TABLE OF CONTENTS

HARDWARE:	**HW**
KITS	1
PC-UNIT	2
PERIPHERALS	3
SOFTWARE:	**SW**
APPLICATION	1
GRAPHIC	2
PRINTER APPLICATION	3
PATCHES	4
INSTALLS	5
OTHER	6
COMMUNICATION	**COM**
APPLICATION	1
PATCHES	2
INSTALL	3
OTHER	4

EXAMPLE: SW3 = Software Printer Application

HARDWARE

HW1-1. PROGRAMMABLE COMMUNICATIONS INTERFACE 2651
HW1-2. INTERRUPT FOR Z80 - K211
HW1-3. OVERVIEW OF PERIPHERAL KITS
HW1-4. K801 RS 232 SWITCHABLE
HW1-5. K803 REAL TIME CLOCK
HW1-6. K804 IEEE488 (IEC) BUS INTERFACE
HW1-7. K806 MOUSE INTERFACE
HW1-8. K215 BUFFERED SYNC/ASYNC INTERFACE
HW1-9. K216 SCC COMMUNICATION INTERFACE
HW1-10. K233 SHARED RAM
HW1-11. K234 68008 PROCESSOR BOARD
HW1-12. K600 OMNINET ADAPTER
HW1-13. K231 EXTERNAL 16-BIT PROCESSOR
HW1-14. K223 DLC INHOUSE INTERFACE

HW2-1. INTERFACE SELECTION
HW2-2. KEYBOARD INTERFACE AND KEYBOARD
HW2-3. 16-BIT PROCESSOR WITH P I C.

HW3-1. C3282 FIX DISK

In the K212 Serial Printer Interface and the K211 Communication Interface we use the Programmable Communication Interface Chip 2651 (NCR Part No. 006-1042033).
Programming look at following pages.

PROGRAMMABLE COMMUNICATIONS INTERFACE (PCI) 2651

2651-1

DESCRIPTION

The Signetics 2651 PCI is a universal synchronous/asychronous data communications controller chip designed for microcomputer systems. It interfaces directly to the Signetics 2650 microprocessor and may be used in a polled or interrupt driven system environment. The 2651 accepts programmed instructions from the microprocessor and supports many serial data communication disciplines, synchronous and asynchronous, in the full or half-duplex mode.

The PCI serializes parallel data characters received from the microprocessor for transmission. Simultaneously, it can receive serial data and convert it into parallel data characters for input to the microcomputer.

The 2651 contains a baud rate generator which can be programmed to either accept an external clock or to generate internal transmit or receive clocks. Sixteen different baud rates can be selected under program control when operating in the internal clock mode.

The PCI is constructed using Signetics n-channel silicon gate depletion load technology and is packaged in a 28-pin DIP.

FEATURES

- **Synchronous operation**
  - 5 to 8-bit characters
  - Single or double SYN operation
  - Internal character synchronization
  - Transparent or non-transparent mode
  - Automatic SYN or DLE-SYN insertion
  - SYN or DLE stripping
  - Odd, even, or no parity
  - Local or remote maintenance loop back mode
  - Baud rate: dc to 0.8M baud (1X clock)

- **Asynchronous operation**
  - 5 to 8-bit characters
  - 1, 1 1/2 or 2 stop bits
  - Odd, even, or no parity
  - Parity, overrun and framing error detection
  - Line break detection and generation
  - False start bit detection
  - Automatic serial echo mode
  - Local or remote maintenance loop back mode
  - Baud rate: dc to 0.8M baud (1X clock)
    - dc to 50k baud (16X clock)
    - dc to 12.5k baud (64X clock)

OTHER FEATURES

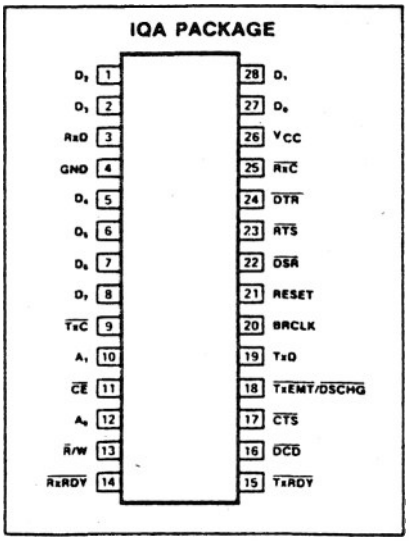
- Internal or external baud rate clock
- 16 internal rates-50 to 19,200 baud
- Double buffered transmitter and receiver
- Full or half duplex operation
-
- TTL compatible inputs and outputs
- Single 5V power supply
- No system clock required
- 28-pin dual in-line package

APPLICATIONS

- Intelligent terminals
- Network processors
- Front end processors
- Remote data concentrators
- Computer to computer links
- Serial peripherals

PIN CONFIGURATION

IQA PACKAGE

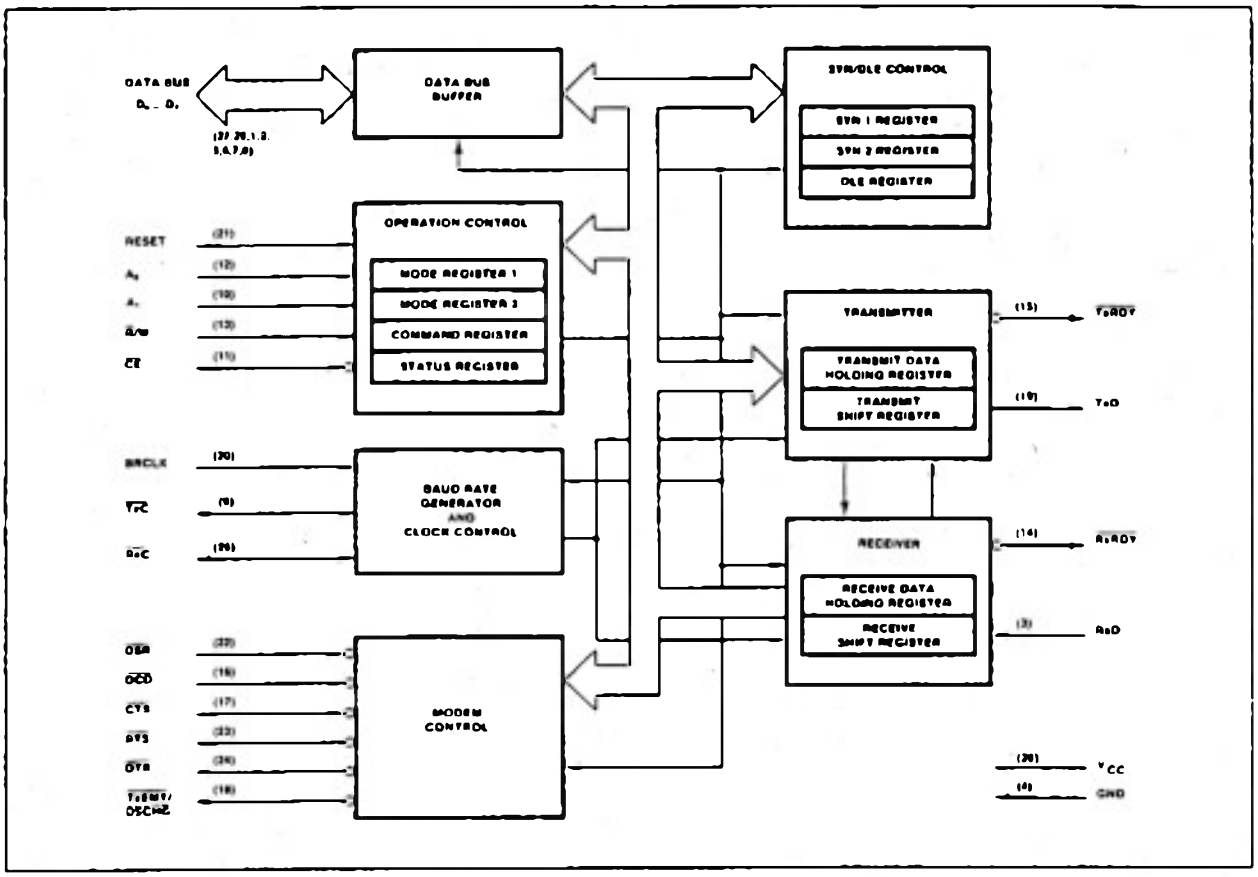


PIN DESIGNATION

PIN NO.	SYMBOL	NAME AND FUNCTION	TYPE
27,28,1,2, 5-8	D_0-D_7	8-bit data bus	I/O
21	RESET	Reset	I
12,10	A_0-A_1	Internal register select lines	I
13	R/W	Read or write command	I
11	CE	Chip enable input	I
22	DSR	Data set ready	I
24	DTR	Data terminal ready	O
23	RTS	Request to send	O
17	CTS	Clear to send	I
16	DCD	Data carrier detected	I
18	TxEMT/DSCHG	Transmitter empty or data set change	O
9	TxC	Transmitter clock	I/O
25	RxC	Receiver clock	I/O
19	TxD	Transmitter data	O
3	RxD	Receiver data	I
15	TxRDY	Transmitter ready	O
14	RxRDY	Receiver ready	O
20	BRCLK	Baud rate generator clock	I
26	V_{CC}	+5V supply	I
4	GND	Ground	I

PROGRAMMABLE COMMUNICATIONS INTERFACE (PCI) 2651

BLOCK DIAGRAM



BLOCK DIAGRAM

The PCI consists of six major sections. These are the transmitter, receiver, timing, operation control, modem control and SYN/DLE control. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the microprocessor data bus via a data bus buffer.

Operation Control

This functional block stores configuration and operation commands from the CPU and generates appropriate signals to various internal sections to control the overall device operation. It contains read and write circuits to permit communications with the microprocessor via the data bus and contains Mode Registers 1 and 2, the Command Register, and the Status Register. Details of register addressing and protocol are presented in the PCI Programming section of this data sheet.

Timing

The PCI contains a Baud Rate Generator (BRG) which is programmable to accept external transmit or receive clocks or to divide an external clock to perform data communications. The unit can generate 16 commonly used baud rates, any one of which can be selected for full duplex operation. See Table 1.

Receiver

The Receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU.

Transmitter

The Transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin.

Modem Control

The modem control section provides interfacing for three input signals and three output signals used for "handshaking" and status indication between the CPU and a modem.

SYN/DLE Control

This section contains control circuitry and three 8-bit registers storing the SYN1, SYN2, and DLE characters provided by the CPU. These registers are used in the synchronous mode of operation to provide the characters required for synchronization, idle fill and data transparency.

INTERFACE SIGNALS

The PCI interface signals can be grouped into two types: the CPU-related signals (shown in Table 2), which interface the 2651 to the microprocessor system, and the device-related signals (shown in Table 3), which are used to interface to the communications device or system.

OPERATION

The functional operation of the 2651 is programmed by a set of control words supplied by the CPU. These control words specify items such as synchronous or asynchronous mode, baud rate, number of bits per character, etc. The programming procedure is described in the PCI Programming section of this data sheet.

After programming, the PCI is ready to perform the desired communications functions. The receiver performs serial to parallel conversion of data received from a modem or equivalent device. The transmitter converts parallel data received from the CPU to a serial bit stream. These actions are accomplished within the framework specified by the control words.

Receiver

The 2651 is conditioned to receive data when the $\overline{DCD}$ input is low and the RxEN bit in the command register is true. In the asynchronous mode, the receiver looks for a high to low transition of the start bit on the RxD input line. If a transition is detected, the state of the RxD line is sampled again after a delay of one-half of a bit time. If RxD is now high, the search for a valid start bit is begun again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input line at one bit time intervals until the proper number of data bits, the parity bit, and the stop bit(s) have been assembled. The data is then transferred to the Receive Data Holding Register, the RxRDY bit in the status register is set, and the $\overline{RxRDY}$ output is asserted. If the character length is less than 8 bits, the high order unused bits in the Holding Register are set to zero. The Parity Error, Framing Error, and Overrun Error status bits are set if required. If a break condition is detected (RxD is low for the entire character as well as the stop bit [s]), only one character consisting of all zeros (with the FE status bit set) will be transferred to the Holding Register. The RxD input must return to a high condition before a search for the next start bit begins.

When the PCI is initialized into the synchronous mode, the receiver first enters the hunt mode. In this mode, as data is shifted into the Receiver Shift Register a bit at a time, the contents of the register are compared to the contents of the SYN1 register. If the two are not equal, the next bit is shifted in and the comparison is repeated. When the two registers match, the hunt mode is terminated and character assembly mode begins. If single SYN operation is programmed, the SYN DETECT status bit is set. If double SYN operation is programmed, the first character assembled after SYN1 must be SYN2 in order for the SYN DETECT bit to be set. Otherwise, the PCI returns to the hunt mode. (Note that the sequence SYN1-SYN1-SYN2 will not achieve synchronization). When synchronization has been achieved, the PCI continues to assemble characters and transfer them to the Holding Register, setting the RxRDY status bit and asserting the $\overline{RxRDY}$ output each time a character is transferred. The PE and OE status bits are set as appropriate. Further receipt of the appropriate SYN sequence sets the SYN DETECT status bit. If the SYN stripping mode is commanded, SYN characters are not transferred to the Holding Register. Note that the SYN characters used to establish initial synchronization are not transferred to the Holding Register in any case.

Transmitter

The PCI is conditioned to transmit data when the $\overline{CTS}$ input is low and the TxEN command register bit is set. The 2651 indicates to the CPU that it can accept a character for transmission by setting the TxRDY status bit and asserting the $\overline{TxRDY}$ output. When the CPU writes a character into the Transmit Data Holding Register, these conditions are negated. Data is transferred from the Holding Register to the Transmit Shift Register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again. Thus, one full character time of buffering is provided.

In the asynchronous mode, the transmitter automatically sends a start bit followed by the programmed number of data bits, the least significant bit being sent first. It then appends an optional odd or even parity bit and the programmed number of stop bits. If, following transmission of the stop bits, a new character is not available in the Transmit Holding Register, the TxD output remains in the marking (high) condition and the $\overline{TxEMT/DSCHG}$ output and its corresponding status bit are asserted. Transmission resumes when the CPU loads a new character into the Holding Register. The transmitter can be forced to output a continuous low (BREAK) condition by setting the Send Break command bit high.

In the synchronous mode, when the 2651 is initially conditioned to transmit, the TxD output remains high and the TxRDY condition is asserted until the first character to be transmitted (usually a SYN character) is loaded by the CPU. Subsequent to this, a continuous stream of characters is transmitted. No extra bits (other than parity, if commanded) are generated by the PCI unless the CPU fails to send a new character to the PCI by the time the transmitter has completed sending the previous character. Since synchronous communications does not allow gaps between characters, the PCI asserts TxEMT and automatically "fills" the gap by transmitting SYN1s, SYN1-SYN2 doublets, or DLE-SYN1 doublets, depending on the command mode. Normal transmission of the message resumes when a new character is available in the Transmit Data Holding Register. If the SEND DLE bit in the command register is true, the DLE character is automatically transmitted prior to transmission of the message character.

PCI PROGRAMMING

Prior to initiating data communications, the 2651 operational mode must be programmed by performing write operations to the mode and command registers. In addition, if synchronous operation is programmed, the appropriate SYN/DLE registers must be loaded. The PCI can be reconfigured at any time during program execution. However, the receiver and transmitter should be disabled if the change has an effect on the reception or transmission of a character. A flowchart of the initialization process appears in Figure 1.

The internal registers of the PCI are accessed by applying specific signals to the $\overline{CE}$, $\overline{R/W}$, A_1 and A_0 inputs. The conditions necessary to address each register are shown in Table 4.

The SYN1, SYN2, and DLE registers are accessed by performing write operations with the conditions $A_1 = 0$, $A_0 = 1$, and $\overline{R/W} = 1$. The first operation loads the SYN1 register. The next loads the SYN2 register, and the third loads the DLE register. Reading or loading the mode registers is done in a similar manner. The first write (or read) operation addresses Mode Register 1, and a subsequent operation addresses Mode Register 2. If more than the required number of accesses are made, the internal sequencer recycles to point at the first register. The pointers are reset to SYN1 Register and Mode Register 1 by a RESET input or by performing a "Read Command Register" operation, but are unaffected by any other read or write operation.

The 2651 register formats are summarized in Tables 5, 6, 7 and 8. Mode Registers 1 and 2 define the general operational characteristics of the PCI, while the Command Register controls the operation within this basic frame-work. The PCI indicates its status in the Status Register. These registers are cleared when a RESET input is applied.

Command Register (CR)

Table 7 illustrates Command Register. Bits CR0 (TxEN) and CR2 (RxEN) enable or disable the transmitter and receiver respectively. If the transmitter is disabled, it will complete the transmission of the character in the Transmit Shift Register (if any) prior to terminating operation. The TxD output will then remain in the marking state (high). If the receiver is disabled, it will terminate operation immediately. Any character being assembled will be neglected.

Bits CR1 (DTR) and CR5 (RTS) control the $\overline{\text{DTR}}$ and $\overline{\text{RTS}}$ outputs. Data at the outputs is the logical complement of the register data.

