Introduction to LEON3, GRLIB

Adi Katav

akatav@kaltech.co.il

6201129 4(0) 972+ Ext 101

Few words about KAL:

- KAL provides professional ASIC consultancy for Digital/Analog ASIC Projects
- Silicon IP services
- Projects
- More info at our web site

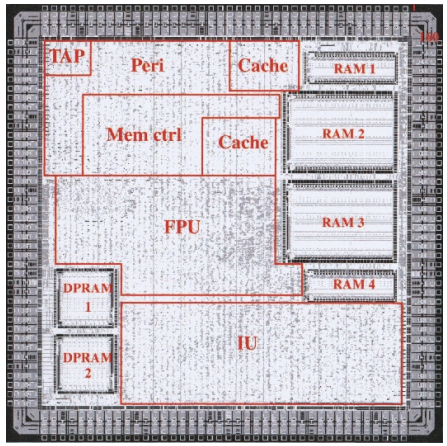
COMPANY INFORMATION

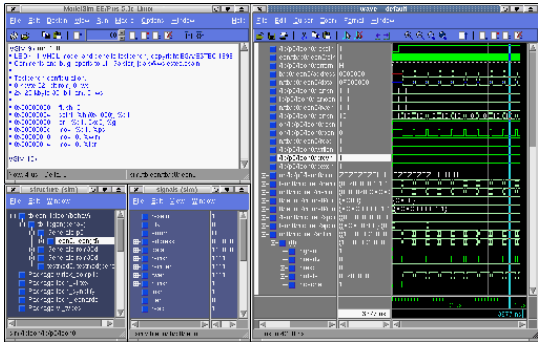
- **Located in Gothenburg, Sweden**
- **Private Company**
- **Management team with 40 years combined experience in the space sector:**
  - Per Danielsson: CEO
  - Jiri Gaisler: Founder and CTO
  - Sandi Habinc: System Design
- **14 engineers with expertise within electronics, ASIC and software design**

GAISLER PRODUCT PORTFOLIO

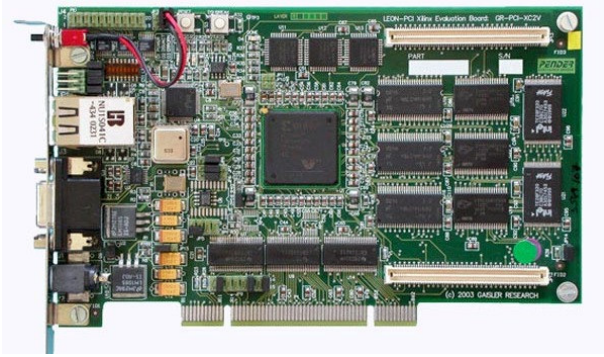
A Reliable ASIC Partner
KAL
Katav Associates

- **LEON3 processor, STD/FT**
- **LEON compatible IP-blocks:**
  - GRFPU, Floating Point Unit
  - Memory controllers
  - PCI, CAN, USB, I2C, SPI
  - 10/100/1000 Mbit Ethernet MAC
  - SpaceWire, 1553





- TSIM, ERC32 and LEON simulator
- GRMON, LEON Debug monitor
- RTOS ports for VxWorks & ThreadX



- **LEON development boards**
- **Test Systems**
- **Technical support and adaptations**
- **Full software development environment based on open source tools**

www.KALtech.co.il

LEON3 FEATURES

A Reliable ASIC Partner

KAL Katav Associates

High Performance

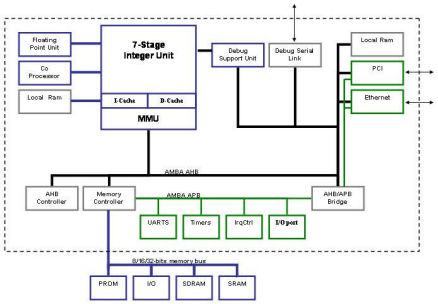
- 400 MHz on 0.13 µm ASIC (std-cell))
- 125 MHz on FPGA

Open Source Standard SW Environment

- GNU C/C++ compiler
- RTEMS real-time kernel
- eCos real-time kernel
- SnapGear Linux

Low Gate Count

- 25 kgates for ASIC
- 3,500 LUT
- 4,000 Cells



Full Simulator and Debug Monitor

- Simulator for developing and debugging SW
- Monitor for HW and SW validation

IP-Library of Cores for SOC Design

- Portable and Vendor independent
- Connection through standard AMBA bus

Customer Approval and Validation

- LEON is the standard processor of ESA
- Numerous commercial customers are using LEON

www.KALtech.co.il

LEON3 SPARC V8 PROCESSOR

- **-7stage pipeline, multi-processor support**
- **Separate multi-set caches with LRU/LRR/RND**
- **On-chip debug support unit with trace buffer**
- **250/400 MHz on 0.18/0.13 um, 250/400 MIPS, 25 Kgates**
- **125 MHz on FPGA, 3500 LUT**
- **Highly configurable:**
  - Cache size 1-256 Kbyte, sets 1-4, LRU/LRR Random
  - Mul/div options, FPU, MMU
  - Pipeline optimisation for specific target technologies
- **SEU tolerance by design for space applications, FT version**

