as close to the output pins as possible (<6mm). The source series termination resistors match the PNX2015 output impedance (~18R) to that of the PCB track impedance. For a 60 Ohm characteristic impedance PCB the series termination resistor would be 43R.

Data and clock tracks for each direction of the Tunnel interface must be line length matched to one another, aiming for a +/- 0.25mm tolerance.

The PNX8550 Tunnel does not have an inverted clock input (CLKN). To reduce cross coupling interference, route both of the CLKP/N tracks as a pair to the PNX8550 but leave the CLKN as an open circuit near the CLKP connection to PNX8550. The PNX8550 Tunnel outputs also have a different impedance (~30R) from the PNX2015 and so should have a 30R termination resistor for a 60 Ohm board.

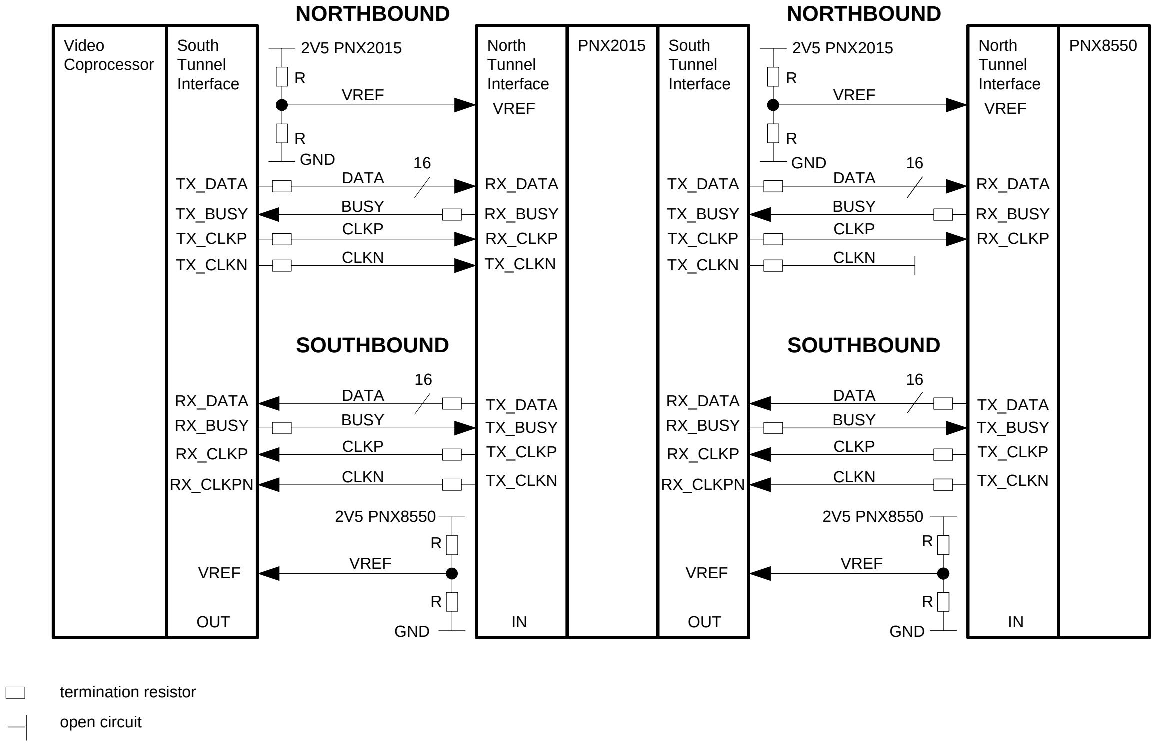


Fig 208.Tunnel interface connections

5.8.4.2 Unused tunnel

When a tunnel is unused, it must be tied off on the PCB to the following safe values.

Table 291: Unused Tunnel Connections

Signal	Direction	Connection
TX_BUSY	INPUT	GND
TX_CLKP	OUTPUT	OPEN
TX_CLKN	OUTPUT	OPEN
TX_DATA[15:0]	OUTPUT	OPEN
RX_BUSY	OUTPUT	OPEN
RX_CLKP	INPUT	2.5V
RX_CLKN	INPUT	GND
RX_DATA[15:0]	INPUT	GND
VREF		1.25V

5.8.5 PNX2015 system setup

Figure 209 is a simple system block diagram showing the DTL/MTL ports.

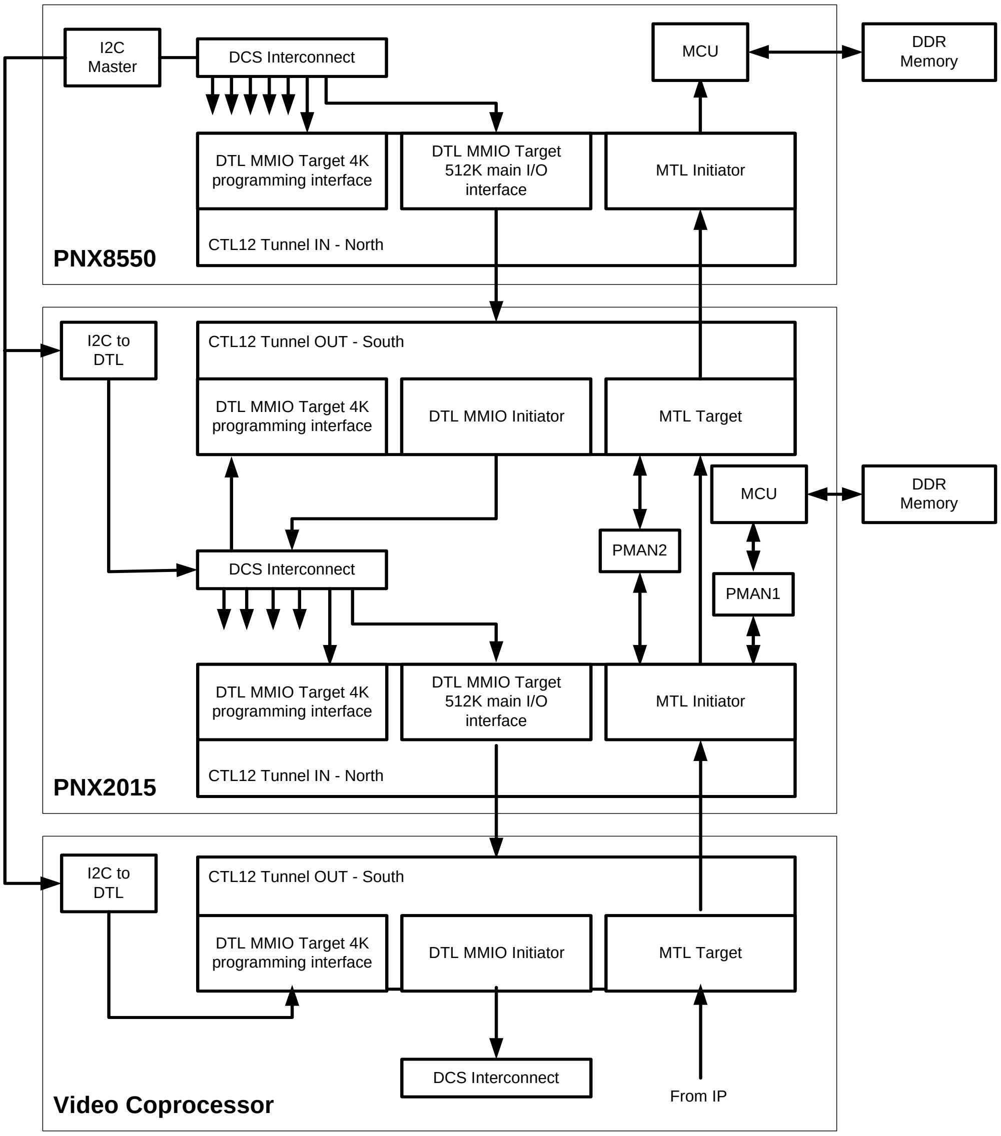


Fig 209.System block diagram

Figure 210 outlines the order in which to initialize the PNX8550, PNX2015 and Video co-processor subsystem. Each block within the diagram below requires one or more MMIO or I^{2}C writes.

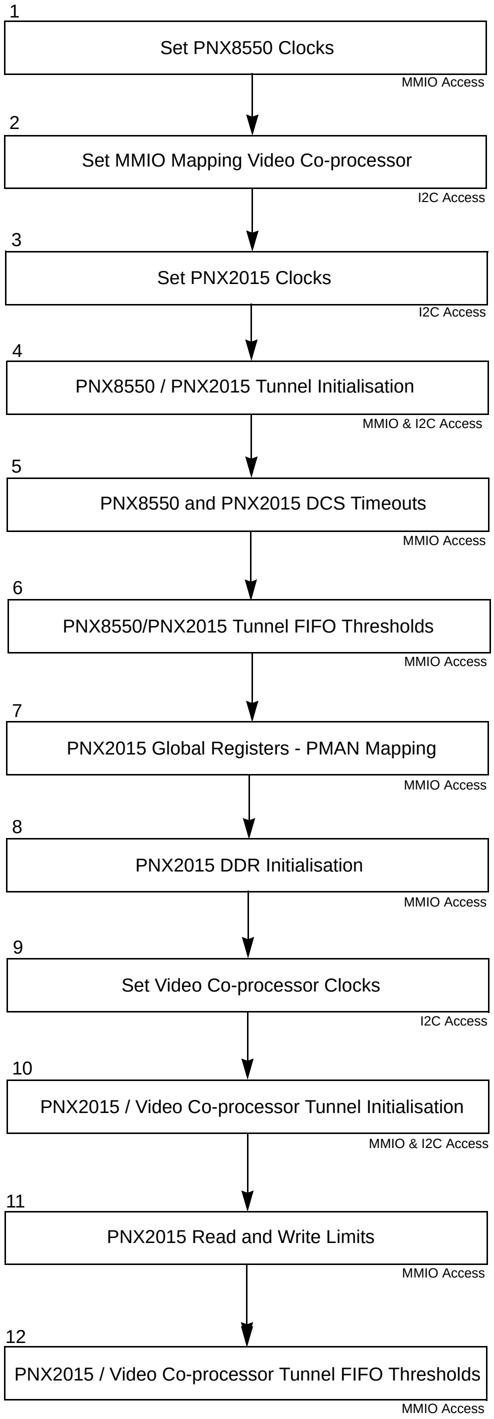

1
Set PNX8550 Clocks
MMIO Access
2
Set MMIO Mapping Video Co-processor
I2C Access
3
Set PNX2015 Clocks
I2C Access
4
PNX8550 / PNX2015 Tunnel Initialisation
MMIO & I2C Access
5
PNX8550 and PNX2015 DCS Timeouts
MMIO Access
6
PNX8550/PNX2015 Tunnel FIFO Thresholds
MMIO Access
7
PNX2015 Global Registers - PMAN Mapping
MMIO Access
8
PNX2015 DDR Initialisation
MMIO Access
9
Set Video Co-processor Clocks
I2C Access
10
PNX2015 / Video Co-processor Tunnel Initialisation
MMIO & I2C Access
11
PNX2015 Read and Write Limits
MMIO Access
12
PNX2015 / Video Co-processor Tunnel FIFO Thresholds
MMIO Access

Fig 210.HD System Setup / Initialisation

1. The PMAN, MDCS, TDCS, DDR and north tunnel clocks need to be set on the PNX8550 device. The registers to configure the tunnel clock are outlined in Table 292.

Table 292: PNX8550 clocks

Register	Offset
IP BLOCK: CLOCKS (Offset = 0x 47000)	
PLL5_CTL	0x014
TUNNEL_CTL	0x21C

2. The MMIO mapping of the video co-processor or PNX2015 may require setting at this point. The intended location of the MMIO for the video co-processor is at the bottom of the 512KB address range e.g. 0 - 128KB. This is to avoid a clash with PNX2015 which defaults to 256KB - 384KB within the 512KB tunnel address range. If the video co-processor defaults to 0-128KB then no action need be taken. If there is a clash, then either the PNX2015 or video co-processor MMIO mapping may be moved.

3. The PMAN, DCS, DDR and south tunnel clocks on the PNX2015 will require setting via the I^{2}C on the PNX2015. The registers required to configure the north and south tunnel clocks are outlined in Table 293.

Table 293: PNX2015 clocks

Register	Offset
IP BLOCK: HDCLOCKS (Offset = 0x 1BFC4000)	
PLL_DDR_CTL	0x014
CLK_TUNNEL_V2_CTL	0x214
CLK_TUNNEL_SPID_CTL	0x21C

4. Initialize the PNX8550 and PNX2015 tunnel.

Prior to initialisation, the north and south tunnel clocks must be set to their intended frequency. Once initialised the clocks must remain stable and therefore cannot be altered. Changing the clock frequency after initialisation may cause a tunnel error and crash the system.

To initialize the tunnel will require access to the 4K of control registers for both the north and south tunnel.

For the PNX8550 north tunnel, access will be via MMIO on the DCS network initiated by one of the processors (MIPS or TRI-MEDIA). Accessing the PNX2015 south tunnel will be done via an I2C access initiated by one of the processors. These accesses are shown in Figure 211.

Assuming the tunnel between PNX8550 and PNX2015 is initialised, there are two options for accessing the PNX2015 north tunnel register. This is via either an I2C access or an MMIO access via the PNX8550 - PNX2015 tunnel.

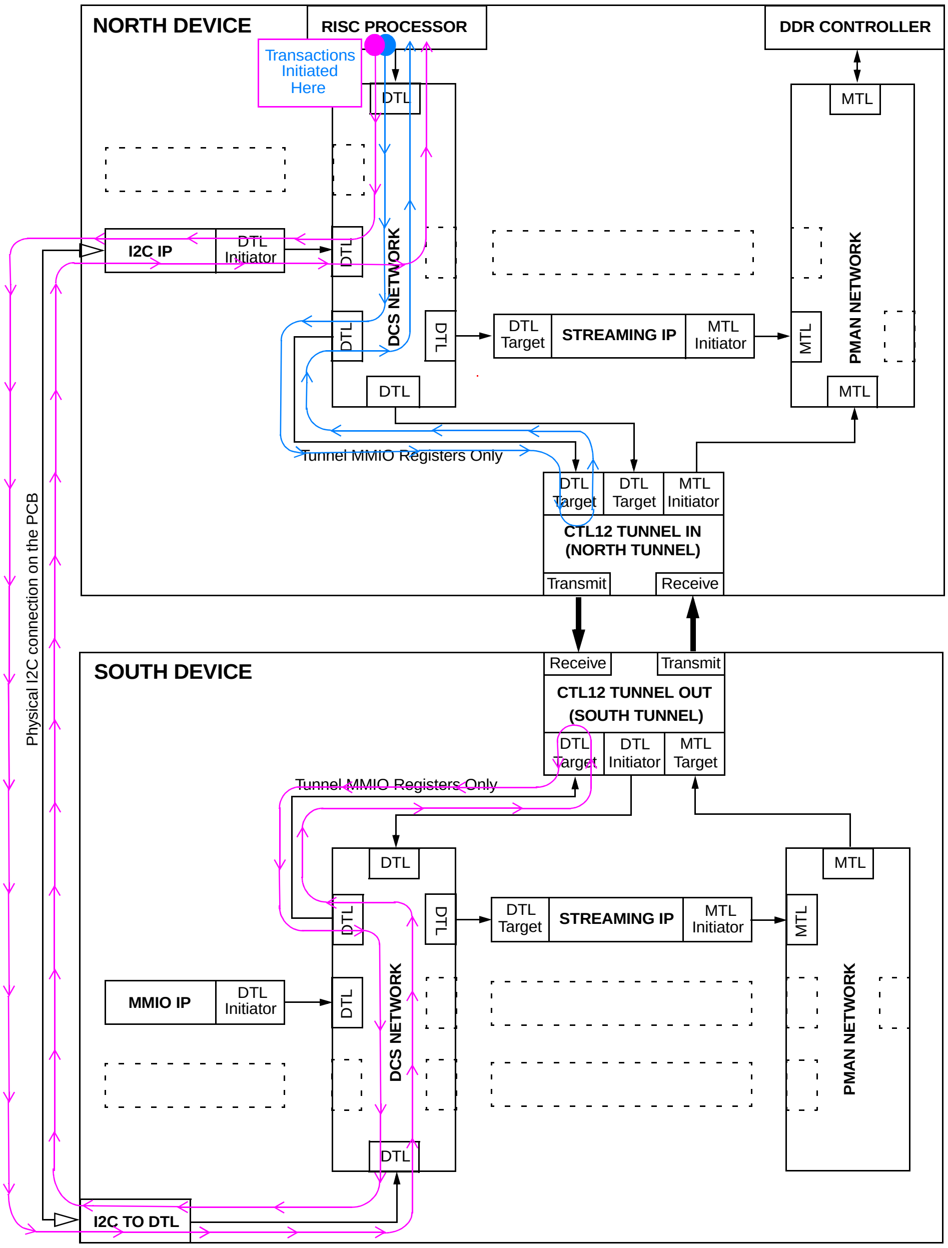


Fig 211.Tunnel initialisation control registers (PNX8550 - PNX2015)

The tunnel powers up in reset. After reset, the tunnel is idle and no clocks are transmitted. Tunnel start-up is under the control of software.

To initialize the tunnel interface software must performs the following steps:

1. Set the “driver en” bit in the north and south tunnel’s "Configuration Register". This will enable the source synchronous tunnel clock, start transmitting sync codes, and reset the receive side logic.

2. Clear the “soft reset” bit in the north and south tunnel’s “Configuration Register. This will take the Tunnel receive logic out of reset.
3. Insert a very small delay here (several clock cycles) to ensure that valid sync codes are being received.
4. Set the “tx/rx en” bit in the north and south tunnel’s “Configuration Register”. This will allow the Tunnel to start accepting commands.
5. Clear the “send sync in” bit in the north and south tunnel’s “Configuration Register”. This will allow the tunnel to actually start sending command/data/status packets. If no command/data/status packets are pending, then Idle packets are transmitted.
6. The tunnel is now ready to operate.

Table 294 shows an initialisation sequence for the tunnel between PNX8550 and PNX2015. Figure 212 graphically shows the activity on the tunnel bus during initialisation.

Table 294: Tunnel Initialisation (PNX8550 - PNX2015)

Order	Access	Type	Device	Sub-Address	Data	Description
Step 1						
1(a)	Write	MMIO	PNX8550	0x0017F00C	0x84020000	Enable driver in the north tunnel
1(b)	Write	I2C	PNX2015	0x0005600C	0x84020000	Enable driver in the south tunnel
Step 2						
2(a)	Write	MMIO	PNX8550	0x0017F00C	0x04020000	Clear soft reset on the north tunnel
2(b)	Write	I2C	PNX2015	0x0005600C	0x04020000	Clear soft reset on the south tunnel
Step 3						
3	Insert a small delay here (several clock cycles) to allow the north and south tunnels to synchronize					
Step 4						
4(a)	Write	MMIO	PNX8550	0x0017F00C	0x04060000	Enable transmit/receive on the north tunnel
4(b)	Write	I2C	PNX2015	0x0005600C	0x04060000	Enable transmit/receive on the south tunnel
Step 5						
5(a)	Write	MMIO	PNX8550	0x0017F00C	0x00060000	Clear send sync on the north tunnel
5(b)	Write	I2C	PNX2015	0x0005600C	0x00060000	Clear send sync on the south tunnel
	Tunnel initialisation complete					

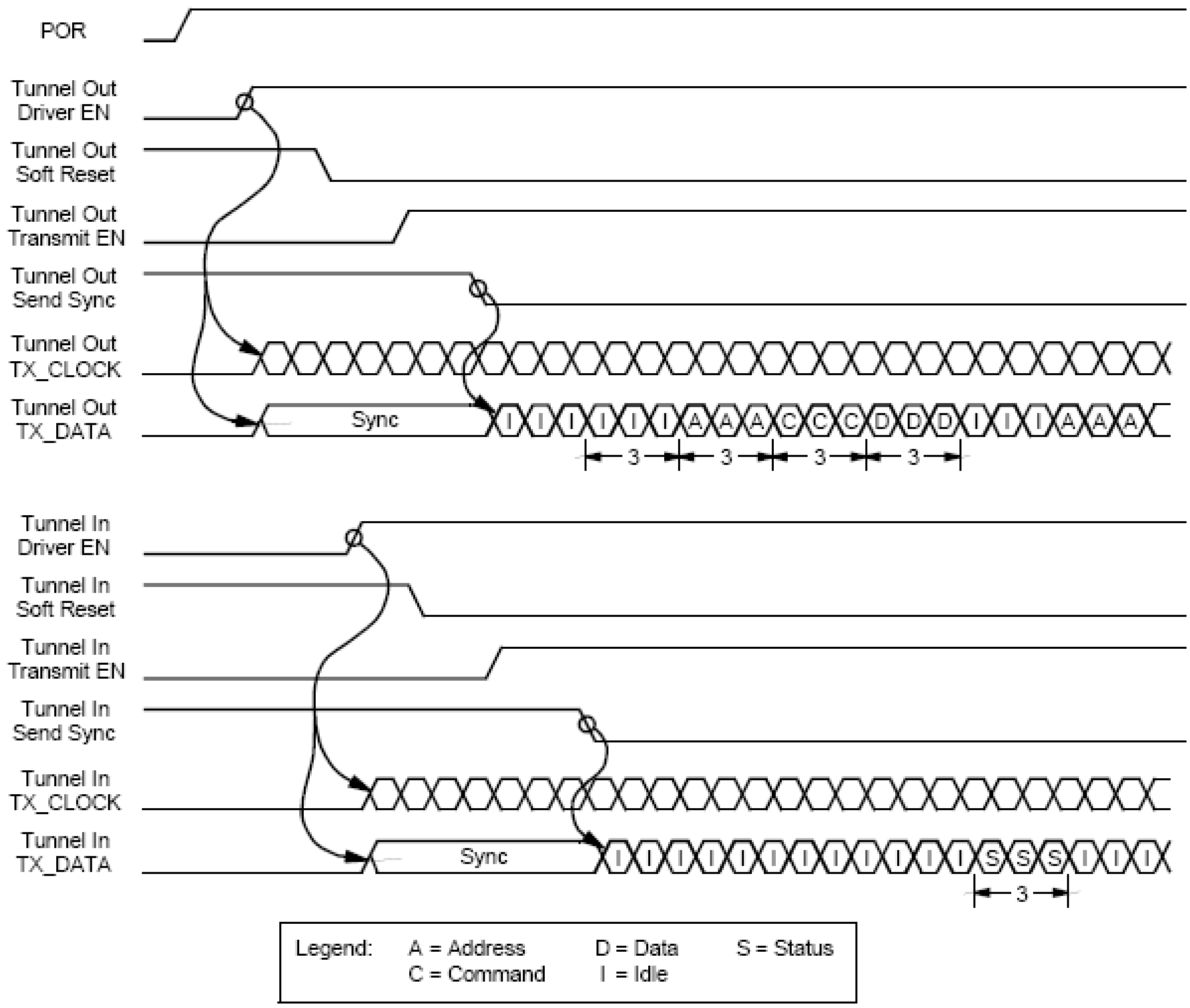


Fig 212.CTL12 tunnel startup sequence

5. The DCS timeouts are disabled by default on power up / reset. Therefore if there is an accidental read or write to a register that does not exist, then this will hang the DCS bus and prevent further MMIO accesses. Essentially, this will hang the system. Therefore, it is important to enable the DCS timeouts. There are three DCS buses that require programming, the PNX8550-MDCS, PNX8550-TDCS and PNX2015-DCS. These registers are highlighted in Table 295 to Table 297.

Table 295: PNX8550 MDCS Control Register

Register	Offset
IP BLOCK: MDCS CONTROL (Offset = 0x 4E000)	
BC_CTRL	0x000

Table 296: PNX8550 TDCS Control Register

Register	Offset
IP BLOCK: MDCS CONTROL (Offset = 0x 103000)	
BC_CTRL	0x000

Table 297: PNX2015 DCS Control Register

Register	Offset
IP BLOCK: DCS CONTROL (Offset = 0x 1BFC0000)	
DCS_CTRL	0x000

6. PNX8550 / PNX2015 tunnel FIFO thresholds

As discussed earlier, the flow control over the Tunnel is handled by the side band busy signals. There is a busy signal in either direction across the tunnel. These signals tell the transmit side to stop transmitting when the 32 deep receiving FIFO reaches a threshold level. This threshold is programmable between 4 and 28 deep. The default level (reset value) is 4 deep. To prevent data stalls and to maximize bandwidth, it is advisable to have the programmable threshold set higher than 4. However, it should be set at a level whereby there is enough room left in the FIFO to accommodate the extra data due to the latency. This is the latency for the transmitting side to detect the busy signal and for it to stop transmitting.

A suitable threshold value would probably be a level of 20. This leaves 12 ranks left in the FIFO for the extra latency data. If a tunnel overflow error is ever detected then try reducing this threshold value. The threshold level is programmed in the tunnel configuration register. Both the north and south tunnels require their FIFO levels programming.

Table 298: PNX8550 North Tunnel Configuration Register

Register	Offset
IP BLOCK: CTL12 TUNNEL IN / NORTH TUNNEL (Offset = 0x 17F000)	
CTL12 IN CONFIGURATION	0x00C

Table 299: PNX2015 South Tunnel Configuration Register

Register	Offset
IP BLOCK: CTL12 TUNNEL OUT / SOUTH TUNNEL (Offset = 0x 1BFD6000)	
CTL12_OUT_CONFIGURATION_REGISTER	0x00C

7. PMAN mapping

The PNX2015 PMAN HUB allows access to either local memory or memory attached to the PNX8550.

The PNX8550 DRAM is always located in the address range 0 -> DRAM_SIZE. For PNX8550, DRAM_SIZE can be 32,64 or 128MB. The PNX2015 local memory sits directly on top of the PNX8550 DRAM address space. Two PNX2015 global registers, PMAN_ROUTE_MASK and PMAN_ROUTE_VALUE, define the start address of the local memory. These two registers require setting at this stage.

Table 300: PNX2015 PMAN_ROUTE_MASK and PMAN_ROUTE_VALUE

Register	Offset
IP BLOCK: GLOBALREG (Offset = 0x 1BFC2000)	
PMAN_ROUTE_MASK	0x810
PMAN_ROUTE_VALUE	0x80C

For example, if PNX8550 has 128MB of DRAM attached, and the PNX2015 32MB. The register LOCAL_BASE = 128MB. Addresses on the HUB less than 128MB would be routed by PNX2015 towards the south tunnel on PMAN2. These addresses would flow across the tunnel to the PNX8550 device. Addresses on the HUB equal to, or above 128MB would be routed towards the PNX2015 local DDR memory on PMAN1.

It is not possible for PNX8550 to address directly via the HUB, the local memory of the PNX2015, since the PNX2015 memory does not appear in the PNX8550 memory map.

There is a very low bandwidth connection from PNX8550 to PNX2015 DDR memory. This is via the PNX2015 DMA_GATE block. The DMA_GATE is configured such that 4K sections of the PNX2015 DDR memory appear as MMIO registers.

8. PNX2015 DDR Initialisation

The DDR will now require initializing. Please note, the DDR clocks for both the PNX8550 and PNX2015 should have been configured earlier. This is important if the tunnel is programmed to run from the same PLL as the DDR which is the case for 225MHz operation. Altering the DDR clock at this stage may crash the tunnel.

9. Video Co-Processor Clocks

The video co-processor south tunnel clock will require setting at this point.

10. PNX2015 and Video Co-Processor Tunnel Initialisation

Table 301: Tunnel Initialisation (PNX8550 - PNX2015)

Order	Access	Type	Device	Sub-Address	Data	Description
Step 1						
1(a)	Write	I2C	PNX2015	0x0005500C	0x84020000	Enable driver in the north tunnel
1(b)	Write	I2C	Video Co-P	STunnel Offset + 00C	0x84020000	Enable driver in the south tunnel
Step 2						
2(a)	Write	I2C	PNX2015	0x0005500C	0x04020000	Clear soft reset on the north tunnel
2(b)	Write	I2C	Video Co-P	STunnel Offset + 00C	0x04020000	Clear soft reset on the south tunnel
Step 3						
3	Insert a small delay here (several clock cycles) to allow the north and south tunnels to synchronize					
Step 4						
4(a)	Write	I2C	PNX2015	0x0005500C	0x04060000	Enable transmit/receive on the north tunnel
4(b)	Write	I2C	Video Co-P	STunnel Offset + 00C	0x04060000	Enable transmit/receive on the south tunnel
Step 5						

Table 301: Tunnel Initialisation *…continued*(PNX8550 - PNX2015)

Order	Access	Type	Device	Sub-Address	Data	Description
5(a)	Write	I2C	PNX2015	0x0005500C	0x00060000	Clear send sync on the north tunnel
5(b)	Write	I2C	Video Co-P	STunnel Offset + 00C	0x00060000	Clear send sync on the south tunnel
	Tunnel initialisation complete					

11 - PNX2015 Read and Write Limits (PMAN2)

An increase in the number of allowable pipelined write transactions is required because the latency of the write acknowledges is increased by the addition of the PNX2015 in to the PNX8550 / Video Co-processor pipeline.

For a single tunnel connection (PNX8550-PNX2015 or PNX8550-Video Co-processor), the time between a request to write 128 bytes to PNX8550 memory and the acknowledge of the completion of this write, is approximately the time taken for two 128 byte write transactions. This is for the case when the PMAN, tunnel and DDR interfaces are all running at approximately the same frequency. So to keep the pipeline full two write requests must be issued at any one time.

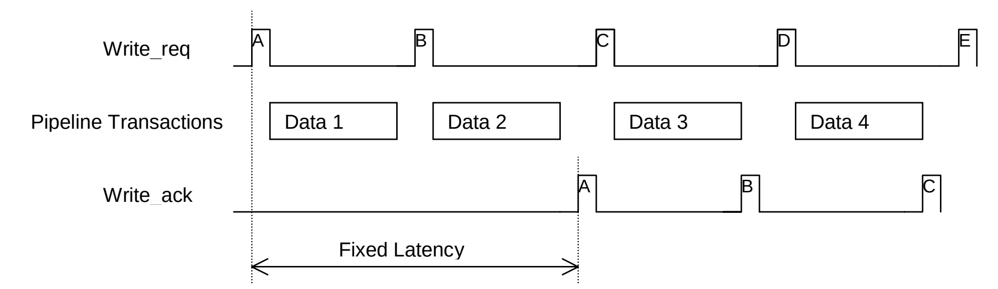


Fig 213.Single tunnel system - 2 devices (write limit = 2 outstanding writes)

The insertion of the PNX2015 in to the pipeline adds an additional latency for the write acknowledge of around two 128byte transactions. This results in a total latency of around four 128 byte transactions. Therefore, if the write limit is not increased then the pipeline will only ever be half full. This has the effect of reducing the usable bandwidth to approximately 50% of the available.

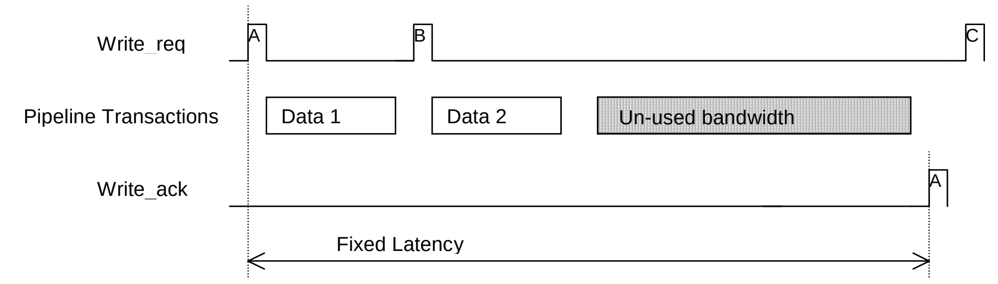


Fig 214.Two tunnel system - 3 devices (write limit = 2 outstanding writes)

When the write limit is increased to 4 transactions, the pipeline is filled and makes use of all the available bandwidth.

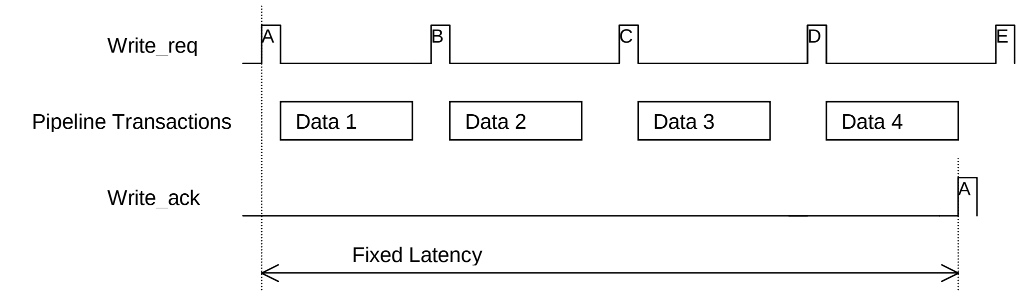


Fig 215.Two tunnel system - 3 devices (write limit = 4 outstanding writes)

When the pipeline is not running at a balanced frequency e.g. if the tunnel between the Video Co-processor and PNX2015 is running at 27MHz, and the rest of the system is 192MHz, then the write limit has no effect.

In a balanced system most of the write acknowledge latency comes from the request path from PNX2015 north tunnel to PNX8550 and back again.

In the unbalanced system, the time it takes to transfer the first 128Byte transaction across the 27MHz tunnel is comparable to the time for the write acknowledge. So the first transaction gets acknowledged before the second transaction has completed. Therefore there is no benefit to increasing the write limit in this type of unbalanced system.

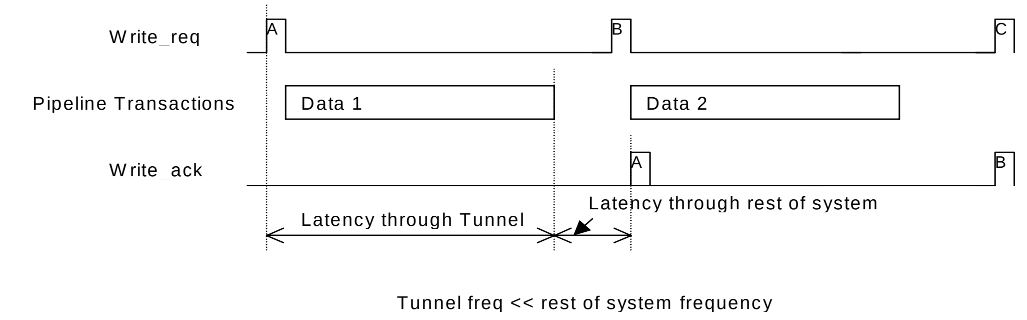


Fig 216.Unbalanced two tunnel system (write limit = 4 outstanding writes)

Therefore, in a system with three devices and two tunnels, it is recommended to set the write limit equal to 4 for the PNX2015 PMAN2.

Similarly, the insertion of the PNX2015 in to the pipeline adds additional latency for the read acknowledges. Therefore, to maximize the bandwidth it is advisable to increase the number of allowed outstanding reads. The recommended value for the PMAN2 read limit is 5.

The read and write limits are set in the global registers (see Table 302).

Table 302: PNX2015 PMAN2 Read and Write Limit

Register	Offset
IP BLOCK: GLOBALREG (Offset = 0x 1BFC2000)	
WRITE_LIMIT_PMAN2	0x808
READ_LIMIT_PMAN2	0x804

Additionally, the video co-processor must be capable of allowing 4 outstanding write requests and 5 outstanding read requests in order to fill the pipeline.

12 - PNX2015 / Video Co-processor Tunnel FIFO Threshold

The threshold level is programmed in the tunnel configuration register as discussed in step 6 in this section. Both the PNX2015 north and video co-processor south tunnels require their FIFO levels programming.

Table 303: PNX2015 North Tunnel Configuration Register

Register	Offset
IP BLOCK: CTL12 TUNNEL IN / NORTH TUNNEL (Offset = 0x 1BFD5000)	
CTL12 IN CONFIGURATION	0x00C

5.8.6 Re-initialisation

Should the system crash, then the only solution is to reset all three devices and then re-initialize the system from scratch. Even if only one of the devices has crashed. It is not possible to un-initialize one side of a tunnel and then re-initialize.

5.8.7 Accessing PNX2015 registers

PNX2015 registers can be accessed either through the tunnel MMIO or via the I^2C.

5.8.7.1 Via tunnel MMIO

Ensure the tunnel between Viper2 and PNX2015 is successfully initialised.

Address = Viper2 MMIO Base + Tunnel Aperture Base within Viper2 MMIO + PNX2015 DCSN Offset + PNX2015 MMIO Offset.

Viper2 MMIO Base = 0x1BE00000

Tunnel Aperture Base within Viper2 MMIO = 0x00180000

PNX2015 DCSN Offset = 0x00040000 (With the DCSN_BASE at 256KB)

For example, to access PMAN_ROUTE_VALUE within the GLOBALREG block of PNX2015 via the tunnel:

1BE00000 + 180000 + 40000 + 280C = 1BFC280C

This is the address format that is listed within the EASI Product Reference for the PNX2015 HD Subsystem.

5.8.7.2 Via I^2C

Address = PNX2015 DCSN Offset + PNX2015 MMIO Offset.

Therefore to access PMAN_ROUTE_VALUE within the GLOBALREG block of PNX2015, the I2C sub-address would be:

40000 + 280C = 0004280C

5.8.8 Register descriptions

5.8.8.1 North tunnel register summary

Table 304: Register summary

Offset	Symbol	Description
0x0	CTL12_IN_Receive_Status_Register	
0x4	CTL12_IN_Packet_Status_Register	
0x8	CTL12_IN_Packet_Register	
0xC	CTL12_IN_Configuration_Register	
0x10	CTL12_IN_Idle_Packet_Status_Register	
0x14	CTL12_IN_TUN_TX_DATA_Register	
0x18	CTL12_IN_TUN_RX_DATA_Register	
0xFF4	CTL12_IN_Power_Down_Register	
0xFFC	CTL12_IN_Module_ID_Register	

5.8.8.2 North tunnel register table

Table 305: North tunnel registers

Bit	Symbol	Access	Value	Description
Offset 0x0 - CTL12_IN_Receive_Status_Register				
31:24	Dcd_err_pipe2[7:0]	R	0x00	6b8b Code in Pipe 2 8-bit encoded value that caused a DCD error in pipe 2
23:16	Dcd_err_pipe1[7:0]	R	0x00	6b8b Code in Pipe 1 8-bit encoded value that caused a DCD error in pipe 1
15	Dcd_err	R/W	0x0	The 6b8b decoder in the CTL12 circuit received an illegal code (did not have 4 ones and 4 zeros). A 1 indicates an error has occurred. This bit is cleared by writing a 1.
14:8	Dcd_err_count[6:0]	R	0x00	This value counts the number of DCD errors received. Once the terminal count is reached, the counter stops. Clearing the dcd_err bit will clear this counter value. (Max count is 127.)
7	Overflow	R/W	0x0	This bit indicates whether or not the CTL12 receiving FIFO overflowed. A 1 indicates the FIFO overflowed. This bit is cleared by writing a 1.
6	Reserved	R	0x0	Reserved for future use
5:0	Rcv_fifo_level[5:0]	R	0x00	Real time level of the CTL12 receiving FIFO
Offset 0x4 - CTL12_IN_Packet_Status_Register				
31:30	Reserved[1:0]	R	0x0	Reserved for future use.
29:24	Tun_err[5:0]	R	0x00	This value describes the error received by the CTL12 IN block. 0: No Error 1: Expect Addr Error 10: Expect Cmd Error 100: Expect 2D Cmd Error 1000: Expect Data

Table 305: North tunnel registers *…continued*

Bit	Symbol	Access	Value	Description
23:4	Reserved2[19:0]	R	0x00000	Reserved for future use
3:0	Pid[3:0]	R	0x0	Represents the 4-bit value that is the packet ID. This value is only updated when a Tunnel error occurs.
Offset 0x8 - CTL12_IN_Packet_Register				
31:0	Rcv_packet[31:0]	R	0x00000000	This register represents the entire packet without the packet ID. This register is only updated when a Tunnel error occurs.
Offset 0xC - CTL12_IN_Configuration_Register				
31	Soft_reset	R/W	0x1	Resets the CTL12 Tunnel In receive logic once a clock is received. It also holds the receive logic outputs at a known value. If no clock is ever received, then this bit should remain high. Once the receive clock is active, this bit should be set low before bringing up the Tunnel interface.
30:27	Reserved[3:0]	R/W	0x0	Reserved for future use.
26	Send_sync_in	R/W	0x1	This bit causes the transmitter in the Tunnel In to stuff sync codes into the stream.
25	Send_idle_in	R/W	0x0	This bit causes the transmitter in the Tunnel In to stuff idle codes into the stream.
24	Test_in	R/W	0x0	When TEST_IN = 1, the Tunnel In transmit pins are driven by the Tunnel In TUN_TX_DATA Test register. If the tunnel is properly connected, the test receive register in the South Chip should match the Tunnel In TUN_TX_DATA Test register.
23	Posting	R/W	0x0	This bit enables the MMIO interface to post one write command.

Table 305: North tunnel registers *...continued*

Bit	Symbol	Access	Value	Description
22:20	Threshold_in[2:0]	R/W	0x0	The number of ranks reserved in Tunnel In receive FIFO to prevent overflow due to flight time and processing delays in flow control between chips. 0: declare almost full when level >= 4 1: declare almost full when level >= 4 10: declare almost full when level >= 8 11: declare almost full when level >= 12 100: declare almost full when level >= 16 101: declare almost full when level >= 20 110: declare almost full when level >= 24 111: declare almost full when level >= 28
19	Reserved2	R/W	0x0	Reserved for future use.
18	Tx_rx_en	R/W	0x0	Transmit/Receive Enable Enables the CTL12 block to start accepting commands.
17	Driver_en	R/W	0x0	Enables the CTL12 block to start the clock and drive out sync codes.
16:0	Reserved3[16:0]	R/W	0x00000	Reserved for future use.
Offset 0x10 - CTL12_IN_Idle_Packet_Status_Register				
31:0	Idle[31:0]	R/W	0xFFFFFFFF	Idle Data Pattern. During IDLE time this pattern is driven out on the transmit pins. This value is before 6b8b encoding. Packet Encoded 0x6FFFFFFFF 0xF45555555555
Offset 0x14 - CTL12_IN_TUN_TX_DATA_Register				
31:16	Reserved[15:0]	R/W	0x0000	Reserved for future use.
15:0	Tun_tx_data[15:0]	R/W	0x5555	When TEST_IN = 1, the contents of this register drive TUN_TX_DATA Tunnel In pins.
Offset 0x18 - CTL12_IN_TUN_RX_DATA_Register				
31:16	Reserved[15:0]	R	0x0000	Reserved for future use.
15:0	Tun_rx_data[15:0]	R	0x0000	This register shows the current state of the TUN_RX_DATA Tunnel In pins.
Offset 0xFF4 - CTL12_IN_Power_Down_Register				

Table 305: North tunnel registers *...continued*

Bit	Symbol	Access	Value	Description
31	Pd	R/W	0x0	Power Down 0: Normal operation 1: Powerdown mode
30:0	Reserved[30:0]	R	0x00000000	Reserved for future use.
Offset 0xFFC - CTL12_IN_Module_ID_Register				
31:16	Module_id[15:0]	R	0xA04A	Module ID value "0xA04A" for the CTL12 Tunnel In block.
15:12	Major_rev[3:0]	R	0x0	Major revision number (0x0)
11:8	Minor_rev[3:0]	R	0x0	Minor revision number (0x0)
7:0	Aperture[7:0]	R	0x00	Aperture size for the configuration space measured in 4 KB. The aperture for the CTL12 Tunnel In block is 4 KB.
Offset 0x0 - CTL12_IN_Receive_Status_Register				
31:24	Dcd_err_pipe2[7:0]	R	0x00	6b8b Code in Pipe 2 8-bit encoded value that caused a DCD error in pipe 2
23:16	Dcd_err_pipe1[7:0]	R	0x00	6b8b Code in Pipe 1 8-bit encoded value that caused a DCD error in pipe 1
15	Dcd_err	R/W	0x0	The 6b8b decoder in the CTL12 circuit received an illegal code (did not have 4 ones and 4 zeros). A 1 indicates an error has occurred. This bit is cleared by writing a 1.
14:8	Dcd_err_count[6:0]	R	0x00	This value counts the number of DCD errors received. Once the terminal count is reached, the counter stops. Clearing the dcd_err bit will clear this counter value. (Max count is 127.)
7	Overflow	R/W	0x0	This bit indicates whether or not the CTL12 receiving FIFO overflowed. A 1 indicates the FIFO overflowed. This bit is cleared by writing a 1.
6	Reserved	R	0x0	Reserved for future use
5:0	Rcv_fifo_level[5:0]	R	0x00	Real time level of the CTL12 receiving FIFO
Offset 0x4 - CTL12_IN_Packet_Status_Register				
31:30	Reserved[1:0]	R	0x0	Reserved for future use.
29:24	Tun_err[5:0]	R	0x00	This value describes the error received by the CTL12 IN block. 0: No Error 1: Expect Addr Error 10: Expect Cmd Error 100: Expect 2D Cmd Error 1000: Expect Data
23:4	Reserved2[19:0]	R	0x00000	Reserved for future use
3:0	Pid[3:0]	R	0x0	Represents the 4-bit value that is the packet ID. This value is only updated when a Tunnel error occurs.

Table 305: North tunnel registers *…continued*

Bit	Symbol	Access	Value	Description
Offset 0x8 - CTL12_IN_Packet_Register				
31:0	Rcv_packet[31:0]	R	0x00000000	This register represents the entire packet without the packet ID. This register is only updated when a Tunnel error occurs.
Offset 0xC - CTL12_IN_Configuration_Register				
31	Soft_reset	R/W	0x1	Resets the CTL12 Tunnel In receive logic once a clock is received. It also holds the receive logic outputs at a known value. If no clock is ever received, then this bit should remain high. Once the receive clock is active, this bit should be set low before bringing up the Tunnel interface.
30:27	Reserved[3:0]	R/W	0x0	Reserved for future use.
26	Send_sync_in	R/W	0x1	This bit causes the transmitter in the Tunnel In to stuff sync codes into the stream.
25	Send_idle_in	R/W	0x0	This bit causes the transmitter in the Tunnel In to stuff idle codes into the stream.
24	Test_in	R/W	0x0	When TEST_IN = 1, the Tunnel In transmit pins are driven by the Tunnel In TUN_TX_DATA Test register. If the tunnel is properly connected, the test receive register in the South Chip should match the Tunnel In TUN_TX_DATA Test register.
23	Posting	R/W	0x0	This bit enables the MMIO interface to post one write command.
22:20	Threshold_in[2:0]	R/W	0x0	The number of ranks reserved in Tunnel In receive FIFO to prevent overflow due to flight time and processing delays in flow control between chips. 0: declare almost full when level >= 4 1: declare almost full when level >= 4 10: declare almost full when level >= 8 11: declare almost full when level >= 12 100: declare almost full when level >= 16 101: declare almost full when level >= 20 110: declare almost full when level >= 24 111: declare almost full when level >= 28

Table 305: North tunnel registers *…continued*

Bit	Symbol	Access	Value	Description
19	Reserved2	R/W	0x0	Reserved for future use.
18	Tx_rx_en	R/W	0x0	Transmit/Receive Enable Enables the CTL12 block to start accepting commands.
17	Driver_en	R/W	0x0	Enables the CTL12 block to start the clock and drive out sync codes.
16:0	Reserved3[16:0]	R/W	0x00000	Reserved for future use.
Offset 0x10 - CTL12_IN_Idle_Packet_Status_Register				
31:0	Idle[31:0]	R/W	0xFFFFFFFF	Idle Data Pattern. During IDLE time this pattern is driven out on the transmit pins. This value is before 6b8b encoding. Packet Encoded 0x6FFFFFFFF 0xF45555555555
Offset 0x14 - CTL12_IN_TUN_TX_DATA_Register				
31:16	Reserved[15:0]	R/W	0x0000	Reserved for future use.
15:0	Tun_tx_data[15:0]	R/W	0x5555	When TEST_IN = 1, the contents of this register drive TUN_TX_DATA Tunnel In pins.
Offset 0x18 - CTL12_IN_TUN_RX_DATA_Register				
31:16	Reserved[15:0]	R	0x0000	Reserved for future use.
15:0	Tun_rx_data[15:0]	R	0x0000	This register shows the current state of the TUN_RX_DATA Tunnel In pins.
Offset 0xFF4 - CTL12_IN_Power_Down_Register				
31	Pd	R/W	0x0	Power Down 0: Normal operation 1: Powerdown mode
30:0	Reserved[30:0]	R	0x00000000	Reserved for future use.
Offset 0xFFC - CTL12_IN_Module_ID_Register				
31:16	Module_id[15:0]	R	0xA04A	Module ID value "0xA04A" for the CTL12 Tunnel In block.
15:12	Major_rev[3:0]	R	0x0	Major revision number (0x0)
11:8	Minor_rev[3:0]	R	0x0	Minor revision number (0x0)
7:0	Aperture[7:0]	R	0x00	Aperture size for the configuration space measured in 4 KB. The aperture for the CTL12 Tunnel In block is 4 KB.

5.8.8.3 South tunnel register summary

Table 306: Register summary

Offset	Symbol	Description
0x0	CTL12_OUT_Receive_Status_Register	
0x4	CTL12_OUT_Packet_Status_Register	
0x8	CTL12_OUT_Packet_Register	
0xC	CTL12_OUT_Configuration_Register	
0x10	CTL12_OUT_Idle_Packet_Status_Register	
0x14	CTL12_OUT_TUN_TX_DATA_Register	
0x18	CTL12_OUT_TUN_RX_DATA_Register	
0xFF4	CTL12_OUT_Power_Down_Register	
0xFFC	CTL12_OUT_Module_ID_Register	

5.8.8.4 South tunnel register table

Table 307: South tunnel registers

Bit	Symbol	Access	Value	Description
Offset 0x0 - CTL12_OUT_Receive_Status_Register				
31:24	Dcd_err_pipe2[7:0]	R	0x00	6b8b Code in Pipe 2 8-bit encoded value that caused a DCD error in pipe 2
23:16	Dcd_err_pipe1[7:0]	R	0x00	6b8b Code in Pipe 1 8-bit encoded value that caused a DCD error in pipe 1
15	Dcd_err	R/W	0x0	The 6b8b decoder in the CTL12 circuit received an illegal code (did not have 4 ones and 4 zeros). A 1 indicates an error has occurred. This bit is cleared by writing a 1.
14:8	Dcd_err_count[6:0]	R	0x00	This value counts the number of DCD errors received. Once the terminal count is reached, the counter stops. Clearing the dcd_err bit will clear this counter value. (Max count is 127.)
7	Overflow	R	0x0	This bit indicates whether or not the CTL12 receiving FIFO overflowed. A 1 indicates the FIFO overflowed. This bit is cleared by writing a 1.
6	Reserved	R	0x0	Reserved for future use
5:0	Rcv_fifo_level[5:0]	R	0x00	Real time level of the CTL12 receiving FIFO
Offset 0x4 - CTL12_OUT_Packet_Status_Register				
31:30	Reserved[1:0]	R	0x0	Reserved for future use.
29:24	Tun_err[5:0]	R	0x00	This value describes the error received by the CTL12 IN block. 0: No Error 1: Expect Addr Error 10: Expect Cmd Error 1000: Expect Data 0x10: Received Write Error on DTL side 0x20: Received Read Error on DTL side

Table 307: South tunnel registers *...continued*

Bit	Symbol	Access	Value	Description
23:4	Reserved2[19:0]	R	0x00000	Reserved for future use
3:0	Pid[3:0]	R	0x0	Represents the 4-bit value that is the packet ID. This value is only updated when a Tunnel error occurs.
Offset 0x8 - CTL12_OUT_Packet_Register				
31:0	Rcv_packet[31:0]	R	0x00000000	This register represents the entire packet without the packet ID. This register is only updated when a Tunnel error occurs.
Offset 0xC - CTL12_OUT_Configuration_Register				
31	Soft_reset	R/W	0x1	Resets the CTL12 Tunnel In receive logic once a clock is received. It also holds the receive logic outputs at a known value. If no clock is ever received, then this bit should remain high. Once the receive clock is active, this bit should be set low before bringing up the Tunnel interface.
30:27	Reserved[3:0]	R/W	0x0	Reserved for future use.
26	Send_sync_in	R/W	0x1	This bit causes the transmitter in the Tunnel In to stuff sync codes into the stream.
25	Send_idle_in	R/W	0x0	This bit causes the transmitter in the Tunnel In to stuff idle codes into the stream.
24	Test_in	R/W	0x0	When TEST_IN = 1, the Tunnel In transmit pins are driven by the Tunnel In TUN_TX_DATA Test register. If the tunnel is properly connected, the test receive register in the South Chip should match the Tunnel In TUN_TX_DATA Test register.
23	Reserved2	R/W	0x0	Reserved for future use.
22:20	Threshold_in[2:0]	R/W	0x0	The number of ranks reserved in Tunnel In receive FIFO to prevent overflow due to flight time and processing delays in flow control between chips. 0: declare almost full when level >= 4 1: declare almost full when level >= 4 10: declare almost full when level >= 8 11: declare almost full when level >= 12 100: declare almost full when level >= 16 101: declare almost full when level >= 20 110: declare almost full when level >= 24 111: declare almost full when level >= 28

Table 307: South tunnel registers *…continued*

Bit	Symbol	Access	Value	Description
19	Big_endian	R/W	0x0	Reserved for future use.
18	Tx_rx_en	R/W	0x0	Transmit/Receive Enable Enables the CTL12 block to start accepting commands.
17	Driver_en	R/W	0x0	Enables the CTL12 block to start the clock and drive out sync codes.
16:0	Reserved3[16:0]	R/W	0x00000	Reserved for future use.
Offset 0x10 - CTL12_OUT_Idle_Packet_Status_Register				
31:0	Idle[31:0]	R/W	0xFFFFFFFF	Idle Data Pattern. During IDLE time this pattern is driven out on the transmit pins. This value is before 6b8b encoding. Packet Encoded 0x6FFFFFFFF 0xF45555555555
Offset 0x14 - CTL12_OUT_TUN_TX_DATA_Register				
31:16	Reserved[15:0]	R/W	0x0000	Reserved for future use.
15:0	Tun_tx_data[15:0]	R/W	0x5555	When TEST_IN = 1, the contents of this register drive TUN_TX_DATA Tunnel In pins.
Offset 0x18 - CTL12_OUT_TUN_RX_DATA_Register				
31:16	Reserved[15:0]	R	0x0000	Reserved for future use.
15:0	Tun_rx_data[15:0]	R	0x0000	This register shows the current state of the TUN_RX_DATA Tunnel In pins.
Offset 0xFF4 - CTL12_OUT_Power_Down_Register				
31	Pd	R/W	0x0	Power Down 0: Normal operation 1: Powerdown mode
30:0	Reserved[30:0]	R	0x00000000	Reserved for future use.
Offset 0xFFC - CTL12_OUT_Module_ID_Register				
31:16	Module_id[15:0]	R	0xA07F	Module ID value "0xFFFF" for the CTL12 Tunnel In block.
15:12	Major_rev[3:0]	R	0x0	Major revision number (0x0)
11:8	Minor_rev[3:0]	R	0x0	Minor revision number (0x0)
7:0	Aperture[7:0]	R	0x00	Aperture size for the configuration space measured in 4 KB. The aperture for the CTL12 Tunnel In block is 4 KB.
Offset 0x0 - CTL12_OUT_Receive_Status_Register				

Table 307: South tunnel registers *…continued*

Bit	Symbol	Access	Value	Description
31:24	Dcd_err_pipe2[7:0]	R	0x00	6b8b Code in Pipe 2 8-bit encoded value that caused a DCD error in pipe 2
23:16	Dcd_err_pipe1[7:0]	R	0x00	6b8b Code in Pipe 1 8-bit encoded value that caused a DCD error in pipe 1
15	Dcd_err	R/W	0x0	The 6b8b decoder in the CTL12 circuit received an illegal code (did not have 4 ones and 4 zeros). A 1 indicates an error has occurred. This bit is cleared by writing a 1.
14:8	Dcd_err_count[6:0]	R	0x00	This value counts the number of DCD errors received. Once the terminal count is reached, the counter stops. Clearing the dcd_err bit will clear this counter value. (Max count is 127.)
7	Overflow	R	0x0	This bit indicates whether or not the CTL12 receiving FIFO overflowed. A 1 indicates the FIFO overflowed. This bit is cleared by writing a 1.
6	Reserved	R	0x0	Reserved for future use
5:0	Rcv_fifo_level[5:0]	R	0x00	Real time level of the CTL12 receiving FIFO
Offset 0x4 - CTL12_OUT_Packet_Status_Register				
31:30	Reserved[1:0]	R	0x0	Reserved for future use.
29:24	Tun_err[5:0]	R	0x00	This value describes the error received by the CTL12 IN block. 0: No Error 1: Expect Addr Error 10: Expect Cmd Error 1000: Expect Data 0x10: Received Write Error on DTL side 0x20: Received Read Error on DTL side
23:4	Reserved2[19:0]	R	0x00000	Reserved for future use
3:0	Pid[3:0]	R	0x0	Represents the 4-bit value that is the packet ID. This value is only updated when a Tunnel error occurs.
Offset 0x8 - CTL12_OUT_Packet_Register				
31:0	Rcv_packet[31:0]	R	0x00000000	This register represents the entire packet without the packet ID. This register is only updated when a Tunnel error occurs.
Offset 0xC - CTL12_OUT_Configuration_Register				
31	Soft_reset	R/W	0x1	Resets the CTL12 Tunnel In receive logic once a clock is received. It also holds the receive logic outputs at a known value. If no clock is ever received, then this bit should remain high. Once the receive clock is active, this bit should be set low before bringing up the Tunnel interface.
30:27	Reserved[3:0]	R/W	0x0	Reserved for future use.

Table 307: South tunnel registers *…continued*

Bit	Symbol	Access	Value	Description
26	Send_sync_in	R/W	0x1	This bit causes the transmitter in the Tunnel In to stuff sync codes into the stream.
25	Send_idle_in	R/W	0x0	This bit causes the transmitter in the Tunnel In to stuff idle codes into the stream.
24	Test_in	R/W	0x0	When TEST_IN = 1, the Tunnel In transmit pins are driven by the Tunnel In TUN_TX_DATA Test register. If the tunnel is properly connected, the test receive register in the South Chip should match the Tunnel In TUN_TX_DATA Test register.
23	Reserved2	R/W	0x0	Reserved for future use.
22:20	Threshold_in[2:0]	R/W	0x0	The number of ranks reserved in Tunnel In receive FIFO to prevent overflow due to flight time and processing delays in flow control between chips. 0: declare almost full when level >= 4 1: declare almost full when level >= 4 10: declare almost full when level >= 8 11: declare almost full when level >= 12 100: declare almost full when level >= 16 101: declare almost full when level >= 20 110: declare almost full when level >= 24 111: declare almost full when level >= 28
19	Big_endian	R/W	0x0	Reserved for future use.
18	Tx_rx_en	R/W	0x0	Transmit/Receive Enable Enables the CTL12 block to start accepting commands.
17	Driver_en	R/W	0x0	Enables the CTL12 block to start the clock and drive out sync codes.
16:0	Reserved3[16:0]	R/W	0x00000	Reserved for future use.
Offset 0x10 - CTL12_OUT_Idle_Packet_Status_Register				

Table 307: South tunnel registers *…continued*

Bit	Symbol	Access	Value	Description
31:0	Idle[31:0]	R/W	0xFFFFFFFF	Idle Data Pattern. During IDLE time this pattern is driven out on the transmit pins. This value is before 6b8b encoding. Packet Encoded 0x6FFFFFFFF 0xF45555555555
Offset 0x14 - CTL12_OUT_TUN_TX_DATA_Register				
31:16	Reserved[15:0]	R/W	0x0000	Reserved for future use.
15:0	Tun_tx_data[15:0]	R/W	0x5555	When TEST_IN = 1, the contents of this register drive TUN_TX_DATA Tunnel In pins.
Offset 0x18 - CTL12_OUT_TUN_RX_DATA_Register				
31:16	Reserved[15:0]	R	0x0000	Reserved for future use.
15:0	Tun_rx_data[15:0]	R	0x0000	This register shows the current state of the TUN_RX_DATA Tunnel In pins.
Offset 0xFF4 - CTL12_OUT_Power_Down_Register				
31	Pd	R/W	0x0	Power Down 0: Normal operation 1: Powerdown mode
30:0	Reserved[30:0]	R	0x00000000	Reserved for future use.
Offset 0xFFC - CTL12_OUT_Module_ID_Register				
31:16	Module_id[15:0]	R	0xA07F	Module ID value "0xFFFF" for the CTL12 Tunnel In block.
15:12	Major_rev[3:0]	R	0x0	Major revision number (0x0)
11:8	Minor_rev[3:0]	R	0x0	Minor revision number (0x0)
7:0	Aperture[7:0]	R	0x00	Aperture size for the configuration space measured in 4 KB. The aperture for the CTL12 Tunnel In block is 4 KB.

5.9 Video Input Processor (VIP)

The VIP handles incoming digital video and processes it for use by other components of the PNX2015. This enables applications such as picture-in-picture and video teleconferencing on the TV screen.

5.9.1 Features

The VIP provides the following functions:

- Receives digital video data from the video port. The data stream may come from a device such as the PNX9975, which can digitize analog video from any source or convert a digital signal from a DVI interface/source into parallel YUV format.

- Features 8/10-bit single channel (single-stream) and 16/20-bit dual channel (dual-stream) capture of CCIR601 YUV 4:2:2 video input with embedded or explicit syncs, supported by a maximum clock frequency of 81 MHz. The dual_stream mode is used to capture a 16 or 20-bit HD stream where 8/10-bit Y and 8/10-bit multiplexed U/V data are received and captured on two separate channels.
- Provides video and auxiliary (AUX, ANC, or RAW) data acquisition and capture. Provides separate acquisition windows for video and for VBI data. Implements two identical dither units capable of either dithering or rounding 9/10 pixel-components in video mode. Enables raw data capture in either 8 or 10 bits for single_stream mode and 8 bits of dual_stream mode. Enables ANC header decoding or window mode for VBI data extraction, only in single stream mode.
- Performs horizontal scaling, cropping and pixel packing on video data from a continuous video data stream or a single field or frame. Performs horizontal down-scaling or zoom-up by 2x, the upscaling being possible only in the single-stream mode. Enables linear horizontal aspect ratio conversion using normal or transposed 6-tap polyphase filter. Enables non-linear horizontal aspect ratio conversion using normal 6-tap polyphase filter. Permits optional linear phase interpolation / nonlinear phase interpolation (as in MBS).
- Allows color-space conversion (mutually exclusive with scaling) on the video path.
- Allows 4:2:2 to 4:4:4 conversions on the video path.
- Provides last-pixel-in signals, for VBI and video data, to the GPIO block for Timestamping.
- Provides an internal Test Pattern Generator with NTSC, PAL, and variable format support.
- Features a wide variety of output formats such as planar YUV 4:4:4, planar YUV 4:2:2, planar RGB, semi-planar YUV 4:2;2, packed UYVY, etc.

5.9.2 Functional description

5.9.2.1 VIP block diagram

The main functional blocks of the VIP and the primary data paths (not including syncs etc.) are shown in Figure 217.

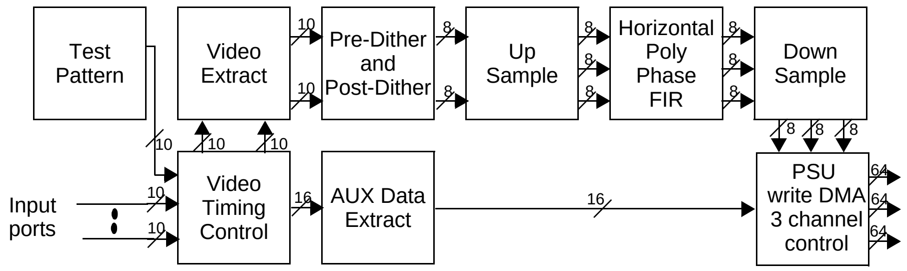


Fig 217.Simplified VIP block diagram

5.9.2.2 Architecture

A brief description of each of the submodules is given in Table 308.

Table 308: VIP submodule descriptions

Submodule	Description
Test Pattern	An internal generator that produces 4:2:2 NTSC/PAL video streams
Video Timing Control	This submodule receives incoming data samples from either the Test Pattern Generator or the Digital Video Port. A tally of the sample is maintained when it conforms to the ITU-R 656 or ITU-R 1364. Video and Aux samples are forwarded to Video Extract and Aux Data Extract respectively.
Video Extract	Video input pipe windower. This submodule: receives video samples from Video Port Input module. captures desired samples in a programmable size rectangular area (window). forwards captured samples to the Pre-Dither unit.
Pre-Dither and Post-Dither	There are two identical Dither units: Pre-Dither and Post-Dither, capable of 10->9, 10->8 and 9->8 dithering/rounding of the video data only.
Up Sample	4:2:2 to 4:4:4 Interpolation FIR Filter for chroma upsampling 8-bit video samples are received from Post-Dither.
Horizontal Poly Phase FIR	Horizontal scaler pipeline
Down Sample	4:4:4 to 4:2:2 Decimation FIR Filter for chroma down sampling.
AUX Data Extract	Video input pipe AUX windower. This submodule: Receives aux samples from Video Port Input module. Captures desired samples in a programmable captured window and/or within a buffer space. Captures ANC packets with matching DID. Captures all valid input samples. Forwards the captured samples to Pixel Packer.
3 Channel Write DMA Control	An interface to the memory agent.

Chip I/O and connections: The VIP module has two 10b data inputs: DV4 and DV5.

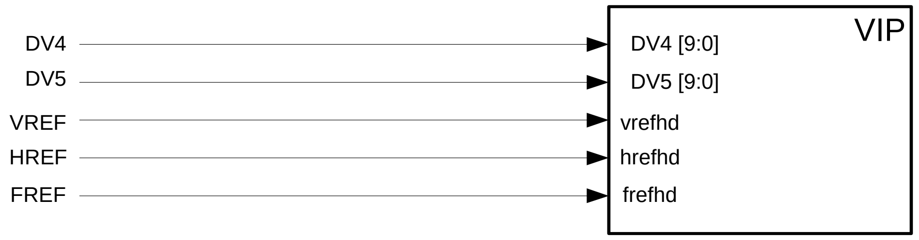


Fig 218.Chip-level digital video inputs, data routing, and VIP blocks

Other signals accompanying the video data are explained in Table 309

Table 309: Chiptop connections

Chiptop Pin Group	Associated signals	Description
DV4	DV4_CLK	Input clock
	DV4_VALID	Input data valid
	DV4_DATA[9:0]	Input data. In VMI mode DV4_DATA[1] is VREF and DV4_DATA[0] is HREF
DV5	DV5_VALID	Input data valid
	{DV5_DATA[7:0],DV5_ERR,DV5_SOP}	Input data. In VMI mode DV5_ERR is VREF and DV5_SOP is HREF

Data routing and video modes:

SD video mode

The VIP supports SD mode, where interleaved data (YUV) comes on the DV4 input. The DV5 is not used in the SD mode.

HD mode

The VIP supports this mode, where Y data is expected on DV5[9:0] and U/V data is expected on DV4[9:0].

5.9.3 VIP basic operation

5.9.3.1 Input source

The VIP accepts data from two sources:

- External Video Input Port or
- Internal Test Pattern Generator

The input signal is selected by the Video Timing Control block, shown in Figure 217

The Video Mode Control Block recovers sync information from the external sync signals or can decode sync signals that are encoded into the video data stream. The incoming encoded sync signal format is based upon D1 video timing reference codes, which also include protection bits (see ITU-R-656). The video timing control block also detects ANC header information according to the ITU-R-1364 standard.

5.9.3.2 Test Pattern Generator

The Test Pattern Generator produces a video stream with a pixel frequency of half the VIP input clock e.g., the 27 MHz encoder clock by programming the clock selection block accordingly.

The sync generation is NTSC-like, with 525 lines per frame and 858 pixels per line. The active video range is 720x462 bordered by a white frame.

The test pattern is shown in Figure 219, and contains the following elements:

- White 2-pixel wide frame– size 720x462
- Color bar– white 100%, yellow 75%, cyan 75%, green 75%, magenta 75%, red 75%, blue 75%, and black 0%
- Grey ramp– full value range 0-255

- Vertical multiburst
- Horizontal multiburst– first rectangle solid in odd, second solid in even field
- Vertical lines
- Moving cursor
- Test pattern

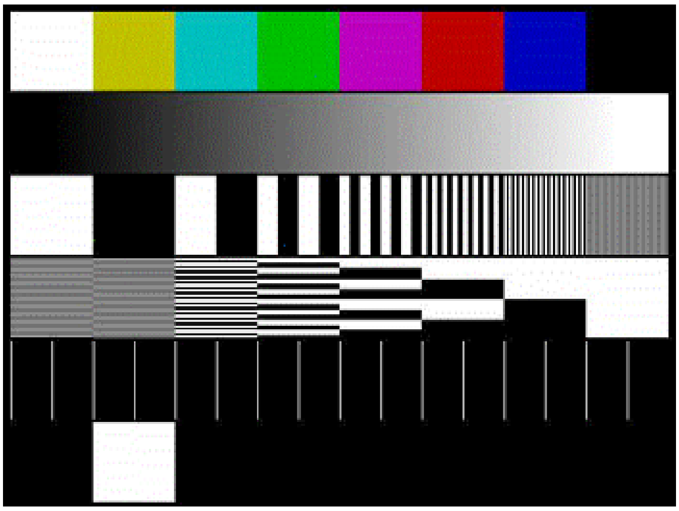


Fig 219.Test Pattern

To capture a picture using the build-in test pattern generator (odd and even field), set up the registers as shown in Table 310 to start capturing at the upper left corner of the white frame.

Table 310: Test Pattern Generator setup

Mode	Reference	Window_Start (0x140)	Window_End (0x144)
NTSC	HREF- / VREF+	8a,0 (138,0)	359,f1 (857,241)
PAL	HREF- / VREF+	90,0 (144,0)	35f,11f (863,287)

5.9.3.3 Input formats

The VIP accepts the following external video input streams:

- 8/10-bit data with encoded [EAV/SAV] syncs YUV 4:2:2 (D1 mode)
- 8-bit data with external [HREF, VREF] syncs YUV 4:2:2 (VMI mode)
- 8/10-bit or 16/20-bit raw data samples (RAW mode)
- 16/20-bit video data on 2 groups of pins for Y and multiplexed U/V with both encoded [SAV/EAV] or explicit [hrefhd,vrefhd,frefhd] syncs (dual_stream or HD mode)

The YUV 4:2:2 sampling scheme assumed by all modes is defined by CCIR 601.

5.9.3.4 D1 mode

The D1 Mode expects an 8/10-bit 4:2:2 video data stream (defined by CCIR 656) with syncs encoded in the video data stream[7]. Timing reference codes recognized are 80h, 9Dh, ABh, B6h, C7h, DAh, ECh and F1h. Single bit errors in the reference codes are corrected, but double bit errors are rejected. The supported mode is shown in Figure 220.

This is strictly a single-stream mode, where VIP captures on (DV4[9:0]) either 10-bit or 8-bit (MSB aligned, with DV4[1:0] unused) multiplexed YUV video data with embedded syncs. The DUAL_STREAM, Video Input Format, 0x100, is programmed to 0 in this mode.

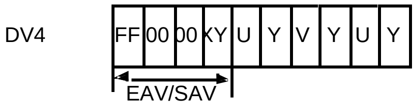


Fig 220.D1 data stream

5.9.3.5 VMI mode

The VMI Mode is an 8-bit YUV (4:2:2) mode with external horizontal and vertical reference signals, which follows the VMI protocol. Chrominance and luminance input samples are multiplexed into a single 8-bit input data stream on DV4. The Field Identifier (FID) is derived from the horizontal and vertical sync timing relation.

This is also a single-stream mode, where VIP captures on (DV4[9:0]) 8-bit VMI data with explicit syncs, where DV4[9:2] correspond to VMI data and DV4[1:0] correspond to VREF and HREF respectively. The DUAL_STREAM, 0x100, is programmed to 0 in this mode.

5.9.3.6 Raw mode

In Raw Mode, valid 8-bit or 10-bit data are continuously captured and written into system memory. Both single and dual streams are supported in this mode. The DUAL_STREAM, 0x100, can therefore be programmed to either 1 or 0 in this mode.

In the single stream mode (DUAL_STREAM 0x100), 8-bit data (DV4[9:2]) is captured as it is but 10-bit data (DV4[9:0]) is extended to 16 bits by either adding leading zeros or by sign-extension.

In the dual stream mode (DUAL_STREAM 0x100), only 8 MSBs of the 10-bit data are valid for each of DV4 and DV5. Two 8-bit data, DV4[9:2] and DV5[9:2], are captured simultaneously from the 8 upper bits of both the channels, for both 8-bit or 10-bit modes, and packed into one 16-bit entity. DV4 and DV5 data occupy the 8 LSBs and 8 MSBs respectively, of the packed 16 bit result.

Raw Mode is only available in the auxiliary capture path of the VIP. It can be enabled independent of D1 or VMI mode.

5.9.3.7 HD mode

This is strictly a dual_stream mode (DUAL_STREAM register = 1), where VIP expects 10-bit Y and 10-bit U/V data on 2 separate inputs (Channels A and B); the U/V data is time multiplexed. In order to support a number of external decoders, this mode has been implemented to work not only with embedded or encoded syncs where EAV and SAV

[7] For compatibility with 8-bit D1 interfaces, the two LSBs are not used for timing reference extraction (as defined in CCIR 656-2)

codes are embedded in the data, but also with explicit syncs where the synchronization reference is provided explicitly via HREF, VREF, and FREF signals (as specified in the implementation of the TDA9975 decoder from Philips and the HMP8117 decoder from Intersil). The supported mode is shown in Figure 221. Note that for detecting the embedded sync in this HD mode, the code is expected to be in the U/V stream; to this end, the current design checks only one of the streams, the U/V stream, for the presence of the embedded codes, assuming that any information embedded in the Y stream is identical (see ITU BT 1120, SMPTE 274M standards). The DUAL_STREAM register is programmed to 1 in this mode.

Remark: The explicit sync signals are used only in the HD or dual_stream mode.

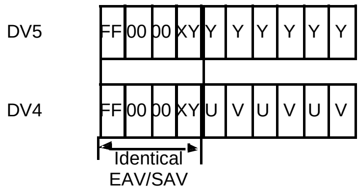


Fig 221.HD dual data stream

Table 311 provides a quick checklist of implemented input formats, where an X designates the presence (support) of the corresponding feature.

5.9.3.8 Video data path

Table 311: Video input formats

Video modes		Single stream (YUV)			Dual stream (Y and U/V)		
		Embedded sync.	Explicit sync.	No Sync.	Embedded Sync.	Explicit Sync.	No Sync.
D1	8-bit	X					
	10-bit	X					
VMI	8-bit		X				
	10-bit						
RAW	8-bit			X			X
	10-bit			X			X
HD	8-bit				X	X	
	10-bit				X	X	

The relation between the video input formats and the supported data stream is shown in Table 312.

Table 312: Relationship Between Input Formats and Video Data Capture

Input Modes		Single stream (YUV) video data	Dual stream (Y and U/V) video data
D1	8-bit	X	
	10-bit	X	

Table 312: Relationship Between Input Formats and Video Data Capture

Input Modes		Single stream (YUV) video data	Dual stream (Y and U/V) video data
VMI	8-bit	X	
HD	8-bit		X
	10-bit		X

5.9.3.9 Video Data Flow

The video datapath dataflow for the VIP is shown in Figure 222

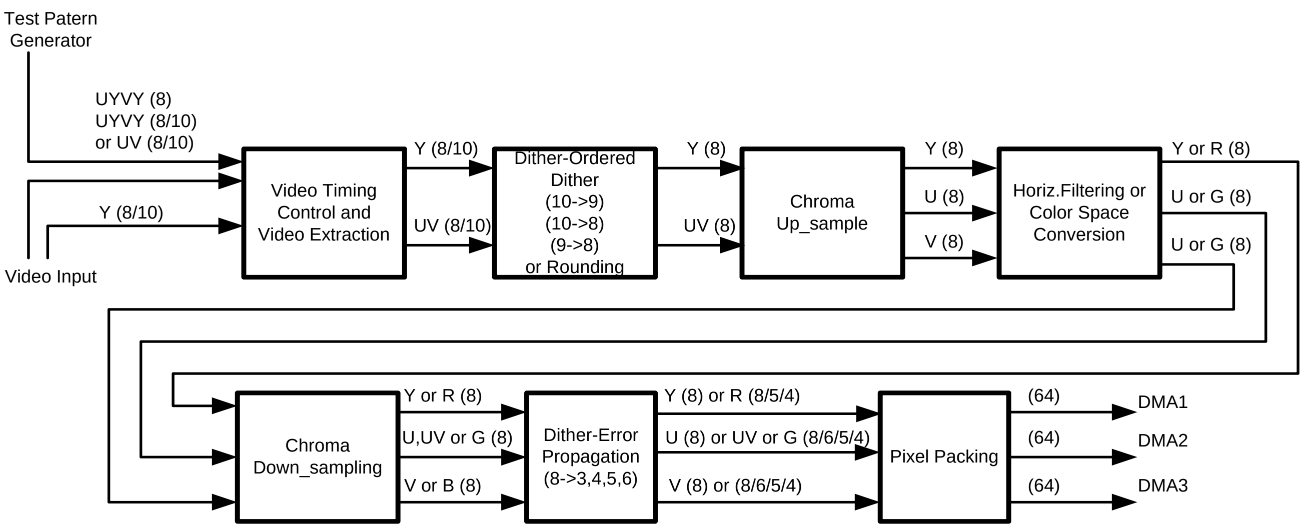


Fig 222.Video Data Flow

5.9.3.10 Video Data Acquisition

The Video and Auxiliary Data Extract block receives a continuous pixel stream from the Video Timing Control block and outputs active window data and synchronization signals. Bit fields in the windowing registers specify the start and end of the source windows relative to the reference edges of H and V syncs and size of the target windows.

5.9.3.11 Internal Timing

Window start is defined relative to either the rising or falling edges of the VREF and HREF inputs (or similar D1 events), as shown in Figure 223

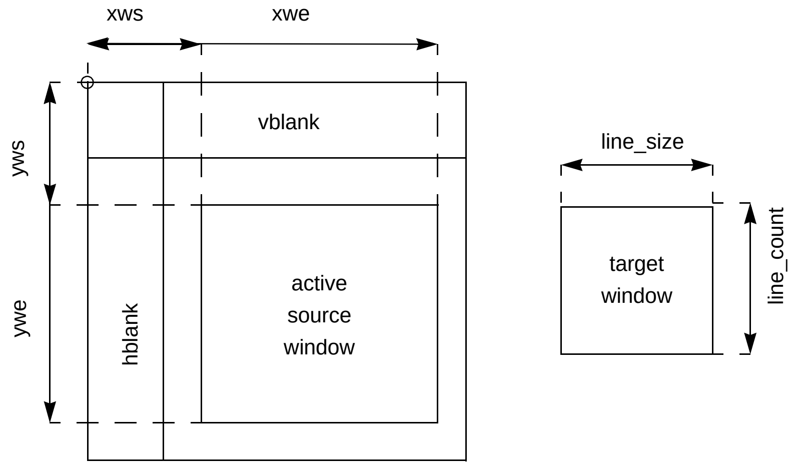


Fig 223.Source and Target Window Parameters

The first qualified data aligned with the REHS reference edge is interpreted as a U sample. If the UYVY data stream is out of sync, it can be realigned with the vsra bits in register 0X100.