In asynchronous mode, setting CR3 will force and hold the TxD output low (spacing condition) at the end of the current transmitted character. Normal operation resumes when CR3 is cleared. The TxD line will go high for a least one bit time before beginning transmission of the next character in the Transmit Data Holding Register. In synchronous mode, setting CR3 causes the transmission of the DLE register contents prior to sending the character in the Transmit Data Holding Register. CR3 should be reset in response to the next TxRDY.

Setting CR4 causes the error flags in the Status Register (SR3, SR4, and SR5) to be cleared. This bit resets automatically.

The PCI can operate in one of four submodes within each major mode (synchronous or asynchronous). The operational sub-mode is determined by CR7 and CR6. CR7-CR6 = 00 is the normal mode, with the transmitter and receiver operating independently in accordance with the Mode and Status Register instructions.

In asynchronous mode, CR7-CR6 = 01 places the PCI in the Automatic Echo mode. Clocked, regenerated received data is automatically directed to the TxD line while normal receiver operation continues. The receiver must be enabled (CR2 = 1), but the transmitter need not be enabled. CPU to receiver communications continues normally, but the CPU to transmitter link is disabled. Only the first character of a break condition is echoed. The TxD output will go high until the next valid start is detected. The following conditions are true while in Automatic Echo mode:

1. Data assembled by the receiver is automatically placed in the Transmit Holding Register and retransmitted by the transmitter on the TxD output.
2. Transmit clock = receive clock.
3. $\overline{\text{TxRDY}}$ output = 1.
4. The $\overline{\text{TxEMT/DSCHG}}$ pin will reflect only the data set change condition.
5. The TxEN command (CR0) is ignored.

Mode Register 1 (MR1)

Table 5 illustrates Mode Register 1. Bits MR11 and MR10 select the communication format and baud rate multiplier. 00 specifies synchronous mode and 1X multiplier. 1X, 16X, and 64X multipliers are programmable for asynchronous format. However, the multiplier in asynchronous format applies only if the external clock input option is selected by MR24 or MR25.

MR13 and MR12 select a character length of 5, 6, 7, or 8 bits. The character length does not include the parity bit, if programmed, and does not include the start and stop bits in asynchronous mode.

MR14 controls parity generation. If enabled, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR15 selects odd or even parity when parity is enabled by MR14.

In asychronous mode, MR17 and MR16 select character framing of 1, 1.5, or 2 stop bits. (If 1X baud rate is programmed, 1.5 stop bits defaults to 1 stop bits on transmit). In synchronous mode, MR17 controls the number of SYN characters used to establish synchronization and for character fill when the transmitter is idle. SYN1 alone is used if MR17 = 1, and SYN1-SYN2 is used when MR17 = 0. If the transparent mode is specified by MR16, DLE-SYN1 is used for character fill, but the normal synchronization sequence is used.

Mode Register 2 (MR2)

Table 6 illustrates Mode Register 2. MR23, MR22, MR21, and MR20 control the frequency of the internal baud rate generator (BRG). Sixteen rates are selectable. When driven by a 5.0688 MHz input at the BRCLK input (pin 20), the BRG output has zero error except at 134.5, 2000, and 19,200 baud, which have errors of +0.016%, +0.235%, and +3.125% respectively.

MR25 and MR24 select either the BRG or the external inputs $\overline{\text{TxC}}$ and $\overline{\text{RxC}}$ as the clock source for the transmitter and receiver, respectively. If the BRG clock is selected, the baud rate factor in asynchronous mode is 16X regardless of the factor selected by MR11 and MR10. In addition, the corresponding clock pin provides an output at 1X the baud rate.

In synchronous mode, CR7-CR6 = 01 places the PCI in the Automatic SYN/DLE Stripping mode. The exact action taken depends on the setting of bits MR17 and MR16:

1. In the non-transparent, single SYN mode (MR17-MR16 = 10), characters in the data stream matching SYN1 are not transferred to the Receive Data Holding Register (RHR).
2. In the non-transparent, double SYN mode (MR17-MR16 = 00), characters in the data stream matching SYN1, or SYN2 if immediately preceded by SYN1, are not transferred to the RHR. However, only the first SYN1 of an SYN1-SYN1 pair is stripped.
3. In transparent mode (MR16 =1), characters in the data stream matching DLE, or SYN1 if immediately preceded by DLE, are not transferred to the RHR. However, only the first DLE of a DLE-DLE pair is stripped.

Note that Automatic Stripping mode does not affect the setting of the DLE Detect and SYN Detect status bits (SR3 and SR5).

Two diagnostic sub-modes can also be configured. In Local Loop Back mode (CR7-CR6 = 10), the following loops are connected internally:

1. The transmitter output is connected to the receiver input.
2. $\overline{DTR}$ is connected to $\overline{DCD}$ and $\overline{RTS}$ is connected to $\overline{CTS}$.
3. Receive clock = transmit clock.
4. The $\overline{DTR}$, $\overline{RTS}$ and TxD outputs are held high.
5. The $\overline{CTS}$, $\overline{DCD}$, $\overline{DSR}$ and RxD inputs are ignored.

Additional requirements to operate in the Local Loop Back mode are that CR0 (TxEN), CR1 (DTR), and CR5 (RTS) must be set to 1. CR2 (RxEN) is ignored by the PCI.

The second diagnostic mode is the Remote Loop Back mode (CR7-CR6 = 11). In this mode:

1. Data assembled by the receiver is automatically placed in the Transmit Holding Register and retransmitted by the transmitter on the TxD output.
2. Transmit clock = receive clock.
3. No data is sent to the local CPU, but the error status conditions (PE, OE, FE) are set.
4. The $\overline{RxRDY}$, $\overline{TxRDY}$, and $\overline{TxEMT/DSCHG}$ outputs are held high.
5. CR1 (TxEN) is ignored.
6. All other signals operate normally.

Status Register

The data contained in the Status Register (as shown in Table 8) indicate receiver and transmitter conditions and modem/data set status.

SR0 is the Transmitter Ready (TxRDY) status bit. It, and its corresponding output, are valid only when the transmitter is enabled. If equal to 0, it indicates that the Transmit Data Holding Register has been loaded by the CPU and the data has not been transferred to the Transmit Shift Register. If set equal to 1, it indicates that the Holding Register is ready to accept data from the CPU. This bit is initially set when the Transmitter is enabled by CR0, unless a character has previously been loaded into the Holding Register. It is not set when the Automatic Echo or Remote Loop Back modes are programmed. When this bit is set, the $\overline{TxRDY}$ output pin is low. In the Automatic Echo and Remote Loop Back modes, the output is held high.

SR1, the Receiver Ready (RxRDY) status bit, indicates the condition of the Receive Data Holding Register. If set, it indicates that a character has been loaded into the Holding Register from the Receive Shift Register and is ready to be read by the CPU. If equal to zero, there is no new character in the Holding Register. This bit is cleared when the CPU reads the Receive Data Holding Register or when the receiver is disabled by CR2. When set, the $\overline{RxRDY}$ output is low.

The TxEMT/DSCHG bit, SR2, when set, indicates either a change of state of the $\overline{DSR}$ or $\overline{DCD}$ inputs or that the Transmit Shift Register has completed transmission of a character and no new character has been loaded into the Transmit Data Holding Register. Note that in synchronous mode this bit will be set even though the appropriate "fill" character is transmitted. It is cleared when the transmitter is enabled by CR0 and does not indicate transmitter condition until at least one character is transmitted. It is also cleared when the Status Register is read by the CPU. When SR2 is set, the $\overline{TxEMT}$/$\overline{DSCHG}$ output is low.

SR3, when set, indicates a received parity error when parity is enabled by MR14. In synchronous transparent mode (MR16 = 1), with parity disabled, it indicates that a character matching the DLE Register has been received. However, only the first DLE of two successive DLEs will set SR3. This bit is cleared when the receiver is disabled and by the Reset Error command, CR4.

The Overrun Error status bit, SR4, indicates that the previous character loaded into the Receive Holding Register was not read by the CPU at the time a new received character was transferred into it. This bit is cleared when the receiver is disabled and by the Reset Error command, CR4.

In asynchronous mode, bit SR5 signifies that the received character was not framed by the programmed number of stop bits. (If 1.5 stop bits are programmed, only the first stop bit is checked.) In synchronous non-transparent mode (MR16 = 0), it indicates receipt of the SYN1 character in single SYN mode or the SYN1-SYN2 pair in double SYN mode. In synchronous transparent mode (MR16 = 1), this bit is set upon detection of the initial synchronizing characters (SYN1 or SYN1-SYN2) and, after synchronization has been achieved, when a DLE-SYN1 pair is received. The bit is reset when the receiver is disabled, when the Reset Error command is given in asynchronous mode, and when the Status Register is read by the CPU in the synchronous mode.

SR6 and SR7 reflect the conditions of the $\overline{DCD}$ and $\overline{DSR}$ inputs respectively. A low input sets its corresponding status bit and a high input clears it.

2651 INITIALIZATION FLOW CHART

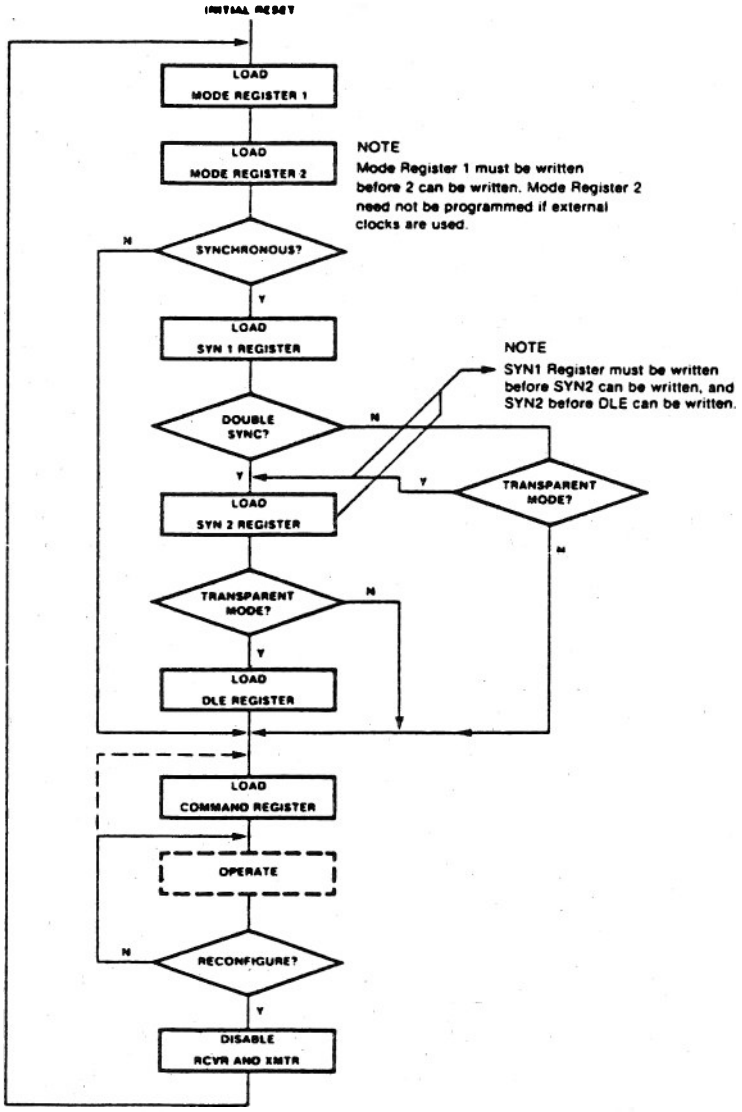
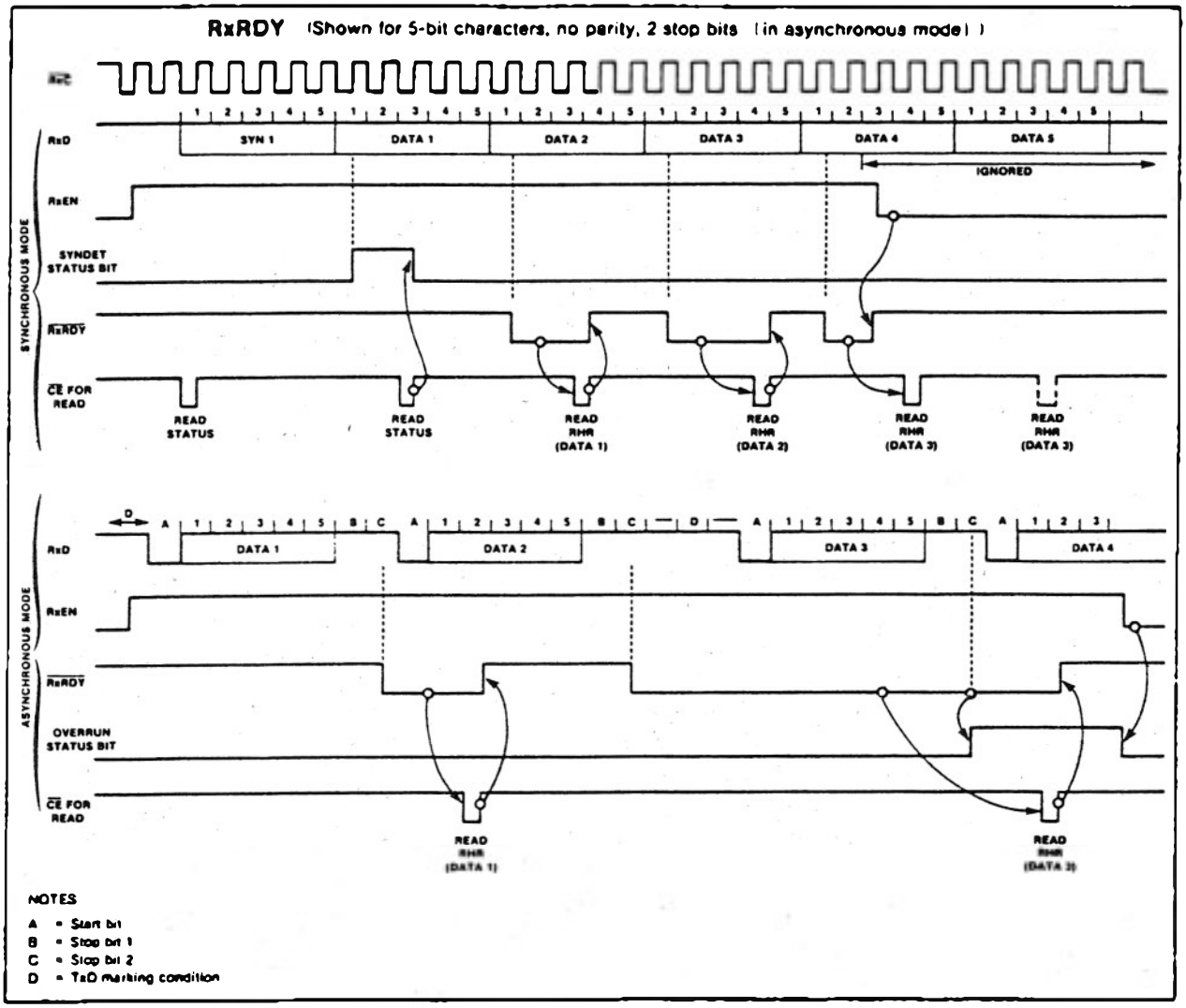


Figure 1

TIMING DIAGRAMS (Cont'd)

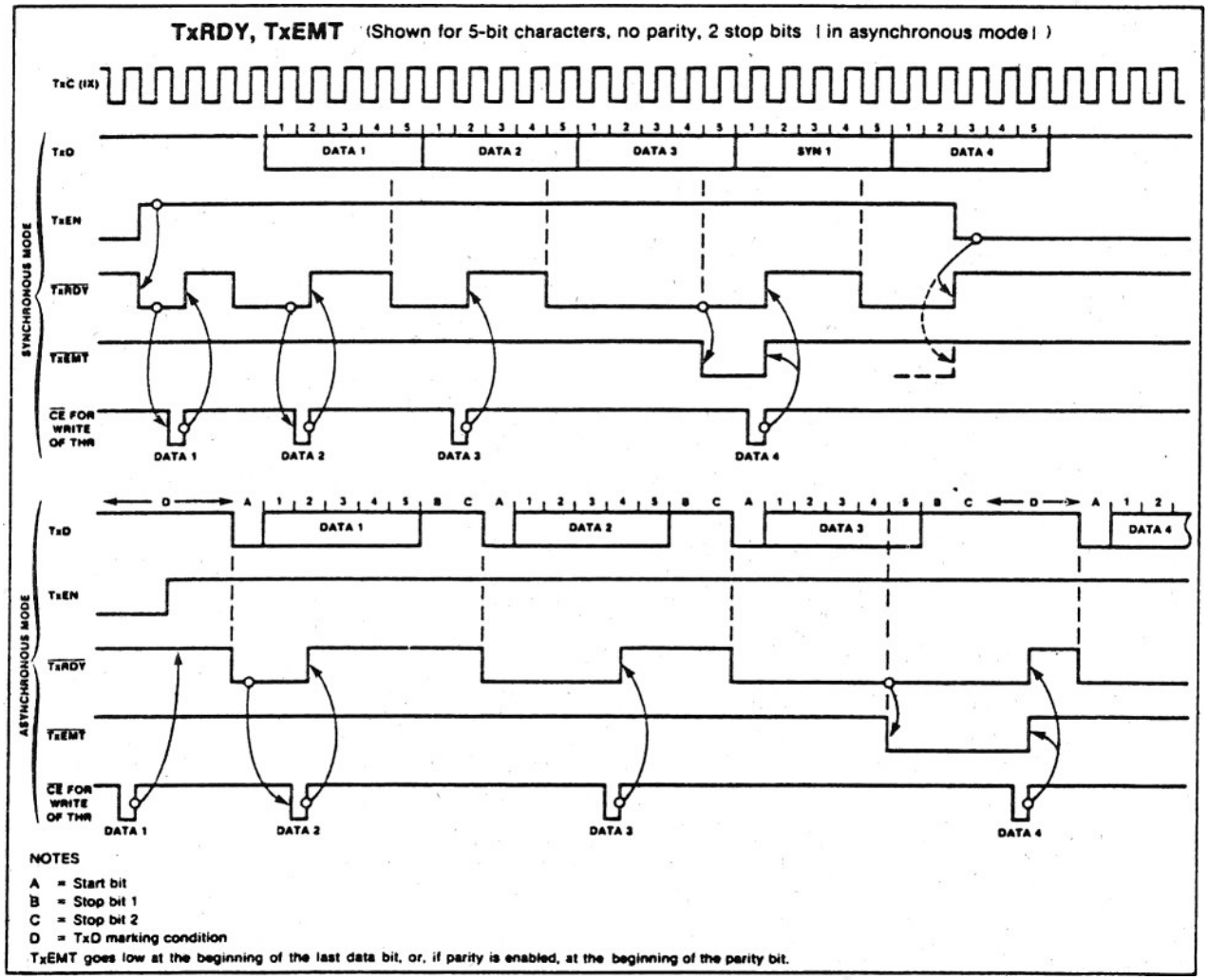
RxRDY (Shown for 5-bit characters, no parity, 2 stop bits (in asynchronous mode))



NOTES
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition

TIMING DIAGRAMS

TxRDY, TxEMT (Shown for 5-bit characters, no parity, 2 stop bits (in asynchronous mode))



NOTES

A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition

TxEMT goes low at the beginning of the last data bit, or, if parity is enabled, at the beginning of the parity bit.