GRLIB IP-LIBRARY

- **GRLIB is a complete design environment:**
  - LEON3 / LEON3-FT
  - Floating Point Unit, Mul/Div
  - Timers, Interrupt Controller
  - Memory controllers (SRAM, SDRAM, DDR)
  - SpaceWire, CAN, Ethernet, PS2, UART, PCI, MIL-STD-1553B, USB 2.0
  - CCSDS Telemetry/Telecommand
- **AMBA on-chip bus with Plug & Play support**
- **Support for many tools and prototyping boards**
- **Support for portability between technologies**

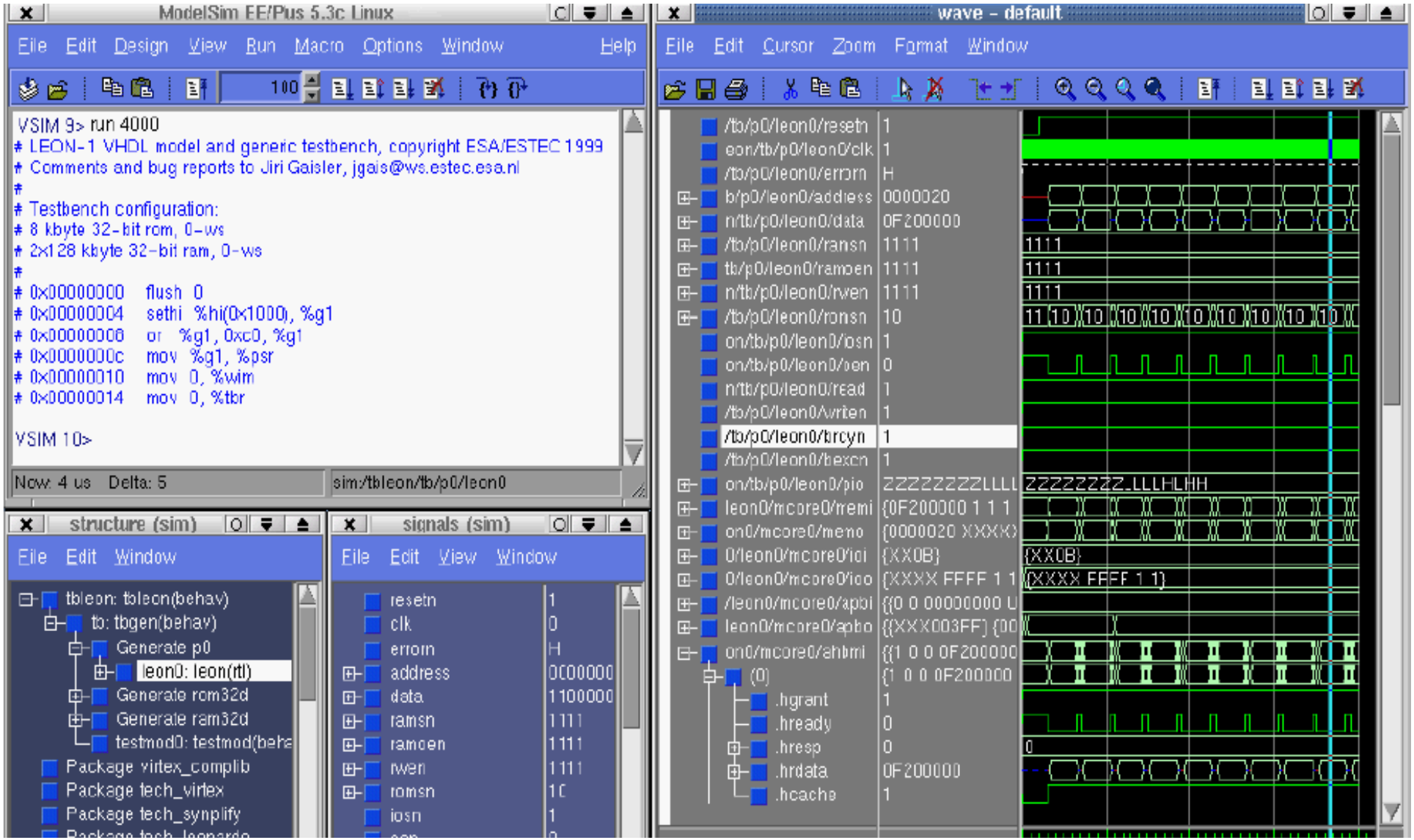
GRLIB PLUG & PLAY

- Fully compliant with AMBA 2.0 AHB/APB buses, with additional sideband signals
- Plug&Play information allows for distributed address decoding, interrupt steering, cachability information, etc.
- No modification of centralized resources, e.g. address decoder, arbiter or interrupt controller.
- Automatic generation of table including vendor and device identifier for each attached core, including version and interrupt information.
- Software can scan table and install the corresponding drivers etc.
- Hardware debuggers can use table for initializing the various IP cores etc.