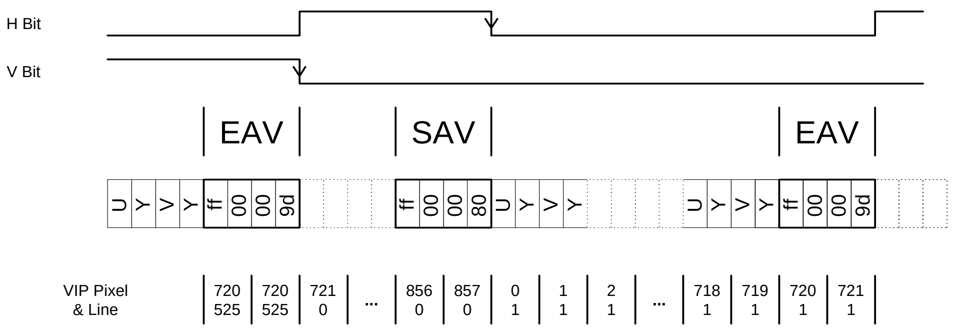


Fig 224.Acquisition Window Counter Reference

For an example showing how to setup the windower and scaler to capture the entire test pattern, refer to Table 310, Figure 223 and Figure 224.

5.9.3.12 Field Identifier Generation

The Field Identifier in the D1 mode is extracted from the F bit in every valid video header, whereas in the VMI mode, the same is derived from the value of the HREF signal during the negative edge of the VREF signal. The Field Identifier timing is illustrated in Figure 225, and Table 313 shows various Field Identified generation modes. Note that instead of using the Field Identifier derived from the video stream, it can also be forced to zero or forced to toggle after each new incoming field; the forced value takes effect after the selected vertical reference edge occurs at the input. The SF bit controls how the Field Identifier value is interpreted. Any change of the Field Identifier interpretation takes effect immediately.

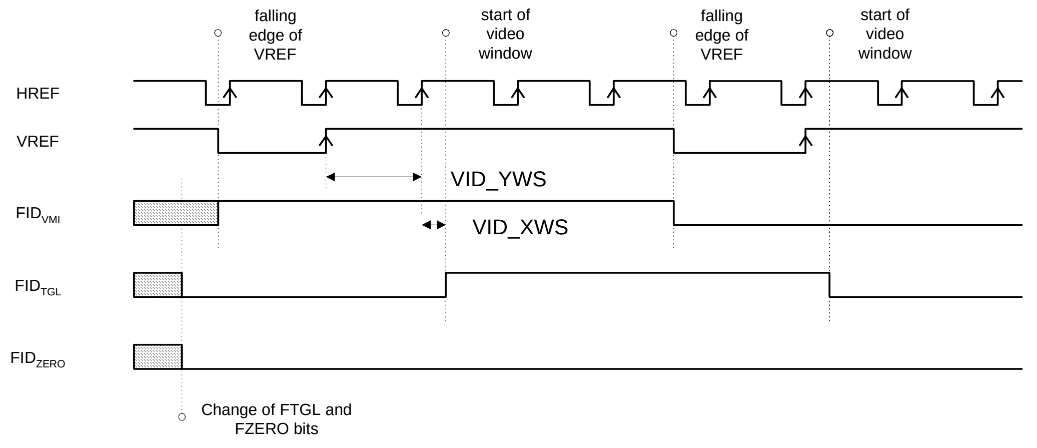


Fig 225.Field Identifier Timing

Table 313: Field Identifier Generation Modes

VDI8	HREF	VSEL	FZERO	FTGL	Change at	FID	FID	FSWP	Meaning
x	f	VMI	0	0	negedge VREF	!f	0	0	Odd
f	x	D1	0	0	valid D1 Header	f	1	0	Even
x	x	x	1	0	immediately	0	0	1	Even
x	x	x	x	1	immediately [1]	0,1,0,..	1	1	Odd

[1] FID toggles after detection of video window start.

5.9.3.13 Video Acquisition Window

The start location of the window to be captured, relative to the input stream, is specified in the Window Start registers, 0X140 (VID_XWS, VID_YWS). The stop location of the window to be captured, relative to the input stream, is specified in the Window End register, 0X144 (VID_XWE, VID_YWE). Additional Target window cropping, which might be necessary after scaling, can be done with the LINE_SIZE and LINE_COUNT values in the Target Window Size register, 0X304.

5.9.3.14 Dithering of the Video Data

There are two identical Dither units, Pre-Dither and Post-Dither, capable of 10->9, 10->8 and 9->8 dithering/rounding with saturating values. These two dither units are cascaded together on the video data path. The two dither units are independent of each other, and controlled with separate MMIO registers. Input samples are assumed to be left (MSB) aligned on the 10 bit input bus. Output samples are left aligned (MSB) on the 10 bit output bus. Both Dither units need to be disabled for an 8-bit input data stream, to avoid unexpected results. The Dither units can be used when the bit precision of samples needs to be limited, while preventing quantization effects in areas with almost uniform levels of shades in a picture.

The following discussion refers to a single Dither unit, either Pre- or Post-Dither

The dither unit processes up to 10 bit inputs. It receives all the three components, Y, U and V, on two 10 bit input buses, and dithers/rounds them down to 8 or 9 bits. Dithering can be enabled separately for luma (Y) and chroma (U and V) components. If the dither unit is enabled but dithering is disabled, rounding, instead of dithering, is performed.

Whenever dithering is enabled, the dithering process alternates its activity between adjacent pixels: either every pixel or every two pixels. Furthermore, any combination of three alternation patterns can be selected: line, field, and frame alternations.

The Dither units are controlled by PRE_DITHER_CTRL and POST_DITHER_CTRL MMIO registers, for the pre- and post-dither units, respectively. Immediately after the unit is enabled, it waits for the beginning of the following captured image before it actually starts to operate.

Enabling the dither unit resets its internal state.

5.9.3.15 The Dither Mechanism

The operation mode is programmable via MODE in the dither-unit control register. The three available modes are 10-bit input down to 8 bits of output, 10-bit input down to 9 bits of output, and 9 bit input down to 8 bits of output. Note that 8-bit input samples are not changed when passed through the Dither unit (the 8 output MSBs are identical to the 8 input MSBs, but the 2 output LSBs are changed by the dither unit).

The Dither unit independently dithers all the three components Y, U, and V in the same way. Each input pixel is processed independently in the sense that the value of the other inputs do not affect the processing of the current input.

The unit is enabled with DITHER_ENABLE. The programmer can select which components are dithered; with DITHER_Y for the luma components, and DITHER_UV, independent of Y, for the chroma components. When the dither unit is enabled, a component that is not selected for dithering goes through rounding. The final value of all components is saturated at 1023, which is the largest value represented by the 10 bit output.

Whenever the dithering operation is enabled, the process of dithering alternates between successive pixel-components, either every pixel or every two pixels, in the same image line. This option is programmable with DOUBLE_PIXEL_ALT for Single or Double pixel alternation.

There are another three dithering options that can be enabled or disabled independently: alternate processing between successive lines, fields and frames.

5.9.3.16 Enabling the Dither Units

Immediately after the Dither unit is enabled or after a reset, the unit waits for the beginning of a newly-captured image. Only then the unit starts dithering. Once the Dither unit is operational (enabled), it keeps track of the order in which the images arrive: refer to the very first image at the unit dither as the even image, the second image as the odd image, and so on. A frame here is defined as two images: an even image followed by an odd image. This maintained state does not depend on the selected alternation options, it is maintained as long as the Dither unit is enabled. Any alternative activity corresponds to the internally-maintained state of a frame and field (even or odd) and has nothing to do if the signal is coming from the top or the bottom field.

Dithering operation also distinguishes between even and odd pixel-components of the same type (either Y, U or V) in a line. The first occurrence of a Y or U or V component in the first line in the first received image is considered to be an even occurrence (or set).

5.9.3.17 Horizontal Video Filters (Sampling, Scaling, Color Space Conversion)

Interpolation Filter (Upsampling): All horizontal video processing is based on equidistant sampled components. All 4:2:2 video streams, therefore, have to be upsampled before being scaled horizontally. The interpolation FIR filter used can interpolate interspersed or co-sited chroma samples. Mirroring of samples at the field boundaries compensate for run-in and run-out conditions of the filter.

The following coefficients are used:

- co-sited: A=(1) and B=(-3,19,19,-3)/32
- interspersed: C=(-1,5,13,-1)/16 and D=(-1,13,5,-1)/16

Decimation Filter (Downsampling): After horizontal processing, the chrominance may be down-sampled to reduce memory bandwidth or allow a higher-quality vertical processing not available otherwise.

The following coefficients are used:

- co-sited: low pass A=(1,2,1)/4 or sub-sample A=(0,1,0)
- interspersed: B=(-3,19,19,-3)/32

Normal Polyphase Filter (Horizontal Scaling): The normal polyphase filter can be used to zoom up (upscale) or downscale a video image. Depending on the number of components, the filter is used with 6 taps (three-component mode) or 3 taps (four-component mode).

Color Space Matrix Mode: In addition to normal and transposed polyphase filtering (scaling), the FIR filter structure can instead be programed to perform color space-conversion. A dedicated set of registers holds the coefficients for the color-space matrix. Horizontal scaling and color space conversion are mutually exclusive.

5.9.3.18 Video Data Write to Memory

The VIP can produce a variety of output formats. Video formats range from a single-component up to three-component formats (like a 4:4:4 YUV). Up to three write planes can be defined. On the input, the video format is restricted to YUV 4:2:2 as defined in ITU-R-656 or 8/10 raw data. On the output, true color and compressed formats are supported.

The Pixel Packing Unit takes care of quantization and packing of the color components into 64-bit units. A list of the most common video formats supported is shown in Table 314. Packing of a pixel into 64 bit units is always done from right to left while bytes within one pixel unit are ordered according to the endianess settings (specified by the global endianess pin; endianess bit in the output format register can, however, invert that signal).

Table 314: Output Pixel Formats

<table>
<tr><th>Format</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td rowspan="3">planar YUV (4:4:4, 4:2:2) or RGB</td><td colspan="24">plane #1</td><td colspan="8">Y8 or R8</td></tr>
<tr><td colspan="24">plane #2</td><td colspan="8">U8 or G8</td></tr>
<tr><td colspan="24">plane #3</td><td colspan="8">V8 or B8</td></tr>
</table>

Table 314: Output Pixel Formats

Format	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
semi planar YUV (4:2:2)	plane #1																								Y8 or R8							
	plane #2																								U8/V8							
packed 4/4/4 RGBa																	alpha				R4				G4				B4			
packed 4/5/3 RGBa																	alpha				R4				G5					B3		
packed 5/6/5 RGB																	R5					G6						B5				
packed YUY2 4:2:2																	U8 or V8								Y8							
packed UYVY 4:2:2																	Y8								U8 or V8							
packed 888 RGB(a)	(alpha)								R8 or Y8								G8 or U8								B8 or V8							
packed 4:4:4 VYU(a)	(alpha)								V8								Y8								U8							

Table 314 shows the location of the first ’pixel unit’ within a 64-bit word in the little endian mode. The selected endianess affects the position of the components within a multi-byte pixel unit.

Remark: VIP does not explicitly support a 4:2:0 memory format. Such a format can be obtained by discarding partial data written to memory.

Capture Enable Mode: Using the cfen bits, video capture can be limited to odd or even or both fields. If both fields are to be captured, the capture starts with the next odd field.

The status of the osm (one-shot) bit in the mode-control-register specifies the capture mode (one-shot or continuous):

- If osm=0, the corresponding incoming video stream is captured continuously. For example, in a video conference application the vanity image would be a continuous stream to the frame buffer.
- If osm=1, the corresponding incoming video stream is captured one field or frame at a time (depending on the cfen bits).

Programming hint: In a video conference application the captured image would be a one-shot stream to the host memory. If you write osm=1 and select field/frame in the register, it is captured on the next VSYNC and cfen bits are cleared to 0. To capture the next image, the cfen and osm bits are reprogrammed.

Address Generation: The line address is generated by loading the base address from the corresponding register set at the beginning of each field and adding the line pitch to it at the beginning of every new line.The lower three bits of the first three base address registers are used as an intra-long-word offset for the left-most pixel components of each line. The offset has to be a multiple of the number of bytes per component.

Double-Buffer Mode: To avoid line tear caused by trying to display a frame at the same time that it is being updated, a double-buffer mode is available. In this double-buffer mode, a second set of DMA base addresses is available. After capturing and storing one complete frame in the location described by one set, the other set is used for the next frame.

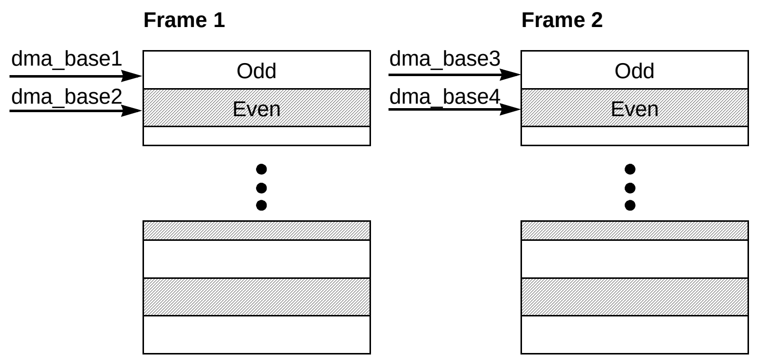


Fig 226.Double Buffer Mode

5.9.3.19 Auxiliary Data Path

The relationship between the input data modes and the supported auxiliary capture modes is shown in Table 315.

Table 315: Relationship Between Input Formats and Data Capture

Input modes		Single-Stream Auxiliary Data			Dual-Stream Auxiliary Data		
		AUX	ANC	RAW	AUX	ANC	RAW
D1	8-bit	X	X				
	10-bit	X	X				
VMI	8-bit	X			X		
RAW	8-bit			X			X
	10-bit			X			X
HD	8-bit				X		
	10-bit				X		

Auxiliary Data Flow: The auxiliary data flow is shown below.

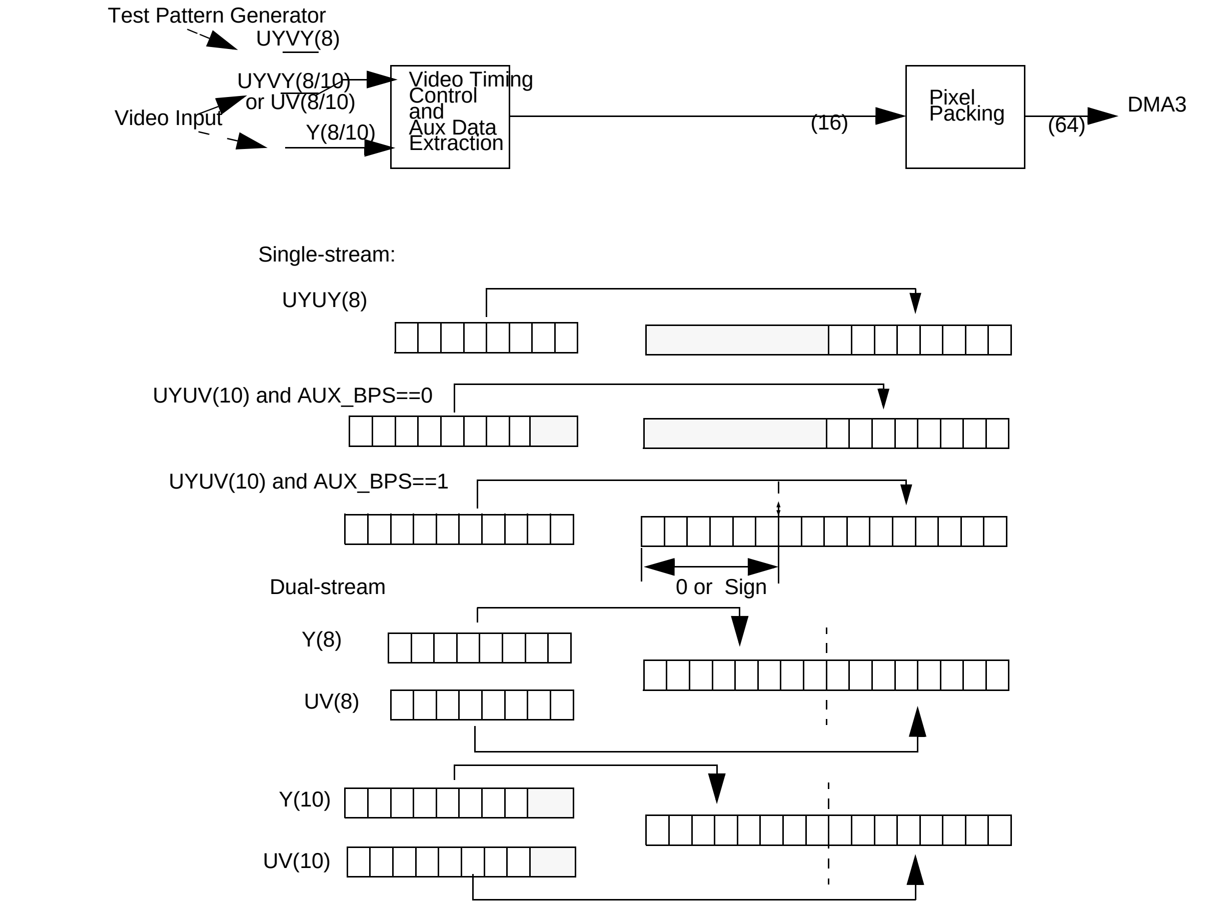


Fig 227.Auxiliary Data Flow

Auxiliary Data Acquisition: Capturing auxiliary data utilizes the same DMA engine used for the third video plane. Capture of overlapping Video and Auxiliary regions is, therefore, only possible when semi-planar or packed formats are being used. Data can be captured in either 8- or 10-bit modes. In the single-stream mode, 10-bit data is extended to 16 bits by either adding leading zeros or by sign-extension. In dual stream mode, only 8 MSBs of a 10-bit data are valid; two such MSB sets (2x8-bit data) are captured simultaneously, for either 8-bit or 10-bit modes, and packed to form a resultant 16-bit unit. thus, DV4 data (8 bits) and DV5 data (8 bits) are located at the 8 LSBs and the 8 MSBs respectively, of the packed 16-bit data.

Three different types of auxiliary data capture are defined:

- Ancillary Data Capture (ANC)
- Auxiliary data acquisition window (AUX)
- Raw data capture (RAW)

A buffer-size register can be used to limit data acquisition by size (one shot mode) or define a ring-buffer length.

Even though ANC and AUX capture can be enabled separately, simultaneous capture of ANC and AUX is not advisable. Timing and sequence of ANC and AUX data are not necessarily related and therefore are likely to lead to unpredictable results if simultaneous capture is attempted.

Ancillary Data Capture: Ancillary Data, embedded in the stream and marked by ITU-R-1364 header codes, can be decoded and extracted for software processing. AUX_BPS register specifies the number of bits to be captured per ANC sample. In the 8-bit mode, two LSBs of the 10-bit data bus are ignored. Ancillary data capture is not supported in the dual stream mode.

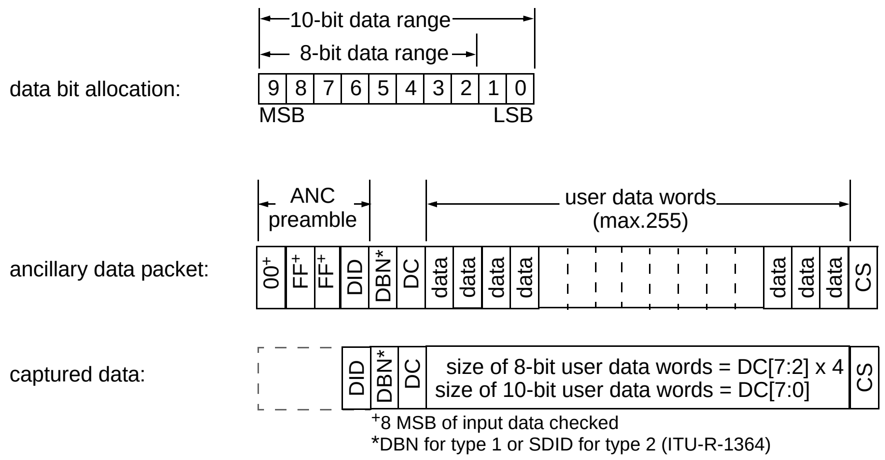


Fig 228.ANC Data Structure

Four sequential ANC preamble bytes, 00, FF, FF and a qualified DID word, enable ANC data capture. A qualified DID word is defined:

- masked AUX_ANC-enabled ID matched (see Figure 228)
- bit 8 is even parity for bit 7-0(10-bit data mode) / bit 7-2(8-bit data mode)
- bit 9 = not bit 8

2 LSBs of both ID_MASK_0 and ID_MASK_1 need to be programmed to 2’b00 in 8-bit data mode to prevent unexpected results.

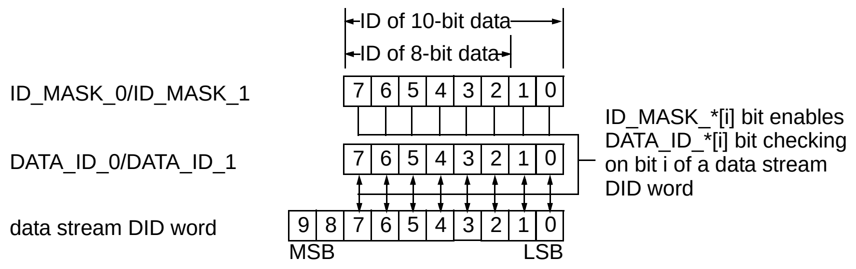


Fig 229.ANC Masked ID Checking

In the type 1 case, the data block number (DBN) distinguishes successive data packets with a common data ID. In the type 2 case, the DID is followed by the secondary identification (SDID). The captured packed length is taken form the data count (DC) byte.

A value of DC=0 captures exactly four data words consisting of DID, DBN (or SDID), DC and checksum (CS). If DC is not equal to 0, additional user data words defined by DC are captured. Parity bits for DBN (or SDID) and DC bytes are not checked.

Auxiliary Data Acquisition Window: The auxiliary data acquisition window can be used to capture either VBI data or an additional region of video data. It provides yet another capture-window. The field identifier is compared against AUX_CFEN bits at the start of the programmed window to control whether a field is captured or not. The start and end points of the auxiliary window are defined by the AUX Window Start and End registers at offsets 0X180 and 0X184 (AUX_XWS, AUX_YWS, AUX_XWE, AUX_YWE).

The AUX_XWS parameter specifies the number of the first pixel to be captured after the HREF reference edge.The AUX_YWS parameter specifies the number of the first line to be captured after AUX reference edge.

The AUX_XWE parameter specifies the number of the last pixel to be captured.The AUX_YWE parameter specifies the number of the last line to be captured.

Pixel and lines start counting at 0.

Raw Data Capture: Raw data capture overrides ANC or AUX data capture modes when enabled. In this raw capture mode, any validated data at the video port is captured regardless of external or embedded synchronization signals.

AUX Data Write to Memory: Auxiliary data capture formats, for writing into the frame buffer, are limited to raw luma and chroma samples in 8 or 10 bit formats (extended to 16 bit). Optionally, writing of chroma samples can be omitted.

Capture Enable Mode: The aux_cfen bits specify the fields from which the device is to capture the AUX data. If cfen=0, no auxiliary data is captured. Once the capture of an auxiliary window has started, resetting these bits has no effect until the end of the video window.

The aux_anc bits specify the type of ancillary data blocks to be fetched. Once the capture of an ANC block has started, resetting of these bits has no effect until the end of the data block.

The aux_raw bit enables continuous capturing of raw samples regardless of external or internal syncs. If raw capture is enabled, aux_cfen and aux_anc bit settings are ignored.

The aux_bsize value specifies, in number of bytes, the size of the buffer available for AUX data.

The aux_pitch bits specify the AUX line pitch, i.e., the difference in the address from a pixel on a line to the same pixel on the next line, when pitch mode is enabled. Pitch mode is also defined for the ANC data capture, where each packet is treated as a new video line.

The aux_osm bit can be used to automatically limit capture by stopping after any wrap condition is reached. End of AUX window wrap condition also applies to ANC capture, even if AUX capture is disabled.

The data buffered in the local FIFOs is flushed when a wrap condition is reached or in pitch mode at the end of each video line or ANC packed. For the raw mode, the data is also flushed when disabling raw capture.

Address Generation: The AUX DMA base address register provides 26 bits to specify a destination address for storing the AUX data in the frame buffer. The address generation is similar to that of video capture, except for the missing double-buffer and field modes.

The relationship between the input data formats and the different video and auxiliary data capture scenarios is shown in Table 316. It is important to note, however, that the current VIP design does not support mixing of AUX and ANC data.

Table 316: Relationship Between Input Formats and Data Capture

Video Modes		Single stream (YUV)				Dual stream (Y and U/V)			
		Video Data	Auxiliary Data			Video Data	Auxiliary Data		
			AUX	ANC	RAW		AUX	ANC	RAW
D1	8-bit	X	X	X					
	10-bit	X	X	X					
VMI	8-bit	X	X			X	X		
RAW	8-bit				X				X
	10-bit				X				X
HD	8-bit					X	X		
	10-bit					X	X		

5.9.3.20 Interrupt Generation

The VIP contains a DVP-compliant interrupt generation mechanism. Interrupts can be generated for the following events:

- start of video
- end of video (written to memory)
- start of AUX in
- end of AUX out (written to memory)
- line threshold
- pipeline error (due to illegal scaling ratio e.g. >2x scaling or memory bus bandwidth error (fifo overflow))

In addition to these interrupts, the VIP module also provides last-pixel-in signals, for the VBI and video capture modes, to the GPIO block for timestamping.

5.9.4 Register descriptions

5.9.4.1 Register summary

Table 317: VIP Register Summary

Offset	Name	Description
0x0	VIP_Mode_Control	
0x20	ANC_Identifier_Codes_Field_0	
0x24	ANC_Identifier_Codes_Field_1	
0x40	VIP_Line_Threshold	
0x100	Video_Input_Format	
0x104	Video_Test_Pattern_Generator_Control	
0x140	Video_Acquisition_Window_Start	
0x144	Video_Acquisition_Window_End	
0x160	Pre_Dither_Control	
0x164	Post_Dither_Control	
0x180	Auxiliary_Acquisition_Window_Start	
0x184	Auxiliary_Acquisition_Window_End	
0x200	Initial_Zoom	
0x204	Phase_Control	
0x208	Initial_Zoom_delta	
0x20C	Zoom_delta_change	
0x220	Color_space_matrix_coefficients_C00_C02	
0x224	Color_space_matrix_coefficients_C10_C12	
0x228	Color_space_matrix_coefficients_C20_C22	
0x22C	Color_space_matrix_offset_coefficients_D0_D2	
0x230	Color_space_matrix_offset_coefficients_E0_E2	
0x284	Color_Key_Components	
0x300	Video_Output_Format	
0x304	Target_Window_Size	
0x340	Target_Base_Address_1	
0x344	Target_Line_Pitch_1	
0x348	Target_Base_Address_2	
0x34C	Target_Line_Pitch_2	
0x350	Target_Base_Address_3	
0x354	Target_Base_Address_4	
0x358	Target_Base_Address_5	
0x35C	Target_Base_Address_6	
0x380	Auxiliary_Capture_Output_Format	
0x390	Auxiliary_Capture_Base_Address	
0x394	Auxiliary_Capture_Line_Pitch	

Table 317: VIP Register Summary *...continued*

Offset	Name	Description
0x800	Horizontal_LS	- 69FCCoefficient Table #1 Taps 0-2 (Horizontal)
0x804	Horizontal_MS	- 69FCCoefficient Table #1 Taps 3-5 (Horizontal)
0xFE0	Interrupt_Status	
0xFE4	Interrupt_Enable	
0xFE8	Interrupt_Clear	
0xFEC	Interrupt_Set	
0xFFC	Module_ID	

5.9.4.2 Register table

Table 318: Video Input Processor (VIP) Registers

Bit	Symbol	Access	Value	Description
Offset 0x0 - VIP_Mode_Control				
31:30	Vid_cfen[1:0]	R/W	0x0	Video window capture field enable 0: capture disabled 1: capture odd only 10: capture even only 11: capture both
29	Vid_osm	R/W	0x0	Video capture one shot mode 0: continuously capture fields selected by CFEN 1: capture fields selected by CFEN only once
28	Vid_fseq	R/W	0x0	video capture field sequence setting has no effect unless VID_CFEN is set to capture both 0: capture fields starting with any field 1: capture fields starting with odd field
27:26	Aux_cfen[1:0]	R/W	0x0	Auxiliary window capture enable 0: capture disabled 1: capture odd only 10: capture even only 11: capture both

Table 318: Video Input Processor (VIP) Registers *…continued*

Bit	Symbol	Access	Value	Description
25	Aux_osm	R/W	0x0	Auxiliary capture one shot mode 0: when auxiliary wrap event is reached, buffer wraps around 1: when auxiliary wrap event is reached, capturing stops
24	Aux_fseq	R/W	0x0	Auxiliary capture field sequence setting has no effect unless AUX_CFEN is set to capture both 0: capture fields starting with any field 1: capture fields starting with odd field
23:22	Aux_anc[1:0]	R/W	0x0	ANC data capture enable 0: no ANC data captured 1: odd ANC field blocks. (masked DATA_ID_0 bit matched) 10: even ANC field blocks. (masked DATA_ID_1 bit matched) 11: odd/even ANC field blocks. (masked DATA_ID_* bit matched)
21	Aux_raw	R/W	0x0	Auxiliary raw capture enable when enabled, AUX_ANC and AUX_CFEN settings are ignored 0: raw capture disabled 1: raw capture enabled, all samples will be captured
20:18	Reserved[2:0]	R/W	0x0	
17	Rst_on_err	R/W	0x0	Reset on error Writing a one into this bit will automatically reset the block in case of a pipeline error (e.g. illegal scaling ratio / FIFO overflow)
16	Soft_reset	W	0x0	Soft reset Writing a one into this bit will reset the block
15	Reserved2	R/W	0x0	
14	Iff_clamp	R/W	0x0	Clamp mode for IFF (affects U/V only) 0: clamp to 0-255 1: clamp to 16 - 240 (CCIR range)
13:12	Iff_mode[1:0]	R/W	0x0	Interpolation mode 00: bypass 01: reserved 10: co-sited 11: interspersed

Table 318: Video Input Processor (VIP) Registers *…continued*

Bit	Symbol	Access	Value	Description
11	Reserved3	R/W	0x0	
10	Dff_clamp	R/W	0x0	Clamp mode for DFF (affects U/V only) 0: clamp to 0-255 1: clamp to 16 - 240 (CCIR range)
9:8	Dff_mode[1:0]	R/W	0x0	Decimation mode 00: bypass 01: co-sited (sub sample) 10: co-sited (low pass) 11: interspersed
7:4	Reserved4[3:0]	R/W	0x0	
3	Hsp_clamp	R/W	0x0	Clamp mode for HSP 0: clamp to 0-255 1: clamp to CCIR range defined by bit 2
2	Hsp_rgb	R/W	0x0	Color space mode, defines CCIR clamping range for HSP 0: processing in YUV color space 1: processing in RGB color space
1:0	Hsp_mode[1:0]	R/W	0x0	Horizontal processing mode 00: bypass mode 01: color space matrix mode 10: normal polyphase mode 11: transposed polyphase mode
Offset 0x20 - ANC_Identifier_Codes_Field_0				
31:16	Reserved[15:0]	R/W	0x0000	
15:8	Id_mask_0[7:0]	R/W	0xFC	Mask for enabling bit checking on ANC identifier code For each ID_MASK_0[i] bit, 1: enable DATA_ID_0[i] bit checking 0: disable DATA_ID_0[i] bit checking
7:0	Data_id_0[7:0]	R/W	0x44	ANC identifier code
Offset 0x24 - ANC_Identifier_Codes_Field_1				
31:16	Reserved[15:0]	R/W	0x0000	
15:8	Id_mask_1[7:0]	R/W	0xFC	Mask for enabling bit checking on ANC identifier code For each ID_MASK_1[i] bit, 1: enable DATA_ID_1[i] bit checking 0: disable DATA_ID_1[i] bit checking
7:0	Data_id_1[7:0]	R/W	0x54	ANC identifier code
Offset 0x40 - VIP_Line_Threshold				
31:11	Reserved[20:0]	R/W	0x0000 00	
10:0	Lcthr[10:0]	R/W	0x000	Video line count threshold Line threshold status bit is set if video line count (SVLC) reaches this value Note: It is possible to have multiple interrupts per field at different line counts, by re-programming the threshold value in this register from the ISR.
Offset 0x100 - Video_Input_Format				

Table 318: Video Input Processor (VIP) Registers *…continued*

Bit	Symbol	Access	Value	Description
31:30	Vsra[1:0]	R/W	0x0	Video stream realignment 1x = reserved 0: normal 1: ignore 1st sample after HREF
29:26	Reserved[3:0]	R/W	0x0	
25	Synchd	R/W	0x0	HD sync select 0: embedded sync 1: explicit sync
24	Dual_stream	R/W	0x0	Dual video data stream enable 0: single video data stream mode 1: dual video data stream mode
23:21	Reserved2[2:0]	R/W	0x0	
20	Nhdaux	R/W	0x0	header detect during AUX window 0: D1 header detection enabled inside AUX window 1: D1 header detection disabled inside AUX window
19	Npar	R/W	0x0	Parity check disable 0: parity check enabled for D1 header detection 1: parity check disabled for D1 header detection
18:16	Reserved3[2:0]	R/W	0x0	
15:14	Vsel[1:0]	R/W	0x0	Video source select 0: reserved 1: video port, encoded sync (D1-Mode) 10: video port, external sync (VMI-Mode) 11: reserved
13	Twos	R/W	0x0	UV data type 0: offset binary 1: twos complement
12	Tpg	R/W	0x0	Test pattern generator 0: video stream selected by VSEL 1: internal test pattern generator
11	Reserved4	R/W	0x0	
10	Fref	R/W	0x0	Field toggle reference mode 0: normal, use VREF 1: toggling Freed bit is used as vertical reference
9	Ftgl	R/W	0x0	Field toggle mode 0: normal 1: free toggle (sequence starts with FID = 0)
8:4	Reserved5[4:0]	R/W	0x00	

Table 318: Video Input Processor (VIP) Registers *…continued*

Bit	Symbol	Access	Value	Description
3	Sf	R/W	0x0	Swap field interpretation 0: odd (first) field = 0, even (second) field = 1 1: odd (first) field = 1, even (second) field = 0
2	Fzero	R/W	0x0	Force FID value to zero 0: field identifier derived from input stream 1: force field identifier value to 0
1	Revs	R/W	0x0	Vertical sync reference edge 0: falling edge / start of active video 1: rising edge / end of active video
0	Rehs	R/W	0x0	Horizontal sync reference edge 0: falling edge / SAV 1: rising edge / EAV
Offset 0x104 - Video_Test_Pattern_Generator_Control				
31	Pal	R/W	0x0	Field generation mode 0: NTSC timing 1: PAL timing
30	Reserved	R/W	0x0	
29	Vsel	R/W	0x0	Vertical timing signal select (will be removed) 0: generate VREF 1: generate VS
28	Hsel	R/W	0x0	Horizontal timing signal select (will be removed) 0: generate HREF 1: generate HS
27	Swap	R/W	0x0	Alternative test pattern 0: normal test pattern 1: test pattern with diagonal patterns, etc.
26	Move	R/W	0x0	Scrolling enable for alternative test pattern 0: no scrolling 1: scrolling enabled
25:0	Reserved2[25:0]	R/W	0x0000000	
Offset 0x140 - Video_Acquisition_Window_Start				
31:28	Reserved[3:0]	R/W	0x0	
27:16	Vid_xws[11:0]	R/W	0x000	Horizontal video window start The pixel co-sited with the reference edge REHS is numbered 0.
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Vid_yws[10:0]	R/W	0x000	Vertical video window start The first line indicated by the reference edge REVS is numbered 0.

Table 318: Video Input Processor (VIP) Registers *…continued*

Bit	Symbol	Access	Value	Description
Offset 0x144 - Video_Acquisition_Window_End				
31:28	Reserved[3:0]	R/W	0x0	
27:16	Vid_xwe[11:0]	R/W	0x000	Horizontal video window end pixels from XWS up to and including XWE are processed
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Vid_ywe[10:0]	R/W	0x000	Vertical video window end lines from YWS up to and including YWE are processed
Offset 0x160 - Pre_Dither_Control				

31	Pre_Dither_enble	R/W	0x0	Dither Control Enable / disable the dither unit.
				Pre_DITHER_ENABLE = 0 : disable;
				Pre_DITHER_ENABLE = 1 : enable.

30	Pre_Dither_y	R/W	0x0	Dither Y Components Enable / disable dithering of Y pixel-components. Pre_DITHER_Y = 0 : disable (round and clip); Pre_DITHER_Y = 1 : enable (dither).
29	Pre_Dither_uv	R/W	0x0	Dither U and V Components Enable / disable dithering of U and V pixel-components. Pre_DITHER_UV = 0 : disable (round and clip); Pre_DITHER_UV = 1 : enable (dither).
28:27	Pre_Mode[1:0]	R/W	0x0	Mode of Operation Select input and output sizes and the computations carried out: Pre_MODE = 0 : 10->9; Pre_MODE = 1 : 10->8; Pre_MODE = 2 : 9->8; Pre_MODE = 3: Reserved.
26:4	Reserved[22:0]	R	0x000000	
3	Pre_Frame_alt	R/W	0x0	Frame Alternate Enable/disable frame alternation while dithering. Pre_FRAME_ALT = 0 : disable; Pre_FRAME_ALT = 1 : enable.
2	Pre_Field_alt	R/W	0x0	Field Alternate Enable/disable field alternation while dithering. Pre_FIELD_ALT = 0 : disable; Pre_FIELD_ALT = 1 : enable.
1	Pre_Line_alt	R/W	0x0	Line Alternate Enable/disable line alternation while dithering. Pre_LINE_ALT = 0 : disable; Pre_LINE_ALT = 1 : enable.
0	Pre_Double_pixel_alt	R/W	0x0	Single or Double Pixel Alternate Select single or double pixel alternation while dithering. Pre_DOUBLE_PIXEL_ALT = 0 : single pixel alternation; Pre_DOUBLE_PIXEL_ALT = 1 : double pixel alternation.
Offset 0x164 - Post_Dither_Control				

31	Post_Dither_enble	R/W	0x0	Dither Control Enable / disable the dither unit.
				Post_DITHER_ENABLE = 0 : disable;
				Post_DITHER_ENABLE = 1 : enable.

30	Post_Dither_y	R/W	0x0	Dither Y Components Enable / disable dithering of Y pixel-components. Post_DITHER_Y = 0 : disable (round and clip); Post_DITHER_Y = 1 : enable (dither).

Table 318: Video Input Processor (VIP) Registers *…continued*

Bit	Symbol	Access	Value	Description
29	Post_Dither_uv	R/W	0x0	Dither U and V Components Enable / disable dithering of U and V pixel-components. Post_DITHER_UV = 0 : disable (round and clip); Post_DITHER_UV = 1 : enable (dither).
28:27	Post_Mode[1:0]	R/W	0x0	Mode of Operation Select input and output sizes and the computations carried out: Post_MODE = 0 : 10->9; Post_MODE = 1 : 10->8; Post_MODE = 2 : 9->8; Post_MODE = 3: Reserved.
26:4	Reserved[22:0]	R	0x000000	
3	Post_Frame_alt	R/W	0x0	Frame Alternate Enable/disable frame alternation while dithering. Post_FRAME_ALT = 0 : disable; Post_FRAME_ALT = 1 : enable.
2	Post_Field_alt	R/W	0x0	Field Alternate Enable/disable field alternation while dithering. Post_FIELD_ALT = 0 : disable; Post_FIELD_ALT = 1 : enable.
1	Post_Line_alt	R/W	0x0	Line Alternate Enable/disable line alternation while dithering. Post_LINE_ALT = 0 : disable; Post_LINE_ALT = 1 : enable.
0	Post_Double_pixel_alt	R/W	0x0	Single or Double Pixel Alternate Select single or double pixel alternation while dithering. Post_DOUBLE_PIXEL_ALT = 0 : single pixel alternation; Post_DOUBLE_PIXEL_ALT = 1 : double pixel alternation.
Offset 0x180 - Auxiliary_Acquisition_Window_Start				
31:28	Reserved[3:0]	R/W	0x0	
27:16	Aux_xws[11:0]	R/W	0x000	Horizontal auxiliary window start The pixel cosited with the reference edge REHS is numbered 0.
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Aux_yws[10:0]	R/W	0x000	Vertical auxiliary window start The line cosited with the reference edge REVS is numbered 0.
Offset 0x184 - Auxiliary_Acquisition_Window_End				
31:28	Reserved[3:0]	R/W	0x0	
27:16	Aux_xwe[11:0]	R/W	0x000	Horizontal auxiliary window end pixels from XWS up to and including XWE are processed
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Aux_ywe[10:0]	R/W	0x000	Vertical auxiliary window end lines from YWS up to and including YWE are processed
Offset 0x200 - Initial_Zoom				
31:29	Hsp_phase_mode[2:0]	R/W	0x0	Phase mode 0: 64 phases 1: 32 phases 2: 16 phases 3: 8 phases 4: 4 phases 5: 2 phases 6: fixed phase 7: linear phase interpolation (only valid for4 component mode)
28:27	Reserved[1:0]	R/W	0x0	

Table 318: Video Input Processor (VIP) Registers *…continued*

Bit	Symbol	Access	Value	Description
26	Hsp_fir_comp	R/W	0x0	Horizontal filter components 0: three components, 6 tap FIR each 1: four components, 3 tap FIR each (4th component unused) In color space matrix mode this value has to remain zero
25:20	Reserved2[5:0]	R/W	0x00	
19:0	Hsp_zoom_0[19:0]	R/W	0x0000 0	Initial zoom for 1st pixel in line (unsigned, LSB = 2-16) 2 0000 (hex): downscale 50% 1 0000 (hex): no scaling = 2 0 0 8000 (hex): zoom 2 x (transposed: downscale 50%)
Offset 0x204 - Phase_Control				
31	Reserved	R/W	0x0	
30:28	Hsp_qshift[2:0]	R/W	0x0	Quantization shift control used to change quantization before being multiplied with HSP_MULTIPLY. 100 (bin): divide by 16 101 (bin): divide by 8 110 (bin): divide by 4 111 (bin): divide by 2 000 (bin): multiply by 1 001 (bin): multiply by 2 010 (bin): multiply by 4 011 (bin): multiply by 8 Warning: A value range overflow caused by an improper quantization shift can not be compensated later by multiplying with a HSP_MULTIPLY value below 0.5!
27:26	Reserved2[1:0]	R/W	0x0	
25	Hsp_qsign	R/W	0x0	Quantization sign bit
24:16	Hsp_qmultiply[8:0]	R/W	0x000	Quantization multiply control used to compensate for different weight sum in transposed polyphase or color space matrix mode, remaining bits are fraction (largest number is 511/512) Value range: . Instead of using values in the range of the quantization shift HSP_QSHIFT should be modified to gain more precision in the truncated result.
15:13	Reserved3[2:0]	R/W	0x0	
12:0	Hsp_offset_0[12:0]	R/W	0x0000	Initial start offset for DTO
Offset 0x208 - Initial_Zoom_delta				
31:26	Reserved[5:0]	R/W	0x00	
25:0	Hsp_dzoom_0[25:0]	R/W	0x0000 000	Initial zoom delta for 1 pixel in line (signed, LSB = 2-27) used for non constant scaling ratios
Offset 0x20C - Zoom_delta_change				
31:29	Reserved[2:0]	R/W	0x0	
28:0	Hsp_ddzoom[28:0]	R/W	0x0000 0000	Zoom delta change (signed, LSB = 2-40) used for non constant scaling ratios
Offset 0x220 - Color_space_matrix_coefficients_C00_C02				
31:30	Unused[1:0]	R/W	0x0	
29:20	Csm_c02[9:0]	R/W	0x000	Coefficient C02, twos complement
19:10	Csm_c01[9:0]	R/W	0x000	Coefficient C01, twos complement
9:0	Csm_c00[9:0]	R/W	0x000	Coefficient C00, twos complement
Offset 0x224 - Color_space_matrix_coefficients_C10_C12				
31:30	Unused[1:0]	R/W	0x0	

Table 318: Video Input Processor (VIP) Registers *…continued*

Bit	Symbol	Access	Value	Description
29:20	Csm_c12[9:0]	R/W	0x000	Coefficient C12, twos complement
19:10	Csm_c11[9:0]	R/W	0x000	Coefficient C11, twos complement
9:0	Csm_c10[9:0]	R/W	0x000	Coefficient C10, twos complement
Offset 0x228 - Color_space_matrix_coefficients_C20_C22				
31:30	Unused[1:0]	R/W	0x0	
29:20	Csm_c22[9:0]	R/W	0x000	Coefficient C22, twos complement
19:10	Csm_c21[9:0]	R/W	0x000	Coefficient C21, twos complement
9:0	Csm_c20[9:0]	R/W	0x000	Coefficient C20, twos complement
Offset 0x22C - Color_space_matrix_offset_coefficients_D0_D2				
31:29	Unused[2:0]	R/W	0x0	
28	Csm_d2_twos	R/W	0x0	Offset coefficient D2 type 0: unsigned 1: signed
27:20	Csm_d2[7:0]	R/W	0x00	Offset coefficient D2
19	Unused2	R/W	0x0	
18	Csm_d1_twos	R/W	0x0	Offset coefficient D1 type 0: unsigned 1: signed
17:10	Csm_d1[7:0]	R/W	0x00	Offset coefficient D1
9	Unused3	R/W	0x0	
8	Csm_d0_twos	R/W	0x0	Offset coefficient D0 type 0: unsigned 1: signed
7:0	Csm_d0[7:0]	R/W	0x00	Offset coefficient D0
Offset 0x230 - Color_space_matrix_offset_coefficients_E0_E2				
31:30	Unused[1:0]	R/W	0x0	
29:20	Csm_e2[9:0]	R/W	0x000	Offset coefficient E2, twos complement
19:10	Csm_e1[9:0]	R/W	0x000	Offset coefficient E1, twos complement
9:0	Csm_e0[9:0]	R/W	0x000	Offset coefficient E0, twos complement
Offset 0x284 - Color_Key_Components				
31:24	Ckey_alpha[7:0]	R/W	0x00	Alpha value Defines the alpha value to be used for keyed samples.
23:0	Reserved[23:0]	R/W	0x000000	
Offset 0x300 - Video_Output_Format				

Table 318: Video Input Processor (VIP) Registers *…continued*

Bit	Symbol	Access	Value	Description
31:30	Psu_bamode[1:0]	R/W	0x0	Base address mode base 1-3 according to number of planes (plane 1-3) base 1-3, odd field (plane 1-3) base 4-6, even field (plane 1-3) packed modes only, frame index is set to 1 if cfen=0, frame index is incremented after capturing even field before capturing odd, base address byte offset is defined in PSU_OFFSET1 base 1, odd field 1st frame (plane 1 only) base 2, even field 1st frame (plane 1 only) base 3, odd field 2nd frame (plane 1 only) base 4, even field 2nd frame (plane 1 only) 0: single set (e.g. progressive video source) 1: reserved 10: alternate sets each field (e.g. interlaced video source) 11: alternate sets each field and frame (e.g. double buffer mode)
29:14	Reserved[15:0]	R/W	0x0000	
13	Psu_endian	R/W	0x0	Output format endianess 0: same as system endianess 1: opposite of system endianess
12	Reserved2	R/W	0x0	
11:10	Psu_dither[1:0]	R/W	0x0	Output format dither mode 00: no dithering 01: error dispersion (never reset pattern) 10: error dispersion (reset pattern at first capture enable) 11: error dispersion (reset pattern every field)
9:8	Psu_alpha[1:0]	R/W	0x0	Output format alpha mode setting 00 is ignored if size of alpha component is less than 8 bits 0: no alpha (alpha byte not written) 1: alpha byte written, value from CKEY_ALPHA (offset 284) 10: reserved 11: reserved
7:0	Psu_opfmt[7:0]	R/W	0x00	Output formats 08 (hex) = YUV 4:2:2, semi-planar 0B (hex) = YUV 4:2:2, planar 0F (hex) = RGB or YUV 4:4:4, planar A9 (hex) = compressed 4/4/4 + (4 bit alpha) AA (hex) = compressed 4/5/3 + (4 bit alpha) AD (hex) = compressed 5/6/5 A0 (hex) = packed YUY2 4:2:2 A1 (hex) = packed UYVY 4:2:2 E2 (hex) = YUV or RGB 4:4:4 + (8 bit alpha) E3 (hex) = VYU 4:4:4 + (8 bit alpha)
Offset 0x304 - Target_Window_Size				
31:28	Reserved[3:0]	R/W	0x0	
27:16	Psu_lsize[11:0]	R/W	0x000	Line size Used for horizontal cropping after scaling 0: cropping disabled 1: one pixel
15:11	Reserved2[4:0]	R/W	0x00	

Table 318: Video Input Processor (VIP) Registers *...continued*

Bit	Symbol	Access	Value	Description
10:0	Psu_lcount[10:0]	R/W	0x000	Line count Used for vertical cropping after scaling 0: cropping disabled 1: one line
Offset 0x340 - Target_Base_Address_1				
31:3	Psu_base1[28:0]	R/W	0x0000 0000	Base address DMA #1 used depending on PSU_BAMODE setting
2:0	Psu_offset1[2:0]	R/W	0x0	Base address byte offset plane 1 bits define pixel offset within multi pixel 64 bit words (e.g. a 16bit pixel can be placed on any 16 bit boundary)
Offset 0x344 - Target_Line_Pitch_1				
31:15	Unused[16:0]	R/W	0x0000 0	
14:3	Psu_pitch1[11:0]	R/W	0x000	Line pitch DMA #1, signed value (twos complement) used for all packed formats and for plane 1
2:0	Unused2[2:0]	R/W	0x0	
Offset 0x348 - Target_Base_Address_2				
31:3	Psu_base2[28:0]	R/W	0x0000 0000	Base address DMA #2 used depending on PSU_BAMODE setting
2:0	Psu_offset2[2:0]	R/W	0x0	Base address byte offset plane 2 bits define pixel offset within multi pixel 64 bit words (e.g. a 16bit pixel can be placed on any 16 bit boundary)
Offset 0x34C - Target_Line_Pitch_2				
31:15	Unused[16:0]	R/W	0x0000 0	
14:3	Psu_pitch2[11:0]	R/W	0x000	Line pitch DMA #2, signed value (twos complement) used for planes 2 and 3
2:0	Unused2[2:0]	R/W	0x0	
Offset 0x350 - Target_Base_Address_3				
31:3	Psu_base3[28:0]	R/W	0x0000 0000	Base address DMA #3 used depending on PSU_BAMODE setting
2:0	Psu_offset3[2:0]	R/W	0x0	Base address byte offset plane 3 bits define pixel offset within multi pixel 64 bit words (e.g. a 16bit pixel can be placed on any 16 bit boundary)
Offset 0x354 - Target_Base_Address_4				
31:3	Psu_base4[28:0]	R/W	0x0000 0000	Base address DMA #4 used depending on PSU_BAMODE setting
2:0	Unused[2:0]	R/W	0x0	
Offset 0x358 - Target_Base_Address_5				
31:3	Psu_base5[28:0]	R/W	0x0000 0000	Base address DMA #5 used depending on PSU_BAMODE setting
2:0	Unused[2:0]	R/W	0x0	
Offset 0x35C - Target_Base_Address_6				

Table 318: Video Input Processor (VIP) Registers *…continued*

Bit	Symbol	Access	Value	Description
31:3	Psu_base6[28:0]	R/W	0x0000 0000	Base address DMA #6 used depending on PSU_BAMODE setting
2:0	Unused[2:0]	R/W	0x0	
Offset 0x380 - Auxiliary_Capture_Output_Format				
31:30	Aux_bamode[1:0]	R/W	0x0	Base address mode 0: pitch mode, wrap at end of buffer or window 1: pitch mode, wrap at end of buffer 10: append mode, wrap at end of buffer or window 11: append mode, wrap at end of buffer
29:27	Reserved[2:0]	R/W	0x0	
26	Aux_sgnex	R/W	0x0	Auxiliary capture sign extension 0: no sign extension 1: sign extension enabled for 10 bit samples
25	Aux_bps	R/W	0x0	Auxiliary capture bits per sample 0: 8 bit samples 1: 10 bit samples
24	Aux_subsample	R/W	0x0	Auxiliary capture sub-sample Not available for ANC data capture 0: all samples 1: luma (even) samples only
23:22	Reserved2[1:0]	R/W	0x0	
21:0	Aux_bzsize[21:0]	R/W	0x0000 00	Auxiliary capture ring buffer size Size of ring buffer in bytes, 0 = unlimited buffer size
Offset 0x390 - Auxiliary_Capture_Base_Address				
31:0	Aux_base[31:0]	R/W	0x0000 0000	Auxiliary capture base address Lower 3 bits define byte offset within 64 bit words, offset has to be a multiple of the byte per unit size (e.g. a 16bit unit can be placed on any 16 bit boundary)
Offset 0x394 - Auxiliary_Capture_Line_Pitch				
31:15	Unused[16:0]	R/W	0x0000 0	
14:3	Aux_pitch[11:0]	R/W	0x000	Auxiliary capture line pitch Signed value
2:0	Unused2[2:0]	R/W	0x0	
Offset 0x800 - Horizontal_LS				
31:30	Unused[1:0]	R/W	0x0	
29:20	Tap_2_x[9:0]	W	0x000	Inverted coefficient, tap #2, twos complement
19:10	Tap_1_x[9:0]	W	0x000	Inverted coefficient, tap #1, twos complement
9:0	Tap_0_x[9:0]	W	0x000	Inverted coefficient, tap #0, twos complement
Offset 0x804 - Horizontal_MS				
31:30	Unused[1:0]	R/W	0x0	

Table 318: Video Input Processor (VIP) Registers *…continued*

Bit	Symbol	Access	Value	Description
29:20	Tap_5_x[9:0]	W	0x000	Inverted coefficient, tap #5, twos complement
19:10	Tap_4_x[9:0]	W	0x000	Inverted coefficient, tap #4, twos complement
9:0	Tap_3_x[9:0]	W	0x000	Inverted coefficient, tap #3, twos complement
Offset 0xFE0 - Interrupt_Status				
31	Stat_fid_aux	R	0x0	Field identifier at start of auxiliary window
30	Stat_fid_vid	R	0x0	Field identifier at start of video window
29	Stat_fid_vpi	R	0x0	Field identifier at video input port
28	Unused	R/W	0x0	
27:16	Stat_line_count[11:0]	R	0x800	Source video line count
15:10	Unused2[5:0]	R/W	0x00	
9	Stat_aux_ovrflw	R	0x0	Auxiliary buffer overflow event
8	Stat_vid_ovrflw	R	0x0	Video buffer overflow event
7	Stat_win_seqbrk	R	0x0	Windower sequence break event
6	Stat_fid_seqbrk	R	0x0	Field identifier sequence break event
5	Stat_line_thresh	R	0x0	Line counter threshold reached event
4	Stat_aux_wrap	R	0x0	Auxiliary capture write pointer wrap around event
3	Stat_aux_start_in	R	0x0	Start of auxiliary data acquisition event
2	Stat_aux_end_out	R	0x0	End of auxiliary data write to memory event
1	Stat_vid_start_in	R	0x0	Start of video data acquisition event
0	Stat_vid_end_out	R	0x0	End of video data write to memory event
Offset 0xFE4 - Interrupt_Enable				
31:10	Unused[21:0]	R/W	0x000000	
9	Ien_aux_ovrflw	R/W	0x0	Auxiliary buffer overflow event
8	Ien_vid_ovrflw	R/W	0x0	Video buffer overflow event
7	Ien_win_seqbrk	R/W	0x0	Windower sequence break event
6	Ien_fid_seqbrk	R/W	0x0	Field identifier sequence break event
5	Ien_line_thresh	R/W	0x0	Line counter threshold reached event
4	Ien_aux_wrap	R/W	0x0	Auxiliary capture write pointer wrap around event
3	Ien_aux_start_in	R/W	0x0	Start of auxiliary data acquisition event
2	Ien_aux_end_out	R/W	0x0	End of auxiliary data write to memory event
1	Ien_vid_start_in	R/W	0x0	Start of video data acquisition event
0	Ien_vid_end_out	R/W	0x0	End of video data write to memory event
Offset 0xFE8 - Interrupt_Clear				
31:10	Unused[21:0]	R/W	0x000000	
9	Clr_aux_ovrflw	W	0x0	Auxiliary buffer overflow event
8	Clr_vid_ovrflw	W	0x0	Video buffer overflow event
7	Clr_win_seqbrk	W	0x0	Windower sequence break event

Table 318: Video Input Processor (VIP) Registers *…continued*

Bit	Symbol	Access	Value	Description
6	Clr_fid_seqbrk	W	0x0	Field identifier sequence break event
5	Clr_line_thresh	W	0x0	Line counter threshold reached event
4	Clr_aux_wrap	W	0x0	Auxiliary capture write pointer wrap around event
3	Clr_aux_start_in	W	0x0	Start of auxiliary data acquisition event
2	Clr_aux_end_out	W	0x0	End of auxiliary data write to memory event
1	Clr_vid_start_in	W	0x0	Start of video data acquisition event
0	Clr_vid_end_out	W	0x0	End of video data write to memory event
Offset 0xFEC - Interrupt_Set				
31:10	Unused[21:0]	R/W	0x000000	
9	Set_aux_ovrflw	W	0x0	Auxiliary buffer overflow event
8	Set_vid_ovrflw	W	0x0	Video buffer overflow event
7	Set_win_seqbrk	W	0x0	Windower sequence break event
6	Set_fid_seqbrk	W	0x0	Field Identifier sequence break event
5	Set_line_thresh	W	0x0	Line counter threshold reached event
4	Set_aux_wrap	W	0x0	Auxiliary capture write pointer wrap around event
3	Set_aux_start_in	W	0x0	Start of auxiliary data acquisition event
2	Set_aux_end_out	W	0x0	End of auxiliary data write to memory event
1	Set_vid_start_in	W	0x0	Start of video data acquisition event
0	Set_vid_end_out	W	0x0	End of video data write to memory event
Offset 0xFFC - Module_ID				
31:16	Mod_id[15:0]	R	0x011A	Module ID Unique 16-bit code
15:12	Rev_major[3:0]	R	0x4	Major revision counter
11:8	Rev_minor[3:0]	R	0x0	Minor revision counter
7:0	App_size[7:0]	R	0x00	Aperture Size 0: 4 kB

5.10 Memory Based Scaler (MBS)

This section describes the MBS of the PNX2015.

Other than real-time scaling, where a scaler is locked to either the video source or the outgoing video stream, memory-based scaling is achieved independently of any video clocks by reading the video data from memory and writing the scaled pictures back to the memory. An advantage of memory-based scaling is that a single scaler can be used to scale more than one video stream.

5.10.1 Features

The memory-based scaler in the PNX2015 (based on the PNX8525 design but significantly restructured) can perform the following operations:

- Vertical and horizontal scaling

Linear and non-linear aspect-ratio conversion

- De-interlacing

  Simple median.

  Majority-selection (median filtering with previous field, spatial temporal average of 2 fields, same position from next or previous field depending on whether three or two field majority selection) [Note: majority selection is done on luma only].

  Field insertion[8] and line doubling (i.e., repeating the same line twice) are attainable under software control.

  Edge-Dependent De-interlacing (EDDI) --- a post-processing step done only on luma.
- Anti-flicker filtering
- Conversions between 4:2:0, 4:2:2 and 4:4:4
- Indexed to true color conversion
- Color expansion/compression (different quantizations for color components, e.g., RGB565 -> RGB32)
- Deplanarization/planarization
- Variable color space conversion with programmable matrix coefficients (mutually exclusive with horizontal scaling)
- Color-key and alpha processing

  Conversion between color-key and alpha.

  Alpha scaling
- Measurements

  Histogram measurement

  Noise estimation

  Blackbar detection

  Blacklevel measurement

  UV bandwidth measurement
- Task-list based programming

Most of the above functions can be performed during a single pass, though the filter quality (length) may vary depending on the performed operations.

Special mode and features:

- Color key to alpha and alpha to color key conversion (color re-keying)
- Non-linear phase interpolation (phase LUT)

[8] The current MBS implementation can not perform film-mode detection.

5.10.2 Functional description

5.10.2.1 MBS block diagram

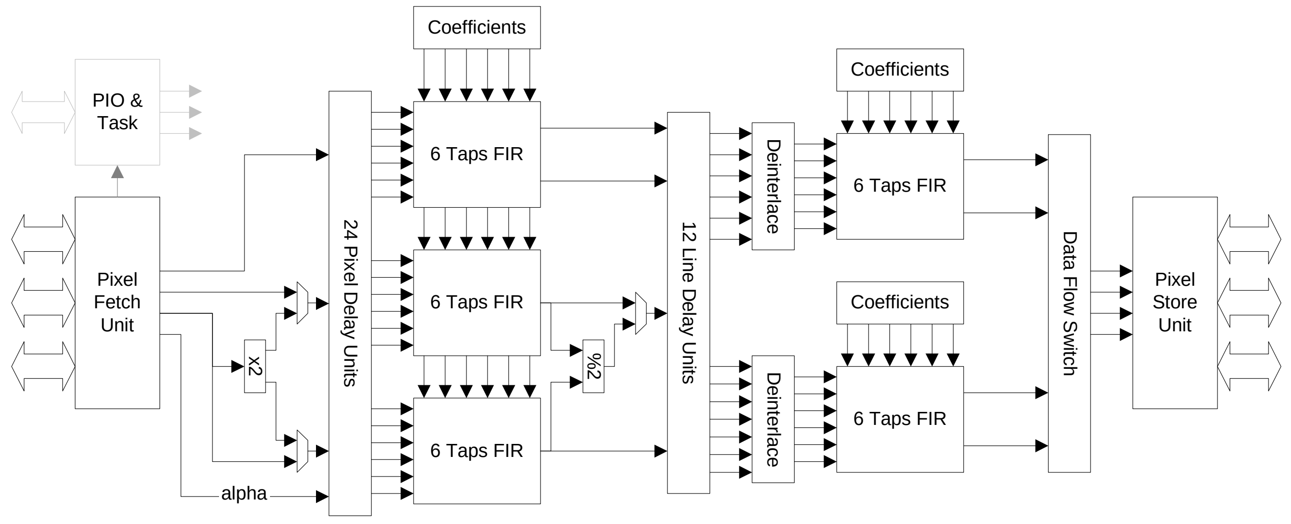


Fig 230.MBS Block Diagram

5.10.3 General Operations

5.10.3.1 Data flow and control

This section provides an overview of the processing functions in the MBS.

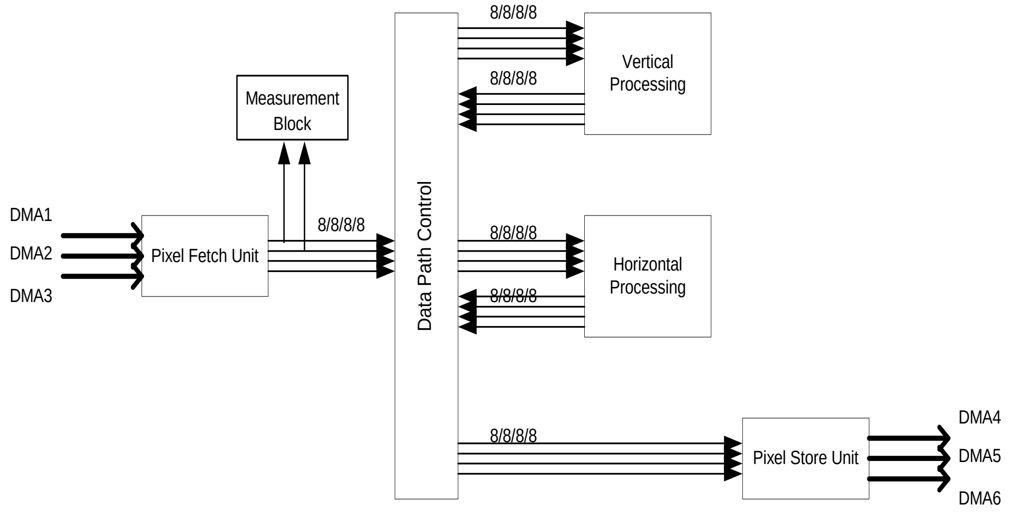


Fig 231.MBS Top Level

Figure 231 shows the top-level structure of MBS. There are two independent processing pipelines – Vertical processing Pipeline and Horizontal Processing Pipeline – that, for most applications, can be used in any order; for 3-field majority selection, however, vertical processing has to be done first.

Horizontal Processing Pipeline: Figure 232 elaborates on the Horizontal Processing Pipeline (HPP). HPP is responsible for chroma up-sampling, horizontal scaling, color-space conversion, and chroma down-sampling. Note that only one of either horizontal scaling or color-space conversion can be done at any one time., i.e., these operations are mutually exclusive. The scaler coefficients are the same for both chroma and luma.

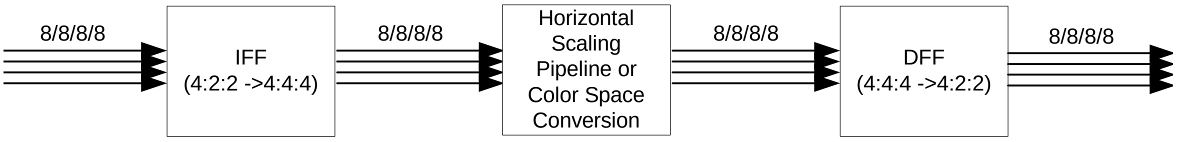


Fig 232.MBS Horizontal Processing Pipeline

Vertical Processing Pipeline

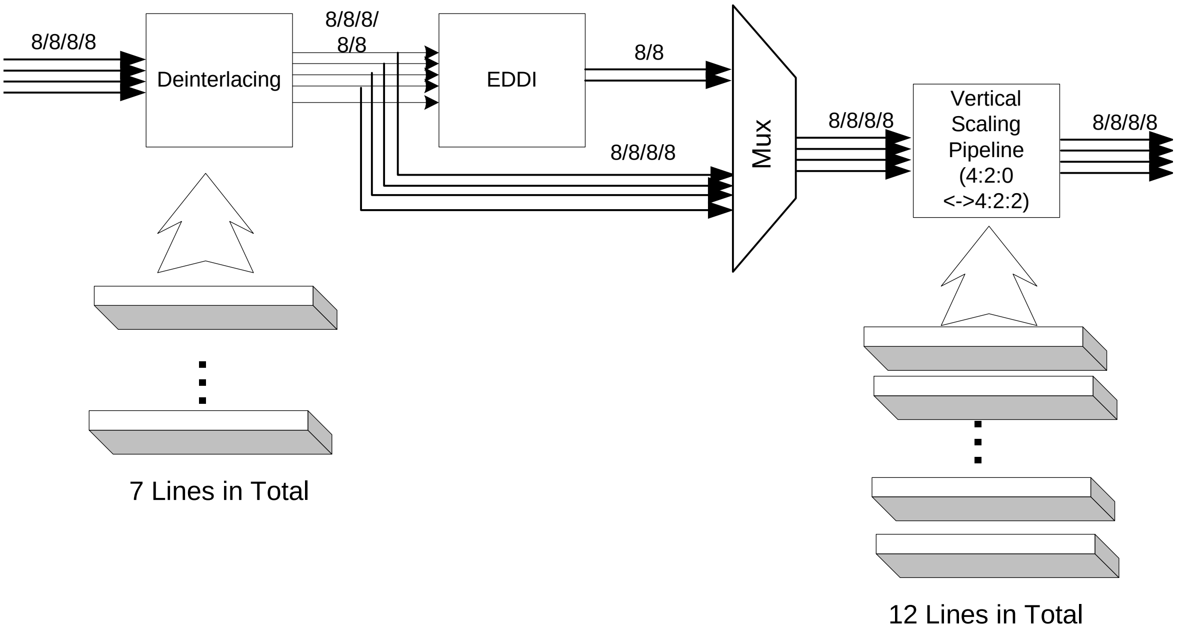


Fig 233.MBS Vertical Processing Pipeline

Figure 233 shows the Vertical Processing Pipeline (VPP). VPP comprises de-interlacing, EDDI-based post-processing, and vertical scaling (that also allows 4:2:0 <-> 4:2:2 sampling conversions for chroma). De-interlacing for luma samples uses majority-select or median filtering; de-interlacing for chroma always uses median filtering. For vertical scaling, there are 2 separate coefficient tables: one for luma and the other for chroma, thereby allowing the use of different filter coefficients for chroma and luma.

5.10.4 Data processing in MBS

Table 319, Table 320 and Table 321 tabulate the various processing modes and orders in the MBS.

Table 319: Pipeline Processing (Horizontal First Mode)

Input	IFF[1]	Horizontal	DFF[2]	Vertical	Output
- 4:2:0 input (semi) planar - 4:2:2 input	up-sample to 4:4:4	- color space conversion (full matrix plus offset) or - scaling only (6 tap direct or transposed polyphase)	down-sample to 4:2:x	-de-interlacing[3] and/or - scaling (6 tap direct polyphase)	- 4:2:0 output (semi) planar - 4:2:2 output (packed, semi-planar, planar)
- 4:4:4 input (8-32 BPP) - LUT mode (8/4/2/1 BPP)[4]	bypass		bypass	- scaling only (4 tap direct polyphase) and/or - Anti Flicker (static 4 tap filter)	- 4:4:4 output (16-32 BPP) - 4:4:4:4 output (16-32 BPP)
- 4:4:4:4 input (16-32 BPP) - LUT mode (8/4/2/1 BPP)[5]	bypass	- scaling only (3 tap direct polyphase)	bypass	- scaling only (3 tap direct polyphase) and/or - Anti Flicker (static 3 tap filter)	- 4:4:4 output (16-32 BPP) - 4:4:4:4 output (16-32 BPP)

[1] Interpolation FIR Filter

[2] Decimation FIR Filter

[3] VTM (luma and chroma) or 2 field MSA (luma) with or without EDDI

[4] Alpha component from LUT is not used

[5] Alpha component from LUT is used

Table 320: Pipeline Processing (Vertical First Mode)

Input	Vertical	IFF	Horizontal	DFF	Output
- 4:2:0 input - 4:2:2 input	- de-interlacing (see Table 321) and/or - scaling (6 tap direct polyphase)	up-sample to 4:4:4	- color space conversion (full matrix plus offset) or - scaling (6 tap direct or transposed polyphase)	down-sample to 4:2:x	- 4:2:0 output (2-3 planes) - 4:2:2 output (packed, semi-planar, planar)
- 4:4:4 input (8-32 BPP) - LUT mode (8/4/2/1 BPP)	- scaling (4 tap direct polyphase) and/or - Anti Flicker (static 4 tap filter)	bypass		bypass	- 4:4:4 output (16-32 BPP) - 4:4:4:4 output (16-32 BPP)
- 4:4:4:4 input (16-32 BPP) - LUT mode (8/4/2/1 BPP)	- scaling (3 tap direct polyphase) and/or - Anti Flicker (static 3 tap filter)	bypass	- scaling (3 tap direct polyphase)	bypass	- 4:4:4 output (16-32 BPP) - 4:4:4:4 output (16-32 BPP)

Table 321: De-Interlacing Mode Maximum Filter Lengths

Input Format	EDDI	MSA 2 Field (Y:UV Taps)	MSA 3 Field (Y:UV Taps)	Median (Y:UV Taps)
4:2:0 or 4:2:2 planar	Yes	6:6	Not supported	6:6
4:2:0 or 4:2:2 semi-planar	Yes	6:6	6:6[1]	6:6
4:2:2 single plane	No	Not supported	Not supported	6:6

[1] Only supported in Vertical-First mode

5.10.5 MBS function

This section provides the details on how the MBS functions. A description of each functional group is provided.

5.10.5.1 Task control

Since the MBS is capable of processing several video streams in sequence, a pipelining mechanism is used for scaling a sequence of tasks. A task is described by a data structure stored in memory. Writing the base address of this task into the Task FIFO schedules the task for execution on completion of the current task.

In addition to the task list in the FIFO, each Task structure in memory can consist of a linked list of sub tasks that are executed in sequence (e.g., HD scaling task via partitioning). The software scheduling algorithm is responsible for preventing the Task FIFO from overflowing. An interrupt can be generated, once the last task in the FIFO gets executed, in order to request new tasks from the scheduler. Other interrupt events help in keeping the task FIFO filled and avoiding overflow in the four available FIFO slots.

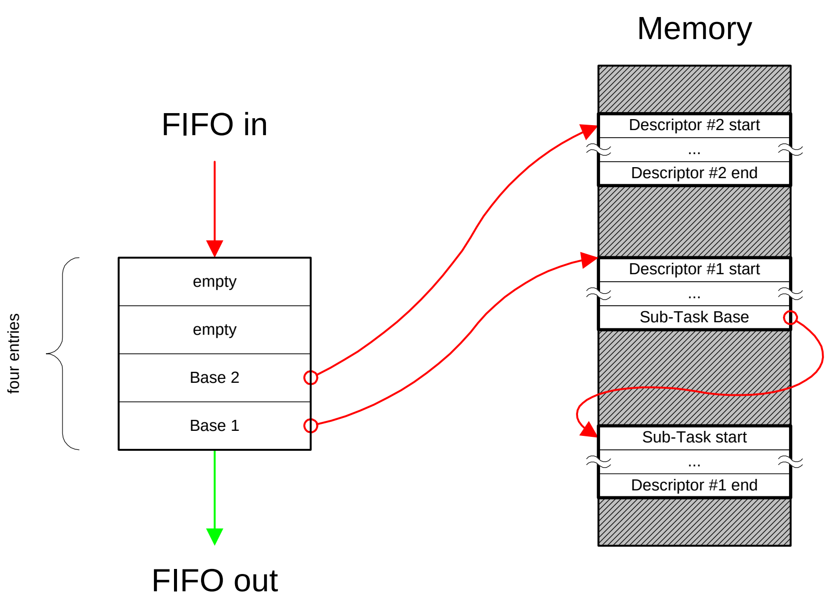


Fig 234.Task FIFO and Linked List

Table 322 shows the opcodes allowed in a descriptor list. The data structure consists of 32 bit words; therefore, endianness rules do apply. The command type is defined in the lower two bits of the 32-bit word. The remaining bits are decoded depending on the command type.

Table 322: Task Descriptor Opcode Table

Command	Bits	Function	Description
Jump	Link to address		
	25:3	address	Defines location where processing of commands continues.
	1:0	opcode	= 00 (binary)
Exec/Stop	End of task list		
	2	stop flag	If this flag is set, processing of the MBS task is stopped.
	1:0	opcode	= 01 (binary)
Queue	Start processing and queue next task		
	25:3	address	Task at this location is started after processing of current task finished.
	1:0	opcode	= 10 (binary)
Load	Load register		
	31:24	count	Define number of registers to be loaded minus one.
	19:16	mask	Write mask, if bit is zero according byte is written, if bit is set byte is not written. Bit 16 is the least significant Byte Mask, Bit 16 is the most significant Byte Mask.
	11:2	Index	Load registers starting at offset.
	1:0	opcode	= 11 (binary)

5.10.5.2 Video source controls

Source video data for the MBS can be fetched via three dedicated DMA engines. It can be fetched from memory in several packed or planar formats. The source window is defined by one set of width and height values which are automatically translated into the corresponding number of 64-bit words to be fetched per line for each plane. Video lines can be fetched in a reverse order as well, thereby allowing horizontal flipping of images for applications like video conferencing.

To allow fetching of interlaced video lines from different locations in memory, each plane can be assigned a second set of base address registers. Pitches are defined separately only for luma and chroma planes. The lower three bits of the first three base address registers are used as an intra-long-word offset for the left-most pixel components of each line. The offset has to be a multiple of the number of bytes per component.

5.10.5.3 Fixed input formats

The fixed input formats, as shown in Table 323, consist of indexed, packed and planar modes.

Table 323: Input Pixel Formats

Format	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1 bit indexed																									I							
2 bit indexed																									2							
4 bit indexed																									4 bit							
8 bit indexed																									8 bit Index							
YUV 4:4:4, planar YUV 4:2:x, planar RGB 4:4:4, planar	plane #1																								Y8 or R8							
	plane #2																								U8 or G8							
	plane #3																								V8 or B8							
YUV 4:2:x, semi planar	plane #1																								Y8							
	plane #2																V8								U8							
packed YUY2 4:2:2																	U8 or V8								Y8							
packed UYVY 4:2:2																	Y8								U8 or V8							
packed variable 16 bit																	variable YUV or RGB 4:4:4															
packed variable 32 bit	variable YUV 4:4:4 , 4:2:2 or RGB 4:4:4																															

For indexed formats, sizes of 1, 2, 4, or 8 bits per pixel can be selected. A 32-bit color look up table is used to expand indexed modes into true color, including alpha values.

Predefined packed modes are available for YUY2 and UYVY formats. Other packed modes can be selected by using one of the three variable input formats described below.

Planar formats can be used to fetch pixel components from different locations in memory. In semi-planar modes, a video field is defined by one plane for Y samples and one shared plane for U/V. In full planar formats, each component is fetched from a separate plane.

When processing a 4:2:2 or 4:2:0 input stream, the first sample fetched from memory has to be a Y/U pair.

5.10.5.4 Variable input formats

In addition to the predefined fixed input pixel formats above, the MBS also includes three variable input format modes for packed 4:4:4 and 4:2:2 pixel extraction. In these modes, the pixel components can be extracted from programmable locations within 32- or 16-bit units.

5.10.5.5 Horizontal video filters

Interpolation Filter - All horizontal video processing is based on equidistant sampled components. All 4:2:2 video streams, therefore, have to be up-sampled before being scaled horizontally. The interpolation FIR filter used can interpolate interspersed or co-sited chroma samples. Mirroring of samples at the field boundaries compensate for run in and run out conditions of the filter.

Decimation Filter - After horizontal processing, chrominance may be down sampled to reduce memory bandwidth or allow for higher quality vertical processing that is not available otherwise. Mirroring of samples at the field boundaries compensate for run in and run out conditions of the filter.

Normal Polyphase Filter Mode - The normal polyphase filter can be used to zoom out or downscale a video image. Depending on the number of components, the filter uses 6 taps (three-component mode) or 3 taps (four-component mode). For the normal polyphase filter, run-in and run-out behavior are compensated by appropriate mirroring of samples at the field boundaries. In order to align the first output pixel with the geometric location of the first input pixel, the start phase has to be chosen as:

dto_offset = 0x200 . filter_length

The scaling ratio for the transposed polyphase *HSP_ZOOM_0 = 0x1 0000h * zoom_x, with zoom_x = pixel_in / pixel_out*

Transposed Polyphase Filter Mode - The transposed polyphase filter can only be used to downscale. By reversing the flow through the normal polyphase filter, the transposed filter takes the output pixels as reference and accumulates input pixels. The advantage of the transposed polyphase filter is the longer effective filter length which allows for filtering out high frequencies that interfere with downscaling. A drawback of this approach is that the filter coefficients have to get optimized for the scaling ratio in order to avoid visible DC ripples. Compensation for run-in and run-out behavior of the filter is not available. Use of the transposed filter is limited to three components only.