MODE REGISTER 1 (MR1)

MR17	MR16	MR15	MR14	MR13	MR12	MR11	MR10
		Parity Type	Parity Control	Character Length		Mode and Baud Rate Factor	
ASYNCH: STOP BIT LENGTH 00 = INVALID 01 = 1 STOP BIT 10 = 1½ STOP BITS 11 = 2 STOP BITS		0 = ODD 1 = EVEN	0 = DISABLED 1 = ENABLED	00 = 5 BITS 01 = 6 BITS 10 = 7 BITS 11 = 8 BITS		00 = SYNCHRONOUS 1X RATE 01 = ASYNCHRONOUS 1X RATE 10 = ASYNCHRONOUS 16X RATE 11 = ASYNCHRONOUS 64X RATE	
SYNCH: NUMBER OF SYN CHAR 0 = DOUBLE SYN 1 = SINGLE SYN	SYNCH: TRANSPARENCY CONTROL 0 = NORMAL 1 = TRANSPARENT						

NOTE
Baud rate factor in asynchronous applies only if external clock is selected. Factor is 16X if internal clock is selected.

MODE REGISTER 2 (MR2)

MR27	MR26	MR25	MR24	MR23	MR22	MR21	MR20
		Transmitter Clock	Receiver Clock	Baud Rate Selection			
Ø	Ø	0 = EXTERNAL 1 = INTERNAL	0 = EXTERNAL 1 = INTERNAL	0000 = 50 BAUD 0001 = 75 0010 = 110 0011 = 134.5 0100 = 150 0101 = 300 0110 = 600 0111 = 1200		1000 = 1800 BAUD 1001 = 2000 1010 = 2400 1011 = 3600 1100 = 4800 1101 = 7200 1110 = 9600 1111 = 19,200	

COMMAND REGISTER (CR)

CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
Operating Mode		Request to Send	Reset Error		Receive Control (RxEN)	Data Terminal Ready	Transmit Control (TxEN)
00 = NORMAL OPERATION 01 = ASYNCH: AUTOMATIC ECHO MODE SYNCH: SYN AND/OR DLE STRIPPING MODE 10 = LOCAL LOOP BACK 11 = REMOTE LOOP BACK		0 = FORCE $\overline{RTS}$ OUTPUT HIGH 1 = FORCE $\overline{RTS}$ OUTPUT LOW	0 = NORMAL 1 = RESET ERROR FLAG IN STATUS REG (FE, OE, PE/DLE DETECT)	ASYNCH: FORCE BREAK 0 = NORMAL 1 = FORCE BREAK SYNCH: SEND DLE 0 = NORMAL 1 = SEND DLE	0 = DISABLE 1 = ENABLE	0 = FORCE $\overline{DTR}$ OUTPUT HIGH 1 = FORCE $\overline{DTR}$ OUTPUT LOW	0 = DISABLE 1 = ENABLE

STATUS REGISTER (SR)

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
Data Set Ready	Data Carrier Detect	FE/SYN Detect	Overrun	PE/DLE Detect	TxEMT/DSCHG	RxRDY	TxRDY
0 = $\overline{DSR}$ INPUT IS HIGH 1 = $\overline{DSR}$ INPUT IS LOW	0 = $\overline{DCD}$ INPUT IS HIGH 1 = $\overline{DCD}$ INPUT IS LOW	ASYNCH: 0 = NORMAL 1 = FRAMING ERROR SYNCH: 0 = NORMAL 1 = SYN CHAR DETECTED	0 = NORMAL 1 = OVERRUN ERROR	ASYNCH: 0 = NORMAL 1 = PARITY ERROR SYNCH: 0 = NORMAL 1 = PARITY ERROR OR DLE CHAR RECEIVED	0 = NORMAL 1 = CHANGE IN $\overline{DSR}$ OR $\overline{DCD}$, OR TRANSMIT SHIFT REGISTER IS EMPTY	0 = RECEIVE HOLDING REG EMPTY 1 = RECEIVE HOLDING REG HAS DATA	0 = TRANSMIT HOLDING REG BUSY 1 = TRANSMIT HOLDING REG EMPTY

2651 REGISTER ADDRESSING

K212	K211	$\overline{CE}$	A_1	A_0	$\overline{R}/W$	FUNCTION
—	—	1	X	X	X	Tri-state data bus
60H	70H	0	0	0	0	Read receive holding register
64H	74H	0	0	0	1	Write transmit holding register
61H	71H	0	0	1	0	Read status register
65H	75H	0	0	1	1	Write SYN1/SYN2/DLE registers
62H	72H	0	1	0	0	Read mode registers 1/2
66H	76H	0	1	0	1	Write mode registers 1/2
63H	73H	0	1	1	0	Read command register
67H	77H	0	1	1	1	Write command register

Use IN; OUT opcodes by Z80, 8088

K801 implements all other port-addresses, as well.
(Compare: Interface Selection)

Module 2661 is used in K801. By assynchron same programming

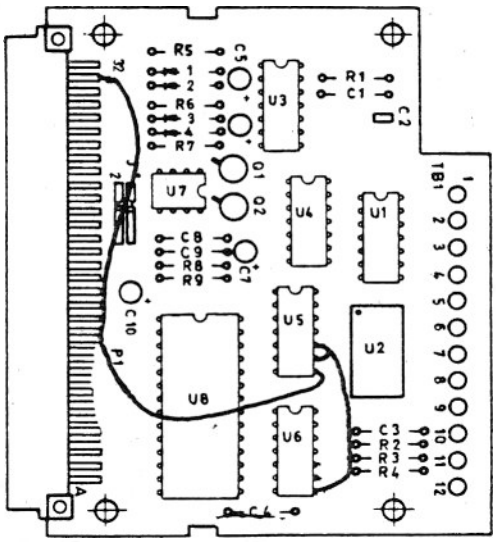
Pin: Clear-To-Send (CTS) will be recognizes more exactly

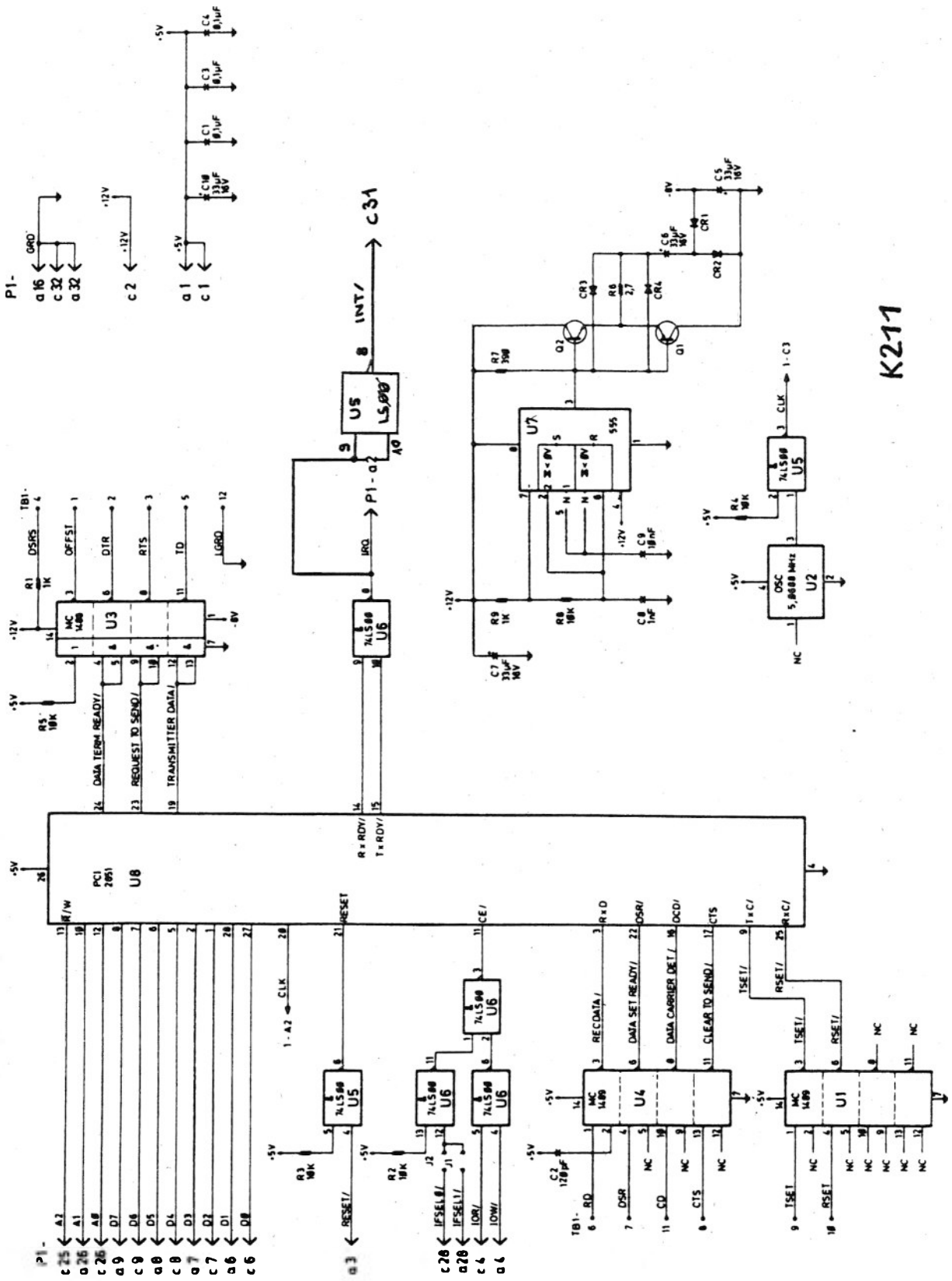
Compare: Hardware-record for printer will be o.k.

Interrupt for Z80 - K211

1. Simple modification for one interrupt in the system. (Bus)

RS-232C ADAPTER

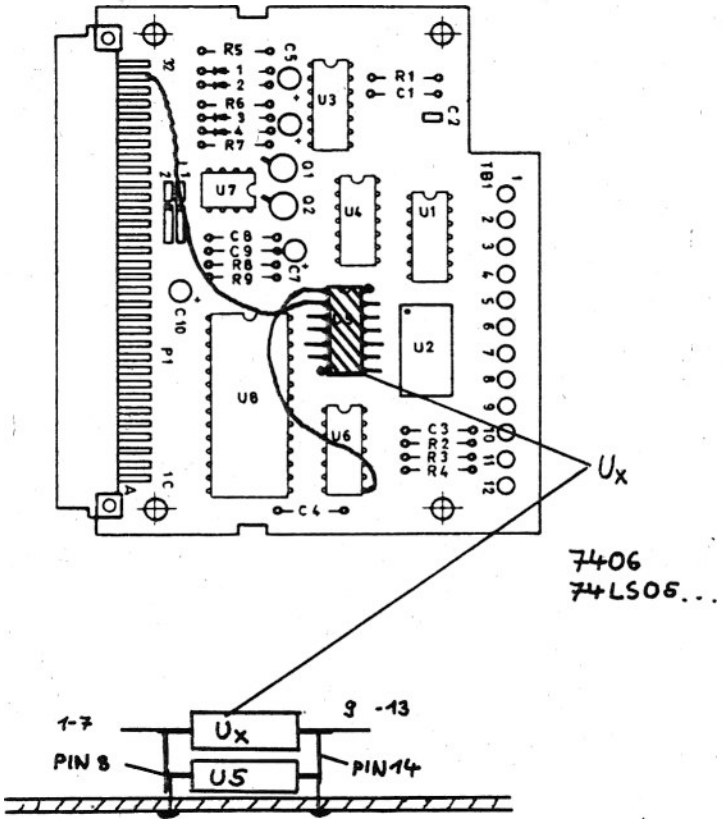




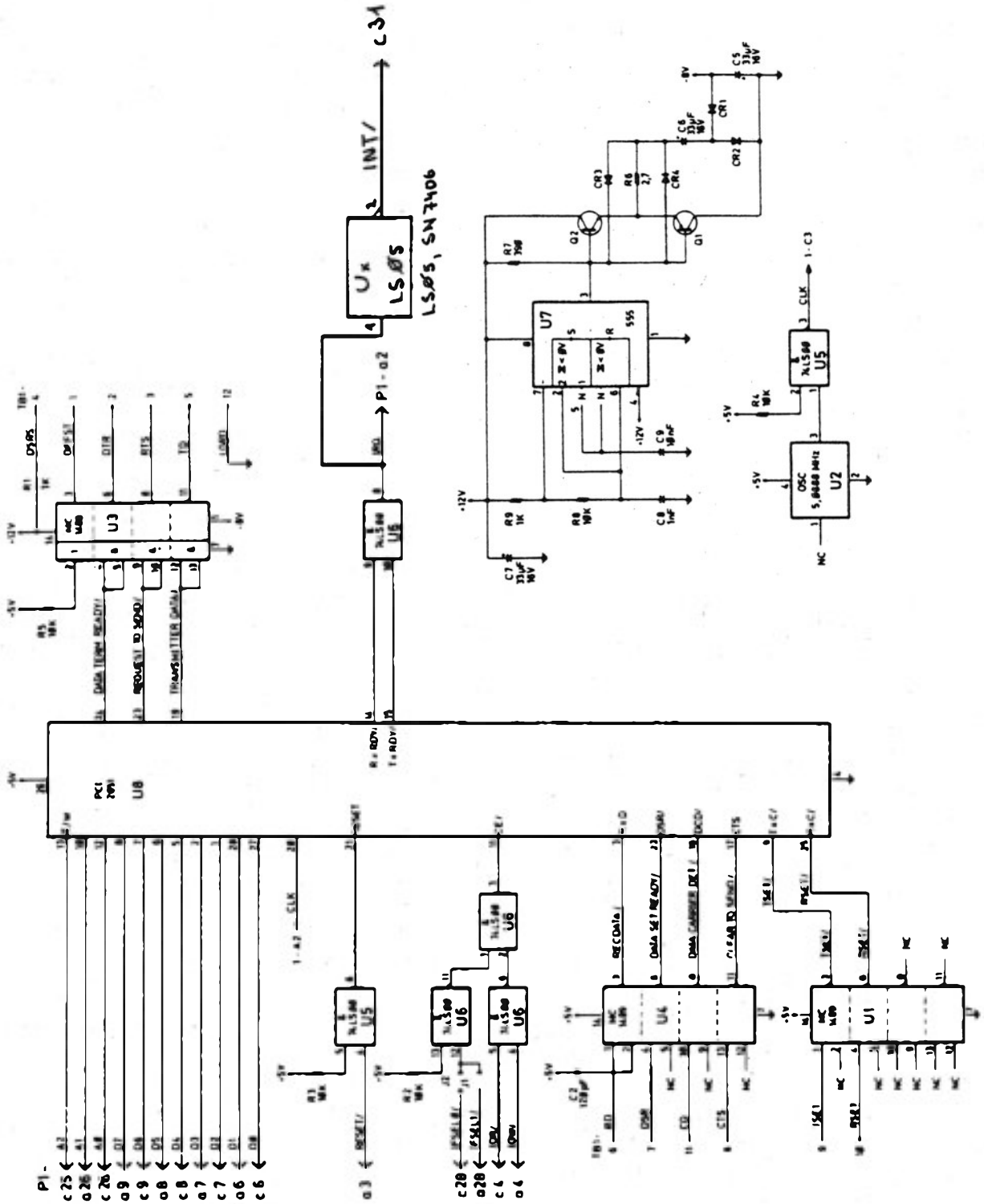
If you have another interrupt interface on the bus, you may destroy the IC U5.