TSIM SIMULATOR

- **Emulates LEON3 system including memory**
  - Configurable cache size/org., PROM/SRAM size/width
  - Break/watchpoints, trace buffer, stack backtrace
  - Code coverage, check-pointing, profiling, GDB I/F
  - Loadable modules (I/O and AHB) for extension, library version
  - 20 MIPS, accuracy better than 5% (1% typical)
  - Memory EDAC emulation
  - SDRAM support up to 512 Mbyte
  - GRFPU timing
  - MMU emulation
- **Used in many projects (space and consumer)**

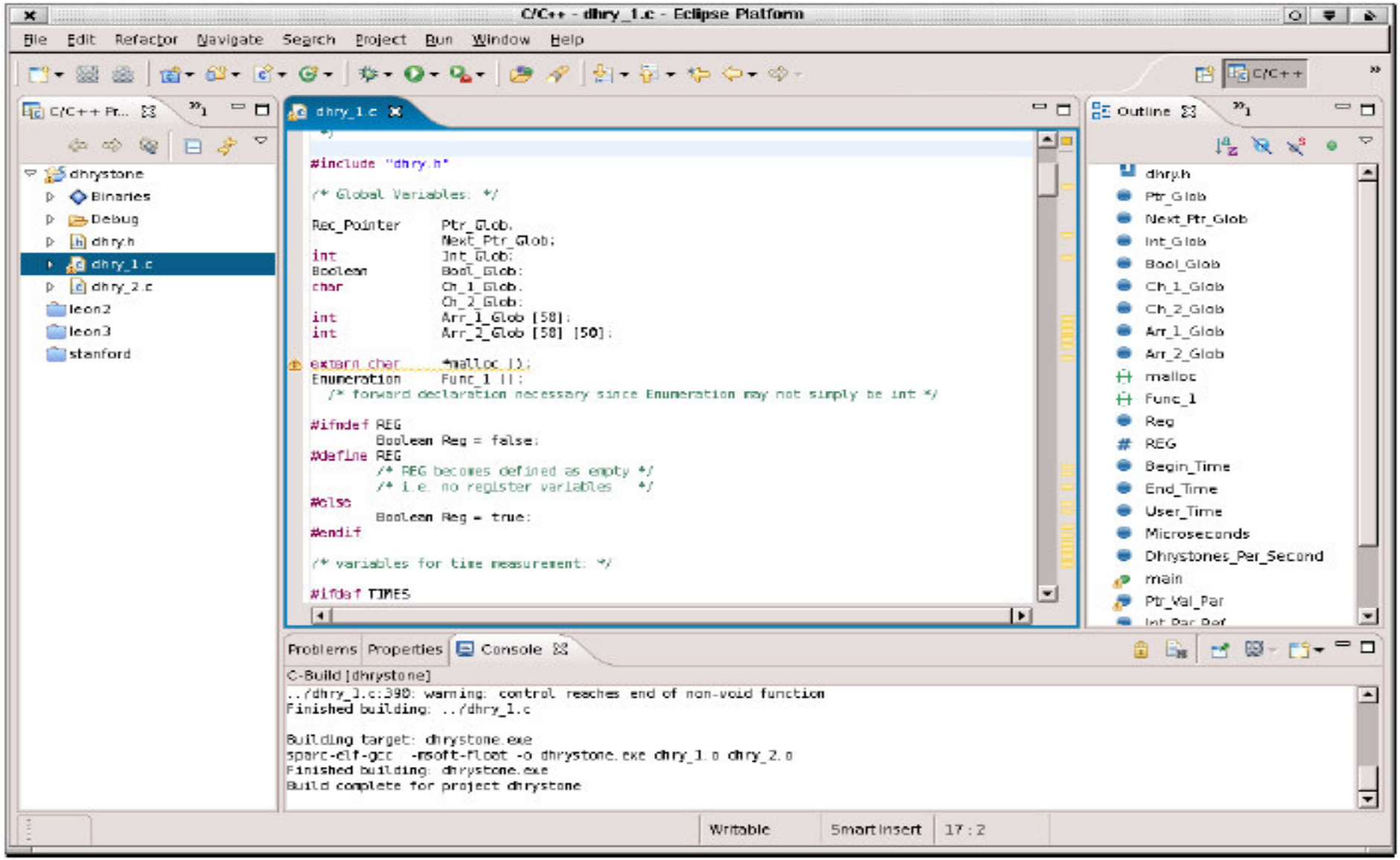
VHDL Simulation