The scaling ratio for the transposed polyphase *HSP_ZOOM_0 = 0x1 0000h * zoom_x, with zoom_x = pixel_out / pixel_in*

Linear Phase Interpolation Mode (LPI) - In the linear phase interpolation mode, samples between two pixels are interpolated by using the linear equation px = (1-phase) * xleft + phase * xright. Linear phase interpolation creates adequate results especially when used with computer generated graphics which, unlike motion video, is usually not bandwidth limited.

The LPI mode is enabled by setting HSP_PHASE_MODE = 7. Horizontal processing mode has to be set to normal polyphase (HSP_MODE = 2) and four-component mode has to be enabled (HSP_FIR_COMP = 1).

Color Space Matrix Mode - In addition to the normal and transposed polyphase filtering (scaling), the FIR filter structure can instead be programed to perform color-space-conversion functions. A dedicated set of registers holds the coefficients for the color space matrix.

5.10.5.6 Vertical video filters

The polyphase filter used for vertical processing can only be operated in the normal polyphase mode.This filter can be used as a two-component 6-tap filter, a three-component 4-tap filter or a four-component 3-tap filter (optionally, linear phase interpolation). In the two-component mode, each filter unit can be run independently, to allow 4:2:0 to 4:2:2 conversion or vice versa.

5.10.5.7 De-interlacing in MBS

The current de-interlacing algorithms, implemented inside the MBS, are as follows:

- Simple median filtering (Vertical Temporal Median -- VTM)
- Majority Selection Algorithm (MSA)
- Field insertion and line doubling

MBS supports both field insertion and line doubling based simple de-interlacing in the pixel-fetch unit.

If de-interlacing is enabled, the previous and current fields have to be fed into the scaler as one interleaved frame. For the MSA, the first line has to originate in the previous field, while in VTM, the first line has to be fetched from the current field. Base Address Mode = 1 can be selected to assist in weaving the two fields together into one frame

5.10.5.8 Edge Dependent De-Interlacing (EDDI)

The purpose of EDDI is to improve the appearance of long diagonal edges, without degrading other parts of the image. EDDI is used as a post-processing unit for the de-interlacer.

5.10.5.9 Color Key Processing

Color key processing, to convert alpha values into color key values and/or vice versa, are only available for the 4:4:4 video formats.

Color Key to Alpha Conversion

To avoid color key values getting altered during video processing, color key matching samples can be tagged as transparent by generating an alpha value of zero at the input of the MBS. Non-keyed samples get assigned a programmable alpha value. After video processing is over, the alpha value can be stored in memory together with the other components or can be used to re-key the keying color back into the video stream (see below).

Color Key Replace

Scaling of video streams containing color keys can lead to artifacts at the corners between keyed and non-keyed samples, due to the “smearing effect of the low pass filter. The effect is heightened by the highly saturated colors normally used for keying. To avoid these artifacts, color keyed samples are replaced with either gray, black, or the previous sample (gray if at the start of the line),.

Alpha to Color Key Convert

If the alpha value, after horizontal and vertical processing, is below a fixed threshold (0x80), the sample is replaced by the color key.

Fixed Alpha Insert

The alpha value defined in the Color Key register is inserted, as the fourth component after horizontal and vertical processing, in the PSU unit (if CKEY_K2A=1).

5.10.5.10 Alpha Processing

Alpha processing is only available when the horizontal and vertical filter blocks are either bypassed or operated in the four-component mode. If the horizontal filter is used for color space conversion, the alpha information is kept time-aligned with the other three components.

5.10.5.11 Video Data Output

After processing the video stream, the MBS can split the video data into one or more (up to three) different streams. For each stream, there is a memory base address. There are two line-pitch registers further defining the DMA streams. The number of streams (planes) is defined by the output-format register. Depending on its value, the video components get packed into 64-bit words. These words then get buffered and transferred to the external memory in more effective clusters. A list of the supported output video formats is shown in Table 324. Packing of a pixel into 64-bit units is always done from right to left, while bytes within one pixel unit are ordered according to the Endian settings (according to the global Endian setting, the Endian bit in the output format register can invert the setting).

Table 324: Output Pixel Formats

Format	3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1
	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0									
planar YUV or RGB (4:4:4, 4:2:2 or 4:2:0)	plane #1																								Y8 or R8						
	plane #2																								U8 or G8						
	plane #3																								V8 or B8						
semi planar YUV (4:2:2 or 4:2:0)	plane #1																								Y8						
	plane #2																V8								U8						
packed 4/4/4 RGBa																	alpha				R4				G4				B4		
packed 4/5/3 RGBa																	alpha				R4				G5					B3	
packed 5/6/5 RGB																	R5					G6						B5			

Table 324: Output Pixel Formats *…continued*

Format	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
packed YUY2 4:2:2																	U8 or V8								Y8						
packed UYVY 4:2:2																	Y8								U8 or V8						
packed 888 RGBa	alpha								R8 or Y8								G8 or U8								B8 or V8						
packed 4:4:4 VYUa	alpha								V8								Y8								U8						

Remark: Table 324 shows location of the first ’pixel unit’ within a 64 bit word in the little endian mode. The selected endianness affects the position of the components within multi-byte pixel units.

5.10.5.12 Address Generation

Each of the three video planes is assigned a set of two base address registers and two line address pointers. Depending on the base address mode, video data corresponding to each plane is written using one pointer, or using both pointers alternating on each line. At the end of the line, the pitch value is added to the active line address pointer. The pitch defines the difference in the address of two vertically adjacent pixels. The pitch values are defined separately for chroma and luma components only. The lower three bits of the first three base address registers are used as an intra-long-word offset for the left-most pixel components of each line. The offset has to be a multiple of the number of bytes per component.

5.10.5.13 Interrupt Generation

The MBS contains a DVP-compliant interrupt generation mechanism. The following interrupt events are defined:

TASK_ERROR

Current processing task leads to a pipeline error, MBS has to be reset to resume with new scaling task(s).

TASK_END

Current task processing is done, but some data might still remain in the output FIFO. This Interrupt denotes the point when the MBS is ready to start the processing of a new task.

TASK_OVERFLOW

Task FIFO overflow - task request written into Task FIFO register got ignored.

TASK_IDLE

Task finished and the Task FIFO is empty.

TASK_EMPTY

Task FIFO runs empty - generation of this interrupt requires that there was a pending task in the Task FIFO. This interrupt is not be generated if tasks are only submitted in the MBS idle state.

TASK_DONE

Current task finished and all writes complete - on reception of this interrupt, all data is accessible in the main memory.

5.10.5.14 Measurement Functions

All measurements in the MBS are based on analyzing a video stream, within a programmable window, using specific algorithms. The measurement results are stored in registers once per field/frame. A special interrupt is generated to indicate that the measurement is done. Based on the measurement results, software sets the control parameters for the picture processing units.

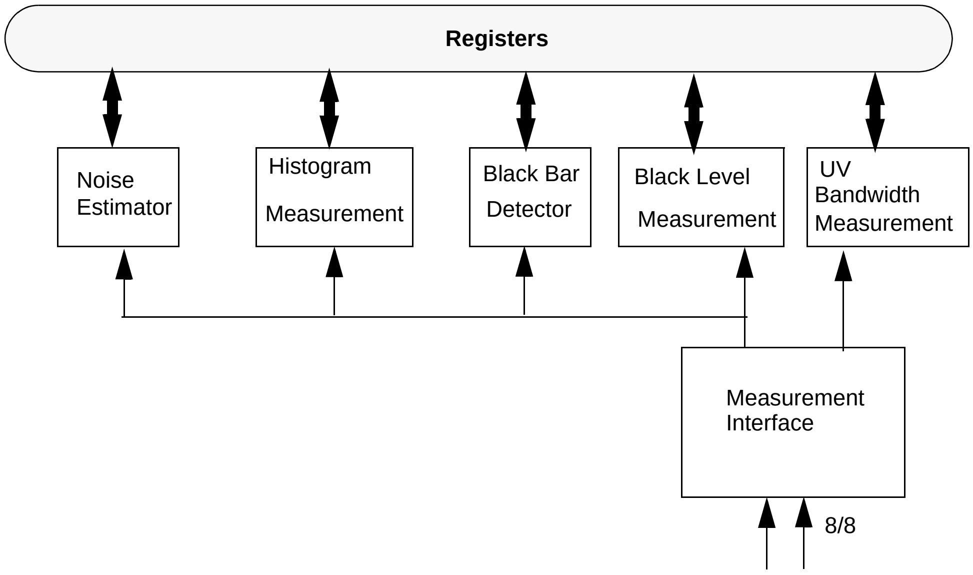


Fig 235.Measurement in the MBS

As shown in Figure 235, the MBS measurement block comprises:

- Measurement Block Interface

  Converts the inputs format (Y, U/V 8bits) into the format needed by the measurement units (Y, U/V 9bits).
- Histogram Measurement

  Analyzes the Y component of the input data stream. Used for dynamic contrast improvement.
- Noise Estimator

  Analyzes the Y component and the result can be used to control a Temporal Noise Reducer.

- Black Bar Detector

  Analyzes the Y component. Results are used to expand the picture to the display size in case of black bars at the top and/or bottom of the picture.
- Black Level Measurement

  Analyzes the Y component. Used for black stretch.
- UV Bandwidth Measurement

  Analyzes the UV component and the result can be used to control a Chrominance Transient Improver.

Remark: MBS and QTNR both use the same measurement functions/block.

5.10.6 Register descriptions

5.10.6.1 Register summary

Table 325: Register Summary

Offset	Name	Description
0x0	MBS_Mode_Control	MBS operation mode
0x10	MBS_FEATURE_REGISTER	
0x40	Task_FIFO	Task queue FIFO
0x44	Task_Status_1	Task queue status
0x48	Task_Status_2	Shows last command executed
0x100	Input_Format	Input format and mode
0x104	Source_Window_Size	Source window size
0x108	Variable_Format_Register	Variable source format
0x140	Source_Base_Address_1	Source base address DMA #1
0x144	Source_Line_Pitch_1	Source line pitch component 1
0x148	Source_Base_Address_2	Source base address DMA #2
0x14C	Source_Line_Pitch_2	Source line pitch component 2 and 3
0x150	Source_Base_Address_3	Source base address DMA #3
0x154	Source_Base_Address_4	Source base address DMA #4
0x158	Source_Base_Address_5	Source base address DMA #5
0x15C	Source_Base_Address_6	Source base address DMA #6
0x200	HSP_Zoom_0	Initial zoom for 1st pixel in line (unsigned)
0x204	HSP_Phase	Horizontal phase control
0x208	HSP_DZoom_0	Initial zoom delta for 1 pixel in line (signed)
0x20C	HSP_DDZoom	Zoom delta change (signed)
0x220	Color_space_matrix_coefficients_C00_C02	Color space matrix coefficients C00 - C02
0x224	Color_space_matrix_coefficients_C10_C12	Color space matrix coefficients C10 - C12
0x228	Color_space_matrix_coefficients_C20_C22	Color space matrix coefficients C20 - C22
0x22C	Color_space_matrix_offset_coefficients_D0_D2	Color space matrix offset coefficients D0-D2
0x230	Color_space_matrix_offset_coefficients_E0_E2	Color space matrix rounding coefficients E0-E2
0x240	VSP_Zoom_0	Initial zoom for 1st pixel in line (unsigned)
0x244	VSP_Phase	Vertical phase control
0x248	VSP_DZoom_0	Initial zoom delta for 1 pixel in line (signed)

Table 325: Register Summary *…continued*

Offset	Name	Description
0x24C	VSP_DDZoom	Zoom delta change (signed)
0x280	Color_Key_Control	Color key and alpha control
0x284	Color_Key_Components	
0x300	Video_Output_Format	
0x304	Target_Window_Size	Target window size
0x340	Target_Base_Address_1	Target base address DMA #1
0x344	Target_Line_Pitch_1	Target line pitch component 1
0x348	Target_Base_Address_2	Target base address DMA #2
0x34C	Target_Line_Pitch_2	Target line pitch component 2
0x350	Target_Base_Address_3	Target base address DMA #3
0x354	Target_Base_Address_4	Target base address DMA #4
0x358	Target_Base_Address_5	Target base address DMA #5
0x35C	Target_Base_Address_6	Target base address DMA #6
0x400	Color_Look_Up_Table	Color look-up table
0x800	Horizontal_LS	Coefficient Table #1 Taps 0-2 (Horizontal)
0x804	Horizontal_MS	Coefficient Table #1 Taps 3-5 (Horizontal)
0xA00	Vertical_Luma_LS	Coefficient Table #2 Taps 0-2 (Vertical - Luma)
0xA04	Vertical_Luma_MS	Coefficient Table #2 Taps 3-5 (Vertical - Luma)
0xC00	Vertical_Chroma_LS	Coefficient Table #3 Taps 0-2 (Vertical - Chroma)
0xC04	Vertical_Chroma_MS	Coefficient Table #3 Taps 3-5 (Vertical - Chroma)
0xE00	Formatter_Control	Formatter control
0xE0C	Flaggen_Control_Registers	
0xE10	Histogram_Control_Register	Histogram control
0xE18	Histogram_Window_Start	Histogram window start
0xE1C	Histogram_Window_End	Histogram window end
0xE20	Noise_Estimator_Control_Register_1	Noise estimation control 1
0xE24	Noise_Estimator_Control_Register_2	Noise estimation control 2
0xE28	Noise_Estimator_Window_Start	Noise estimation window start
0xE2C	Noise_Estimator_Window_End	Noise estimation window end
0xE30	Black_Bar_Detector_Control_Register	Black bar detection control
0xE38	Black_Bar_Detection_Window_Start	Black bar detection window start
0xE3C	Black_Bar_Detection_Window_End	Black bar detection window end
0xE40	Black_Level_Detection_Control	Black level detection control
0xE48	Black_Level_Detection_Window_Start	Black level window start
0xE4C	Black_Level_Detection_Window_End	Black level window end
0xE50	Bandwidth_Detection_Control	UV Bandwidth detection control
0xE58	Bandwidth_Detection_Window_Start	UV Bandwidth window start
0xE5C	Bandwidth_Detection_Window_End	UV Bandwidth window end
0xE8C	Histogram_Data_Output_1	Histogram status
0xE90	Histogram_Data_Output_2	Histogram data output
0xE94	Histogram_Data_Output_3	Histogram data output

Table 325: Register Summary *…continued*

Offset	Name	Description
0xE98	Histogram_Data_Output_4	Histogram data output
0xE9C	Histogram_Data_Output_5	Histogram data output
0xEA0	Histogram_Data_Output_6	Histogram data output
0xEA4	Histogram_Data_Output_7	Histogram data output
0xEA8	Histogram_Data_Output_8	Histogram data output
0xEAC	Histogram_Data_Output_9	Histogram data output
0xEB0	Noise_Estimator_Data_Output_1	Noise estimation status 1
0xEB4	Noise_Estimator_Data_Output_2	Noise estimation status 2
0xEB8	Black_Bar_Detection_Data_Output_1	Black bar detection status 1
0xEBC	Black_Bar_Detection_Data_output_2	Black bar detection status 2
0xEC0	Black_Level_Detection_Control_Output	Black level detection status
0xEC4	Bandwidth_Detection_Output_1	UV Bandwidth detection status 1
0xEC8	Bandwidth_Detection_Data_Output_2	UV Bandwidth detection status 2
0xFE0	Interrupt_Status	Interrupt status register
0xFE4	Interrupt_Enable	Interrupt enable register
0xFE8	Interrupt_Clear	Interrupt clear register
0xFEC	Interrupt_Set	Interrupt set register
0xFFC	Module_ID	Module Identification and revision information

5.10.6.2 Register table

Table 326: Memory Based Scaler (MBS) Registers

Bit	Symbol	Access	Value	Description
Offset 0x0 - MBS_Mode_Control				
31:30	Dpm_seq[1:0]	R/W	0x0	Data path sequence 0x: bypass all filter stages (format conversion only) 10: HSP -> VSP 11: VSP -> HSP
29:22	Reserved[7:0]	R/W	0x00	
21:20	Coeff_lut_mode[1:0]	R/W	0x0	Coefficient look-up table access mode 00: each table gets written separately 01: writes to coeff table 2 are copied into table 3 10: writes to coeff table 1 are copied into table 2 11: writes to coeff table 1 are copied into table 2 and 3
19	Disable_irq	R/W	0x0	1: disables the MBS interrupts- task done and task end. Used for concatenated tasks.
18	No_auto_skip	R/W	0x0	Disable Auto Skip When set to zero (default) the MBS will automatically skip all remaining operations within a task once the TASK_DONE interrupt was generated. If enable the MBS will continue until all pipeline stages are idle.
17	Skip_task	W	0x0	Skip current task Writing a one into this bit will reset the current task and continue with previously scheduled tasks

Table 326: Memory Based Scaler (MBS) Registers *...continued*

Bit	Symbol	Access	Value	Description
16	Soft_reset	W	0x0	Soft reset Writing a one into this bit will reset the block and all outstanding scaling tasks
15	Reserved2	R/W	0x0	
14	Iff_clamp	R/W	0x0	Clamp mode for IFF (affects U/V only) 0: clamp to 0-255 1: clamp to 16 - 240 (CCIR range)
13:12	Iff_mode[1:0]	R/W	0x0	Interpolation mode 00: bypass 01: reserved 10: co-sited 11: interspersed
11	Reserved3	R/W	0x0	
10	Dff_clamp	R/W	0x0	Clamp mode for DFF (affects U/V only) 0: clamp to 0-255 1: clamp to 16 - 240 (CCIR range)
9:8	Dff_mode[1:0]	R/W	0x0	Decimation mode 00: bypass 01: co-sited (sub sample) 10: co-sited (low pass) 11: interspersed
7	Vsp_clamp	R/W	0x0	Clamp mode for VSP 0: clamp to 0-255 1: clamp to CCIR range defined by VSP_RGB (bit 6)
6	Vsp_rgb	R/W	0x0	Color space mode, defines CCIR clamping range for VSP 0: processing in YUV color space (CCIR range: 16 - 235(Y), 16 - 240(U/V)) 1: processing in RGB color space (CCIR range: 16 - 235)
5:4	Vsp_mode[1:0]	R/W	0x0	Vertical processing mode 00: bypass mode 01: reserved 10: normal polyphase mode 11: reserved
3	Hsp_clamp	R/W	0x0	Clamp mode for HSP 0: clamp to 0-255 1: clamp to CCIR range defined by HSP_RGB (bit 2)
2	Hsp_rgb	R/W	0x0	Color space mode, defines CCIR clamping range for HSP 0: processing in YUV color space (CCIR range: 16 - 235(Y), 16 - 240(U/V)) 1: processing in RGB color space (CCIR range: 16 - 235)

Table 326: Memory Based Scaler (MBS) Registers *…continued*

Bit	Symbol	Access	Value	Description
1:0	Hsp_mode[1:0]	R/W	0x0	Horizontal processing mode 00: bypass mode 01: color space matrix mode 10: normal polyphase mode 11: transposed polyphase mode
Offset 0x10 - MBS_FEATURE_REGISTER				
31:21	Reserved[10:0]	R/W	0x000	
20	Mea	R	0x1	Measurement Block
19	Eddi	R	0x0	EDDI
18	Deinterlace	R	0x0	Deinterlace
17	Vs_aff	R	0x1	Veridical Scaler/ Anti flicker filtering
16	Hs_csm	R	0x1	0 - Horizontal Scaler/CSM not present 1 - Horizontal Scaler/CSM present
15:7	Reserved2[8:0]	R/W	0x000	
6:4	Max_line_len[2:0]	R	0x7	Indicates maximum pixels per line supported 7 - HD line (up to 2047 pixels) 6 - HD line (up to 1280 pixels) 5 - SD line (up to 780 pixels) 4 - up to 512 pixels 0,1,2,3 - Reserved
3:0	Reserved3[3:0]	R/W	0x0	
Offset 0x40 - Task_FIFO				
31:3	Task_base[28:0]	W	0x00000000	Scale task FIFO Must be aligned to 8 byte boundary
2	Reserved	R/W	0x0	
1:0	Task_cmd[1:0]	W	0x0	Command mode 00: process task descriptor at given base 01: start scaling with current register settings 1x: reserved
Offset 0x44 - Task_Status_1				
31:4	Reserved[27:0]	R/W	0x0000000	
3:0	Task_pending[3:0]	R	0x0	Number of pending scaling tasks (including current)
Offset 0x48 - Task_Status_2				
31:0	Last_command[31:0]	R	0x00000000	Last Command executed
Offset 0x100 - Input_Format				
31:30	Pfu_bamode[1:0]	R/W	0x0	Base address mode base 1-3 according to number of planes (plane 1-3) base 1-3, first line in, etc. (plane 1-3) base 4-6, second line in, etc. (plane 1-3) 1x = reserved 0: single set (e.g. progressive video source) 1: alternate sets each line (e.g. interlaced video source)
29:16	Reserved[13:0]	R/W	0x0000	
15	Pfu_hflip	R/W	0x0	Mirror input mode 0: normal 1: mirrored input
14	Reserved2	R/W	0x0	
13	Pfu_endian	R/W	0x0	Input format Endianness 0: same as system Endianness 1: opposite of system Endianness

Table 326: Memory Based Scaler (MBS) Registers *…continued*

Bit	Symbol	Access	Value	Description
12:8	Reserved3[4:0]	R/W	0x00	
7:0	Pfu_ipfmt[7:0]	R/W	0x00	Input formats 00 (hex) = YUV 4:2:0, semi-planar 03 (hex) = YUV 4:2:0, planar 08 (hex) = YUV 4:2:2, semi-planar 0B (hex) = YUV 4:2:2, planar 0F (hex) = RGB or YUV 4:4:4, planar 24 (hex) = 1-bit indexed 45 (hex) = 2-bit indexed 66 (hex) = 4-bit indexed 87 (hex) = 8-bit indexed A0 (hex) = packed YUY2 4:2:2 A1 (hex) = packed UYVY 4:2:2 AC (hex) = 16-bit variable contents 4:4:4 E8 (hex) = 32-bit variable contents 4:2:2 EC (hex) = 32-bit variable contents 4:4:4
Offset 0x104 - Source_Window_Size				
31:27	Reserved[4:0]	R/W	0x00	
26:16	Pfu_lsize[10:0]	R/W	0x000	Line size Defines size of input window 1: one pixel
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Pfu_lcount[10:0]	R/W	0x000	Line count Defines size of input window 1: one line
Offset 0x108 - Variable_Format_Register				
31:29	Pfu_size_c4[2:0]	R/W	0x0	Size component #4 (alpha or V) Number of bits minus 1 (e.g. 7 = 8 bits per component)
28:24	Pfu_offs_c4[4:0]	R/W	0x00	Offset component #4 (alpha or V) Index of MSB position within 32 bit word (0-31)
23:21	Pfu_size_c3[2:0]	R/W	0x0	Size component #3 (V or B or Y2) number of bits minus 1 (e.g. 7 = 8 bits per component)
20:16	Pfu_offs_c3[4:0]	R/W	0x00	Offset component #3 (V or B or Y2) index of MSB position within 32 bit word (0-31)
15:13	Pfu_size_c2[2:0]	R/W	0x0	Size component #2 (U or G) number of bits minus 1 (e.g. 7 = 8 bits per component)
12:8	Pfu_offs_c2[4:0]	R/W	0x00	Offset component #2 (U or G) index of MSB position within 32 bit word (0-31)
7:5	Pfu_size_c1[2:0]	R/W	0x0	Size component #1 (Y or R) number of bits minus 1 (e.g. 7 = 8 bits per component)
4:0	Pfu_offs_c1[4:0]	R/W	0x00	Offset component #1 (Y or R) index of MSB position within 32 bit word (0-31)
Offset 0x140 - Source_Base_Address_1				
31:28	Unused[3:0]	R/W	0x0	
27:3	Pfu_base1[24:0]	R/W	0x0000000	Base address DMA #1 used depending on PFU_BAMODE setting
2:0	Pfu_offset1[2:0]	R/W	0x0	Base address byte offset DMA #1 bits define pixel offset within multi pixel 64 bit words (e.g. a 16bit pixel can be placed on any 16 bit boundary)
Offset 0x144 - Source_Line_Pitch_1				

Table 326: Memory Based Scaler (MBS) Registers *…continued*

Bit	Symbol	Access	Value	Description
31:15	Unused[16:0]	R/W	0x00000	
14:3	Pfu_pitch1[11:0]	R/W	0x000	Line pitch DMA #1, signed value (twos complement) Used for all packed formats and for plane 1.
2:0	Unused2[2:0]	R/W	0x0	
Offset 0x148 - Source_Base_Address_2				
31:28	Unused[3:0]	R/W	0x0	
27:3	Pfu_base2[24:0]	R/W	0x0000000	Base address DMA #2 Used depending on PFU_BAMODE setting.
2:0	Pfu_offset2[2:0]	R/W	0x0	Base address byte offset DMA #2 Bits define pixel offset within multi pixel 64-bit words (e.g., a 16-bit pixel can be placed on any 16-bit boundary).
Offset 0x14C - Source_Line_Pitch_2				
31:15	Unused[16:0]	R/W	0x00000	
14:3	Pfu_pitch2[11:0]	R/W	0x000	Line pitch DMA #2, signed value (twos complement) Used for planes 2 and 3.
2:0	Unused2[2:0]	R/W	0x0	
Offset 0x150 - Source_Base_Address_3				
31:28	Unused[3:0]	R/W	0x0	
27:3	Pfu_base3[24:0]	R/W	0x0000000	Base address DMA #3 Used depending on PFU_BAMODE setting.
2:0	Pfu_offset3[2:0]	R/W	0x0	Base address byte offset DMA #3 Bits define pixel offset within multi pixel 64-bit words (e.g., a 16-bit pixel can be placed on any 16-bit boundary).
Offset 0x154 - Source_Base_Address_4				
31:28	Unused[3:0]	R/W	0x0	
27:3	Pfu_base4[24:0]	R/W	0x0000000	Base address DMA #4 Used depending on PFU_BAMODE setting.
2:0	Unused2[2:0]	R/W	0x0	
Offset 0x158 - Source_Base_Address_5				
31:28	Unused[3:0]	R/W	0x0	
27:3	Pfu_base5[24:0]	R/W	0x0000000	Base address DMA #5 Used depending on PFU_BAMODE setting.
2:0	Unused2[2:0]	R/W	0x0	
Offset 0x15C - Source_Base_Address_6				
31:28	Unused[3:0]	R/W	0x0	
27:3	Pfu_base6[24:0]	R/W	0x0000000	Base address DMA #6 Used depending on PFU_BAMODE setting.
2:0	Unused2[2:0]	R/W	0x0	
Offset 0x200 - HSP_Zoom_0				

Table 326: Memory Based Scaler (MBS) Registers *…continued*

Bit	Symbol	Access	Value	Description
31:29	Hsp_phase_mode[2:0]	R/W	0x0	Phase mode 0: 64 phases 1: 32 phases 2: 16 phases 3: 8 phases 4: 4 phases 5: 2 phases 6: fixed phase 7: linear phase interpolation (only valid for 4 component mode)
28	Hsp_no_crop	R/W	0x0	Disable line length cropping 0: cropping enabled (default) 1: cropping disabled, used for striped scaling tasks
27	Hsp_uv_seq	R/W	0x0	Input sample re-sequence 0: normal sequence (default) 1: skip first input sample (used for striped scaling tasks in 4:2:x transposed mode if stripe would start on an unwanted pixel location)
26	Hsp_fir_comp[1:0]	R/W	0x0	Horizontal filter components 0: three components, 6 tap FIR each 1: four components, 3 tap FIR each In color space matrix mode this value has to remain zero
25	Hsp_ri_crop	R/W	0x0	Enable IFF run-in cropping 0: cropping disabled (default) 1: cropping enabled (used for striped scaling tasks in 4:2:x transposed mode to skip an additional three samples at the start of each line)
24:20	Reserved[4:0]	R/W	0x00	
19:0	Hsp_zoom_0[19:0]	R/W	0x00000	Initial zoom for 1st pixel in line (unsigned, LSB = 2-16) 2 0000 (hex): downscale 50% 1 0000 (hex): no scaling = 100% 0 8000 (hex): zoom 2 x (transposed: downscale 50%)
Offset 0x204 - HSP_Phase				
31	Reserved	R/W	0x0	
30:28	Hsp_qshift[2:0]	R/W	0x0	Quantization shift control Used to change quantization before being multiplied with HSP_MULTIPLY. 100 (bin): divide by 16 101 (bin): divide by 8 110 (bin): divide by 4 111 (bin): divide by 2 000 (bin): multiply by 1 001 (bin): multiply by 2 010 (bin): multiply by 4 011 (bin): multiply by 8 Warning: A value range overflow caused by an improper quantization shift can not be compensated later by multiplying with a HSP_MULTIPLY value below 0.5!
27:26	Reserved2[1:0]	R/W	0x0	
25	Hsp_qsign	R/W	0x0	Quantization sign bit
24:16	Hsp_qmultiply[8:0]	R/W	0x000	Quantization multiply control Used to compensate for different weight sums in transposed polyphase or color space matrix mode, remaining bits are fractions (largest number is 511/512) Value range: 0<=m<1.0 . Instead of using values in the range of m<0.5 the quantization shift HSP_QSHIFT should be modified to gain more precision in the truncated result.
15:14	Reserved3[1:0]	R/W	0x0	
13:0	Hsp_offset_0[13:0]	R/W	0x0000	Initial start offset for DTO
Offset 0x208 - HSP_DZoom_0				
31:26	Reserved[5:0]	R/W	0x00	

Table 326: Memory Based Scaler (MBS) Registers *…continued*

Bit	Symbol	Access	Value	Description
25:0	Hsp_dzoom_0[25:0]	R/W	0x0000000	Initial zoom delta for 1 pixel in line (signed, LSB = 2-27) Used for non constant scaling ratios.
Offset 0x20C - HSP_DDZoom				
31:29	Reserved[2:0]	R/W	0x0	
28:0	Hsp_ddzoom[28:0]	R/W	0x00000000	Zoom delta change (signed, LSB = 2-40) used for non constant scaling ratios
Offset 0x220 - Color_space_matrix_coefficients_C00_C02				
31:30	Unused[1:0]	R/W	0x0	
29:20	Csm_c02[9:0]	R/W	0x000	Coefficient C02, twos complement
19:10	Csm_c01[9:0]	R/W	0x000	Coefficient C01, twos complement
9:0	Csm_c00[9:0]	R/W	0x000	Coefficient C00, twos complement
Offset 0x224 - Color_space_matrix_coefficients_C10_C12				
31:30	Unused[1:0]	R/W	0x0	
29:20	Csm_c12[9:0]	R/W	0x000	Coefficient C12, twos complement
19:10	Csm_c11[9:0]	R/W	0x000	Coefficient C11, twos complement
9:0	Csm_c10[9:0]	R/W	0x000	Coefficient C10, twos complement
Offset 0x228 - Color_space_matrix_coefficients_C20_C22				
31:30	Unused[1:0]	R/W	0x0	
29:20	Csm_c22[9:0]	R/W	0x000	Coefficient C22, twos complement
19:10	Csm_c21[9:0]	R/W	0x000	Coefficient C21, twos complement
9:0	Csm_c20[9:0]	R/W	0x000	Coefficient C20, twos complement
Offset 0x22C - Color_space_matrix_offset_coefficients_D0_D2				
31:29	Unused[2:0]	R/W	0x0	
28	Csm_d2_twos	R/W	0x0	Offset coefficient D2 type 0: unsigned 1: signed
27:20	Csm_d2[7:0]	R/W	0x00	Offset coefficient D2
19	Unused2	R/W	0x0	
18	Csm_d1_twos	R/W	0x0	Offset coefficient D1 type 0: unsigned 1: signed
17:10	Csm_d1[7:0]	R/W	0x00	Offset coefficient D1
9	Unused3	R/W	0x0	
8	Csm_d0_twos	R/W	0x0	Offset coefficient D0 type 0: unsigned 1: signed
7:0	Csm_d0[7:0]	R/W	0x00	Offset coefficient D0
Offset 0x230 - Color_space_matrix_offset_coefficients_E0_E2				
31:30	Unused[1:0]	R/W	0x0	

Table 326: Memory Based Scaler (MBS) Registers *…continued*

Bit	Symbol	Access	Value	Description
29:20	Csm_e2[9:0]	R/W	0x000	Offset coefficient E2, twos complement
19:10	Csm_e1[9:0]	R/W	0x000	Offset coefficient E1, twos complement
9:0	Csm_e0[9:0]	R/W	0x000	Offset coefficient E0, twos complement
Offset 0x240 - VSP_Zoom_0				
31:29	Vsp_phase_mode[2:0]	R/W	0x0	Phase mode 0: 64 phases 1: 32 phases 2: 16 phases 3: 8 phases 4: 4 phases 5: 2 phases 6: fixed phase 7: linear phase interpolation (only valid for4 component mode)
28	Reserved	R/W	0x0	
27:26	Vsp_fir_comp[1:0]	R/W	0x0	Vertical filter components 00: two components, 6 tap FIR each[1] ([1] Filter lengths differ when in de-interlacing mode) 01: reserved 10: three components, 4 tap FIR each 11: four components, 3 tap FIR each
25:20	Reserved2[5:0]	R/W	0x00	
19:0	Vsp_zoom_0[19:0]	R/W	0x00000	Initial zoom for 1st pixel in line (unsigned, LSB = 2-16) 2 0000 (hex): downscale 50% 1 0000 (hex): no scaling = 2 0 0 8000 (hex): zoom 2 x
Offset 0x244 - VSP_Phase				
31	Reserved	R/W	0x0	
30:28	Vsp_qshift[2:0]	R/W	0x0	Quantization shift control Used to change quantization before being multiplied with 0.5. 100 (bin): divide by 16 101 (bin): divide by 8 110 (bin): divide by 4 111 (bin): divide by 2 000 (bin): multiply by 1 001 (bin): multiply by 2 010 (bin): multiply by 4 011 (bin): multiply by 8
27:26	Reserved2[1:0]	R/W	0x0	
25	Vsp_qsign	R/W	0x0	Quantization sign bit
24:21	Vsp_ldiff_c[3:0]	R/W	0x0	Line offset for chroma line count (signed) (needed for slicing only)
20	Reserved3	R/W	0x0	
19:14	Vsp_offset_c[13:8]	R/W	0x00	Initial start offset for chroma DTO (used for 4:2:0 scaling and deinterlacing only)
13:0	Vsp_offset_0[13:0]	R/W	0x0000	Initial start offset for DTO
Offset 0x248 - VSP_DZoom_0				
31:26	Reserved[5:0]	R/W	0x00	
25:0	Vsp_dzoom_0[25:0]	R/W	0x0000000	Initial zoom delta for 1 pixel in line (signed, LSB = 2-27) Used for non-constant scaling ratios.
Offset 0x24C - VSP_DDZoom				
31:29	Reserved[2:0]	R/W	0x0	
28:0	Vsp_ddzoom[28:0]	R/W	0x00000000	Zoom delta change (signed, LSB = 2-40) Used for non-constant scaling ratios.
Offset 0x280 - Color_Key_Control				

Table 326: Memory Based Scaler (MBS) Registers *…continued*

Bit	Symbol	Access	Value	Description
31:30	Ckey_k2a[1:0]	R/W	0x0	Color key to alpha convert Alpha value for non-key sample is taken from CKEY_ALPHA register, alpha value for key sample is set to zero. 0: no alpha manipulation 1: fixed alpha at output 10: reserved 11: color key to alpha convert
29:28	Ckey_a2k[1:0]	R/W	0x0	Alpha mode to color key convert Samples with alpha component below 80 (hex) are replaced with values defined in CKEY_COMP 1-3. 0: no alpha manipulation 1: reserved 10: reserved 11: alpha to color key convert
27:26	Ckey_replace[1:0]	R/W	0x0	Color keying replace mode 0: no color component manipulation 1: replace keyed color components with black (10, 10, 10) 10: replace keyed color components with gray (80, 80, 80) 11: replace keyed color components with last non-key value
25:24	Reserved[1:0]	R/W	0x0	
23:16	Ckey_mask1[7:0]	R/W	0x00	Color key mask component 1 Defines bits of color component that are compared against color key value setting to key sample.
15:8	Ckey_mask2[7:0]	R/W	0x00	Color key mask component 2 Defines bits of color component that are compared against color key value setting to key sample.
7:0	Ckey_mask3[7:0]	R/W	0x00	Color key mask component 3 Defines bits of color component that are compared against color key value setting to key sample.
Offset 0x284 - Color_Key_Components				
31:24	Ckey_alpha[7:0]	R/W	0x00	Alpha value Defines the alpha value to be used for keyed samples.
23:16	Ckey_comp1[7:0]	R/W	0x00	Color key component 1 Defines value of color key for component 1 (red or Y).
15:8	Ckey_comp2[7:0]	R/W	0x00	Color key component 2 Defines value of color key for component 2 (green or U).
7:0	Ckey_comp3[7:0]	R/W	0x00	Color key component 3 Defines value of color key for component 3 (blue or V).
Offset 0x300 - Video_Output_Format				

Table 326: Memory Based Scaler (MBS) Registers *...continued*

Bit	Symbol	Access	Value	Description
31:30	Psu_bamode[1:0]	R/W	0x0	Base address mode base 1-3 according to number of planes (plane 1-3) base 1-3, first line out, etc. (plane 1-3) base 4-6, second line out, etc. (plane 1-3) 0: single set (e.g. progressive video source) 1: alternate sets each line (e.g. anti-flicker mode) 10: reserved 11: reserved
29:14	Reserved[15:0]	R/W	0x0000	
13	Psu_endian	R/W	0x0	Output format endianness 0: same as system endianness 1: opposite of system endianness
12	Reserved2	R/W	0x0	
11:10	Psu_dither[1:0]	R/W	0x0	Output format dither mode 00: no dithering 01: error dispersion (never reset pattern) 10: error dispersion (reset pattern at startup) 11: error dispersion (reset pattern every field)
9:8	Psu_alpha[1:0]	R/W	0x0	Output format alpha mode 00: no alpha (alpha byte not written to memory) 01: alpha byte written (see Color Keying Control) 10: reserved 11: reserved
7:0	Psu_opfmt[7:0]	R/W	0x00	Output formats 00 (hex) = YUV 4:2:0, semi-planar 03 (hex) = YUV 4:2:0, planar 08 (hex) = YUV 4:2:2, semi-planar 0B (hex) = YUV 4:2:2, planar 0F (hex) = RGB or YUV 4:4:4, planar A9 (hex) = compressed 4/4/4 + (4 bit alpha) AA (hex) = compressed 4/5/3 + (4 bit alpha) AD (hex) = compressed 5/6/5 A0 (hex) = packed YUY2 4:2:2 A1 (hex) = packed UYVY 4:2:2 E2 (hex) = YUV or RGB 4:4:4 + (8 bit alpha) E3 (hex) = VYU 4:4:4 + (8 bit alpha)
Offset 0x304 - Target_Window_Size				
31:27	Reserved[4:0]	R/W	0x00	
26:16	Psu_lsize[10:0]	R/W	0x000	Line size Used for horizontal cropping after scaling. 0: cropping disabled 1: one pixel
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Psu_lcount[10:0]	R/W	0x000	Line count Used for vertical cropping after scaling. 0: cropping disabled 1: one line
Offset 0x340 - Target_Base_Address_1				
31:28	Unused[3:0]	R/W	0x0	
27:3	Psu_base1[24:0]	R/W	0x0000000	Base address DMA #1 Used depending on PSU_BAMODE setting.
2:0	Psu_offset1[2:0]	R/W	0x0	Base address byte offset DMA #1 Bits define pixel offset within multi pixel 64-bit words (e.g., a 16-bit pixel can be placed on any 16-bit boundary).

Table 326: Memory Based Scaler (MBS) Registers *...continued*

Bit	Symbol	Access	Value	Description
Offset 0x344 - Target_Line_Pitch_1				
31:15	Unused[16:0]	R/W	0x00000	
14:3	Psu_pitch1[11:0]	R/W	0x000	Line pitch DMA #1, signed value (twos complement) Used for all packed formats and for plane 1.
2:0	Unused2[2:0]	R/W	0x0	
Offset 0x348 - Target_Base_Address_2				
31:28	Unused[3:0]	R/W	0x0	
27:3	Psu_base2[24:0]	R/W	0x0000000	Base address DMA #2 Used depending on PSU_BAMODE setting.
2:0	Psu_offset2[2:0]	R/W	0x0	Base address byte offset DMA #2 Bits define pixel offset within multi pixel 64-bit words (e.g., a 16-bit pixel can be placed on any 16-bit boundary).
Offset 0x34C - Target_Line_Pitch_2				
31:15	Unused[16:0]	R/W	0x00000	
14:3	Psu_pitch2[11:0]	R/W	0x000	Line pitch DMA #2, signed value (twos complement) Used for planes 2 and 3.
2:0	Unused2[2:0]	R/W	0x0	
Offset 0x350 - Target_Base_Address_3				
31:28	Unused[3:0]	R/W	0x0	
27:3	Psu_base3[24:0]	R/W	0x0000000	Base address DMA #3 Used depending on PSU_BAMODE setting.
2:0	Psu_offset3[2:0]	R/W	0x0	Base address byte offset DMA #3 Bits define pixel offset within multi pixel 64-bit words (e.g., a 16-bit pixel can be placed on any 16-bit boundary).
Offset 0x354 - Target_Base_Address_4				
31:28	Unused[3:0]	R/W	0x0	
27:3	Psu_base4[24:0]	R/W	0x0000000	Base address DMA #4 Used depending on PSU_BAMODE setting.
2:0	Unused2[2:0]	R/W	0x0	
Offset 0x358 - Target_Base_Address_5				
31:28	Unused[3:0]	R/W	0x0	
27:3	Psu_base5[24:0]	R/W	0x0000000	Base address DMA #5 Used depending on PSU_BAMODE setting.
2:0	Unused2[2:0]	R/W	0x0	
Offset 0x35C - Target_Base_Address_6				
31:28	Unused[3:0]	R/W	0x0	
27:3	Psu_base6[24:0]	R/W	0x0000000	Base address DMA #6 Used depending on PSU_BAMODE setting
2:0	Unused2[2:0]	R/W	0x0	
Offset 0x400 - Color_Look_Up_Table				
31:24	Lut_alpha_x_[7:0]	W	0x00	Alpha

Table 326: Memory Based Scaler (MBS) Registers *…continued*

Bit	Symbol	Access	Value	Description
23:16	Lut_red_x[7:0]	W	0x00	Red or Y
15:8	Lut_green_x[7:0]	W	0x00	Green or U
7:0	Lut_blue_x[7:0]	W	0x00	Blue or V
Offset 0x800 - Horizontal_LS				
31:30	Unused[1:0]	R/W	0x0	
29:20	Tap_2_x[9:0]	W	0x000	Inverted coefficient, tap #2, twos complement
19:10	Tap_1_x[9:0]	W	0x000	Inverted coefficient, tap #1, twos complement
9:0	Tap_0_x[9:0]	W	0x000	Inverted coefficient, tap #0, twos complement
Offset 0x804 - Horizontal_MS				
31:30	Unused[1:0]	R/W	0x0	
29:20	Tap_5_x[9:0]	W	0x000	Inverted coefficient, tap #5, twos complement
19:10	Tap_4_x[9:0]	W	0x000	Inverted coefficient, tap #4, twos complement
9:0	Tap_3_x[9:0]	W	0x000	Inverted coefficient, tap #3, twos complement
Offset 0xA00 - Vertical_Luma_LS				
31:30	Unused[1:0]	R/W	0x0	
29:20	Tap_2_x_9_0__1563877[9:0]	W	0x000	Inverted coefficient, tap #2, twos complement
19:10	Tap_1_x_9_0__1563887[9:0]	W	0x000	Inverted coefficient, tap #1, twos complement
9:0	Tap_0_x_9_0__1563897[9:0]	W	0x000	Inverted coefficient, tap #0, twos complement
Offset 0xA04 - Vertical_Luma_MS				
31:30	Unused[1:0]	R/W	0x0	
29:20	Tap_5_x_9_0__1563837[9:0]	W	0x000	Inverted coefficient, tap #5, twos complement
19:10	Tap_4_x_9_0__1563847[9:0]	W	0x000	Inverted coefficient, tap #4, twos complement
9:0	Tap_3_x_9_0__1563857[9:0]	W	0x000	Inverted coefficient, tap #3, twos complement
Offset 0xC00 - Vertical_Chroma_LS				
31:30	Unused[1:0]	R/W	0x0	
29:20	Tap_2_x_9_0__1563967[9:0]	W	0x000	Inverted coefficient, tap #2, twos complement
19:10	Tap_1_x_9_0__1563977[9:0]	W	0x000	Inverted coefficient, tap #1, twos complement
9:0	Tap_0_x_9_0__1563987[9:0]	W	0x000	Inverted coefficient, tap #0, twos complement
Offset 0xC04 - Vertical_Chroma_MS				
31:30	Unused[1:0]	R/W	0x0	
29:20	Tap_5_x_9_0__1563927[9:0]	W	0x000	Inverted coefficient, tap #5, twos complement

Table 326: Memory Based Scaler (MBS) Registers *…continued*

Bit	Symbol	Access	Value	Description
19:10	Tap_4_x_9_0__1563937[9:0]	W	0x000	Inverted coefficient, tap #4, twos complement
9:0	Tap_3_x_9_0__1563947[9:0]	W	0x000	Inverted coefficient, tap #3, twos complement
Offset 0xE00 - Formatter_Control				
31:9	Reserved[22:0]	R/W	0x000000	
8:7	Line_valid[1:0]	W	0x3	Determines the lines to be measured 0: reserved 1: only passes even occurrences of lines for measurements 10: only passes odd occurrences of lines for measurements 11: passes all occurrences of lines for measurements
6	Input_format_uv	W	0x0	UV input range type selector: 0 (unsigned) = 0 (minimum) 32 (negative, 100% saturation) 256 (uncolored) 480 (positive, 100% saturation) 511 (maximum) 1 (signed) = -256 (minimum) -224 (negative, 100% saturation) 0 (uncolored) 224 (positive, 100% saturation) 255 (maximum)
5	Input_format_y	W	0x0	Y input range type selector: 0 (unsigned) = 0 (minimum) 32 (black) 470 (white, 100%) 511 (maximum) 1 (signed) = -256 (minimum) -224 (black) 214 (white, 100%) 255 (maximum)
4:3	Output_format_uv[1:0]	W	0x0	UV output type selector The output range is always interpreted as signed 0: Output is 9 bit, with LSB fixed to 0 (true 8 bit) 1: Output is 9 bit, with LSB copied from LSB + 1 10: Output is 9bit, from undithered 8 bit 11: Output is true 9 bit
2:1	Output_format_y[1:0]	W	0x0	Y output type selector The output range is always interpreted as unsigned 0: Output is 9 bit, with LSB fixed to 0 (true 8 bit) 1: Output is 9 bit, with LSB copied from LSB + 1 10: Output is 9bit, from undithered 8 bit 11: Output is true 9 bit
0	Reserved2	R/W	0x0	
Offset 0xE0C - Flaggen_Control_Registers				
31:6	Reserved[25:0]	R/W	0x0000000	
5	Eaf_bbar_enable	R/W	0x0	1 : meas_finish is only generated if eaf_bbar has occurred 0 : meas_finish is independent from eaf_bbar

Table 326: Memory Based Scaler (MBS) Registers *...continued*

Bit	Symbol	Access	Value	Description
4	Eaf_blklvl_enable	R/W	0x0	1 : meas_finish is only generated if eaf_blklvl has occurred 0 : meas_finish is independent from eaf_blklvl
3	Eaf_histo_enable	R/W	0x0	1 : meas_finish is only generated if eaf_histo has occurred 0 : meas_finish is independent from eaf_histo
2	Eaf_noise_enable	R/W	0x0	1 : meas_finish is only generated if eaf_noise has occurred 0 : meas_finish is independent from eaf_noise
1	Eaf_uvbw_enable	R/W	0x0	1 : meas_finish is only generated if eaf_uvbw has occurred 0 : meas_finish is independent from eaf_uvbw
0	Eaf_film_enable	R/W	0x0	1 : meas_finish is only generated if eaf_film has occurred 0 : meas_finish is independent from eaf_film
Offset 0xE10 - Histogram_Control_Register				
31:27	Reserved[4:0]	R/W	0x00	
26	Subsample_8x	R/W	0x0	8x subsample or 4x subsample of input data 0: 4x subsample (for standard definition input) 1: 8x subsample (for high definition input)
25	Repair_ac_error	R/W	0x0	Enable measurement on gradual slope 0: disable (default) 1: enable
24:17	Reserved2[7:0]	R/W	0x00	
16:8	Histo_thres[8:0]	R/W	0x000	Threshold value for the measurement algorithm (0...511)
7:4	Histo_gain[3:0]	R/W	0xF	0 to 10 = Gain for selection of the 8 output bits 11 to 14 = not used 1111: Calculate HISTO_GAIN in hardware
3	Reserved3	R/W	0x0	
2	Histo_noise_red	R/W	0x1	Enable / disable noise reduction feature 0: disable 1: enable
1	Filt2_enable	R/W	0x1	Enable / disable the use of filter 2 0: disable 1: enable
0	Filt1_enable	R/W	0x1	Enable / Disable the use of filter1 0: disable 1: enable
Offset 0xE18 - Histogram_Window_Start				
31:27	Reserved[4:0]	R/W	0x00	
26:16	Histo_xws[10:0]	R/W	0x001	Horizontal histogram measurement window start
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Histo_yws[10:0]	R/W	0x001	Vertical histogram measurement window start
Offset 0xE1C - Histogram_Window_End				

Table 326: Memory Based Scaler (MBS) Registers *…continued*

Bit	Symbol	Access	Value	Description
31:27	Reserved[4:0]	R/W	0x00	
26:16	Histo_xwe[10:0]	R/W	0x2D0	Horizontal histogram measurement window end
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Histo_ywe[10:0]	R/W	0x120	Vertical histogram measurement window end
Offset 0xE20 - Noise_Estimator_Control_Register_1				
31:22	Reserved[9:0]	R/W	0x000	
21	Nest_stall	W	0x0	Stall of measurement unit 0: measurement unit active (default) 1: measurement unit stalled
20:19	Clip_offset[1:0]	W	0x0	Selection of clipping levels
18	Sel_sob_neglect_ext	W	0x0	Selection of external SOB_NEGLECT_FLAG
17	Sob_neglect_ext	W	0x0	External SOB_NEGLECT flag
16:13	Compensate[3:0]	W	0x0	Signed value added to NEST before low pass filtering
12:10	Gain_upbnd[2:0]	W	0x0	Selection of coupling to low boundary
9:8	Ypscale[1:0]	W	0x0	Scaling factor of the prefilter unit
7:0	Wanted_value[7:0]	W	0x46	Controls the up/down counting of images. When the number of times ce_SOB and ce_SAD are in agreement (equal 1) in an image, is smaller than WANTED_VALUE, unit are counting down; otherwise counting up.
Offset 0xE24 - Noise_Estimator_Control_Register_2				
31:24	Reserved[7:0]	R/W	0x00	
23:16	Lb_detail[7:0]	W	0xE2	Lower limit for absolute difference between two adjacent pixels
15:8	Reserved2[7:0]	R/W	0x00	
7:0	Upb_detail[7:0]	W	0xBE	Upper limit for absolute difference between two adjacent pixels
Offset 0xE28 - Noise_Estimator_Window_Start				
31:27	Reserved[4:0]	R/W	0x00	
26:16	Nest_xws[10:0]	W	0x001	Horizontal Noise Estimator window start
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Nest_yws[10:0]	W	0x001	Vertical Noise Estimator window start
Offset 0xE2C - Noise_Estimator_Window_End				
31:27	Reserved[4:0]	R/W	0x00	
26:16	Nest_xwe[10:0]	W	0x2D0	Horizontal NOise Estimator window end
15:11	Reserved2[4:0]	R/W	0x00	
10:0	nest_ywe[10:0]	W	0x120	Vertical Noise Estimator window end
Offset 0xE30 - Black_Bar_Detector_Control_Register				
31:24	Bbd_event_value2[7:0]	W	0x10	If number of black pixels > BBD_EVENT_VALUE2 the line is considered as black.

Table 326: Memory Based Scaler (MBS) Registers *…continued*

Bit	Symbol	Access	Value	Description
23:16	Bbd_slice_level2[7:0]	W	0x20	If luminance level < BBD_SLICE_LEVEL2 (multiplied by 2) the pixel is considered as black.
15:8	Bbd_event_value1[7:0]	W	0x15	If number of black pixels > BBD_EVENT_VALUE1 the line is considered as black.
7:0	Bbd_slice_level1[7:0]	W	0x55	If luminance level < BBD_SLICE_LEVEL1 (multiplied by 2) the pixel is considered as black.
Offset 0xE38 - Black_Bar_Detection_Window_Start				
31:27	Reserved[4:0]	R/W	0x00	
26:16	Bbd_xws[10:0]	W	0x001	Horizontal Black Bar Detection window start
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Bbd_yws[10:0]	W	0x001	Vertical Black Bar Detection window start
Offset 0xE3C - Black_Bar_Detection_Window_End				
31:27	Reserved[4:0]	R/W	0x00	
26:16	Bbd_xwe[10:0]	W	0x2D0	Horizontal Black Bar Detection window end
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Bbd_ywe[10:0]	W	0x120	Vertical Black Bar detection window end
Offset 0xE40 - Black_Level_Detection_Control				
31:1	Reserved[30:0]	R/W	0x00000000	
0	Filt1_enable_1565528	W	0x1	Enable / Disable the use of filter1
Offset 0xE48 - Black_Level_Detection_Window_Start				
31:27	Reserved[4:0]	R/W	0x00	
26:16	Bld_xws[10:0]	W	0x001	Horizontal black level detection window start
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Bld_yws[10:0]	W	0x001	Vertical black level detection window start
Offset 0xE4C - Black_Level_Detection_Window_End				
31:27	Reserved[4:0]	R/W	0x00	
26:16	Bld_xwe[10:0]	W	0x2D0	Horizontal black level detection window end. Pixels from XWS up to and including XWE are processed.
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Bld_ywe[10:0]	W	0x120	Vertical black level detection window end. Lines from YWS up to and including YWE are processed.
Offset 0xE50 - Bandwidth_Detection_Control				
31:9	Reserved[22:0]	R/W	0x000000	
8:0	Max_delta_bw[8:0]	W	0x1FF	Slope of the rectifier: 0: no slope 511: maximum slope (default)
Offset 0xE58 - Bandwidth_Detection_Window_Start				
31:27	Reserved[4:0]	R/W	0x00	
26:16	Bwd_xws[10:0]	W	0x001	Horizontal bandwidth detection window start

Table 326: Memory Based Scaler (MBS) Registers *...continued*

Bit	Symbol	Access	Value	Description
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Bwd_yws[10:0]	W	0x001	Vertical bandwidth detection window start
Offset 0xE5C - Bandwidth_Detection_Window_End				
31:27	Reserved[4:0]	R/W	0x00	
26:16	Bwd_xwe[10:0]	W	0x2D0	Horizontal bandwidth detection window end
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Bwd_ywe[10:0]	W	0x120	Vertical bandwidth detection window end
Offset 0xE8C - Histogram_Data_Output_1				
31:29	Reserved[2:0]	R/W	0x0	
28:20	Y_max[8:0]	R	0x000	Maximum luminance value
19	Reserved2	R/W	0x0	
18:10	Y_min[8:0]	R	0x000	Minimum luminance value
9:8	Reserved3[1:0]	R/W	0x0	
7:0	Histo_max_value[7:0]	R	0x00	Level of the maximum histogram data
Offset 0xE90 - Histogram_Data_Output_2				
31:24	Histo_data_3[7:0]	R	0x00	Measured histogram data (bin 3, range 46...63)
23:16	Histo_data_2[7:0]	R	0x00	Measured histogram data (bin 3, range 32...47)
15:8	Histo_data_1[7:0]	R	0x00	Measured histogram data (bin 1, range 16...31)
7:0	Histo_data_0[7:0]	R	0x00	Measured histogram data (bin 0, range 0...15)
Offset 0xE94 - Histogram_Data_Output_3				
31:24	Histo_data_7[7:0]	R	0x00	Measured histogram data (bin 7, range 112...127)
23:16	Histo_data_6[7:0]	R	0x00	Measured histogram data (bin 6, range 96...111)
15:8	Histo_data_5[7:0]	R	0x00	Measured histogram data (bin 5, range 80...95)
7:0	Histo_data_4[7:0]	R	0x00	Measured histogram data (bin 4, range 64...79)
Offset 0xE98 - Histogram_Data_Output_4				
31:24	Histo_data_11[7:0]	R	0x00	Measured histogram data (bin 11, range 176...191)
23:16	Histo_data_10[7:0]	R	0x00	Measured histogram data (bin 10, range 160...175)
15:8	Histo_data_9[7:0]	R	0x00	Measured histogram data (bin 9, range 144...159)
7:0	Histo_data_8[7:0]	R	0x00	Measured histogram data (bin 8, range 128...143)
Offset 0xE9C - Histogram_Data_Output_5				
31:24	Histo_data_15[7:0]	R	0x00	Measured histogram data (bin 15, range 240...255)
23:16	Histo_data_14[7:0]	R	0x00	Measured histogram data (bin 14, range 224...239)
15:8	Histo_data_13[7:0]	R	0x00	Measured histogram data (bin 13, range 208...223)
7:0	Histo_data_12[7:0]	R	0x00	Measured histogram data (bin 12, range 192...207)
Offset 0xEA0 - Histogram_Data_Output_6				
31:24	Histo_data_19[7:0]	R	0x00	Measured histogram data (bin 19, range 304...319)
23:16	Histo_data_18[7:0]	R	0x00	Measured histogram data (bin 18, range 288...303)
15:8	Histo_data_17[7:0]	R	0x00	Measured histogram data (bin 17, range 272...287)
7:0	Histo_data_16[7:0]	R	0x00	Measured histogram data (bin 16, range 256...271)

Table 326: Memory Based Scaler (MBS) Registers *...continued*

Bit	Symbol	Access	Value	Description
Offset 0xEA4 - Histogram_Data_Output_7				
31:24	Histo_data_23[7:0]	R	0x00	Measured histogram data (bin 23, range 368...383)
23:16	Histo_data_22[7:0]	R	0x00	Measured histogram data (bin 22, range 352...367)
15:8	Histo_data_21[7:0]	R	0x00	Measured histogram data (bin 21, range 336...351)
7:0	Histo_data_20[7:0]	R	0x00	Measured histogram data (bin 20, range 320...335)
Offset 0xEA8 - Histogram_Data_Output_8				
31:24	Histo_data_27[7:0]	R	0x00	Measured histogram data (bin 27, range 432...447)
23:16	Histo_data_26[7:0]	R	0x00	Measured histogram data (bin 26, range 416...431)
15:8	Histo_data_25[7:0]	R	0x00	Measured histogram data (bin 25, range 400...415
7:0	Histo_data_24[7:0]	R	0x00	Measured histogram data (bin 24, range 384...399)
Offset 0xEAC - Histogram_Data_Output_9				
31:24	Histo_data_31[7:0]	R	0x00	Measured histogram data (bin 31, range 496...511)
23:16	Histo_data_30[7:0]	R	0x00	Measured histogram data (bin 30, range 480...495)
15:8	Histo_data_29[7:0]	R	0x00	Measured histogram data (bin 29, range 464...479)
7:0	Histo_data_28[7:0]	R	0x00	Measured histogram data (bin 28, range 448...463)
Offset 0xEB0 - Noise_Estimator_Data_Output_1				
31:28	Reserved[3:0]	R/W	0x0	
27:24	Nest[3:0]	R	0x0	The number of times in an image that the number of events (ce_SOB = ce_SAD) is lower than WANTED_VALUE.
23:16	Nest_filt[7:0]	R	0x00	Recursive filtered NEST value
15:8	Detail_cnt_l[7:0]	R	0x00	LSBs of the number of times in an image that the difference between two adjacent pixels falls within the range of 2*LB_DETAIL and 2*UPB_DETAIL
7:0	Detail_cnt_h[7:0]	R	0x00	MSBs of the number of times in an image that the difference between two adjacent pixels falls within the range of 2*LB_DETAIL and 2*UPB_DETAIL
Offset 0xEB4 - Noise_Estimator_Data_Output_2				
31:8	Reserved[23:0]	R/W	0x000000	
7:0	Grey_count[7:0]	R	0x00	MSBs of the number of pixels with values within the gray range
Offset 0xEB8 - Black_Bar_Detection_Data_Output_1				
31:27	Reserved[4:0]	R/W	0x00	
26:16	Bbd_last_vid_line1[10:0]	R	0x000	Number of last video line detected (first detector)
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Bbd_first_vid_line1[10:0]	R	0x000	Number of first video line detected (first detector)
Offset 0xEBC - Black_Bar_Detection_Data_output_2				
31:27	Reserved[4:0]	R/W	0x00	

Table 326: Memory Based Scaler (MBS) Registers *…continued*

Bit	Symbol	Access	Value	Description
26:16	Bbd_last_vid_line2[10:0]	R	0x000	Number of last video line detected (second detector)
15:11	Reserved2[4:0]	R/W	0x00	
10:0	Bbd_first_vid_line2[10:0]	R	0x000	Number of first video line detected (second detector) 0: disable 1: enable
Offset 0xEC0 - Black_Level_Detection_Control_Output				
31:9	Reserved[22:0]	R/W	0x000000	
8:0	Smartblack[8:0]	R	0x000	Minimum luminance level
Offset 0xEC4 - Bandwidth_Detection_Output_1				
31:29	Reserved[2:0]	R/W	0x0	
28:20	V_max[8:0]	R	0x000	Signed maximum chrominance V value
19	Reserved2	R/W	0x0	
18:10	V_min[8:0]	R	0x000	Signed minimum chrominance V value
9	Reserved3	R/W	0x0	
8:0	Uv_bandwidth[8:0]	R	0x000	Estimated value of bandwidth detection
Offset 0xEC8 - Bandwidth_Detection_Data_Output_2				
31:29	Reserved[2:0]	R/W	0x0	
28:20	U_max[8:0]	R	0x000	Signed maximum chrominance U value
19	Reserved2	R/W	0x0	
18:10	U_min[8:0]	R	0x000	Signed minimum chrominance U value
9:0	Reserved3[9:0]	R/W	0x000	
Offset 0xFE0 - Interrupt_Status				
31:30	Stat_psu[1:0]	R	0x0	Status of pixel store unit (test signal)
29:27	Reserved[2:0]	R/W	0x0	
26:16	Stat_psu_line[10:0]	R	0x000	Status of pixel store unit, line currently written
15:14	Stat_vsp[1:0]	R	0x0	Status of vertical scale pipe (test signal)
13:12	Stat_dff[1:0]	R	0x0	Status of decimation FIR filter (test signal)
11:10	Stat_hsp[1:0]	R	0x0	Status of horizontal scale pipe (test signal)
9:8	Stat_iff[1:0]	R	0x0	Status of interpolation FIR filter (test signal)
7	Stat_pfu_0_	R	0x0	Status of pixel fetch unit (test signal)
6	Stat_meas_done	R	0x0	Status of measurement progress
5	Stat_task_error	R	0x0	Processing error
4	Stat_task_end	R	0x0	Current task processing done
3	Stat_task_overflow	R	0x0	Task FIFO overflow
2	Stat_task_idle	R	0x0	Task finished and the task FIFO is empty.
1	Stat_task_empty	R	0x0	Task FIFO runs empty.
0	Stat_task_done	R	0x0	Current task finished and all write complete.

Table 326: Memory Based Scaler (MBS) Registers *…continued*

Bit	Symbol	Access	Value	Description
Offset 0xFE4 - Interrupt_Enable				
31:7	Reserved[24:0]	R/W	0x0000000	
6	Ien_meas_done	R/W	0x0	Measurement processing done
5	Ien_task_error	R/W	0x0	Processing error
4	Ien_task_end	R/W	0x0	Current task processing done.
3	Ien_task_overflow	R/W	0x0	Task FIFO overflow
2	Ien_task_idle	R/W	0x0	Task finished and the task FIFO is empty.
1	Ien_task_empty	R/W	0x0	Task FIFO runs empty.
0	Ien_task_done	R/W	0x0	Current task finished and all write complete.
Offset 0xFE8 - Interrupt_Clear				
31:7	Reserved[24:0]	R/W	0x0000000	
6	Clr_meas_done	W	0x0	Measurement processing complete
5	Clr_task_error	W	0x0	Processing error
4	Clr_task_end	W	0x0	Current task processing done.
3	Clr_task_overflow	W	0x0	Task FIFO overflow
2	Clr_task_idle	W	0x0	Task finished and the task FIFO is empty.
1	Clr_task_empty	W	0x0	Task FIFO runs empty.
0	Clr_task_done	W	0x0	Current task finished and all write complete.
Offset 0xFEC - Interrupt_Set				
31:7	Reserved[24:0]	R/W	0x0000000	
6	Set_meas_done	W	0x0	Measurement processing complete
5	Set_task_error	W	0x0	Processing error
4	Set_task_end	W	0x0	Current task processing done.
3	Set_task_overflow	W	0x0	Task FIFO overflow
2	Set_task_idle	W	0x0	Task finished and the task FIFO is empty.
1	Set_task_empty	W	0x0	Task FIFO runs empty.
0	Set_task_done	W	0x0	Current task finished and all write complete.
Offset 0xFFC - Module_ID				
31:16	Mod_id[15:0]	R	0x0119	Module ID; unique 16-bit code
15:12	Rev_major[3:0]	R	0x4	Major revision counter
11:8	Rev_minor[3:0]	R	0x0	Minor revision counter
7:0	App_size[7:0]	R	0x00	Aperture Size: 0 = 4K

5.11 MPEG video decoder

5.11.1 Introduction

The MPEG unit is a slice-level video decoder comprised of a pipeline of functional blocks. It supports MPEG2 video decoding at a maximum resolution of “main profile” at “high-level” (MP@HL) and also MPEG1 video decoding. The MPEG Video Decoder does not support the D picture-type in MPEG1; however, the decoding of a D picture is handled by the CPU and software. It is capable of decoding all eighteen video formats of the ATSC digital television standard.

The video lines of each field (top or bottom) of all picture types (I, P and B) are normally stored in one contiguous main memory block. To optimize main memory usage in decoding MPEG2 video, however, each field of a type B picture can be optionally divided into two, four or eight chunks of video lines, where each chunk can be stored in a location in main memory which is non-contiguous from other chunks. Chunking of type B picture fields is done in order to save the main memory usage, by reusing the buffer space of a chunk immediately after it has been displayed.

Another important feature of the MPEG Video Decoder is in its ability to perform error concealment. For terrestrial ATSC broadcast, errors in transmission can be quite frequent especially in areas having less than ideal reception. Being able to conceal short bursts of errors in the ATSC video stream is a key advantage in these areas. Error concealment is performed by patching the missing pixels (or errors) with corresponding pixels from the stored backward or forward reference frames.

The MPEG Video Decoder is a slice-level decoder. It decodes all the macroblocks in a given slice, reconstructs the pixels of each macroblock and stores the pixels in designated locations in main memory. All other aspects of MPEG decoding functions are carried out by the CPU with proper programming. This includes parsing of the sequence header down to the slice header of the MPEG stream. The relevant information decoded by software (picture type, structure, quantization matrix) is programmed into corresponding control registers to direct the MPEG Video Decoder to decode the macroblocks within each slice.

Other functions related to MPEG decoding that are performed by the CPU include

- Frame buffer management.
- Trick modes, such as frame/field freeze implementation.
- Decoding of I frames only for mosaic implementation.
- Slow motion and fast forward or fast reverse implementation.
- Write-back of run-level pairs and macroblock headers to separate main-memory buffers.

5.11.2 Standard features

A mode in which each macroblock is stored in memory at 1/2 the normal horizontal resolution of image is supported in order to reduce the overall memory and bandwidth requirements of the system. The algorithm does, however, produce minor artifacts and is only to be used for reduced resolution applications (Picture-In-Picture, etc.)

5.11.2.1 Clock Programming

In the PNX8550, the clock for the VMPG is set to 143 MHz.

5.11.2.2 Reset-related issues

The MPEG Video Decoder is reset by a hardware reset or a software reset. The hardware reset signal is generated from the external PIN. The software reset is initiated by writing a ‘Reset the MPEG Video Decoder’ command in the VLD_COMMAND register. See Table 327 for the details on the software reset procedure. The COMMAND_DONE bit in the VLD_MC_STATUS register is set upon completion of a software reset, which leads to an interrupt to the CPU if the corresponding interrupt is enabled.

NOTE: The VLD_INP_CNT is cleared after reset. However, this is not treated as a DMA_INPUT_DONE condition in the VLD and the CPU is not be interrupted by the VLD after reset with a DMA_INPUT_DONE condition. The MPEG video bitstream buffer is refilled as needed and the VLD_INP_CNT rewritten with a valid value before issuing new commands to the VLD after reset.

Table 327: Software Reset Procedure

Cycle No.	Action	Remarks
i	CPU issues the ‘Reset the MPEG Video Decoder’ command by writing the corresponding command code into the VLD_COMMAND register.	
i to j	1) VLD will complete any memory DMA transactions in progress 2) Any new DMA transactions are aborted. 3) Then VLD raises the vld_ready_to_reset signal	Any DMA transactions, once started, will not be aborted in the middle
i to k	1) MC will complete any memory DMA transactions in progress. 2) Any new DMA transactions are aborted. 3) Then MC raises the mc_ready_to_reset signal	Any DMA transactions, once started, will not be aborted in the middle
k	If (vld_ready_to_reset & mc_ready_to_reset) VLD and MC will perform the full reset.	Assumes k > j; otherwise it is jth cycle. The full reset resets all internal buffers, state machines. After the reset, The VLD_MC_STATUS register will have 0x1 value and the following control registers will have 0x0 value: VLD_COMMAND, VLD_CTL, VLD_BIT_CNT, MC_STATUS, MC_PFCOUNT, MC_COMMAND, VLD_INP_CNT, VLD_MBH_CNT, VLD_RL_CNT, and VLD_STATUS has 0x1 value.

5.11.2.3 Interrupt Processing

Since the VMPG is a “legacy” controller from a prior generation, the standard DVP interrupt features are not included; i.e., the Interrupt Status, Clear and Set registers. Instead, the interrupt processing is accomplished via bits in the VLD_MC_Status register bits [7:0]. The VLD_IE register bits [7:0] provide the interrupt mask bits - if a bit is set to “1” then the corresponding status interrupt is “enabled”, otherwise the status bit is disabled.

5.11.2.4 Power Management

The VMPG does include the standard power management control bit, implemented by the POWER_DOWN register (offset 0x10 5FF4). When the POWERDOWN bit is set, the controller responds to all MMIO read transactions with the read data set to 0xDEADABBA, except for an access to the POWER_DOWN register. In addition, all write transactions are ignored except a write to the POWER_DOWN register. This makes it possible for the system to set the POWERDOWN bit and then stop the clock to the VMPG in order to conserve power. All MMIO accesses still complete correctly even though the core logic is “disabled”.

5.11.3 Functional description

The MPEG Video Decoder consists of a Variable Length Decoder (VLD) block, a Run-Level Decoder and Inverse Scan (RL/IS) block, an Inverse Quantization (IQ) block, an Inverse Discrete Cosine Transform (IDCT) block and a Motion Compensation (MC) block.

The VLD decodes the Huffman encoded MPEG1 and MPEG2 video elementary bitstreams. The VLD unit, enabled by the CPU, operates independently during the slice-level decoding. The remaining decoding of bitstream is carried out by the CPU and appropriate software and the VLD assists the CPU in slave mode. The VLD outputs a stream of macroblock headers and a stream of run-level pairs. The VLD sends the macroblock header stream to the MC block and the run-level stream to the RL/IS block for further decoding.

Expansion of the run-level pairs produced by the VLD into actual quantized DCT values and rearrangement of these values in natural order are carried out by the RL/IS block.

The IQ block is responsible for restoring or “dequantizing” the quantized DCT values by multiplying them by the corresponding values in the 8x8 DCT quantization matrix. The output of the IQ block is passed to the IDCT block, which performs the inverse discrete cosine transformation on each of the 8x8 dequantized block. The output data from the IDCT process is either the video samples or the differential values to be used by the MC block to reconstruct the final video samples.

MC reconstructs the final video samples from the macroblock header data decoded by the VLD, the reference frame data stored in main memory and the differential data from the IDCT. Error concealment is carried out mainly in the MC block.

An optional feature allows the run-level pairs and macroblock headers to be output to separate main-memory buffers so that alternative processing can be done with the TriMedia™ or other processors. When this option is enabled, processing beyond the VLD (including the RL/IS, IDCT and MC) is disabled.

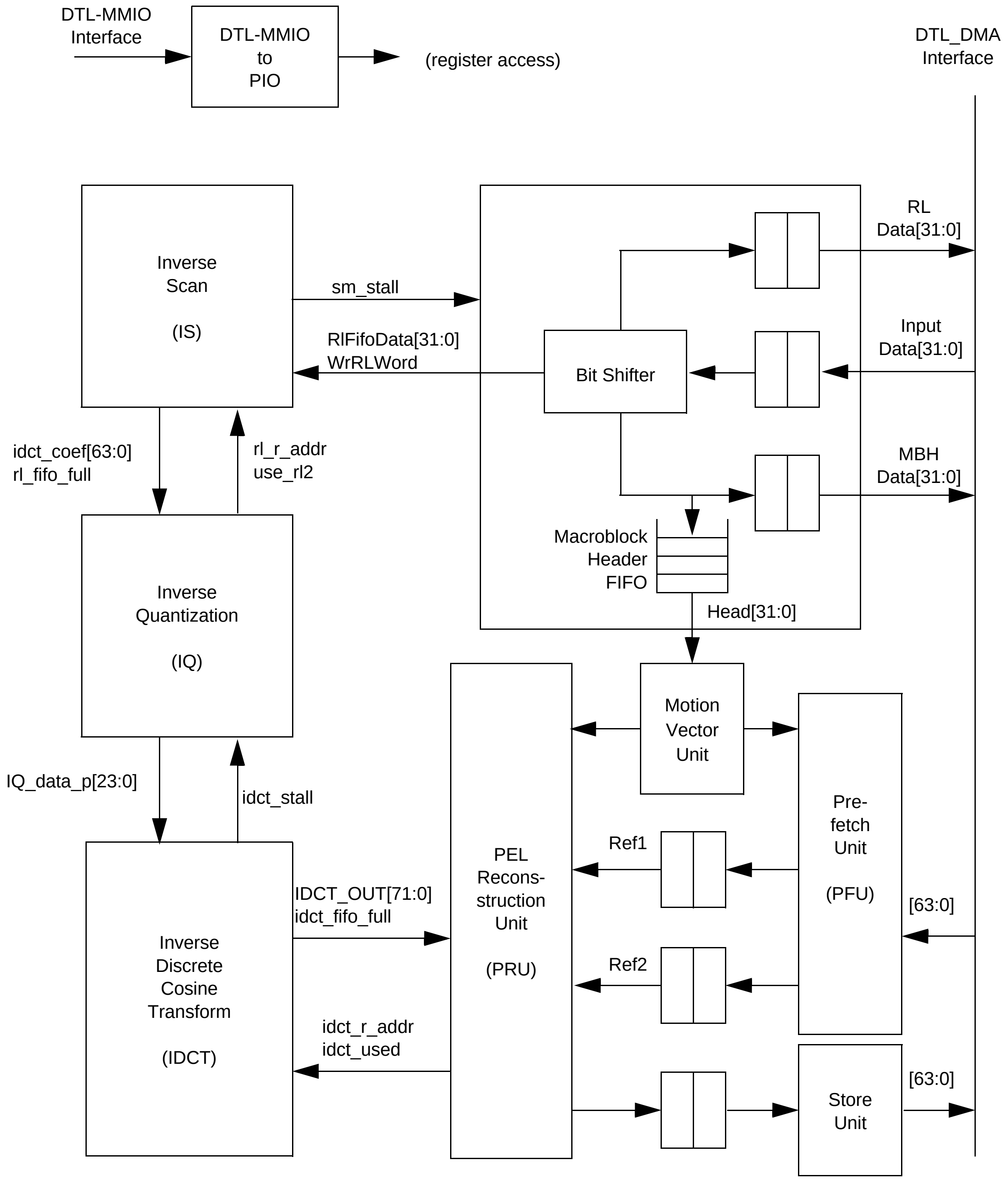


Fig 236.VMPG Block Diagram

5.11.4 Architecture

5.11.4.1 VLD only Mode

In this mode only the VLD and two write-back buffers, Run-Length (RL) data and Macroblock Header (MBH), are activated. The Input, RL and MBH DMA registers are initialized and the VLD is given a Parse or Parselong command. This initiates Variable-Length decoding of the MPEG- or MPEG2-encoded Elementary Stream data. The Run-Length code data is separated and transferred back to main memory via the RL DMA. The Macroblock Header data is separated and transferred back to main memory via the MBH DMA. When a start code is encountered, an interrupt is typically generated. This mode assumes that software performs the Inverse Scan, Quantization, IDCT and Motion Compensation algorithms. Therefore, the Inverse Scan, Quantization, IDCT and Motion Compensation hardware blocks are not used.

5.11.4.2 Full MPEG/MPEG2 Decoding with Motion Control

Section 5.11.5 that follows describes full MPEG/MPEG2 Decoding with Motion Control.

5.11.5 Operation

5.11.5.1 Variable Length Decoder (VLD)

After initialization, the CPU controls the VLD through the VLD command register. There are currently nine commands supported by the VLD:

- Shift the bit stream by some number of bits.
- Parse some number of macroblocks.
- Search for the next start code.
- Reset the MPEG Video Decoder.
- Initialize the VLD.
- Search for the given start code.
- Parse one row of macroblocks.
- Flush VLD output buffers in ‘write to memory mode.
- Parse macroblocks without stopping at the slice header.

The normal mode of operation is for the CPU to request the VLD to parse some number of macroblocks. Once the VLD has begun parsing macroblocks it may stop for any one of the following reasons:

- The command was completed with no exceptions.
- A start code was detected.
- An error was encountered in the bit stream.
- The VLD input DMA completed and the VLD is stalled waiting for more data.
- One of the output DMA for RL or HDR processing has been completed.

Under normal circumstances, the CPU is interrupted whenever the VLD halts.

5.11.5.2 Run-Length Decoder/Inverse Scan (RL/IS)

Run-Length Decoder/Inverse Scanning (RL/IS), and Inverse Quantization (IQ) follow the VLD. This takes VLD run-level output, generates scanned 8x8 DCT coefficient blocks, returns them to natural order and carries out inverse quantization. It then passes the reconstructed 8x8 two dimensional matrices to the IDCT block.

The RL-Decoder finishes a block when one of the following is encountered:

- A run value of EOB (end of block) means successful decode of a block.
- The total accumulated run is greater or equal to 64. For this case, the runlength decoder has encountered an out of bound error, the RL-Decoder error flag in the status register is set.