2. Modification for more interrupt interfaces

RS-232C ADAPTER



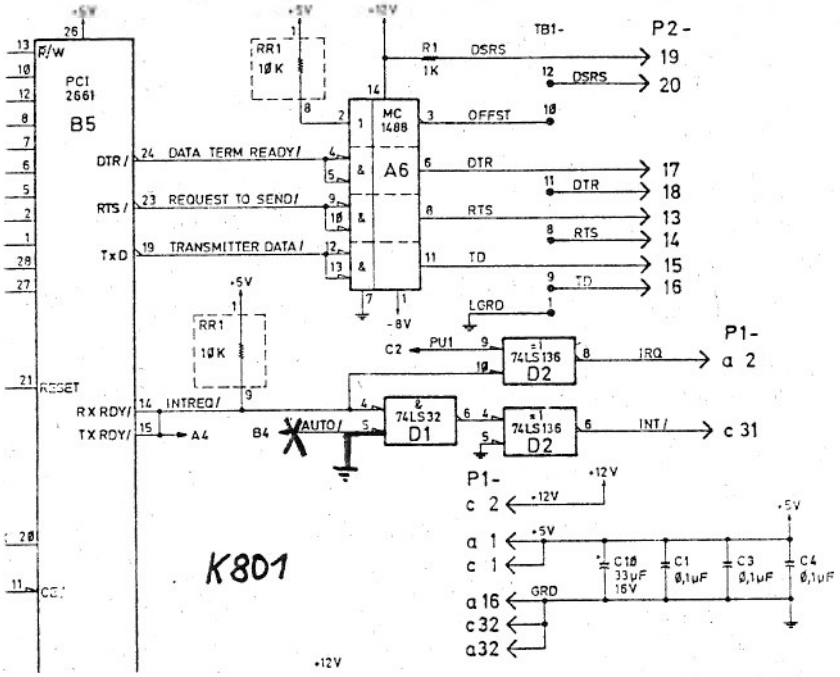
You can use this interrupt only if you have a special application. For other applications you can´t use this interface.



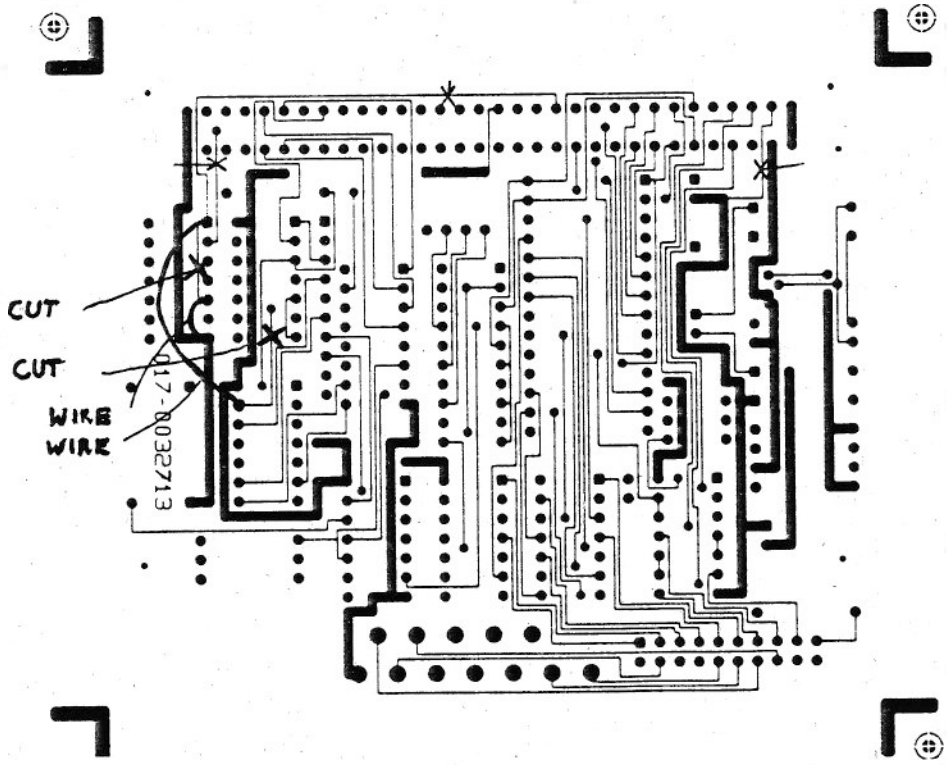
K801

Interrupt for Z80 - K801

modification for Interrupt in the system. (Bus)



LÖTSEITE - SOLDERSIDE



You can use this interrupt only if you have a special application.
For other applications you can't use this Interface.

Do not use with K235

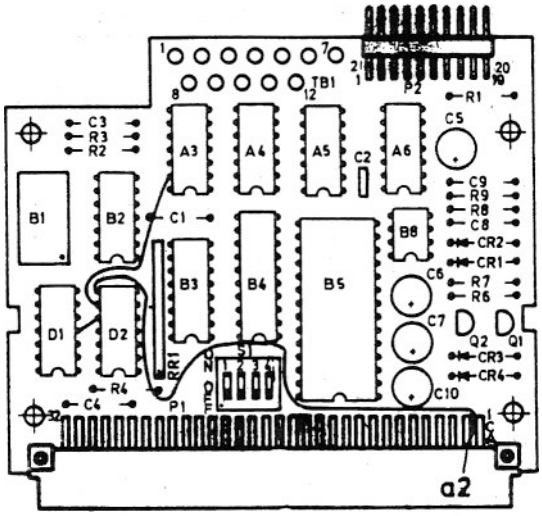
ÜBERSICHT der Peripherie-Kits für DMV
Overview of peripheral kits

Kit-Nr. / Kit-No.	Beschreibung / Description
C3282-101	Freistehende Festplatte, 10MB, nicht erweiterbar Winchester disk drive, not additional
C3282-102	Freistehende Festplatte, 10MB, erweiterbar Winchester disk drive, additional with -103
C3282-103	Freistehende Festplatte, 10MB, für -102 Winchester disk drive, additional for -102
K 018	Erweiterung auf 2. Flexdisk-Laufwerk Second flexible disk upgrade
K 200	64 auf 128 kB Speichererweiterung memory upgrade
K 202	64 auf 256 kB Speichererweiterung memory upgrade
K 208	64 auf 512 kB Speichererweiterung memory upgrade
K 210	Centronics Interface centronics parallel I/F
K 211	RS-232C Interface, Modem-Version serial I/F
K 212	RS-232C Interface, Drucker-Version serial I/F
K 213	RS-232C Interface, Plotter-Version serialI/F
K 214	Leer-Kit mit Platine und Busstecker blank interface adapter & bus connector
K 215	RS-232C, synchron/asynchron, gepuffert buffered sync/async RS-232C adapter
K 216	SCC, 2-fach RS-232C SCC communication adapter
K 219	Integriertes Modem(nur USA) integrated modem(only USA)
K 220	Diagnose-Modul diagnostic module
K 223	DLC Einbauinterface DLC inhouse I/F adapter
K 225	Diagnose-Diskette diagnostic diskette
K 231	8/16-bit Prozessor-Erweiterung(Einsteckmodul) dual 8/16-bit processor upgrade
K 232	Arithmetik-Coprozessor(8087) numeric coprocessor
K 233	Gemeinsames RAM, Modul 16kB shared RAM cartridge
K 234	68008-Prozessor-Modul(32Bit CPU) processor board

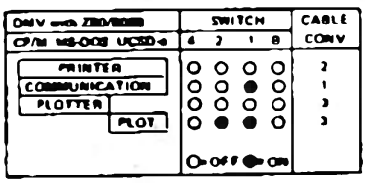
Kit-Nr. Kit-No.	Beschreibung Description
K 235	16-bit Prozessor mit Interrupt(intern) 16-bit processor with interr. controller(inhouse)
K 240	Kippvorrichtung für DMV tilt device
K 600	Omninet-Transporter Interface omninet transporter
K 801	RS-232C, programmierbar switchable RS-232C adapter
K 803	Echtzeit-Uhr real-time-clock
K 804	IEEE-488(IEC-625)-Interface IEEE-488 adapter
K 806	Maus-Interface mouse adapter
K 880	Einschubverriegelung cartridge lock

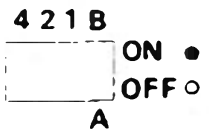
K801

SWITCHABLE RS-232 ADAPTER
(K801)



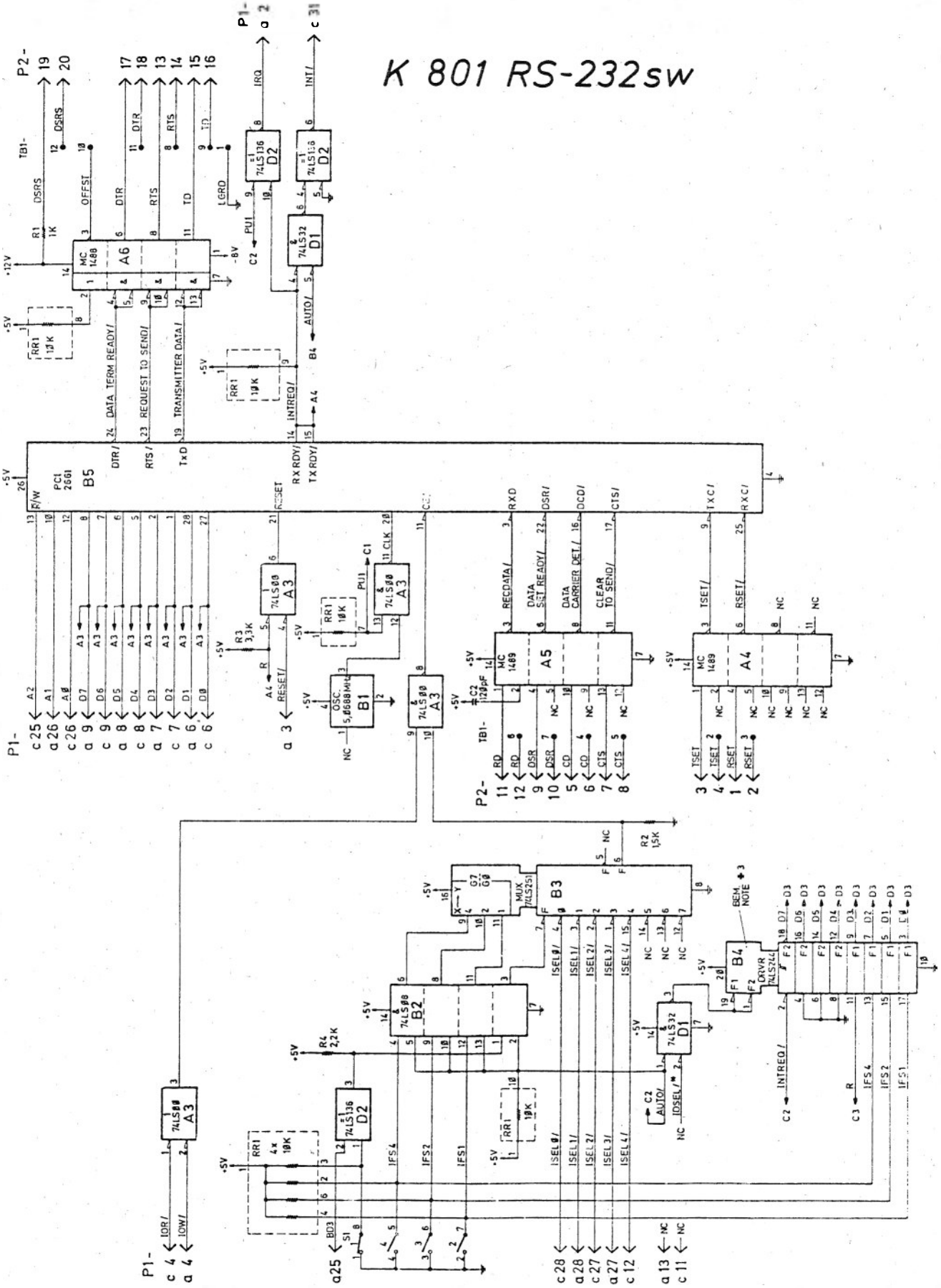
IFSEL	SWITCH 4 2 1 B	PORT-ADDR.
0 A	○ ○ ○ ○	60H – 67H
0 B	○ ○ ○ ●	68H – 6FH
1 A	○ ○ ● ○	70H – 77H
1 B	○ ○ ● ●	78H – 7FH
2 A	○ ● ○ ○	30H – 37H
2 B	○ ● ○ ●	38H – 3FH
3 A	○ ● ● ○	B0H – B7H
3 B	○ ● ● ●	B8H – BFH
4 A	● ○ ○ ○	C0H – C7H
4 B	● ○ ○ ●	C8H – CFH