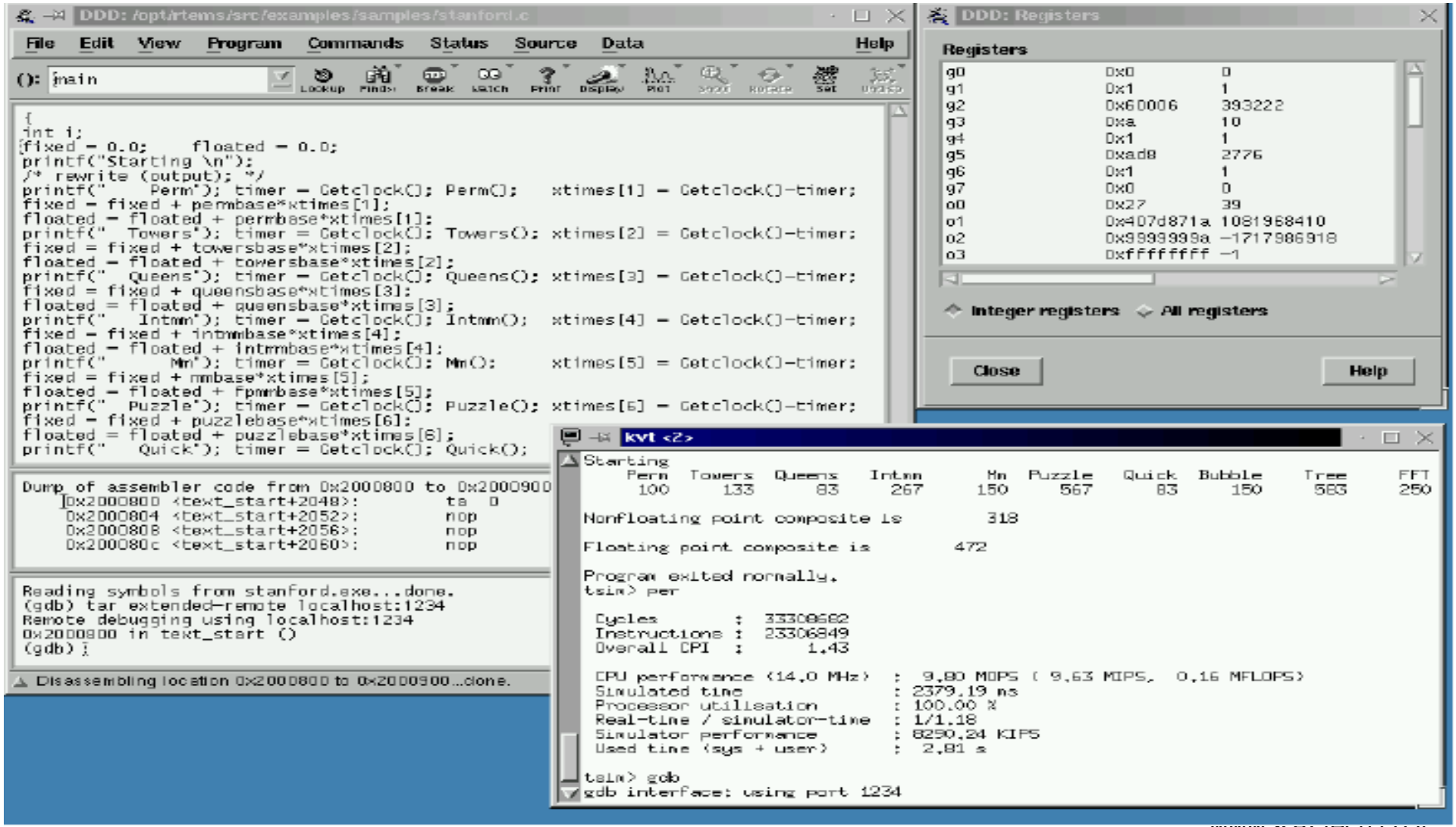
GRGMON

- **Debug monitor for LEON2 systems providing non-intrusive debug environment on real target hardware**
  - Read/write access to all LEON registers and memories
  - Download and execution of applications
  - Built-in disassembler and trace buffer
  - Breakpoint and watchpoint management
  - Command-line mode or graphical user interface
  - Remote connection to GNU debugger (GDB)
  - Supported debug interfaces: PCI, USB, Ethernet, JTAG, UART and SpaceWire
- **Used in many projects (space and consumer)**