After the RL-Decoder step, inverse scan is performed on the RL-Decoder output. There are two types of scans: Zig-Zag Scan and Alternate Scan. The type of inverse scan used is controlled by the ALTERNATE_SCAN bit in the IQ control register.

Inverse Quantization is the step that dequantizes the DCT coefficients based on the quantization matrix, the quantizer_scale, sign of the quantized DCT coefficient, and MPEG coding type (MPEG1/MPEG2). After the arithmetic process, saturation and mismatch control is performed on the results to generate the final IQ output and pass the results to the IDCT block.

Run-length decoder: If an error is encountered, such as the DCT coefficient index is out of bound, the Runlength Decoder sets the "RL Overflow" error flag in the VLD_MC_Status register and generates an interrupt to inform the CPU of the error. The corresponding interrupt enable bit is set in the control register for the CPU to receive the interrupt.

Inverse Scan: The inverse scan step is processed in the same pipeline stage with the Runlength Decoder. There is one bit ALTERNATE_SCAN in the control register that determines if zig-zag scan or alternate scan is applied for the current picture. If the bit is set, alternate scan is used. Otherwise, zig-zag scan is used.

The zig-zag scan table is as follows:

Table 328: Zig-Zag Scan Table

Zig-Zag Scan							
0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

The alternate scan is shown in Table 329 :

Table 329: Alternate Scan Table

Alternate Scan							
0	4	6	20	22	36	38	52
1	5	7	21	23	37	39	53
2	8	19	24	34	40	50	54
3	9	18	25	35	41	51	55
10	17	26	30	42	46	56	60
11	16	27	31	43	47	57	61
12	15	28	32	44	48	58	62
13	14	29	33	45	49	59	63

5.11.5.3 Inverse Quantization (IQ)

Inverse quantization summary

- Two quantization matrices are mapped into control register memory space, initialized with default matrices but can be overwritten by downloadable matrices at picture level and sequence level.
- Global quantizer scale can be updated at slice and MB levels.
- Support both MPEG1 and MPEG2 mode Inverse Quantization.
- Inverse quantization for DCT coefficients other than Intra DC coefficients.

Inverse quantization is a process that takes natural ordered, quantized DCT coefficients in 8x8 matrix format, and produces a reconstructed 8x8 DCT coefficient matrix. There are three steps involved in this process. The first step is to select the dequantization matrix, performs the arithmetic as required based on video coding algorithm (MPEG1 or MPEG2) and macroblock type. The result is then going through a saturation process to limit the reconstructed DCT coefficients to twelve bit signed values. Lastly a mismatch control step is performed in an effort to alleviate picture quality degradation due to different IDCT implementations in the encoder and decoder.

For all other DCT coefficients including DC coefficients for inter-blocks, the reconstructed DCT coefficients are recovered by arithmetic involving the current quantizer_scale_code and the quantization matrix (weighting matrix). The default quantization matrix for intra blocks is shown in Table 330 and the default quantization matrix for non-intra blocks is shown in Table 331.

Table 330: Default quantization matrix for intra blocks

Default quantization matrix for intra blocks							
8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

Table 331: Default quantization matrix for non-intra blocks

Default quantization matrix for non-intra blocks							
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16

The quantizer_scale_code can be updated on the slice and macroblock levels. It is updated whenever a new quantizer_scale_code is decoded by the VLD. A new quantization matrix can only be loaded on the sequence or picture levels. The currently active quantization matrices are loaded into control registers so the Inverse Quantization block can access them. Note that for MPEG2 YUV4:2:2 and YUV4:4:4 mode, chrominance macroblocks use different quantization matrices than the ones used by luminance macroblocks. Therefore up to four quantization matrices can be used. For the PNX2015, YUV4:2:2 and YUV4:4:4 are not supported.

When a loadable quantizer matrix is downloaded, it has to go through an inverse zig-zag scan process. Note that alternate scan does not apply to the downloadable quantizer matrices.

The flags in bit 2 and 3 of Extra_Picture_Info indicate that the default or downloaded quantizer matrices are currently being used by the Inverse Quantization block.

When a downloadable quantization matrix is loaded into the proper control registers, the corresponding Default_Quantizer_Matrix flag in Extra_Picture_Info is also cleared. At the beginning of a picture, the two Default_Quantizer_Matrix bits are set after the default matrices are loaded.

5.11.5.4 Motion Compensation (MC)

The Motion Compensation (MC) unit is part of a slice-level MPEG Video Decoder in the PNX2015. The MC block is compatible with MPEG1 and MPEG2 MP@ML and MP@HL (ATSC compliant) video standards.

The MC block requires macroblock headers from the VLD block, YUV data from the IDCT block, and reference YUV data from main memory (SDRAM), in order to perform motion compensation. The output of motion compensation is stored directly into main memory. Picture and sequence level parameters required for motion compensation and SDRAM base addresses of reference and destination frame buffers are obtained from the control registers which are programmed by the CPU.

Reference and decoding pictures are stored on a per field basis. That is, top-field and bottom-field of each picture are stored in a separate buffer. The Y-component of each field is stored in a separate buffer, while the U- and V- components of each field are interleaved in one buffer in the order of UVUVUV. Hence up to 12 field buffers are required to decode one motion-compensated picture. The width of all field buffers in number of bytes should be the same and is specified by the user in the MMI_LINE_SIZE control register. MMI_LINE_SIZE can be larger than the actual picture width if necessary.

NOTE: MMI_LINE_SIZE is a multiple of 16 bytes.

- Supports motion compensation operations in MEPG1 and MPEG2 MP@ML and MP@HL
- Error concealment under CPU control
- Error detection
- Optimal ‘B’ frame buffer switching at macroblock row level
- Allows pipeline flush under CPU control
- Optional short-cuts available to reduce memory bandwidth usage

Error concealment operation: The MC saves the starting macroblock column position in the start_mb_col field of the MP_MC_STATUS register to allow the CPU to perform further error checking.

The MC also maintains an internal database of concealment motion vectors during normal operation, to prepare for error concealment operation as described in the next subsection.

In error concealment mode, the MC reconstructs a number of macroblocks where the motion parameters to reconstruct a macroblock are derived from the internal concealment motion vector database.

Two error concealment commands are available: errcon1_cmd, which is used when error concealment is applied from a user-programmed starting macroblock position to the beginning of the next slice being parsed by VLD (where the starting column position of the new slice is unknown to the user at the moment); and errcon2_cmd, which is used when error concealment is applied from a user-programmed starting macroblock position to a user-programmed ending macroblock position. As described below, both commands allow error concealment to span over multiple macroblock rows, and the MC_COMMAND register is cleared automatically when either command is done.

The errcon1_cmd bit is checked when the MVU fetches a mb_header which is the beginning of a new slice. If errcon1_cmd is 1, the MVU performs error concealment before processing the current mb_header. The error concealment starts at (err_mb_col, err_mb_row), as programmed by the CPU, until the macroblock just before the given new slice. On the other hand, the errcon2_cmd bit is checked by the MC while it is waiting for a new macroblock header from the VLD. If this bit is set, MC performs error concealment from the programmed starting macroblock position to the ending macroblock position. Upon completion of either command, the MVU clears the MC_COMMAND register, sets the DONE_ERRCON bit (in VLD_MC_STATUS register), and waits in an idle loop for CPU acknowledgement of the condition by the CPU writing a 1 into the DONE_ERRCON bit. If the user sets the DONE_ERRCON_IE bit in the VLD_IE register, the VLD also interrupts the CPU when the DONE_ERRCON bit is set. The MC then resumes its normal course of operation. Notice that no macroblocks are repaired if (err_mb_col + err_mb_row) >= (start_mb_col + slice_start_code - 1) for errcon1_cmd, or if (err_mb_col + err_mb_row) >= (err_end_mb_col + err_end_mb_row) for errcon2_cmd. For these cases, the MC declares completion of error concealment immediately.

The DONE_ERRCON bit is set only when all the error-concealed macroblocks are stored into main memory. That is, the MPEG pipe is flushed and idle upon completion of an error concealment operation.

When a bitstream error is detected, the user sends a command to clean out the MPEG pipe before attempting error concealment of the missing macroblocks. The command can be a MPEG-RESET or a MC FLUSH command, depending on the error situation at hand. After the MPEG pipe is flushed or reset, the CPU software commands the VLD to scan the bit stream for the next undamaged slice start code. At that point, the CPU software can decide whether to perform an error concealment operation to “repair” the damaged macroblocks from the point where the error was first detected up to the beginning of the new slice.

Two cases are possible here.

Case 1: The error macroblocks to be repaired and the new slice are all located in the current destination buffer; the CPU software can perform the following steps to repair the missing macroblocks: i) write the desired starting position of error concealment into err_mb_row and err_mb_col fields of the MC_PICINFO2 register, ii) set the errcon1_cmd bit in MC_COMMAND register, iii) send a command to the VLD to start parsing the new slice, iv) wait for the DONE_ERRCON flag in the VLD_MC_STATUS register to be set by the MC, and v) acknowledge DONE_ERRCON by writing 1 into this bit-field.

Case 2: Some of the error macroblocks and/or the new slice data are not located in the current destination buffer; the CPU software can perform the following steps to conceal the missing macroblocks: i) write the desired starting position of error concealment into err_mb_row and err_mb_col fields of the MC_PICINFO2 register, and the beginning position of the macroblock row right below the last row of the current destination buffer into the err_end_mb_row and err_end_mb_col fields of the MC_PICINFO2 register, ii) set the errcon2_cmd bit, iii) wait for DONE_ERRCON flag to be set by the MC, iv) acknowledge DONE_ERRCON, v) program the new buffer address (and any new picture-information if the buffer corresponds to a new picture) into the corresponding control registers, vi) if there are more macroblocks to be repaired in the new buffer, write the position of the first macroblock of the new buffer into the err_mb_row and err_mb_col, and repeat steps (ii) through (v) in Case 1; otherwise, send a command to the VLD to start parsing the new slice.

Bitstream Error Detection: Before processing a macroblock header received from the VLD, the MC checks the contents of header for certain parameter errors. Table 332 lists the codes of the error types checked by the MVU, which are explained below:

Table 332: MC Error Types

Error Code	Description
000000	no error
000001	invalid motion type
000010	invalid macroblock address increment (>1)
000100	macroblock overflow
001000	reserved
010000	pre-fetch coordinates out of bound
100000	no prediction for B skipped macroblocks

Invalid Motion Type: This error is detected when dual prime motion is specified for a B picture.

Invalid Macroblock Address Increment: This error is detected when the given macroblock address increment in a macroblock header is too large, such that it goes beyond the end of current row in MPEG2, or beyond the end of current picture in MPEG1.

Macroblock Overflow: This error is raised when a slice continues while the last coded macroblock is already at the end of the current row for MPEG2, or at the end of the current picture for MPEG1.

Pre-fetch Coordinates Out-Of-Bound: This error is detected when the computed pre-fetch request parameters refer to a reference macroblock which goes beyond the boundary of the reference picture.

No Prediction for B Skipped Macroblocks: This can occur when the last coded values of mv_forward and mv_backward are both 0 when skipped macroblocks are encountered in a B picture. In other words, skipped macroblocks are immediately preceded by an intra macroblock in a B picture, which is not allowed.

When any of these bit-stream errors are detected, the MVU writes the corresponding error code into the mc_error_flags field of the VLD_MC_STATUS register. This eventually causes the MPEG pipe to go into an idle state awaiting reset.

Short cuts to reduce memory bandwidth utilization: A few optional short cuts are provided to reduce the usage of memory bandwidth:

Drop the vertical half-pel flags of Y reference in all inter-macroblocks in a B picture, to reduce the number of 24x5 DMA requests. This is controlled by setting the no_y_yhalf bit in the MC Control Register.

Drop the vertical half-pel flags of UV reference in all inter-macroblocks in a B picture, to reduce the number of 24x5 DMA requests. This is controlled by setting the no_uv_yhalf bit in the MC Control Register.

Drop backward prediction for all inter-macroblocks requiring both forward and backward predictions. This is controlled by setting the no_backward bit in the MC Control Register.

Fetching macroblocks: For each macroblock the MC fetches 2-D reference data blocks from one of the 8 reference field buffers (Y or UV, top-field or bottom-field, forward or backward).

For bandwidth efficiency in DMA memory access, the MC makes special 2-D memory fetch requests to the MMI to read 24x4 or 24x5 bytes per request. It then selects the desired reference data from the fetched data, and stores the data into internal reference macroblock buffers. The extra data that is fetched is discarded. A list of all possible pre-fetch sizes is shown in Table 333.

Table 333: Possible Pre-Fetch Size

No. of Pre-Fetch Lines Per Request	No. of DMA Requests	Fetch Sizes*
4	1	wx4
5	1	wx5
8	2	wx4, wx4
9	2	wx4, wx5

Table 333: Possible Pre-Fetch Size *…continued*

No. of Pre-Fetch Lines Per Request	No. of DMA Requests	Fetch Sizes*
16	4	wx4, wx4, wx4, wx4
17	4	wx4, wx4, wx4, wx5
* w = 24 in full resolution mode		

In order to assist the CPU in determining memory bandwidth utilization due to the fetching of reference data, the MC increments the count_24x4 and count_24x5 control registers for every wx4 and wx5 MMI request respectively (where w = 16 or 24). These two (16-bit unsigned) counters can be reset by the CPU by writing 0 directly into the MC_PFCOUNT register.

PEL Reconstruction Unit: The PEL Reconstruction Unit (PRU) is responsible for combining the pre-fetched reference macroblocks with the corresponding 8x8 IDCT blocks to produce motion compensated macroblocks in the output macroblock buffers. This operation is driven by information within the macroblock header information that is delivered via a FIFO from the VLD unit. The PRU reconstruction engine applies bilinear interpolation on the pre-fetched reference data blocks as needed. It is also responsible for 2:1 compression of the IDCT block in half-resolution mode. It stores the reconstructed macroblock into an internal output buffer where it is written to main memory by the Storage Unit (SU). This SU is double-buffered and each buffer is capable of holding 32x24 8-bit unsigned values, or two 16x24 reconstructed macroblocks in full-resolution mode or up to four 8x24 reconstructed macroblocks in half_resolution mode.

The PRU continuously moves completed macroblocks to main memory as the SU buffers fill. A flush operation occurs if the current macroblock is at the end of the current row, or if a flush command is received.

Storage unit: It is the function of the MC to store the reconstructed macroblocks into main memory. The MC computes the physical main memory addresses to store the macroblocks based on the base address of the destination buffer and MC_LINE_SIZE parameters programmed into the control registers.

After the macroblocks are successfully stored into SDRAM, the MC writes the macroblock position of the last macroblock stored into the state_mb_col and state_mb_row fields of the MC Status Register.

For bandwidth efficiency in DMA access, the MC makes special 2-D storage requests to write 32x4 bytes into main memory per request. To cover the cases in which the output buffer is not completely filled, however, special byte-enable masks are devised to write only the first 8x4, 16x4, or 24x4 bytes of the 32x4 bytes sent via DMA into main memory.

Since every MPEG2 picture slice starts and ends in the same row and each field is decoded sequentially starting from the first row, it is possible to divide the entire top or bottom field into multiple destination buffers. For instance, a complete picture field may be divided into 4 uniform quarters such that each quarter is stored in a different main-memory buffer. Software can take advantage of this multi-buffer approach to reduce the total memory buffer size to hold the decoded picture. Hence, for each new destination buffer, the CPU programs the starting macroblock row to which the first row of the destination buffer corresponds into the mb_row_offset register. The MC computes the main memory address of the macroblocks in the output buffer taking into account also the mb_offset parameter.

It is clear that mb_row should not be less than mb_row_offset. The CPU should check the validity of mb_row (through slice_start_code) for every new slice to detect any potential error in the decoder bit stream.

Remark: splitting a destination picture field into multiple buffers is done only when decoding a B picture. Since I and P pictures are used as reference, it is mandatory that the entire I or P picture field is stored in a single contiguous memory buffer and mb_row_offset should be 0 in these cases. Finally, splitting a destination picture field into multiple buffers is never done for any picture type when decoding a MPEG1 bit stream.

MC/VLD error recovery: During a software reset of the MPEG Video Decoder, all the MPEG units first enter an idle state as quickly as possible. As soon as all the MPEG units become idle, all of their internal states are reset synchronously. For the MC block, the MC_STATUS, MC_COMMAND, and MC_PFCOUNT control registers are also cleared after reset (while other MC control registers are unchanged). Then all the MPEG units resume normal operation after the CPU software over-writes the RESET command in the VLD_COMMAND register.

Remark: The first macroblock header sent to the MC after a software reset should be the first of a slice. The MC will not check for this condition, however. It is the responsibility of the software to make sure that this is the case when restarting the MPEG unit after a soft reset.

MPEG Software Reset: During a software reset of the MPEG Video Decoder, all the MPEG units first enter an idle state as quickly as possible. As soon as all the MPEG units become idle, all of their internal states are reset synchronously. For the MC block, the MC_STATUS, MC_COMMAND, and MC_PFCOUNT control registers are also cleared after reset (while other MC control registers are unchanged). Then all the MPEG units resume normal operation after the CPU software overwrites the RESET command in the VLD_COMMAND register.

MC flush: There are two situations under which the CPU needs to make sure that the MPEG2 pipe is empty with all pending macroblocks reconstructed and saved into main memory: a) When the VLD detects the start code of a new slice which starts in a new destination frame buffer, and b) When the VLD is stalled right before the start of an error packet, as controlled by the CPU. Under either situation, the CPU issues a flush command to the VLD/MC by setting the flush_cmd bit in the MC Command Register.

Upon completion of a flush command, the MC clears the MC_COMMAND register and sets the DONE_FLUSH bit in the VLD_MC_STATUS register. The VLD also interrupts the CPU if the DONE_FLUSH_IE bit (in VLD_IE register) is 1. From then on, the entire MPEG Video Decoder is empty and idle. The CPU determines the completion of the flush operation by reading the DONE_FLUSH status bit. The CPU acknowledges the flush-completion condition by writing 1 into the DONE_FLUSH status bit, which clears the bit and resumes normal MC operation. The CPU may also send a software reset command to reset the MPEG Video Decoder. Software reset is necessary to ensure proper MPEG2 operation after flushing the DMA data paths if the flush command is sent to handle a packet error.

MC timeout: Whenever the state_mb_col or state_mb_row fields of the MC_STATUS register are updated by the MC, or when the reset_mc_timeout signal transitions from high to low, the internal 15-bit mc_timeout_counter in the MC block is reset to N = mc_timeout_period*2048, where mc_timeout_period is a 4-bit value between 0 and 15 programmed into the MC_PICINFO0 register by the CPU. If the reset_mc_timeout signal is low, the mc_timeout_counter is decremented for every CPU clock cycle; otherwise count-down is stalled. When the mc_timeout_counter transitions from 1 to 0, the internal mc_timeout one-shot signal between the MC and VLD is triggered to indicate to the VLD that the MC has not been storing any output macroblocks for the last N clocks. After it reaches 0, the mc_timeout_counter is stuck at 0 until state_mb_col or state_mb_row is updated, or when the reset_mc_timeout signal transitions from high to low, or after a MPEG software reset operation. Notice that the timeout mechanism is disabled if mc_timeout_period equals to 0. The maximum timeout period is 30720 cycles.

If enabled by the CPU, the VLD interrupts the CPU in response to the mc_timeout signal if it is in the parsing mode.

Timeout does not occur if the MC is carrying out an ERRCON2 command (type 2 error concealment) or the VLD is in the idle state.

5.11.6 VLD register programming guidelines

5.11.6.1 VLD Status (VLD_MC_STATUS)

The VLD_MC_STATUS register contains current status information which is most pertinent to the normal operation of an MPEG video decode application. Writing a logic ‘1’ to any of the status bits other than bit-0 clears the corresponding bit. Writing a logic ‘0’ has no effect. Exception: Bit 0 (Command Done) is cleared only by issuing a new command. Writing a logic ‘1’ to bit zero of the status register results in undefined behavior of the VLD. Note that several status bits may be asserted simultaneously. Table 334 lists the function of each status fields. See Table 340 for details on the other status flags related to the motion compensation unit.

Table 334: VLD STATUS (R/W)

Name	Size (Bits)	Description
VLD Command Done	1	Logic ‘1’ indicates successful completion of current command. This bit is cleared by issuing a new command.
Start Code Detected	1	Logic ‘1’ indicates VLD encountered 0x000001 while executing current command. This bit is cleared by writing a logic ‘1’ to it.
Bitstream Error	1	Logic ‘1’ indicates VLD encountered an illegal Huffman code or an unexpected start code. See Section 5.11.11 for details on the error handling procedure. This bit is cleared by writing a logic ‘1’ to it.
DMA Input Done	1	Conditions for setting this bit depends on the value of the DMA_Input_Done field in the VLD_CTL register. See Table 335 and Section 5.11.9 for details. This bit is cleared by writing a logic ‘1’ to it.
DMA Macroblock Header Output Done	1	Logic ‘1’ indicates that the macroblock header DMA write transfer has completed. This is valid only when the “Write to memory” bit is set in the VLD_CTL register. This bit is cleared by writing a logic ‘1’ to it.
DMA Run/Level Output Done	1	Logic ‘1’ indicates that the Run/Level DMA write transfer has completed. This is valid only when the “Write to memory” bit is set in the VLD_CTL register. This bit is cleared by writing a logic ‘1’ to it.
RL overflow Error	1	Logic ‘1’ indicates Overflow of run/level values with in a block. See Section 5.11.11 for details on the error handling procedure. This bit is cleared by writing a logic ‘1’ to it.
Timeout	1	Logic ‘1’ indicates MPEG Video Decoder timed out. See Section ‘MC timeout’ for details on the timeout mechanism. This bit is cleared by writing a logic ‘1’ to it.

When an error occurs in the VLD or RL/IS, the corresponding error flag (Bitstream error or RL overflow error) is set and an interrupt is generated if corresponding bits in the VLD_IE register is set.

5.11.6.2 VLD Interrupt Enable (VLD_IE)

This VLD_IE read/write register allows the CPU to control the initiation of the interrupt for the corresponding bits in the VLD_MC_STATUS register. Writing a one to any of these bits in the VLD_IE register enables the interrupt for the corresponding bit in the status register.

5.11.6.3 VLD Control (VLD_CTL)

The “Write to memory” bit is for controlling the output path of the VLD. When this bit is set, the outputs of the VLD are transferred back to main memory buffers instead of being passed further down the MPEG pipeline. This gives some flexibility for software error concealment or for software MPEG decoding starting from the VLD output.

For the “Write to memory” bit to change from zero to one, a VLD command is first issued by the CPU to flush out the data currently remaining in the VLD output buffers, and the “slice start code strobe” bit is set to zero. Also, the “Write to memory” bit is not changed from 1 to 0 until the VLD input buffers to main memory are fully consumed.

When VLD detects a new slice start code in the bit-stream, it writes the lower 8-bits of the start code into the slice_start_code field of the VLD_CTL register before interrupting the CPU. When re-started, the VLD reads the slice_start_code from the VLD_CTL register and writes this value into bits 16-23 of the last word in the first mb_header and sets the mb_first bit to 1. To allow the CPU to switch bitstream at the slice level, CPU can write the desired slice start code and slice_start_code_strobe in the VLD control register. The value of ‘1’ in the slice_start_code_strobe causes the VLD to update the slice_start_code field with the given slice_start_code value. The other fields in the VLD_CTL register are not updated when the input data contains a value of ‘1’ in the slice_start_code_strobe field. The slice_start_code_strobe bit is always read as 0. The CPU should write the slice_start_code only when the VLD is not active.

In order to update the “Write to memory” and the DMA_INPUT_DONE_MODE fields, the slice_start_code_strobe bit value is set to ‘0’.

The use of the DMA_INPUT_DONE_MODE bit is described in Table 335.

Table 335: VLD Control (R/W)

Name	Size (Bits)	Description
DMA_input_done_mode	1	When this bit is ‘0’, VLD sets the DMA_INPUT_DONE flag (in VLD_MC_STATUS register) when the DMA_INP_CNT transitions from non-zero to zero. When this bit is ‘1’, the same flag is set only with the additional condition that both DMA input buffers are empty. The slice_start_code_strobe bit field should be set to ‘0’ in order to update this field).
slice_start_code	8	slice start code when the VLD is restarted; the slice_start_code_strobe bit field should be set to ‘1’ in order to update this field.
slice_start_code_strobe	1	When CPU writes 1 into this field, VLD copies the value of slice_start_code into its internal register. CPU should do this only when the VLD is stopped. This bit is always read as 0.

5.11.6.4 VLD DMA Current Read Address (VLD_INP_ADR) and VLD DMA Current Read Count (VLD_INP_CNT)

The CPU writes the main memory buffer address from which bitstream to be read by VLD in VLD_INP_ADR register. The number of bytes to be read by the VLD is updated by the CPU in the VLD_INP_CNT register.

The VLD unit uses two 64-byte buffers to store the input bitstream. The VLD reads the bitstream data from the main memory and updates the VLD_INP_ADR and the VLD_INP_CNT register. The content of the VLD_INP_ADR register reflects the next or the current fetch address of the bitstream data.

The VLD interrupts the CPU when it has consumed all the given bitstream data in the main memory (the DMA_INPUT_DONE condition). The value of the DMA_INPUT_DONE_MODE bit in the VLD_CTL register is used to select the condition for raising the DMA_INPUT_DONE flag.

The VLD input address is word (32-bit) aligned and the count value in number of bytes is also word aligned.

5.11.6.5 VLD DMA macroblock header current write address (VLD_MBH_ADR)

The CPU writes the main memory macroblock header buffer address in the VLD_MBH_ADR register in order to output the macroblock header data in main memory. This is valid only when the “Write to memory” bit is set in the VLD_CTL register. The VLD updates this address whenever data is transferred to main memory via the DMA logic. The address always represents the next write address of the macroblock header data. The macroblock header buffer address should be 64-byte aligned.

5.11.6.6 VLD DMA macroblock header current write count

The CPU writes the main memory macroblock header buffer size formatted as the number of 8-byte words into the VLD_MBH_CNT register in order to output the macroblock header data in main memory. The is valid only when the “Write to memory” bit is set in the VLD_CTL register. The VLD updates the buffer size whenever data is transferred to main memory via the DMA logic. The buffer size always represents the remaining empty buffer space.

Note that in MPEG2 when Macroblock Headers are written to main memory, they are written in groups of six 4-byte vectors (24 bytes).

5.11.6.7 VLD DMA run-level current write address (VLD_RL_ADR)

The CPU writes the main memory run-level pairs buffer address in the VLD_RL_ADR register in order to output the run-level pairs in main memory. This is valid only when the “Write to memory” bit is set in the VLD_CTL register. The VLD updates this address whenever data is transferred to main memory via the DMA logic. The address always represents the next write address of the run-level pairs. The macroblock header buffer address should be 32-bit aligned.

5.11.6.8 VLD DMA run-level current write count

The CPU writes the main memory run-level pair buffer size formatted as the number of words into the VLD_RL_CNT register in order to output the run-level pairs in main memory. This is valid only when the “Write to memory” bit is set in the VLD_CTL register. The VLD updates the buffer size whenever data is transferred to main memory via the DMA logic. The buffer size always represents the remaining empty buffer space.

5.11.6.9 VLD command (VLD_COMMAND)

This read/write register indicates the next action to be taken by the VLD. A command is sent to the VLD by writing the corresponding 4-bit command code in the COMMAND field of the VLD_COMMAND register. Some commands require an associated count value which resides in the least significant 8 bits of this register. The following 9 VLD commands are currently available:

- Shift the video bitstream by “count” bits (where “count” is given in the COUNT field of the register and should be between 0 and 15 inclusive).
- Parse “count” macroblocks (where “count” is given in the COUNT field of the register and should be between 0 and 255 inclusive).
- Search for the next start code.
- Reset the MPEG Video Decoder.
- Initialize the VLD (to clear the VLD_BIT_CNT register).
- Search for the specific 8-bit start code pattern given in the COUNT field of the register.
- Flush the VLD output buffers to main memory in ‘write to memory’ mode.
- Parse one row of macroblocks.
- Parse macroblocks continuously (COUNT field is unused, but cannot be programmed to 0).

Table 336: VLD Commands

Code	Command	Flags Set (after completion of the Command)	Description
0x1	Shift the bitstream by “count” bits	Command Done	VLD shifts the number of bits in its internal shift register. The shift register value is available in the VLD_SR register. The flag is reset by issuing the new command.
0x2	Parse for a given number of macroblocks	Command Done and/or Start Code Detected	VLD parses for a given number of macroblocks; however, if VLD encounters a start code, the parsing action will be terminated and VLD sets only the start code detected flag. If VLD parses the given number of macroblocks without encountering a start code, VLD will set the command done flag. The start code detected flag is reset by writing a ‘1’ value to the flag. The command done flag is reset by issuing the new command
0x3	Search for the next start code	Start Code Detected and Command Done	VLD search for a start code. The search code has 0x000001 prefix and additional 8-bit value; a 32-bit value with 0x000001 prefix. The start code detected flag is reset by writing a ‘1’ value to the flag. The command done flag is reset by issuing the new command
0x4	Reset the MPEG Video Decoder	Command Done	See Section 5.11.11.

Table 336: VLD Commands *…continued*

Code	Command	Flags Set (after completion of the Command)	Description
0x5	Initialize the VLD	None	The bit count register is initialized to zero. The initialization action is immediate without any delay.
0x6	Search for the given start code	Start Code Detected and Command Done	VLD search for a start code with a given 8-bit lsb of the 32-bit start code. The search code has 0x000001 prefix and the additional 8-bit value is given in the ‘count’ field of the VLD_COMMAND register. The start code detected flag is reset by writing a ‘1’ value to the flag. The command done flag is reset by issuing the new command
0x7	Parse one row of macroblocks	Start Code Detected	This command instructs the VLD to parse one complete row of macroblocks. If the row contains more than one slice, VLD parses the intermediate slice headers without CPU intervention, provided these slice headers have a 0 bit after the 5-bit quantizer_scale_code. If the VLD encounters a start code different from the start code of the current slice, or if the slice header has a 1 bit after the quantizer_scale_code, it sets the Start-Code-Detected flag and ends the operation. WARNING: The ‘Count’ field of the VLD_COMMAND register is still in effect as in the ‘Parse A Number Of Macroblocks’ Command: VLD stops and sets the Command-Done flag after ‘Count’ macroblock headers are parsed. ‘Count’ should be set to at least mb_width (number of macroblocks per row in the picture) to guarantee the entire row is parsed before the VLD stops.
0x8	Flush Write FIFOs	Command Done	The command is valid only when the “Write to memory” bit is set in the VLD_CTL register. The VLD flushed the remaining macroblock header data and the remaining run-level data to the main memory.
0x9	Parse macroblock continuously	Start Code Detected	This command instructs the VLD to parse macroblock continuously. The VLD parses the slice headers without CPU intervention if the slice header is valid and has a 0 bit after the 5-bit quantizer_scale_code. If VLD encounters a start code which is not a valid slice start code or has a 1 bit after the 5-bit quantizer_scale_code, it sets the Start-Code-Detected flag and ends the operation. The ’Count’ field of the VLD_COMMAND is unused for this command and these bits cannot be programmed to 0.

The CPU should wait for the VLD to halt before the next command can be issued. Note that there are several ways in which a command may complete. Only a successful completion is indicated by the command done bit in the status register. A command may complete unsuccessfully if a start code or an error is encountered before the requested number of items have been processed. Note also that the expiration of a DMA count does not constitute the completion of a command. When a DMA count expires the VLD is stalled as it waits for a new DMA to be initiated. It is not halted.

The ‘Initialize VLD’ command initializes the VLD_BIT_CNT register (the bit counter) to zero.

The ‘Search for the given start code’ command searches for the start code pattern given in the COUNT field (in the least significant 8 bits) of the VLD_COMMAND register. A valid MPEG start code is a 32-bit pattern with the upper 24-bits equal to 0x000001. For each start code encountered in the bitstream, the 8-bits following the 0x000001 pattern is compared against the given 8-bit start code pattern until a perfect match is obtained.

Once the given start-code is found, the VLD sets the COMMAND_DONE bit in the VLD_MC_STATUS register. At that point, the VLD interrupts the CPU if the corresponding bit in the VLD_IE register is also set.

Table 337: VLD Command Register

Name	Size (Bits)	Description
Count	8	For the ‘Shift Bitstream’ command, only the lower 4 bits are used; the upper 4 bits should be set to 0. All 8 bits are used for the ‘Parse macroblocks’ and ‘Search for given start code’ commands
Command	4	Command code of the VLD command to be executed

5.11.6.10 VLD Shift Register (VLD_SR)

This read only register is a shadow of the VLD’s operational shift register and it allows the CPU to access the bitstream through the VLD. Bits 0 through 15 are the current contents of the VLD shift register. Bit 16 to 31 are RESERVED and should be treated as undefined by the programmer.

5.11.6.11 VLD Quantizer Scale (VLD_QS)

This 5-bit register read/write register contains the quantization scale code to be output by the VLD until it is overridden by a macroblock quantizer scale code. The quantizer scale code is part of the macroblock header output.

5.11.6.12 VLD Picture Info (VLD_PI)

This 32-bit read/write register contains the picture layer information necessary for the VLD (and MC) to parse the macroblocks within that picture. Again, the values of each of these fields are determined by the appropriate standard (MPEG or MPEG2).

5.11.6.13 VLD Bit Count (VLD_BIT_CNT)

The number of bits consumed by the VLD is updated in the VLD_BIT_CNT register. VLD_BIT_CNT can be initialized to zero by issuing the ‘Initialize VLD’ command. It counts upward when bits are shifted out and consumed by the VLD. This counter wraps around after reaching the maximum value.

5.11.7 RL/ IQ registers

To ensure compatibility with future devices, any undefined register bits should be ignored when read, and written as zeroes.

5.11.7.1 Runlength Decoder Statistics Registers

The Runlength Decoder has a status register in the register space called RL_STAT. This is a read/write register with two fields. These fields are used only for performance analysis.

5.11.7.2 Total Symbol Count

This register counts the number of symbols (non-zero DCT coefficients) in multiples of 1024 in the decoded bitstream. The CPU is responsible for initializing, reading, and resetting this value.

5.11.7.3 Total_Coded_Block_Count

This register counts the number of coded blocks in multiples of 1024 in the current bitstream decoder process. The CPU is responsible for initializing, reading and resetting this value.

5.11.7.4 Extra Picture Info (Extra_Pic_Info)

There are a number of parameters that are used by the Runlength Decoder, Inverse Scan, Inverse Quantization and Inverse DCT. Bits 2 and 3 of the Extra_Pic_Info register indicate if the default or loaded quantizer matrices are being used by the Inverse Quantization block. The CPU sets the appropriate value for the Default_intra_quant_matrix and the Default_non_intra_quant_matrix fields. At the beginning of a picture, the CPU sets the value of ‘1’ for the Default_intra_quant_matrix and the Default_non_intra_quant_matrix fields.

5.11.8 Motion compensation registers

5.11.8.1 MC control registers

The MC_PICINFO0 and MC_PICINFO1 registers contain parameters which are extracted or derived from the decoding MPEG video bit stream at the picture layer and above (such as mb_width and picture_type). (Note that MC_PICINFO1 is the same as the VLD_PI register in the VLD block.) MC_PICINFO2 register contains parameters for error concealment operation and picture_stride, which can be different from the coded picture width if necessary. The rest of the control registers store the SDRAM base addresses of the forward, backward, and destination YUV field buffers.

Note that all SDRAM base addresses should be at least 16-byte aligned.

Remark: The allocation for all MPEG picture data (together) should be in chunks of 4 kB with 4 kB-aligned addresses.

5.11.8.2 MC command register

This register contains a single 3-bit command field through which the CPU instructs the MC to perform its operations. The definition of the command code and the corresponding operation is given in Table 338. The command field is automatically cleared by the MC once the last command is completed.

Table 338: Definition of MC command code

Name	Code	Description
cmd_done	000	If a command is issued previously, this indicates completion of the last requested operation.
flush_cmd	001	When the VLD is stopped (due to a start code for instance), the CPU may issue this command to flush the MPEG pipe. The MC continues to read and process macroblock headers from the mb_header_FIFO and will initiate an internal flush sequence when the FIFO becomes empty. The MC will be stalled and the DONE_FLUSH bit (in the VLD_MC_STATUS register) set after the last macroblock in the pipe is stored into SDRAM. The CPU will be interrupted also if the DONE_FLUSH_IE bit (in VLD_IE register) is 1. The CPU should acknowledge the DONE_FLUSH condition by writing 1 into the DONE_FLUSH status bit to resume the MC normal operation. The MC will clear the MC_COMMAND register once the flush operation is completed.
errcon1_cmd	010	When the VLD is stopped due to a new slice start code, the CPU issues this command to perform error concealment. Before restarting the VLD to parse the new slice, the CPU sets this command code and writes the desired macroblock position (row and column) to start error concealment into the err_mb_row and the err_mb_col registers. Once the VLD restarts parsing (from the beginning of the new slice), the MC first determines the starting macroblock position of the new slice based on the first macroblock header of the slice parsed by the VLD and performs error concealment from (err_mb_row, err_mb_col) to just before the starting position of the new slice. At that point, the MC clears the MC_COMMAND register, sets the DONE_ERRCON bit (in the VLD_MC_STATUS register), and waits in the idle state for the CPU acknowledgement of the condition. If DONE_ERRCON_IE (in VLD_IE register) is 1, the CPU will also be interrupted by the VLD when the DONE_ERRCON bit is set. The CPU acknowledges the DONE_ERRCON condition by writing 1 into the DONE_ERRCON status bit. The MC then resumes normal operation and continues the processing of the new slice. Note that error concealment is allowed to span multiple macroblock rows.
errcon2_cmd	100	This command is similar to errcon1_cmd, except that CPU needs to specify both the starting macroblock position (in err_mb_row and err_mb_col) and the ending macroblock position (in err_end_mb_row and err_end_mb_col) for the desired error concealment operation. In this case, the MC performs error concealment from the given starting position to the macroblock just before the given ending macroblock position. If error concealment is up to and include the last macroblock of the current picture, the user should specify err_end_mb_col as 0 and err_end_mb_row as mb_height for frame pictures or the value (mb_height>>1) for field pictures.

5.11.8.3 MC status register

The MC Status Register is a read-only register which stores the status information listed in Table 339. It is updated continuously while the MC is in operation and is normally read by the CPU when the block is completely idle and the pipe empty. The (state_mb_col, state_mb_row) status indicates the macroblock position of the last macroblock successfully reconstructed and stored into SDRAM. start_mb_col is the macroblock column position of the first macroblock of the last slice (this information allows the CPU to detect, for example, if there are any “holes” in the decoding slice).

Table 339: MC status register (read only)

Name	Bits	Description
state_mb_col	8	Macroblock column position in the decoding picture where the last reconstructed macroblock is stored in SDRAM
state_mb_row	8	Macroblock row position in the decoding picture where the last reconstructed macroblock is stored in SDRAM
start_mb_col	8	Macroblock column position in the decoding picture where the current slice starts

5.11.8.4 MC_PFCOUNT

The MC_PFCOUNT register contains two counters, namely, count_24x4 and count_24x5, which record respectively the total number of 24x4 and 24x5 2-D memory fetches generated by the Pre-Fetch Unit. The CPU can reset these counters by writing 0 into the register. These values reflect the memory bandwidth utilization for motion compensation and allow the CPU to derive the contingency strategy to assure graceful degradation in picture quality when the memory bandwidth utilization is at the limit.

5.11.8.5 Line size

This register contains the size in bytes of the Y component for one video line stored in memory. For example, if the system is receiving and decoding a 1920 by 1080 pixels MPEG2 bitstream in full resolution mode, then MC_LINE_SIZE should contain 1920.

Some limitations apply to the value MC_LINE_SIZE can contain:

MC_LINE_SIZE cannot be 512, 680, 1024, 1360, 1368, 1536 or 2040.

Recommended values are 320, 352, 368, 640, 704, 720, 960, 1280, 1440 or 1920.

If the desired MC_LINE_SIZE value is not in this recommended values list, it is recommended to use the higher value and lose some memory space in order to use one of the recommended values listed above.

This register should be set before the motion compensation unit is enabled. The stored value has to be 8 bytes aligned (i.e. the three least significant bits are forced to 0s).

5.11.8.6 VLD_MC_STATUS

The MC updates the fields shown in Table 340in the VLD_MC_STATUS register. For a complete description of this status register see Section 5.11.9.

When the MC detects a bit-stream error (see Section on page 620), it writes the corresponding error code into the mc_error_flags field, which causes an interrupt to the CPU if the MC ERROR interrupt is enabled. The DONE_FLUSH bit is set when the MC completes a flush command. If the DONE_FLUSH_IE bit (in VLD_IE register) is set, a VLD interrupt is sent to CPU as well. The DONE_ERRCON bit is set when the MC completes an error concealment operation. The CPU also receives a VLD interrupt if the DONE_ERRCON_IE bit (in VLD_IE register) is set.

Table 340: MC-updated fields in VLD_MC_STATUS register

Name	Bits	Description
mc_error_flags	6	MC error code. See Table 332.
DONE_FLUSH	1	A ‘1’ signals the completion of the last flush command. The CPU writes 1 to this field to acknowledge the condition and to clear this flag.
DONE_ERRCON	1	A ‘1’ signals the completion of the last error concealment command. The CPU writes 1 to this field to acknowledge the condition and to clear this flag

5.11.9 VLD operations

After initialization, the CPU controls the VLD through the VLD command register. There are currently nine commands supported by the VLD:

- Shift the bit stream by some number of bits.
- Parse some number of macroblocks.
- Search for the next start code.
- Reset the MPEG Video Decoder.
- Initialize the VLD.

- Search for the given start code.
- Parse one row of macroblocks.
- Flush VLD output buffers in ‘write to memory mode’.
- Parse macroblocks without stopping at the slice header.

The normal mode of operation is for the CPU to request the VLD to parse some number of macroblocks. Once the VLD has begun parsing macroblocks it may stop for any one of the following reasons:

- The command was completed with no exceptions.
- A start code was detected.
- An error was encountered in the bit stream.
- The VLD input DMA completed and the VLD is stalled waiting for more data.
- One of the output DMA for RL or HDR processing has completed.

Under normal circumstances, the CPU is interrupted whenever the VLD halts.

Consider the case in which the VLD has encountered a start code. At this point, the VLD halts and sets the status flag which indicates that a start code has been detected. This flag generates an interrupt to the CPU. Upon entering the interrupt service routine, the CPU reads the VLD status register to determine the source of the interrupt. Once it has been determined that a start code has been encountered, the CPU reads 8 bits from the VLD shift register to determine the type of start code that has been encountered. If a slice start code has been encountered, the CPU reads from the shift register the slice quantization scale code and any extra slice information (from the slice header). The slice quantization scale code is then written back to the VLD_QS register. Before exiting the interrupt service routine, the CPU clears the start code detected status bit in the status register and issues a new command to process the remaining macroblocks.

5.11.9.1 VLD input

The VLD reads the video bitstream from the main memory and performs the variable length decoding process. The CPU writes the main memory buffer address from which bitstream to be read by VLD in VLD_INP_ADR register. The number of bytes to be read by the VLD is updated by the CPU in the VLD_INP_CNT register.

The VLD unit uses two 64-byte buffers to store the input bitstream. The VLD reads the bitstream data from the main memory and updates the VLD_INP_ADR and the VLD_INP_CNT register. The content of the VLD_INP_ADR register reflects the next fetch address of the bitstream data. The content of the VLD_INP_CNT register reflects the number of bytes to be read from the main memory. When the number of bytes to be read from the main memory transitions from non-zero to zero, the DMA_INPUT_DONE flag in the VLD_MC_STATUS is set. An interrupt is sent to the CPU also if the corresponding interrupt enable bit in the VLD_IE register is set. The CPU will then provide the new bitstream buffer address and the number of bytes in the bitstream buffer to the VLD.

The VLD unit also updates a bit counter in the VLD_BIT_CNT register to keep track of the number of bits consumed in the decoding process. The bit counter is updated only after a successful decoding of a symbol. The CPU can read this bit counter from the VLD_BIT_CNT register. This register can be initialized to zero by sending the ‘Initialize VLD’ command.

5.11.9.2 VLD output

The output of the VLD is sent to the downstream pipeline blocks. Macroblock headers are passed to the MC block and the run-level encoded DCT coefficients to the RL/IS block for further decoding of the macroblocks.

If the "write_to_mem" option is enabled, then the run-level pairs and macroblock headers are written to main memory, and the downstream pipeline blocks are disabled.

5.11.9.3 Restart the VLD parsing

The restart the parsing command at the last decoded symbol position is done through the use of the VLD_BIT_CNT register to count the number of bits from the main memory VLD buffer that have been consumed. The VLD_BIT_CNT register can be initialized to zero by issuing the VLD_INIT command. It counts up when bits are shifted out of the VLD. This counter wraps around after reaching the maximum value.

5.11.10 Interrupt

The Vector-interrupt source number for the MPEG Video Decoder is 14.

5.11.11 Error handling

The VLD can generate two types of errors and the MC can generate six types of errors. The VLD_MC_STATUS register has two VLD error flags and the six motion compensation error flags.The VLD errors are 1) bitstream parsing error and 2) run-level overflow error. See Table 341 for the details on the VLD error handling procedure when full MPEG Video Decoder is in operation. i.e., when ‘write-to-memory’ bit is not set in the VLD_CTL register.

5.11.11.1 Unexpected start code

When VLD encounters an unexpected start code, VLD sets the ‘Start Code Detected’ and ‘Bitstream Error’ flags (in VLD_MC_STATUS register). The start code value is left in the shift register (VLD_SR). VLD interrupts the CPU, if one of the corresponding interrupt bits in the VLD_IE register is enabled.

Table 341: VLD Error Handling

Cycle No.	Action	Remarks
i	VLD sets the appropriate error bit in the VLD_MC_STAUS register	The vld_mc_error signal is formed by OR’ing together all of the error bits in the VLD_MC_STATUS register is the Hence any MC error also drives the vld_mc_error signal high and the following error handling steps still apply.
i to j	When the vld_mc_error signal is high, VLD completes any pending control or memory hwy. transactions. The valid data in the DMA output buffers, in ‘write-to-memory’ mode, will be flushed to the main memory. Then VLD asserts the vld_ready_to_reset signal and waits for the CPU to reset the MPEG Video Decoder.	Any DMA transactions, once started, will not be aborted in the middle.

Table 341: VLD Error Handling *...continued*

Cycle No.	Action	Remarks
i to k	MC after receiving the vld_mc_error signal, will complete any pending control or memory hwy. transactions. Then MC asserts the mc_ready_to_reset signal and waits for the CPU to reset the MPEG Video Decoder	Any DMA transactions, once started, will not be aborted in the middle.
k	If (vld_ready_to_reset & mc_ready_to_reset) then VLD interrupts the CPU.	Assumes k > j; otherwise it is jth cycle. The corresponding IE bit in the VLD_IE register is '1' for the VLD to raise the interrupt.
l	CPU will perform the software reset.	See Table 327 for the software reset procedure.

The error handling in the VLD is synchronized with the error handling in the MC block as well. Synchronization of error handling in VLD and MC is achieved through the use of 3 interface signals between the MC and the VLD: the vld_mc_error, the vld_ready_to_reset, and the mc_ready_to_reset signal. These internal signals are active high and are cleared after hardware or software reset.

As soon as it encounters an error, the VLD sets the appropriate error flag in the VLD_MC_STATUS register. The vld_mc_error signal is obtained by OR'ing together all the MC and VLD error flags. Hence any MC error also drives the vld_mc_error signal high and the subsequent VLD error handling steps apply as well. As soon as the vld_mc_error signal is high, the VLD completes any outstanding DMA (control or main memory) transactions and then asserts the vld_ready_to_reset signal to the MC. Meanwhile MC also performs similar error handling steps when the vld_mc_error_signal goes high and eventually asserts the mc_ready_to_reset signal to the VLD. When the vld_ready_to_reset and mc_ready_to_reset signals are both high, the VLD interrupts the CPU if any of the error interrupt enable bits (in the VLD_IE register) is set. Then the CPU can issue a 'Flush the VLD output buffers' command to optionally empty the VLD output buffer before sending a software reset command to the VLD.

5.11.11.2 MC flush

The MPEG Video Decoder stores the intermediate data at various stages within the MPEG Video Decoder. In order to terminate the decoding process at a particular point of the bitstream, the CPU issues the flush_cmd by setting the flush_cmd bit in the MC_COMMAND register. The flush_cmd is issued in two cases:

1. The bitstream contains error bits for a known amount of bytes and would like to terminate the decoding at a particular byte-offset of the bitstream buffer, and
2. When CPU decides to switch the bitstream at the start of a new slice-start-code in a new row.

The CPU sets the DMA_input_done_mode bit to '1' in the VLD_CTL register for the first case.

The flush_cmd is issued when the VLD stops after interrupting the CPU for the dma_input_done reason in the first case, and when the VLD stops after interrupting the CPU for the start_code_detected reason in the second case. When VLD detects the assertion of the flush_cmd bit in the MC_COMMAND (refer to Table 334) register, the RL/IS, IQ, and IDCT blocks continue their operations until they have exhausted their input data buffers and go into their respective idle states. Then the VLD unit raises the vld_mc_done_flush signal to the MC. The MC block processes the flush command and sets the DONE_FLUSH bit in the VLD_MC_STATUS register when the flush operation is completed. The VLD unit de-asserts the vld_mc_done_flush signal only after the CPU clears the DONE_FLUSH bit in the VLD_MC_STATUS register. Any status bit can be cleared by writing a ‘1’ into the status bit.

There may be some partial data in the RL/IS-IQ-IDCT chain and in the macroblock header buffers in the first case and there is not be any partial data in the RL/IS-IQ-IDCT chain or in the macroblock header buffers in the second case. The CPU will issue a ‘Reset MPEG Video Decoder’ command in when flush is completed in the first case. Software reset is optional in the second case.

5.11.11.3 Timeout

The timeout mechanism is used by the MPEG Video Decoder in order to interrupt the CPU when the MPEG Video Decoder does not make the expected progress in decoding the given bitstream. The expected MPEG Video Decoder performance is given in the MC_PICINFO0 register.

The current implementation of this timeout mechanism relies on two interface signals, mc_timeout and reset_mc_timeout, between the VLD and the MC and a timeout down-counter in the MC block. The reset_mc_timeout signal is obtained by OR’ing together all the bits in the VLD_MC_STATUS register. As long as reset_mc_timeout is low, the timeout counter continues to decrement once per cycle until it reaches 0. The timeout counter stops if its current value is 0 or when reset_mc_timeout is high. At the falling edge of reset_mc_timeout, the timeout counter is reset to a number N according to the value specified in the 4-bit mc_time_period field of the MC_PICINFO0 register, where N = 2048*mc_timeout_period. The mc_timeout_period is set to zero in the ‘Write-to-memory’ mode to effectively disable the timeout mechanism.

MC asserts the mc_timeout signal to the VLD for one cycle when no progress is detected in the MC. When the mc_timeout signal is asserted, VLD sets the TIMEOUT bit in the VLD_STATUS register only if the VLD is also in a parsing mode (that is, when the VLD is in the middle of executing a parse command). When the TIMEOUT bit is set, VLD also raises an interrupt to the CPU, if the corresponding interrupt enable bit is set in the VLD_IE register.

Timeout does not occur when the MC is carrying out an ERRCON2 command (type 2 error concealment) for the VLD is idle in this case.

5.11.12 Coefficient Selection for Half Resolution Mode

The MPEG Video Decoder supports a two-to-one compression mode to reduce the main memory buffering requirement for video decoding by 50%. This mode is enabled by setting the half_resolution_mode bit in the MC_PICINFO1 register. If this mode is not enabled then the coefficients from the VLD are to be selected. The half resolution mode involves frequency domain filtering in order to improve the quality of the video image. The filtering algorithm either selects a coefficient value generated from the VLD or it selects a zero coefficient value. The coefficient selection algorithm is programmable and operates on an 8 x 8 block level. The processor should write 32-bit values in the MP_IQ_SEL_0 and the MP_IQ_SEL_1 to select either a coefficient value from the VLD or a zero coefficient value.

5.11.13 Register descriptions

5.11.13.1 Register summary

Table 342: Register summary

Offset	Symbol	Description
0x0	VLD_COMMAND	Variable length decoder command
0x4	VLD_SR	VLD shift register (shadow)
0x8	VLD_QS	Quantization scale code to be output by the VLD
0xC	VLD_PI	VLD picture information
0x10	VLD_MC_STATUS	VLD and MC status register
0x14	VLD_IE	VLD interrupt enable
0x18	VLD_CTL	VLD control register
0x1C	VLD_INP_ADR	VLD input memory address
0x20	VLD_INP_CNT	Gives the number of bytes to be read from main memory.
0x24	VLD_MBH_ADR	VLD macroblock header write back address
0x28	VLD_MBH_CNT	VLD macroblock header write back count
0x2C	VLD_RL_ADR	VLD run-level write back address
0x30	VLD_RL_CNT	VLD run-level write back count
0x34	VLD_BIT_CNT	Gives the number of bits consumed by the VLD.
0x38	LINE_SIZE	Contains the size in bytes of the Y component for one video line stored in memory
0x40	W_TBL0_W0	Quantization matrix table 0 W0 through W15
0x44	W_TBL0_W1	
0x48	W_TBL0_W2	
0x4C	W_TBL0_W3	
0x50	W_TBL0_W4	
0x54	W_TBL0_W5	
0x58	W_TBL0_W6	
0x5C	W_TBL0_W7	
0x60	W_TBL0_W8	
0x64	W_TBL0_W9	
0x68	W_TBL0_W10	
0x6C	W_TBL0_W11	
0x70	W_TBL0_W12	
0x74	W_TBL0_W13	
0x78	W_TBL0_W14	
0x7C	W_TBL0_W15	

Table 342: Register summary *…continued*

Offset	Symbol	Description
0x80	W_TBL1_W0	Quantization matrix table 1 W0 through W15
0x84	W_TBL1_W1	
0x88	W_TBL1_W2	
0x8C	W_TBL1_W3	
0x90	W_TBL1_W4	
0x94	W_TBL1_W5	
0x98	W_TBL1_W6	
0x9C	W_TBL1_W7	
0xA0	W_TBL1_W8	
0xA4	W_TBL1_W9	
0xA8	W_TBL1_W10	
0xAC	W_TBL1_W11	
0xB0	W_TBL1_W12	
0xB4	W_TBL1_W13	
0xB8	W_TBL1_W14	
0xBC	W_TBL1_W15	
0xC0	Extra_Pic_Info	Extra picture information
0xC4	RL_STATS	Run Length Statistics
0xC8	MP_IQ_SEL_0	Inverse Quantization Select 0
0xCC	MP_IQ_SEL_1	Inverse Quantization Select 1
0x200	MC_PICINFO0	Motion Control Picture Information 0
0x208	MC_PICINFO2	Motion Control Picture Information 2
0x20C	MC_FREFY0	MC Forward Reference base address for the Y component of the top field
0x210	MC_FREFY1	MC Forward Reference base address for the Y component of the bottom field
0x214	MC_FREFUV0	MC Forward Reference base address for the UV component of the top field
0x218	MC_FREFUV1	MC Forward Reference base address for the UV component of the bottom field
0x21C	MC_BREFY0	MC Backward Reference base address for the Y component of the top field
0x220	MC_BREFY1	MC Backward Reference base address for the Y component of the bottom field
0x224	MC_BREFUV0	MC Backward Reference base address for the UV component of the top field
0x228	MC_BREFUV1	MC Backward Reference base address for the UV component of the bottom field
0x22C	MC_DESTY0	MC Destination Reference base address for the Y component of the top field
0x230	MC_DESTY1	MC Destination Reference base address for the Y component of the bottom field
0x234	MC_DESTUV0	MC Destination Reference base address for the UV component of the top field

Table 342: Register summary *…continued*

Offset	Symbol	Description
0x238	MC_DESTUV1	MC Destination Reference base address for the UV component of the bottom field
0x23C	MC_COMMAND	MC Command
0x240	MC_PFCOUNT	Records the total number of 2D memory fetches generated
0x244	MC_STATUS	MC Status
0xFF4	VMPG_POWER_DOWN	Power Down Register
0xFFC	VMPG_MODULE_ID	Module Identification Register

5.11.13.2 Register table

Table 343: MPEG video decoder registers

Bit	Symbol	Access	Value	Description
Offset 0x0 - VLD_COMMAND				
31:12	Reserved[19:0]	R	0x00000	
11:8	Command[3:0]	R/W	0x0	Command code of the VLD command to be executed 1: Shift Bitstream by Shift Count bits 10: Parse Macroblock 11: Search for next Start Code 100: Reset MPEG Video Decoder 101: Initialize VLD 110: Search for the Start Code given in bits [7:0]. 111: Parse Macroblock Row 1000: Flush Write FIFOs 1001: Parse Macroblock Long
7:0	Mblock_shift_count_or_start_code[7:0]	R/W	0x00	For the Shift Bitstream command, only the lower 4 bits are used; the upper 4 bits should be set to 0. All 8 bits are used for the Parse macroblocks and Search for given start code commands. Note that when Command = 0x9 (Parse Macroblock Long), these bits cannot be programmed to 0.
Offset 0x4 - VLD_SR				
31:16	Reserved[15:0]	R	0xXXXX	
15:0	Shift_register[15:0]	R	0xXXXX	This read-only register is a shadow of the VLDs operational shift register and it allows the DSPCPU to access the bitstream through the VLD. Bits 15 through 0 are the current contents of the VLD shift register
Offset 0x8 - VLD_QS				
31:5	Reserved[26:0]	R	0x0000000	
4:0	Quant_scale[4:0]	R/W	0xXX	This 5-bit register read/write register contains the quantization scale code to be output by the VLD until it is overridden by a macroblock quantizer scale code. The quantizer scale code is part of the macroblock header output.
Offset 0xC - VLD_PI				

Table 343: MPEG video decoder registers *…continued*

Bit	Symbol	Access	Value	Description
31:28	Vertical_back_rsize[3:0]	R/W	0xX	Number of bits per backward vectical motion vector that are residual in the picture
27:24	Horizontal_back_rsize[3:0]	R/W	0xX	Number of bits per backward horizontal motion vector that are residual in the picture
23:20	Vertical_for_rsize[3:0]	R/W	0xX	Number of bits per forward vertical motion vector that are residual in the picture
19:16	Horizontal_for_rsize[3:0]	R/W	0xX	Number of bits per forward horizontal motion vector that are residual in the picture
15	Reserved	R	0x0	
14	No_backward	R/W	0xX	0: keeps backward prediction in all macroblocks with bi-directional motion prediction. 1: drops backward prediction in all macroblocks with bi-directional motion prediction.
13	Mpeg2mode	R/W	0xX	0: indicates if the current sequence is MPEG1(for error checking only) 1: indicates if the current sequence is MPEG2
12	No_uv_yhalf	R/W	0xX	0: keeps vertical half-pel flag for UV component in all macroblocks of a B picture. 1: drops vertical half-pel flag for UV component in all macroblocks of a B picture.
11	No_y_yhalf	R/W	0xX	0: keeps vertical half-pel flag for Y component in all macroblocks of a B picture. 1: drops vertical half-pel flag for Y component in all macroblocks of a B picture.
10	Top_field_first	R/W	0xX	0: indicates if the bottom of a frame picture is captured earlier than the other field. 1: indicates if the top of a frame picture is captured earlier than the other field.
9	Full_pel_backward	R/W	0xX	0: indicates backward motion vectors in a picture with half pel units 1: indicates backward motion vectors in a picture with full pel units.

Table 343: MPEG video decoder registers *…continued*

Bit	Symbol	Access	Value	Description
8	Full_pel_forward	R/W	0xX	0: indicates forward motion vectors in a picture with half pel units 1: indicates forward motion vectors in a picture with full pel units.
7	half_resolution_mode	R/W	0xX	1 = images are stored in memory with 1/2 the normal horizontal resolution. 0 = images are stored in memory at the normal full horizontal resolution.
6	Mv_concealment	R/W	0xX	0: indicates forward motion vectors are not coded in all intra macroblock headers of a picture. 1: indicates forward motion vectors are coded in all intra macroblock headers of a picture.
5	Intra_vlc	R/W	0xX	Use DCT table zero (intra_vlc = 0)or one (intra_vlc = 1)
4	Frame_prediction_frame_dct	R/W	0xX	If 1, motion_type = FRAME and dct_type = 0. If 0, motion_type and dct_type follow the decoded values in the mb_header from the VLD. CPU should set it to 0 for Field Pictures and to 1 for MPEG.
3:2	Picture_structure[1:0]	R/W	0xX	0: reserved. 1: top-field 10: bottom-field 11: frame picture
1:0	Picture_type[1:0]	R/W	0xX	0: D (MPEG1 only) 1: I 10: P 11: B
Offset 0x10 - VLD_MC_STATUS				
31:24	Reserved[7:0]	R	0x00	
23	Done_flush	R/W	0x0	1 signals the completion of the last mc flush command. CPU writes 1 into this field to acknowledge the condition and to clear this flag.
22	Eor_errcon	R/W	0x0	1 signals the completion of the last error concealment command. CPU writes 1 into this field to acknowledge the condition and to clear this flag
21:16	Mc_error_flags[5:0]	R/W	0x00	MC error code.These bits are cleared by writing a logic 1 to each bit.
15:8	Reserved2[7:0]	R/W	0x00	

Table 343: MPEG video decoder registers *…continued*

Bit	Symbol	Access	Value	Description
7	Timeout	R/W	0x0	Logic 1 indicates MPEG Video Decoder timed out. Refer to <Hyperlink>Section 3.10.9 on page 31</Hyperlink>for details on the timeout mechanism. This bit is cleared by writing a logic 1 to it.
6	Rl_overflow	R/W	0x0	Logic 1 indicates Overflow of run/level values with in a block. Refer to <Hyperlink>Section 3.8 on page 20</Hyperlink>for details on the error handling procedure. This bit is cleared by writing a logic 1 to it.
5	Dma_rl_output_done	R/W	0x0	Logic 1 indicates that the Run Length data FIFO has been written to main memory. This bit is cleared by writing a logic 1 to it.
4	Dma_header_output_done	R/W	0x0	Logic 1 indicates that the Run Length data FIFO has been written to main memory. This bit is cleared by writing a logic 1 to it.
3	Dma_input_done	R/W	0x0	Conditions for setting this bit depends on the value of the DMA_Input_Done field in the VLD_CTL register. Refer to and <Hyperlink>Section 3.6 on page 18</Hyperlink>for details. This bit is cleared by writing a logic 1 to it.
2	Bitstream_error	R/W	0x0	Logic 1 indicates VLD encountered an illegal Huffman code or an unexpected start code. Refer to <Hyperlink>Section 3.8</Hyperlink>for details on the error handling procedure. This bit is cleared by writing a logic 1 to it.
1	Start_code_detected	R/W	0x0	Logic 1 indicates VLD encountered 0x000001 while executing current command. This bit is cleared by writing a logic 1 to it.
0	Vld_command_done	R/W	0x0	Logic 1 indicates successful completion of current command. This bit is cleared by issuing a new command.
Offset 0x14 - VLD_IE				
31:24	Reserved[7:0]	R/W	0x00	
23	Done_flush_ie	R/W	0x0	Each of these bits enables the matching bit from the status reg 0x010 to issue an IR to the CPU.
22	Eor_errcon_ie	R/W	0x0	Each of these bits enables the matching bit from the status reg 0x010 to issue an IR to the CPU.
21:16	Mc_int_enables[5:0]	R/W	0x00	Each of these bits enables the matching bit from the status reg 0x010 to issue an IR to the CPU.
15:8	Reserved2[7:0]	R/W	0x00	-
7:0	Vld_int_enables[7:0]	R/W	0x00	Each of these bits enables the matching bit from the status reg 0x010 to issue an IR to the CPU.
Offset 0x18 - VLD_CTL				
31:17	Reserved[14:0]	R/W	0x0000	
16	Slice_start_code_strobe	R/W	0x0	When CPU writes 1 into this field, VLD copies the value of slice_start_code into its internal register. CPU should do this only when the VLD is stopped. This bit is always read as 0

Table 343: MPEG video decoder registers *…continued*

Bit	Symbol	Access	Value	Description
15:8	Slice_start_code[7:0]	R/W	0x00	Slice start code when the VLD is restarted; the slice_start_code_strobe bit field must be set to 1 in order to update this field
7:3	Reserved2[4:0]	R/W	0x00	
2	Dma_input_done_mode	R/W	0x0	When this bit is 0, VLD sets the DMA_INPUT_DONE flag (in VLD_MC_STATUS register) when the DMA_INP_CNT transitions from non-zero to zero. When this bit is 1, the same flag is set only with the additional condition that both highway input buffers are empty. The slice_start_code_strobe bit field must be set to 0 in order to update this field)
1	Write_to_memory	R/W	0x0	The slice_start_code_strobe bit field must be set to 0 in order to update this field. 0: Run Length and Macroblock Header codes are not written back to main memory. 1: Run Length and Macroblock Header codes are written to main memory, using the VLD_RL_ADR, VLD_RL_CNT, VLD_MBH_ADR, VLD_MBH_CNT registers to define the memory areas.
0	Little_endian	R/W	0x1	This bit only affects the Run Length and Macroblock Header codes that are written to main memory, when VLD_CTL[1] is set. The slice_start_code_strobe bit field must be set to 0 in order to update this field. 0: Big Endian 1: Little Endian
Offset 0x1C - VLD_INP_ADR				
31:0	Vld_input_memory_address[31:0]	R/W	0x00000000	Memory address from which VLD is reading (updated when DMA read transfer is completed) needs to be 32-bit word aligned.
Offset 0x20 - VLD_INP_CNT				
31:15	Reserved[16:0]	R	0x00000	
14:0	Vld_input_count[14:0]	R/W	0x0000	Number of bytes to be read from main memory
Offset 0x24 - VLD_MBH_ADR				
31:0	Mb_header_memory_address[31:0]	R/W	0xXXXXXXXX	Memory address to which the VLD writes macroblock headers when VLD_CTL[1] is set. Needs to be 32-bit word aligned.
Offset 0x28 - VLD_MBH_CNT				
31:15	Reserved[16:0]	R	0xXXXXX	
14:0	Mb_header_count[14:0]	R/W	0xXXXX	Number of MB Header bytes to be written to main memory when VLD_CTL[1] is set.
Offset 0x2C - VLD_RL_ADR				
31:0	Vld_rl_writeback_address[31:0]	R/W	0xXXXXXXXX	Memory address to which the VLD writes Run Length data when VLD_CTL[1] is set. Needs to be 32-bit word aligned.

Table 343: MPEG video decoder registers *…continued*

Bit	Symbol	Access	Value	Description
Offset 0x30 - VLD_RL_CNT				
31:15	Reserved[16:0]	R	0xXXXXX	
14:0	Vld_rl_count[14:0]	R/W	0xXXXX	Number of RL bytes to be written to main memory when VLD_CTL[1] is set.
Offset 0x34 - VLD_BIT_CNT				
31:18	Reserved[13:0]	R/W	0x0000	
17:0	Actual_bit_count[17:0]	R	0x00000	Bits consumed
Offset 0x38 - LINE_SIZE				
31:13	Reserved[18:0]	R	0x00000	
12:3	Line_size_x_16[9:0]	R/W	0x000	The line size is a multiple of 16.
2:0	Line_size_zero[2:0]	R/W	0x0	They are always 0.
Offset 0x40 - W_TBL0_W0				
31:24	W_0[0:3]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 040 to 07C shows either the default_intra_matrix or the downloaded_intra_matrix.
23:16	W_0[0:2]	R/W	0xXX	
15:8	W_0[0:1]	R/W	0xXX	
7:0	W_0[0:0]	R/W	0xXX	
Offset 0x44 - W_TBL0_W1				
31:24	W_0[0:7]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 040 to 07C shows either the default_intra_matrix or the downloaded_intra_matrix.
23:16	W_0[0:6]	R/W	0xXX	
15:8	W_0[0:5]	R/W	0xXX	
7:0	W_0[0:4]	R/W	0xXX	
Offset 0x48 - W_TBL0_W2				
31:24	W_0[1:3]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 040 to 07C shows either the default_intra_matrix or the downloaded_intra_matrix.
23:16	W_0[1:2]	R/W	0xXX	
15:8	W_0[1:1]	R/W	0xXX	
7:0	W_0[1:0]	R/W	0xXX	
Offset 0x4C - W_TBL0_W3				

Table 343: MPEG video decoder registers *…continued*

Bit	Symbol	Access	Value	Description
31:24	W_0[1:7]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 040 to 07C shows either the default_intra_matrix or the downloaded_intra_matrix.
23:16	W_0[1:6]	R/W	0xXX	
15:8	W_0[1:5]	R/W	0xXX	
7:0	W_0[1:4]	R/W	0xXX	
Offset 0x50 - W_TBL0_W4				
31:24	W_0[2:3]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 040 to 07C shows either the default_intra_matrix or the downloaded_intra_matrix.
23:16	W_0[2:2]	R/W	0xXX	
15:8	W_0[2:1]	R/W	0xXX	
7:0	W_0[2:0]	R/W	0xXX	
Offset 0x54 - W_TBL0_W5				
31:24	W_0[2:7]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 040 to 07C shows either the default_intra_matrix or the downloaded_intra_matrix.
23:16	W_0[2:6]	R/W	0xXX	
15:8	W_0[2:5]	R/W	0xXX	
7:0	W_0[2:4]	R/W	0xXX	
Offset 0x58 - W_TBL0_W6				
31:24	W_0[3:3]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 040 to 07C shows either the default_intra_matrix or the downloaded_intra_matrix.
23:16	W_0[3:2]	R/W	0xXX	
15:8	W_0[3:1]	R/W	0xXX	
7:0	W_0[3:0]	R/W	0xXX	
Offset 0x5C - W_TBL0_W7				
31:24	W_0[3:7]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 040 to 07C shows either the default_intra_matrix or the downloaded_intra_matrix.
23:16	W_0[3:6]	R/W	0xXX	
15:8	W_0[3:5]	R/W	0xXX	

Table 343: MPEG video decoder registers *…continued*

Bit	Symbol	Access	Value	Description
7:0	W_0[3:4]	R/W	0xXX	
Offset 0x60 - W_TBL0_W8				
31:24	W_0[4:3]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 040 to 07C shows either the default_intra_matrix or the downloaded_intra_matrix.
23:16	W_0[4:2]	R/W	0xXX	
15:8	W_0[4:1]	R/W	0xXX	
7:0	W_0[4:0]	R/W	0xXX	
Offset 0x64 - W_TBL0_W9				
31:24	W_0[4:7]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 040 to 07C shows either the default_intra_matrix or the downloaded_intra_matrix.
23:16	W_0[4:6]	R/W	0xXX	
15:8	W_0[4:5]	R/W	0xXX	
7:0	W_0[4:4]	R/W	0xXX	
Offset 0x68 - W_TBL0_W10				
31:24	W_0[5:3]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 040 to 07C shows either the default_intra_matrix or the downloaded_intra_matrix.
23:16	W_0[5:2]	R/W	0xXX	
15:8	W_0[5:1]	R/W	0xXX	
7:0	W_0[5:0]	R/W	0xXX	
Offset 0x6C - W_TBL0_W11				
31:24	W_0[5:7]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 040 to 07C shows either the default_intra_matrix or the downloaded_intra_matrix.
23:16	W_0[5:6]	R/W	0xXX	
15:8	W_0[5:5]	R/W	0xXX	
7:0	W_0[5:4]	R/W	0xXX	
Offset 0x70 - W_TBL0_W12				
31:24	W_0[6:3]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 040 to 07C shows either the default_intra_matrix or the downloaded_intra_matrix.

Table 343: MPEG video decoder registers *…continued*

Bit	Symbol	Access	Value	Description
23:16	W_0[6:2]	R/W	0xXX	
15:8	W_0[6:1]	R/W	0xXX	
7:0	W_0[6:0]	R/W	0xXX	
Offset 0x74 - W_TBL0_W13				
31:24	W_0[6:7]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 040 to 07C shows either the default_intra_matrix or the downloaded_intra_matrix.
23:16	W_0[6:6]	R/W	0xXX	
15:8	W_0[6:5]	R/W	0xXX	
7:0	W_0[6:4]	R/W	0xXX	
Offset 0x78 - W_TBL0_W14				
31:24	W_0[7:3]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 040 to 07C shows either the default_intra_matrix or the downloaded_intra_matrix.
23:16	W_0[7:2]	R/W	0xXX	
15:8	W_0[7:1]	R/W	0xXX	
7:0	W_0[7:0]	R/W	0xXX	
Offset 0x7C - W_TBL0_W15				
31:24	W_0[7:7]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 040 to 07C shows either the default_intra_matrix or the downloaded_intra_matrix.
23:16	W_0[7:6]	R/W	0xXX	
15:8	W_0[7:5]	R/W	0xXX	
7:0	W_0[7:4]	R/W	0xXX	
Offset 0x80 - W_TBL1_W0				
31:24	W_1[0:3]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 080 to 0FC shows either the default_non_intra_matrix or downloaded_non_intra_matrix.
23:16	W_1[0:2]	R/W	0xXX	
15:8	W_1[0:1]	R/W	0xXX	
7:0	W_1[0:0]	R/W	0xXX	
Offset 0x84 - W_TBL1_W1				

Table 343: MPEG video decoder registers *…continued*

Bit	Symbol	Access	Value	Description
31:24	W_1[0:7]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 080 to 0FC shows either the default_non_intra_matrix or downloaded_non_intra_matrix.
23:16	W_1[0:6]	R/W	0xXX	
15:8	W_1[0:5]	R/W	0xXX	
7:0	W_1[0:4]	R/W	0xXX	
Offset 0x88 - W_TBL1_W2				
31:24	W_1[1:3]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 080 to 0FC shows either the default_non_intra_matrix or downloaded_non_intra_matrix.
23:16	W_1[1:2]	R/W	0xXX	
15:8	W_1[1:1]	R/W	0xXX	
7:0	W_1[1:0]	R/W	0xXX	
Offset 0x8C - W_TBL1_W3				
31:24	W_1[1:7]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 080 to 0FC shows either the default_non_intra_matrix or downloaded_non_intra_matrix.
23:16	W_1[1:6]	R/W	0xXX	
15:8	W_1[1:5]	R/W	0xXX	
7:0	W_1[1:4]	R/W	0xXX	
Offset 0x90 - W_TBL1_W4				
31:24	W_1[2:3]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 080 to 0FC shows either the default_non_intra_matrix or downloaded_non_intra_matrix.
23:16	W_1[2:2]	R/W	0xXX	
15:8	W_1[2:1]	R/W	0xXX	
7:0	W_1[2:0]	R/W	0xXX	
Offset 0x94 - W_TBL1_W5				
31:24	W_1[2:7]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 080 to 0FC shows either the default_non_intra_matrix or downloaded_non_intra_matrix.
23:16	W_1[2:6]	R/W	0xXX	
15:8	W_1[2:5]	R/W	0xXX	

Table 343: MPEG video decoder registers *…continued*

Bit	Symbol	Access	Value	Description
7:0	W_1[2:4]	R/W	0xXX	
Offset 0x98 - W_TBL1_W6				
31:24	W_1[3:3]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 080 to 0FC shows either the default_non_intra_matrix or downloaded_non_intra_matrix.
23:16	W_1[3:2]	R/W	0xXX	
15:8	W_1[3:1]	R/W	0xXX	
7:0	W_1[3:0]	R/W	0xXX	
Offset 0x9C - W_TBL1_W7				
31:24	W_1[3:7]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 080 to 0FC shows either the default_non_intra_matrix or downloaded_non_intra_matrix.
23:16	W_1[3:6]	R/W	0xXX	
15:8	W_1[3:5]	R/W	0xXX	
7:0	W_1[3:4]	R/W	0xXX	
Offset 0xA0 - W_TBL1_W8				
31:24	W_1[4:3]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 080 to 0FC shows either the default_non_intra_matrix or downloaded_non_intra_matrix.
23:16	W_1[4:2]	R/W	0xXX	
15:8	W_1[4:1]	R/W	0xXX	
7:0	W_1[4:0]	R/W	0xXX	
Offset 0xA4 - W_TBL1_W9				
31:24	W_1[4:7]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 080 to 0FC shows either the default_non_intra_matrix or downloaded_non_intra_matrix.
23:16	W_1[4:6]	R/W	0xXX	
15:8	W_1[4:5]	R/W	0xXX	
7:0	W_1[4:4]	R/W	0xXX	
Offset 0xA8 - W_TBL1_W10				
31:24	W_1[5:3]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 080 to 0FC shows either the default_non_intra_matrix or downloaded_non_intra_matrix.