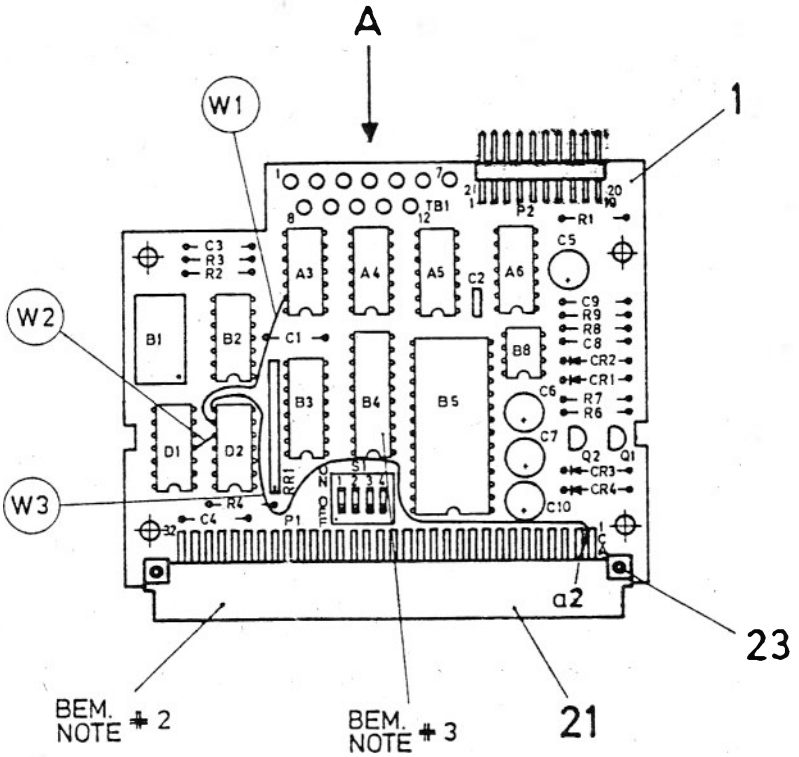


IF/ID SELECTION

K 801 RS-232sw



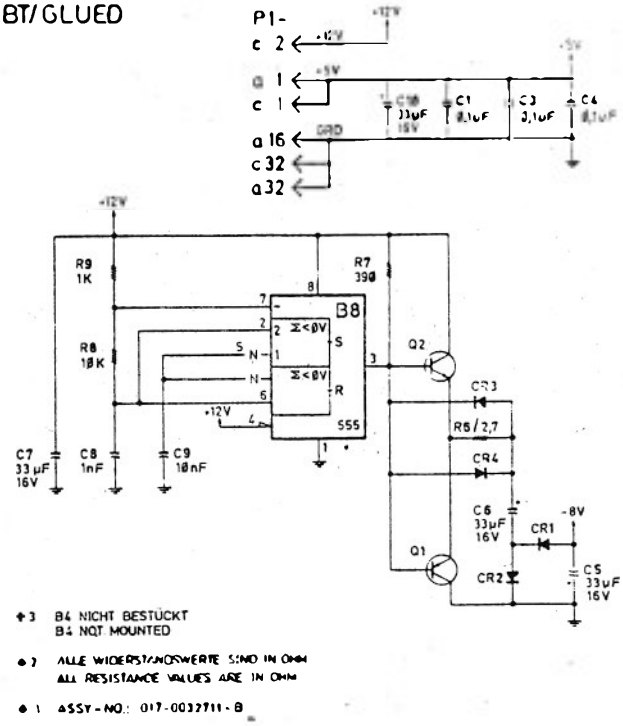
K 801 SWITCH RS-232C



◆ 3 B4 NICHT BESTÜCKT
B4 NOT MOUNTED

2 LABEL „ASSY 017-0032711 - A" AUFGEKLEBT/ GLUED
LABEL „SCHM 017-0032712 - A" AUFGEKLEBT/ GLUED

1 SCHEMATIC - NO.: 017- 0032712 - A



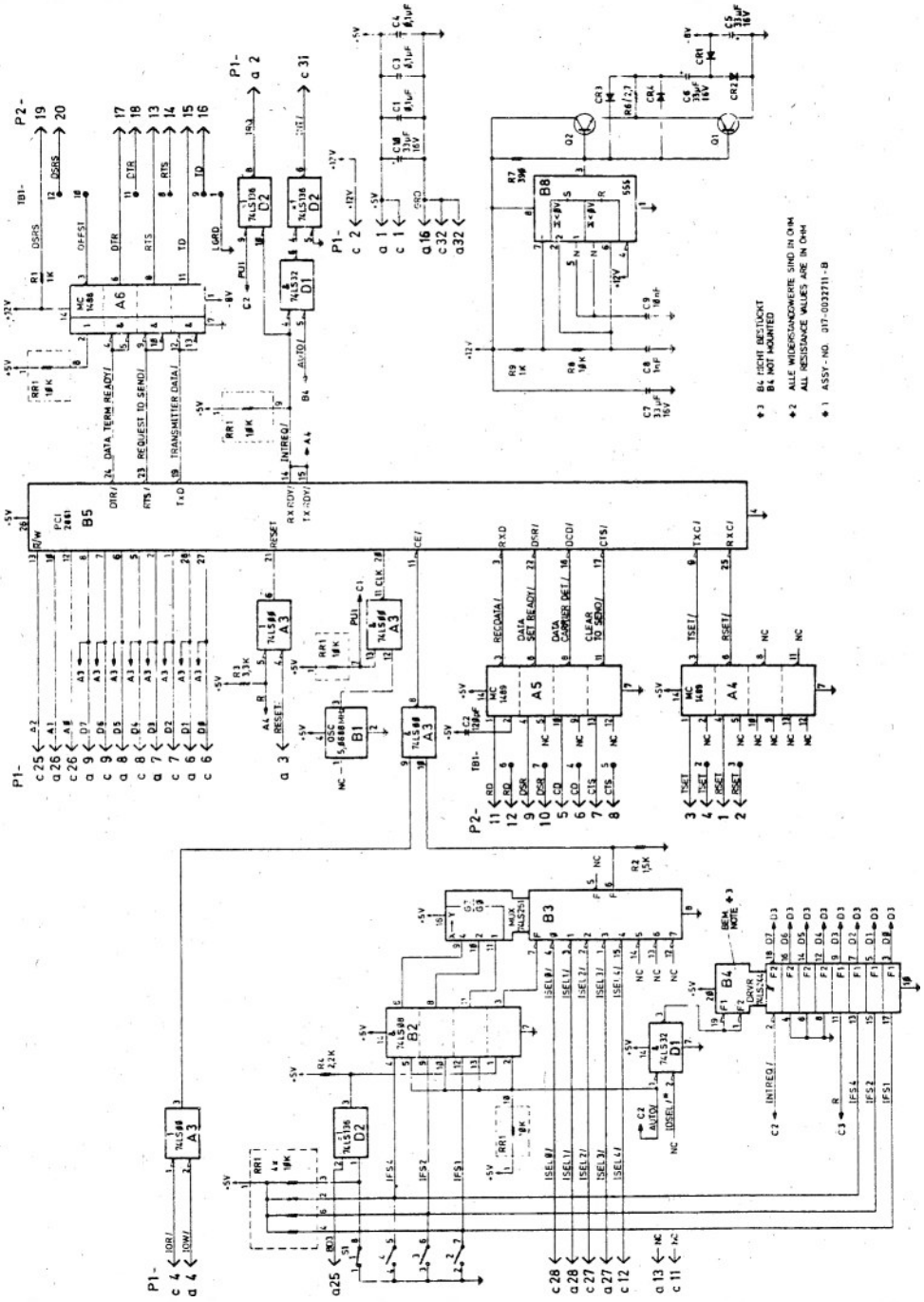
◆ 3 B4 NICHT BESTÜCKT
B4 NOT MOUNTED

2 ALLE WIDERSTANDSWERTE SIND IN OHM
ALL RESISTANCE VALUES ARE IN OHM

1 ASSY-NO.: 017-0032711-B

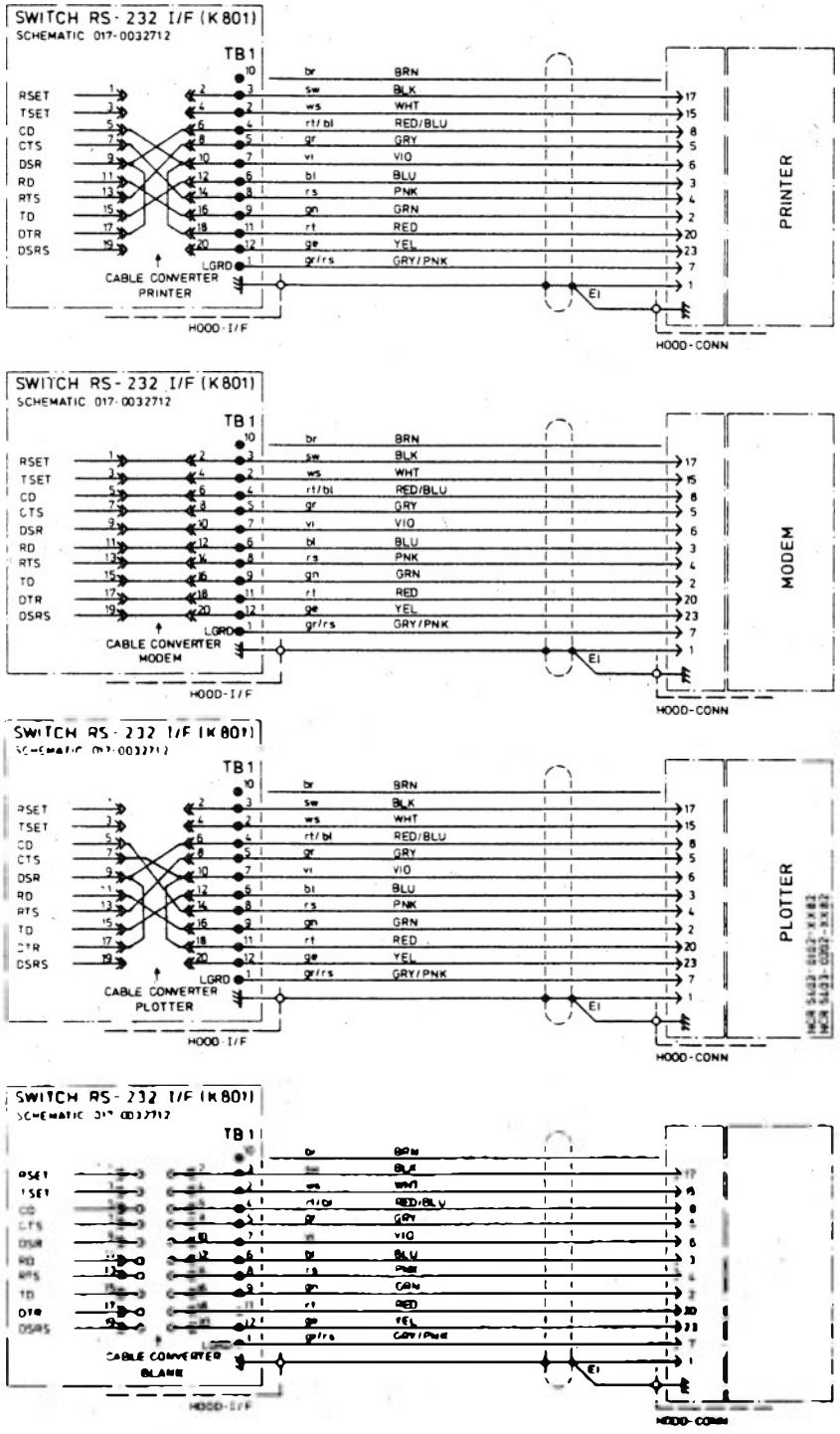
PARTS LIST of K 801 : switch RS-232C IF

REQD	PART DESCRIPTION		
1	PC-BOARD		
1	EPCI 2661		B5
2	MC 1489	LINE-RECEIVER	A4,A5
1	MC 1488	LINE-DRIVER	A6
1	74LS00	QUAD-NAND	A3
1	555	TIMER	B8
1	5.0688 MHz	OSCILL.-QUARZ	B1
1	BC 337	TRANSISTOR NPN	Q2
1	BC 237	" PNP	Q1
2	BZX/C1V4	DIODE	CR3,CR4
2	1 A -	RECTIFIER	CR1,CR2
4	33 UF/16V	CAPACITOR-T.	C5,6,7,10
1	120 PF/63V	" CER.	C2
3	.1 UF/50V	" -C.	C1,3,4
1	1000PF/100V	CAPACITOR CER.	C8
1	.01UF/50V	" -C.	C9
1	10 KOHM	RESISTOR	R8
2	1 KOHM	"	R1,9
1	390 OHM	"	R7
1	2.7 OHM	"	R6
1	2*32POL	CONNECTOR-PLUG	P1
12	TERMINAL-WIRE		TB1
1	9*10KOHM	RESISTOR NETWORK	RR1
1	3.3 KOHM	RESISTOR	R3
1	SWITCH		S1
1	74LS08	QUAD-AND	B2
1	74LS32	QUAD-OR	D1
1	74LS251	8 TO 1 MUX	B3
1	74LS136	QUAD-XOR	D2
1	1.5KOHM	RESISTOR	R2
1	2.2KOHM	"	R4
1	8POL	SOCKET,IC	FOR S1

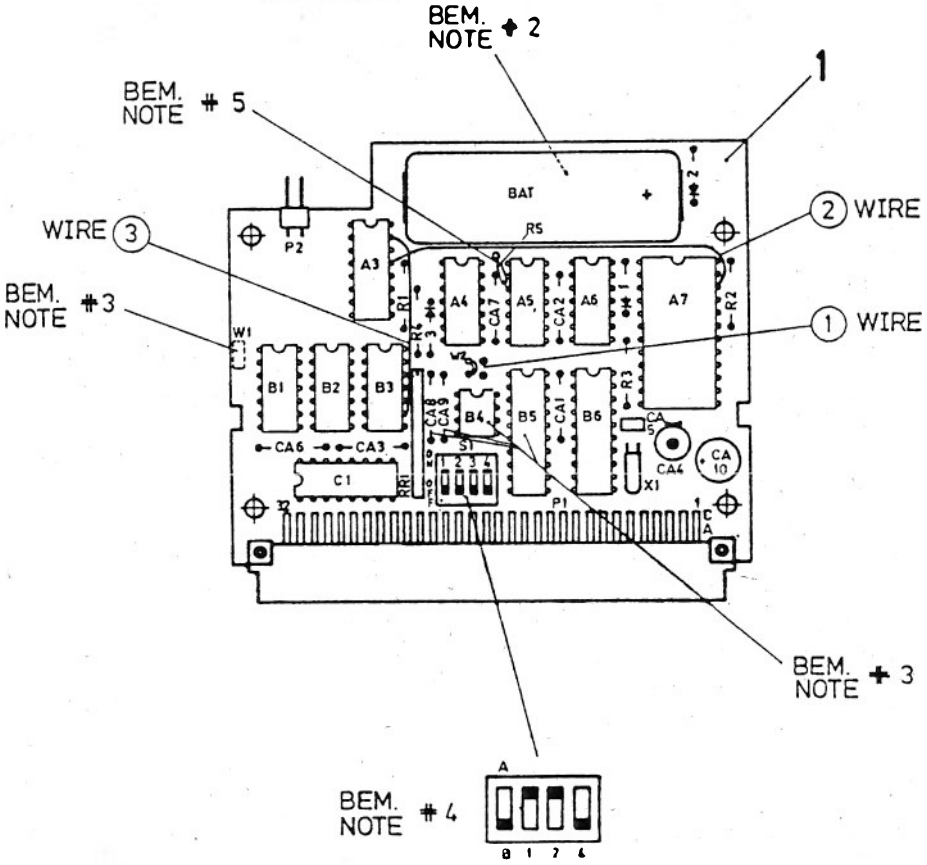


RS-232 C Switchable Interface (K801) 017-0032712 Rev. B

PIN ASSIGNMENTS & STRAPPING



K 803 REAL TIME CLOCK



✱ 5 WIDERSTAND POS. 031 : A5-3 NACH +5V
RESISTOR ITEM 031 : A5-3 TO +5V DC

✱ 4 SCHALTERGRUNDEINSTELLUNG: 4B
DEFAULT SWITCH POSITION : 4B

✱ 3 W1, CA8, CA9, B4, B5 NICHT BESTÜCKT
W1, CA8, CA9, B4, B5 NOT MOUNTED

✱ 2 LABEL „ ASSY 017-0032702-B "
AUFGEKLEBT / GLUED
LABEL „ SCHM. 017-0032703-B "
AUFGEKLEBT / GLUED

✱ 1 SCHEMATIC - NO.: 017-0032703-B

✱6 B4, CA8, CA9, B5 NICHT BESTÜCKT
B4, CA8, CA9, B5 NOT MOUNTED

✱5 W2/1 NACH W2/3 VERBINDEN
W2/1 TO W2/3 CONNECTION

✱4 W2/3 NACH W2/4 UNTERBROCHEN
W2/3 TO W2/4 CUT

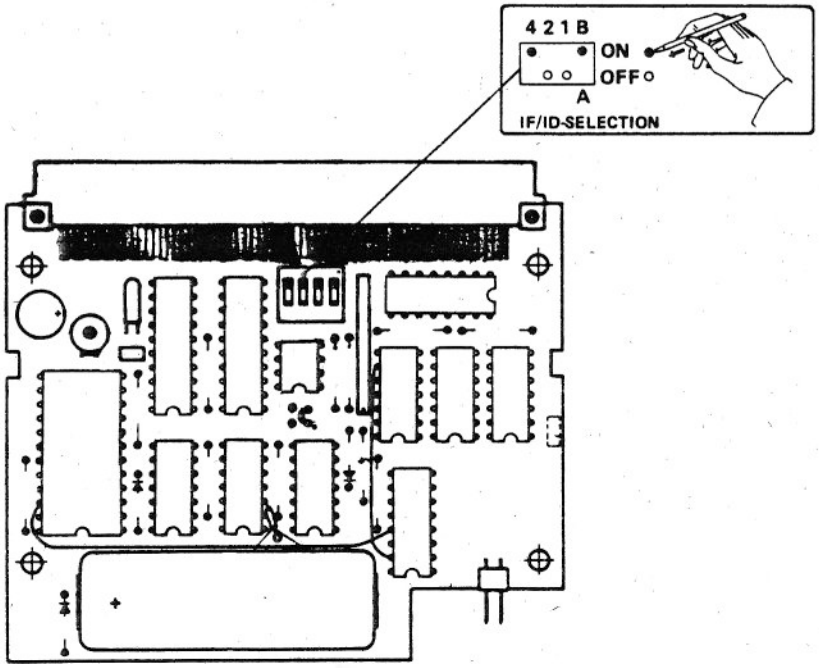
✱3 W1 NICHT BESTÜCKT
W1 NOT MOUNTED

✱2 ALLE WIDERSTANDSWERTE SIND IN OHM
ALL RESISTANCE VALUES ARE IN OHM

✱1 ASSY-NO.: 017-0032702-B

PIN ASSIGNMENTS & STRAPPING

REAL-TIME-CLOCK
(K 803)