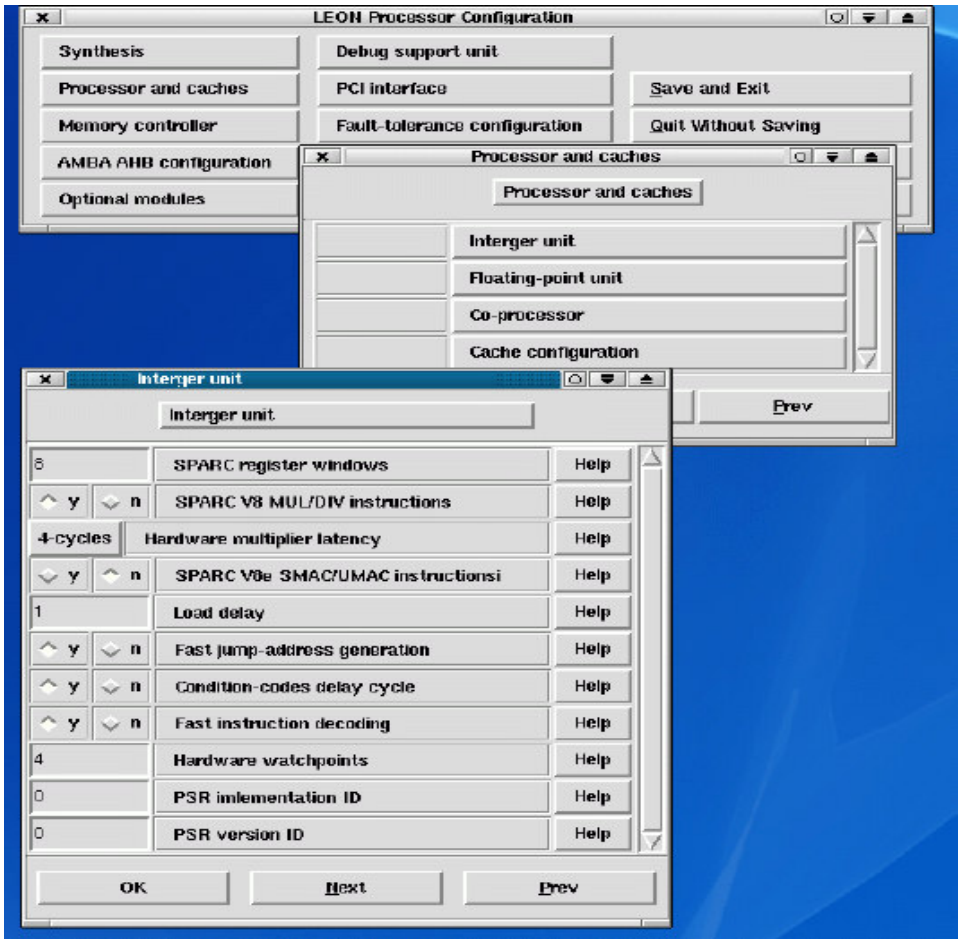
Eclipse


C/C++ - dhry_1.c - Eclipse Platform
File Edit Refactor Navigate Search Project Run Window Help
C/C++
C/C++ Pr...
dhrystone
Binaries
Debug
dhry.h
dhry_1.c
dhry_2.c
leon2
leon3
stanford
dhry_1.c
#include "dhry.h"
/* Global Variables: */
Rec_Pointer Ptr_Glob,
Next_Ptr_Glob;
int Int_Glob;
Boolean Bool_Glob;
char Ch_1_Glob,
Ch_2_Glob;
int Arr_1_Glob [50];
int Arr_2_Glob [50] [50];
extern char *malloc ();
Enumeration Func_1 ();
/* forward declaration necessary since Enumeration may not simply be int */
#ifndef REG
Boolean Reg = false;
#define REG
/* REG becomes defined as empty */
/* i.e. no register variables */
#else
Boolean Reg = true;
#endif
/* variables for time measurement: */
#ifdef TIMES
Outline
dhry.h
Ptr_Glob
Next_Ptr_Glob
Int_Glob
Bool_Glob
Ch_1_Glob
Ch_2_Glob
Arr_1_Glob
Arr_2_Glob
malloc
Func_1
Reg
REG
Begin_Time
End_Time
User_Time
Microseconds
Dhrystones_Per_Second
main
Ptr_Val_Par
Problems Properties Console
C-Build [dhrystone]
../dhry_1.c:390: warning: control reaches end of non-void function
Finished building: ../dhry_1.c
Building target: dhrystone.exe
sparc-elf-gcc -msoft-float -o dhrystone.exe dhry_1.o dhry_2.o
Finished building: dhrystone.exe
Build complete for project dhrystone
Writable
Smart Insert
17 : 2