Table 343: MPEG video decoder registers *…continued*

Bit	Symbol	Access	Value	Description
23:16	W_1[5:2]	R/W	0xXX	
15:8	W_1[5:1]	R/W	0xXX	
7:0	W_1[5:0]	R/W	0xXX	
Offset 0xAC - W_TBL1_W11				
31:24	W_1[5:7]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 080 to 0FC shows either the default_non_intra_matrix or downloaded_non_intra_matrix.
23:16	W_1[5:6]	R/W	0xXX	
15:8	W_1[5:5]	R/W	0xXX	
7:0	W_1[5:4]	R/W	0xXX	
Offset 0xB0 - W_TBL1_W12				
31:24	W_1[6:3]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 080 to 0FC shows either the default_non_intra_matrix or downloaded_non_intra_matrix.
23:16	W_1[6:2]	R/W	0xXX	
15:8	W_1[6:1]	R/W	0xXX	
7:0	W_1[6:0]	R/W	0xXX	
Offset 0xB4 - W_TBL1_W13				
31:24	W_1[6:7]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 080 to 0FC shows either the default_non_intra_matrix or downloaded_non_intra_matrix.
23:16	W_1[6:6]	R/W	0xXX	
15:8	W_1[6:5]	R/W	0xXX	
7:0	W_1[6:4]	R/W	0xXX	
Offset 0xB8 - W_TBL1_W14				
31:24	W_1[7:3]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 080 to 0FC shows either the default_non_intra_matrix or downloaded_non_intra_matrix.
23:16	W_1[7:2]	R/W	0xXX	
15:8	W_1[7:1]	R/W	0xXX	
7:0	W_1[7:0]	R/W	0xXX	
Offset 0xBC - W_TBL1_W15				

Table 343: MPEG video decoder registers *…continued*

Bit	Symbol	Access	Value	Description
31:24	W_1[7:7]	R/W	0xXX	The two quantizer matrices that are being used are mapped into control register space. Each value in the 8x8 quantizer matrix occupies eight bits. These two matrices take 128 bytes total. Offset 080 to 0FC shows either the default_non_intra_matrix or downloaded_non_intra_matrix.
23:16	W_1[7:6]	R/W	0xXX	
15:8	W_1[7:5]	R/W	0xXX	
7:0	W_1[7:4]	R/W	0xXX	
Offset 0xC0 - Extra_Pic_Info				
31:7	Reserved[24:0]	R	0x0000000	
6	Matrix_rd_en	R/W	0x0	If set, software may read the Quantization Tables (W_TBL*); otherwise the hardware controls the read address to the table memories. This bit is for diagnostics.
5:4	Intra_dc_precision[1:0]	R/W	0x0	Used to compute intra_dc_mult and reset values of intra_dc_pred.
3	Default_non_intra_q_matrix	R/W	0x0	If set, indicates the non-intra default quantizer matrix is being used.
2	Default_intra_q_matrix	R/W	0x1	If set, indicates the default intra quantizer matrix is being used.
1	Alt_scan	R/W	0x0	If set, the alternate scan table is used. Otherwise, the Zig-Zag scan table is used for the current block.
0	Quant_scale_type	R/W	0x0	This field is used to select the quantizer scale table.
Offset 0xC4 - RL_STATS				
31:20	Total_coded_block_ct[11:0]	R/W	0x000	Total number of coded blocks in multiples of 1024. Can be initialized by writing the required values through a register write transaction.
19:0	Total_symbol_cnt[19:0]	R/W	0x00000	Total number of non-zero DCT coefficients in multiples of 1024. This field can be initialized by writing the required values through a register write transaction.
Offset 0xC8 - MP_IQ_SEL_0				
31:0	sel_0[31:0]	R/W	0x00000000	The MP_IQ_SEL_0 and MP_IQ_SEL_1 is initialized to zero after hardware or software reset The coefficient selection scheme is programmable and operates at a 8-by-8 block level. DSPCPU will write 64-bit values in to the MP_IQ_SEL_0 and MP_IQ_SEL_1 control registers. Each bit in the control register is used to select either a coefficient value from VLD or a zero coefficient value with in a 8-by-8 block. The bit value of 0 selects the coefficient from the VLD and the bit values of 1 selects the zero value.
Offset 0xCC - MP_IQ_SEL_1				

Table 343: MPEG video decoder registers *…continued*

Bit	Symbol	Access	Value	Description
31:0	sel_1[31:0]	R/W	0x00000000	The MP_IQ_SEL_0 and MP_IQ_SEL_1 is initialized to zero after hardware or software reset The coefficient selection scheme is programmable and operates at a 8-by-8 block level. DSPCPU will write 64-bit values in to the MP_IQ_SEL_0 and MP_IQ_SEL_1 control registers. Each bit in the control register is used to select either a coefficient value from VLD or a zero coefficient value with in a 8-by-8 block. The bit value of 0 selects the coefficient from the VLD and the bit values of 1 selects the zero value.
Offset 0x200 - MC_PICINFO0				
31	Bsw	R/W	0xX	Set to 1 to swap bytes of each 32-bit word read from or write to SDRAM by MC to match current endianess of the DSPCPU. Note that swapping bytes for 16-bit word is not supported. Programming of this bit is required only if DSPCPU will access the frame pixels decoded by the MPEG Pipe. In that case, set bsw to 0 if DSPCPU is in little endian mode, or to 1 if DSPCPU is in big endian mode. Also, if the decoded pixels are to be displayed/processed by HDVO, then the byte swapping bit for the HDVO unit must be set the same as the bsw bit.
30:24	Mb_row_offset[6:0]	R/W	0xXX	Starting macroblock row in the decoding picture which the first macroblock row of the current destination field buffers corresponds to. For I and P pictures, this number should always be 0 because reference pictures are not allowed to span multiple discontiguous SDRAM buffers. For B pictures, the decoding picture is allowed to span multiple discontiguous SDRAM buffers; this offset allows the Storage Unit to compute the correct storage coordinates in the current destination buffers
23:20	Reserved[3:0]	R	0x0	
19:16	Mc_timeout_period[3:0]	R/W	0xX	If non-zero, VLD interrupts the CPU if no macroblocks are stored into SDRAM by the MC within mc_timeout_period*2048 CPU cycles while the VLD is in parsing mode; 0 has no effect
15:8	Mb_height[7:0]	R/W	0xXX	Number of luminance (Y) macroblocks per column in a frame picture. This is a fixed number for a given video sequence. MC relies on this number to detect out-of-bound reference errors
7:0	Mb_width[7:0]	R/W	0xXX	Number of macroblocks per row in a picture in the current sequence
Offset 0x208 - MC_PICINFO2				
31:24	Err_end_mb_row[7:0]	R/W	0xXX	Ending macroblock row in error concealment (for each errcon2_cmd)
23:16	Err_end_mb_col[7:0]	R/W	0xXX	Ending macroblock column in error concealment (for each errcon2_cmd)

Table 343: MPEG video decoder registers *…continued*

Bit	Symbol	Access	Value	Description
15:8	Err_mb_row[7:0]	R/W	0xXX	Macroblock row to start error concealment (for each errcon1_cmd or errcon2_cmd)
7:0	Err_mb_col[7:0]	R/W	0xXX	Macroblock column to start error concealment (for each errcon1_cmd or errcon2_cmd)
Offset 0x20C - MC_FREFY0				
31:0	Fb_fref_y_f0[31:0]	R/W	0xXXXXXXXX	SDRAM base address of forward reference top-field buffer for the Y-component
Offset 0x210 - MC_FREFY1				
31:0	Fb_fref_y_f1[31:0]	R/W	0xXXXXXXXX	SDRAM base address of forward reference bottom-field buffer for the Y-component
Offset 0x214 - MC_FREFUV0				
31:0	Fb_fref_uv_f0[31:0]	R/W	0xXXXXXXXX	SDRAM base address of forward reference top-field buffer for the UV-component
Offset 0x218 - MC_FREFUV1				
31:0	Fb_fref_uv_f1[31:0]	R/W	0xXXXXXXXX	SDRAM base address of forward reference bottom-field buffer for the UV-component
Offset 0x21C - MC_BREFY0				
31:0	Fb_bref_y_f0[31:0]	R/W	0xXXXXXXXX	SDRAM base address of backward reference top-field buffer for the Y-component
Offset 0x220 - MC_BREFY1				
31:0	Fb_bref_y_f1[31:0]	R/W	0xXXXXXXXX	SDRAM base address of backward reference bottom-field buffer for the Y-component
Offset 0x224 - MC_BREFUV0				
31:0	Fb_bref_uv_f0[31:0]	R/W	0xXXXXXXXX	SDRAM base address of backward reference top-field buffer for the UV-component
Offset 0x228 - MC_BREFUV1				
31:0	Fb_bref_uv_f1[31:0]	R/W	0xXXXXXXXX	SDRAM base address of backward reference bottom-field buffer for the UV-component
Offset 0x22C - MC_DESTY0				
31:0	Fb_dest_y_f0[31:0]	R/W	0xXXXXXXXX	SDRAM base address of destination top-field buffer for the Y-component
Offset 0x230 - MC_DESTY1				
31:0	Fb_dest_y_f1[31:0]	R/W	0xXXXXXXXX	SDRAM base address of destination bottom-field buffer for the Y-component
Offset 0x234 - MC_DESTUV0				
31:0	Fb_dest_uv_f0[31:0]	R/W	0xXXXXXXXX	SDRAM base address of destination top-field buffer for the UV-component
Offset 0x238 - MC_DESTUV1				
31:0	Fb_dest_uv_f1[31:0]	R/W	0xXXXXXXXX	SDRAM base address of destination bottom-field buffer for the UV-component
Offset 0x23C - MC_COMMAND				

Table 343: MPEG video decoder registers *…continued*

Bit	Symbol	Access	Value	Description
31:3	Reserved[28:0]	R	0x00000000	
2:0	Mc_cmd[2:0]	R/W	0x0	The 3-bit command field through which the CPU instructs the MC to perform certain special operations. Refer to .
Offset 0x240 - MC_PFCOUNT				
31:16	Count_24x5[15:0]	R/W	0x0000	Records the total number of 24x5 2-D memory fetches generated.
15:0	Count_24x4[15:0]	R/W	0x0000	Records the total number of 24x4 2-D memory fetches generated.
Offset 0x244 - MC_STATUS				
31:24	Reserved[7:0]	R	0x00	
23:16	Start_mb_col[7:0]	R	0xXX	Macroblock column position in the decoding picture where the current slice starts
15:8	State_mb_row[7:0]	R	0x00	Macroblock row position in the decoding picture where the last reconstructed macroblock is stored in SDRAM
7:0	State_mb_col[7:0]	R	0x00	Macroblock column position in the decoding picture where the last reconstructed macroblock is stored in SDRAM
Offset 0xFF4 - VMPG_POWER_DOWN				
31	Power_down_vmpg_ip	R/W	0x0	Power Down indicator 0: Power up 1: Powerdown
30:0	Reserved[30:0]	R	0xXXXXXXXX	
Offset 0xFFC - VMPG_MODULE_ID				
31:16	Module_id_vmpg_ip[15:0]	R	0x0100	MPEG Video Decoder Module ID register
15:12	Major_rev[3:0]	R	0x2	Major Revision 1: Implements RL & HDR write-back 10: Implements Matrix_rd_en bit.
11:8	Minor_rev[3:0]	R	0x0	Minor Revision
7:0	Aperature_size[7:0]	R	0x00	MPEG Video Aperture = 4 kB

5.12 Interrupt controller

The Interrupt controller used in the PNX2015 consists of a Generic Interrupt Controller (GIC), which has 16 inputs that can be routed towards two interrupt outputs. These outputs are available as pins on the PNX2015 device. They are provided to allow interrupts to be separated into high/low priority towards the PNX8550. Two levels of priority are implemented for each output in the controller. The interrupt sources and location in the controller is shown in Table 344.

Table 344: Interrupt sources

Input	Source
1	DCS-Network
2	TSU
3	PMAN1-Monitor
4	PMAN2-Monitor
5	VO-1
6	VO-2
7	AVIP-1
8	AVIP-2
9	Clock Module
10	MBS
11	VIP
12	PMAN1 Security
13	PMAN2 Security
14	VMPG
15	External 1
16	External 2

The external interrupt inputs (15 and 16) can be programmed to be active high or low level. The interrupt output pins are always active low and have open drain output pads.

5.12.1 Features

Features include:

- Level active interrupt request inputs with programmable polarity
- Software interrupt request capability associated to each request input
- Observable interrupt request state before masking
- Software programmable priority assignment to requests
- Priority level masking scheme with HW-configurable number of priority levels
- Fast identification of interrupt request through vector
- Support for nesting of interrupt service routines

5.12.2 Overview

The PNX2015 supports two external interrupt outputs that can be routed towards an external processor for servicing. Figure 237 shows the application of the Interrupt controller.

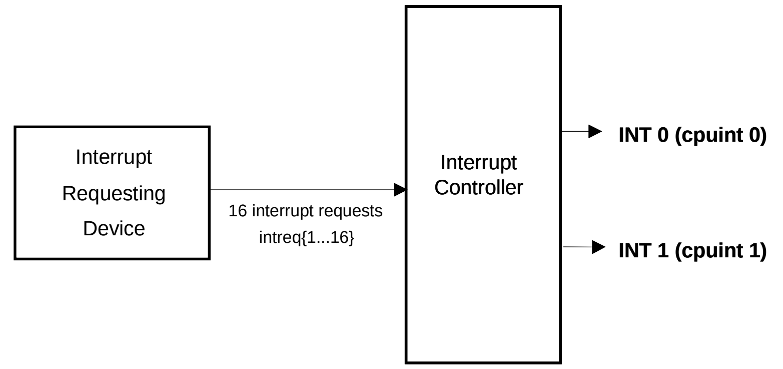


Fig 237.PNX2015 Interrupt Controller Application

5.12.3 Functional description

Configuration of interrupt controller in PNX2015

The interrupt controller is a generic IP block configured as follows in the PNX2015

N=16 Configure parameter N: number of interrupt request inputs

P=15 Configure parameter P: number of priority levels supported (minus one)

T=1 Configure parameter T: number of interrupt targets supported (minus one)

Figure 238 presents the functionality of the Interrupt Controller.

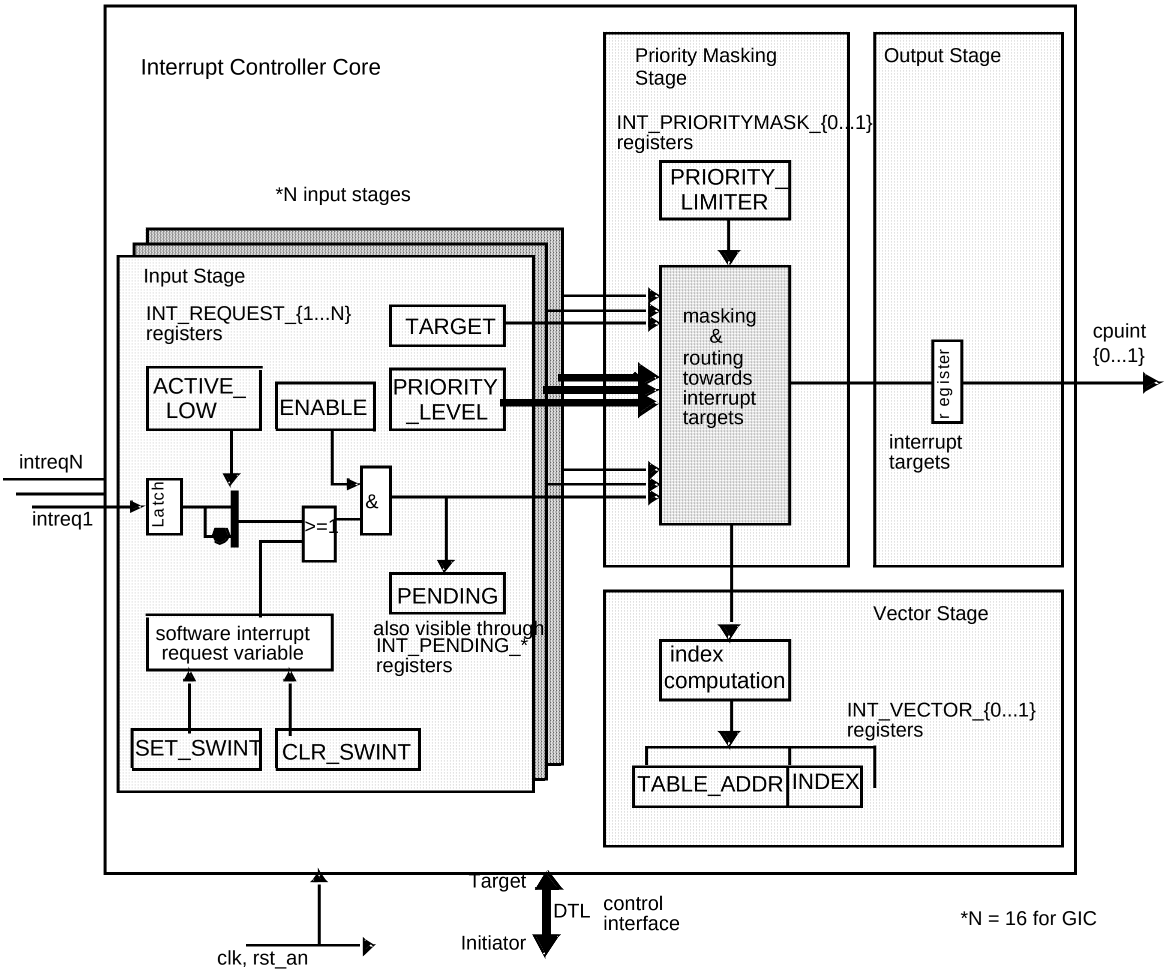


Fig 238.Interrupt Controller Block Diagram

5.12.3.1 Architecture

Input stage: An input stage performs the following tasks:

- Input of one interrupt request (intreq) signal
- Latch the interrupt request state during computation of the interrupt vector, otherwise keep the latch transparent.
- Invert the request polarity if the interrupt request signal is active low (controlled by variable ACTIVE_LOW).
- Combine the interrupt request with the state of a local software interrupt request variable.
- Enable or disable the resulting interrupt request (controlled by variable ENABLE).
- Forward the request to the priority masking stage together with attributes characterizing the interrupt request. These attributes are:

  The priority level assigned to the request (variable PRIORITY_LEVEL).

  The interrupt target defined for the request (variable TARGET).

In addition, the input stage provides the means to set and clear the software interrupt request variable (SET_SWINT and CLR_SWINT commands) and to observe the request status before priority masking (variable PENDING).

While no interrupt vector is being computed, the signal path of an interrupt request throughout the input stage (including the latch) is asynchronous and requires no active interrupt controller clock. Immediately before vector computation the latch synchronously captures the state of the intreq line and thereby blocks any signal changes to propagate into the vector stage, which might lead to incorrect index computation. Sufficient time to resolve potential metastability of the latched request is allowed. After vector computation, the latch becomes transparent again.

There is one input stage per interrupt request input. The vector stage references an input stage through an index. There is no input stage defined for index 0 as this index is reserved for a special purpose by the vector stage.

Priority masking stage: The priority masking stage performs the following tasks:

- For each interrupt target, input all interrupt requests selected for the target and mask pending interrupts which are at lower or equal priority than a target specific priority threshold (PRIORITY_LIMITER).
- For each interrupt target, combine pending interrupt requests with priority above the priority threshold through a logical OR and route the result towards the interrupt target.

The signal path of interrupt requests throughout the priority masking stage towards the output stage is asynchronous and requires no active interrupt controller clock

Output stage: The output stage performs the following task:

For each interrupt target, produce processor interrupt request output signals cpuint{0...1} at both active high and low level by registering the interrupt request information of the priority masking stage.

Remark: The contribution of the interrupt controller to the overall interrupt latency (which is the time between the assertion of an interrupt request by an interrupt device and the start of an Interrupt Service Routine (ISR) by the processor), is small. It equals the sum of combinatorial delays through the input (priority masking) and output stages, plus a maximum of 1 clk period.

Vector stage: The vector stage provides one vector register per interrupt target (INT_VECTOR_{0...1}). It performs the following tasks triggered by a read action to one of these registers.

1. For a read of register INT_VECTOR_t, process the PRIORITY information of input states with pending interrupt requests selected for TARGET = t.
2. Identify the input stage with the highest PRIORITY value above the target specific PRIORITY_LIMITER threshold (if this condition is true for a multitude of input stages, then the input stage with the highest index is taken).
3. Present the index of that input stage through the INDEX variable in the INT_VECTOR_t register. If no interrupt request exceeds the PRIORITY_LIMITER threshold, then INDEX=0 is given.

The above process is performed upon any INT_VECTOR_* read action. There is no storage of a previously computed vector.

The information from the INT_VECTOR_* register facilitates a generic software ISR in identifying the interrupt requesting device to be serviced. To invoke the ISR of that device, the INDEX variable can be taken as an offset into a table of address pointers towards a device-specific Interrupt Service Routine. Alternatively, the total content of the INT_VECTOR_* register, consisting of a table base address (variable TABLE_ADDR) plus INDEX, can be taken as pointer into a table.

INDEX = 0 identifies the special case that no interrupt request requires service when the INT_VECTOR_* register is read.

For correct vector computation, it is required that the ISR always reads the INT_VECTOR_* register that corresponds to the interrupt target.

5.12.4 Register descriptions

5.12.4.1 Register summary

The interrupt controller provides the registers (below) to give software access to its internal variables. The interrupt controller address space (aperture) is 4 kB.

Table 345: GIC register summary

Offset	Symbol	Description
0x0	INT_PRIORITYMASK_0	interrupt target 0 priority threshold
0x4	INT_PRIORITYMASK_1	interrupt target 0 priority threshold
0x100	INT_VECTOR_0	interrupt target 0 vector
0x104	INT_VECTOR_1	interrupt target 1 vector
0x200	INT_PENDING_1_16	status of interrupt requests 131
0x300	INT_FEATURES	Interrupt controller configuration features
0x404	INT_REQUEST_1	Interrupt request 1 (dcs_network), configuration/software interrupt control
0x408	INT_REQUEST_2	Interrupt request 2 (tsu), configuration/software interrupt control
0x40C	INT_REQUEST_3	Interrupt request 3 (pman1_monitor), configuration/software interrupt control -
0x410	INT_REQUEST_4	Interrupt request 4 (pman2_monitor), configuration/software interrupt control
0x414	INT_REQUEST_5	Interrupt request 5 (vo-1), configuration/software interrupt control
0x418	INT_REQUEST_6	Interrupt request 6 (vo-2), configuration/software interrupt control
0x41C	INT_REQUEST_7	Interrupt request 7 (avip-1), configuration/software interrupt control
0x420	INT_REQUEST_8	Interrupt request 8 (avip-2), configuration/software interrupt control
0x424	INT_REQUEST_9	Interrupt request 9 (clock module), configuration/software interrupt control
0x428	INT_REQUEST_10	Interrupt request 10 (mbs), configuration/software interrupt control -

Table 345: GIC register summary *…continued*

Offset	Symbol	Description
0x42C	INT_REQUEST_11	Interrupt request 11 (vip), configuration/software interrupt control
0x430	INT_REQUEST_12	Interrupt request 12 (pman1_security), configuration/software interrupt control -
0x434	INT_REQUEST_13	Interrupt request 13 (pman2_security), configuration/software interrupt control
0x438	INT_REQUEST_14	Interrupt request 14 (vmpg), configuration/software interrupt control-
0x43C	INT_REQUEST_15	Interrupt request 15 (external 1), configuration/software interrupt control
0x440	INT_REQUEST_16	Interrupt request 16 (external 2), configuration/software interrupt control
0xFFC	INT_MOD_ID	interrupt controller module ID

5.12.4.2 Register tables

Interrupt targets: The PNX2015 contains two "interrupt targets," which are connected to pin outs of the PNX2015.

The interrupt target is configured for each interrupt request input of the interrupt controller through the TARGET variable in the INT_REQUEST_* registers.

Interrupt priority: Interrupt request masking is performed individually per interrupt target by comparing the priority level assigned to a specific interrupt request input (variable PRIORITY_LEVEL in the INT_REQUEST_* registers) with a target specific priority threshold (variable PRIORITY_LIMITER in the INT_PRIORITYMASK_* registers).

Priority levels are defined as follows:

- Priority level 0 corresponds to ‘masked.’ Interrupt requests with priority 0 do not lead to an interrupt request towards processor or power management controller.
- Priority level 1 corresponds to lowest priority.
- Priority level F corresponds to highest priority.
- Programming the INT_REQUEST_* register variable ENABLE = 0 is an alternative to PRIORITY_LEVEL = 0 which is typically applied when an interrupt request input shall be temporarily disabled without the need to save and restore the current PRIORITY_LEVEL setting.

Software interrupts: Software interrupt support is provided through variables in the INT_REQUEST_* registers. Software interrupts can be applied as follows:

- Testing the RTOS interrupt handling without using a device-specific ISR
- Software emulation of an interrupt requesting device, including interrupts

Interrupt priority mask registers: This set of registers defines the threshold for priority level masking. There is one INT_PRIORITYMASK_* register per interrupt target.

Table 346: INT_PRIORITYMASK_{0,1} registers

Bit	Symbol	Access	Value	Description
Offset 0x0 - INT_PRIORITYMASK_0				
31:3	RSD[31:3]	R	0xXXXX XXXX	reserved for future extensions; should be written as 0
2:0	PRIORITY_LIMITER_0[2:0]	R/W	0xX	Priority Limiter: this variable determines a priority threshold that incoming interrupt requests must exceed to trigger interrupt requests towards CPU. Legal PRIORITY_LIMITER values are 0 .. P; other values are reserved and lead to undefined behavior. PRIORITY_LIMITER = 0: incoming interrupt requests with priority > 0 can trigger interrupt requests towards CPU PRIORITY_LIMITER = n: only interrupt requests at a priority level above n can trigger interrupt requests towards CPU PRIORITY_LIMITER = P: no incoming interrupt requests can trigger interrupt requests towards CPU. high order bits not required for PRIORITY_LIMITER encoding are read-only 0
Offset 0x4 - INT_PRIORITYMASK_1				
31:3	RSD[31:3]	R	0xXXXX XXXX	reserved for future extensions; should be written as 0
2:0	PRIORITY_LIMITER_1[2:0]	R/W	0xX	Priority Limiter: this variable determines a priority threshold that incoming interrupt requests must exceed to trigger interrupt requests towards CPU. Legal PRIORITY_LIMITER values are 0 .. P; other values are reserved and lead to undefined behavior. PRIORITY_LIMITER = 0: incoming interrupt requests with priority > 0 can trigger interrupt requests towards CPU PRIORITY_LIMITER = n: only interrupt requests at a priority level above n can trigger interrupt requests towards CPU PRIORITY_LIMITER = P: no incoming interrupt requests can trigger interrupt requests towards CPU. high order bits not required for PRIORITY_LIMITER encoding are read-only 0

The PRIORITY_LIMITER variable can be used to define the minimum priority level for nesting interrupts: typically, the PRIORITY_LIMITER variable is set to the priority level of the ISR that is currently being executed. By doing this, only interrupt requests at a higher priority level lead to a nested interrupt service. Nesting can be disabled by setting PRIORITY_LEVEL = F or by disabling interrupt exceptions within the processor.

Interrupt vector registers: These registers identify - individually for each interrupt target - the highest priority- enabled pending interrupt request that is present at the time a register is being read.

Table 347: INT_VECTOR_{0,1} registers

Bit	Symbol	Access	Value	Description
Offset 0x100 - INT_VECTOR_0				
31:11	TABLE_ADDR_0 [20:0]	R/W	0xXXXXXX	Table start address: indicates the lower address boudary of a 2048 byte aligned interrupt vector table in memory
10:7	RSD[10:7]	R	0xX	Reserved
6:3	INDEX_0[3:0]	R	0xX	Index: indicates the intreq line number of the interrupt request to be served by the processor: INDEX = 0 no interrupt request to be served INDEX = 1: serve interrupt request at input intreq1 INDEX = 2: serve interrupt request at input intreq2 INDEX = N: serve interrupt request at input intreqN
2:0	NULL_0[2:0]	R	0x0	bit field always read as 0

Table 347: INT_VECTOR_{0,1} registers

Bit	Symbol	Access	Value	Description
Offset 0x104 - INT_VECTOR_1				
31:11	TABLE_ADDR_1 [20:0]	R/W	0xXXXXXX	Table start address: indicates the lower address boudary of a 2048 byte aligned interrupt vector table in memory
10:7	RSD[10:7]	R	0xX	Reserved
6:3	INDEX_1[3:0]	R	0xX	Index: indicates the intreq line number of the interrupt request to be served by the processor: INDEX = 0 no interrupt request to be served INDEX = 1: serve interrupt request at input intreq1 INDEX = 2: serve interrupt request at input intreq2 INDEX = N: serve interrupt request at input intreqN
2:0	NULL_1[2:0]	R	0x0	bit field always read as 0

The software ISR should always read the vector register that corresponds to the interrupt target, for example:

read INT_VECTOR_0 for interrupt target 0 service

read INT_VECTOR_1 for interrupt target 1 service

The INT_VECTOR_* register content can be used as a vector into a memory based table. This table has N+1 entries. To be able to use the register content as a full 32 bit address pointer, the table should be aligned to a 2048 byte address boundary (see Figure 239). If only the INDEX variable is used as offset into the table, then this address alignment is not required.

Each table entry has 64-bit data. It is recommended to pack per table entry - the start address of a device specific ISR, plus the associated priority limiter value (if nesting of ISR shall be performed).

64-bit packing optimizes the speed of nested interrupt handling by processors that implement caches. In the (likely) case of a cache miss when reading data from the table, the priority limiter value to be programmed into the INT_PRIORITYMASK register is loaded into the cache along with the ISR start address, saving several clock cycles of interrupt processing time compared to a solution where the priority limiter value would have to be established from an INT_REQUEST_* register.

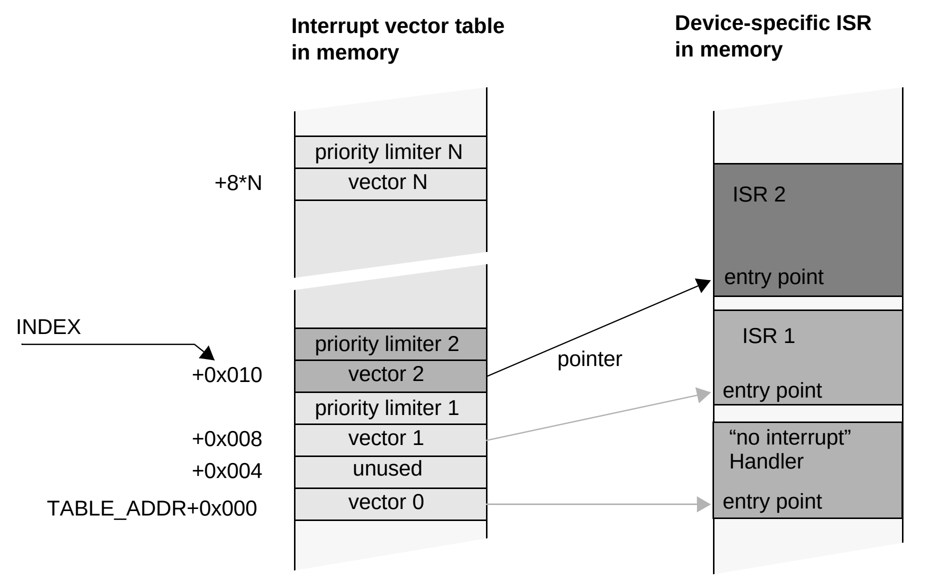


Fig 239.Memory-based interrupt vector/priority table

A vector with INDEX = 0 indicates that no interrupt with priority above the priority threshold is pending. The vector table should implement for this entry a “no interrupt” handler to treat this special case.

Note that due to the special purpose of INDEX = 0, no interrupt request input intreq0 signal exists, and thus no INT_REQUEST_0 register exists.

Interrupt pending registers: This set of registers gathers the PENDING variables of all interrupt requests.

Software can make use of INT_PENDING registers to gain a faster overview on pending interrupt requests than by reading individual INT_REQUEST_* registers. For certain ISR software this may lead to a benefit in interrupt processing time.

There are 3 registers. INT_PENDING_1_16 reflects the state of signals intreq1...16. Bit positions within the INT_PENDING_* registers increase with the index of interrupt request signals.

Table 348: INT_PENDING registers

Bit	Symbol	Access	Value	Description
Offset 0x200 - INT_PENDING_1_16				
31:17	RSD[31:17]	R	0xXXXX	Reserved
16:1	PENDING_INTERRUPT [15:0]	R	0xXXXX	Pending Interrupt Request: This variable reflects the state of the intreq[i] line (if needed, converted to active high) OR'ed by the state of the local software interrupt request variable at the time the register is read. Note that the PENDING variable is also reflected by the INT_REQUEST_*registers individually present for each interrupt request input. PENDING = 0: no interrupt request PENDING = 1: interrupt request is pending
0	RSD_Bit0	R	0xX	Reserved

Note that bit 0 and bits 17:31 of register INT_PENDING_1_16 are always read as 0 because the interrupt controller does not implement an interrupt request input 0.

Interrupt features register: This register indicates the hardware configuration parameters chosen during the creation of the interrupt controller. A generic ISR can make use of the INT_FEATURES register to implement interrupt controller configuration-specific behavior.

Table 349: INT_FEATURES register

Bit	Symbol	Access	Value	Description
Offset 0x300 - INT_FEATURES				
31:22	RSD[31:22]	R	0xXXX	Reserved
21:16	T[5:0]	R	0x01	Configuration parameter T: number of interrupt targets supported (minus one)
15:8	P[7:0]	R	0x07	Configuration parameter P: number of priority levels supported (minus one)
7:0	N[7:0]	R	0x10	Configuration parameter N: number of interrupt request inputs

Interrupt Request Registers: This set of registers holds configuration information related to interrupt request inputs of the interrupt controller and allows to issue software interrupt requests.

Table 350: INT_REQUEST_{1...16} registers

Bit	Symbol	Access	Value	Description
Offset 0x404 - INT_REQUEST_1				
31	PENDING_1	R	0xX	Pending Interrupt Request: This variable reflects the state of the intreq line (if needed, converted to active high) OR'ed by the state of the local software interrupt request variable at the time the register is read. Note that the PENDING variable is also visible from the INT_PENDING_*registers. PENDING = 0: no interrupt request PENDING = 1: interrupt request pending
30	SET_SWINT_1	W	0x0	Set software interrupt request SET_SW_INT = 0 (write): no effect on the state of the local software interrupt request variable SET_SWINT = 1 (write): set the state of the local software interrupt request variable to 1 SET_SWINT is always read as 0
29	CLR_SWINT_1	W	0x0	Clear software interrupt request: CLR_SWINT = 0 (write): no effect on the state of the local software interrupt request variable CLR_SWINT = 1 (write): clear the state of the local software interrupt request variable to 0. CLR_SWINT is always as 0
28	WE_PRIORITY_LEVEL_1	W	0xX	Write enable PRIORITY_LEVEL WE_PRIORITY_LEVEL = 0 (write): no change of PRIORITY_LEVEL variable state WE_PRIORITY_LEVEL = 1 (write): PRIORITY_LEVEL variable state may be changed WE_PRIORITY_LEVEL is always read as 0
27	WE_TARGET_1	W	0xX	Write enable target WE_TARGET = 0 (write): no change of TARGET variable state WE_TARGET = 1 (WRITE): TARGET variable state may be changed WE_TARGET is always read as 0
26	WE_ENABLE_1	W	0xX	Write enable ENABLE WE_ENABLE = 0 (write): no change of ENABLE variable state WE_ENABLE = 1 (write): ENABLE variable state may be changed WE_ENABLE is always read as 0

Table 350: INT_REQUEST_{1...16} registers *…continued*

Bit	Symbol	Access	Value	Description
25	WE_ACTIVE_LOW_1	W	0xX	Write enable ACTIVE_LOW WE_ACTIVE_LOW = 0 (write): no change of ACTIVE_LOW variable state WE_ACTIVE_LOW = 1 (write): ACTIVE_LOWvariable state may be changed WE_ACTIVE_LOW is always read as 0
24:18	RSD[24:18]	R	0xXX	Reserved: should be written as 0
17	ACTIVE_LOW_1	R/W	0x0	Active Low: This variable selects the polarity of the interrupt request input signal. See also WE_ACTIVE_LOW. ACTIVE_LOW = 1: the intreq signal is interpreted as active low ACTIVE_LOW = 0: the intreq signal is interpreted as active high
16	ENABLE_1	R/W	0x0	Enable interrupt request: This variable controls whether an interrupt request is enabled for further processing by the interrupt controller. See also WE_ENABLE. Enable = 0: the interrupt request is discarded. It cannot cause a processor or power management interrupt request ENABLE = 1: the interrupt request may cause a processor or power management interrupt request when further conditions for this become true.
15:14	RSD[15:14]	R	0xX	Reserved: should be written as 0
13:8	TARGET_1[5:0]	R/W	0x00	Interrupt target: This variable defines the interrupt target of an interrupt request. Legal values are 0 .T. Other values are reserved and lead to undefined behavior. See also WE_TARGET. TARGET = 0: the interrupt request shall lead to a processor interrupt request 0 (cpuint0) TARGET = 1: the interrupt request shall lead to a processor interrupt request 1 (cpuint1) TARGET = T: the interrupt request shall lead to a processor interrupt request T (cpuintt) High order bits not required for TARGET encoding are read-only 0. For a single-target configuration (T=0), the TARGET field must always be written as 0
7:0	PRIORITY_LEVEL_1[7:0]	R/W	0xXX	Priority level: this variable determines the priority level of the interrupt request. Legal values are 0P. Other values are reserved and lead to undefined behavior. See also WE_PRIORITY_LEVEL. PRIORITY_LEVEL = 0: the interrupt request has priority level 0 (masked): it is ignored PRIORITY_LEVEL =1: the interrupt request has priority level 1 (lowest) PRIORITY_LEVEL = P: the interrupt request has priority level P (highest) High order bits not required for PRIORITY_LEVEL encoding are read-only 0
0x03 E408int_request_reg_2				
IPC TM32-1register, Target 1. This register is identical to the int_request_1 register (Offset 0x03 E404)				
0x03 E40Cint_request_reg_3				
IPC TM32-2 register, Target 1. This register is identical to the int_request_1 register (Offset 0x03 E404)				
0x03 E410int_request_reg_4				
Reserved				
0x03 E414int_request_reg_5				
Universal Serial Bus Interrupt register. This register is identical to the int_request_1 register (Offset 0x03 E404)				
0x03 E418int_request_reg_6				
GPIO Interrupt Event Queue 0 register. This register is identical to the int_request_1 register (Offset 0x03 E404)				
0x03 E41Cint_request_reg_7				
GPIO Interrupt Event Queue 1 register. This register is identical to the int_request_1 register (Offset 0x03 E404)				
0x03 E420int_request_reg_8				

Table 350: INT_REQUEST_{1...16} registers *...continued*

Bit	Symbol	Access	Value	Description
GPIO Interrupt Event Queue 2 register. This register is identical to the int_request_1 register (Offset 0x03 E404)				
0x03 E424int_request_reg_9				
GPIO Interrupt Event Queue 3 register. This register is identical to the int_request_1 register (Offset 0x03 E404)				
0x03 E428int_request_reg_10				
GPIO Interrupt Event Queue 4 register. This register is identical to the int_request_1 register (Offset 0x03 E404)				
0x03 E42Cint_request_reg_11				
GPIO Interrupt Event Queue 5 register. This register is identical to the int_request_1 register (Offset 0x03 E404)				
0x03 E430int_request_reg_12				
Reserved				
0x03 E434int_request_reg_13				
Quality Video Composition Processor Interrupt 1 register. This register is identical to the int_request_1 register (Offset 0x03 E404)				
0x03 E438int_request_reg_14				
Quality Video Composition Processor Interrupt 2 register. This register is identical to the int_request_1 register (Offset 0x03 E404)				
0x03 E43Cint_request_reg_15				
1^{2}C 1 Interrupt register. This register is identical to the int_request_1 register (Offset 0x03 E404)				
0x03 E440int_request_reg_16				
1^{2}C 2 Interrupt register. This register is identical to the int_request_1 register (Offset 0x03 E404)				

For changing the TARGET variable state dynamically, software should first disable the interrupt request (ENABLE = 0), then change TARGET and finally re-enable the request (ENABLE = 1).

Write enable commands are provided to allow the modification of individual INT_REQUEST_* variables by simple write operations instead of automatic read-modify-write operations. This feature allows to access INT_REQUEST_* registers simultaneously by multiple software threads.

Interrupt module ID register: This register identifies the type and revision of the interrupt controller. A generic ISR can make use of the INT_MOD_ID register to implement interrupt controller type or revision specific behavior.

Table 351: (INT_MOD_ID) register

Bit	Symbol	Access	Value	Description
Offset 0xFFC - INT_MOD_ID				
31:16	ID[15:0]	R	0x1106	Identification number: Unique module identifier (assigned) by the Philips Semiconductors ReUse Technology Group) indicating the type of interrupt controller.
15:12	MAJOR_REV[3:0]	R	0x1	Major Revision: Major revision of module implementation
11:8	MINOR_REV[3:0]	R	0x0	Minor revision: Module revision of module implementation, starting at 0.
7:0	APERTURE_SIZE[7:0]	R	0x00	Aperture Size: 4 Kbyte address aperture.

5.13 Video Output (VO) module

In a memory-based architecture, the video processing is carried out on the active pixels and the intermediate results from this video processing are usually stored in memory. This memory-based pixel data is later converted into a desired video signal format at the output display interface. The output formatting is often achieved by a simple video output unit that can fetch active (and processed) pixel data from memory via a DMA protocol and output it directly in the desired format.

For the above video output unit, the ability for it to be slaved to an external clock and external timing control signal(s) is an important feature. The current Video Output IP (VO) is designed in a way that it supports both a master (where it generates the required timing control signals depending on the raster) and a slave mode (where it takes as input and recognizes external timing control signals).

5.13.1 Configuration of VO stages in PNX2015

In the PNX2015 the background colour is displayed if there are no valid pixels available, or first frame/field or there is an underflow interrupt. Underflow can be caused by inappropriate Hub arbitration settings or incorrect VO STG settings.

The VO IP block is configured with 16 bit video out, not 30 bit as selectable in the IP block.

The clocks Pixel Clock (clk_pix), Processing Clock (clk_proc) are provided by the HD subsystem, clock generation block.

The VO can receive vip_stream_clk, Hsync, Vsync and FREF to operate in slave mode.

The VO does not generate explicit syncs only embedded syncs.

The VO STG register must be programmed so that it's Hsync and Vsync lie within the appropriate blanking interval.

5.13.2 Features

VO has the following main features:

Selectable input video formats (8-bit semi-planar YUV 4:2:2, 8-bit semi-planar YUV 4:2:0 using line repetition, 8-bit packed YUV 4:2:2;(YUY2,UYVY).

Selectable input external sync formats(HB(HREF),VB(VREF) and FREF as one set of sync format or HS and VS as another set of sync formats).

Flexible screen timing generation adopted to any display requirements (SD-TV standards, HD-TV standards, progressive and interlaced formats).

Output digital video format generation in 8-bit YUV 4:2:2 and 16-bit YUV 4:2:2.

VBI Insertion in the horizontal blanking interval.

5.13.3 Functional description

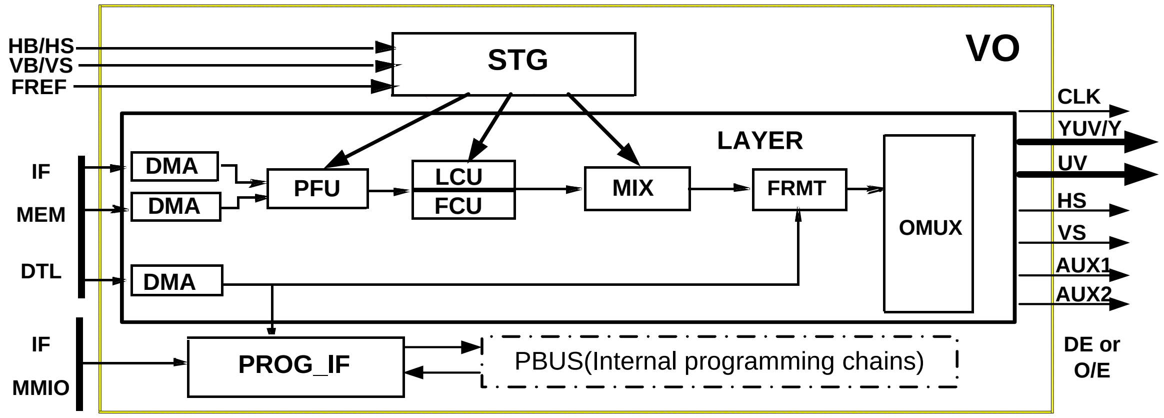


Fig 240.VO Block Diagram

5.13.3.1 Architecture

As shown in the block diagram above, VO contains mainly 3 blocks; LAYER, STG and PROG_IF(Programming Interface).

LAYER fetches the video data from the memory using the DTL protocol and sends out the formatted data along with the embedded syncs according to the outmode programmed. Refer to Section “Formatter (FRMT)” for more details on the outmodes.

Refer to Section “Direct Memory Access Control (DMA)” for all the components that comprise the LAYER.

STG provides timing control to VO either in a standalone mode (Master) or Slave mode and, in addition, generates the external syncs driven to the VO output.

Refer to Section “Screen Timing Generator (STG)” for more details on the STG.

PROG_IF contains all the control logic required to hookup the internal programming chains to the MMIO Interface.

Direct Memory Access Control (DMA): VO has 3 DMA agents, each of which connects to a 512-Byte buffer in the DMA adapter. DMA agent 1 is meant for the video data fetch. DMA agent 2 is used for supporting the semi-planar input format. DMA agent 3 is used to fetch a VBI packet or a data packet for DMA-based control-register programming.

Pixel Formatter Unit (PFU): The Pixel Formatter/Fetch Unit (PFU) interfaces to the device-transaction-protocol block which translates the memory-system protocol to device-level-transaction protocol. The PFU retrieves the raw data stream, for a particular image source, from the system memory and formats it according to the specified pixel format. Table 352 summarizes the various native pixel formats supported by the PFU.

Table 352: Summary of Native Pixel Formats

Format	Description
Packed YUV 4:2:2 (UYVY)	16-b 2 successive units contain 2 horizontally-adjacent pixels, no alpha
Packed YUV 4:2:2 (YUY2,2vuy)	16-b 2 successive units contain 2 horizontally-adjacent pixels, no alpha
Semiplanar YUV 4:2:2	Separate memory planes for Y and UV

LCU&FCU: LCU contains all the registers and the control logic necessary to send the required data at the correct screen co-ordinates to the Mixer. It also takes care of the clipping, cropping of any out of area images depending on the size of the layer and the negative images starts of the layer.

FCU controls the data that goes to mixer and generates the underflow interrupt whenever it runs out of data.

Mixer (MIX): Mixer sends out the correct data to the FRMT for formatting.

If there are no valid pixels, a programmable background color (offset 0x01C) is displayed. There is also programmable register, Solid Color (offset 0x320) which has an enable and when turned on replaces the layer data with the programmed solid color.

Formatter (FRMT): The output formatter (FRMT) produces the final output stream in the required output format. It also inserts the VBI data into the output D1 streams.

VO supports the following output formats.

Single D1 Mode: In this mode, only the following 8 video data signals are used out of 30. Rest of the signals carry 0’s.

data_out[9:2]: image stream with multiplexed components

The output looks as follows in this mode:

Upper(data_out[29:22]): 00000000000000 ....

Middle(data_out[19:12]) :000000000000 ...

Lower(data_out[9:2]): UYVYUYVYUYVYUYVYUYVY ....

Double D1 Mode: In this mode, only 16 video data signals are used out of the 30. VO outputs YUV422 by splitting Y and UV into two separate streams, Y uses 8 signals and UV uses the other 8 signals. Rest of the signals carry 0’s.

data_out[19:12]: UV component of image stream

data_out[9:2]: Y component of image stream

The output looks as follows in this mode:

Upper(data_out[29:22]): 00000000000000 ....

Middle(data_out[19:12]) : UVUVUVUVUVUVUVUV ....

Lower(data_out[9:2]): YYYYYYYYYYYYYY ...

OMUX: This module does all the multiplexing of the correct video in to the desired output format. It also takes care of the polarity control of various sync signals.

OMUX outputs the following:

CLK (Output Clock)

Output Data in the programmed format (Single D1 or Double D1 mode)

HS (Horizontal Sync)

VS (Vertical Sync)

AUX1 and AUX2 as defined below

VO has two auxiliary (Aux) output ports, each of which can be programmed for:

Composite blanking, where the value on the port is asserted HIGH (1) if either Vblanking is TRUE or Hblanking is TRUE or both are TRUE. The complement of the value on the port can, therefore, be used as the indicator of valid/active pixels.

Odd/even indication, where the field polarity (odd field=0 and even field=1) is indicated in the interlaced mode. The value on the port is 0 in progressive mode.

Screen Timing Generator (STG): The Screen Timing Generator (STG) creates the required synchronization signals for the monitor or other display devices for both progressive and interlaced modes. The screen timing generator usually operates as the timing master in the system. However, it is also possible to synchronize the operation of the screen timing generator to external events i.e., a field synchronization signal and/or a vertical synchronization signal and/or Horizontal synchronization signal. The screen timing generator also defines the active display region. The coordinate system for the STG is (x, y), with (0, 0) referring to the top left of the display surface. The coordinate (Horizontal Total, Vertical Total) defines the bottom right of the display surface. Horizontal and vertical blanking intervals, synchronization signals, and the visible display are within these boundaries.

5.13.4 General operations

5.13.4.1 Data flow and control

The input to VO is the video data that is fetched by the 3 DMA engines. For video data fetches, the request block size is equal to the layer width. If start_fetch is disabled (i.e., Enable bit 31 of register 0x2C8 is programmed to zero), the first DMA request starts right after the layer_enable is asserted (bit 0 of the register 0x240 is set to 1) and VO works as if prefetch is enabled. However, if start_fetch is enabled (i.e., Enable bit 31 of register 0x2C8 is programmed to one), then the DMA starts fetching only when VO’s internal line counter reaches the 12-bit line threshold programmed in the Fetch Start bits [11:0] of register 0x2C8. Data fetched for the first field (interlaced) or frame (progressive) is not used and is flushed at the FCU (Fetch Control Unit). Thereafter, the pixels for the second field/frame start marching into the FCU, waiting for the correct layer position. The FCU releases pixels only if the (x,y) coordinates generated by the Screen Timing Generator (STG) match the layer position. In case of an interlaced output, the field ID LSB is also checked. The DMA fetch request for the next active video line starts as soon as the last active pixel of the current line moves from the adapter FIFO into the processing pipeline and this request should be served in time to guarantee that the first active pixel of this new line is ready at FCU before the STG signals the start of active video for the new line.

5.13.4.2 Standard features

Clock Programming: VO's datapath contains 3 clock domains, viz. Pixel Clock(clk_pix), Processing Clock(clk_proc) and Output Clock(clk_out).

Out of these 3 clocks, clk_pix and clk_proc are the same. clk_ out=Vip Stream Clock or clk_out = n x clk_pix, where n = 1, 2 depending on the output interface mode (i.e., whether it is Single D1 mode or Double D1).

Out Clock Selection: VO Output Clock (dv_clk_out) can be selected to have either a normal clock or an inverted clock using the bit 13 of the OUT_CTRL register (offset, 03C). Bit 13 is "out_clk_ctl" which has the default value of 1 so that the default output clock is inverted.

Clock Frequencies: For both pixel and out clocks, the frequencies are as follows:

Minimum Frequency = 13.5 MHz

Maximum Frequency = 86 MHz

Reset-Related Issues: After the Reset, all the Registers are initialized to their default values. No sync or data is available on the output ports till the MMIO programming is finished.

Interrupt Processing: VO generates the following 8 interrupts based on their respective events and sends them out. VO doesn't receive and process the interrupts from outside.

EXT_HSYNC_CHANGE : This interrupt indicates that the line length derived from external HSYNC has changed. It is very useful when VO works as a synchronization slave in a system.

LAYER_DONE : This interrupt is generated to indicate that the last pixel has been sent out and the layer has been completely displayed.

BUF_DONE : This interrupt is generated when the DMA channel is done fetching all data for the layer.

FCU_UNDERFLOW : This interrupt signals that there is Underflow in FCU for the layer. This occurs whenever VO is short of pixel and it has to pump out one. It means that there is not enough system memory bandwidth allocated to VO.

VINTB : Vertical line interrupt issued if Y position matches VLINTB

VINTA : Vertical line interrupt issued if Y position matches VLINTA

VBI_DONE_INT : VBI/Register load is done with all the current list of packets

VBI_PACKET_INT : This interrupt is generated when the bit 15 (GI) of the VBI packet is set and VBI/Register reload has sent a packet.

Power Management: Setting the Powerdown bit (register offset 0xFF4) puts VO in to powerdown mode.

During Powerdown,

VO retains the settings of all the registers and allows read responses to DTL MMIO read transactions for only the Powerdown register.

At system level, all the clocks to VO can be stopped to reduce the power consumed by VO.

MMIO and Task Based Programming: In order for the VO to function properly, its various blocks have to be configured. Each functional unit contains a set of programming registers. A more detailed description of the various registers can be found in Section 6.6 of this document.

The registers are divided in to global registers and sub-module specific registers. The global register space accommodates functions such as screen timing and output format related functions.

There are two ways to access the VO registers:

The first and primary way to get read/write access to the registers is via the DTL-MMIO, which maps the registers into the overall system address space.

The second way to get write-only access to the registers is via data structures fetched through an additional DMA access port. The DMA port can also be used to fetch VBI data which get inserted into the output data stream. Differentiation between VBI and programming data is accomplished via a different header.

The DMA-based register-control programming only needs to be done once for a particular display scenario. The data structure to be used contains a header consisting of a pointer to the next packet in memory. A null pointer indicates the last packet in a linked list. The header also contains a field ID field which allows field synchronized insertion of VBI or re-programming packets. Packet insertion can cause an interrupt if the appropriate header flag is set. A detailed view of the packet format can be found in Table 353.

Each data packet consists of an 8-byte descriptor followed by data.

Table 353: Data packet descriptor

Bit	Name	Description
12:0		Data byte count (It doesn’t include the header size, only the following data)
13		Unused
14	WV	1=wait for proper vertical field 0=send data on current field without considering the field ID (for a series of packets to be inserted in the same field, this bit should only be set for the first packet and not for subsequent ones. If this bit is set for all packets, they will be inserted with one field delay each).
15	GI	1=generate interrupt when this packet is transmitted 0=don’t generate packet interrupt
27:16		Screen line in which to insert the data packet 0=first line after rising edge of VSYNC 0xFFF=line compare disabled. The packet is inserted without consideration of the line counter. all other values = The packet is inserted at line number specified by this value.
30:28		Field ID for this packet to be sent on

Table 353: Data packet descriptor *…continued*

Bit	Name	Description
31	DT	Data type 0=VBI data 1=register reprogram data for internal VO registers
63:32		Next packet address
61:...		Data if bit 31=0, the data block consists of byte VBI data if bit 31=1, each qword in the data portion is: 15:0 VO register address (offset) 31:16 unused 63:32 register data

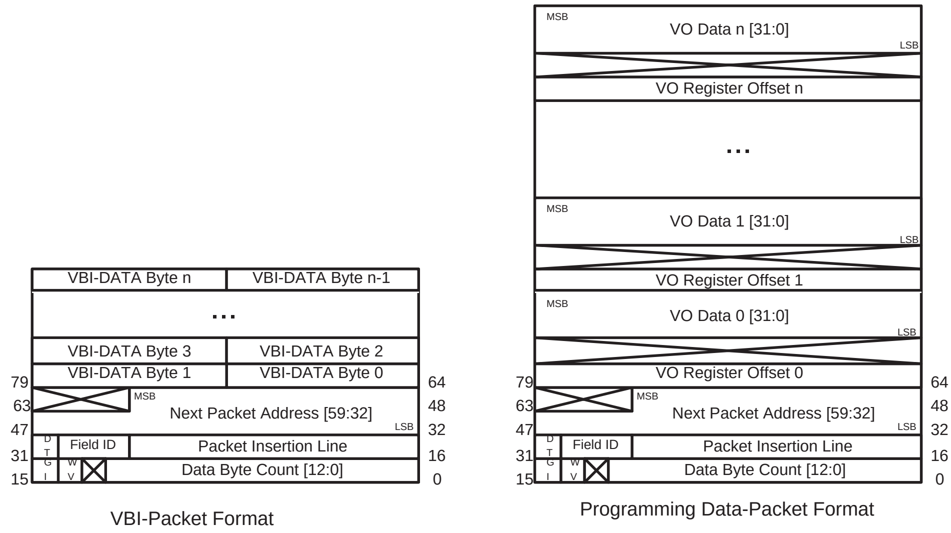


Fig 241.VBI/programming data packet formats

5.13.4.3 Module-specific capabilities

VBI: VO is designed such that VBI packets are only inserted in the horizontal blanking interval and only one VBI packet is allowed in any one horizontal blanking interval. To insert this packet, there are two DMA requests. The first request has a block size of 1 since it is used to fetch only the header (which contains the size information). The second request is meant to fetch actual data of the required size and so, the maximum DMA request size (for the second request) is equal to the length of the Data Byte Count. The VBI data for the current horizontal blanking interval is always fetched in advance and stored in the DMA buffer (in the adapter). After sending out this prefetched data, the VBI DMA control unit (DMA3) requests a prefetch of the next packet (and correct operation requires that the sequence of the next two fetches should complete before the start of the next horizontal blanking interval). In the current design, the VBI packet insertion begins immediately after EAV and continues till exhausting all data in the packet or SAV is reached, whichever occurs first.

When Line number field is other than 0xFFF, In case that VBI packet is smaller than the hblank interval, unused hblank spaces are padded with standard D1 blanking values (UYVY=0x80,0x10,0x80,0x10).

However, when VBI packet is larger than the hblank interval, the packet data that exceeded the size of hblank is discarded.

When line number field is 0xFFF,

In case that VBI packet is smaller than the hblank interval, unused hblank spaces are padded with standard D1 blanking values (UYVY=0x80,0x10,0x80,0x10).

However, when VBI packet is larger than the hblank interval, exceeded packets are inserted into the next hblank.

Shadow registers: Whenever any picture setting needs to be changed, it always a good idea to make it a seamless transition, i.e. no noticeable artifacts should be observed by the general audience. For most use cases, the goal is to change settings in between fields/frames, or during non-active video lines (e.g. VBI).

VO provides two mechanisms (programmable/selectable via a register bit) for register shadowing, whereby certain registers are shadowed to prevent screen artifacts during the re-configuration operations.

One method allows all new setting changes to really take effect at any line location assigned by user. By using a duplicated set of the acting registers --- the shadow registers as they are commonly called ---, any register changes first get buffered in these registers, wait for the correct time, i.e. the programmed line to be the current line being processed, and then be passed to acting registers. The contents of the shadow registers are transferred to the corresponding active registers at the line location indicated by 0x1F0[11:0]. The user has R/W access only to the shadow registers, but not the active registers.

Besides a trigger from the line location indicated by the register at 0x1F0[11:0], the second method comprises shadowing on a positive edge of the layer-enable signal (i.e., when a disabled layer is enabled). The “positive edge” of layer enable implies “when the layer_enable register changes from 0->1”.

In conclusion, a shadow register transfers its content to an active register at

the positive edge of layer-enable, or

when the output line number is equal to the line number specified in the register at 0x1F0[11:0]

An unwanted situation can arise when shadowing starts (as a result of the trigger) before the user has finished programming a complete sequence of register changes. To prevent this from happening, the complete register re-programming should be followed by a “FINISH” which should really trigger the shadowing. The “FINISH” is activated via re-writing a value of “1” to the Layer_Enable, which originally has value “1”. This is sometimes called re-hitting the Layer_Enable bit. By making use of this mechanism, any “UNFINISHED” programming is only ready for shadowing when the Layer_Enable is re-hit.

Remark: Once the shadowing is complete, the Layer upload bit is set again.

Figure 242 and Figure 243 illustrate the shadowing procedure.

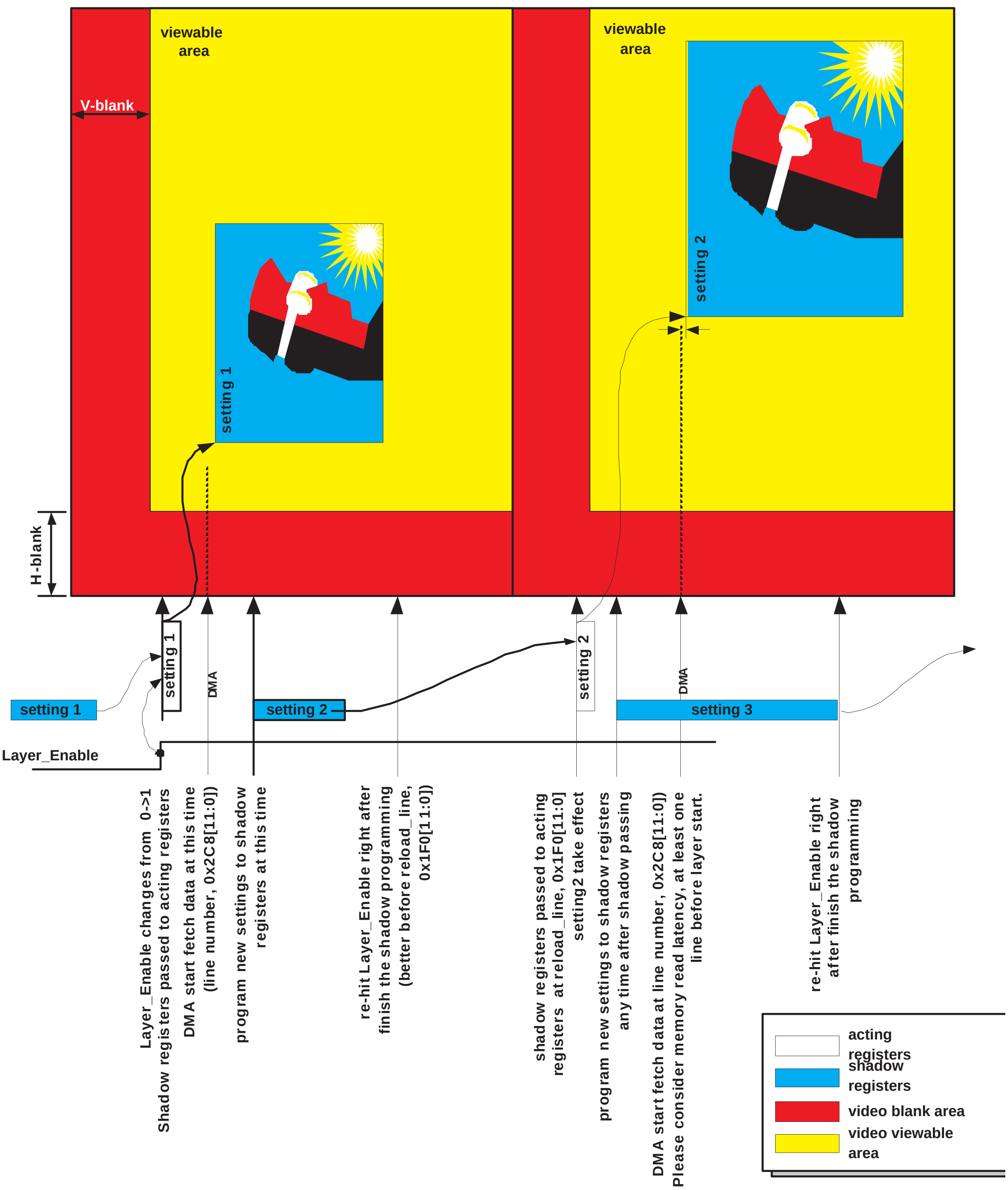

viewable area
viewable area
V-blank
H-blank
setting 1
setting 2
setting 1
setting 2
setting 3
DMA
DMA
Layer_Enable
Layer_Enable changes from 0->1
Shadow registers passed to acting registers
DMA start fetch data at this time (line number, 0x2C8[11:0])
program new settings to shadow registers at this time
re-hit Layer_Enable right after finish the shadow programming (better before reload_line, 0x1F0[11:0])
shadow registers passed to acting registers at reload_line, 0x1F0[11:0] setting2 take effect
program new settings to shadow registers any time after shadow passing
DMA start fetch data at line number, 0x2C8[11:0]) Please consider memory read latency, at least one line before layer start.
re-hit Layer_Enable right after finish the shadow programming
□ acting registers
🟦 shadow registers
🟥 video blank area
🟨 video viewable area

Fig 242.Shadow mechanism

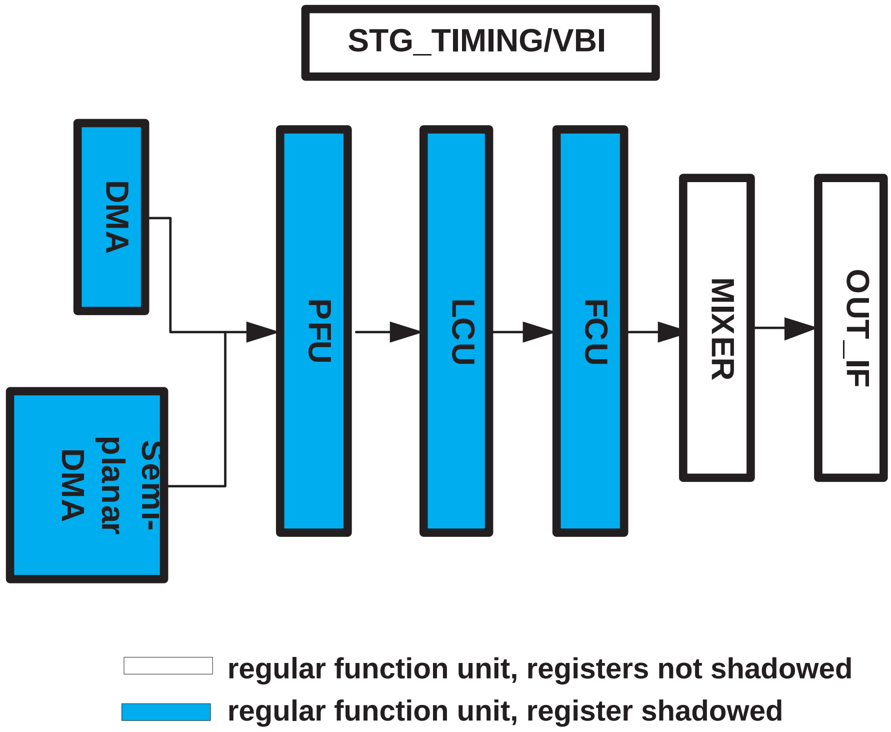

STG_TIMING/VBI
DMA
Semi-planar DMA
PFU
LCU
FCU
MIXER
OUT_IF
regular function unit, registers not shadowed
regular function unit, register shadowed

Fig 243.Shadowing of registers

Table 354 lists all shadowed registers

Table 354: Shadow registers

Register	Used by
Dummy Pixel Count (0x214)	Pixel Fetch Unit (PFU)
Layer Size (Initial) (0x234)	Pixel Fetch Unit (PFU)
Formats (0x2BC)	Pixel Fetch Unit (PFU)
Layer Source Address A (0x200,0x218)	DMA,Semi-Planar DMA
Layer Pitch A (0x204, 0x228)	DMA,Semi-Planar DMA
Layer Source Width (0x208, 0x22C)	DMA,Semi-Planar DMA
Layer Source Address B (0x20C,0x21C)	DMA,Semi-Planar DMA
Layer Pitch B (0x210, 0x228)	DMA,Semi-Planar DMA
Dummy Pixel Count (0x214)	Pixel Fetch Unit (PFU)
Line Increment (0x220, 0x224)	DMA, Semi-Planar DMA
Layer Start (0x230)	LCU
Layer Pixel Processing (0x23C)	Pixel Fetch Unit (PFU)
Layer Size (final) (0x2B4)--same as 0x234	Layer Control
Layer data manipulation (0x2B8)	Pixel Fetch Unit (PFU)
Start Fetch (0x2C8)	DMA

Fast access registers: The architecture of the VO MMIO access results in module dependent latencies for the various configuration registers. For most of the registers this does not present a problem since their content is usually rather static or only updated once per field/frame. Some registers, however require access with relatively low latency. VO configuration registers which can be accessed with low latency are:

Field_Info (0x1F8)

XY_Position (0x1FC)

Interrupt Status (0xFE0)

Interrupt Enable (0xFE4)

Interrupt Clear (0xFE8)

Interrupt Set (0xFEC)

Powerdown (0xFF4)

Module_ID (0xFFC)

Signature analysis: Signature Analysis is a feature where VO calculates a 16-bit signature on the upper 8 bits of each mixer output channel separately (Upper Path, Middle Path and Lower Path) using a specific CRC algorithm. The register, Signature3 (offset 0x58) has a bit, “sig_enable” which when set to 1 enables this feature. Once VO finishes calculating the signature, it sets the bit“sig_done” of the same register and the separate signatures can be read from the registers, “Signature1” (offset 0x050) and “Signature2” (offset 0x054).

VO can calculate the signature for the control signals (vysnc, hsync, blank etc.) along with the data.The result is reported in the Signature3 register (offset 0x58) as misc signature.

Signature analysis is used extensively in the emulation to validate the real-life use-cases that are very difficult to simulate on RTL.

Data endianness handling: The endinness is handled inside VO as follows:

VO gets the “big_endian” bit from the system which indicates that the data is being handled in the big-endian format. This system ”big_endian” bit is XOR’ed with the PF_ENDIAN bit (see register with offset 0x2BC) to generate an internal big_endian bit. If this internal big_endian bit is set and if the pixel format is packed YUY2 4:2:2 or packed UYVY 4:2:2, PFU inside VO swaps the input data word within a 16-bit boundary.

5.13.5 Application notes

5.13.5.1 Programming the STG

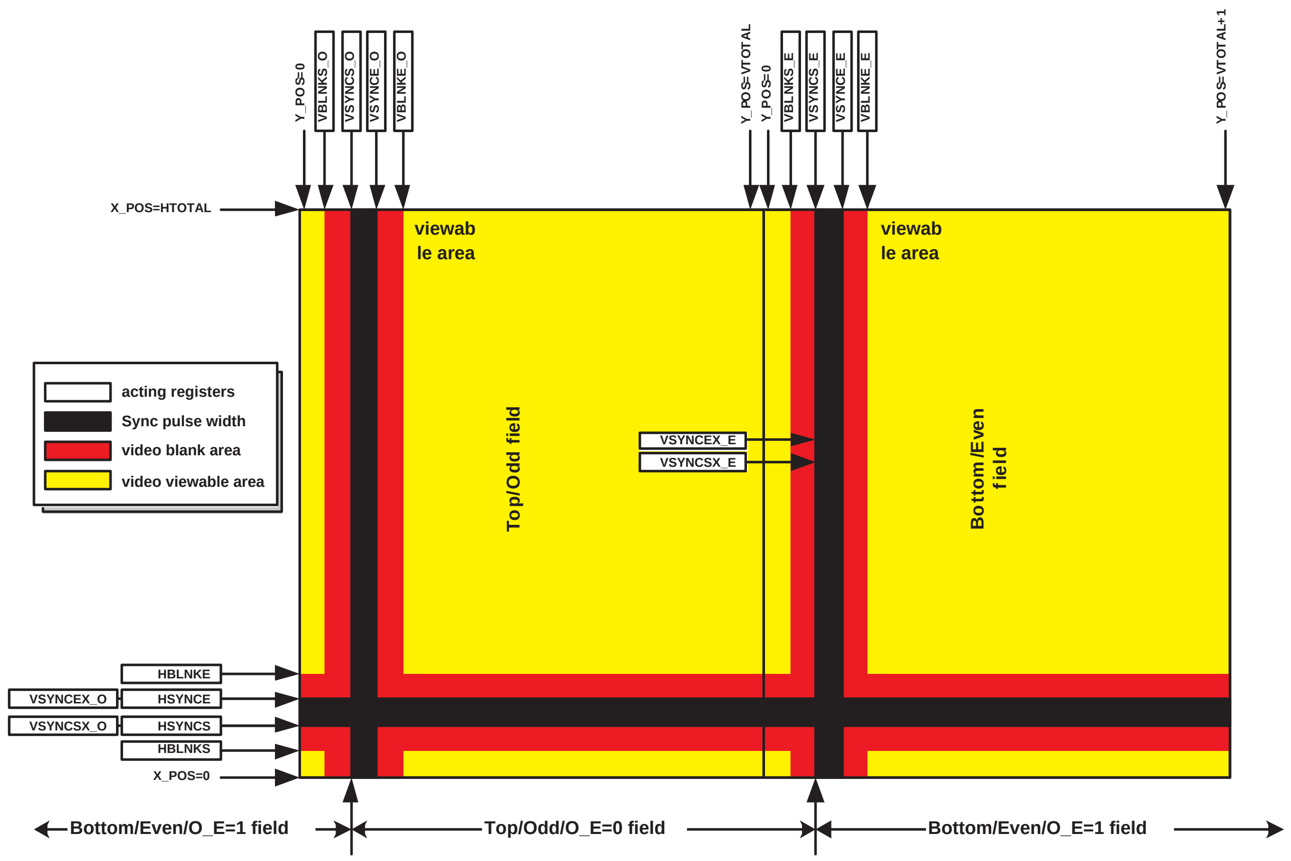


Fig 244.Typical video frame screen timing

Master mode: Whenever the bits settings {ExtFTrigger,ExtVTrigger,ExtHTrigger}=3'b000(These bits are from the control register with offset 0x020), the STG works in master mode.

Refer to Figure 244 for all the parameters that are defined below:

HTOTAL (0x000[27:16])= Total no. of pixels per line minus one

VTOTAL (0x000[11:0])= Total no. of lines per field minus one

HBLANKS/E (0x004)=Start/End pixel position of horizontal blanking interval

HSYNCS/E (0x00C)= Start/End pixel position of horizontal sync (Hsync)

VSYNC(S/E)_O (0x010)= Start/End line position of Vsync for top (odd) field in interlaced mode or progressive frame

VSYNC(S/E)X_O (0x028,0x2C)=Start/End pixel position of Vsync for top field in interlaced mode or progressive frame.

VSYNC(S/E)_E (0x0D4)= Start/End line position of Vsync for bottom (even) field only.

VSYNC(S/E)X_E (0x028,0x2C)=Start/End pixel position of Vsync for bottom field only

VBLNK(S/E)_O (0x008)= Start/End line position of vertical blanking interval for top (odd) field in interlaced mode or progressive frame.

VBLNK(S/E)_E (0x0D0)= Start/End line position of vertical blanking interval for bottom (even) field only

Because the STG coordinate system begins at (0,0), it is necessary to program certain registers to one less than the desired value. For example, if a scan line has 800 pixels in total, the horizontal total (HTOTAL) is set to 799 because 0-799 is a total of 800. The same applies to programming the vertical total (VTOTAL).

In the vertical domain, there are three main timing intervals to set: vertical active time, vertical blank time, and vertical sync time. The position of the vertical sync registers VSYNC(S/E)_O and VSYNC(S/E)_E define the vertical front and back porches. Note that the vertical sync interval (and therefore vertical blanking) is a minimum of one line in duration. Besides that for most normal operations in NTSC or PAL mode, VSYNC(S/E)_E should be set to the same value as VSYNC(S/E)_O.

Non-blanked area occurs when the currently active line is not within the vertical blanking interval or in the column of the horizontal blanking interval. Display layer can be programmed to reside on any portion of the screen. Any non-blanked screen position that does not have an active display layer pixel assigned to it results in the background color being displayed.

In addition to the standard progressive VO display mode, another mode called "Interlaced" can be switched on by setting the Interlaced control bit. In this mode the VTotal register no longer specifies the height of a frame but the height of a field. The field height alternates by one line depending on whether an odd or even field needs to be processed by the VO. Four registers are provided for this mode to specify the actual location of the vsync signal within a line in odd and even fields.

Slave mode: Whenever the bits settings {ExtFTrigger,ExtVTrigger,ExtHTrigger} !=3’b000(These bits are from the control register with offset 0x020), the STG works in slave mode.

STG slave mode is used for VO to synchronize with external timing signals, Fref, Vref and Href. VO recognizes two types of external timing signals defined in registers 0x020[30] TriggerType. Triggertype=0 indicates the three signals resembles those flags in SAV/EAV code. TriggerType=1 indicates industry standard V/HSYNC timing signals.

{ExtFTrigger, ExtVTrigger,ExtHTrigger} (= TriggerMode[2:0] in the table below) can be programmed in different combinations to achieve different level of slave control on STG. “111” is used whenever the input reference signals (Vref,Href,Fref) has reliable timing characteristics, e.g. interchip connecting of V/HSYNC. Otherwise, the signals deemed unstable can be disabled from triggering (presetting) X_POS or Y_POS counter.

Table 355: STG modes

TriggerMode[2:0]	O_E	Y_POS	X_POS	Comments
000	Generated[1][3]	Generated[4]	Generated[5]	Master Mode
001	Generated	Generated	Preset by Href[6]	
010	Generated	Preset by Vref[7]	Preset by Fref	
011	Generated	Preset by Vref	Preset by Href	
100	Preset by Fref[2]	Preset by Fref	Preset by Fref	
101	Preset by Fref[8]	Preset by Fref	Preset by Href	
110	Preset by Fref	Preset by Vref	Preset by Fref	
111	Preset by Fref	Preset by Vref	Preset by Href	Absolute Slave Mode

[1] Generated = to produce signals according to internal state regardless of external signals

[2] Preset = to reload preset values at active edge of external signals

[3] Generated O_E : O_E is produced according to VSYNCS_O and VSYNCS_E settings.

[4] Generated Y_POS: Y_POS is counting up whenever X_POS=HTOTAL. Y_POS=0~VTOTAL for top field and Y_POS=0~VTOTAL+1 for bottom field

[5] Generated X_POS: X_POS is counting up for every pixel clock. X_POS=0~HTOTAL

[6] Href presets X_POS: Active edge of Href preload X_POS with value specified in ExtTriggerX register (0x0D8)

[7] Vref presets Y_POS: Active edge of Vref preload Y_POS with value specified in ExtTriggerY register.(0x0D8)

[8] Fref presets O_E: negative edge of Fref preload O_E with value specified in ExtFieldSwap. (0x020)

Remark: In the slave mode, VO generates a new field/frame depending on the external VSYNC. The distance between two consecutive external VSYNCs can vary. The actual distance between two syncs depends on the use case.

V/HSYNC consideration (trigger type =1): For interlaced video in slave mode, if the leading edges of VSYNC and HSYNC are coincident, the STG recognizes this as top (odd) field. If the leading edge of VSYNC occurs mid-line, the field is bottom (even) field.

In a inter-chip video transmit and receive environment, the V/HSYNC mostly has good timing characteristics and the field can therefore be detected without any problem. However, if the video source is from a different or remote system, the coincident of edges may not hold true. Depending upon the timing path difference in between VSYNC or HSYNC, the misalignment may be up to several pixels. VO is designed to tolerate this kind of misalignment. By setting register ExtSYNCMisalign[3:0] (0x020[15:12]), VO provides a tolerable misalignment “window” of size 2*ExtSYNCMisalign[3:0]+1. If both edges occur within this window, the field is still recognized as top (odd) field. Set ExtSYNCMisalign[3:0] to 15 gives the largest window of 31 pixels.

Changing timing: All register settings to the STG take effect immediately and are not clock re- synchronized. (The start/stop bit is the exception. It takes effect immediately and is clock re-synchronized.) The only safe way to change screen timing is as follows:

Turn off the screen timing generator (STG).

Program all registers needed in the new display mode.

Turn the timing generator back on. In the process, the entire display pipeline is reset. The display layers gets reset, and the screen timing starts at the vertical total, which guarantees a complete vertical blank period and vertical sync signal at the start of any mode change.

5.13.5.2 Positioning the picture

Picture cropping: The VO provides clipping support at the edge of the defined H- and V-total. If a layer is positioned in a way that some part of it would exceed the overall screen dimensions, no wrapping occurs but the pixel layering in this area is marked as invalid, hence they are not being displayed. The layer size should be smaller than HTOTAL and VTOTAL.

Negative positioning: The VO also supports negative screen positions i.e., top and left side clipping of layers. For negative x and y layer start positions, the following equations should be used:

```
if StartX < 0 then
    StartX = Xtotal + 1 - ABS(StartX)
    set StartX sign bit
    StartY = Ydisplay - 1

else

    StartX = StartX
    StartY = StartY

if StartY < 0 then
    StartY = Ytotal + 1 - ABS(StartY)
    set StartY sign bit

else
    StartY = StartY
```

StartY and fine bit: The “FINE” bit is used to convert StartY from Frame to Field position in interlaced mode.

Fine=0 : No frame to field conversion for StartY, ie, StartY=100 displays the layer at STG_Y_POS=100 position.

Fine=1 : Do frame to field conversion for StartY, e.g. LayerNStartY=100 is converted to display at STG_Y_POS=100/2=50 position for top field/bottom field. However, LayerNStartY=101 is converted to display at STG_Y_POS=101/2=50 position for top field and STG_Y_POS=101/2+1=51 position for bottom field.

Fine=1 is recommended in interlaced mode.

Fine=0 is mandatory for progressive mode.

5.13.5.3 Setup sequence for VO

The following order is recommended for setting up the VO for a particular display scenario:

The screen timing generator setup is performed first since this is usually fixed for the target display. Once the screen timing is set up the screen timing generator should still stay disabled until all other settings are complete.