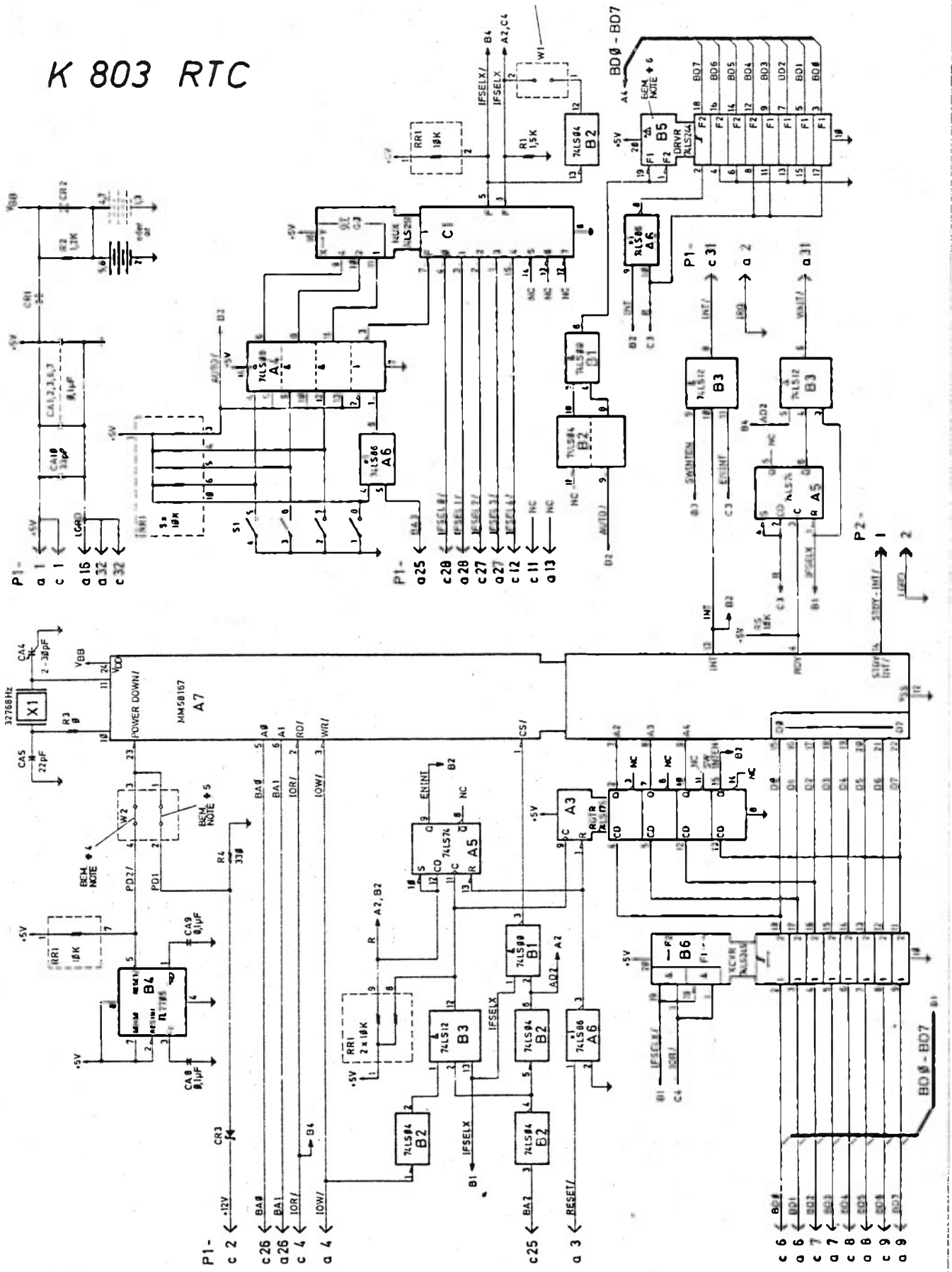


IFSEL switches

IFSEL	SWITCH 4 2 1 B	PORT HEX	DEC
0A	○ ○ ○ ○	60-67H	96-103
0B	○ ○ ○ ●	68-6FH	104-111
1A	○ ○ ● ○	70-77H	112-119
1B	○ ○ ● ●	78-7FH	120-127
2A	○ ● ○ ○	30-37H	48- 55
2B	○ ● ○ ●	38-3FH	56- 63
3A	○ ● ● ○	B0-B7H	176-183
3B	○ ● ● ●	B8-BFH	184-191
4A	● ○ ○ ○	C0-C7H	192-199
4B	● ○ ○ ●	C8-CFH	200-207

IFSEL switch settings

K 803 RTC



PARTS LIST of K 803 : Real-time-clock IF

REQD	PART DESCRIPTION		
1	PC-BOARD		
1	MM 58167A RTC-uP		A7
1	74LS251	8 to 1 MUX	C1
1	74LS08	QUAD-AND	A4
1	74LS86	QUAD-XOR	A6
1	74LS175	QUAD-D-FF	A3
1	74LS00	QUAD-NAND	B1
1	74LS04	HEX INVERTER	B2
1	74LS12	TRIPPLE-NAND	B3
1	9*10kOHM	RESISTOR NETWORK	RR1
1	33 UF/16V	CAPACITOR-T.	CA10
1	15 PF/63V	" CER.	CA5
5	.1 UF/50V	" -C.	CA1,2,3,6,7
1	32768 Hz	CRISTAL	X1
1	200 KOHM	RESISTOR	R3
2	1N4148	DIODE	CR1,2
1	1.2 KOHM	RESISTOR	R2
1	3.6V	NC-BATTERY	
1	2*32POL	CONNECTOR-PLUG	P1
1	DIP-SWITCH		S1
1	1.5 KOHM	RESISTOR	R1
1	74LS74	DUAL-D-FF	A5
1	74LS245	OKTAL-BUS-TRANSCEIVER	B6
1	2-30 PF	CAPACITOR, VAR.	CA4
1	8.2V/.5W	ZENER-DIODE	CR3
1	330 OHM	RESISTOR	R4
1	10 KOHM	"	R5

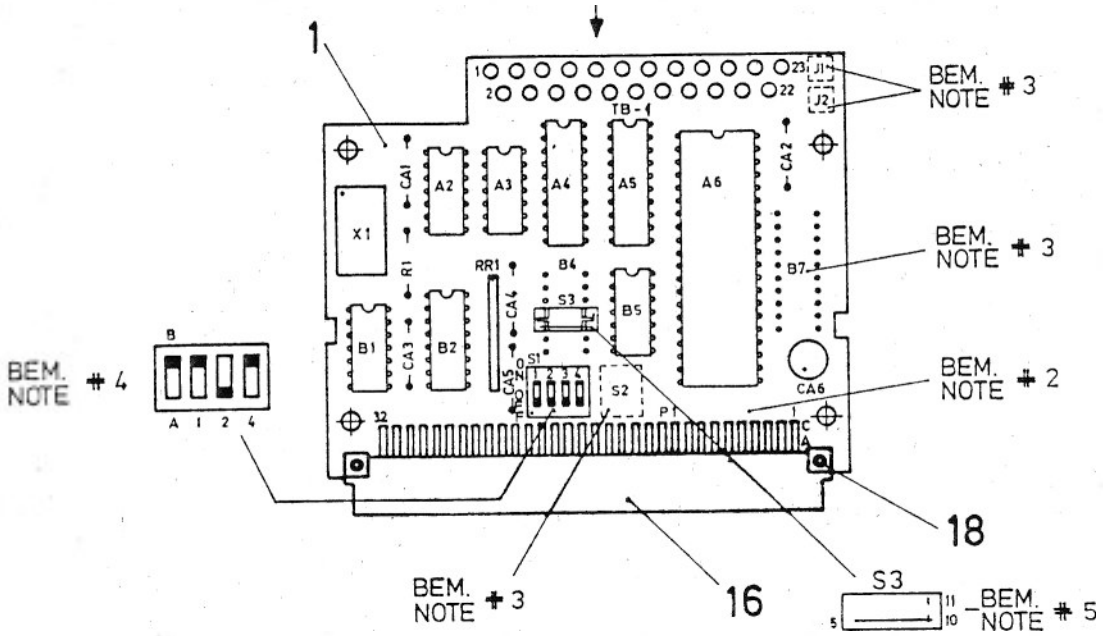
a		c
+5V	1	+5V
IRQ	2	+12V
RESET/	3	
IOW/	4	IOR/
	5	
BD1	6	BD0
BD3	7	BD2
BD5	8	BD4
BD7	9	BD6
	10	
	11	IDSEL/
	12	IFSEL4
AUTO/	13	
	14	
	15	
LGRD	16	
	17	
	18	
	19	
	20	
	21	
	22	
	23	
	24	
BA3	25	BA2
BA1	26	BA0
IFSEL3	27	IFSEL2
IFSEL1	28	IFSEL0
	29	
	30	
WAIT/	31	INT/
LGRD	32	LGRD

Pin assignments P1

P2-1	STBY-INT/
-2	LGRD

Standby interrupt connector

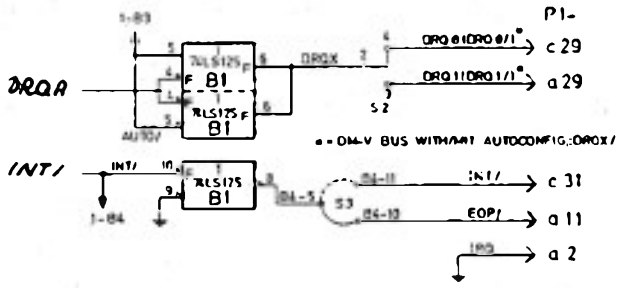
K 804 IEEE 488(IEC) BUS IF



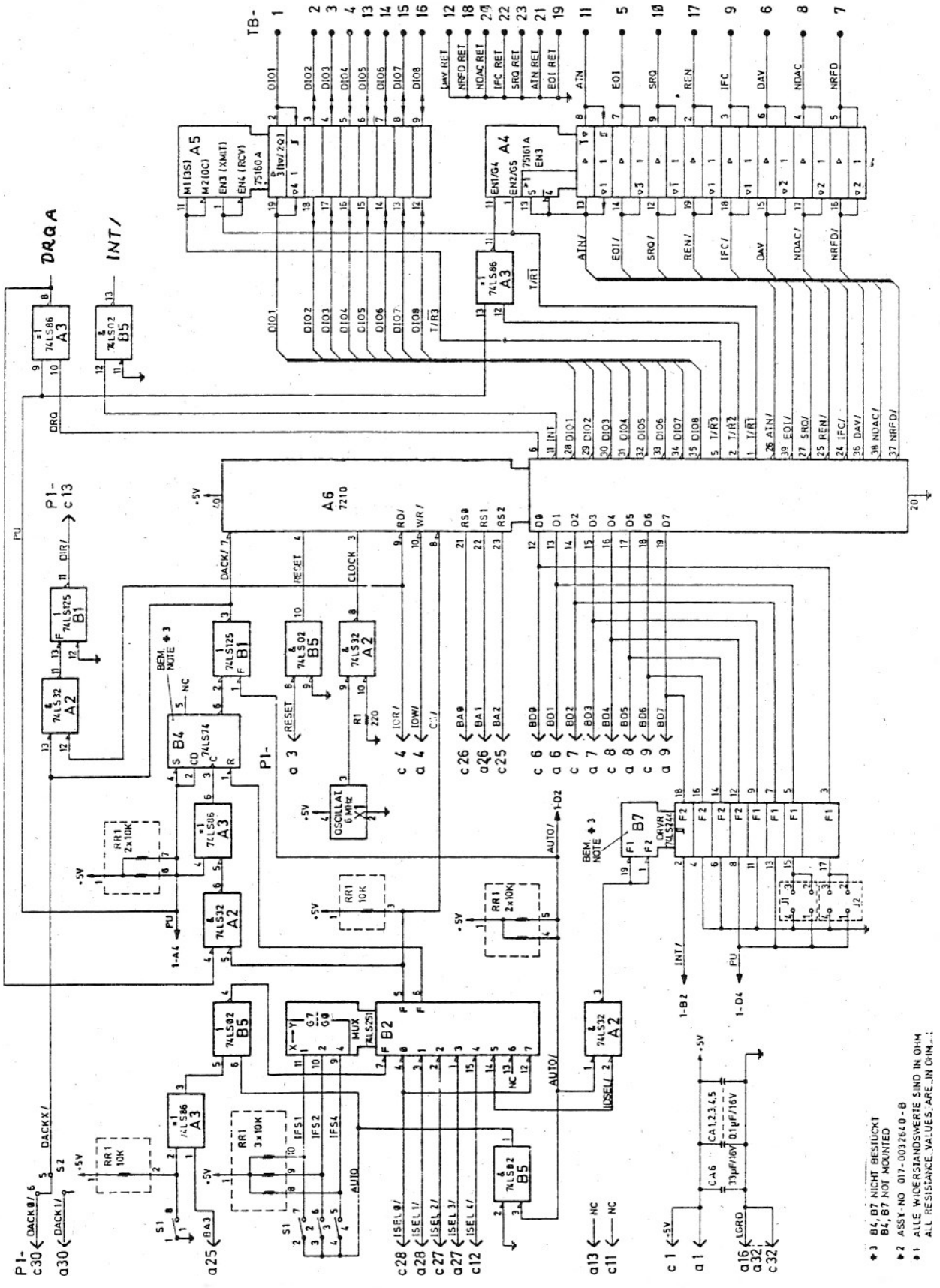
5 DRAHT-SCHALTER GRUNDEINSTELLUNG:
GESCHLOSSEN S3/5 – S3/10
WIRE-SWITCH DEFAULT:
CLOSED S3/5 – S3/10

4 SCHALTERGRUNDEINSTELLUNG: 2B
DEFAULT SWITCH POSITION: 2B

3 J1, J2, S2, B7 NICHT BESTÜCKT
J1, J2, S2, B7 NOT MOUNTED



K 804 IEEE 488

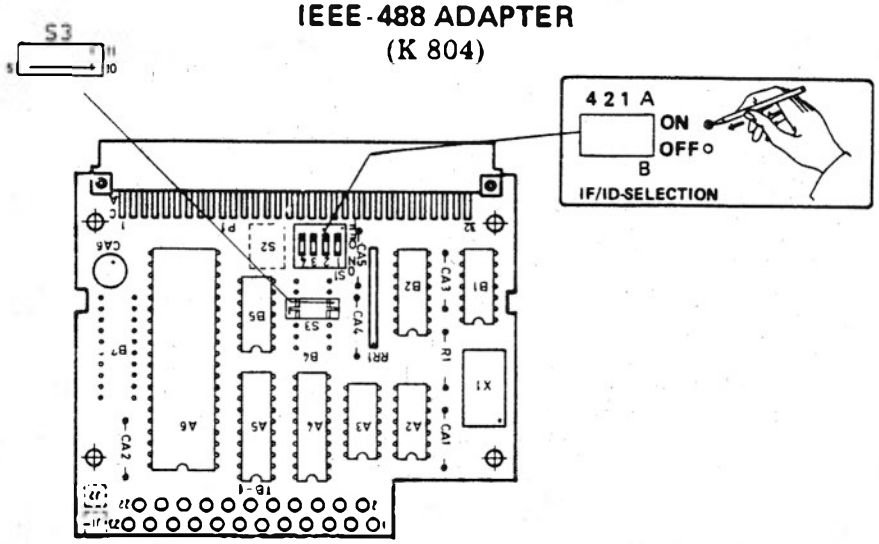


PARTS LIST of K 804 : IEEE-488 IF

REQD	PART DESCRIPTION		
1	PC-BOARD		
1	uPD7210	GPIB-INTERFACE CONTROLLER	A6
1	75161	" -TRANSCEIVER	A4
1	75160	" -"	A5
1	74LS251	8 to 1 MUX	B2
1	74LS86	QUAD-XOR	A3
1	74LS32	QUAD-OR	A2
1	74LS125	QUAD-BUFFER	B1
1	6,0 MHz	CLOCK OSZILLATOR	X1
1	DIP-SWITCH		S1
1	9*10kOHM	RESISTOR NETWORK	RR1
1	33 UF/16V	CAPACITOR-T.	CA6
5	.1 UF/50V	" -C.	CA1,2,3,4,5
1	2*32POL	CONNECTOR-PLUG	P1
1	74LS02	QUAD-NOR	B5
1	220 OHM	RESISTOR	R1

PIN ASSIGNMENTS & STRAPPING

IEEE-488 ADAPTER
(K 804)

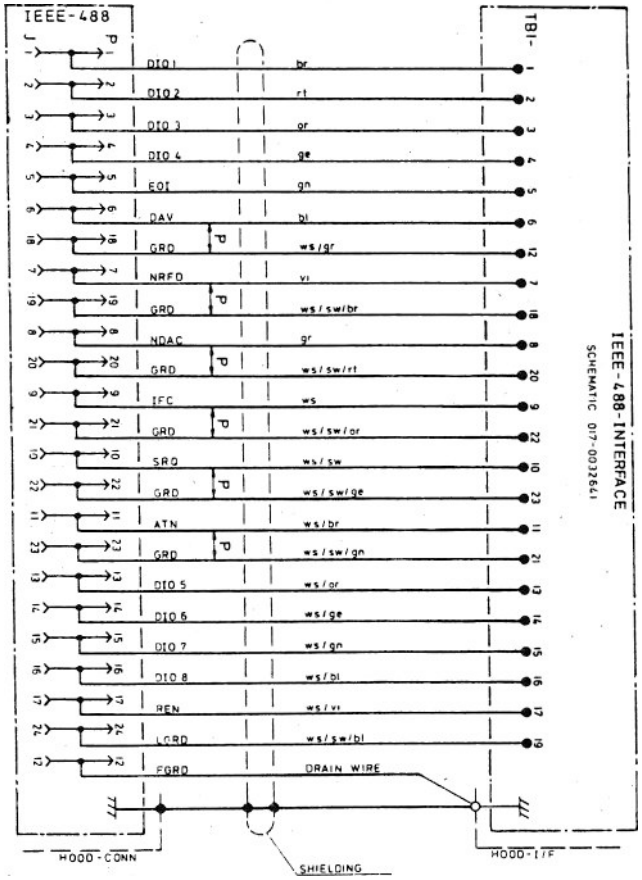


a		c
+5V	1	+5V
IRQ	2	
Reset/	3	
IOW/	4	IOR/
	5	
BD1	6	BD0
BD3	7	BD2
BD5	8	BD4
BD7	9	BD6
	10	
EOP/	11	
	12	ISEL4/
	13	DIR/
	14	
	15	
LGRD	16	
	17	
	18	
	19	
	20	
	21	
	22	
	23	
	24	
BA3	25	BA2
BA1	26	BA0
ISEL3/	27	ISEL2/
ISEL1/	28	ISEL0/
DRQ1	29	DRQ0
DACK1/	30	DACK0/
	31	INT/
LGRD	32	LGRD