GDB/DDD


DDD: /opt/rtems/src/examples/samples/stanford.c
File Edit View Program Commands Status Source Data Help
main
Lookup Find Break Watch Print Display Plot Set
{
int i;
fixed = 0.0; floated = 0.0;
printf("Starting \n");
/* rewrite (output); */
printf(" Perm"); timer = Getclock(); Perm(); xtimes[1] = Getclock()-timer;
fixed = fixed + permbase*xtimes[1];
floated = floated + permbase*xtimes[1];
printf(" Towers"); timer = Getclock(); Towers(); xtimes[2] = Getclock()-timer;
fixed = fixed + towersbase*xtimes[2];
floated = floated + towersbase*xtimes[2];
printf(" Queens"); timer = Getclock(); Queens(); xtimes[3] = Getclock()-timer;
fixed = fixed + queensbase*xtimes[3];
floated = floated + queensbase*xtimes[3];
printf(" Intmm"); timer = Getclock(); Intmm(); xtimes[4] = Getclock()-timer;
fixed = fixed + intmmbase*xtimes[4];
floated = floated + intmmbase*xtimes[4];
printf(" Mm"); timer = Getclock(); Mm(); xtimes[5] = Getclock()-timer;
fixed = fixed + mmbase*xtimes[5];
floated = floated + fpmmbase*xtimes[5];
printf(" Puzzle"); timer = Getclock(); Puzzle(); xtimes[6] = Getclock()-timer;
fixed = fixed + puzzlebase*xtimes[6];
floated = floated + puzzlebase*xtimes[6];
printf(" Quick"); timer = Getclock(); Quick();
Dump of assembler code from 0x2000800 to 0x2000900
0x2000800 <text_start+2048>: ta 0
0x2000804 <text_start+2052>: nop
0x2000808 <text_start+2056>: nop
0x200080c <text_start+2060>: nop
Reading symbols from stanford.exe...done.
(gdb) tar extended-remote localhost:1234
Remote debugging using localhost:1234
0x2000900 in text_start ()
(gdb)
Disassembling location 0x2000800 to 0x2000900...done.
DDD: Registers
Registers
g0 0x0 0
g1 0x1 1
g2 0x60006 393222
g3 0xa 10
g4 0x1 1
g5 0xad8 2776
g6 0x1 1
g7 0x0 0
o0 0x27 39
o1 0x407d871a 1081968410
o2 0x9999999a -1717986918
o3 0xffffffff -1
Integer registers All registers
Close Help
kvt <2>
Starting
Perm Towers Queens Intmm Mm Puzzle Quick Bubble Tree FFT
100 133 83 267 150 567 83 150 583 250
Nonfloating point composite is 318
Floating point composite is 472
Program exited normally.
tsim> per
Cycles : 33308562
Instructions : 23306849
Overall CPI : 1.43
CPU performance (14.0 MHz) : 9,80 MOPS (9,63 MIPS, 0,16 MFLOPS)
Simulated time : 2379.19 ms
Processor utilisation : 100.00 %
Real-time / simulator-time : 1/1.18
Simulator performance : 8250.24 KIPS
Used time (sys + user) : 2,81 s
tsim> gdb
gdb interface: using port 1234

GRLIB SOC Config Tool



www.KALtech.co.il

OPERATING SYSTEMS FOR LEON

- **Available today**
  - RTEMS-4.6.5+
  - VxWorks 5.4, 6.3, 6.4
  - ThreadX
  - eCos
  - Linux-2.6.21, uClinux-2.0.x
- **Under development**
  - Nucleus
  - LynxOS

RTEMS FOR LEON

- **RCC tool-chain, based on RTEMS-4.6.5 with extensions**
- **Initially ported from ERC32 BSP, maintained by GR**
- **Updates are being merged with RTEMS CVS (4.8)**
- **Supports AT697 with standard peripherals and any LEON3FT configuration with plug&play**
- **GRLIB drivers: GRPCI, GRSPW, CANOC, 1553, GRETH**
- **New drivers added for RASTA and GR701**
  - InSilicon-PCI, GRSPW, HCAN, PCI-F
  - LAN91C111 Ethernet MAC

VXWORKS FOR LEON

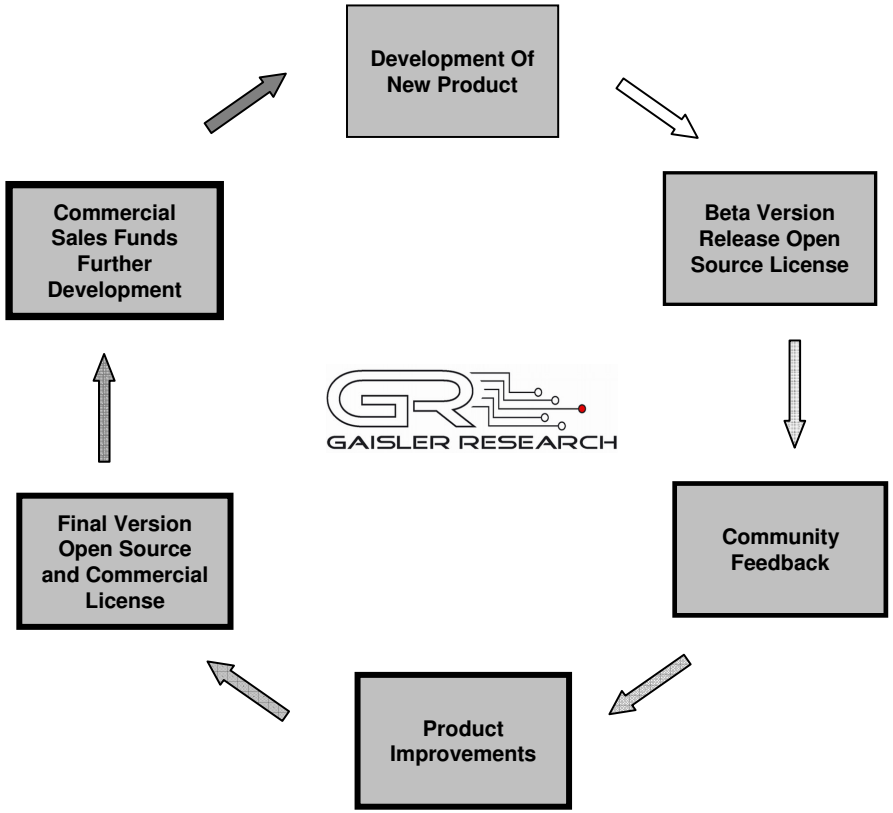
- **Port and BSP available for VxWorks 5.4, 6.3 and 6.4**
- **Full source code for BSP and port (6.x) provided**
- **Compiled with DIAB or GCC (VxWorks-5.4)**
- **LEON2 Drivers:**
  - standard peripherals, LAN91C111
  - Supports MMU in COLE
  - GR701 support in development (PCI, SPW, 1553, CAN)
- **LEON3 Drivers:**
  - GRLIB: standard peri. + GRETH, GRSPW, CAN, 1553BRM
  - LEON3FT MMU support, LAN91C111
- **Workbench support on Linux and Windows hosts**