Once all layer specific functions are set up, the output interface needs to be programmed in order to correctly interface with the display controller chip. After performing all these tasks the screen timing generator may be enabled. Once running, the layer can be enabled. This concludes the VO setup. Once the VO is set up for a certain scenario and images are displayed, a number of operations can be reconfigure on the fly.

5.13.6 Register descriptions

If writes are performed to an unoccupied spot of all the registers (Unused bits) they are discarded and Reads will return zero.

All the Reserved bits in the R/W or W registers should be written, only the default values and the reads return the default values.

5.13.6.1 Register summary

Table 356: Register summary

Offset	Name	Description
0x0	TOTAL	
0x4	HBLANK	
0x8	VBLANK_O	
0xC	HSYNC	
0x10	VSYNC_O	
0x14	VINTERRUPT	
0x1C	DEFAULT_BACKGROUND_COLOR	
0x20	CONTROL	
0x28	INTLCTRL1	
0x2C	INTLCTRL2	
0x30	VBI_SRC_Address	
0x34	VBI_CTRL	
0x38	VBI_SENT_OFFSET	
0x3C	OUT_CTRL	
0x50	Signature1	
0x54	Signature2	
0x58	Signature3	
0xD0	VBLANK_E	
0xD4	VSYNC_E	
0xD8	Ext_Trigger_XY	
0xDC	Ext_trigger_Status_Select	
0xE0	Ext_Trigger_Status	
0xE4	Ext_SYNC_Alignment_Status	
0x1EC	STG_Shadow_Reload	
0x1F0	Shadow_Reload	
0x1F8	Field_Info	
0x1FC	XY_Position	
0x200	Layer_Source_address_A	Layer Source Address A (Packed/Semi Planar Y)

Table 356: Register summary *…continued*

Offset	Name	Description
0x204	Layer_Pitch_A	Layer Pitch A (Packed/Semi Planar Y)
0x208	Layer_Source_Width	Layer Source Width (Packed/Semi Planar-Y)
0x20C	Layer_Source_Address_B	Layer Source Address B (Packed/Semi Planar-Y)
0x210	Layer_Pitch_B	Layer Pitch B (Packed/Semi Planar Y)
0x214	Dummy_Pixel_Count	
0x218	Layer_Source_Address_A_Semi_Planar_UV	Layer Source Address A (Semi Planar UV)
0x21C	Layer_Source_Address_B_Semi_Planar_UV	Layer Source Address B (Semi Planar UV)
0x220	Line_Increment_Packed	Line Increment (Packed)
0x224	Line_Increment_Semi_Planar	Line Increment (Semi Planar)
0x228	Layer_Pitch_Semi_Planar	Layer Pitch (Semi Planar)
0x22C	Layer_Source_Width_Semi_Planar	Layer Source Width (Semi Planar)
0x230	Layer_Start	
0x234	Layer_Size_Initial	Layer Size (Initial)
0x23C	Layer_Pixel_Processing	
0x240	Layer_Status_Control	
0x2B4	Layer_size_final	Layer Size (final)
0x2B8	Layer_data_manipulation	
0x2BC	Formats	
0x2C8	Start_Fetch	
0x320	Layer_Solid_Color	
0x3F8	DTL_Valid_STATUS	
0x3FC	Layer_Boundary	
0xFE0	Interrupt_Status_NHP_VO	
0xFE4	Interrupt_Enable_NHP_VO	
0xFE8	Interrupt_Clear_NHP_VO	
0xFEC	Interrupt_Set_NHP_VO	
0xFF4	VO_Powerdown	
0xFFC	VO_Module_ID	

5.13.6.2 Register tables

Table 357: NHP_VO registers

Bit	Symbol	Access	Reset value	Description
Offset 0x0 - TOTAL				
31:28	Unused[3:0]	R/W	0x0	
27:16	Htotal[11:0]	R/W	0x000	Horizontal Total sets the number of horizontal pixels for the display. Total # of pixels per line = HTOTAL+1.
15:12	Unused2[3:0]	R/W	0x0	

Table 357: NHP_VO registers *…continued*

Bit	Symbol	Access	Reset value	Description
11:0	Vtotal[11:0]	R/W	0x000	Vertical Total sets the number of vertical pixels for the display. For Progressive: Total # of lines per frame = VTOTAL +1. For Interlace: Total # of lines for odd field = VTOTAL +1. Total # of lines for even field = VTOTAL +2.
Offset 0x4 - HBLANK				
31:28	Unused[3:0]	R/W	0x0	
27:16	Hblanks[11:0]	R/W	0x000	Horizontal Blank Start sets the pixel location where horizontal blanking starts. Limitation: HTOTAL+2 >= HBLANKS >= 2
15:12	Unused2[3:0]	R/W	0x0	
11:0	Hblnke[11:0]	R/W	0x000	Horizontal Blank End sets the pixel location where horizontal blanking ends. Limitation: HTOTAL >= HBLNKE >=0
Offset 0x8 - VBLANK_O				
31:28	Unused[3:0]	R/W	0x0	
27:16	Vblanks_o[11:0]	R/W	0x000	Vertical Blank Start sets the pixel location where vertical blanking starts for the top field in the interlace mode or a frame in progressive mode.. Limitation: VTOTAL+1 >=VBLANKS >=1.
15:12	Unused2[3:0]	R/W	0x0	
11:0	Vblanke_o[11:0]	R/W	0x000	Vertical Blank End sets the pixel location where vertical blanking ends for a progressive frame or, if interlace, start for the top field. Limitation: VTOTAL >= VBLANKE >=0.
Offset 0xC - HSYNC				
31:28	Unused[3:0]	R/W	0x0	
27:16	Hsyncs[11:0]	R/W	0x000	Horizontal Sync Start sets the pixel location where horizontal sync starts. Limitation: HTOTAL+2 >= HSYNCS >=2.
15:12	Unused2[3:0]	R/W	0x0	
11:0	Hsynce[11:0]	R/W	0x000	Horizontal Sync End sets the pixel location where horizontal sync ends. Limitation: HTOTAL >= HSYNCE >=0.
Offset 0x10 - VSYNC_O				
31:28	Unused[3:0]	R/W	0x0	
27:16	Vsyncs_o[11:0]	R/W	0x000	Vertical Sync Start sets pixel location where Vertical sync starts for a progressive frame or, if interlace, start for the top field. Limitation: VTOTAL+1 >= VSYNCS_O >=1.
15:12	Unused2[3:0]	R/W	0x0	
11:0	Vsynce_o[11:0]	R/W	0x000	Vertical Sync End sets pixel location where Vertical sync ends for top field in interlaced mode or a frame in progressive mode. Limitation: VTOTAL >= VSYNCE_O >=0.
Offset 0x14 - VINTERRUPT				
31:28	Unused[3:0]	R/W	0x0	
27:16	Vlinta[11:0]	R/W	0x000	Vertical Line Interrupt A sets a vertical line number where an interrupt will be generated when the scan line matches this value. The interrupt is monitored by the Event Monitor (EVM). Limitation: VTOTAL >= VLINTA >=0.

Table 357: NHP_VO registers *...continued*

Bit	Symbol	Access	Reset value	Description
15:12	Unused2[3:0]	R/W	0x0	
11:0	Vlintb[11:0]	R/W	0x000	Vertical Line Interrupt B sets a vertical line number where an interrupt will be generated when the scan line matches this value. The interrupt is monitored by the Event Monitor (EVM).Limitation: VTOTAL >= VLINTB >=0.
Offset 0x1C - DEFAULT_BACKGROUND_COLOR				
31:24	Unused[7:0]	R/W	0x00	
23:16	Upper_default_background[7:0]	R/W	0x00	Background color of the upper channel (Y)
15:8	Middle_default_background[7:0]	R/W	0x00	Background color of the middle channel (U)
7:0	Lower_default_background[7:0]	R/W	0x00	Background color of the lower channel (V)
Offset 0x20 - CONTROL				
31	Reserved	R/W	0x0	Reserved
30	Triggertype	R/W	0x0	Select the external trigger method in slave mode. 0: SAV/EAV mode; Fref=FID;Vref=VBLANK;Href= HBLANK mode 1: Explicit Sync edge detection mode;Fref=derived from VSYNC,HSYNC;Vref=VSYNC;Href=HSYNC
29	Interlaced	R/W	0x0	Interlaced mode bit Field height = VTotal for odd fields. Field height = VTotal+1 for even fields. O_E flag = 0 for odd (bottom) fields O_E flag =1 for even (top) fields 0: Non-interlaced mode; VTotal=frame height. 1: Interlaced mode
28	Blankpol	R/W	0x0	Output BLANK Polarity 0: Positive blank 1: Negative blank
27	Exthsyncpol	R/W	0x0	Input external Hsync polarity (used in slave mode to trigger STG). Applies only on TriggerMode=1 (VSYNC/HSYNC mode). 0 : low active 1 : high active
26	Hsyncpol	R/W	0x0	Output HSYNC Polarity 0: Positive going 1: Negative going
25	Extvsyncpol	R/W	0x0	Input external Vsync polarity (used in slave mode to trigger STG). Applies only on TriggerMode=1 (VSYNC/HSYNC mode). 0 : low active 1 : high active
24	Vsyncpol	R/W	0x0	Output VSYNC Polarity 0: Positive going 1: Negative going
23:22	Unused[1:0]	R/W	0x0	

Table 357: NHP_VO registers *…continued*

Bit	Symbol	Access	Reset value	Description
21	Extfieldswap	R/W	0x0	to swap the detected Field ID in slave mode. Applies to all TriggerMode.
20	Blankctl	R/W	0x0	Blank Control allows either normal blanking or forces blanking to occur immediately. 0: Blank output is equivalent to BlankPol setting 1: Normal Blank
19	Exthtrigger	R/W	0x0	Input External HSYNC/HREF active edge triggers the pixel counter to value of ExtTriggerX.The active edge for HSYNC is defined in ExtHSYNCPol register bit whereas the active edge for HBLANK is always rising edge
18	Hsyncctl	R/W	0x0	HSYNC Control enables or disables the horizontal sync output of the chip. 0: HSYNC output is equivalent to HSYNCPol setting 1: Enable
17	Extvtrigger	R/W	0x0	Input External Vsync active edge triggers the line counter to value of ExtTriggerY.The active edge for VSYNC is defined in ExtVSYNCPol register bit whereas the active edge for VBLANK is always rising edge
16	Vsyncctl	R/W	0x0	VSYNC Control enables or disables vertical sync output of the chip. 0: VSYNC output is equivalent to VSYNCPol setting 1: Enable
15:12	Extsyncmisalign[3:0]	R/W	0xF	the maximum misalignment errors allowed between vsync and hsync. 0 means the input VSYNC and HSYNC active edge has to be perfectly aligned to be detected as top field. Maximum number 31 to define a misalignment window of 63 pixels for HSYNC and VSYNC.
11:10	Reserved2[1:0]	R/W	0x0	
9:8	Auxctrl2[1:0]	R/W	0x0	[9:8] = 2b00 => output acts like a composite blanking signal controlled with BlankPol and BlankCtl [9:8] = 2b01 => output Odd/Even signal in interlaced modes, zero in progressive modes [9:8] = 2b10 => reserved [9:8] = 2b11 => reserved
7:6	Reserved3[1:0]	R/W	0x0	
5:4	Auxctrl1[1:0]	R/W	0x0	[5:4] = 2b00 => output acts like a composite blanking signal controlled with BlankPol and BlankCtl [5:4] = 2b01 => outputs Odd/Even signal in interlaced modes, zero in progressive modes [5:4] = 2b10 => reserved [5:4] = 2b11 => reserved
3	Data_oen	R/W	0x0	Output enable control for video data bus 0: Data outputs enabled (normal operation) 1: Data outputs disabled (tri-state)
2	Reserved4	R/W	0x0	

Table 357: NHP_VO registers *…continued*

Bit	Symbol	Access	Reset value	Description
1	Extftrigger	R/W	0x0	Input External derived Field_ID falling edge triggers the O_E (offset 0x1FC[31])to value of ExtFieldSwap. Please keep in mind that the three bits {ExtFTrigger, ExtVTrigger,ExtHTrigger} = TriggerMode[2:0] can be programmed to different combination to achieve different level of control on STG. 000 : Master mode (Generated O_E,Y_POS,X_POS) 001 : Generated O_E, Generated Y_POS, Href presets X_POS 010 : Generated O_E. Vref presets Y_POS. Fref presets X_POS 011 : Generated O_E, Vref preset Y_POS. Href presets X_POS 100 : Fref presets O_E,Y_POS,X_POS 101 : Fref presets O_E,Y_POS. Href presets X_POS 110 : Fref presets O_E,X_POS. Vref presets Y_POS. 111 : Fref presets O_E,Vref presets Y_POS. Href presets X_POS The Generated here means to produce signals according to internal counter. Preset here means counters reload preset values at active edge of event. Generated O_E : O_E is produced according to VSYNCS_O and VSYNCS_E settings. Generated Y_POS: Y_POS is counting up whenever X_POS=HTOTAL. Y_POS=0~VTOTAL for top field and Y_POS=0~VTOTAL+1 for bottom field. Generated X_POS: X_POS is counting up for every pixel clock. X_POS=0~HTOTAL. Href presets X_POS: Active edge of Href preload X_POS with value specified in Trigger_X_POS register. Vref presets VCNT: Active edge of Vref preload Y_POS with value specified in Trigger_Y_POS register. Fref presets O_E: negative edge of Fref preload O_E with value specified in ExtFieldSwap.
0	Stg_enable	R/W	0x0	Timing generator ENABLE 0: Disable 1: Enable (VO is reset on the transition from disable to enable)
Offset 0x28 - INTLCTRL1				
31:28	Unused[3:0]	R/W	0x0	
27:16	Vsyncsx_e[11:0]	R/W	0x000	Horizontal offset for VSYNC start even field (interlaced mode only) Vsync appears at INT_START_E + 1.
15:12	Unused2[3:0]	R/W	0x0	
11:0	Vsyncsx_o[11:0]	R/W	0x000	Horizontal offset for VSYNC start odd field (interlaced mode only) Vsync appears at INT_START_O + 1.
Offset 0x2C - INTLCTRL2				
31:28	Unused[3:0]	R/W	0x0	
27:16	Vsyncex_e[11:0]	R/W	0x000	Horizontal offset for VSYNC end even field (interlaced mode only)
15:12	Unused2[3:0]	R/W	0x0	
11:0	Vsyncex_o[11:0]	R/W	0x000	Horizontal offset for VSYNC end odd field (interlaced mode only)
Offset 0x30 - VBI_SRC_Address				

Table 357: NHP_VO registers *...continued*

Bit	Symbol	Access	Reset value	Description
31:0	Vbi_src_addr[31:0]	R/W	0x00000000	VBI data source address
Offset 0x34 - VBI_CTRL				
31:1	Unused[30:0]	R	0x00000000	
0	Vbi_en	R/W	0x0	Enable VBI data fetch engine.
Offset 0x38 - VBI_SENT_OFFSET				
31:12	Unused[19:0]	R/W	0x00000	
11:0	Vbi_sent_offset_value[11:0]	R/W	0x000	This programming value specifies the number of lines to add to the linecnt value in the packet identifier.
Offset 0x3C - OUT_CTRL				
31:28	Unused[3:0]	R/W	0x0	
27:20	Reserved[7:0]	R/W	0xFC	
19	Unused2	R/W	0x0	
18	Qualifier	R/W	0x0	 0: hsync is put out 1: slice qualifier is put out
17:16	Outmode[1:0]	R/W	0x0	00 = output interface runs is single d1 mode 01 = output interface runs in double-d1 mode (middle:Y, lower:UV) 10,11 = reserved.
15	Reserved2	R/W	0x0	Reserved
14	Reserved3	R/W	0x0	
13	Out_clk_ctl	R/W	0x1	NHP_VO output clock selector (Refer to Table 6 for DV_CLK1) 0: dv_clk_out is non-inverted version of the input clock 1: dv_clk_out is inverted version of the input clock
12	Reserved4	R/W	0x1	Reserved (Note the default value is not 0)
11	Reserved5	R/W	0x0	
10	D1_mode	R/W	0x1	1 = 4:2:2 D1 mode, 0 = Reserved 0: 4:4:4 D1 mode 1: 4:2:2 D1 mode
9:8	Reserved6[1:0]	R/W	0x0	
7	Unused3	R/W	0x0	
6:4	Reserved7[2:0]	R/W	0x0	
3	Unused4	R/W	0x0	
2:0	Reserved8[2:0]	R/W	0x0	
Offset 0x50 - Signature1				
31:16	Middle_signature[15:0]	R	0x0000	Middle path signature
15:0	Lower_signature[15:0]	R	0x0000	Lower path signature

Table 357: NHP_VO registers *…continued*

Bit	Symbol	Access	Reset value	Description
Offset 0x54 - Signature2				
31:16	Reserved[15:0]	R/W	0x0000	
15:0	Upper_signature[15:0]	R	0x0000	Upper path signature
Offset 0x58 - Signature3				
31:16	Misc_signature[15:0]	R	0x0000	Other signature
15:9	Unused[6:0]	R/W	0x00	
8	Sig_done	R	0x0	Signature done
7:6	Unused2[1:0]	R/W	0x0	
5:3	Sig_select[2:0]	R/W	0x0	Signature select, 3'b000 = layer output selected for signature analysis. All other values are reserved.
2:1	Unused3[1:0]	R/W	0x0	
0	Sig_enable	R/W	0x0	Signature enable
Offset 0xD0 - VBLANK_E				
31:28	Unused[3:0]	R/W	0x0	
27:16	Vblanks_e[11:0]	R/W	0x000	Vertical Blank Start sets the pixel location where vertical blanking starts for the bottom field in interlaced mode only.. Limitation: VTOTAL+1 >=VBLANKS >=1. Value of VTOTAL+2 is equivalent to Y_POS=0.
15:12	Unused2[3:0]	R/W	0x0	
11:0	Vblanke_e[11:0]	R/W	0x000	Vertical Blank End sets the pixel location where vertical blanking ends for the bottom field in interlaced mode only. Limitation: VTOTAL >= VBLANKE >=0.
Offset 0xD4 - VSYNC_E				
31:28	Unused[3:0]	R/W	0x0	
27:16	Vsyncs_e[11:0]	R/W	0x000	Vertical Sync Start sets pixel location where Vertical sync starts for bottom field in interlaced mode only. Limitation: VTOTAL+1 >= VSYNCS_E >=1. Value of VTOTAL+2 is equivalent to Y_POS=0
15:12	Unused2[3:0]	R/W	0x0	
11:0	Vsynce_e[11:0]	R/W	0x000	Vertical Sync End sets pixel location where Vertical sync ends for bottom field in interlaced mode only Limitation: VTOTAL >= VSYNCE_E >=0.
Offset 0xD8 - Ext_Trigger_XY				
31:28	Unused[3:0]	R/W	0x0	
27:16	Exttriggerx[11:0]	R/W	0x000	The x position counter reset/trigger value whenever stg_enable=0 or external trigger happens in slave mode.
15:12	Unused2[3:0]	R/W	0x0	
11:0	Exttriggery[11:0]	R/W	0x000	The y position counter reset/trigger value whenever stg_enable=0 or external trigger happens in slave mode.
Offset 0xDC - Ext_trigger_Status_Select				

Table 357: NHP_VO registers *…continued*

Bit	Symbol	Access	Reset value	Description
31:6	Reserved[25:0]	R/W	0x0000000	
5:4	ExtHTriggerStatus _sel[1:0]	R/W	0x0	Select the kind of status shown in the next registers (0E0), whenever there is an external H Trigger event. Please refer to register offset 0x0E0 for more information
3:2	Reserved2[1:0]	R/W	0x0	
1:0	ExtVTriggerStatus _sel[1:0]	R/W	0x0	Select the kind of status shown in the next registers (0E0), whenever there is an external V Trigger event. Please refer to register offset 0x0E0 for more information
Offset 0xE0 - Ext_Trigger_Status				
31:29	Reserved[2:0]	R	0x0	
28:16	ExtHTriggerStatus [12:0]	R	0x0000	Depending upon settings of the previous registers (0DC), this register can have different status. ExtHTriggerStatus_sel[1:0]=... 00: signed value. (H Trigger X_POS[11:0]) - (ExtTriggerX)+1. Whenever ExtHTrigger=1 (register offset 0x20) and ExtTriggerx=0 (register offset 0x0D8), this value represent the total length in pixels in between two external H triggers. For any settings otherwise, this value represents the discrepancy between the external line length and the internal HTOTAL settings. 0 means they are exactly match, positive 1 means external line length is one pixel more than HTOTAL and so on. 01: Difference between current line and the previous in pixels. 10: Ext V trigger O_E and X_POS. 11: Ext V trigger O_E and Y_POS
15:13	Reserved2[2:0]	R	0x0	Unused
12:0	ExtVTriggerStatus [12:0]	R	0x0000	Depending upon settings of the previous registers (0DC), this register can have different status. ExtVTriggerStatus_sel[1:0]=... 00: signed value. (V Trigger Y_POS[11:0]) - (ExtTriggerY)+1. Whenever ExtVTrigger=1 (register offset 0x20) and ExtTriggerY=0 (register offset 0x0D8), this value represent the total length in pixels in between two external V triggers. For any settings otherwise, this value represents the discrepancy between the external field/frame length and the internal VTOTAL settings. 0 means they are exactly match, positive 1 means external field/frame length is one line more than VTOTAL and so on. 01: signed value. (current Vtrigger Y_POS) - (Previous Vtrigger Y_POS). Difference between current field/frame length and the previous in lines. 10: Ext V trigger at {O_E, X_POS[11:0]}. 11: Ext V trigger at {O_E, Y_POS[11:0]}
Offset 0xE4 - Ext_SYNC_Alignment_Status				
31:5	Reserved[26:0]	R	0x0000000	
4	VsyncLead	R	0x0	This bit indicates that Vsync leads Hsync when read as 1.
3:0	SyncMisalignment [3:0]	R	0x0	This 4-bit value indicates the misalignment between VSYNC and HSYNC. 0 means perfect alignment, 1 means misalignment by 1 pixel and so on.

Table 357: NHP_VO registers *…continued*

Bit	Symbol	Access	Reset value	Description
Offset 0x1EC - STG_Shadow_Reload				
31	StgShadowEnable	R/W	0x0	Enable the STG settings shadowing feature.
30:12	Reserved[18:0]	R/W	0x00000	
11:0	StgShadowLine[11:0]	R/W	0x000	STG shadow reload line number where new settings in STG shadow-able registers are taking effect. This feature only available when StgShadowEnable=1.
Offset 0x1F0 - Shadow_Reload				
31	Reserved	R/W	0x0	should always set to 0
30	Reserved2	R/W	0x0	
29:12	Unused[17:0]	R	0x00000	
11:0	Reload_line[11:0]	R/W	0x000	line count number where shadow reload occurs. Please make sure reload line is set to a position earlier than layer start Y position given in 0x230.
Offset 0x1F8 - Field_Info				
31:3	Unused[28:0]	R/W	0x00000000	
2:0	Field_id[2:0]	R	0x0	Field_ID is reset by disabling the screen timing generator Field_ID is incremented with each rising edge of VSYNC and wraps around after reaching the value 0x7 which yields a sequence of 8 fields which could be differentiated by using the Field_ID register.
Offset 0x1FC - XY_Position				
31	O_e	R	0x0	Odd/Even flag status (interlaced mode) 0: First field (odd/top field) 1: Second field (even/bottom field)
30:28	Unused[2:0]	R/W	0x0	
27:16	Y_pos[11:0]	R	0x000	Current vertical position of screen timing generator
15:12	Unused2[3:0]	R/W	0x0	
11:0	X_pos[11:0]	R	0x000	Current horizontal position of screen timing generator
Offset 0x200 - Layer_Source_address_A				
31:0	Layer_source_start_address_a[31:0]	R/W	0x00000000	Layer Source Data Start Address A in bytes. This sets starting address A for data transfers from the linear Frame Buffer memory to the Layer . For semi planar mode this address points to the Y plane. Note: It should be aligned on a 128-byte boundary for memory performance reasons. It has to be 8-byte aligned.
Offset 0x204 - Layer_Pitch_A				
31:23	Unused[8:0]	R/W	0x000	

Table 357: NHP_VO registers *…continued*

Bit	Symbol	Access	Reset value	Description
22:0	Layer_data_pitch_a[22:0]	R/W	0x000000	Layer Source Data Pitch B in bytes. This sets pitch A for data transfers from the linear Frame Buffer memory to the Layer. For semi planar mode this determines the pitch for the Y plane. The value has to be rounded up to the next 64-bit word.
Offset 0x208 - Layer_Source_Width				
31:13	Unused[18:0]	R/W	0x00000	
12:0	Layer_source_data_width[12:0]	R/W	0x0000	Layer source width in bytes. For semi planar mode this determines the source data with in bytes for the Y plane. The value has to be rounded up to the next 64-bit word.
Offset 0x20C - Layer_Source_Address_B				
31:0	Layer_source_start_address_b[31:0]	R/W	0x00000000	Layer Source Data Start Address B in bytes. This sets starting address B for data transfers from the linear Frame Buffer memory to the Layer. For semi planar mode this address points to the Y plane. Note : It should be aligned on a 128-byte boundary. It has to be 8-byte aligned.
Offset 0x210 - Layer_Pitch_B				
31:23	Unused[8:0]	R/W	0x000	
22:0	Layer_data_pitch_b[22:0]	R/W	0x000000	Layer Source Data Pitch B in bytes sets pitch B for data transfers from the linear Frame Buffer memory to the Layer. For semi planar mode this determines the pitch for the Y plane. The value has to be rounded up to the next 64-bit word.
Offset 0x214 - Dummy_Pixel_Count				
31:8	Unused[23:0]	R/W	0x000000	
7:0	Dcnt[7:0]	R/W	0x00	Number of dummy pixels to be inserted between layer video lines
Offset 0x218 - Layer_Source_Address_A_Semi_Planar_UV				
31:0	Layer_source_address_asemi_planar_uv[31:0]	R/W	0x00000000	Layer Source Data Start Address A in bytes. This sets starting address A for data transfers from the linear Frame Buffer memory to the Layer. This Register holds the source address for the UV plane in semi planar modes. Note: It should be aligned on a 128-byte boundary. It has to be 8-byte aligned.
Offset 0x21C - Layer_Source_Address_B_Semi_Planar_UV				
31:0	Layer_source_address_bsemi_planar_uv[31:0]	R/W	0x00000000	Layer Source Data Start Address B in bytes. This sets starting address B for data transfers from the linear Frame Buffer memory to the Layer. This Register holds the source address for the UV plane in semi planar modes. Note: It should be aligned on a 128-byte boundary. It has to be 8-byte aligned.
Offset 0x220 - Line_Increment_Packed				
31:16	Unused[15:0]	R/W	0x0000	
15:0	Line_increment_packed_repeat[15:0]	R/W	0xFFFF	This register determines whether a layer line is repeatedly fetched from memory or not. Round Down(2^{16}/(Line Increment Packed))= #of times the same line is fetched i.e., 0x8000H would fetch each line exactly twice (line doubling).
Offset 0x224 - Line_Increment_Semi_Planar				

Table 357: NHP_VO registers *…continued*

Bit	Symbol	Access	Reset value	Description
31:16	Unused[15:0]	R/W	0x0000	
15:0	Line_increment_semi_planar_repeat[15:0]	R/W	0xFFFF	This register determines whether a layer line is repeatedly fetched from memory or not. Round Down(2^{16}/(Line Increment Semi Planar))= #of times the same line is fetched i.e., 0x8000H would fetch each line exactly twice (line doubling).
Offset 0x228 - Layer_Pitch_Semi_Planar				
31:23	Unused[8:0]	R/W	0x000	
22:0	Layer_data_pitch_semi_planar[22:0]	R/W	0x000000	Layer Source Data Pitch in bytes. This sets pitch for data transfers from the linear Frame Buffer memory to the Layer for semi planar modes. The value is used independent of whether buffer A or B is used. The value has to be rounded up to the next 64-bit word.
Offset 0x22C - Layer_Source_Width_Semi_Planar				
31:13	Unused[18:0]	R/W	0x00000	
12:0	Layer_source_widthsemi_planar[12:0]	R/W	0x0000	Layer source width in bytes for semi planar modes. The value is used independent of whether buffer A or B is used. The value has to be rounded up to the next 64-bit word.
Offset 0x230 - Layer_Start				
31	Fine	R/W	0x0	For interlaced modes, to convert StartY from Frame to Field position Fine=0 : No frame to field conversion for StartY, ie, StartY=100 will display the layer at STG_Y_POS=100 position. Fine=1 : Do frame to field conversion for StartY, eg. LayerNStartY=100 will be converted to display at STG_Y_POS=100/2=50 position for top field/bottom field. However, LayerNStartY=101 will be converted to display at STG_Y_POS=101/2=50 position for top field and STG_Y_POS=101/2+1=51 position for bottom field. Fine=1 is recommended in interlaced mode. Fine=0 is mandatory for progressive mode.
30:29	Unused[1:0]	R/W	0x0	
28:16	Layerstartx[12:0]	R/W	0x0000	Layer Start x position (from zero at left edge) in pixels. Negative X start position is possible.
15:13	Unused2[2:0]	R/W	0x0	

Table 357: NHP_VO registers *…continued*

Bit	Symbol	Access	Reset value	Description
12:0	Layerstarty[12:0]	R/W	0x0000	Layer Start y position (from zero at top) in lines. Negative Y position is allowed. Note: In interlaced modes the following rules apply: Fine=0 : LayerStartY is always relative to frame position, ie, LayerStartY=100 will display the layer at STG_Y_POS=100 position. Fine=1 : LayerStartY is always relative to field position, ie. LayerStartY=100 will be translated to display layer at STG_Y_POS=100/2=50 position. Fine=1 is recommended in interlaced mode. Fine=0 is recommended in progressive mode. Whenever layer y position is changed, please make sure other y position sensitive register settings are still satisfied, such as : start fetch register 10E2C8, shadow reload position 10E1F0 layer start field register 10E23C (for interlaced mode)
Offset 0x234 - Layer_Size_Initial				
31:28	Unused[3:0]	R/W	0x0	
27:16	Layerheight[11:0]	R/W	0x000	Layer height in lines for (frame/field) for progressive and interlaced respectively.
15:12	Unused2[3:0]	R/W	0x0	
11:0	Layerwidth[11:0]	R/W	0x000	Initial Layer width, in pixels. Limitation: LayerWidth > 2.
Offset 0x23C - Layer_Pixel_Processing				
31:6	Unused[25:0]	R/W	0x0000000	
5	Buffer_toggle	R/W	0x0	This bit controls the DMA buffer mode: 0: No buffer toggle, always fetch from buffer spec A. 1: Always toggle between buffer A and B (A=odd field, B=even field).
4	Layer_start_field	R/W	0x1	Field in which the layer gets actually enabled once the Layer_Enable bit is set. This bit is used to invert the internal odd/even signal. If the result of the operation Layer_Start_Field x or OE is true the layer is enabled, otherwise the layer stays disabled until the OE signal changes. In non-interlaced modes: this bit is ignored. In interlaced modes: LayerStartY (10E230) >= 0, set this bit to 0 LayerStartY (10E230) < 0, set this bit to 1
3:0	Reserved[3:0]	R/W	0x0	
Offset 0x240 - Layer_Status_Control				
31:10	Unused[21:0]	R/W	0x000000	
9	Layer_upload	R	0x0	This bit indicates if the register upload into the shadow area is still in progress. 0: Upload in progress, DO NOT reprogram any registers as the results are undetermined 1: New register upload possible, previous upload is complete
8:1	Unused2[7:0]	R/W	0x00	

Table 357: NHP_VO registers *…continued*

Bit	Symbol	Access	Reset value	Description
0	Layer_enable	R/W	0x0	This register reads always 0 if the screen timing generator is not enabled 0: Disable the layer 1: Enable the layer
Offset 0x2B4 - Layer_size_final				
31:12	Unused[19:0]	R/W	0x00000	
11:0	Layerwidth_final[11:0]	R/W	0x000	Final Layer width, in pixels This value should be same as the LayerWidth programmed in the register with offset 0x10E234(bits 11:0)
Offset 0x2B8 - Layer_data_manipulation				
31:24	Unused[7:0]	R/W	0x00	
23:9	Reserved[14:0]	R/W	0x0000	
8:6	Unused2[2:0]	R/W	0x0	
5:3	Pf_offset2[2:0]	R/W	0x0	Defines pixel offset (in bytes) within a multi pixel 64 bit words, for channel 2 for semi-planar mode. 0 to 7 for 8-bit YUV 4:2:2 semi-planar The number will be truncated to the closest even number for the channel 2.
2:0	Pf_offset1[2:0]	R/W	0x0	Defines pixel offset (in bytes) within a multi pixel 64 bit words. 0, 2, 4 or 6 for 8-bit (16bpp) packed YUV 4:2:2 0 to 7 for all the other formats.
Offset 0x2BC - Formats				
31:14	Unused[17:0]	R/W	0x00000	
13	Pf_endian	R/W	0x0	Input format endianness 0: Same as system endianness 1: Opposite of system endianness
12	Unused2	R/W	0x0	
11:10	Reserved[1:0]	R/W	0x0	
9	Unused3	R/W	0x0	
8	Reserved2	R/W	0x0	
7:0	Pf_ipfmt[7:0]	R/W	0x00	Input Formats 08 (hex) = YUV 4:2:2/4:2:0* semi-planar A0 (hex) = packed YUY2 4:2:2 A1 (hex) = packed UYVY 4:2:2 * See register, 0x10E220 for details on enabling 4:2:0 mode
Offset 0x2C8 - Start_Fetch				
31	Enable	R/W	0x0	Set this bit to delay the DMA data fetch timing until line number specified in bit 11:0 is reached. If disabled, DMA will pre-fetch data for the next field at the end of current field. Please note that if NHP_VO is operating as a downstream receiver in a streaming mode, eg. MBS->VIP->NHP_VO, this feature has to be enabled to prevent from early underflow.
30:28	Unused[2:0]	R	0x0	unused

Table 357: NHP_VO registers *…continued*

Bit	Symbol	Access	Reset value	Description
27:16	Flushcount[11:0]	R/W	0x030	The number of flush pixels to be inserted after the end of a field. If Start Fetch is enabled this register must contain a large enough value to flush all pixels out of the pipeline after the last pixel entered the pixel formatter. (approx. 50)
15:12	Unused2[3:0]	R	0x0	
11:0	Fetch_start[11:0]	R/W	0x000	If enabled (by setting bit 31 to 1), the data fetched from memory will be delayed until line number set here is reached, i.e. the data pre-fetch is disabled. The number given here must be set to a value earlier in Y position than LayerStartY in 10E230 to prevent from layer underflow. In non-interlaced mode: this value is relative to FRAME position. For example, if LayerStartY=100, a start fetch position of 98 is deemed earlier position. In interlaced mode: this value is relative to FIELD position. For example, if LayerStartY=100, a start fetch position of 52 is deemed one line too late to start the fetch, because LayerStartY=100 is equivalent to field position 100/2=50. Therefore, a start fetch position of 48 is a proper one.
Offset 0x320 - Layer_Solid_Color				
31	SC_enable	R/W	0x0	This bit enables the replacement of the layer input by the specified color. 1: Replace 0: Use layer input.
30:24	Unused[6:0]	R/W	0x00	
23:16	Upper_solid_color [7:0]	R/W	0x00	Upper channel of the replacement of the color (Y); unsigned and the range is 16 to 235
15:8	Middle_solid_colo r[7:0]	R/W	0x00	Middle channel of the replacement of the color (U); unsigned and the range is 16 to 240
7:0	Lower_solid_color [7:0]	R/W	0x00	Lower channel of the replacement of the color (V); unsigned and the range is 16 to 240
Offset 0x3F8 - DTL_Valid_STATUS				
31:28	Unused[3:0]	R	0x0	
27:16	FirstPixelValid_X[11:0]	R	0x000	Status showing the X_POS where the first pixel in a field/frame is valid from DTL.
15:12	Unused2[3:0]	R	0x0	
11:0	FirstPixelValid_Y[11:0]	R	0x000	Status showing the Y_POS where the first pixel in a field/frame is valid from DTL.
Offset 0x3FC - Layer_Boundary				
31:28	Unused[3:0]	R/W	0x0	
27:16	FX_Boundary[11: 0]	R/W	0x000	Layer X boundary in a field/frame. Any pixel crossing this boundary will be cropped.
15:12	Unused2[3:0]	R/W	0x0	
11:0	FY_Boundary[11: 0]	R/W	0x000	Layer Y boundary in a field/frame. Any pixel crossing this boundary will be cropped.
Offset 0xFE0 - Interrupt_Status_NHP_VO				

Table 357: NHP_VO registers *…continued*

Bit	Symbol	Access	Reset value	Description
31:24	Unused[7:0]	R/W	0x00	
23:21	Reserved[2:0]	R	0x0	
20	Unused2	R/W	0x0	
19:17	Reserved2[2:0]	R/W	0x0	
16	Unused3	R/W	0x0	
15:13	Reserved3[2:0]	R/W	0x0	
12	Unused4	R/W	0x0	
11:9	Reserved4[2:0]	R/W	0x0	
8	Ext_Hsync_Change	R	0x0	The line length derived from external HSYNC has changed.
7	Layer_done	R	0x0	The layer has been completely displayed
6	Buf_done	R	0x0	DMA channel is done fetching all data for the layer
5	Fcu_underflow	R	0x0	Underflow in FCU FIFO for the layer
4	Unused5	R/W	0x0	
3	Vintb	R	0x0	Vertical line interrupt issued if Y position matches VLINTB
2	Vinta	R	0x0	Vertical line interrupt issued if Y position matches VLINTA
1	Vbi_done_int	R	0x0	VBI/Register load is done with the current packet list
0	Vbi_packet_int	R	0x0	VBI/Register reload has sent a packet with the IRQ request bit set in the packet header
Offset 0xFE4 - Interrupt_Enable_NHP_VO				
31:24	Unused[7:0]	R/W	0x00	
23:9	Reserved[14:0]	R/W	0x0000	
8:0	Interrupt_enables[8:0]	R/W	0x000	A 1 in the appropriate bit will enable the interrupt according to the specification in register 0xFE0.
Offset 0xFE8 - Interrupt_Clear_NHP_VO				
31:24	Unused[7:0]	R/W	0x00	
23:9	Reserved[14:0]	W	0x0000	
8:0	Interrupt_clears[8:0]	W	0x000	A 1 in the appropriate bit will clear the interrupt according to the specification in register FE0.
Offset 0xFEC - Interrupt_Set_NHP_VO				
31:24	Unused[7:0]	R/W	0x00	
23:9	Reserved[14:0]	W	0x0000	
8:0	Interrupt_sets[8:0]	W	0x000	A 1 in the appropriate bit will set the interrupt according to the specification in register FE0.
Offset 0xFF4 - VO_Powerdown				
31	Powerdown_nhp_VO	R/W	0x0	This bit is only used for returning 0xDEADABBA on any MMIO DTL read except from this register.
30:0	Unused[30:0]	R/W	0x00000000	

Table 357: NHP_VO registers *…continued*

Bit	Symbol	Access	Reset value	Description
Offset 0xFFC - VO_Module_ID				
31:16	Module_id_nhp_VO[15:0]	R	0xA07B	Unique revision number
15:12	Rev_major[3:0]	R	0x0	Major revision counter
11:8	Rev_minor[3:0]	R	0x0	Minor revision counter
7:0	App_size[7:0]	R	0x00	Aperture Size 0 = 4 kB

6. LVDS transmitter

Low Voltage Differential Signaling (LVDS) is a low-power, low-noise differential technology for high speed data transmission over two PCB traces, or a balanced cable. LVDS allows single-channel data transmission at hundreds, or even up to a thousand Mbps. Low swing and current-mode driver outputs create low noise and provide very low power consumption across frequency ranges.

The LVDS transmitter IP provides a connection interface to FPDs.

6.1 Features

The following features are supported in the LVDS transmitter IP:

- Single-link transmission of RGB video pixel data.

  Up to 30-bits of RGB pixel data, synchronization signals (HS and VS), data valid indication signal (DE), and up to two user-defined control bits (UD1& UD2) sampled at the input using a 13.5-86 MHz input clock.

  Sampled input and clock sent over up to six differential channels using a 7x output clock.
- Transmission of two special control bits with 4 selectable values for each.

  Choice of control bits from the set {0, 1, UD1, UD2}.

  Programmable inversion of the above bits from the input to the output.

  Support of ALIS MODE for FHP by using UD1 input for field information.
- Choice of three: 30, 24, or 18-bits-per-pixel (i.e. 10, 8, or 6-bits-per-component) - selectable video data transmission formats.
- Selectable polarity for the data valid (DE) input signal.

  Programmable High or low level to designate (qualify) active video data
- Support for selectable output transmission formats:

  National™ semiconductor, THine™, and formats for 18-24-30-bits-per-pixel modes.
- Optional substitution of invalid pixel data by zero (RGB) values.
- Multiple selectable diagnostic modes.

  Stress test mode

  Pattern test mode
- Selectable clock strobe.

  Positive or negative edge of the clock for input data sampling

6.2 System perspective

In an LVDS-based system, one or more transmitters transmit serial data that is received by one or more receivers. Figure 245 shows an example system setup where a video composition processor (e.g., QVCP) in the host is connected to an LVDS transmitter that serializes the RGB pixel data and sends the data, along with the clock and some control signals, over multiple differential channels (streams) to an LVDS receiver in an LCD panel.

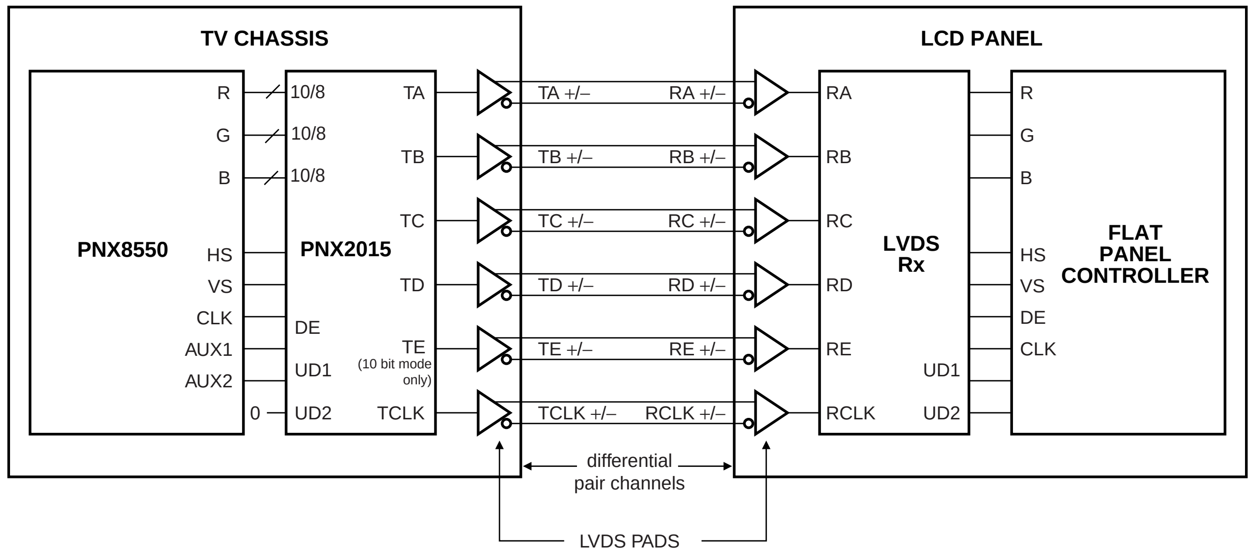


Fig 245.Overall system diagram

6.2.1 Data transmission

The output interface of the transmitter IP consists of a single link (i.e., one-input-one-output) comprising six simplex signals: five data and one clock. The data signals encode up to 30 bits of data into five LVDS data streams (channels). A phase-locked transmit clock is transmitted in parallel with the data streams over a sixth LVDS channel. Every cycle of the transmit clock (min:13.5 MHz to max:86 MHz), input data --- 30(= 3 components x 10 bits per component), 24(= 3 components x 8bits per component) or 18(= 3 components x 6 bits per component) bits of RGB pixel data, 3 bits of LCD timing data (HS, VS, and DE), and two extra user-defined input bits (UD1 and UD2) --- are sampled and transmitted at a rate of min:13.5x7=94.5 Mbps to max:86x7=602 Mbps. More specifically, the input data is mapped to a 5x7 (five data channels by seven bits per channel, for the 30-bit pixel data) or 4x7 (four data channels by seven bits per channel, for the 24-bit pixel data) or 3x7 (three data channels by seven bits per channel, for the 18-bit pixel data) output grid, where the seven bits in each channel are transmitted serially (the bits of the different channels being transmitted in parallel) using a 7x clock. The transmitter is programmable for both rising and falling-edge strobes (i.e., rising and falling edge of the input clock) through a programmable control register. One can also select between a high and a low level for the DE signal to signify valid active-video data.

6.2.2 Data formats

There are two predominant LVDS data formats: National™ semiconductor and THine™. The transmitter IP supports both of these formats for various pixel resolutions: 24b and 18b pixels for National™ and 30b, 24b, and 18b pixels for THine™. In addition, the transmitter also supports the minimum mandatory (DC unbalanced) set of the 24b and 18b multiplexing schemes for use in the flat panel monitor and related industries.

A special National™ Semiconductors transmission format (NS Mode) is also supported.

6.3 Functional description

6.3.1 LVDS transmitter block diagram

Figure 246 shows a conceptual block diagram of the LVDS transmitter IP (nhp_lvds_ltx).

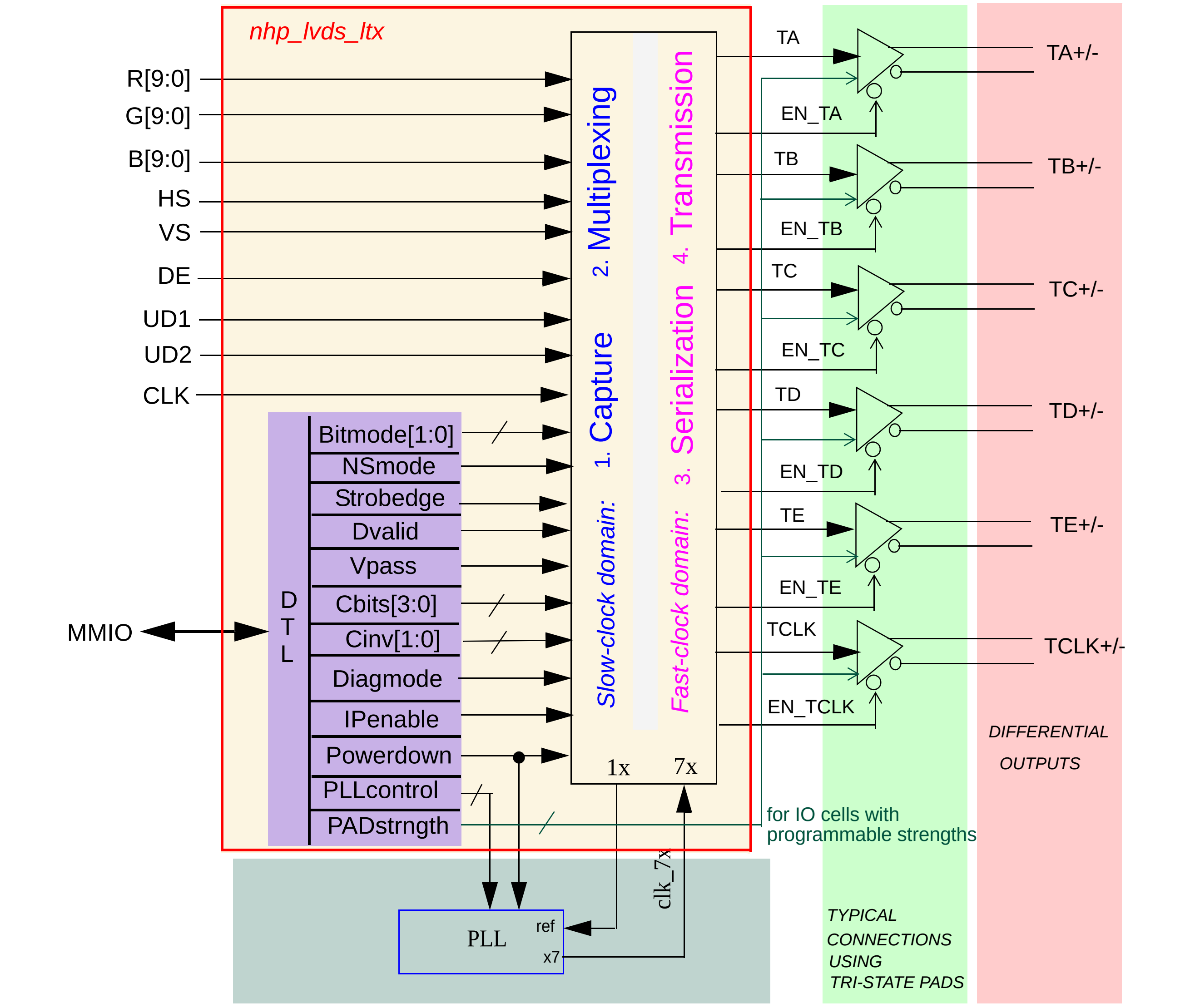


Fig 246.Functional block diagram

The Output clock (TCLK) is generated from the 7x input clock and has a fixed pattern “1100011” continually sent over in phase with the output data lines.

6.3.2 Architecture

The LVDS transmitter IP comprises three distinctive parts: a set of control registers with a DTL MMIO interface, a low-speed (i.e., low frequency) digital part --- the input module ---- to sample and format the input, and a high-speed (high frequency) digital part --- the output module --- that serializes the sampled data and transmits the serial data and clock on the output links. Using the low-frequency (1x) sampling clock from the IP as a reference, an external PLL generates and feeds back a high-frequency (7x) clock that is used to transmit the sampled data serially over multiple high-speed serial links.

The transmitter IP accepts at its input 30(=3x10) bits of RGB video pixel data (R[9:0], G[9:0], and B[9:0]), three bits of timing data in the form of horizontal/vertical synchronization and data-enable signals (HS, VS, and DE), two user-defined bits (UD1 and UD2), and an input clock whose frequency ranges from 13.5 MHz to 86 MHz. The video, timing, and user-defined data are sampled using the programmed edge (rising or falling) of the input clock. The sampling clock, derived from the input clock, has the same frequency but the phase difference is equal to either zero or 180o depending on whether the rising or the falling edge is to be used for sampling; in case of the latter, the derived clock is an inverted version of the sampling clock. For every cycle of the 1x sampling clock, the generated 7x clock is used to transmit the sampled data serially at a rate of 86x7=602 Mbps over 5 data links: TA, TB, TC, TD, and TE. A phase-locked transmit clock is transmitted in parallel with the data streams over a sixth link: TCLK.

The 6 serial output data streams from the logic-layer transmitter connect to external LVDS physical pads that, when enabled using the pad-enable signals EN_Tj ( ), convert the digital data streams to LVDS data streams (TA+/-, TB+/-, TC+/-, TD+/-, TE+/-, and TCLK+/-).

6.3.2.1 Control signals

The control registers of the LVDS transmitter IP are programmed via an MMIO DTL interface. The various control signals, explained below for clarification purposes, are later mapped to specific register bits.

The Bitmode[1:0] vector allows a selection between the 30-b, 24-b, or 18-b pixel data modes; the programmable mode selection allows disabling of the output pads corresponding to the unused data streams.

The NSmode signal, when asserted in the 24-b mode, allows the choice of a special National™ semiconductors transmission format.

The Strobedge signal controls the clock edge (rising or falling) used for sampling of the input data. Figure 247 shows simplified clock scheme:

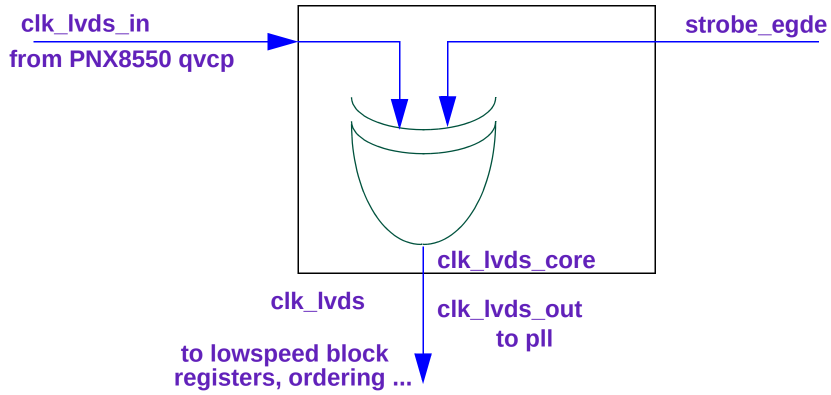


Fig 247.Clock scheme

The Dvalid signal qualifies the polarity of the DE signal in the sense that it specifies whether input video data is valid for DE=1 or for DE=0.

The Vpass signal identifies what data to consider and send for invalid inputs: for Vpass=0, the RGB input is passed to the output as is, even when the input is not valid; for Vpass=1, the input RGB data is sent only when input data is valid and zero values are substituted and sent when the input data is invalid. (Note that whether a data is valid or invalid is determined by a combination of DE and Dvalid).

The mapping of the full-resolution RGB video input data (at 10 bits per component), along with 3 bits of LCD timing, to the 5x7 output pattern (grid) allows two special control bits to be transmitted (for 8 bits per component mode, one special control bit can be sent. None in case of 6 bits per component mode). It is the Cbits[3:0] vector that determines whether each of the two special control bits transmitted is 0, 1, user-defined bit UD1 or user-defined bit UD2. The Cinv[1:0] vector allows selective inversion of the UD1 and the UD2 bits as they are mapped from the input to the output; this allows the flexibility, for example, of choosing the correct polarity of the field-identifier when one or more of the UD bits are used for this purpose.

The Diagmode signal, when asserted, programs the IP in a diagnostic mode where the IP loops continuously and transmits a pre-determined pattern. Note that no valid screen raster is sent in diagnostic mode, DE, VS, HS (1) and (2) are constant but programmable. The number of output data channel is function of the number of bits per pixel and therefore is programmable. This mode can be used for link testing and verification.

A combination of various control register settings, programmed via an MMIO DTL interface, determines whether and which of the LVDS pads are enabled (and, hence, which pads are disabled or tri-stated with the corresponding output floating to a high-impedance state). For example, an 18 bits per pixel (= 6 bits per component) mode disables (tri-states) both the TD and the TE pads (channels), a 24 bits per pixel (= 8 bits per component) mode disables (tri-states) only the TE pads (channels), whereas a global powerdown (Powerdown=1) or disabling of the LVDS IP (IPenable=0) disables (tri-states) all 6 of the LVDS pads (channels); as explained before, the Bitmode vector is used for determining the mode and, hence, selective disabling (tri-stating) of the pads.

The PLLcontrol register provides the necessary control signals to the PLL that receives the input clock (range: 13.5 to 86 MHz) and produces a 7x transmit clock (range: 7x13.5=94.5 to 7x86=602 MHz) which is used to stream out the serial data. For a PLL with a fixed clock ratio (7x), a PLLcontrol register is not necessary, but the control register is designed for future-proofing, e.g., to implement several different clock ratios. The PLL is disabled or programmed in a bypass mode during powerdown.

The PADstrngth register specifies the strength of the IO cell; when such pads are available, this allows greater flexibility in choosing the IO drive strength that is appropriate for the application.

Note: Most common LVDS pads are enabled by a LOW and disabled by a HIGH pad_enable signal (i.e., the pad enable should be 0 for normal operation and 1 when the pad is powered down and the pad output floats). The IP should assume the presence of such active-low LVDS pads. Consequently, there is no need of any special control signal to specify whether the pads are enabled by a LOW or a HIGH pad_enable signal.

6.3.2.2 Pixel format and bit mapping

This section describes the mapping of MSBs and LSBs from inputs to outputs for the 30, 24 and 18 bits per pixel modes. The programming of the special control bits is also explained.

Input Map Table 358 shows the definition of the input bits (that are used to illustrate the input-output mapping in the next section).

Output Map Figure 248 shows the mapping of the LVDS input bits to the output cycles (formats). There are several key observations:

- The TE channel (pad) is disabled for the 24-bits-per-pixel (8-bits-per-component) mode.
- Both TD and TE channels (pads) are disabled for the 18-bits-per-pixel (6-bits-per-component) mode.
- Positions marked as (1), (2) are special control-bit positions whose values are determined by UD1, UD2, Cbits, and Cinv.
- The suffix “n” in Xn refers to the “next” cycle (in order to illustrate the cycle relationship); so, X and Xn designate the values of X in the current and the next cycle, respectively.

Table 358: Input bit map

RGB component data bits			10 bits per component	8 bits per component	6 bits per component
R9	G9	B9	MSB	MSB	MSB
R8	G8	B8			
R7	G7	B7			
R6	G6	B6			
R5	G5	B5			
R4	G4	B4			LSB

Table 358: Input bit map *…continued*

RGB component data bits			10 bits per component	8 bits per component	6 bits per component
R3	G3	B3			
R2	G2	B2		LSB	
R1	G1	B1			
R0	G0	B0	LSB		

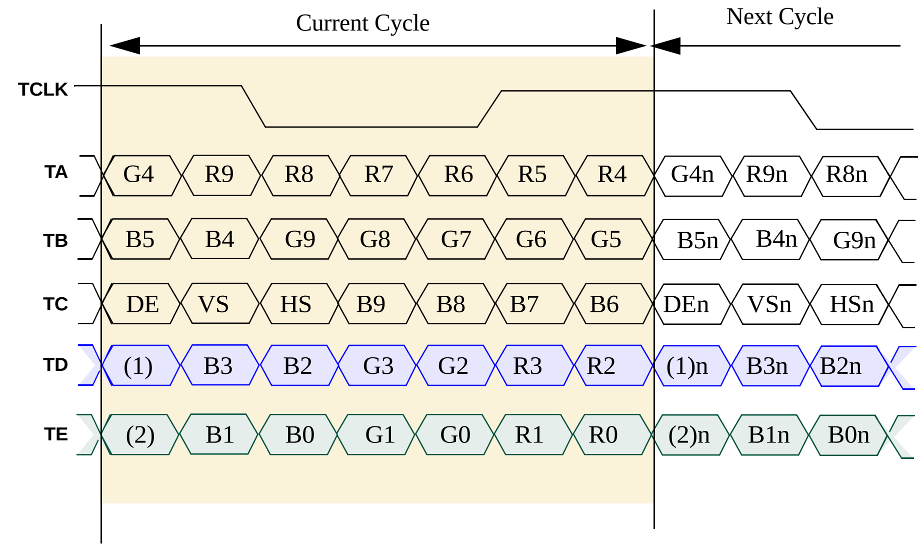

Current Cycle
Next Cycle
TCLK
TA
G4 R9 R8 R7 R6 R5 R4 G4n R9n R8n
TB
B5 B4 G9 G8 G7 G6 G5 B5n B4n G9n
TC
DE VS HS B9 B8 B7 B6 DEn VSn HSn
TD
(1) B3 B2 G3 G2 R3 R2 (1)n B3n B2n
TE
(2) B1 B0 G1 G0 R1 R0 (2)n B1n B0n

Fig 248.LVDS output bit map (NSmode=0)

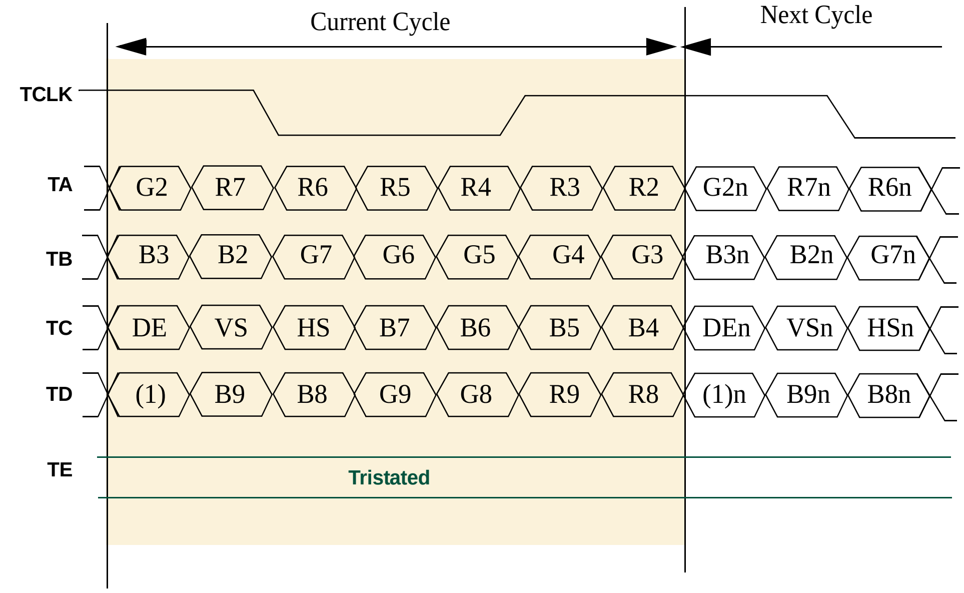


Fig 249.LVDS output bit map (NSmode=1 & Bitmode=24-b)

6.3.2.3 Specification of the special control bits

The special control bit position (1) should be programmable to be equal to 0, 1, UD1 or its inverse, and UD2 or its inverse. Similarly, control bit position (2) should be programmable to 0, 1, UD1 or its inverse, and UD2 or its inverse.

Table 359: Mapping of special control bit (1)

Cbits1[1:0]	Value at (1)	
	Cinv1=0	Cinv1=1
00	0	1
01	1	0
10	UD1	UD1
11	UD2	UD2

Table 360: Mapping of special control bit (2)

Cbits1[1:0]	Value at (2)	
	Cinv2=0	Cinv2=1
00	0	1
01	1	0
10	UD1	UD1
11	UD2	UD2

6.3.2.4 Input mapping for diagnostic mode

When in diagnostic mode, the inpuy RGB data and the control signals (DE,HS, VS..) are no longer used but substituted with internally-generated values according to programming of the DCONFIG register.

Figure 250 and Figure 251 illustrate the input bit mapping for two example diagnostic modes. It is worth noting that in the diagnostic mode only bits 9-4 of any color is used in case of the 6-bit mode and bits 9-2 in case of the 8-bit mode. Also R, G, and B values are the identical.

When PATTERN=1, a grayscale pattern is applied. A counter counts from 0 to Maxvalue (specified in the register that follows) and the count is treated as the R, G and B values; however, instead of directly connecting the counter output as R, G, and B values, the connection is bit-reversed so that the output lines corresponding to the upper bits are exercised. Figure 252 details on the input mapping for the color data values.

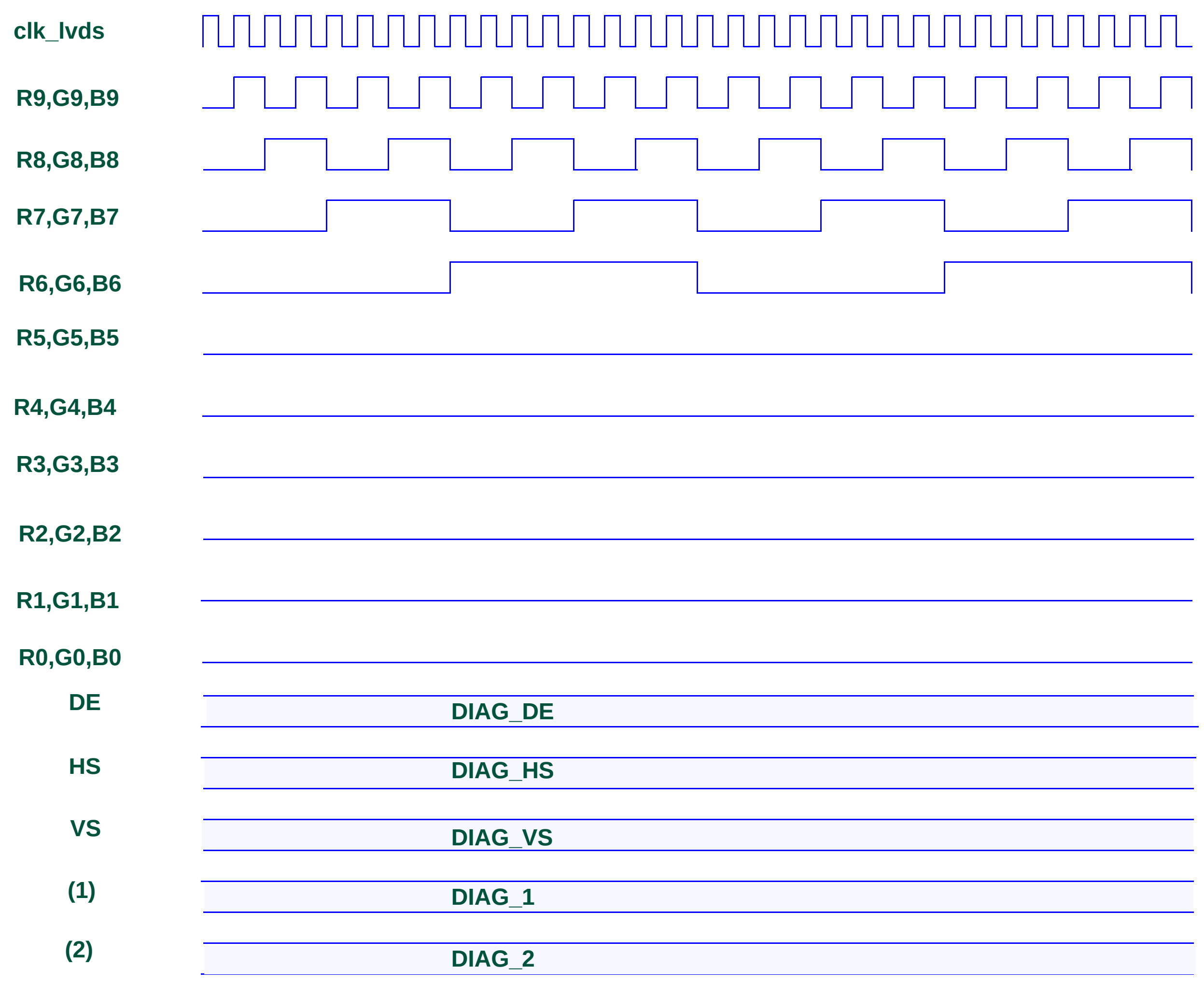


Fig 250.Example of input mapping for diagnostic mode: PATTERN=1, MAXVALUE=15

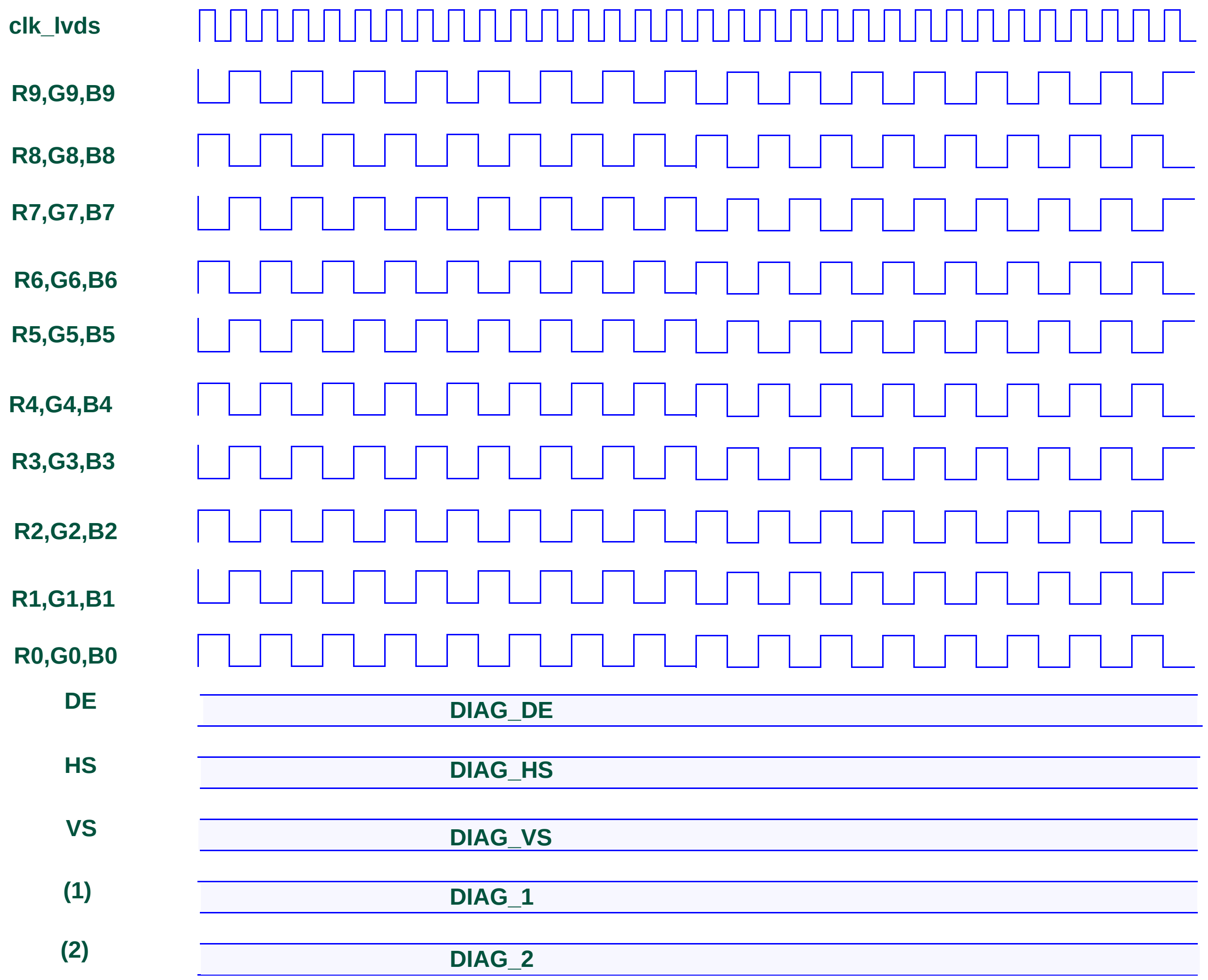


Fig 251.Example of input mapping for diagnostic mode: PATTERN=0, PATTERN1=h155, PATTERN2=h2AA

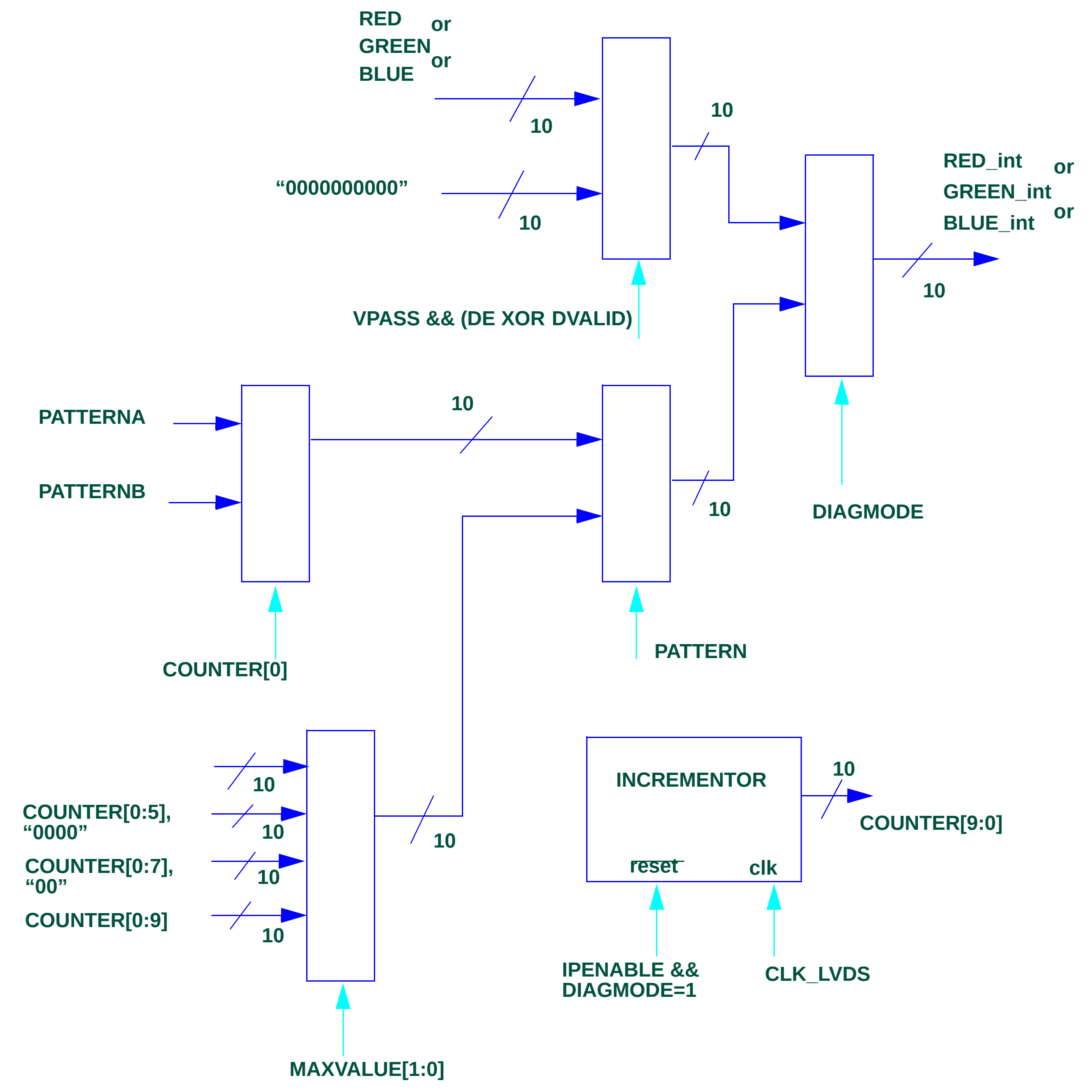


Fig 252.Input mapping for each color

6.4 General operations

The LVDS transmitter, once programmed and enabled via a MMIO DTL-based programming of its control registers, continues to run (transmit) for ever till it is re-programmed via a soft-reset. The control programming is not dynamic; once set up, the parameters are expected to remain unchanged till reset or powerdown.

The normal mode of operation is simple: RGB data and control signals at the input are sampled (and captured) using a low-frequency clock. Thereafter, the sampled data is serialized and sent out on multiple output links using a high-frequency clock. The only complication arises in keeping the relative skew between the output data and clock links within acceptable limits. The high-frequency clock is generated by an external PLL using the low-frequency clock as the reference.

A one-way asynchronous handshake (implemented via ping-pong buffers) between the low-speed and the high-speed logic relaxes the phase alignment constraint between the low- and high-frequency clocks, thereby making the design tolerant of the jitter and instantaneous phase misalignments; as long as the PLL-generated high-frequency clock is seven times the frequency of the low-frequency clock, the design functions correctly.

6.4.1 Data flow and control

This section describes the control programming and data flow.

6.4.1.1 Normal mode

1. The transmitter is powered on.
2. A reset is applied and subsequently the IP is taken out of reset.
3. All control registers show up with their default values.
4. The desired control registers are programmed with intended parameters (in the functional mode).
5. The output pads remain disabled till the IP is enabled --- the enable control register bit is programmed last.
6. Once enabled, the IP starts to capture valid data at its inputs and transmit serial data bits over the output links.
7. Till valid data appears at its inputs (the first time) and makes it to the output, the IP sends zero values (i.e., R=G=B=0) on its output links.
8. Once valid data starts flowing through, any invalid data that now comes along is either passed through or replaced by zero values depending on the programming of a particular control register bit.

6.4.1.2 Diagnostic mode

1. The transmitter is powered on.
2. A reset is applied and subsequently the IP is taken out of reset.
3. All control registers show up with their default values.
4. The desired control registers are programmed with intended parameters (in the diagnostic mode).
5. The output pads remain disabled till the IP is enabled --- the enable control register bit is programmed last.
6. Once enabled, the IP starts to transmit a fixed pattern without any capture of data at its inputs. There is a slack time for the actual test pattern to appear at the output; the slack time is filled by zero RGB values.

In order to move (transition) from a Functional to a Diagnostic mode and vice versa apply a software reset. However, once a software reset is applied, the transmitter is disabled (and the output pads tristated). It now needs to be enabled explicitly (hopefully, after reprogramming its registers with desired parameters).

For all changes of register settings such as BitMode, Vpass, etc., a soft reset is advisable to prevent undefined behavior (and transmission of erroneous data), even though the hardware does implement some precautionary measures in certain cases.

In order to turn the transmitter off, it is also advisable to apply a soft reset that automatically disables the IP. Only setting and/or resetting the IP enable register bit, without resetting (software reset) the IP, is not advisable; it leaves the transmitter in an undefined state and leads to incorrect transmission.

6.4.2 Standard features

6.4.2.1 Clock programming

The transmitter IP receives an input clock along with the RGB video data. This input clock is synchronous with the RGB data and is, in all probabilities, an output from the video out block driving the transmitter. The supported input clock frequency ranges from 13.5 to 86 MHz. It is ultimately the responsibility of the integrator to choose a PLL with the desired range of frequencies.

Depending on which edge of the input clock is to be used to sample the incoming data, the input clock needs to be inverted (in case negative-edge strobing is programmed). Thus, the sampling clock is derived from either the input clock directly or a complemented version of it. In order to keep the clock delay (between the sampling and the input clocks) the same for sampling based on both the rising and the falling edges, the sampling clock is actually derived by xor’ing the input clock with the strobedge control bit.

The sampling clock is not only used to sample and capture the RGB data and synchronization signals at the input of the transmitter, but is also sent (out of the IP) as the reference 1x clock to an external PLL. The PLL generates a faster (seven times the sampling clock) 7x clock that is fed back to the transmitter IP where it is used for serialization and output data transmission.

6.4.2.2 Reset-related Issues

For an LCD display, the correct LVDS data (that is zeroed out if not qualified by the data enable signal) and valid synchronization signals should be available within 50 milliseconds of power-on. SW, however, takes about 2~5 seconds to set up the system after power-on. Therefore, in order to have the active video ready within 50 milliseconds of power-on, the screen-timing generator of the video source (e.g., video out) has to be set up as part of the boot script or the LCD panel has to be powered on later. In order to alleviate the situation, the default IP behavior at startup and a hard or soft reset is designed to transmit 0 values for R, G, and B. Note that since the required output frequency and the type of the LCD panel are not known a priority (before the control programming is complete), the implemented (simple) scheme of sending hard-coded values for HS, VS, and DE may (is likely) produce a wrong raster.

6.4.2.3 Interrupt processing

The LVDS transmitter IP does not generate any interrupts.

6.4.2.4 Power management

In the powerdown mode (i.e., when the powerdown register bit is set), the pads get tristated. So, setting powerdown register in the middle of operation abruptly turns the transmission off (and this behavior is the same as suddenly applying a software reset). The difference, however, is that a software reset sets all control registers with their default values, whereas setting the powerdown bit retains the settings of the registers and allows read responses to DTL MMIO read transactions for the powerdown register (see Section 6.6).