Pin assignments P1

PIN ASSIGNMENTS & STRAPPING

IEEE Cable

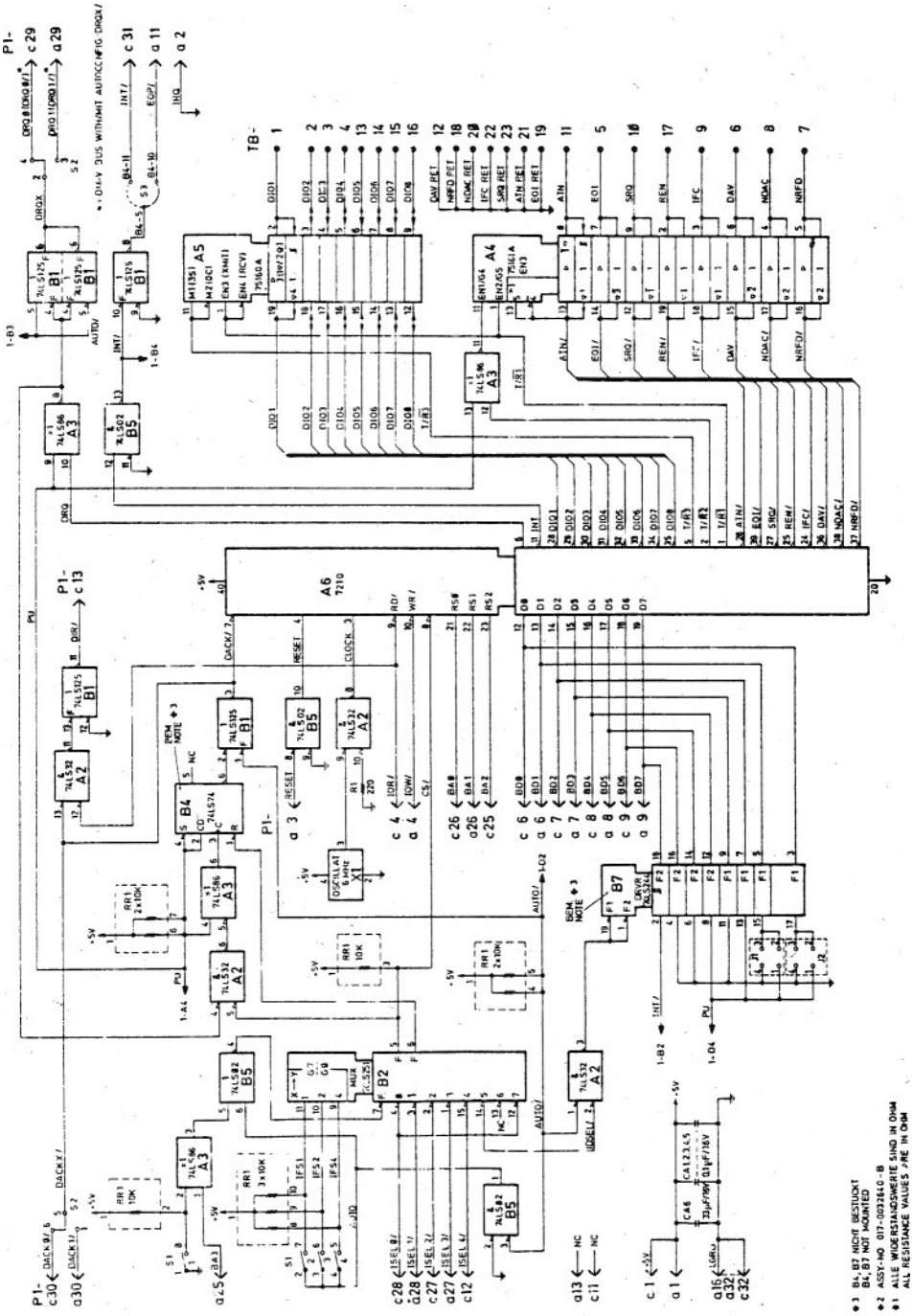


IEEE Cable

Follow instructions inside adapter cover: ignore any markings on switch assembly.

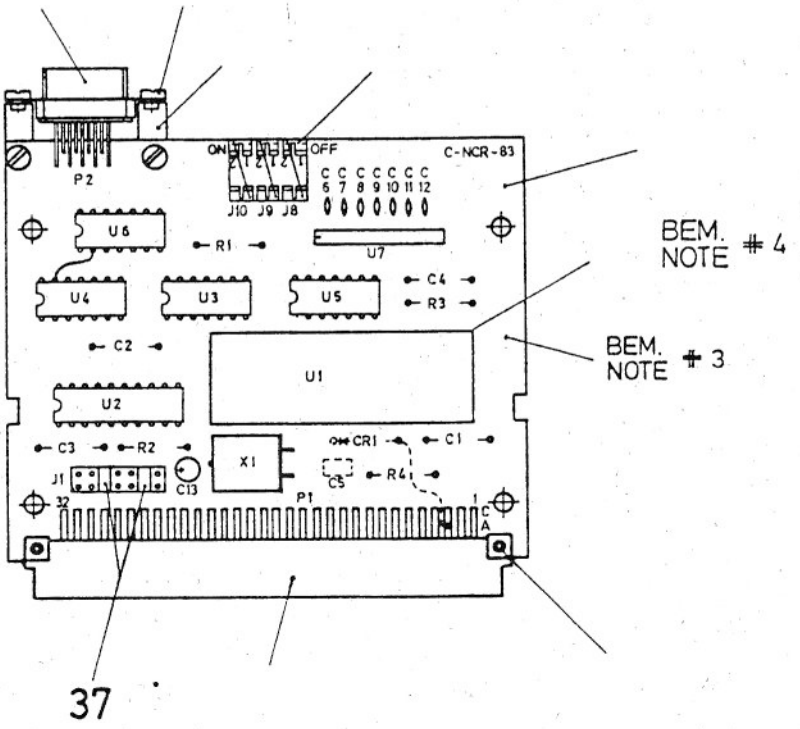
IFSEL	Switch 4	2	1	4	Port Addresses
0A	○	○	○	●	60-67 Hex
0B	○	○	○	○	68-6F Hex
1A	○	○	●	●	70-77 Hex
1B	○	○	●	○	78-7F Hex
2A	○	●	○	●	30-37 Hex
2B	○	●	○	○	38-3F Hex
3A	○	●	●	●	B0-B7 Hex
3B	○	●	●	○	B8-BF Hex
4A	●	○	○	●	C0-C7 Hex
4B	●	○	○	○	C8-CF Hex
○ = Open ● = Closed					

IFSEL Switch settings



IEEE-488 Interface (K804) 017-0032641 Rev. B

K 806 Mouse Interface



JUMPER STRAPPING

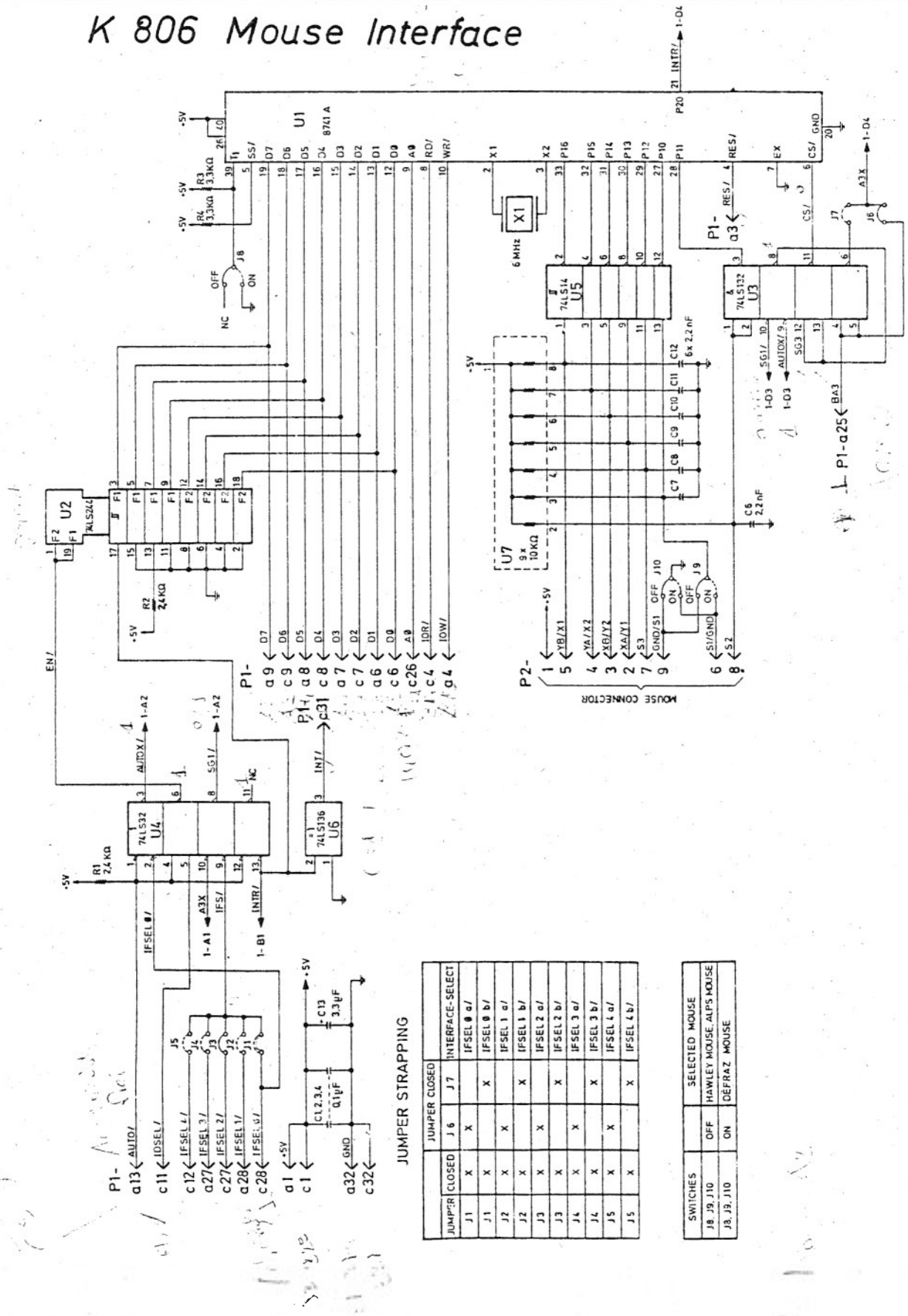
		JUMPER CLOSED		
JUMPER	CLOSED	J6	J7	INTERFACE-SELECT
J1	X	X		IFSELØa/
J1	X		X	IFSELØb/
J2	X	X		IFSEL 1a/
J2	X		X	IFSEL 1b/
J3	X	X		IFSEL 2a/
J3	X		X	IFSEL 2b/
J4	X	X		IFSEL 3a/
J4	X		X	IFSEL 3b/
J5	X	X		IFSEL 4a/
J5	X		X	IFSEL 4b/

SWITCHES		SELECTED MOUSE
J8, J9, J10	OFF	HAWLEY MOUSE, ALPS MOUSE
J8, J9, J10	ON	DEPRAZ MOUSE

4 U1 IN K806 AUFGEFÜHRT
U1 LISTED IN K 806

2 CR1, C5 NICHT BESTÜCKT
CR1, C5 NOT MOUNTED

K 806 Mouse Interface



JUMPER STRAPPING

JUMPER	CLOSED	JUMPER CLOSED J6	JUMPER CLOSED J7	INTERFACE-SELECT
J1	X	X		IFSEL 0 a/
J1	X		X	IFSEL 0 b/
J2	X	X		IFSEL 1 a/
J2	X		X	IFSEL 1 b/
J3	X	X		IFSEL 2 a/
J3	X		X	IFSEL 2 b/
J4	X	X		IFSEL 3 a/
J4	X		X	IFSEL 3 b/
J5	X	X		IFSEL 4 a/
J5	X		X	IFSEL 4 b/

SWITCHES		SELECTED MOUSE
J8, J9, J10	OFF	HAWLEY MOUSE, ALPS MOUSE
J8, J9, J10	ON	DEFRAZ MOUSE

PIN ASSIGNMENTS & STRAPPING

P1	+5V
2	XA/Y1
3	XB/Y2
4	YA/X2
5	YB/X1
6	S1/BND
7	S3
8	S2
9	GND/S1

Mouse connector pin assignments

	Jumpers Closed
IFSEL 0a/	J1, J6
IFSEL 0b/	J1, J7
IFSEL 1a/	J2, J6
IFSEL 1b/	J2, J7
IFSEL 2a/	J3, J6
IFSEL 2b/	J3, J7
IFSEL 3a/	J4, J6
IFSEL 3b/	J4, J7
IFSEL 4a/	J5, J6
IFSEL 4b/	J5, J7

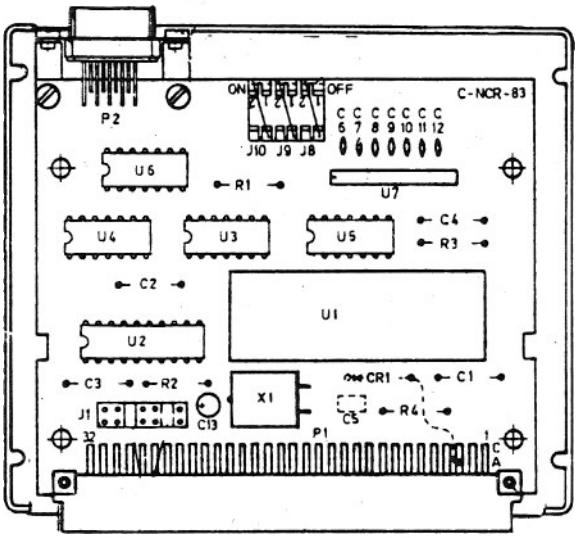
IFSEL Jumpers

Alps Mouse, Hawkey Mouse	J8, J9, J10 Off
Depraz Mouse	J8, J9, J10 On

Mouse selection jumpers

PIN ASSIGNMENTS & STRAPPING

MOUSE INTERFACE
(K 806)



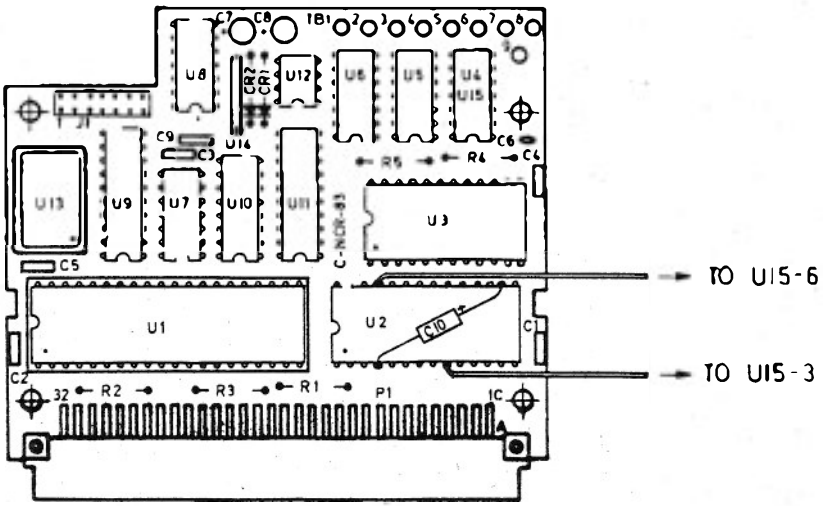
a		
+5V	1	+5V
	2	
RESET/	3	
IOW/	4	IOR/
	5	
D1	6	D0
D3	7	D2
D5	8	D4
D7	9	D6
	10	
	11	IDSEL/
	12	IFSEL4/
AUTO/	13	
	14	
	16	
	17	
	18	
	19	
	20	
	21	
	22	
	23	
	24	
BA3	25	
	26	A0
IFSEL3/	27	IFSEL2/
IFSEL1/	28	IFSEL0/
	29	
	30	
	31	INT/
GND	32	GND

Pin assignments P1

PARTS LIST of K 806 : MOUSE IF

REQD	PART DESCRIPTION		
1	PC-BOARD		
1	uPD8741	PERIPHERAL INTERFACE,PROGR.	U1
1	74LS244	OKTAL-BUS-DRIVER	U2
1	74LS132	QUAD-NAND SCHMITT-TRIGG.	U3
1	74LS32	QUAD-OR	U4
1	74LS14	HEX INVERTER SCHMITT-TRIGG.	U5
1	74LS136	QUAD-XOR	U6
1	9*10kOHM	RESISTOR NETWORK	U7
2	2.4 kOHM	RESISTOR	R1,2
2	3.3 kOHM	"	R3,4
1	6.0 MHZ	CRISTAL OSC.	X1
7	2200PF	CAPACITOR-CERAMIC	C6,7,8,9,10,11,12
1	3.3 UF/16V	CAPACITOR-T.	C13
4	.1 UF/50V	" -C.	CA1,2,3,4
1	2*32POL	CONNECTOR-PLUG	P1
1	9POL	RECEPTACLE	P2
1	40POL	SOCKET, IC	FOR U1