THREADX FOR LEON

- **Popular RTOS for deeply embedded systems**
- **LEON Port and BSP available for ThreadX-5.0**
- **Full source code for Kernel, BSP and port provided**
- **Small foot-print, GCC tool-chain**
- **Network stack (NetX) under testing**
- **USB stack (USBX) to be ported Q4-2007**
- **Has been used by NASA with ERC32 and Mongoose**
  - NASA Deep Impact Sensor
- **Device driver development done by end-user**

LICENSING MODEL

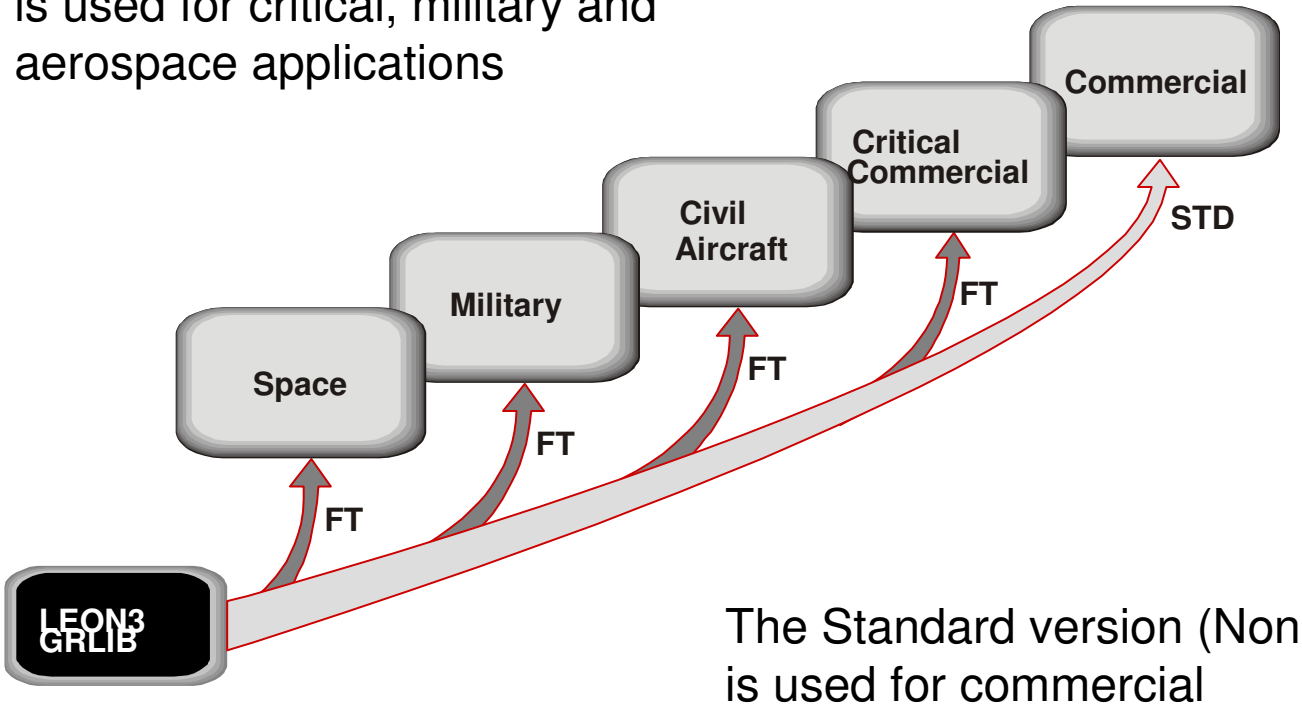
Gaisler Research uses an open source business model based on Dual Licensing. Dual licensing allows to provide commercial licenses for a fee, while at the same time offering the source code under open source licenses.



- Free access for the academic research community
- Free evaluation possibilities for companies
- Large user base gives sizable and quick feedback
- Commercial licenses for IP cores to companies not willing to comply with the GPL license
- Commercial licenses for the LEON3 FT IP core library
- Provide technical support and development tools

MARKET AND APPLICATIONS

The Fault Tolerant (FT) version is used for critical, military and aerospace applications



The Standard version (Non-FT) is used for commercial applications

CUSTOMERS AND APPLICATION

- **Commercial examples**
  - GPS receiver
  - Set top boxes
  - Sensors
  - RF-ID
  - Printers
  - Wireless
  - Power transmission
  - Video games
- **Space**
  - All European space companies
  - US, China, Canada, Korea, Israel, India, Taiwan
- **Research and Universities**
  - Used by hundreds of universities and research centres around the world