To power down the transmitter, it is recommended that one (SW) first applies a soft reset or disables the IP and then re-programs the IP in the powerdown mode (mainly to save power, if possible, by sending the powerdown signal to the external PLL); this resetting allows clean power up. Unfortunately, this implies re-programming the transmitter after power up. (Powering down without a soft reset is allowed, but then a soft reset is applied after powerup and this is followed by re-programming of the control registers and re-enabling of the IP. Note however, that since the IP retains its register values at powerdown, the latter scenario of powering down without a reset allows the possibility of the IP transmitting some non-zero left-over garbage values for the brief period between powerup and soft reset.)

The powerdown register bit is also an output from the transmitter IP module so that it is possible to program the external PLL in a powerdown mode (e.g.,1:1 bypass mode or PLL output=0 mode) provided such a powerdown mode exists.

6.4.2.5 Register programming guidelines

The special NS format, mentioned earlier, is valid only in the 24-b-per-pixel mode. So, the pixel mode has to be separately and correctly programmed when using the special NS format.

The following table briefly describes some of the programming recommendations for and transitional activities during reset and powerdown.

Table 361: Reset and powerdown guide

Action	Achieved by	Effect(s)
Turn transmitter off	Apply soft reset	Pads get tristated IP gets disabled Control registers get set to default values IP needs to be re-programmed and re-enabled to start transmission.
Power transmitter down	Set powerdown register bit	Pads get tristated IP stops responding to MMIO writes (Only the powerdown register continues to respond to MMIO reads) Clocks need to be set, Powerdown bit needs to be reset and the IP needs to be reset, re-programmed, and re-enabled to start re-transmission. (As mentioned before, applying a soft reset followed by powerdown provides a cleaner solution).

6.5 Application note

Table 362 shows the values of the special control bit positions for a number of commercial chips and standards. Any desired value (0,1,UD1 and UD2) can be programmed into (1) and (2) and their correct polarity can be achieved using Cinv[1:0]. For safety reasons, the Reserved bits are programmed as 0.

Table 362: Values of special bits for available chips

Bit (Position)	Value	Vendor/Spec/Standard ....
(1)	Reserved	OpenLDI 24-bit single pixel unbalanced mode
	Reserved	OpenLDI 24-bit single pixel unbalanced mode
	0	THine™ THC63LVDM83A 24-bit single pixel unbalanced
	0	THine™ THC63LVD823 24-bit single pixel unbalanced mode
	0	THine™ THC63LVD104 30-bit single pixel unbalanced mode
	Reserved	DS90C385 24-bit single pixel unbalanced mode
	Field id	Fujitsu-Hitachi Plasma ALISformat to insert correct field id (polarity controlled using Cinv[0])
(2)	0	THine™ THC63LVD104 30-bit single pixel unbalanced mode

Recommended Cable Connections are specified in chapter 6 of an LVDS Owners Manual See Ref. 9

6.6 Register descriptions

6.6.1 Register summary

Table 363: Register summary

Offset	Name	Description
0x0	OCONTROL	Programmable LVDS operating mode control register.
0x4	FCONFIG	Functional mode configuration.
0x8	DCONFIG	Diagnostic mode configuration.
0xC	ACONTROL	Programmable analog (pad and pll) control.
0xFF4	LVDS_POWERDOWN	Powerdown and softreset.
0xFFC	Module_ID	Module Identification and revision information.

6.6.2 Register tables

Table 364: NHP_LVDS_LTX registers

Bit	Symbol	Access	Reset value	Description
Offset 0x0 - OCONTROL				
31	Softreset	W	0x0	Writing a one into this bit will reset the block. Read 0.
30:5	Reserved[25:0]	R/W	0x0000000	
4	Nsmode	R/W	0x0	Output bit Map 1 : Special NS scheme (BITMODE must be 01) 0 : National, Thine, and OpenLDI modes

Table 364: NHP_LVDS_LTX registers *...continued*

Bit	Symbol	Access	Reset value	Description
3:2	Bitmode[1:0]	R/W	0x1	00 : 6bits per component - 18 bits per pixel. 01 : 8bits per component - 24 bits per pixel. 10 : 10bits per component - 30 bits per pixel. 11 : reserved
1	Diagmode	R/W	0x0	0 : Functional mode. 1 : Diagnostic mode.
0	Ipenable	R/W	0x0	Enabling of the IP 1: IP enabled (pads get enabled depending on BITMODE) 0: IP disabled (pads get disabled)
Offset 0x4 - FCONFIG				
31:10	Reserved[21:0]	R/W	0x000000	
9	Cinv2	R/W	0x0	Inversion of control bit (2) from input to output: 1: Control bit (2) is inverted 0: Control bit (2) is not inverted but passed as is.
8	Cinv1	R/W	0x0	Inversion of control bit (1) from input to output: 1: Control bit (1) is inverted 0: Control bit (1) is not inverted but passed as is.
7:6	Cbit2[1:0]	R/W	0x0	Mapping of special control bit (2): 00 : 0 01 : 1 10 : UD1 11 : UD2
5:4	Cbit1[1:0]	R/W	0x0	Mapping of special control bit (1): 00 : 0 01 : 1 10 : UD1 11 : UD2
3:2	Reserved2[1:0]	R/W	0x0	
1	Vpass	R/W	0x0	When set to 1, RGB substituted by 0 when DE is not valid, otherwise passed as is.
0	Dvalid	R/W	0x0	Polarity of DE signal. 1 : DE=1 signifies active video and DE=0 implies inactive video. 0 : DE=0 signifies active video and DE=1 implies inactive video.
Offset 0x8 - DCONFIG				
31:30	Reserved[1:0]	R/W	0x0	
29:20	Pattern1[9:0]	R/W	0x000	Pattern1 for Worst case pattern mode. In case of 8 or 6 bit mode, or line mode, only upper bits are used.
19:10	Pattern2[9:0]	R/W	0x000	Pattern2 for Worst case pattern mode. In case of 8 or 6 bit mode, or line mode, only upper bits are used.
9:8	Reserved2[1:0]	R/W	0x0	
7	Diag_de	R/W	0x1	Binary values to transmit for DE
6	Diag_hs	R/W	0x1	Binary values to transmit for HS
5	Diag_vs	R/W	0x1	Binary values to transmit for VS
4	Diag_1	R/W	0x0	Binary values to transmit for (1)
3	Diag_2	R/W	0x0	Binary values to transmit for (2)
2:1	Maxvalue[1:0]	R/W	0x0	Maximum count value to transmit WHEN pattern =0: 00 : 15. 01 : 63. 10 : 255. 11 : 1023.

Table 364: NHP_LVDS_LTX registers *...continued*

Bit	Symbol	Access	Reset value	Description
0	Pattern	R/W	0x0	0 : Transmit a grayscale pattern; a counter counts from 0 to MAXVALUE and the count is treated as R, G, and B values and connected that way but bit-reversed so that the output lines corresponding to the upper bits are always exercised. 1 : Transmit worst case pattern, also called doublechecker pattern, that alternates between PATTERN1 and PATTRN2.
Offset 0xC - ACONTROL				
31:9	Reserved[22:0]	R/W	0x000000	
8	Strobedge	R/W	0x0	Capture edge of clock 0 : posedge 1 : negedge.
7:4	Pllcntl[3:0]	R/W	0x0	Control for PLL (register value is passed to output as is).
3:0	Padstrength[3:0]	R/W	0x0	Control for PADS (register value is passed to output as is).
Offset 0xFF4 - LVDS_POWERDOWN				
31	Powerdown_lvds_ltx	R/W	0x0	
30:0	Reserved[30:0]	R/W	0x00000000	
Offset 0xFFC - Module_ID				
31:16	Mod_id[15:0]	R	0xA07C	Module ID; unique 16-bit code
15:12	Rev_major[3:0]	R	0x0	Major revision counter
11:8	Rev_minor[3:0]	R	0x0	Minor revision counter
7:0	App_size[7:0]	R	0x00	Aperture Size: 0 = 4K

7. TV microcontroller

7.1 Overview

The TV Microcontroller is a derivative of the 80C51 Microcontroller family. It consists of an 80C51 CPU core, together with embedded memory and peripherals required for TV systems. For details of 8051 instruction set see Ref. 6.

The TV Microcontroller has the capability to support the following functions:

- User interfacing via keyboard or remote control.
- System power management; control and detection of power supplies.
- Communication with external sources via P50.
- Control of system peripherals.

Remark: P50 (Philips Project 50) standard for signalling on SCART (AVlink, Easylink, SmartLink). This provides a simple signalling system between devices and has been implemented on many analogue VCRs.

The TV Microcontroller subsystem is isolated from the other sub systems within the PNX2015 (See Section 2.5.5). It has its own power supply (1.2V & 3.3V), together with separate clocking (16MHz) and reset. This allows for it to be active while all other sub systems are either inactive, via clock being disabled or powered down. A functional block diagram is shown in Figure 253.

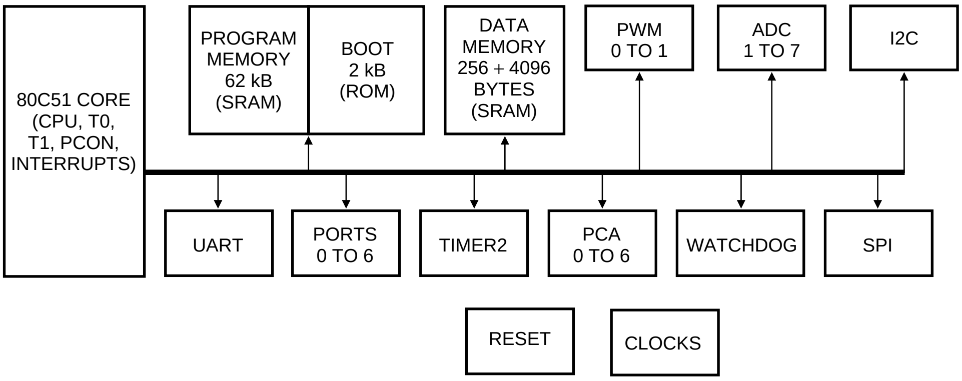


Fig 253.TV microcontroller

The CPU core is a derivative of the 80C51 that operates at frequencies of 16/24 MHz. It executes a maximum of 1 instructions per 6 clock cycles (referred to as CPU state 1 through 6). It has 15 interrupt sources (2 external and 13 internal), with 2 priority levels. Three timers are available (T0/T1 and T2).

The program code memory is 64KB of which 0 to 62KB is implemented as SRAM (PRAM) and 62KB to 64KB is implemented as ROM (PROM) for booting of the TV Microcontroller. This external Flash memory access during the boot process requires a specific Emulator when developing program code. The boot process requires a specific configuration of the internal data memory is the standard 256 Bytes for Direct/Indirect addressing (IDATA), and 4096 Bytes of Auxiliary (XDATA). No external data memory interface is provided.

The TV Microcontroller provides two pulse width modulator outputs (PWM 0-1), eight Analog to Digital Converters (ADC0-7), an I^{2}C, a UART, a Programmable Counter Array (PCA) with seven capture/compare modules, timers (Timer0/1 and Timer2), a WatchDog timer and a Serial Peripheral Interface (SPI). It also provides for seven 8-bit ports. Some of these ports have special purposes, and others are general purpose I/Os.

Table 365 describes the TV Microcontroller pads. Figure 254 shows a typical connection between these pads and the outside circuitry.

Table 365: TV Microcontroller pads

Symbol	Type	Description
P0_0, …, P7	I/O	Port P0
P1_0, …, P1_7	I/O	Port P1
P2_0, …, P2_7	I/O	Port P2
P3_0, …, P3_7	I/O	Port P3
P4_0, …, P4_7	I/O	Port P4
ADC3V3VDDA		ADC 3V3 reference
P5_0,…, P5_7	I	ADC pins
ADC_VSSA		ADC GND
SPI_SDO/P6.0	O	Motorola compliant SPI (64k Flash)
SPI_SDI/P6.1	I	Motorola compliant SPI (64k Flash)
SPI_CLK/P6.2	O	Motorola compliant SPI (64k Flash)
SPI_CSB/P6.3	O	Motorola compliant SPI (64k Flash)
MODE0/P6_4	I/O	Boot mode selection
MODE1/P6_5	I/O	Write Enable to Flash Memory
SCL_MC/P6.6	I/O	I^{2}C Clock
SDA_MC/P6.7	I/O	I^{2}C Data
PWM0	O	Dedicated PWM output
PWM1	O	Dedicated PWM output
PSEN	I/O	Emulation (Program Store Enable)
ALE	I/O	Emulation (Address Latch Enable)
EA	I/O	Emulation (External Access)
MC_RESET	I	Reset (active HIGH)
$V_{DDD(MCIO)}$	PWR	MCIO TV Microcontroller supply (3.3 V)
$V_{DDD(MC\_CORE)}$	PWR	MC_CORE TV Microcontroller supply (1.2 V)

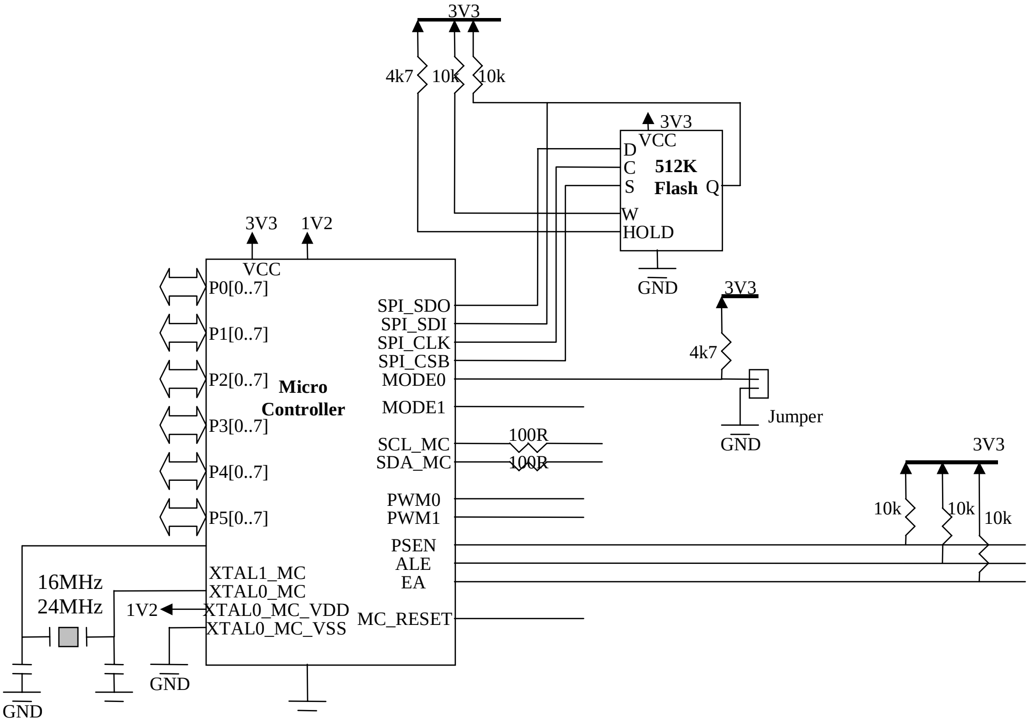


Fig 254. TV Microcontroller pads and typical connection to an outside circuitry

7.2 Reset and Clocks

7.2.1 Reset

The TV Microcontroller reset (MC_RESET) is active high and is held high for at least 30 clock cycles after the crystal frequency has become stable. Operating mode selection occurs at reset (for more details about mode selection, see Section 7.16 Boot Process and Mode Selection).

7.2.2 Clock

The TV Microcontroller subsystem operates on a single clock, which is supplied by an external crystal or clock source. The supported frequencies are 16MHz and 24MHz. Unlike the standard 8051 (Ref. 6) the TV Microcontroller machine cycle consists of 6 clock cycles, instead of 12.

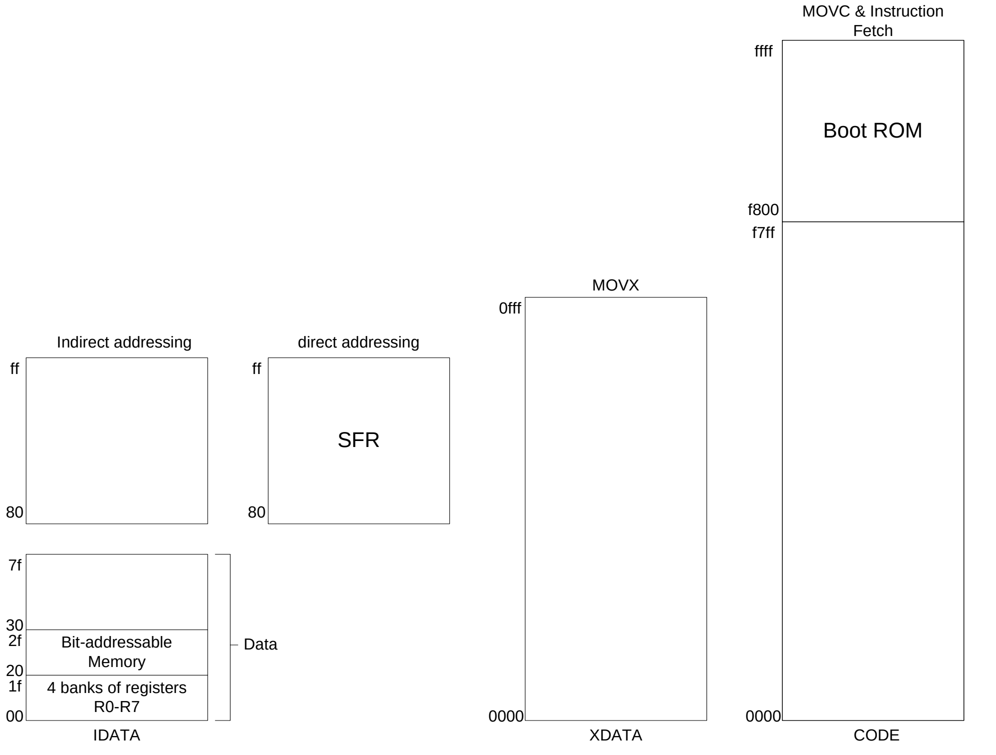
7.3 Memory

The TV Microcontroller memory spaces are as follows:

Table 366: TV Microcontroller memory

Memory	Description
DATA	128 bytes of internal data memory space (00h..7Fh) accessed via direct or indirect addressing, using instructions other than MOVX and MOVC. All or part of the Stack may be in this area.
IDATA	Indirect Data. 256 bytes of internal data memory space (00h:FFh) accessed via indirect addressing using instructions other than MOVX and MOVC. All or part of the Stack may be in this area. This area includes the DATA area and the 128 bytes immediately above it.
SFR	Special Function Registers. Selected CPU registers and peripheral control and status registers, accessible only via direct addressing.
XDATA	“External” Data or Auxiliary RAM. Duplicates the classic 80C51 64KB memory space addressed via the MOVX instruction using the DPTR, R0, or R1. The TV Microcontroller has 4096 bytes of on-chip XDATA memory. Off-chip XDATA memory is not supported.
CODE	64 KB of Code memory space, accessed as part of program execution and via the MOVC instruction. The TV Microcontroller has 64 KB of on-chip CODE memory. Of this, 62 KB is implemented as RAM, which is preloaded from an external flash memory via the SPI interface immediately after reset. The remaining 2048 bytes of CODE memory are implemented as a ROM, which contains routines for booting from the external flash memory and reprogramming the flash memory. Off-chip CODE memory is not supported.

Remark: The above memory organization is not applicable during system boot, while the CODE memory is being preloaded from the external flash memory.



The above memory organization is not applicable during system boot, while the CODE memory is being preloaded from the external flash memory.

Fig 255.Memory map

7.4 SFR (Special Function Registers)

The SFR map is shown in Table 367.

Table 367: SFR map

Low order nibble	High order nibble							
	8	9	A	B	C	D	E	F
0	P0	P1	P2	P3	PCACCON	PSW	ACC	B
	Section 7.6	Section 7.6	Section 7.6	Section 7.6	Section 7.10	Section 7.4	Section 7.4	Section 7.4
1	SP	-	-	-	PCACMOD	-	SPICON	SPIDAT
	Section 7.4				Section 7.10		Section 7.14	Section 7.14
2	DPL	PWML	PWMCCON	PWMM0CCL	PWMM0CCH	PWMM0MOD	SPISTAT	SPICLK
	Section 7.4	Section 7.11	Section 7.11	Section 7.11	Section 7.11	Section 7.11	Section 7.14	Section 7.14
3	DPH	PWMH	PWMCMOD	PWMM1CCL	PWMM1CCH	PWMM1MOD	-	ADCSEL
	Section 7.4	Section 7.11	Section 7.11	Section 7.11	Section 7.11	Section 7.11		Section 7.15
4	P0CFGA	P1CFGA	P2CFGA	P3CFGA	P4CFGA	-	P6CFGA	ADCTIML
	Section 7.6	Section 7.6	Section 7.6	Section 7.6	Section 7.6		Section 7.6	Section 7.15
5	POCFGB	P1CFGB	P2CFGB	P3CFGB	P4CFGB	ADCON	P6CFGB	ADCTIMH
	Section 7.6	Section 7.6	Section 7.6	Section 7.6	Section 7.6	Section 7.15	Section 7.6	Section 7.15
6	ADCRL0	ADCRL1	ADCRL2	ADCRL3	ADCRL4	ADCRL5	ADCRL6	ADCRL7
	Section 7.15	Section 7.15	Section 7.15	Section 7.15	Section 7.15	Section 7.15	Section 7.15	Section 7.15
7	PCON	-	-	-	P4	P5	P6	ADCRH
	Section 7.4				Section 7.6	Section 7.6	Section 7.6	Section 7.15
8	T01CON	S0CON	IE0	IP0	T2CON	I2C0CON	IE1	IP1
	Section 7.7	Section 7.13	Section 7.5	Section 7.5	Section 7.8	Section 7.12	Section 7.5	Section 7.5
9	T01MOD	S0BUF	PCAM0CCL	PCAM0CCH	T2MOD	I2C0STA	PCAM0MOD	-
	Section 7.7	Section 7.13	Section 7.10	Section 7.10	Section 7.8	Section 7.12	Section 7.10	
A	T0L	S0FECON	PCAM1CCL	PCAM1CCH	T2RCL	I2C0DAT	PCAM1MOD	XRAMP
	Section 7.7	Section 7.13	Section 7.10	Section 7.10	Section 7.8	Section 7.12	Section 7.10	Section 7.4
B	T1L	-	PCAM2CCL	PCAM2CCH	T2RCH	I2C0ADR	PCAM2MOD	-
	Section 7.7		Section 7.10	Section 7.10	Section 7.8	Section 7.12	Section 7.10	
C	T0H	-	PCAM3CCL	PCAM3CCH	T2L	-	PCAM3MOD	-
	Section 7.7		Section 7.10	Section 7.10	Section 7.8		Section 7.10	
D	T1H	-	PCAM4CCL	PCAM4CCH	T2H	-	PCAM4MOD	BOOT
	Section 7.7		Section 7.10	Section 7.10	Section 7.8		Section 7.10	Section 7.4.1.7
E	PCAL	-	PCAM5CCL	PCAM5CCH	-	-	PCAM5MOD	WDTKEY
	Section 7.10		Section 7.10	Section 7.10			Section 7.10	Section 7.9
F	PCAH	-	PCAM6CCL	PCAM6CCH	-	-	PCAM6MOD	WDT
	Section 7.10		Section 7.10	Section 7.10			Section 7.10	Section 7.9

7.4.1 Miscellaneous SFRs

7.4.1.1 PSW: Program Status Word

Table 368: PSW

Bit	Symbol	Description
7	C	carry bit
6	AC	auxiliary carry bit
5	F0	general purpose flag 0
4	RS1	register bank selection bit 1
3	RS0	register bank selection bit 0
2	OV	overflow flag
1	F1	general purpose flag 1
0	P	parity bit

7.4.1.2 ACC: Accumulator

Destination register for most cpu instructions.

7.4.1.3 B: B register

Operand register for MUL and DIV instructions.

7.4.1.4 SP: Stack Pointer

Contains address of the top of the stack (in IDATA memory).

7.4.1.5 DPH/DPL: Data Pointer High/Low

Concatenated to form 16-bit XDATA address for MOVX instructions using DP.

7.4.1.6 XRAMP: XDATA Page

Forms the upper 8 bits of the 16-bit XDATA address for MOVX instructions using registers R0/R1.

7.4.1.7 BOOT

Table 369: BOOT

Bit	Symbol	Description
7	BOOT	Selects TV Microcontroller memory map. When this bit is zero, the normal memory map (as Section 7.3) applies. When this bit is one, an alternative memory mapping is applied, to allow the CODE memory to be preloaded from an external Flash memory (See Section 7.17). This register should not be used by application code.
6	-	Reserved

7.4.1.8 PCON

The PCON module contains one Special Function Register (SFR) that controls various parts of the TV Microcontroller. All bits can be read and written by the CPU. In one machine cycle, the software load overrides the hardware load. Bit 4 of the PCON register has one shadow bit implemented.

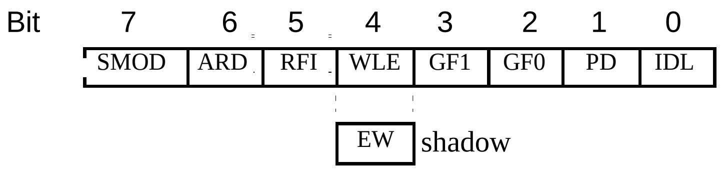


Fig 256.PCON register

IDL (IDLe mode)

When this bit becomes set, the TV Microcontroller goes into in IDLE mode.

In IDLE mode, power is conserved by disabling the clock to the cpu, ADC, and internal memory. The following peripherals remain enabled in idle mode: timer2, pca, pwm, uart, i2c, spi, wdt and timer01.

A reset or any interrupt terminates IDLE mode, returns to normal mode and clears the PCON.IDL bit.

PD (Power Down mode)

When this bit becomes set, the TV Microcontroller goes into Power Down mode, when the clocks for the cpu and all peripherals are disabled.

A reset or an external interrupt terminates Power Down mode and restarts the TV Microcontroller. Note that the external interrupt is enabled and set to level sensitive. This bit is automatically reset when waking up from Power Down mode.

Setting this bit is inhibited when the watchdog timer is enabled. The WDTKEY register is used to disable the watchdog timer before entering power-down mode.

GF0 and GF1 (general purpose flags 0 and 1)

These two bits are free for the software to use and have no influence on the hardware part of the TV Microcontroller. Setting and clearing these bits can only be done by software.

WLE (Watchdog Load Enable)

The WLE bit can be changed by software and by hardware. Setting this bit will allow the CPU to load the watchdog timer. The watchdog timer will clear this bit after a load.

EW (Enable Watchdog)

This bit is a shadow bit of the PCON WLE bit. The shadow bit is set when software writes to the WLE bit for the first time (i.e. when the shadow bit EW = 0). Setting EW will enable the watchdog timer. Only a synchronous reset can clear this bit.

RFI (Radio Frequency Interference)

The RFI bit can be changed by software only. Setting this bit will disable the ALE toggling during on-chip program or data memory access.

ARD (Auxiliary RAM Disable)

Reserved, should be written as 0.

SMOD (Serial MODe)

Reserved

7.5 Interrupts

7.5.1 Interrupt vectors

The TV Microcontroller has fifteen interrupt sources which can be individually enabled and assigned to one of two priority levels. A low priority interrupt can be interrupted by a high priority interrupt, but not by another low priority interrupt. A high priority interrupt cannot be interrupted by any other interrupt source. If two requests of different priority levels are received simultaneously, the request with the higher priority level is serviced. If requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced.

Each interrupt source corresponds to an interrupt vector address in the CPU program memory address space. The processor acknowledges an interrupt request by executing a hardware generated LCALL to the interrupt service routine located at the corresponding interrupt vector address.

A RETI (Return from Interrupt) instruction causes the program to continue at the address of the instruction immediately after the point at which the interrupt was detected. If a lower, or same level interrupt has been pending when the RETI instruction is executed, then one further instruction of the interrupted program is executed before the pending interrupt is processed.

The available interrupt sources are shown in the following table, along with the associated interrupt vector and polling order.

Table 370: Interrupt Sources

Interrupt source	Interrupt number	Vector	Polling Priority Order
EX0	0	0003H	0 (highest)
I2C	5	002BH	1
ADC	10	0053H	2
T0	1	000BH	3
PCA_CC0	6	0033H	4
PCA_CC4	11	005BH	5
EX1	2	0013H	6
PCA_CC1	7	003BH	7
PCA_CC5	12	0063H	8
T1	3	001BH	9
PCA_CC2	8	0043H	10
PCA_CC6	13	006BH	11
UART	4	0023H	12
PCA_CC3	9	004BH	13
T2	14	0073H	14 (lowest)

7.5.2 IE registers (Interrupt Enable registers)

Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the interrupt enable special function registers IE0 and IE1. The IE0 register also contains a global interrupt enable bit which can be used to disable all interrupts. The interrupt enable registers are shown in the following table:

Table 371: Interrupt Enable Registers

Register Bit	Interrupt Number	Register Bit	Interrupt Number
IE0[0]	0	IE1[0]	7
IE0[1]	1	IE1[1]	8
IE0[2]	2	IE1[2]	9
IE0[3]	3	IE1[3]	10
IE0[4]	4	IE1[4]	11
IE0[5]	5	IE1[5]	12
IE0[6]	6	IE1[6]	13
IE0[7]	global interrupt enable	IE1[7]	14

7.5.3 IP register (Interrupt Priority registers)

Each interrupt source can be assigned to the high (or low) priority level by setting (or clearing) the corresponding bit in the interrupt priority special function registers IP0 and IP1. The interrupt priority registers are shown in the following table:

Table 372: Interrupt Priority Register

Register Bit	Interrupt Number	Register Bit	Interrupt Number
IP0[0]	0	IP1[0]	7
IP0[1]	1	IP1[1]	8
IP0[2]	2	IP1[2]	9
IP0[3]	3	IP1[3]	10
IP0[4]	4	IP1[4]	11
IP0[5]	5	IP1[5]	12
IP0[6]	6	IP1[6]	13
IP0[7]	not used	IP1[7]	14

7.6 I/O, Ports 0:6

The TV microcontroller has 56 general purpose I/O lines, organized as seven 8-bit ports: P0 - P6. The lines are multiplexed with the I/O signals for the integrated peripheral interfaces, as shown in Table 373.

Table 373: Alternate port functions

** denotes function only required for emulation mode; no alternative in normal mode.*

	Port0	Port1	Port2	Port3	Port4	Port5	Port6
Px.0	AD0*	CC0	A8*	RXD	-	ADC0	SPI_SDO
Px.1	AD1*	CC1	A9*	TXD	-	ADC1	SPI_SDI
Px.2	AD2*	CC2	A10*	INT0_N	-	ADC2	SPI_CLK
Px.3	AD3*	CC3	A11*	INT1_N	-	ADC3	SPI_CSB
Px.4	AD4*	CC4	A12*	T0	-	ADC4	Boot mode

Table 373: Alternate port functions *…continued*

** denotes function only required for emulation mode; no alternative in normal mode.*

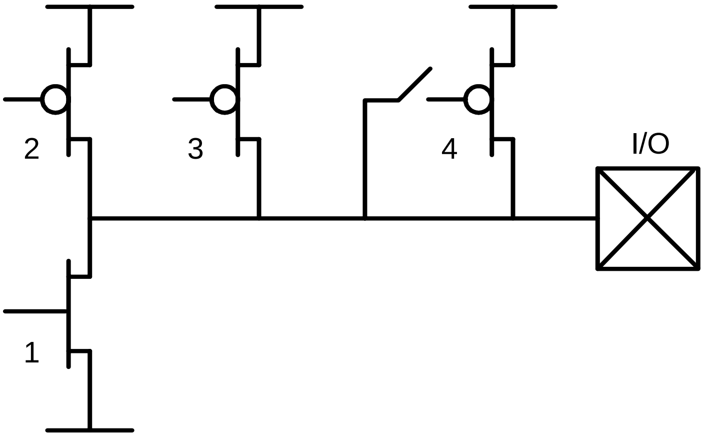
	Port0	Port1	Port2	Port3	Port4	Port5	Port6
Px.5	AD5*	CC5	A13*	T1	-	ADC5	-
Px.6	AD6*	CC6	A14*	-	-	ADC6	SCL_MC
Px.7	AD7*	T2	A15*	-	-	ADC7	SDA_MC

Remark: The PWM0, PWM1, PSEN, ALE and EA signals have dedicated pinning and cannot be used as general purpose I/O.

7.6.1 Pads

7.6.1.1 Software Configurable Pads

Ports 0 - 4 and bits 0 - 5 of Port 6 are implemented as software-configurable pads which can be and can be individually programmed to operate in one of four modes: Open-Drain, Quasi-Bidirectional, Push-Pull, and High-Impedance. The operating mode is selected by the port configuration SFRs. A schematic of the pad is given in Figure 257.



1 - NMOS strong pull-down.

2 - PMOS strong pull-up.

3 - Very weak pull-up.

4 - Weak hold (latch).

Fig 257.Software-configurable pad schematic

Mode 0: Open Drain (Input/Output): Only the NMOS strong pull-down transistor of the pad cell is driven. When the output data is '0', the pad cell pin is pulled down to '0'. When the output data is '1', the pad cell pin is in the high impedance state. An external pull-up resistor is required to produce a logic '1'. This enables the pad to be used as an input when the corresponding port SFR bit is '1'.

Mode 1: Quasi-Bidirectional (Input/Output): The PMOS strong pull-up transistor and the NMOS strong pull-down transistor are driven individually. When the output data is '0', the pad cell pin is pulled down to '0' by the NMOS strong pull-down transistor. When the output data transitions from '0' to '1', the pad cell pin is pulled up to '1' for one clock cycle by the PMOS strong pull-up transistor. After that transition, this state is held by the weak hold transistor, which implements a latch function. Because of the weaker nature of this hold transistor, the pad can now act as an input as well. This enables the pad to be used as an input when the corresponding port SFR bit is '1'. The very weak pull up transistor ensures that an open input is read as '1'.

Mode 2: High Impedance (Input): All transistors are placed in the high impedance state, allowing the pad to be used as in input.

Mode 3: Push-Pull (Output): The PMOS strong pull up transistor and NMOS strong pull down transistor are driven according to the output data, and the hold transistor and the weak pull-up transistor are placed in the high impedance state. Thus, the pad drives a "hard 0" or "hard 1" value continuously, and input is impossible.

Mode Switching: When switching from one mode to another, two SFR write cycles are necessary to switch from open-drain and push-pull, or between quasi-bidirectional and high impedance. After the first write cycle, the port is configured in an undesired mode. Special care must be taken when switching between quasi-bidirectional and high impedance. This must always be done by switching to open-drain at the first write cycle, because push-pull may damage external devices.

7.6.1.2 Analog Input Pads

Port 5 serves as the analog input to the ADC. Reads from this port return a "digital" representation of the analog voltage present on the pads, thus this port can provide additional digital inputs. Port 5 cannot be used for output.

7.6.1.3 I^{2}C Pads

Bits 6 and 7 of Port 6 are implemented as I^{2}C pads, which are functionally similar to open-drain pads. When the output data is '0', the pad cell pin is pulled down to '0', otherwise the pad cell pin is in the high impedance state. Note that although the corresponding configuration register bits are physically present (as for software configurable pads), these pads do not support the push-pull or quasi-bidirectional modes.

7.6.2 Alternate Functions

For ports which have an alternate *output* function, the alternate data is ANDed with the value of the port SFR. Thus, the port SFR is written as '1' to enable the alternate output.

Remark: This restriction does not apply to the alternate functions for P0 and P2, which are only used in Metalink emulation mode.

7.6.3 SFRs

The general purpose I/O signals are accessed via the P# SFRs (where # represents the port number). Writes to these registers set the data for output on the I/O signals (when the pads are configured as outputs). Reads from the P# registers return the data value present on the pad pins (regardless of the pad mode). Read-modify-write (RMW) instructions (shown in Table 374) access the value of the port SFR rather than the pad value. The P#.n SFR is written as '1' to enable the corresponding alternate function.

Table 374: RMW Instructions

Instruction	Arguments
anl orl xrl	direct, A direct, #data
clr cpl setb	bit
dec inc	direct
djnz	direct, rel
jbc	bit, rel
mov	bit, c

[1] **direct** - 128 main RAM locations and any SFR.

[2] **#data** - 8-bit constant.

[3] **bit** - direct addressed bit in main RAM or SFR.

[4] **rel** - signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is -128 to +128 bytes relative to the first byte of the following instruction.

[5] **A** - accumulator

[6] **c** - carry bit of PSW (PSW.7).

Table 375: P# register description

Symbol	Position	Description
P#	7:0	Port # Data Register

The software-configurable pad operation is controlled by the P#CFGA and P#CFGB SFRs. The mode selection is given in Table 376.

Table 376: Software-configurable pad settings

P#CFGA.n	P#CFGB.n	P#.n mode
0	0	Open Drain
0	1	Quasi-Bidirectional
1	0	High Impedance
1	1	Push Pull

Table 377: P# configuration register A

Symbol	Position	Description
P#CFGA	7:0	Port # Configuration Register A

Table 378: P# configuration register A

Symbol	Position	Description
P#CFGB	7:0	Port # Configuration Register B

7.7 Timer 0/1

7.7.1 Programmer’s view

Timer 1 basic functionality is identical to Timer 1 of the 8XC51 products (described in *Databook IC20, Chapter 2*). The Timer 1 module has two 16-bit Timer/Counter registers: Timer 0 and Timer 1. Both can be configured as timers or event counters.

In the “timer” function, the register is incremented every machine cycle. The count rate is 1/6 of the clock clk frequency.

In the “counter” function, the register is incremented in response to a 1-to-0 transition at its input pin P3.4 (Timer 0) or P3.5 (Timer 1). In this function the external input is sampled during cpu state 5 of every machine cycle. When the sample shows a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register in cpu state 3 of the machine cycle following the one in which the transition was detected. The maximum count rate is 1/12 of the clock clk frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it is held for at least one full machine cycle.

In addition to timer/counter selection, Timer 0 and Timer 1 have four operating modes from which to select (see Section 7.7.4). Both Timer 0 and Timer 1 modes can be chosen independently to each other, their modes need not be the same. However mode 3 has some exceptions, see Section 7.7.4.2.

The overflow output of Timer 1 can be used as a baud rate generator to drive the UART. The Timer1 interrupt (T1) should be disabled in this application. In the most typical applications, when driving the UART, Timer 1 is configured for ’timer’ operation, in the auto-reload mode (see Section 7.7.4.1).

The overflow output can also be used as a baud-rate generator for the I^{2}C interface, and as a count trigger for the PCA.

The operation of the two external interrupts EX0 and EX1 is also controlled by the timer01 block.

7.7.2 Special Function Registers

7.7.2.1 T01CON register

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Fig 258.Timer/Counter 2 control (T01CON) register

Table 379: T01CON Register Description

Symbol	Position	Name	Description
TF1	T01CON.7	Timer 1 overflow flag	Set by hardware on a Timer 1 overflow. Cleared by hardware when the interrupt is processed.
TR1	T01CON.6	Timer 1 run control bit	Set/cleared by software only. When set the timer/counter 1 is turned on. When cleared the timer/counter 1 is turned off.
TF0	T01CON.5	Timer 0 overflow flag	Set by hardware on a Timer 0 overflow. Cleared by hardware when the interrupt is processed.

Table 379: T01CON Register Description *…continued*

Symbol	Position	Name	Description
TR0	T01CON.4	Timer 0 run control bit	Set/cleared by software only. When set the timer/counter 0 is turned on. When cleared the timer/counter 0 is turned off.
IE1	T01CON.3	Interrupt 1 edge flag	Set by hardware when an external interrupt (EX1) is detected on input *P3.3*. Cleared by hardware when the interrupt is processed.
IT1	T01CON.2	Interrupt 1 type control bit	Set/cleared by software only. When set, interrupt EX1 triggers on a falling edge of input *P3.3*. When cleared, interrupt EX1 triggers on a low level of input *P3.3*.
IE0	T01CON.1	Interrupt 0 edge flag	Set by hardware when an external interrupt (EX0) is detected on input *P3.2*. Cleared by hardware when the interrupt is processed.
IT0	T01CON.0	Interrupt 0 type control bit	Set/cleared by software only. When set, interrupt EX0 triggers on a falling edge of input *P3.2*. When cleared, interrupt EX0 triggers on a low level of input *P3.2*.

7.7.2.2 T01MOD register

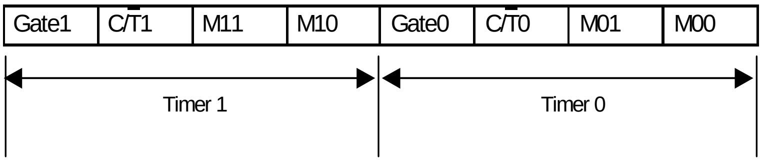


Fig 259.Timer/Counter 2 control (T01MOD) registers

Table 380: T01MOD Register Description

Symbol	Position	Name	Description
Gate1	T01MOD.7	Timer 1 gating control	Set/cleared by software only. When set, timer/counter 1 is enabled only when input *P3.3* is high and T01CON.TR1 is set. When cleared timer/counter 1 is enabled whenever T01CON.TR1 is set.
C/T1	T01MOD.6	Timer 1 or Counter 1 selector	Set/cleared by software only. When set, timer/counter 1 is set to counter operation (input from input *P3.5*). When cleared timer/counter 1 is set to timer operation (input from system clock).
M11	T01MOD.5	Timer/counter 1 mode	Set/cleared by software only. The timer/counter1 has four modes.
M10	T01MOD.4	Timer/counter 1 mode	Mode 0: 8048-compatible 13-bit timer Mode 1: 16-bit timer/counter mode 2: 8-bit timer/counter with auto-reload Mode 3: Halt
Gate0	T01MOD.3	Timer 0 gating control	Set/cleared by software only. When set, timer/counter 0 is enabled only when input *P3.2* is high and T01CON.TR0 is set. When cleared timer/counter 0 is enabled whenever T01CON.TR0 is set.
C/T0	T01MOD.2	Timer 0 or Counter 0 selector	Set/cleared by software only. When set, timer/counter 0 is set to counter operation (input from input *P3.4*). When cleared timer/counter 0 is set to timer operation (input from system clock).
M01	T01MOD.1	Timer/counter 0 mode	Set/cleared by software only. The timer/counter0 has four modes.
M00	T01MOD.0	Timer/counter 0 mode	Mode 0: 8048-compatible 13-bit timer Mode 1: 16-bit timer/counter mode 2: 8-bit timer/counter with auto-reload Mode 3: dual 8-bit timer/counter

7.7.2.3 T0H, T0L registers

These two 8-bit registers are normal registers in the SFR space. These are the actual timer/counter registers for Timer 0. T0L is the least significant byte of timer/counter 0, T0H is the most significant byte of timer/counter 0.

7.7.2.4 T1H, T1L registers

These two 8-bit registers are normal registers in the SFR space. These are the actual timer/counter registers for Timer 1. T1L is the least significant byte of timer/counter 1, T1H is the most significant byte of timer/counter 1.

7.7.3 Timer modes in general

7.7.3.1 Incrementer

The two 16-bit timer/counters are built around one 8-bit incrementer. The registers are incremented in the first four states (overflow flags are set in state 2 and 4).

1. T0L is incremented if timer operation of timer/counter 0 is active, or when a 1-to-0 transition is detected on *P3.4* input and counter operation is active.
2. T0H is incremented if T0L overflows. The overflow flag T01CON.TF0 is updated.
3. T1L is incremented if timer operation of timer/counter 1 is active, or when a 1-to-0 transition is detected on *P3.5* input and counter operation is active.
4. T1H is incremented if T1L overflows. The overflow flag T01CON.TF1 is updated.

7.7.3.2 Overflow detection

An overflow is detected by comparing the incremented value of the most significant bit with its previous value. If the bit changed from 1 to 0, the register overflowed. An overflow detection of the lower byte register is clocked into a flip-flop and is used in the next state as the increment enable of the upper byte registers. An overflow detection of the upper byte registers sets the corresponding overflow bit in the T01CON register. The upper byte overflow is also clocked into a flip-flop to generate the overflow output signal for the UART, I2C, and PCA.

Overflow T01CON.TF0 is loaded during cpu state 2 and overflow T01CON.TF1 during cpu state 4. The interrupt controller scans all requests at cpu state 2. Thus, an overflow of Timer 0 or Timer 1 is detected one cycle after it overflowed. Thereafter, if the request is serviced, the interrupt routine is called and the overflow flag is cleared. Execution of the interrupt routine starts in the fourth cycle after the timer overflowed. If in one cycle an overflow occurs and an acknowledge comes from the cpu, then the overflow flag is set. So a set overrules a reset.

7.7.3.3 Emulator break

In Enhanced Hooks emulation mode, the emulator break disables timer01 incrementing and edge detection of external timer and interrupt inputs.

7.7.4 Timer/Counter 0 and 1 modes

Table 381: Timer 0 Modes

Mode	Mode function	Description
0	13-bit counter	In mode 0, timer0 acts as a 8048-compatible 13 bit timer. T0L acts as a 5-bit prescaler.
1	16-bit timer/counter	T0L and T0H are cascaded to form a 16-bit timer/counter.
2	8-bit timer/counter with autoreload	T0H holds a value which is automatically loaded into T0L each time T0L overflows.
3	dual 8-bit timer/counter	Timer/counter 0 is split into two 8-bit timer/counters, T0H and T0L. T0H is controlled by bit TR1 of the T01CON register. T0L is controlled by bit TR0 of the T01CON register and bit Gate0 of the T01MOD register.

Table 382: Timer 1 Modes

Mode	Mode function	Description
0	13-bit counter	In mode 0, timer1 acts as a 8048-compatible 13 bit timer. T1L acts as a 5-bit prescaler.
1	16-bit timer/counter	T1L and T1H are cascaded to form a 16-bit timer/counter.
2	8-bit timer/counter with autoreload	T1H holds a value which is automatically loaded into T1L each time T1L overflows.
3	Halt	Timer1 is stopped, and holds its previous count value.

Remark: If timer1 is not in mode 3, placing timer0 in mode 3 also enables timer1, regardless of the value of TR1. (GATE1 is still in effect.) The interrupt is not available to timer1 when timer0 is in mode 3.

7.7.4.1 Event detection on external timer pins

In the counter function the timer register is incremented in response to a 1-to-0 transition on the external event pin of the timer (inputs *P3.4/P3.5*). To detect negative transitions, *P3.4* and *P3.5* are sampled during cpu state 5 of every machine cycle (except during emulator break). When samples show a high in one cycle and a low in the next, the counter is incremented.

7.7.4.2 External interrupts EX0 and EX1

The Timer 1 samples *P3.2* and *P3.3* every cpu state 5. Depending on bits T01CON.IT0/IT1, the external interrupts (EX0 and EX1 respectively) are triggered by a low level, or by a 1-to-0 transition. The bits T01CON.IE0 and T01CON.IE1 are set in cpu state 1 and reset in cpu state 2. During emulator break, bits T01CON.IE0 and T01CON.IE1 are not set or reset by hardware.

7.8 Timer 2

7.8.1 Programmer’s view

Timer 2 has six Special Function Registers (SFR) that can be read and written by the CPU. These registers are: T2CON, T2MOD, T2H, T2L, T2RCH and T2RCL. Timer 2 basic functionality is identical to the Timer 2 module of the 80C51FB product. In addition to the 80C51FB the uCMS Timer 2 module has extra bits in the T2CON register to use as baudrate clock signals for the UART in the TV Microcontroller. Timer 2 can operate either as a timer or as an event counter. This is selected by bit T2CON.C/T2. Timer 2 can operate in one of four different modes: Capture mode, Auto-reload mode, Baud rate generation mode (for the Uart) or Clock output mode. The four modes are selected by several bits of the T2CON and T2MOD registers. The four modes are described in detail in Section 7.8.3 to Section 7.8.3.4.

Timer 2 register values can change by hardware or by software. In one machine cycle, the write by software takes place in state 6 of the machine cycle. When in one machine cycle both an update by hardware and software occurs in one of the registers T2H, T2L, T2RCH or T2RCL, the update by software has precedence over the update by hardware. Each increment or decrement of Timer 2 occurs in state 1 except when in baud rate generation mode and configured as a counter. In this mode Timer 2 increments each clock cycle.

7.8.2 Special Function Registers

7.8.2.1 T2CON register

TF2	EXF2	RCLK0	TCLK0	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

Fig 260.Timer/Counter 2 control (T2CON) register

Table 383: T2CON register description

Symbol	Position	Name	Description
TF2	T2CON.7	Timer 2 overflow flag	Set by a Timer 2 overflow and should be cleared by software. TF2 will not be set when baud rate generation mode, clock out mode or metalink mode is selected (see Section 7.8.3).
EXF2	T2CON.6	Timer 2 External Flag	Set on a negative transition on *P1.7* and T2CON. EXEN2='1'. In auto-reload mode it is toggled on an under- or overflow (not in metalink mode, intd = 0). This bit should be cleared by software.
RCLK0	T2CON.5	Receive clock UART flag	Set by software only. When set, it causes UART0 to use Timer 2 overflow pulses. When reset, it causes UART0 to use overflow pulses from Timer 1.
TCLK0	T2CON.4	Transmit clock UART flag	Set by software only. When set, it causes UART0 to use Timer 2 overflow pulses. When reset, it causes UART0 to use overflow pulses from Timer 1.
EXEN2	T2CON.3	Timer 2 external enable flag	Set by software only. When set, allows a capture or reload to occur, together with an interrupt, as a result of a negative transition on input *P1.7* if in capture mode or auto reload mode with DCEN reset. If in auto reload mode and DCEN is set, the EXEN2 bit has no influence. In the other modes EXF2 is set and an interrupt is generated on a 1 to 0 transition on P1.7 pin. When EXEN2 is reset, Timer 2 ignores events at *P1.7* in all modes.
TR2	T2CON.2	Start/stop control for timer	Set by software only. When set, the timer is started. When reset the timer is stopped.
C/T2	T2CON.1	Timer/counter select for timer	Set by software only. When set the counter function is selected, when reset the timer function is selected.
CP/RL2	T2CON.0	Capture/Reload flag	Set by software only. Selection of mode capture or reload. When set the capture function is selected, when reset the reload function is selected. When baud rate generation mode is selected this bit is ignored and Timer 2 is forced to auto-reload on an overflow.

Remark: The external count input and trigger input both share the I/O pin P1.7 Consequently, it is not useful to enable the external capture/reload trigger (EXEN2 = 1) when the counter function is selected (C/T2 = 1).

7.8.2.2 T2MOD register

–	–	-	-	–	T2RD	T2OE	DCEN

Fig 261.Timer/Counter 2 mode control (T2MOD) register

Table 384: T2MOD register description

Symbol	Position	Name	Description
	T2MOD.7	Not implemented.	-
	T2MOD.6	Not implemented.	-
RCLK1	T2MOD.5	Not implemented.	-
TCLK1	T2MOD.4	Not implemented.	-
	T2MOD.3	Not implemented.	-
T2RD	T2MOD.2	Timer 2 Read flag.	Set/reset by hardware and software. This bit is set by hardware if a T2L read operation is followed by an increment of T2H before a T2H read operation. This bit is reset on the trailing edge of next T2L read. This bit is used to indicate that the16 bit Timer 2 register is not read properly since the T2H part was incremented by hardware before it was read.
T2OE	T2MOD.1	Timer 2 output enable bit	Set by software only. When set and T2CON.TF2 is reset and T2CON.EXF2 is reset, output *P1.7* outputs a clock signal. The clock output is half the overflow frequency of Timer 2 (see Section 7.8.3.4).
DCEN	T2MOD.0	Down Count Enable flag	Set by software only. When this bit is set and input P1.7 is set, Timer 2 can be configured (in auto_reload mode) as an up counter. When this bit is reset or input *P1.7* is reset, Timer 2 can be configured (in auto-reload mode) as a down counter.

7.8.2.3 T2H, T2L registers

These registers are normal registers in the SFR space. They are the actual timer/counter registers. Momentary reading can give a false value since T2H can be changed after T2L is read and before T2H is read. This situation is indicated by flag T2RD in T2MOD. In all cases the two 8-bit registers operate as one 16-bit timer/counter register.

7.8.2.4 T2RCH, T2RCL registers

These registers are normal registers in the SFR space. They are the capture and reload registers depending on the chosen operation mode. In the capture mode the T2RCH/T2RCL registers are loaded with the value of the T2H/T2L registers. In the reload mode the T2H/T2L registers are loaded with the value of the T2RCH/T2RCL registers.

7.8.3 Timer modes

Timer 2 can operate in four different modes:

- Capture mode.
- Auto-reload mode.
- Baud rate generation mode.
- Clock output mode.

Table 385 gives the list of T2CON and T2MOD register bits which set the Timer 2 mode of operation. Clock output mode and baud rate generation mode can happen simultaneously.

Table 385: Timer 2 Mode Setting

RCLK0 + TCLK0 + RCLK1 + TCLK1	CP/RL2	T2OE	C/T2	Operating Mode
0	0	0	X	16-bit Auto Reload
0	1	0	X	16-bit Capture
1	X	X	X	Baud Rate Generation
X	0	1	0	Clock Out

In most of the modes the timer/counter operates on events detected on input *P1.7* . The u80c51_timer2 module synchronizes the input *P1.7* in cpu state 5.

7.8.3.1 Capture mode

In the capture mode, registers T2RCH/T2RCL are used to capture the T2H/T2L register data. There are two options selected by the T2CON.EXEN2 bit. This bit enables or disables the events of the external trigger input *P1.7*.

Table 386: Capture mode

Register	Description
T2CON.C/T2 = 1	Timer 2 is a 16-bit counter. The counter increments at each low to high event on input *P1.7* at a maximum rate of one each 12 clk cycles. Increments are done in cpu state 1.
T2CON.C/T2 = 0	Timer 2 is a 16-bit timer. The timer increments each machine cycle (increments each 6 clk cycles). Increments are done in cpu state 1.
T2CON.EXEN2 = 1	The external trigger input *P1.7* is enabled. Timer 2 is a 16-bit timer or counter. If T2MOD.DCEN = 0, a 1-to-0 transition at input *P1.7* causes the current Timer 2 value (T2H/T2L data) to be captured into T2RCH/T2RCL, and bit T2CON.EXF2 becomes set. If T2MOD.DCEN = 1, bit T2CON.EXEN2 has no influence. Overflowing of Timer 2 sets bit T2CON.TF2.
T2CON.EXEN2 = 0	The external trigger input *P1.7* is disabled. Timer 2 is a 16-bit timer or counter. Overflowing of Timer 2 sets bit T2CON.TF2.

7.8.3.2 Auto-reload mode

In auto-reload mode, when the C/T2 bit is set, Timer2 acts as a 16-bit up-counter. The counter increments at each low to high event on the input P1.7, at a maximum rate of one increment per 12 clock cycles. The DCEN and EXEN2 bits are cleared in counter mode, because the count input and external trigger input both share the P1.7 I/O pin. Overflowing Timer2 causes the value in T2RCH/T2RCL to be reloaded into the T2H/T2L registers and sets bitTF2.

Counting Up (DCEN=0): In the auto-reload mode and counting up, registers T2RCH/T2RCL are used to hold a reload value for T2H/T2L. The external trigger input (P1.7) is enabled when the EXEN2 bit is set. The T2 interrupt is asserted if the EXF2 bit is set or if the TF2 bit is set.

Table 387: DCEN=0; counting up

Register	Description
T2CON.EXEN2=1	The external trigger input P1.7 is enabled. Timer 2 is a 16-bit timer. A high to low event at input P1.7 causes the value in T2RCH/T2RCL to be reloaded into the Timer2 T2H/T2L registers, and the EXF2 bit is set. Also, an overflow of Timer2 causes the value in T2RCH/T2RCL to be reloaded in the T2H/T2L registers and sets bit TF2.
T2CON.EXEN2=0	The external trigger input P1.7 is disabled. Timer2 is a 16-bit timer. Overflowing of Timer2 causes the value in T2RCH/T2RCL to be reloaded in the T2H/T2L registers and sets bit TF2.

Counting Up (DCEN=1 and P1.7=1): The high value of the input P1.7 sets Timer2 to a count up mode. The registers T2RCH/T2RCL are used to hold a reload value for T2H/T2L. Overflowing of Timer2 causes the values in T2RCH/T2RCL to be reloaded in the T2H/T2L registers, sets bit TF2, and toggles bit EXF2 (thus EXF2 can be used as a 17th timer bit if desired). The T2 interrupt is generated if TF2 is set.

Counting Down (DCEN=1 and P1.7=0): The low value of the input P1.7 sets Timer2 to a count down mode. The registers T2RCH/T2RCL are used to hold a value for detecting an underflow of T2H/T2L. Underflow occurs if the contents of T2H/T2L matches the contents of T2RCH/T2RCL. Upon underflow, bit TF2 is set and registers T2H/T2L are loaded with 0sffff, bit TF2 is set, and bit EXF2 toggles. Note that a Timer2 roll over from 0x0000 to 0xffff is not considered an underflow (unless T2RCH/T2RCL = 0x0000). The T2 interrupt is generated if TF2 is set.

7.8.3.3 Baud rate generation mode

When one of the T2CON.RCLK0 or T2CON.TCLK0 bits are set, Timer 2 is in baud rate generation mode.

- T2CON.TCLK0 = 1, Timer 2 is used as the serial port transmitter baud rate generation for the Uart.
- T2CON.RCLK0 = 1, Timer 2 is used as the serial port receiver baud rate generation for the Uart.

T2CON.C/T2 = 1:Timer 2 is a 16 bit counter. The counter increments at each low to high event on input *P1.7* at a maximum rate of one each 12 clk cycles.

The baud rate of the UART is:

$$Baud\ rate\ =\ \frac{Timer2overflowrate}{16}$$

T2CON.C/T2 = 0:Timer 2 is a 16 bit timer. The timer increments each clk cycle.

The baud rate of the UART is:

$$Baud\ rate\ =\ \frac{f_{clk}}{16 \times [65536 - (T2RCH,\ T2RCL)]}$$

In this mode, a timer overflow loads T2H/T2L with the contents of T2RCH/T2RCL and does not set TF2.

Bit T2CON.EXF2 is set if T2CON.EXEN2 is set and a high to low transition is detected on *P1.7* pin. The T2 interrupt is set only if T2CON.EXF2 is set. This makes an extra external interrupt available.

Remark: The external interrupt is not usable when C/T2 is set, as the count input and the external trigger input share the P1.7 I/O pin.

7.8.3.4 Clock output mode

In the clock output mode, the output *P1.7* is enabled as a clock output. A timer overflow causes T2H/T2L to be loaded with T2RCH/T2RCL and toggle output *P1.7*. The frequency of pin *P1.7* is half the overflow frequency.

It is possible to configure Timer 2 in clock output mode and baud rate generation mode simultaneously.

7.9 Watchdog Timer (WDT)

7.9.1 Overview

The watchdog timer is a counter that on overflow forces the microcontroller into a hard reset. When enabled, the watchdog timer generates a system reset if the user program fails to reload the watchdog within a certain (software programmable) time, known as the watchdog interval. The watchdog timer consists of an 8-bit timer and a 13 bit prescaler.

7.9.2 General description

The prescaler and the WDT SFR are combined into one long timer. The prescaler bits are the low order bits of the timer, the WDT register are the highest order bits of the timer. The timer is clocked in every cpu state 2. The prescaler is reset when the watchdog timer is initially enabled, when the WDT register is written, or when the watchdog timer is disabled (via the WDTKEY register). In emulation mode the prescaler holds its current value.

The 13 bit prescaler overflows every ‘t’ seconds, where:

$$t = \frac{\text{states / cycle} \ * \ 2^{\text{pre\_width}}}{f_{\text{clk}}} = \frac{6 \ * \ 2^{13}}{f_{\text{clk}}} = 3.072 \text{ ms at } 16 \text{ MHz}$$

7.9.3 Operation

After a reset, the WDT contains all zeros and keeps this value until it is enabled. The WDT is enabled by the first time the PCON.WLE bit (bit 4) is set. Once enabled, the WDT remains enabled.

Once the WDT is enabled it starts counting. If no actions are taken it overflows after a certain period of time and resets the TV Microcontroller. To avoid this reset, the WDT has to be reloaded before it overflows. The user program should therefore continually execute sections of code which reload the WDT. The period of time elapsed between execution of these sections may never exceed the watchdog timer interval.

The watchdog timer is reloaded in two stages:

1. The WLE bit in the PCON SFR must be set. This is a safety mechanism to prevent erroneous software from reloading the watchdog timer.
2. The WDT register is loaded. This automatically clears the WLE bit.

The WDT remains active in idle-mode. In order to get to power down mode the watchdog timer should be disabled by the WDT key register WDTKEY.

7.9.4 Programmer's view

7.9.4.1 WDT registers

The WDT contains two Special Function Registers (SFRs) that can be read and written by the CPU.

The WLE and EW bits in the PCON SFR also affect the operation of the watchdog timer (see Section 7.9).

7.9.4.2 WDT register

This register is the lower part of the watchdog counter. The incremented value is clocked into the WDT register in cpu state 2. The WDT register can only be written when the WLE bit in the PCON register is set.

7.9.4.3 WDTKEY register

This register is the watchdog key register. When the value 55h is written, the WDT is disabled. This also inhibits setting the PD bit in the PCON SFR. The only way to enable the watchdog again is to load WDTKEY with a different value than the key value 55h.

7.10 PCA (Programmable Counter Array)

7.10.1 Functional description

The PCA contains a special timer module that has seven 16-bit Capture/Compare (CC) modules associated with it. Each of these modules can be programmed to operate in one of four modes (described in detail in Section 7.10.4 and Section 7.10.5). The modes are:

- Capture mode, using rising and/or falling edges of the module input P1.x. This captures the 16-bit value of the main timer into the 16-bit CC-register on a pre-defined event.
- Software timer/compare mode.This compares a 16-bit value stored in the CC-register against the contents of the main timer. A flag is set when the values match.
- Toggle output mode. This is similar to the timer mode, except a port pin is toggled when the CC-register and main timer values match.
- Pulse width modulation mode. The lower 8-bit of the CC-register are compared against the lower 8-bits of the main timer. If the CC-register is less than or equal to the main timer, output pin is low else output pin is high.

All modules share the timer/counter as their timebase. The timer/counter can select as trigger source one of three sources (described in detail in Section 7.10.4).

- Trigger on negative edge of the cexin input pin.
- Trigger on an active ovfin input pin.
- Trigger on a divided clock clk (division factors: 1/6, 1/12, 1/24, 1/48, 1/96, 1/192).

The PWM contains four Special Function Registers (SFR's) in the timer/counter module and additional 3 SFR's for each CC module. The register bit definitions are described in Section 7.10.2 and Section 7.10.3.

The PCA can generate 7 interrupts; one individual interrupt signal for each CC module. Interrupts are stored in the PCACCON SFR and can be identified and cleared by software. Interrupts are described in detail in Section 7.10.6.

7.10.2 PCA timer/counter registers

7.10.2.1 PCACMOD register

-	CR	-	-	CP2	CP1	CP0	ECF

Fig 262.PWMCMOD Register

Table 388: PCACMOD register description

Symbol	Position	Name	Description
	7	Not implemented	-
CR	6	Counter run control bit	Set/cleared by software only. When set, the PCA counter is turned on. When cleared, the PCA counter is turned off.
	5	Not implemented	-
	4	Not implemented	-
CPS2	3	Count source select bit 2	See description at PCACMOD.1 bit.
CPS1	2	Count source select bit 1	See description at PCACMOD.1 bit.
CPS0	1	Count source select bit 0	Set/cleared by software only. The 8 combinations of CPS[2:0] can select 8 different increment control sources for the PWM counter. See Section 7.10.4 for details.
ECF	0	Not implemented	-

7.10.2.2 PCACCON register

CF	CCF6	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0

Fig 263.PCACCON register

Table 389: PCACCON register description

Symbol	Position	Name	Description
CF	7	PWM Counter overflow flag	Set by hardware when the counter rolls over. Cleared by software only.
CCF6	6	PWM module 6 interrupt flag	Set by hardware when the module register matches or captures the PCA counter value. Cleared by software only.
CCF5	5	PWM module 5 interrupt flag	Set by hardware when the module register matches or captures the PCA counter value. Cleared by software only.
CCF4	4	PWM module 4 interrupt flag	Set by hardware when the module register matches or captures the PCA counter value. Cleared by software only.

Table 389: PCACCON register description *…continued*

Symbol	Position	Name	Description
CCF3	3	PWM module 3 interrupt flag	Set by hardware when the module register matches or captures the PCA counter value. Cleared by software only.
CCF2	2	PWM module 2 interrupt flag	Set by hardware when the module register matches or captures the PCA counter value. Cleared by software only.
CCF1	1	PWM module 1interrupt flag	Set by hardware when the module register matches or captures the PCA counter value. Cleared by software only.
CCF0	0	PWM module 0interrupt flag	Set by hardware when the module register matches or captures the PCA counter value. Cleared by software only.

The PCACCON SFR supports a locking mechanism to prevent software read-modify-write instructions to overwrite the contents while hardware is modifying the contents of the register.

7.10.2.3 PCAH/PCAL registers

The PCAH/PCAL registers form the 16-bit counter. This counter is a common time base for all modules and can be programmed to increment at one of the following:

- the divided frequencies of the input clock
- the overflow output of timer 1
- the input P1.7

The timer count source is determined from the three PCACMOD.CPS bits.

The PCAH and PCAL registers can change by hardware and by software. When in one machine cycle both an update by hardware and software occurs, the update by software has precedence over the update by hardware. The PCAH/PCAL registers can be read by software.

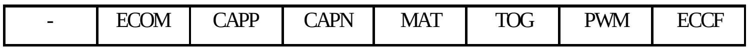
To enable the PCA-counter the PCACMOD.CR bit should be set by software. The PCA-counter is disabled (not cleared) by clearing the PCACMOD.CR bit and is disabled in metalink emulation mode.

7.10.3 PCA module registers

The PCA contains three SFRs for each CC module: PCAM#MOD, PCAM#CCH, and PCAM#CCL (# represents the module-number: 0-6).

7.10.3.1 PCAM#MOD register

Each instantiated module has a PCAM#MOD register.



-	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF

Fig 264.PCAM#MOD register

Table 390: PCAM#MOD register description

Symbol	Position	Name	Description
	7	Not implemented	
ECOM	6	Comparator enable bit	Set/cleared by software and by hardware. When set, the PCA module comparator is enabled. When cleared, the PCA module comparator is disabled. When the PCA#CCL register is written by software, the ECOM bit is cleared by hardware. When the PCA#CCH register is written by software, the ECOM bit is set by hardware.
CAPP	5	Capture positive bit	Set/cleared by software only. When set, the negative edge capture is enabled. When cleared, the negative edge capture is disabled.
CAPN	4	Capture negative bit	Set/cleared by software only. When set, the negative edge capture is enabled. When cleared, the negative edge capture is disabled.
MAT	3	Match bit	Set/cleared by software only. When set, a match between the PCA counter and the CC module register causes the PCACCON.CCF# bit to be set. When cleared, the PCACCON.CCF# bit is not set.
TOG	2	Toggle bit	Set/cleared by software only. When set, a match between the PCA counter and the CC module register causes the *P1.#* output pin to toggle. When cleared, the *P1.#* output is not toggled by a match.
PWM	1	Pulse width modulation mode	Set/cleared by software only. When set, the *P1.#* module output is enabled to be used as a pulse width modulated output. When cleared, the *P1.#* output is disabled to be used as pwm output.
ECCF	0	Enable CCF interrupt	Filled by software only. When set, The PCACCON.CCF# bit is enabled to generate an interrupt, when cleared disabled.

Table 391 lists the valid combinations of the PCAM#MOD register bits

Table 391: Valid combinations of PCAM#MOD register bits

Reserve	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	Module function
-	-	0	0	0	0	0	-	no operation
-	-	1	0	0	0	0	-	16-bit capture pos. edge of P1.7#
-	-	0	1	0	0	0	-	16-bit capture neg. edge of P1.7#
-	-	1	1	0	0	0	-	16-bit capture dual. edge of P1.7#
-	1	0	0	1	0	0	-	16-bit software timer (enabled compare)
-	0	0	0	1	0	0	-	16-bit software timer (disabled compare)
-	1	0	0	1	1	0	-	16-bit high-speed output (enabled compare)
-	0	0	0	1	1	0		16-bit high-speed output (disabled compare)
-	1	0	0	0	0	1	0	8-bit PWM
All other bit combinations								reserved: not specified

7.10.3.2 PCAM#CCH/PCAM#CCL registers

Each instantiated module has two 8-bit registers: PCAM#CCH and PCAM#CCL (# represents module-number 6...0).

The PCAM#CCH/CCL registers are used to capture the 16-bit value of the PCAH/PCAL counter registers. The PCAM#CCH/CCL registers are also compared with the PCAH/PCAL registers. The PCAM#CCH/CCL registers can change by hardware and by software. When in one machine cycle both an update by hardware and software occurs, the update by software has precedence over the update by hardware. When enabled, a capture of the PCAH/PCAL values occurs 2 clock cycles after the edge on input P1.7#. The PCAM#CCH/CCL registers can be read by software.

7.10.4 PCA timer/counter operation

The PCAH/PCAL register is a 16-bit timer counter controlled by bits in the PCACMOD register. The CPS[2:0] Count Pulse Selection bits select the source which is used to increment the PCA counter. There are three different sources:

- The falling edge of the asynchronous input pin P1.7# is used to increment the PCA-counter. The maximum toggle frequency of the input is smaller than half the clock frequency.
- The overflow output from timer 1 is used to increment the PCA-counter.
- The input clock clk is divided by a local clock-divider, the resulting signal is used to increment the PCA counter. Available division factors are: 1/6, 1/12, 1/24, 1/48, 1/96 and 1/192.

The PCACMOD register functionality is listed in Table 392.

Table 392: PCACMOD register functionality

CR	CPS2	CPS1	CPS0	ECF	Module function
0	-	-	-	-	stop PWM counter
1	-	-	-	-	run PWM counter
1	0	0	0	-	increment counter: clk/192
1	0	0	1	-	increment counter: clk/96
1	0	1	0	-	increment counter: clk/48
1	0	1	1	-	increment counter: clk/24
1	1	0	0	-	increment counter: clk/12
1	1	0	1	-	increment counter: clk/6
1	1	1	0	-	increment counter: timer01 overflow
1	1	1	1	-	increment counter: input pin P1.7 (negative edge)

7.10.5 PCA modules

Each PCA module has three special function registers associated with it PCAM#MOD, PCAM#CCL and PCAM#CCH.

The registers PCAM#CCL and PCAM#CCH contain the Capture/Compare values for each module. The PCAM#MOD register contains the bits that control the mode in which each module operates.

7.10.5.1 Capture modes

To use a Capture/Compare module in the capture mode either one or both the PCAM#MOD.CAPN and CAPP bits are set. In the capture mode, the external cexin# input is sampled for a transition. When a valid transition occurs, the value of the PCA-counter (PCAH/PCAL) is clocked into the module capture registers (PCAM#CCH/CCL). The PCACCON.CCF# bit for the module is set and if the PCAM#MOD.ECCF bit is set, an interrupt is generated. When multiple captures occur, the previous captured values are overwritten.

The PCAM#MOD bits CAPN and CAPP determine the active edge of the input P1.7# that triggers the capture. The CAPN bit enables the negative edge, and the CAPP bit enables the positive edge. If both bits are set, both edges of P1.7# trigger a capture.

7.10.5.2 Software timer mode

A PCA module can be used as a software timer by setting bits PCAM#MOD.ECOM/MAT (the PCAM#MOD.CAPN/CAPP/PCA bits should be 0). In the software timer mode, the PCA counter value is compared with the module compare registers (PCAM#CCH/CCL) and when a match occurs, the PCACCON.CCF# bit becomes set. When the corresponding PCAM#MOD.ECCF bit is set an interrupt is generated.

When the modules compare SFRs require reloading, the following steps are to be taken:

1. PCAM#CCL is reloaded, this clears the PCAM#MOD.ECOM bit.
2. PCAM#CCH is reloaded, this sets the PCAM#MOD.ECOM bit.

7.10.5.3 Toggle output mode

The toggle output mode is selected by setting bits PCAM#MOD.ECOM/MAT/TOG (the PCAM#MOD.CAPN/CAPP/PCA bits should be 0). In this mode the P1.# output pin associated with the PCA module toggles each time a match occurs between the PCA counter and the module’s compare registers (PCAM#CCH/CCL).

When the modules compare SFRs require reloading, the following steps are to be taken:

1. PCAM#CCL is reloaded, this clears the PCAM#MOD.ECOM bit.
2. PCAM#CCH is reloaded, this sets the PCAM#MOD.ECOM bit.

To keep the output toggling, the compare registers have to be loaded with new data. This can be done in an interrupt service routine.

7.10.5.4 Pulse width modulator mode

The pulse width modulator mode is selected by setting the PCAM#MOD.ECOM/PCA bits (the PCAM#MOD.CAPN/CAPP/MAT/TOG bits should be 0). In this mode a clock signal is output on module pin *PCA#*. The frequency of the output depends on the source for the PCA counter. All modules set to PCA mode have the same frequency since they share the PCA timer. The duty cycle of each module is independently variable using the low part of the module’s compare register PCAM#CCL. In PCA mode, the PCA# output is 0 if PCAL < PCAM#CCL, and 1 if PCAL >= PCAM#CCL.

When the PCA counter PCAL overflows from FFh to 00h, PCAM#CCL is reloaded with the value in PCAM#CCH. This mechanism allows updating of the PCA without glitches.

When the compare SFRs require reloading only PCAM#CCH should be reloaded, software should not re-define PCAM#CCL while this may cause glitches on *P1.#*.

7.10.6 Interrupt structure

The PCA has the following interrupt register bits:

- PCACCCON.CF
- PCACCON.CCF0

- ......
- PCACCON,CCF6

The PCACCON.CF bit is set by hardware when the PCA counter overflows. The CF bit can only be cleared by software. The PCACCON.CCF# bits (0 through 6) are the interrupt flags for the modules. They are set by hardware when either a match or a capture occurs. These flags can only be cleared by software.

The PCA_CC# interrupt is asserted when the corresponding PCACCON.CCF# and PCAM#MOD.ECCF SFR bits are set.

The PCACCON register is implemented as a bit-addressable SFR. When bit-write instructions are used for PCACCON, interrupt requests are never cleared without being detected.

7.11 PWM (Pulse Width Modulator)

7.11.1 Functional description

The PWM is a cut-down version of the PCA module, which is intended for use as a pulse-width modulator. The PWM has contains two CC modules and has two dedicated output pins (PWM0 and PWM1) that are not shared with general purpose I/O signals from the Ports module. The PWM also has the following restrictions (relative to the main PCA module): the main counter cannot trigger from the timer1 overflow or an external input, the CC modules do not have external capture inputs, and no interrupts are available.

The PWM contains a special timer module that has two 16-bit Capture/Compare (CC) modules associated with it. Each of these modules can be programmed to operate in one of three modes (described in detail in Section 7.10.4 and Section 7.10.5). The modes are:

- Software timer/compare mode.This compares a 16-bit value stored in the CC-register against the contents of the main timer. A flag is set when the values match.
- Toggle output mode. This is similar to the timer mode, except a port pin is toggled when the CC-register and main timer values match.
- Pulse width modulation mode. The lower 8-bit of the CC-register are compared against the lower 8-bits of the main timer. If the CC-register is less than or equal to the main timer, output pin is low else output pin is high.

All modules share the timer/counter as their timebase. The timer/counter can trigger on a divided clock clk (division factors: 1/6, 1/12, 1/24, 1/48, 1/96, 1/192).

The PWM contains four Special Function Registers (SFR’s) in the timer/counter module and additional 3 SFR’s for each CC module. The register bit definitions are described in Section 7.10.2 and Section 7.10.3.

7.11.2 PWM timer/counter registers

7.11.2.1 PWMCMOD register

-	CR	-	-	CP2	CP1	CP0	ECF

Fig 265.PWMCMOD Register

Table 393: PWMCMOD register description

Symbol	Position	Name	Description
	7	Not implemented	-
CR	6	Counter run control bit	Set/cleared by software only. When set, the PWM counter is turned on. When cleared, the PWM counter is turned off.
	5	Not implemented	-
	4	Not implemented	-
CPS2	3	Count source select bit 2	See description at PWMCMOD.1 bit.
CPS1	2	Count source select bit 1	See description at PWMCMOD.1 bit.
CPS0	1	Count source select bit 0	Set/cleared by software only. The 8 combinations of CPS[2:0] can select 8 different increment control sources for the PWM counter. See Section 7.10.4 for details.
ECF	0	Not implemented	-

7.11.2.2 PWMCCON register

CF						CCF1	CCF0

Fig 266.PWMCCON register

Table 394: PWMCCON register description

Symbol	Position	Name	Description
CF	7	PWM Counter overflow flag	Set by hardware when the counter rolls over. Cleared by software only.
CCF6	6	Not implemented	-
CCF5	5	Not implemented	-
CCF4	4	Not implemented	-
CCF3	3	Not implemented	-
CCF2	2	Not implemented	-
CCF1	1	PWM module 1interrupt flag	Set by hardware when the module register matches or captures the PWM counter value. Cleared by software only.
CCF0	0	PWM module 0interrupt flag	Set by hardware when the module register matches or captures the PWM counter value. Cleared by software only.

The PWMCCON SFR supports a locking mechanism to prevent software read-modify-write instructions to overwrite the contents while hardware is modifying the contents of the register.

7.11.2.3 PWMH/PWML registers

The PWMH/PWML registers form the 16-bit counter. This counter is a common time base for all modules and can be programmed to increment at divided frequencies of the input clock.

The clock division ratio is determined from the three PWMCMOD.CPS bits.

The PWMH and PWML registers can change by hardware and by software. When in one machine cycle both an update by hardware and software occurs, the update by software has precedence over the update by hardware. The PWMH/PWML registers can be read by software.

To enable the PWM-counter the PWMCMOD.CR bit should be set by software. The PWM-counter is disabled (not cleared) by clearing the PWMCMOD.CR bit and is disabled in metalink emulation mode.

7.11.3 PWM module registers

The PWM contains three SFRs for each CC module: PWMM#MOD, PWMM#CCH, and PWMM#CCL (# represents the module-number: 0-1).

7.11.3.1 PWMM#MOD register

Each instantiated module has a PWMM#MOD register.

-	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF

Fig 267.PWMM#MOD register

Table 395: PWMM#MOD register description

Symbol	Position	Name	Description
	7	Not implemented	
ECOM	6	Comparator enable bit	Set/cleared by software and by hardware. When set, the PWM module comparator is enabled. When cleared, the PWM module comparator is disabled. When the PWM#CCL register is written by software, the ECOM bit is cleared by hardware. When the PWM#CCH register is written by software, the ECOM bit is set by hardware.
CAPP	5	Not implemented	-
CAPN	4	Not implemented	-
MAT	3	Match bit	Set/cleared by software only. When set, a match between the PWM counter and the CC module register causes the PWMCCON.CCF# bit to be set. When cleared, the PWMCCON.CCF# bit is not set.