Other processors vs. Leon

PRODUCT	ARM 9	ARM 11	ARC 700	Xtensa LX	LEON3
Clock frequency (MHz)	250	400	400	350	400
Pipe-line stages	5	8	7	5	7
Gate count	000'100 <	000'100 <	000'100	000'28	000'25
Synthesizable to FPGA	No	No	Yes	Yes	Yes
Third party SW development tools	Yes	Yes	No	No	Yes
Open Source VHDL	No	No	No	No	Yes
License Cost	High	High	Medium	Medium	Low

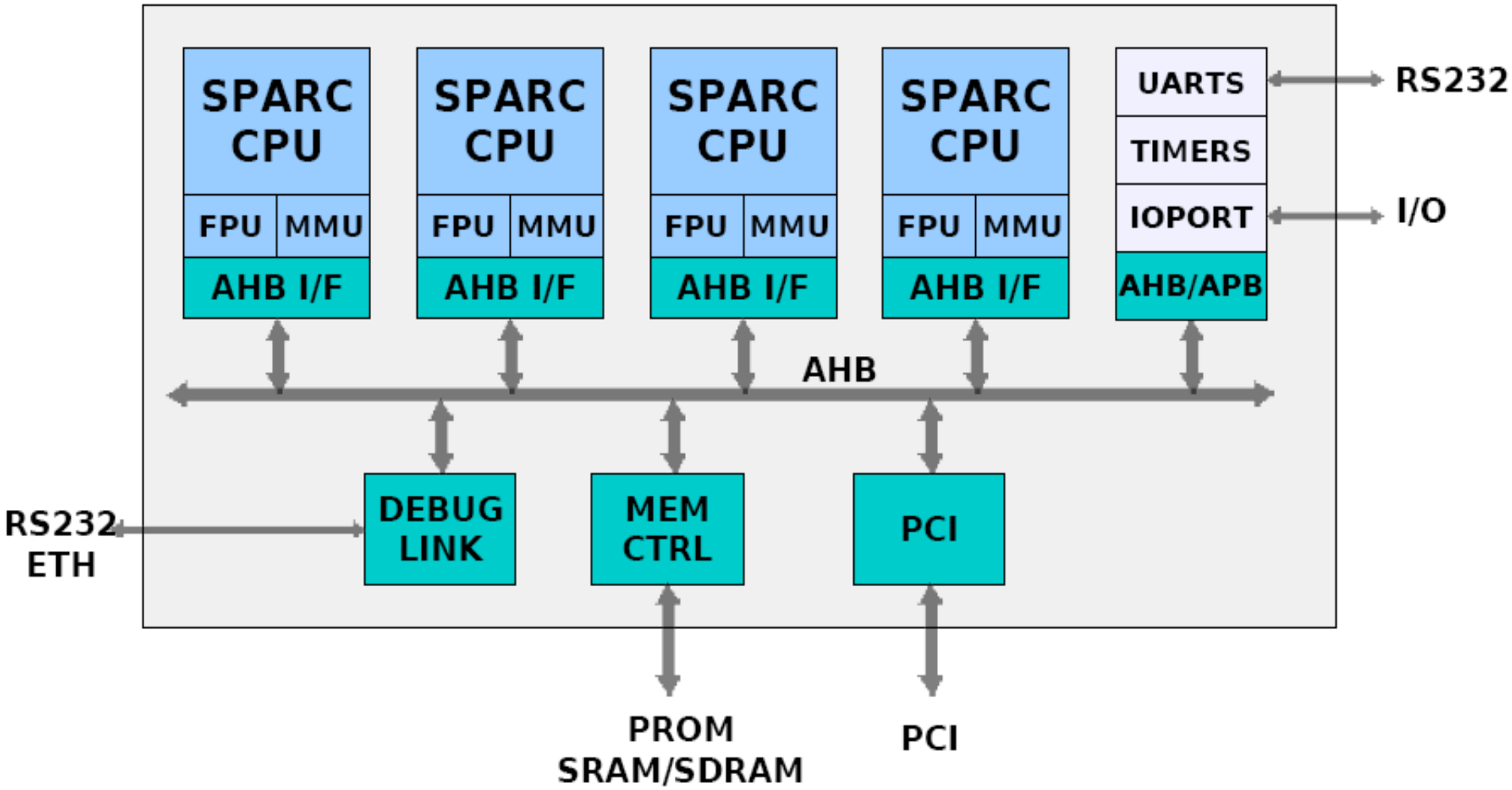
Clock frequency: Depends on process and technology, figures are approximate for 0.13 µm process

Gate count: Depends on processor configuration, figures are for bare processor

VERIFICATION STATUS

- LEON3 independently certified by Sparc International
- Verified for space use according to the stringent requirements of the European Space Agency
- Used as reference design in the low power design package
- Used as reference design by major tool vendors: (Synopsis, Synplify, Mentor, Spirit)
- Promoted by Cadence through the open choice programme
- Partnership for military and space FPGA applications

LEON3 Multi Processing



Example on the NX750LP