Table 395: PWMM#MOD register description *…continued*

Symbol	Position	Name	Description
TOG	2	Toggle bit	Set/cleared by software only. When set, a match between the PWM counter and the CC module register causes the *PWM#* output pin to toggle. When cleared, the *PWM#* output is not toggled by a match.
PWM	1	Pulse width modulation mode	Set/cleared by software only. When set, the *PWM#* module output is enabled to be used as a pulse width modulated output. When cleared, the PWM# output is disabled to be used as pwm output.
ECCF	0	Not implemented	-

Table 391 lists the valid combinations of the PWMM#MOD register bits

Table 396: Valid combinations of PWMM#MOD register bits

Reserve	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	Module function
-	-	0	0	0	0	0	-	no operation
-	1	0	0	1	0	0	-	16-bit software timer (enabled compare)
-	0	0	0	1	0	0	-	16-bit software timer (disabled compare)
-	1	0	0	1	1	0	-	16-bit high-speed output (enabled compare)
-	0	0	0	1	1	0		16-bit high-speed output (disabled compare)
-	1	0	0	0	0	1	0	8-bit PWM
All other bit combinations								reserved: not specified

7.11.3.2 PWMM#CCH/PWMM#CCL registers

Each instantiated module has two 8-bit registers: PWMM#CCH and PWMM#CCL (# represents module-number 1...0).

The PWM#CCH/CCL registers contain a value to be compared with the PWMH/PWML registers. The PWMM#CCH/CCL registers can change by hardware and by software. When in one machine cycle both an update by hardware and software occurs, the update by software has precedence over the update by hardware. The PWMM#CCH/CCL registers can be read by software.

7.11.4 PWM timer/counter operation

The PWMH/PWML register is a 16-bit timer counter controlled by bits in the PWMCMOD register. The CPS[2:0] Count Pulse Selection bits select the clock division ratio which is used to increment the PWM counter. Available division factors are: 1/6, 1/12, 1/24, 1/48, 1/96 and 1/192.

The PWMCMOD register functionality is listed in Table 392.

Table 397: PWMCMOD register functionality

CR	CPS2	CPS1	CPS0	ECF	Module function
0	-	-	-	-	stop PWM counter
1	-	-	-	-	run PWM counter
1	0	0	0	-	increment counter: clk/192
1	0	0	1	-	increment counter: clk/96

Table 397: PWMCMOD register functionality *…continued*

CR	CPS2	CPS1	CPS0	ECF	Module function
1	0	1	0	-	increment counter: clk/48
1	0	1	1	-	increment counter: clk/24
1	1	0	0	-	increment counter: clk/12
1	1	0	1	-	increment counter: clk/6
-	-	-	-	-	Reserved
1	1	1	1	-	Reserved

7.11.5 PWM modules

Each PWM module has three special function registers associated with it PWMM#MOD, PWMM#CCL and PWMM#CCH.

The registers PWMM#CCL and PWMM#CCH contain the Capture/Compare values for each module. The PWMM#MOD register contains the bits that control the mode in which each module operates.

7.11.5.1 Software timer mode

A PWM module can be used as a software timer by setting bits PWMM#MOD.ECOM/MAT (the PWMM#MOD.CAPN/CAPP/PWM bits should be 0). In the software timer mode, the PWM counter value is compared with the module compare registers (PWMM#CCH/CCL) and when a match occurs, the PWMCCON.CCF# bit becomes set.

When the modules compare SFRs require reloading, the following steps are to be taken:

1. PWMM#CCL is reloaded, this clears the PWMM#MOD.ECOM bit.
2. PWMM#CCH is reloaded, this sets the PWMM#MOD.ECOM bit.

7.11.5.2 Toggle output mode

The toggle output mode is selected by setting bits PWMM#MOD.ECOM/MAT/TOG (the PWMM#MOD.CAPN/CAPP/PWM bits should be 0). In this mode the PWM# output pin associated with the PWM module toggles each time a match occurs between the PWM counter and the module’s compare registers (PWMM#CCH/CCL).

When the modules compare SFRs require reloading, the following steps are to be taken:

1. PWMM#CCL is reloaded, this clears the PWMM#MOD.ECOM bit.
2. PWMM#CCH is reloaded, this sets the PWMM#MOD.ECOM bit.

To keep the output toggling, the compare registers have to be loaded with new data.

7.11.5.3 Pulse width modulator mode

The pulse width modulator mode is selected by setting the PWMM#MOD.ECOM/PWM bits (the PWMM#MOD.CAPN/CAPP/MAT/TOG bits should be 0). In this mode a clock signal is output on module pin *PWM#*. The frequency of the output depends on the clock division ratio for the PWM counter. All modules set to PWM mode have the same frequency since they share the PWM timer. The duty cycle of each module is independently variable using the low part of the module’s compare register PWMM#CCL. In PWM mode, the PWM# output is 0 if PWML < PWMM#CCL, and 1 if PWML >= PWMM#CCL.

When the PWM counter PWML overflows from FFh to 00h, PWMM#CCL is reloaded with the value in PWMM#CCH. This mechanism allows updating of the PWM without glitches.

When the compare SFRs require reloading only PWMM#CCH should be reloaded, software should not re-define PWMM#CCL while this may cause glitches on *PWM#*.

7.11.6 Flag registers

The PWM has the following flag register bits:

- PWMCCCON.CF
- PWMCCON.CCF0
- PWMCCON,CCF1

The PWMCCON.CF bit is set by hardware when the PWM counter overflows. The CF bit can only be cleared by software. The PWMCCON.CCF# bits (0 through 1) are the flags for the modules. They are set by hardware when a match occurs. These flags can only be cleared by software.

7.12 I^{2}C

The I^{2}C module implements a master/slave I^{2}C bus interface with integrated shift register, shift timing generation and slave address recognition. It is compliant to the I^{2}C bus specification IC20/Jan 92. I^{2}C standard mode (100 kHz SCLK) and fast mode (400 kHz SCLK) are supported. Extended 10-bit addressing is not supported. The I^{2}C interface uses pins P6.7 and P6.6 as SDA and SCL, respectively.

7.12.1 Programmer's view

For a detailed description of the I^{2}C bus protocol refer to Philips Integrated Circuits Data Handbook IC20 8XC552.

The programmer's view of the I^{2}C library function is, with one exception, identical to that of the 8XC552 TV Microcontroller. Only bit rate frequency selection in I2C0CON and the handling of the Timer1 overflow information deviates to accommodate 400kHz operation.

The CPU interfaces to the I^{2}C logic - referred to as 'SIO1' - via the following four special function registers:

- I2C0CON (I^{2}C control register)
- I2C0DAT (I^{2}C data register)
- I2C0STA (I^{2}C status register)
- I2C0ADR (I^{2}C slave address register)

7.12.2 I2C0CON

The CPU can read from and write to this 8-bit SFR. Two bits are affected by the I^{2}C hardware: the SI bit is set when a serial interrupt is requested, and the STO bit is cleared when a STOP condition is present on the I^{2}C bus. The STO bit is also cleared when ENS1 = '0'. Reset initializes I2C0CON to 00H.

7	6	5	4	3	2	1	0
CR2	ENS1	STA	STO	SI	AA	CR1	CR0

Fig 268.I2C0CON

7.12.2.1 ENS1 - the SIO1 enable bit

When ENS1 is '0', the SDA and SCL I/Os are in the high impedance state, the I^{2}C interface is in the "not addressed" slave state, and the STO bit in I2C0CON is forced to '0'. No other bits are affected. The SDA and SCL pins (P6.7 and P6.6) may be used as open-drain general purpose I/O ports.

When ENS1 is ‘1’ the I^{2}C interface is enabled. The port latches associated to SDA and SCL are set to logic 1.

ENS1 is not to be used to temporarily release the I^{2}C interface from the I^{2}C bus since, when ENS1 is reset, the I^{2}C bus status is lost. The AA flag is used instead.

7.12.2.2 STA - the START flag

When the STA bit is set to enter a master mode, the I^{2}C hardware checks the status of the I^{2}C bus and generates a START condition if the bus is free. If the bus is not free, then the I^{2}C interface waits for a STOP condition (which frees the bus) and generates a START condition after a delay of a half clock period of the internal serial clock generator.

If STA is set while the I^{2}C interface is already in a master mode and one or more bytes are transmitted or received, the I^{2}C interface transmits a repeated START condition. STA may be set at any time. STA may also be set when the I^{2}C interface is an addressed slave.

When the STA bit is reset, no START condition or repeated START condition is generated.

7.12.2.3 STO - the STOP flag

When the STO bit is set while the I^{2}C interface is in a master mode, a STOP condition is transmitted to the I^{2}C bus. When the STOP condition is detected on the bus, the I^{2}C hardware clears the STO flag. In a slave mode, the STO flag may be set to recover from an error condition. In this case, no STOP condition is transmitted to the I^{2}C bus. However, the SIO1 hardware behaves as if a STOP condition has been received and switches to the defined “not addressed” slave receiver mode. The STO flag is automatically cleared by the hardware.

If the STA and STO bits are both set, the STOP condition is transmitted to the I^{2}C bus if the I^{2}C interface is in a master mode (in a slave mode, the I^{2}C interface generates an internal STOP condition which is not transmitted). the I^{2}C interface then transmits a START condition.

When the STO bit is reset, no STOP condition is generated.

7.12.2.4 SI - the serial interrupt flag

When the SI flag is set, then, if the I^{2}C interrupt is enabled, the CPU receives an interrupt. SI is set by hardware when any one of 25 of the possible 26 I^{2}C states are entered.The only state that does not cause SI to be set is state F8H, which indicates that no relevant state information is available.

The I2C interface status states are described in detail in Section 7.12.4.

While SI is set, the low period of the serial clock on the SCL line is stretched, and the serial transfer is suspended. A high level on the SCL line is unaffected by the serial interrupt flag. SI should be reset by software.

When the SI flag is reset, no serial interrupt is requested, and there is no stretching of the serial clock on the SCL line.

7.12.2.5 AA - the assert acknowledge flag

If the AA flag is set, an acknowledge (low level to SDA) is returned during the acknowledge clock pulse on the SCL line when:

- The “own slave address” has been received.
- The general call address has been received while the general call bit (GC) in I2C0ADR is set.
- A data byte has been received while SIO1 is in the master receiver mode.
- A data byte has been received while SIO1 is in the addressed slave receiver mode.

When SIO1 is in the addressed slave transmitter mode, state C8H is entered after the last serial bit is transmitted. When SI is cleared, the I2C interface leaves state C8H, enters the not addressed slave receiver mode, and the SDA line remains at a high level. In state C8H, the AA flag can be set again for future address recognition.

When the I2C interface is in the not addressed slave mode, its own slave address and the general call address are ignored. Consequently, no acknowledge is returned, and a serial interrupt is not requested. Thus, the I2C interface can be temporarily released from the I^2C bus while the bus status is monitored. While the I2C interface is released from the bus, START and STOP conditions are detected, and serial data is shifted in. Address recognition can be resumed at any time by setting the AA flag. If the AA flag is set when the part’s own slave address or the general call address has been partly received, the address is recognized at the end of the byte transmission.

7.12.2.6 CR2, 1, 0 - the clock rate bits

These three bits determine the serial clock frequency when the I2C interface is in a master mode. The available frequencies are given in Table 398.

Table 398: I2C clock frequency settings

CR2:0	Division ratio	I^2C bit frequency (kHz) for 16MHz system clock
000	10	1600
001	20	800
010	30	533
011	40	400
100	80	200
101	120	132
110	160	100
111	T1[1]	-

[1] When CR2:0 = 111, the I2C bit rate is 1/2 the Timer01 overflow rate.

7.12.3 I2C0DAT

I2C0DAT contains a byte of serial data to be transmitted or a byte which has just been received. The CPU can read from and write to this 8-bit SFR while it is not in the process of shifting a byte. This occurs when SIO1 is in a defined state and the serial interrupt flag is set. Data in I2C0DAT remains stable as long as SI is set. Data in I2C0DAT is always shifted from right to left: the first bit to be transmitted is the MSB (bit 7), and, after a byte has been received, the first bit of received data is located at the MSB of I2C0DAT. While data is being shifted out, data on the bus is simultaneously being shifted in; I2C0DAT always contains the last data byte present on the bus. Thus, in the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data in I2C0DAT. Reset initializes I2C0DAT to 00H.

7	6	5	4	3	2	1	0
SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0

Fig 269.I2C0DAT

7.12.3.1 SD7 - SD0 - the data bits

Eight bits to be transmitted or just received. A logic ‘1’ in I2C0DAT corresponds to a high level on the I^{2}C bus, and a logic ‘0’ corresponds to a low level on the bus. Serial data shifts through I2C0DAT from right to left.

I2C0DAT and the ACK flag form a 9-bit shift register which shifts in or shifts out an 8-bit byte, followed by an acknowledge bit. The ACK flag is controlled the I^{2}C interface and cannot be accessed by the CPU. Serial data is shifted through the ACK flag into I2C0DAT on the rising edges of clock pulses on the SCL line. When a byte has been shifted into I2C0DAT, the serial data is available in I2C0DAT, and the acknowledge bit is returned by the control logic during the ninth clock pulse. Serial data is shifted out from I2C0DAT via a buffer on the falling edges of clock pulses on the SCL line.

When the CPU writes to I2C0DAT, the buffer is loaded with the contents of I2C0DAT(7) which is the first bit to be transmitted to the SDA line. After nine serial clock pulses, the eight bits in I2C0DAT are transmitted to the SDA line, and the acknowledge bit is present in ACK. Note that the eight transmitted bits are shifted back into I2C0DAT.

7.12.4 I2C0STA

I2C0STA is an 8-bit read-only special function register. The three least significant bits are always zero. The five most significant bits contain the status code. There are 26 possible status codes. When I2C0STA contains F8H, no relevant state information is available and no serial interrupt is requested. Reset initializes I2C0STA to F8H. All other I2C0STA values correspond to defined I^{2}C states. When each of these states is entered, a serial interrupt is requested (SI = ‘1’), this can happen in any CPU cycle, and a valid status code is present in I2C0STA. This status code remains present in I2C0STA until SI is cleared by software.

Remark: I2C0STA changes one clock cycle after SI changes, so the new status can be visible in the same machine cycle SI changes or possibly (in one out of six cases) the machine cycle after that. This should not be a problem since you should not read I2C0STA before either polling SI or entry of the interrupt handler (which in itself takes several machine cycles).

7	6	5	4	3	2	1	0
ST7	ST6	ST5	ST4	ST3	0	0	0

Fig 270.I2C0STA

7.12.4.1 ST7 - ST3 - the status bits

Table 399 lists all possible status codes returned in a certain mode (master transmitter, master receiver, slave transmitter, slave receiver) plus some miscellaneous status codes that can be returned at any time.

Table 399: Master transmitter mode status codes

Status code I2C0STA	Status of the I^2C and I^2C interface	Application software response					Next action taken by the I^2C interface
		To /fromI2C0DAT	To I2C0CON				
			STA	STO	SI	AA	
08H	A START condition has been transmitted	Load SLA+W	X	0	0	X	SLA+W will be transmitted ACK will be received.
10H	A repeated START condition has been transmitted	Load SLA+W or	X	0	0	X	As above
		Load SL+R	X	0	0	X	SLA+W will be transmitted; SIO1 will be switched to MST/(TRX or REC) mode.
18H	SLA+W has been transmitted; ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received.
		no I2C0DAT action or	1	0	0	X	Repeated START will be transmitted;
		no I2C0DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
		no I2C0DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
20H	SLA+W has been transmitted; NOT ACK has been received	Load data byte or	0	0	0	X	Data bye will be transmitted; ACK bit will be received.
		no I2C0DAT action or	1	0	0	X	Repeated START will be transmitted.
		no I2C0DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
		no I2C0DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
28H	Write data byte in I2C0DAT has been transmitted; ACK has been received	Load data byte or	0	0	0	X	Data bye will be transmitted; ACK bit will be received.
		no I2C0DAT action or	1	0	0	X	Repeated START will be transmitted.
		no I2C0DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
		no I2C0DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.

Table 399: Master transmitter mode status codes *…continued*

Status code I2C0STA	Status of the I^{2}C and I^{2}C interface	Application software response					Next action taken by the I^{2}C interface
		To /fromI2C0DAT	To I2C0CON				
			STA	STO	SI	AA	
30H	Write data byte in I2C0DAT has been transmitted; NOT ACK has been received	Load data byte or	0	0	0	X	Data bye will be transmitted; ACK bit will be received.
		no I2C0DAT action or	1	0	0	X	Repeated START will be transmitted.
		no I2C0DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
		no I2C0DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
38H	Arbitration lost in SLA+R/W or Data bytes	no I2C0DAT action or	0	0	0	X	1^{2}C will be released; a slave mode will be entered.
		no I2C0DAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free.

Table 400: Master Receiver Mode status codes

Status code I2C0STA	Status of the I^{2}C and I^{2}C interface	Application software response					Next action taken by the I^{2}C interface
		To /fromI2C0DAT	To I2C0CON				
			STA	STO	SI	AA	
08H	A START condition has been transmitted	Load SLA+W	X	0	0	X	SLA+W will be transmitted ACK will be received.
10H	A repeated START condition has been transmitted	Load SLA+W or	X	0	0	X	As above.
		Load SL+R	X	0	0	X	SLA+W will be transmitted; SIO1 will be switched to MST/(TRX or REC) mode.
38H	Arbitration lost in SLA+R/W or Data bytes	no I2C0DAT action or	0	0	0	X	1^{2}C will be released; a slave mode will be entered.
		no I2C0DAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free.
40H	SLA+R has been transmitted; ACK has been received	no I2C0DAT action or	0	0	0	0	Data byte will be received; NOT ACK bit will be returned.
		no I2C0DAT action	0	0	0	1	Data byte will be received; ACK bit will be returned.
48H	SLA+R has been transmitted; NOT ACK has been received	no I2C0DAT action or	1	0	0	X	Repeated START condition will be transmitted.
		no I2C0DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
		no I2C0DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
50H	Read data byte has been received; ACK has been returned	Read data byte or	0	0	0	0	Data byte will be received; NOT ACK bit will be returned.
		read data byte	0	0	0	1	Data byte will be received; ACK bit will be returned.

Table 400: Master Receiver Mode status codes *…continued*

Status	Status of the	Application software response					Next action taken by the I^{2}C interface
code	I^{2}C and I^{2}C interface	To /fromI2C0DAT	To I2C0CON				
I2C0STA			STA	STO	SI	AA	
58H	Read data byte has been received; NOT ACK has been returned	Read data byte or	1	0	0	X	Repeated START condition will be transmitted.
		Read data byte or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
		Read data byte	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.

Table 401: Slave Receiver Mode status codes

Status	Status of the	Application software response					Next action taken by the I^{2}C interface
code	I^{2}C and I^{2}C interface	To /fromI2C0DAT	To I2C0CON				
I2C0STA			STA	STO	SI	AA	
60H	Own SLA+W has been received; ACK has been returned.	no I2C0DAT action or	X	0	0	0	Data byte will be received an NOT ACK will be returned.
		no I2C0DAT action	X	0	0	1	Data bye will be received and ACK will be returned.
68H	Arbitration lost in SLA+R/W as master; Own SLA+W has been received, ACK returned.	no I2C0DAT action or	X	0	0	0	Data byte will be received an NOT ACK will be returned.
		no I2C0DAT action	X	0	0	1	Data bye will be received and ACK will be returned.
70H	General call address (00H) has been received; ACK has been returned.	no I2C0DAT action or	X	0	0	0	Data byte will be received an NOT ACK will be returned.
		no I2C0DAT action	X	0	0	1	Data bye will be received and ACK will be returned.
78H	Arbitration lost in SLA+R/W as master; General call address has been received, ACK has been returned.	no I2C0DAT action or	X	0	0	0	Data byte will be received an NOT ACK will be returned.
		no I2C0DAT action	X	0	0	1	Data bye will be received and ACK will be returned.
80H	Previously addressed with own SLA; Write data byte has been received; ACK has been returned.	Read data byte or	X	0	0	0	Data byte will be received an NOT ACK will be returned.
		Read data byte	X	0	0	1	Data byte will be received and ACK will be returned.

Table 401: Slave Receiver Mode status codes *…continued*

Status code I2C0STA	Status of the I^{2}C and I^{2}C interface	Application software response					Next action taken by the I^{2}C interface
		To /fromI2C0DAT	To I2C0CON				
			STA	STO	SI	AA	
88H	Previously addressed with own SLA; Write data byte has been received; NOT ACK has been returned.	Read data byte or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address.
		Read data byte	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if I2C0ADR(0) = ‘1’.
		Read data byte or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free Switched to not addressed SLV mode.
		Read data byte	1	0	0	1	Own SLA will be recognized; General call address will be recognized if I2C0ADR(0) = ‘1’. A START condition will be transmitted when the bus becomes free.
90H	Previously addressed with General Call; Write data byte has been received; ACK has been returned.	Read data byte or	X	0	0	0	Data byte will be received and NOT ACK will be returned.
		Read data byte	X	0	0	1	Data byte will be received and ACK will be returned.
98H	Previously addressed with General Call; Write data byte has been received; NOT ACK has been returned.	Read data byte or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address.
		Read data byte	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if I2C0ADR(0) = ‘1’.
		Read data byte or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free Switched to not addressed SLV mode; Own SLA will be recognized.
		Read data byte	1	0	0	1	General call address will be recognized if I2C0ADR(0) = ‘1’. A START condition will be transmitted when the bus becomes free.

Table 401: Slave Receiver Mode status codes *…continued*

Status code I2C0STA	Status of the I^{2}C and I^{2}C interface	Application software response					Next action taken by the I^{2}C interface
		To /fromI2C0DAT	To I2C0CON				
			STA	STO	SI	AA	
AOH	A STOP condition or repeated START condition has been received while still addressed as SLV/(REC or TRX).	Read data byte or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address.
		Read data byte	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if I2C0ADR(0) = ‘1’.
		Read data byte or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free Switched to not addressed SLV mode.
		Read data byte	1	0	0	1	Own SLA will be recognized; General call address will be recognized if I2C0ADR(0) = ‘1’. A START condition will be transmitted when the bus becomes free.

Table 402: Slave Transmitter Mode status codes

Status code I2C0STA	Status of the I^{2}C and I^{2}C interface	Application software response					Next action taken by the I^{2}C interface
		To /fromI2C0DAT	To I2C0CON				
			STA	STO	SI	AA	
A8H	Own SLA+R has been received; ACK has been returned.	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received.
		Load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received.
BOH	Arbitration lost in SLA+R/W as master; Own SLA+R has been received. ACK has been returned.	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received.
		Load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received.
B8H	Read data bye in I2C0DAT has been transmitted; ACK has been received.	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received.
		Load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received.

Table 402: Slave Transmitter Mode status codes *…continued*

Status code I2C0STA	Status of the I^{2}C and I^{2}C interface	Application software response					Next action taken by the I^{2}C interface
		To /fromI2C0DAT	To I2C0CON				
			STA	STO	SI	AA	
COH	Read data byte in I2C0DAT has been transmitted; NOT ACK has been received.	no I2C0DAT action or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address.
		no I2C0DAT action or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if I2C0ADR(0) = ‘1’.
		no I2C0DAT action or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free Switched to not addressed SLV mode.
		no I2C0DAT action	1	0	0	1	Own SLA will be recognized; General call address will be recognized if I2C0ADR(0) = ‘1’. A START condition will be transmitted when the bus becomes free.
C8H	Last read data byte in I2C0DAT has been transmitted (AA=0); ACK has been received.	no I2C0DAT action or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address.
		no I2C0DAT action or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if I2C0ADR(0) = ‘1’.
		no I2C0DAT action or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free Switched to not addressed SLV mode.
		no I2C0DAT action	1	0	0	1	Own SLA will be recognized; General call address will be recognized if I2C0ADR(0) = ‘1’. A START condition will be transmitted when the bus becomes free.

Table 403: Miscellaneous status codes

Status code I2C0STA	Status of the I^{2}C and I^{2}C interface	Application software response					Next action taken by the I^{2}C interface
		To /fromI2C0DAT	To I2C0CON				
			STA	STO	SI	AA	
OOH	Bus error.	No I2C0DAT action	X	1	0	X	Hardware will enter the “not addressed” slave mode.
F8H	No information available.	No I2C0DAT action	-	-	-	-	-

7.12.5 I2C0ADR

The CPU can read from and write to this 8-bit SFR. I2C0ADR is not affected by the I^{2}C interface hardware. The contents of this register are irrelevant when the I^{2}C interface is in a master mode. In the slave modes, the seven most significant bits are loaded with the TV Microcontroller's own slave address, and, if the least significant bit is set, the general call address (00H) is recognized; otherwise it is ignored. Reset initializes I2C0ADR to 00H.

7	6	5	4	3	2	1	0
SA6	SA5	SA4	SA3	SA2	SA1	SA0	GC

Fig 271.I2C0ADR

7.12.5.1 SA6 - SA0 - the slave address bits

These bits correspond to the 7-bit slave address which is recognized on the incoming data stream from the I^{2}C bus. When the slave address is detected and the interface is enabled, a serial interrupt is generated to the CPU.

7.12.5.2 GC - the general call bit

When set, this bit causes the I^{2}C logic to watch for the general call address to be transmitted on the I^{2}C bus. If a general call address is detected and this bit is set, a serial interrupt is generated to the CPU.

7.12.6 Modes of operation

The I^{2}C logic may operate in any of the following four modes:

- Master Transmitter
- Master Receiver
- Slave Receiver
- Slave Transmitter

As a master, the I^{2}C logic will generate all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the I^{2}C bus will not be released.

Two types of data transfers are possible on the I^{2}C bus:

- Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte.
- Data transfer from a slave transmitter to a master receiver. The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next follows the data bytes transmitted by the slave to the master. The master returns an acknowledge bit after each received byte except the last byte. At the end of the last received byte, a "not acknowledge" is returned.

In a given application, the I^{2}C interface may operate as a master and as a slave. In the slave mode, the I^{2}C interface looks for its own slave address and the general call address. If one of these addresses is detected, an interrupt is requested. When the TV

Microcontroller wishes to become the bus master, the hardware waits until the bus is free before the master mode is entered so that a possible slave action is not interrupted. If bus arbitration is lost in the master mode, the I^{2}C interface switches to the slave mode immediately and can detect its own slave address in the same serial transfer.

7.12.6.1 Master transmitter mode

Serial data is output through SDA while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7-bit SLA) and the data direction bit. In this case the data direction bit (R/W) is logic ‘0’ (W). Serial data is transmitted 8 bits at a time. After each byte is transmitted, an acknowledge bit is received. START and STOP conditions are output to indicate the beginning and the end of a serial transfer.

In the master transmitter mode, a number of data bytes can be transmitted to the slave receiver. Before the master transmitter mode can be entered, I2C0CON is initialized with the ENS1 bit set and the STA, STO and SI bits reset. ENS1 is set to enable the I^{2}C interface. If the AA bit is reset, the I^{2}C interface will not acknowledge its own slave address or the general call address if they are present on the bus. This will prevent the I^{2}C interface from entering a slave mode.

The master transmitter mode may now be entered by setting the STA bit. The I^{2}C interface will then test the I^{2}C bus and generate a start condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SI) is set, and the status code in the status register (I2C0STA) will become 08H. This status code should be used to vector to an interrupt service routine that loads I2C0DAT with the slave address and the data direction bit (SLA+W). The SI bit in I2C0CON is then reset before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in I2C0STA are possible. The appropriate action to be taken for any of the status codes is detailed in the table. After a repeated start condition (state 10H), the I^{2}C interface may switch to the master receiver mode by loading I2C0DAT with SLA+R.

7.12.6.2 Master receiver mode

The first byte transmitted contains the slave address of the transmitting device (7-bit SLA) and the data direction bit. In this case the data direction bit (R/W) is logic 1 (R). Serial data is received via SDA while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are output to indicate the beginning and end of a serial transfer.

In the master receiver mode, a number of data bytes are received from a slave transmitter. The transfer is initialized as in the master transmitter mode. When the START condition has been transmitted, the interrupt service routine should load I2C0DAT with the 7-bit slave address and the data direction bit (SLA+R). The SI bit in I2C0CON should then be cleared before the serial transfer can continue.

When the slave address and the data direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes are possible in I2C0STA. The appropriate action to be taken for each of the status codes is detailed in the table. After a repeated start condition (state 10H), the I^{2}C interface may switch to the master transmitter mode by loading I2C0DAT with SLA+W.

7.12.6.3 Slave receiver mode

Serial data and the serial clock are received through SDA and SCL. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit.

In the slave receiver mode, a number of data bytes are received from a master transmitter. To initiate the slave receiver mode, I2C0ADR should be loaded with the 7-bit slave address to which the I^{2}C interface will respond when addressed by a master. Also the least significant bit of I2C0ADR should be set if the interface should respond to the general call address (00H). The control register, I2C0CON, is initialized with ENS1 and AA set and STA, STO, and SI reset in order to enter the slave receiver mode. Setting the AA bit will enable the logic to acknowledge its own slave address or the general call address and ENS1 will enable the interface.

When I2C0ADR and I2C0CON have been initialized, the I^{2}C interface waits until it is addressed by its own slave address followed by the data direction bit which should be ‘0’ (W) for the I^{2}C interface to operate in the slave receiver mode. After its own slave address and the W bit have been received, the serial interrupt flag (SI) is set and a valid status code can be read from I2C0DAT. This status code should be used to vector to an interrupt service routine, and the appropriate action to be taken for each of the status codes is detailed in the table. The slave receiver mode may also be entered if arbitration is lost while the I^{2}C interface is in the master mode.

If the AA bit is reset during a transfer, the I^{2}C interface will return a not acknowledge (logic ‘1’) to SDA after the next received data byte. While AA is reset, the I^{2}C interface does not respond to its own slave address or a general call address. However, the I^{2}C bus is still monitored and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate the I^{2}C interface from the I^{2}C bus.

7.12.6.4 Slave transmitter mode

The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. Serial data is transmitted via SDA while the serial clock is input through SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer.

In the slave transmitter mode, a number of data bytes are transmitted to a master receiver. Data transfer is initialized as in the slave receiver mode. When I2C0ADR and I2C0CON have been initialized, the I^{2}C interface waits until it is addressed by its own slave address followed by the data direction bit which should be ‘1’ (R) for SIO1 to operate in the slave transmitter mode. After its own slave address and the R bit have been received, the serial interrupt flag (SI) is set and a valid status code can be read from I2C0STA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status codes is detailed in the table. The slave transmitter mode may also be entered if arbitration is lost while the I^{2}C interface is in the master mode.

If the AA bit is reset during a transfer, the I^{2}C interface will transmit the last byte of the transfer and enter state C0H or C8H. The I^{2}C interface is switched to the not addressed slave mode and will ignore the master receiver if it continues the transfer. Thus the master receiver receives all ‘1’s as serial data. While AA is reset, the I^{2}C interface does not respond to its own slave address or a general call address. However, the I^{2}C bus is still monitored, and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate the I^{2}C interface from the I^{2}C bus.

7.13 UART (Universal Asynchronous Receiver Transmitter)

7.13.1 UART description

The UART is a full duplex serial port, which means it can transmit and receive simultaneously. The serial port is receive-buffered: it can commence reception of a second byte before a previously received byte has been read from the receive buffer. However, if the first byte has not been read by the time reception of the second byte is complete, one of the bytes is lost.

The serial port receive and transmit data registers are both accessed by software at the address of Special Function Register S0BUF. Writing to S0BUF loads the transmit register, reading from S0BUF accesses a physically separate receive register.

The UART also supports framing error detection.

7.13.1.1 Modes

The serial port can operate in 4 modes. These modes are selected by programming special function register S0CON bits SM0 and SM1:

Mode 0: Data is transmitted and received through pin P3.0 (RXD). The shift clock is transmitted on pin P3.1 (TXD). 8 bits are transmitted/received (LSB first).

- Baud rate: fixed at 1/6 of the frequency of the system clock.

Mode 1: 10 bits are transmitted through P3.1 (TXD) or received through P3.0 (RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1).

- Receive: the received stop bit is stored into special function register bit S0CON.RB8.
- Baud rate: variable, set by the overflow rate of either Timer 1 or Timer 2.

Mode 2: 11 bits are transmitted through P3.1 (TXD) or received through P3.0 (RXD): start bit (0), 8 data bits (LSB first), a 9th data bit, and a stop bit (1).

- Transmit: the 9th data bit is taken from S0CON.TB8, it can be assigned the value of 0 or 1. For example, the parity bit could be loaded into S0CON.TB8.
- Receive: the 9th data bit is stored into S0CON.RB8, while the stop bit is ignored.
- Baud rate: programmable to either 1/16 or 1/32 the frequency of the system clock.

Mode 3: 11 bits are transmitted through P3.1 (TXD) or received through P3.0 (RXD): a start bit (0), 8 data bits (LSB first), a 9th data bit, and a stop bit (1).In fact, mode 3 is the same as mode 2 in all aspects except the baud rate.

- Transmit: as mode 2, the 9th data bit is taken from S0CON.TB8.

- Receive: as mode 2, the 9th data bit is stored into S0CON.RB8.
- Baud rate: variable, set by the overflow rate of either Timer 1 or Timer 2.

The UART initiates transmission and/or reception as follows:

- Transmission is initiated, in modes 0, 1, 2, 3, by any instruction that uses S0BUF as destination.
- Reception initiated in mode 0, if S0CON.RI = 0 and S0CON.REN = 1.
- Reception initiated in modes 1, 2, 3 by the incoming start bit if S0CON.REN = 1.

7.13.1.2 Multiprocessor communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, the 10th bit is the stop bit. Setting S0CON.SM2 enables that the UART interrupt is activated only if S0CON.RB8 = 1 when the stop bit is received. A way to use this feature in multiprocessor systems is as follows:

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte: the 9th bit is 1 in an address byte and 0 in a data byte. With S0CON.SM2 = 1, no slave is interrupted by a data byte reception. An address byte, however, interrupts all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its S0CON.SM2 bit and prepare to receive the data bytes that will follow. The slaves that weren’t being addressed leave their S0CON.SM2 bits set and ignore the incoming data bytes.

Bit S0CON.SM2 has no effect in mode 0. In mode 1 it can be used to check the validity of the stop bit. When receiving in mode 1, if S0CON.SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

7.13.1.3 Framing error detection

A framing error occurs whenever a stop bit is missing in the communication. This is when bit 8 is 0 in mode 1 or bit 9 is 0 in mode 2 or 3. Under these conditions the S0FECON.FE bit is set when enabled during reception. Even if a received character causes a framing error, it can still set S0CON.RI.

7.13.2 Programmer’s view

7.13.2.1 UART registers

The UART contains three Special Function Registers (SFRs) that can be read and written by the CPU.

7.13.2.2 S0CON register

The Special Function Register S0CON is the control and status register of the UART. This register contains the mode selection bits (SM2, SM1, SM0), the 9th data bit for transmit and receive (TB8 and RB8), and the UART interrupt bits (TI and RI).

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Fig 272.S0CON register

Table 404: S0CON register description

Symbol	Position	Name	Description
SM0	S0CON.7	Mode selection bit 0	Set/cleared by software only. The UART has 4 modes:
SM1	S0CON.6	Mode selection bit 1	Mode 0: SM0=0 SM1=0 Shift register Baud rate fclk/6 Mode 1: SM0=0 SM1=1 8 bit UART Variable baud rate Mode 2: SM0=1 SM1=0 9 bit UART Baud rate fclk/64 or fclk/32 Mode 3: SM0=1 SM1=1 9 bit UART Variable baud rate
SM2	S0CON.5	Multi-processor communication enable	Enables the multi-processor communication feature in modes 2 and 3. Set/cleared by software only. In modes 2 or 3, if SM2 is set, then RI will not be activated and RB8 and S0BUF will not be loaded if the received 9th data bit is 0, if SM2 is cleared it has no influence n the activation of RI and RB8. In mode 1, if SM2 is set, then RI will not be activated and RB8 and S0BUF will not be loaded if no valid stop bit was received, if SM2 is cleared it has no influence on the activation of RI and RB8. In mode 0, SM2 has no influence.
REN	S0CON.4	Serial reception enable	Set/cleared by software only. Enables serial reception. Set by software enables reception. Cleared by software disables reception.
TB8	S0CON.3	Transmit data bit	Set/cleared by software only. The value of TB8 is transmitted as the 9th data bit in modes 2 and 3. The TB8 bit is not used in modes 0 and 1. Set or cleared by software as desired.
RB8	S0CON.2	Receive data bit	Set/cleared by hardware and by software (if SM2=1 loading RB8 can be blocked, see bit description of SM2 above). In modes 2 or 3, the hardware stores the 9th data bit that was received in RB8. In mode 1, the hardware stores the stop bit that was received in RB8. In mode 0, the hardware does not change RB8.
TI	S0CON.1	Transmit interrupt flag	In modes 2 or 3, when transmitting the hardware sets the transmit interrupt flag TI at the end of the 9th bit time. In modes 0 or 1, when transmitting the hardware sets the transmit interrupt flag TI at the end of the 8th bit time. TI should be cleared by software.
RI	S0CON.0	Receive interrupt flag	In modes 2 or 3, when receiving, the hardware sets the receive interrupt flag 1 clock period after sampling the 9th data bit (if SM2=1 setting RI can be blocked, see bit description of SM2 above). In mode 1, when receiving, the hardware sets the receive interrupt flag 1 clock period after sampling the stop bit (if SM2=1 setting RI can be blocked, see bit description of SM2 above). In mode 0, when receiving the hardware sets RI at the end of cpu state 1 of the 9th machine cycle after the machine cycle where data reception started by a write to S0CON. RI should be cleared by software.

The S0CON register supports a locking mechanism to prevent software read-modify-write instructions to overwrite the contents while hardware is modifying the contents of the register.

7.13.2.3 S0BUF registers

Writing to S0BUF writes to the transmit buffer. Reading from S0BUF reads from the receive buffer. Only hardware can read from the transmit buffer and write to the receive buffer.

7.13.2.4 S0FECON register

The framing error control register S0FECON controls the framing error detection. It also contains the double baud rate bit SMOD.

-	-	-	-	-	SMO	FE_E	FE

Fig 273.Framing Error Control (S0FECON) register

Table 405: S0FECON Register description

Symbol	Position	Name	Description
SMOD	S0FECON.2	Double baud rate bit	Set/cleared by software only. If set the baud rate in mode 1, 2 and 3 is doubled. In mode 0 SMOD is not used.
FE_EN	S0FECON.1	Framing error enable bit	Set/cleared by software only. If set the framing error detection is enabled. If reset the framing error detection is disabled.
FE	S0FECON.0	Framing error bit	Set by hardware halfway through the stop bit time in mode 1, 2 and 3 if the received stop bit is zero and the received character did not contain all zeros. In mode 0 this bit is not used. The FE bit supports a locking mechanism to prevent software read-modify-write instructions from overwriting the contents while hardware is modifying the contents of this register.

7.13.3 Baud rates

The serial port can operate with different baud rates depending on its mode:

Mode 0: the baud rate is fixed at f_{clk}/ 6.

Mode 2: the baud rate in mode 2 depends on the value of bit SMOD.

If SMOD = 0, the baud rate is $f_{clk}/32$

If SMOD = 1, the baud rate is $f_{clk}/16$

$$\text{Baud rate} = \frac{2^{SMOD}}{32} \times f_{clk}$$

Mode 1 and 3: The baud rates are determined by the overflow rate of either Timer 1 or Timer 2.

The register bit T2CON.RCLK0 selects between Timer 1 and Timer 2 for receive baud rate generation, and the register bit T2CON.TCLK0 selects between Timer 1 and Timer 2 for transmit baud rate generation (see Section 7.8 “Timer 2”). The following tables give the available baud rate options.

Table 406: Mode 1, 3 TX baud rate

TCLK0	SMOD	baud
0	0	1/32 * Timer1 overflow
0	1	1/16 * Timer1 overflow
1	-	1/16 * Timer2 overflow

Table 407: Mode 1, 3 RX baud rate

RCLK0	SMOD	baud
0	0	1/32 * Timer1 overflow
0	1	1/16 * Timer1 overflow
1	-	1/16 * Timer2 overflow

7.13.3.1 Baud rates using Timer 1 (UART mode 1 and 3)

The overflow output of Timer1 can be used to generate the UART transmit or receive baud rate. The UART is set in mode 1 or 3 and the T2CON.TCLK0 bit is 0 to select Timer 1 for transmitting and the T2CON.RCLK0 bit is 0 to select Timer 1 for receiving. The T1 interrupt is disabled in this application. The Timer 1 itself can be configured for either ’timer’ or ’counter’ operation, and in any of its 3 running modes. In the most typical applications, it is configured for ’timer’ operation, in the auto-reload mode (Timer 1 mode 2: high nibble of TMOD = 0010b). In that case the baud rate is given by the formula:

Baud Rate =

$$\frac{2^{SMOD}}{32} \times \frac{f_{clk}}{6 \times (256 - T1H)}$$

When rewriting this formula, the value for the Timer 1 reload value T1H is calculated from the desired baud rate as follows:

Timer 1 reload value T1H =

$$256 - \frac{2^{SMOD} \times f_{clk}}{32 \times 6 \times Baudrate}$$

Very low baud rates can be achieved with Timer 1 by leaving the T1 interrupt enabled, and configuring the timer to run as a 16-bit timer (high nibble of TMOD = 0001b), and using the T1 interrupt to do a 16-bit software reload.

For details on programming Timer 1 to function as baud rate generator for the UART see Section 7.7 “Timer 0/1”.

7.13.3.2 Baud rates using Timer 2 (UART mode 1 and 3)

The overflow output of Timer 2 can be used to generate the UART transmit or receive baud rate. The UART is set in mode 1 or 3 and the T2CON.TCLK0 bit is 1 to select Timer 2 for transmitting and the T2CON.RCLK0 bit is 1 to select Timer 2 for receiving. Timer 2 has a programming mode to function as baud rate generator for the UART. In this mode the baud rate is given by formula:

Baud Rate =

$$\frac{f_{clk}}{16 \times [65536 - (T2RCH,T2RCL)]}$$

When rewriting this formula, the value for the Timer 2 reload values T2RCH/L is calculated from the desired baud rate as follows:

Reload value T2RCH/L =

$$65536 - \frac{f_{clk}}{16 \times Baudrate}$$

For details on programming Timer 2 to function as baud rate generator for the UART see Section 7.8 “Timer 2”.

7.14 SPI (Serial Peripheral Interface)

7.14.1 General Description

The SPI interface is compliant with the Motorola™ Serial Peripheral Interface (SPI) specification. It provides SPI master operation, with synchronous, full-duplex serial communication. The maximum data bit rate is one half the system clock frequency. The SPI interface does not generate the SS (Slave Select) signal required by the SPI specification; this should be handled by software using a separate general-purpose I/O signal.

The SPI interface is primarily intended for interfacing to a serial flash memory (e.g. ST MP2505-A). This is used by the bootloader to preload the TV Microcontroller application software into the CODE memory at system boot (for more information about the boot process, see section 16 - Boot Process and Mode Selection). The bootloader uses the P6.3 pin to implement the SS signal to the flash memory.

A data transfer is initiated by writing a byte of data to the SPIDAT SFR. The SPI interface simultaneously transmits the data byte while receiving another data byte from the SPI bus. The SPIF bit in the SPISTAT SFR is used as a handshaking mechanism to ensure the previous transfer has completed before a new transfer is initiated.

The SPI interface also permits burst mode operation, which minimizes the software overhead for each byte of data transferred, allowing the aggregate data transfer rate to approach the theoretical maximum line rate of the SPI bus. In burst mode, the SPIF handshaking mechanism is disabled, and each access (read or write) to the SPIDAT SFR initiates a one-byte transfer on the SPI bus. As a result, burst transfers are transmit-only or receive-only; duplex operation is not supported. When doing burst transmit operations the received data is discarded, and when doing burst receive operations, continuous zero data is transmitted. In burst mode, software should allow enough delay for one transfer to complete before initiating a second transfer.

7.14.2 Special Function Registers

Table 408: SPICON - SPI control register

Symbol	Position	Access	Value	Description
RSRV	7	R/O	0b0	Reserved
LSBF	6	R/W	0b0	LSB - first enable. This bit controls which direction each byte is shifted when transferred. 1- SPI data is transferred LSB (bit 0) first 0- SPI data is transferred MSB (bit 7) first
RSRV	5	R/O	0b0	Reserved
CPOL	4	R/W	0b0	Clock polarity. This bit controls the polarity of the SPI_SCK clock. 1 - SPI_SCK is active low 0 - SPI_SCK is active high
CPHA	3	R/W	0b0	Clock phase. This bit controls the relationship between the data and the clock on SPI transfers. 0 - Data is always sampled on the first clock edge of SPI_SCK. A transfer starts and ends with activation and deactivation of the SPI_SS signal 1 - Data is always sampled on the second clock edge of the SPI_SCK A transfer starts and ends with the first clock edge, and last sampling clock edge when the SPI_SS signal is active.
BURST	2	R/W	0b0	Burst mode enable. This bit enables burst mode operation. In burst mode, the SPIF handshaking mechanism is disabled, and any access (read or write) to the SPIDAT register initiates a transfer on the SPI bus. 1 - Burst mode is enabled 0 - Burst mode is disabled
RSRV	1	R/O	0b0	Reserved
FPEN	0	R/W	0b0	Fast path enable. Enable higher SPI clock frequencies (controlled by the SPICLK SFR). This bit should be set to '1' 1 - Maximum SPI clock frequency is 1/2 of the system clock frequency 0 - Maximum SPI clock frequency is 1/8 of the system clock frequency

Table 409: SPISTAT - SPI status register

Symbol	Position	Access	Value	Description
SPIF	7	R/O	0b0	SPI transfer complete flag. This bit indicates when a SPI data transfer is complete. This bit is set at the end of the last cycle of the transfer. This bit is cleared by first reading this register, then accessing the SPI data register. 1 - Data transfer has complete 0 - No data transfer completion indicated
WCOL	6	R/O	0b0	Write collision. This bit indicates that a write collision has occurred. There is no write buffer between the SPI block bus interface, and the internal shift register. As a result, data should not be written to the SPI data register when a SPI data transfer is currently in progress. The time frame where data cannot be written to the SPI data register is from when the transfer starts, until after the status register has been read when the SPIF status is active. If the SPI data register is written in this time frame, the write data will be lost, and the write collision (WCOL) bit in the status register will be activated. This bit is cleared by reading this register, then accessing the SPI data register. 1 - Write collision 0 - No write collision
ROVR	5	R/O	0b0	Read overrun. This bit indicates that a read overrun has occurred. A read overrun occurs when the SPI block internal read buffer contains data that has not been read by the processor, and a new transfer has completed. The read buffer containing valid data is indicated by the SPIF bit in the status register being active. When a transfer completes, the SPI block needs to move the received data to the read buffer. If the SPIF bit is active (the read buffer is full), the new receive data will be lost, and the ROVR bit will be activated. This bit is cleared by reading this register. 1 - Read overrun 0 - No read overrun
RSRV	4:0	R/O	0b0	Reserved

Remark: The contents of the SPISTAT register are not relevant during burst mode operation (when the BURST bit in the SPICON SFR is set).

Table 410: SPDISTAT - SPI Bi-directional status register

Symbol	Position	Access	Value	Description
SPIBI	7:0	R/W	0x00	SPI Bi-directional data port. The SPI data register is used to provide the transmit and receive data bytes. An internal shift register in the SPI block logic is used for the actual transmission and reception of the serial data. Data is written to the SPI data register for the transmit case. There is no buffer between the data register and the internal shift register. A write to the data register goes directly into the internal shift register. Therefore, data should only be written to this register when a transmit is not currently in progress. Read data is buffered. When a transfer is complete, the receive data is transferred to a single byte data buffer, where it is later read. A read of the SPI data register returns the value of the read data buffer. In burst mode operation, a read or write to this register will start a SPI data transfer. Writing to the SPIDAT register will initiate a SPI transfer which will transmit the write data. Reading the SPIDAT register will return the received data from the previous SPI transfer, and initiate a new transfer to receive the next byte of data from the SPI bus. When an SPI transfer is initiated by a read of the SPIDAT register, the transmitted data will be 8 zero bits. Software should allow enough delay for one transfer to finish before accessing this register a second time. In non-burst mode operation, a write to this register will start a SPI data transfer. Writes to this register will be blocked from when a data transfer starts to when the SPIF status bit is set, and the status register has not been read. Reads from this register will not initiate a transfer on the SPI bus.

Table 411: SPICLK - SPI clock register

Symbol	Position	Access	Value	Description
SPI	7:0	R/W	0x00	Clock counter setting. This register is used to generate the SPI_SCK signal. The register indicates the number of system clock cycles that make up a SPI clock. The value should always be an even number. As a result, bit 0 is reserved and will always be 0. When the FPEN bit in the SPICON register is cleared, the value of this register should be greater than or equal to 8.

7.14.3 SPI Operation

The following sequence describes how to process a data transfer with the SPI block. This process assumes that any prior data transfer has already completed.

1. Set the SPICLK register to the desired clock rate.
2. Set the SPICON register to the desired settings (BURST=0).
3. Clear SS signal (using Port SFR).
4. Write the data to be transmitted to the SPI data register. This write starts the SPI data transfer.
5. Poll SPISTAT register, until SPIF bit is set.
6. Read the received data from the SPI data register (optional).
7. Go to step 4 if more data is required to be transmitted.
8. Clear SS signal to end SPI transaction.

Remark: A read or write of the SPI data register is required in order to clear the SPIF status bit. Therefore, if the optional read of the SPI data register does not take place, a write to this register is required in order to clear the SPIF status bit.

7.14.3.1 SPI Burst read operation

The following sequence applies:

1. Set the SPICLK register to the desired clock rate.
2. Set the SPICON register to the desired settings (BURST=1).
3. Clear SS signal (using Port SFR).
4. Read or Write byte to (from) the SPI data register. This starts the SPI data transfer.
5. Software delay.
6. Read the received data from the SPI data register. This read also starts the next SPI data transfer.
7. Go to step 5 if more data is required to be received.
8. Set SS signal to end SPI transaction.

7.14.3.2 SPI Burst write operation

The following sequence applies:

1. Set the SPICLK register to the desired clock rate.
2. Set the SPICON register to the desired settings (BURST=1).
3. Clear SS signal (using Port SFR).

4. Write the data to be transmitted to the SPI data register. This write starts the SPI data transfer.
5. Software delay.
6. Go to step 4 if more data is required to be transmitted.
7. Set SS signal to end SPI transaction.

Remark: In burst mode operation, an SPI transfer is initiated by either a read or a write to the SPIDAT register. Thus, read and write operations (receive and transmit, respectively) may be interleaved within one burst transaction.

7.14.3.3 Transfer latency

Transfer latency calculation for burst mode operation: The SPICLK register value gives the number of system clock cycles per bit transmitted on the SPI bus. Thus, the latency to transfer one byte of data is:

tx_latency = 8 * SPICLK

Thus, software should ensure that at least tx_latency system clock cycles elapse before accessing the SPIDAT register. Note that most 8051 instructions execute in one or two cpu cycles (6 or 12 system clock cycles).

Example latency calculation:

Software Latency:

MOVA, SPIDAT	= 1 cpu cycle
MOVX@DPTR, A	= 2 cpu cycles
INCDPL	= 1 cpu cycle
MOVA, SPIDAT	
sw_latency	= 1+2+1 = 4 cpu cycles
4 * 6	= 24 system clock cycles

SPI Transfer Latency:

SPICLK	= 2
tx_latency	= 8 * 2 = 16

24 > 16, Thus the above software sequence is acceptable for SPICLK=2

SPICLK	= 4
tx_latency	= 8 * 4 = 32

24 < 32, Thus the above software sequence is NOT acceptable for SPICLK=4

7.15 ADC (Analog to Digital Converter)

7.15.1 Overview

The ADC is implemented as a 10-bit successive approximation design. It has 8 multiplexed inputs, which can be individually selected for conversion to be performed on. The design allows for measurement to be performed continuously without software control. The results of the last completed 10-bit conversion are stored in SFRs. The ADC has its dedicated power supply ground and 3V3 reference voltage located on pads ADC_VSSA and ADC3V3VDDA respectively.

7.15.2 ADC Registers

7.15.2.1 ADCON Register

The ADC control register (ADCON, offset $D5) sets the characteristics of the ADC conversion. This register is common to all the 8 ADC channels. Table 412 describes the content of this register.

Table 412: ADCON Register Description

Symbol	Position	Name	Significance
adc_clk_mode	7	ADC clock mode	‘0’ -> clk/4, ‘1’-> clk/8. With 16MHz micro clock use /4 option. For 19 MHz to 36 MHz use /8 option. R/W field. Reset value=’0’.
Int_status	6	Interrupts status	Write ‘0’ to clear / reads status ‘0’. R/W field.
adc_resol	05:04	ADC resolution	ADC resolution sets ADC resolution. '00' -> 4bit, '01' ->6 bits, '10' -> 8bits, '11'->10bits. Reset value='00'. R/W field.
adc_en	3	ADC enable	ADC enable 0 -> non-operational mode. '1-> operational mode $0. R/W field.
adc_c_scan	2	ADC continuous scan	ADC continuous scan should always be 1 for continuous scan. 0 is single scan which is not supported in PNX2015 $1. R/W field.
adc_abort (W)	1	ADC abort	ADC abort write 0 -> no effect, write 1->abort AD conversion progress, always read 0. Write only field.
adc_stasta	0	ADC start	ADC start (wr) / status (rd) write 0 -> no effect, write 1->Start AD conversion, read 0 when NO conversion in progress, read 1 read 0 when conversion in progress. Reset value=$0. R/W field.

7.15.2.2 ADCTIMH and ADCTIML Registers

The ADCTIMH (offset $F5) and ADCTIML (offset $F4) registers set the time characteristics of the ADC conversion. Table 413 and Table 414 give a description of the constituent bit fields. Typical settings for ADCTIMH and ADCTIML are: ADCTIMH=053H, ADCTIML=04AH.

Table 413: ADCTIMH Register Description

Symbol	Position	Name	Significance
adc_start_time	7:4	ADC start time	Time required by suddac to become fully operational after power up. ADC start time = no of clock cycles x 64. Start_time = 0 corresponds to 1024 clock cycles. Reset value=$F. R/W field.
adc_comp_time	3:0	ADC compare time	Time required to compare sampled voltage against internal DAC voltage. ADC compare time = number of clock cycles. Comp_time = 0 corresponds to 32 clk cycles. Reset value=$F. R/W field.

Table 414: ADCTIML Register Description

Symbol	Position	Name	Significance
adc_dac_time	7:5	ADC dac time	Time required for internal DAC voltage to settle. Dac_time = number of clock cycles. Dac_time = 0 corresponds to 32 clk cycle. Reset value=$F. R/W field.
adc_sample_time	4:0	ADC sample time	Time period of which analog input voltage is being sampled. Sample_time = number of clock cycles. Sample_time = 0 corresponds to 32 clock cycles. Sample_time 1 and 2 are illegal. Reset value=$F. R/W field.

7.15.2.3 ADCSEL Register

The ADCSEL (offset $F3) register selects the channels for which the ADC conversion is applied. Table 415 gives a description of the constituent bit fields.

Table 415: ADCSEL Register Description

Symbol	Position	Name	Significance
adc_csel7	7	Channel select 7	ADC channel select. 0-> ADC will skip channel 8 during conversion scan 1-> this channel is included in the conversion scan. Reset value= $0. R/W field.
adc_csel6	6	Channel select 6	ADC channel select. 0-> ADC will skip channel 7 during conversion scan 1-> this channel is included in the conversion scan. Reset value=$0. R/W field.
adc_csel5	5	Channel select 5	ADC channel select. 0-> ADC will skip channel 6 during conversion scan 1-> this channel is included in the conversion scan. Reset value=$0. R/W field.
adc_csel4	4	Channel select 4	ADC channel select. 0-> ADC will skip channel 5 during conversion scan 1-> this channel is included in the conversion scan $
adc_csel3	3	Channel select 3	ADC channel select. 0-> ADC will skip channel 4 during conversion scan 1-> this channel is included in the conversion scan. Reset value=$0. R/W field.
adc_csel2	2	Channel select 2	ADC channel select. 0-> ADC will skip channel 3 during conversion scan 1-> this channel is included in the conversion scan. Reset value=$0. R/W field.
adc_csel1	1	Channel select 1	ADC channel select. 0-> ADC will skip channel 2 during conversion scan 1-> this channel is included in the conversion scan. Reset value=$0. R/W field.
adc_csel0	0	Channel select 0	ADC channel select. 0-> ADC will skip channel 1 during conversion scan 1-> this channel is included in the conversion scan. Reset value=$0. R/W field.

7.15.2.4 ADCRL# and ADCH Registers

The ADCRL0..7 and ADCH give the ADC conversion result low and high bytes. Table 416 and Table 417 give a description of the constituent bit fields.

Table 416: ADCRL0..7 Register Description

Symbol	Position	Name	Significance
ADC_RL#	7:0	ADC result register for channel #	Least significant 8-bits of ADC conversion result for channel 0. 4-bit resolution adc_result -> ADCRH(1:0)&ADCRL0(7:6), 6 bits resolution -> ADCRH(1:0)&ADCRL0(7:4), 8 bits resolution -> ADCRH(1:0)&ADCRL0(7:2), 10 bits resolution -> ADCRH(1:0)&ADCRL0(7:0). Read only register.

Table 417: ADCRH Register Description

Symbol	Position	Name	Significance
ADC_RH	1:0	ADC result register shared between channels	Most significant 2-bits of ADC conversion result shared by all channels. Last read LSB value of channel x will load this register with it's remaining Most significant bits. Read only register.

7.16 Boot Process and Mode Selection

7.16.1 Boot Process

Figure 274 shows the state diagram of the TV Microcontroller boot process. The diagram features the following operating modes: Emulation mode Section 7.18, Program mode Section 7.17 and Normal Mode.

Table 418: Operating modes

Operating Mode	PSEN	ALE	EA	P2.0	P2.1	MODE
Normal Operation	1	1	1	1	1	1
Flash Programming	1	1	1	1	1	0
In-Circuit Emulation	0	0	0	0	0	X

Remark: the selection of In-Circuit Emulation mode (PSEN, ALE, EA, P2.0, and P2.1) takes place when MC_RESET is deasserted, while the selection between Normal and Flash Programming modes takes place several microseconds later.

After Microcontroller reset, the normal/program or emulation mode is selected according to signals PSEN, ALE, EA, P2.0, and P2.1. If emulation mode is selected, then the TV Microcontroller executes code from the emulator’s memory. Otherwise, it executes code from the internal Boot ROM (PROM). The ROM contains boot loader code for both reading from and writing to Serial-Flash memory attached to the TV Microcontroller via the SPI. The first instruction is to check the value of Mode0 pin and select either Normal or Flash-Programming mode. If Normal mode is selected, then the content of the external flash is copied via SPI to internal Program RAM (PRAM). On completion of copying, the code is executed from PRAM. If Flash-Programming mode is selected, an application code image is downloaded from the UART, and written into the external flash memory (see Section 7.17 Flash Programming).

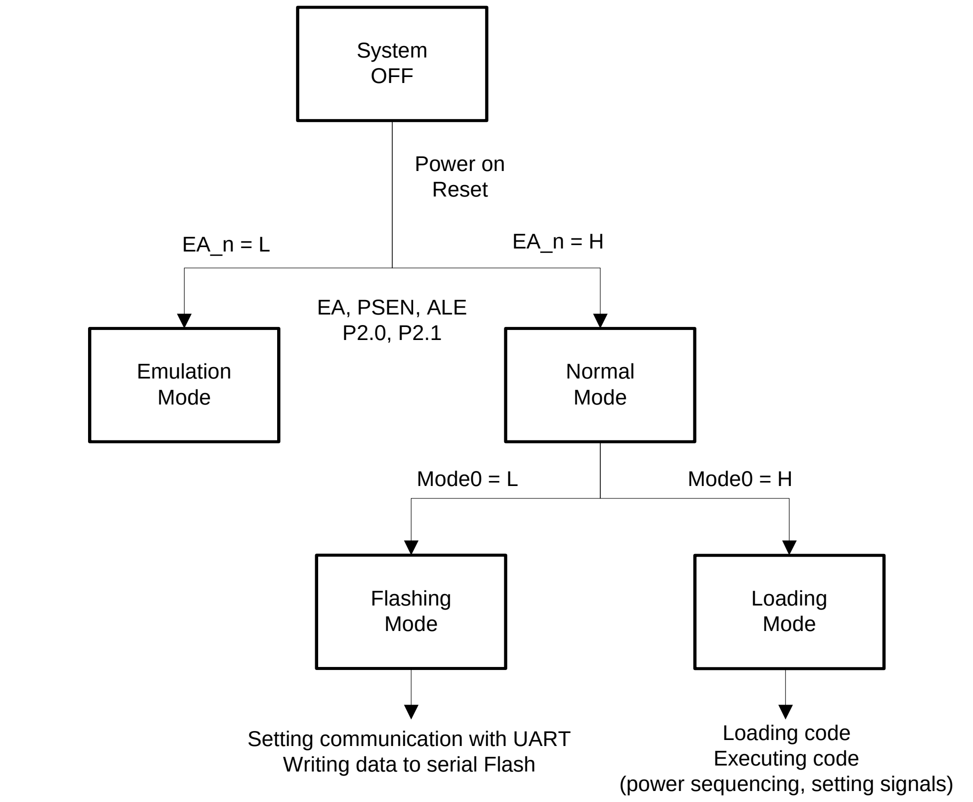


Fig 274. Operation Modes Emulation, Program and Normal Mode

Table 419 defines the mode selection pins for programming the external flash memory.

Table 419: Mode selection

Symbol	Description	Function mode	Value
P6_4	mode select input pin	programme	0
		normal	1
P6_5	mode select output pin	programme flash error	toggle[1]

[1] A square waveform at f = $2/2^{16}$ is generated to indicate a programming error of the flash memory.

7.17 Flash programming

The following hardware and software equipment are typically needed for developing code for the TV Microcontroller:

- PNX2015 module: with embedded PNX2015 chip, a Motorola compatible Flash memory, a 16 MHz clock circuitry, a UART cable connector (connected to the Microcontroller UART pins), a reset circuitry, a jumper-based circuitry to set pin Mode0 to either GND or 3V3.
- External power supply (1V2 and 3V3)
- IBM PC compatible, with standard serial port (32-bit version of Windows operating system)
- Compiler to write and compile code

- Source-level Debugger V2.0.5.C or higher

7.17.1 Code download procedure

1. Connect the PNX2015 module to a PC via a UART cable
2. Set Mode0 pin to GND
3. Open a Hyperterminal window and configure it with the following parameters:

   Bitrate=115200, Databits=8, Parity=None, Stopbit=1, FlowControl=None, Ascci sending: ‘Echo typed characted locally’ box checked.
4. Apply power (3V3 and 1V2)
5. Press the reset button on the PNX2015 module
6. Press the letter 'U' on the Hyperterminal window, Hyperterminal should show:

   PNX2015 Loader V1.0, 19-09-2003, DEVID=0x05, Erasing, MCSUM=0x
7. Download the hex file by using the menu 'download text file’

   Remark: maximum size of code that can be downloaded on PNX2015 is 62K
8. The file content and then the checksum is viewed on the Hyperterminal window
9. If the download is successful, then Hyperterminal should show the following message

   DCSUM=0x, OK, MCSUM-0x, Programming, PCSUM=0x, Finished

7.17.2 Running the code

1. Set pin Mode0 high by removing the programming jumper
2. Power on the PNX2015 module
3. Press the reset button on the PNX2015 module

7.18 Emulation

The TV Microcontroller uses the Enhanced Hooks protocol for system emulation. Emulation tool support for the PNX2015 TV Microcontroller are available from Ashling Microsystems and Hitex Solutions:

- Ashling Microsystems provide the Ultra-51 ICE Enhanced Hooks Probe and PNX2015 adaptor together with PathFinder Software (order code: ULTRA - PNX2015). Users with Ashling Ultra-51 require a re-configuration kit Ref. 2.
- Hitex Solutions provide the Dprobe HS-PEH ICE and Extended Probe (ZS-PEH-CON) together with HiTop Software. Also Dtrace16 or Dbox16 are available for trigger/trace functionality Ref. 3.

8. Limitations

8.1 General Limitations of PNX2015

The 27Mhz clock must be common between PNX8550 and PNX2015.

For correct video communication with PNX8550, the PNX2015 must connect to PNX8550 27 MHz clock output.

A potential divider is required as PNX8550 outputs a 3V3 clock while the PNX2015 needs a 1V2 clock.

For limitations updates for the PNX2015 contact Philips Semiconductors customer support: DTV.support@philips.com

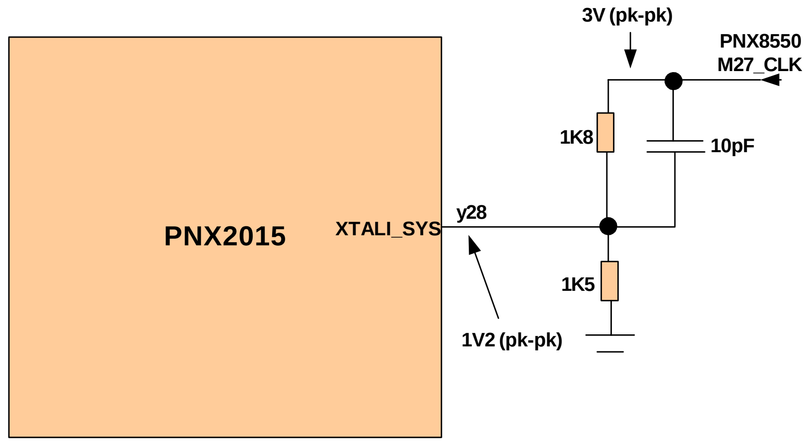


Fig 275. PNX8550 27MHz clock signal reduction from 3V3 to 1V2

8.2 AVIP

8.2.1 I2S Slave-Auto Mode.

The I2S "Slave-Auto" Mode in AVIP1 is not operational. This mode is not a required function.

8.2.2 Sound Core

The Sound Core has the following limitations:

1. Holding the Sound Core in Reset (using the SND_CLK_POR_N bit in GPR) causes the DAC outputs to go to 0.5Vref. When released from reset, they fall to 0V then ramp back up to 0.5Vref. This causes a click from the audio outputs at power-up.
2. On releasing the Sound Core from reset, the reset values in the H/W control registers are sometimes undefined.

A software workaround has been accepted, which reads all affected registers after releasing the reset, and assert the reset again if any non-zero values are found.

8.2.3 PLL divider settings in AVIP1 GPR.

The PLLs in AVIP1 require programming with application specific settings to overwrite the default values.

8.2.4 AVIP UV Swap for within 2015

Incorrect colors are displayed for 2FH modes 1080i@50Hz and 1080i@60Hz. This is for embedded syncs.

Set following register values are required to fix the problem:

1080i@50Hz: hrefs_2fh = 0x3CD, hrefb_2fh = 0x3BF

1080i@60Hz: hrefs_2fh = 0x29, hrefb_2fh = 0x3BF

8.2.5 New settings for AVIP register

The values for Viddec register SUBPIX_PLL_SYNC3 (0x14C) for video modes 1080i@50Hz and 1080i@60Hz should be:

i@50Hz : 0x000F37BF

i@60Hz : 0x0000A7BF see AVIP limitation 5.

8.2.6 Audio distortion at low VDD

The 1152i test pattern in Columbus is not generated correctly. This is associated with the new Australian standard.

No impact on functionality.

8.2.7 I2S Auto-Master mode

The audio clock system has a mode in which it automatically switches between I2S Master and Slave modes, depending on whether an external clock is supplied. This automatic switching is not operational.

This is not a required function.

8.2.8 Audio power up

The audio is not muted correctly on power-up, resulting in an audible 'plop' as the audio outputs are switched on.

An external mute circuit is required during the power up phase.

8.2.9 Register map in sound cores

Due to additional registers in the audio core, the register allocation was partially re-mapped.

The changes are as follows:

7F0000C: New register added - INF_CLIP_MGT_REG. All subsequent registers shifted by one place in address map.

7F00010: New bits added to control polarity switches for surround output channels

7F00038: New bits added to control polarity switches for ADC inputs and DAC (MPIF) outputs

7F00048: SNDMOD value of $B changed from Active Surround to Multichannel

Post-Processing (since Active Surround is removed). Also added bits to control in which channels the EQ is active.

7F0004C: Added CANDEOENABLE bit

7F0005C: Changed default value of VDDMIXLEV from 0 to 4. Also added VDDBALANCE control.

7F00060: Inserted register DPL2_MM_CON_REG (was previously at address 7F000F8). All subsequent registers shifted by another place in address map.

7F00064: Changed default MASTVOL value from 0dB to Mute.

7F0009C - 7F000E4: Numerous changes to Equalizer control registers due to new equalizer design. Addresses of subsequent registers (MON_SEL_REG, MPI_CONTROL_REG, ACC_MAINCH_REG, ACC_CENTRECH_REG) also shifted.

7F000E8: Default values of ASMOCP1 and ASMOCP2 modified

All registers relating to Active Surround (previously in addresses 7F000E0 - 7F000F4) removed.

These differences apply to both AVIP instances.

8.2.10 DCCANCEL bit

DCCANCEL bit causes -6dB amplitude change

A -6 dB amplitude change occurs when the DCCANCEL bit is enabled. The recommended software solution is to increase the values of all the following registers by 6 (to increase gain by 6dB). This amplifies the signal to noise but this is acceptable in this point in the audio module.

In DemDec DSP:

DEM_LEVELADJUST_REG / EXTAM_GAIN

In Audio DSP:

LEV_ADJ_ADC_DEMDEC_IIS_REG / PIPMONOLEV

LEV_ADJ_ADC_IIS_REG / ADCL_ALEV

LEV_ADJ_ADC_IIS_REG / ADCR_BLEV

Enabling the DCCANCEL bit in the SND_DD_M00 register of the Sound Core causes an amplitude reduction of 6dB on all MPIF ADC audio inputs (excluding the SIF input). The 6dB amplitude change does not add any other audio degradation, and DC cancellation still works.

8.2.11 Audio Core update

The CANDEO feature is fully functional in Stereo mode, but there is a 3dB level difference between CANDEO ON and OFF. This is apparent in all DPL and Virtualizer related modes except VDD522 and VDD523.

That means all levels are normal with CANDEO OFF. But with CANDEO switched ON the output level increases by +3dB in all involved output channels (except center output). The recommendation (except for VDD522/523) is to reduce this +3dB gain with the volume control whenever CANDEO is switched ON, with a further adjustment when switching OFF.

8.3 HD Subsystem

8.3.1 Supply sequencing and rise time

PNX2015 does not tolerate incorrect supply sequencing and rise time.

For successful operation of the PNX2015 either of the following power up sequences are required:

1. If the supplies rise over the same time period of 6ms and all start rising at the same time.
2. If the supplies are applied in ascending order. While the time interval between supply switch on is not restricted there is a 6ms rise time requirement for each supply.

8.3.2 PMAN2 (tunnel) monitor

The PMAN2 (tunnel) monitor measurements are not reliable as different block transactions are measured from the block identified. Functionality accepted.

8.3.3 HD Subsystem I^{2}C block

HD Subsystem I^{2}C block unable to handle burst read/write. Functionality accepted.

8.3.4 Vip - Vo streaming

The Vip-VO module does not support mode changes from Direct streaming mode to Memory streaming mode. Use Memory streaming mode for both main and sub streams.

8.3.5 VO clock update

There is a limitation in the VO supporting HDMI format changes when in slave mode using the VIP derived clock. When in memory streaming mode the VO clock must use the DDS clock and not the VIP clock.

8.3.6 3D comb filter restriction on DDR

The Columbus 3D comb filter can not operate concurrently with other video processes that also require DDR access. Buffering restrictions in the 3D comb filter require specific latencies for DDR accesses.

8.4 LVDS

The LVDS FRS describes clk_vo1_proc' and 'clk_vo2_proc' to control LVDS PLL

Clock select 'LVDS PLL' source is 'clk_lvdsx7'. The clk_vo1/2_proc clocks no longer drive any circuitry on the PNX2015.

9. Glossary

ADC – Analog to Digital Converter

AES – Advanced Encryption Standard

AGC – Automatic Gain Control

ANC – Ancillary

ARD – Auxiliary RAM Disable

ASD – Auto Standard Detection

ATSC – Advanced Television Systems Committee

AUX – Auxiliary

AVIP – Audio Video Input Processor

AVL – Auto Volume Level

BCU – Bus Control Unit

BSDL – Boundary-Scan Description Language

BTSC – Broadcast TV System Committee

CC – Closed Caption

CGU – Clock Generation Unit

COLMUX – Color Multiplexer

CRT – Cathode-Ray Tube

DACS – Digital Access and Cross-connect System

DAF – Discriminating Averaging Filter

DBE – Dynamic Base Enhancement

DCS – Data Capture Status

DCSN – Device, Control and Status Network

DCU – Data Capture Unit

DDEP – Demodulator and Decoder Easy Programming

DDR – Double Data Rate

DEMDEC – Demodulator Decoder

DLINK – I^{2}D Link

DMA – Direct Memory Access

DPL – Dolby Pro Logic

DQPSK – Differential Quadrature Phase Shift Keying

DSP – Digital Signal Processor

DTL – Data Transition Level

DUB – Dynamic UltraBass

DVB – Digital Video Broadcasting

DVD – Digital Video Disc

DVP – Digital Video Platform

EAV – End Active Video

EBU – European Broadcasting Union

EIAJ – Electronic Industries Association of Japan

EMC – Electro Magnetic Compatibility

Fref – Video field indicator

GTU – Global Task Unit

HD – High Definition

Hsync – Video horizontal synchronisation signal

ICE – In-Circuit Emulator

I/O – Input/Output

IP – Intellectual Property

ITU – International Telecommunication Union

LORE – LOcal REgression (noise reduction)

LSB – Least Significant Bit

LVDS – Low Voltage Differential Signalling

MC – Motion Compensation

MMIO – Memory Mapped Input / Output

MPEG – Moving Picture Experts Group

MPIF – Multiple Platform Interface

MSB – Most Significant Bit

MTL – Memory Transition Level

NICAM – Near Instantaneously Companded Audio Multiplex

NTSC – National Television System Committee

PAL – Phase Alternating Line

PCA – Programmable Counter Array

PCB – Printed Circuit Board

PCON – Power Control

PI – Peripheral Interface

PLL – Phase Locked Loop

PMAN – Pipelined Memory Access Network

PNX – Philips NeXperia

POR – Power On Reset

PWM – Pulse Width Modulation

RAW – Raw data capture

RFI – Radio Frequency Interference

RGB – Red Green Blue digital/analog color information

RSL – Register Summary List

RST – Reset

SAD – Sum of Absolute Differences

SAV – Start Active Video

SCL – Serial Clock

SDRAM – Synchronous DRAM

SECAM – Systeme Electronique Couleur Avec Memoire

SFR – Special Function Register

SMOD – Serial Mode

SMR – Sum of Magnitudes of Residues

SPI – Serial Peripheral Interface

SRC – Sample Rate Conversion

SSS – Static Standard Selection

STG – Screen Timing Generator

SWAN – Spatial temporal Weighted Averaging Noise reduction

Syncs – Video synchronisation signals

TAP – Test Access Port

TCB – Test Control Block

TDA9975 – Video input decoder device

TNR – Temporal Noise Reduction

TSU – Time Stamp Unit

UART – Universal Asynchronous Receiver Transmitter

URT – Universal Register Toolkit

VBI – Vertical Blanking Interval

VIDDEC – Video Decoder

VIP – Video Input Processor

VITC – Vertical Interval Time Code

VMPG – Video MPEG decoder

VO – Video Output

VPS – Virtual Private Server

Vsync – Video vertical synchronisation signal

WDT – Watchdog Timer

WLE – Watchdog Load Enable

WST – Western Standard Time

WSS – Wide Screen Signaling

10. References

[1] **I^{2}C –** I^{2}C specification v2.1, Jan 2000, Document order number:9398 393 40011 www.semiconductors.philips.com/buses/i2c

[2] **Ashling Microsystems –** Contact Ashling Microsystems Ltd. National Technology Park, Limerick, Ireland. www.ashling.com

[3] **Hitex Solutions –** Contact Hitex Development Tools, Karlsruhe, Germany. www.hitex.com

[4] **AV link –** Domestic and similar electronic equipment interconnection requirements. AV.link - Part 1, Part 2-1, Part 2-2 and Part 2-3.EN 50157-1, EN 50157-2-1, EN 50157-2-2, EN 50157-2-3, Cenelec

[5] **DCS Network –** DCS Network Specification, Version 1.0, Hardware Architecture Board, 4-Jan-2002

[6] **80C51 –** 80C51-Based 8-Bit Microcontrollers, Philips Semiconductors, 1993

[7] **PNX2015 family Data Sheet –** document number 9397 750 13379.

[8] **PNX2015 Register Summary List –** document number AN10334.

[9] **LVDS Owners Manual Low-Voltage Differential Signalling –** Spring 2000, 3rd Edition, http://www.national.com/appinfo/lvds/files/ownersmanual.pdf.

[10] **I2S bus specification – February 1986 Revised:June 5, 1996**, http://www.semiconductors.philips.com/acrobat_download/various/I2SBUS.pdf.

[11] Rec. ITU-R BT.656-4 1 **–** RECOMMENDATION ITU-R BT.656-4 INTERFACES FOR DIGITAL COMPONENT VIDEO SIGNALS IN 525-LINE AND 625-LINE TELEVISION SYSTEMS OPERATING AT THE 4:2:2 LEVEL OF RECOMMENDATION ITU-R BT.601 (PART A, http://www.inst.eecs.berkeley.edu/~cs150/documents/ITU656.pdf.

11. Disclaimers

Life support – These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes – Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

12. Licenses

Dolby Laboratories

'Dolby' and 'Pro Logic' are trademarks of Dolby Laboratories, San Francisco, USA. Products are available to licensees of Dolby Laboratories Licensing Corp., 100 Potrero Avenue, San Francisco, CA, 94103, USA. Tel: 1-415-558-0200, Fax: 1-415-863-1373.

Supply of this implementation of Dolby Technology does not convey a license, nor imply a right under any patent to use this implementation in any final product. A license for such use is required from Dolby Laboratories.

BBE Sound

BBE is a registered trademark of BBE Sound Inc., 5381 Production Drive, Huntington Beach, CA, 92649, USA. The use of BBE needs licensing from BBE Sound Inc. Tel: 1-714-897-6766, Fax: 1-714-895-6728.

dbx - TV noise reduction

A Set-Maker License is required for use of this product under one (or more) of the following patents: US4,539,526; 5,796,842; 6,118,879 and U.S. Patent Application 09/638245 . For further information contact THAT Corporation, 45 Sumner Street, Milford, Massachusetts 01757-1656, USA. Tel: 1-508-478-9200, FAX: 1-508-478-0990

13. Trademarks

Nexperia – is a trademark of Koninklijke Philips Electronics N.V.
TriMedia – is a trademark of TriMedia Technologies Inc.
Motorola SPI – is a trademark of Motorola Inc.
Dolby and Pro Logic – are trademarks of Dolby Laboratories.
THine – is a trademark of THine Electronics Inc.
National – is a trademark of National Semiconductor Corp.
MIPS – is a trademark of Mips Technologies Inc.
BBE – is a trademark of BBE Sound Inc.

14. Figures

continued >>

continued >>

continued >>

continued >>

15. Contents

continued >>

continued >>

continued >>

continued >>

continued >>

continued >>

Date of release: 6 May 2005
Document number: UM10113

Published in The Netherlands