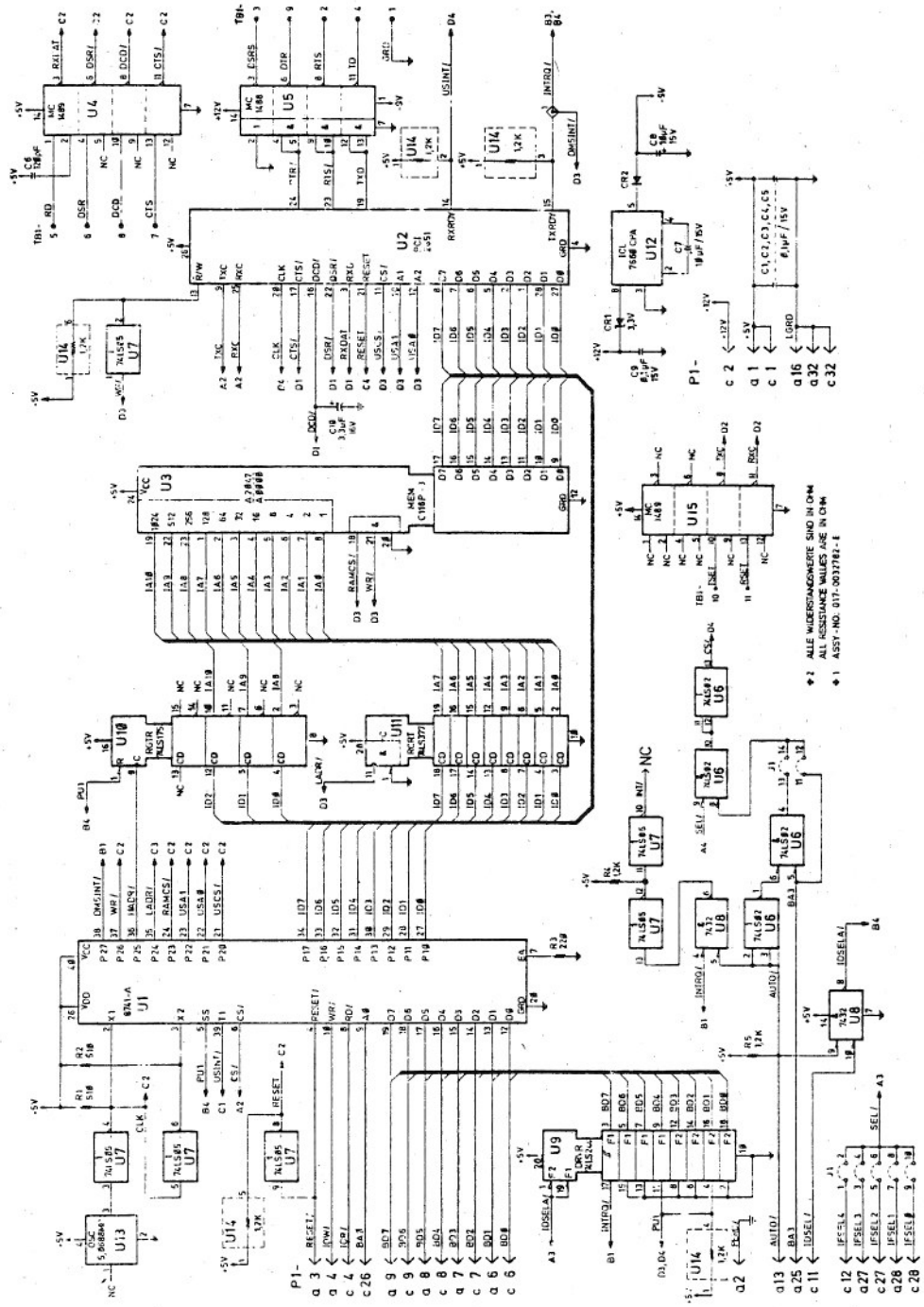
BUFFERED SYNC/ASYNC ADAPTER
(K 215)



a		c
+5V	1	+5V
PERC/	2	+12V
RESET/	3	
IOW/	4	IOR/
	5	
BD1	6	BD0
BD3	7	BD2
BD5	8	BD4
BD7	9	BD6
	10	
	11	IDSEL/
	12	IFSEL4
AUTO/	13	
	14	
	15	
LGRD	16	
	17	
	18	
	19	
	20	
	21	
	22	
	23	
	24	
BA3	25	
	26	BA0
IFSEL3	27	IFSEL2
IFSEL1	28	IFSEL0
	29	
	30	
	31	
LGRD	32	LGRD

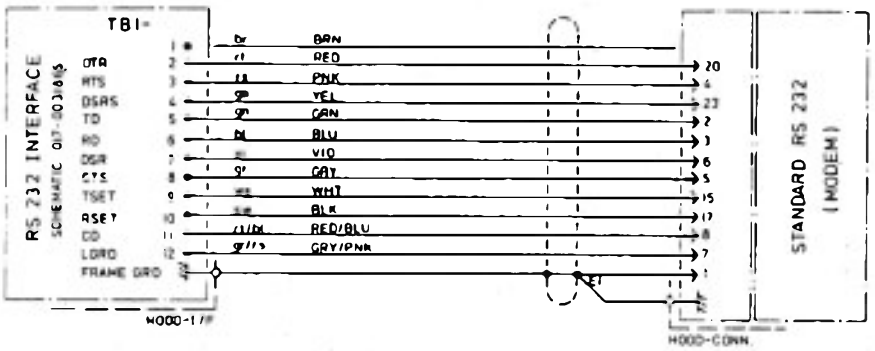
Pin assignment P1

SCHEMATICS

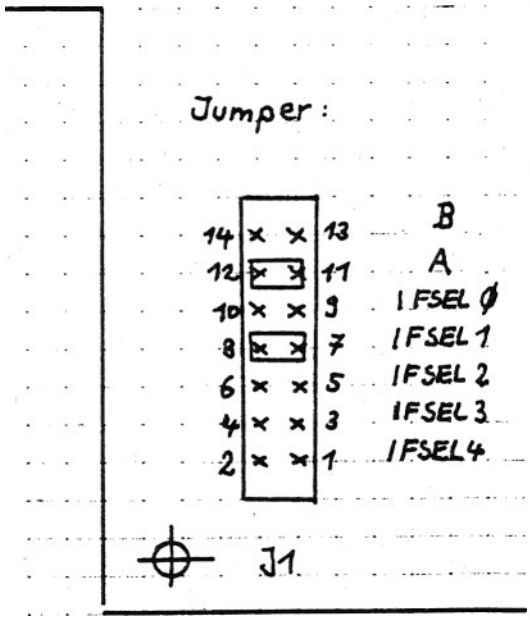


Buffered Async/Bisync Interface (K215) 017-0032792 Rev. C

PIN ASSIGNMENTS & STRAPPING



IFSEL	J1	Port Address
0A	11-12 and 9-10	60-67 hex
0B	13-14 and 9-10	68-6F hex
1A	11-12 and 7-8	70-77 hex
1B	13-14 and 7-8	78-7F hex
2A	11-12 and 5-6	30-37 hex
2B	13-14 and 5-6	38-3F hex
3A	11-12 and 3-4	B0-B7 hex
3B	13-14 and 3-4	B8-BF hex
4A	11-12 and 1-2	C0-C7 hex
4B	13-14 and 1-2	C8-CF hex



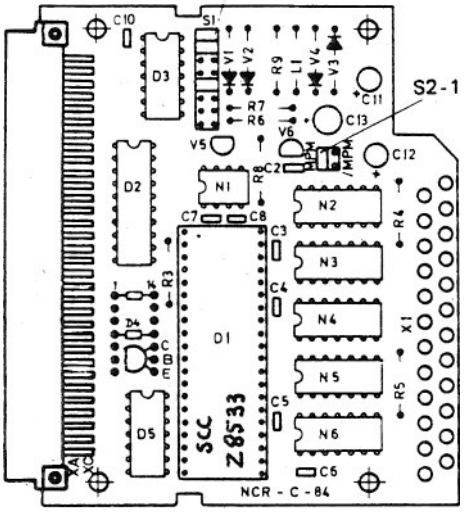
$\triangleq 70_H - 7F_H$

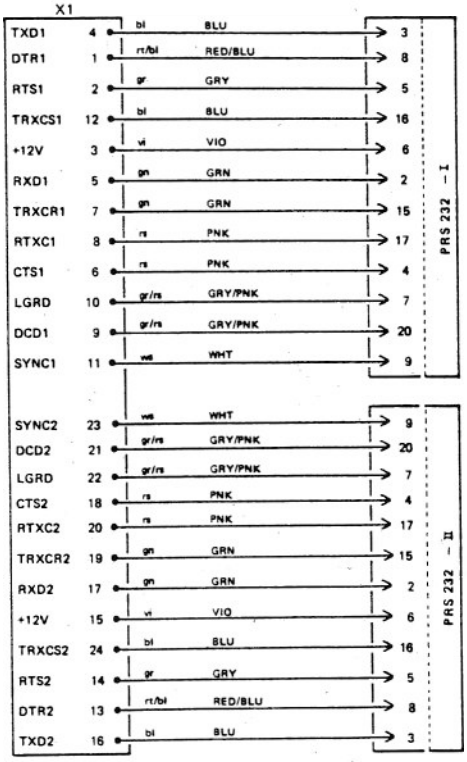
PARTS LIST of K 215 : buffered sync/async RS-232C IF

REQD	PART DESCRIPTION		
1	PC-BOARD		
1	8741 PROCESSOR WITH FIRMWARE		U1
1	PCI 2651		U2
1	6116	RAM 2K*8	U3
2	MC 1489	LINE-RECEIVER	U4,15
1	MC 1488	LINE-DRIVER	U5
1	74LS02	QUAD-NOR	U6
1	74LS05	HEX INVERTER	U7
1	74LS32	QUAD-OR	U8
1	74LS244	OCTAL-BUS-DRIVER	U9
1	74LS175	QUAD D-FF	U10
1	74LS377	OCTAL D-FF	U11
1	ICL7660	VOLTAGE-REGUL.	U12
1	5.0688 MHz	OSCILL.-QUARZ	U13
1	5*1.2kOHM	RESISTOR-NETWORK	U14
2	510 OHM	RESISTOR	R1,2
1	220 OHM	"	R3
2	1.2 kOHM	"	R4,5
2	10 UF/16V	CAPACITOR-T.	C7,8
1	3.3 UF/15V	" -T.	C10
1	120 PF/63V	" CER.	C6
6	.1 UF/50V	" -C.	C1,2,3,4,5,9
1	3.3V	DIODE-ZENER	CR1
1	1A	RECTIFIER	CR2
1	40POL	SOCKET,IC	FOR U1
1	2*32POL	CONNECTOR-PLUG	P1

SCC COMMUNICATION INTERFACE (K216)

S1-1





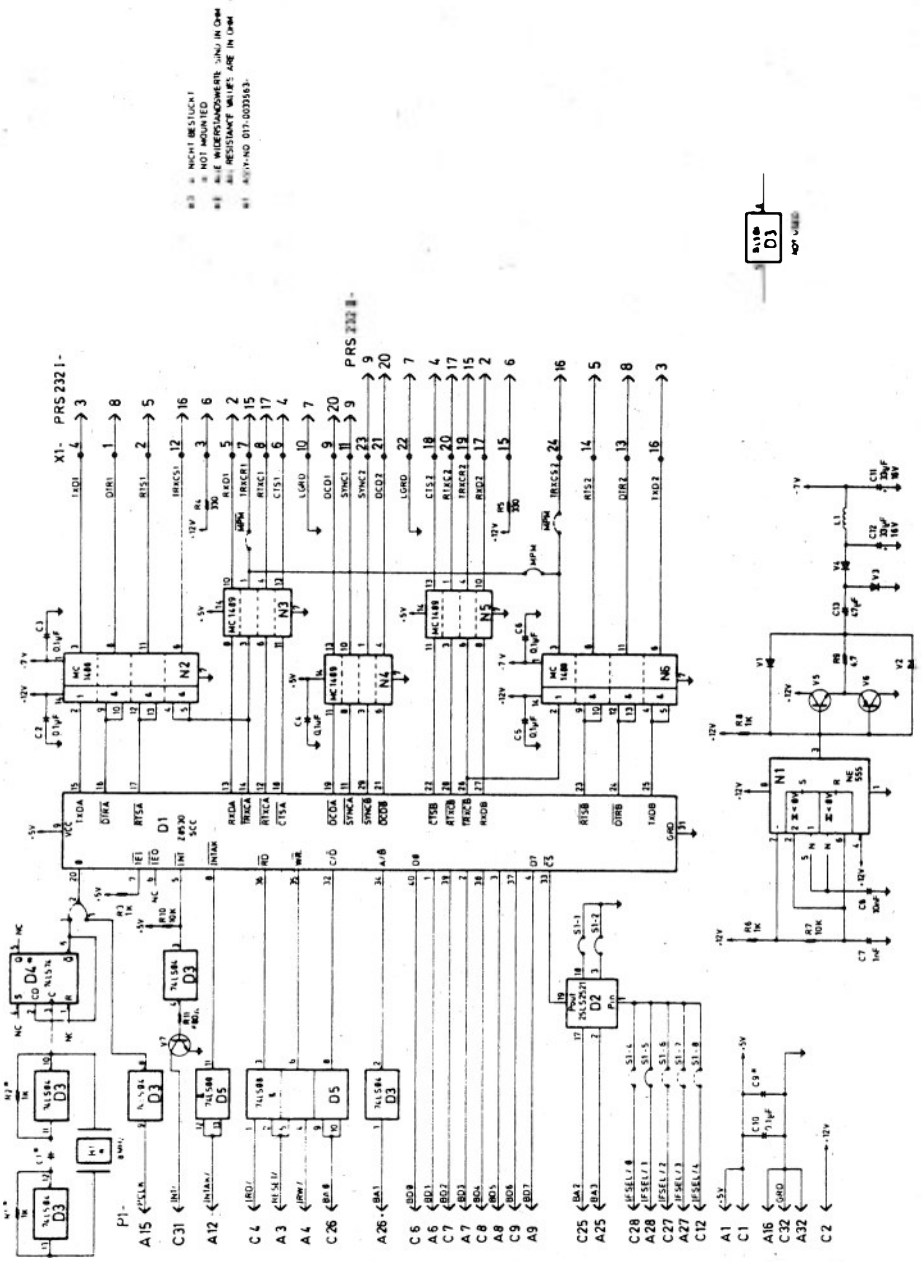
2 CHANNEL SERIAL CABLE

K 216 SCC COMMUNICATION I/F

Description at a glance:

- With usage of the Z 8530 SCC communication controller two independent serial channels (full duplex) can be used (channel select by adress line 1 / BA1)
- Switchable Portadress: I/F SEL 0 - 4 (internal by jumbers)
- Asynchronous mode with 5 - 8 bit data, 1, 1 1/2 or 2 stopbits
- Programmable for NRZ, NRZI or FM data encoding
- Break detection and generation
- Parity, overrun and framing error detection
- Synchronous mode, CRC generation and checking
- SDLC/HDLC mode
- Local loopback and auto echo modes
- Switchable for MP/M (internal jumber)
- Interconnection cables are shielded, length 10 m, peripheral plug type D-subminiatur (25 pins)
- Description of interface signals:

PIN 1	FRAMEGRD	frameground
PIN 2	TXD	data transmitted (+/-3 V to +/-12 V)
Pin 3	RXD	data received
PIN 4	RTS	request to sent, I if DM V wishes to transmit
PIN 5	CTS	clear to sent, I for data transmission (input)
PIN 7	LGRD	logic ground
PIN 8	DCD	carrier detect, I for receiving data (input)
PIN 9	SYNC	
PIN 15	TRXCR	
PIN 16	TRXCS	
PIN 17	RTXC	
PIN 20	DTR	data terminal ready: DM V ready to receive data



SCC Communication Interface (K216) 017-0033564 Rev. B

PIN ASSIGNMENTS & STRAPPING

K216 SCC

K216 - V001	MPM	MPM/	S1-3
MPM Operation	●	o	o
Two Channel RS-232	o	●	●

K216 - V002	S2-1	S2-2	S2-3
MPM Operation	●	o	o
Two Channel RS-232	o	●	●

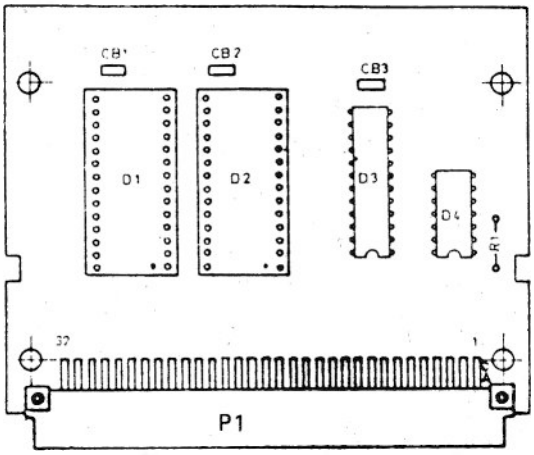
Strapping SCC communication interface

IFSEL		Switch S1 1 2 4 5 6 7 8	Port Address
IFSEL 0	0A1	x x x o o o o	60 - 63 Hex
	0A2	x o x o o o o	64 - 67 Hex
	0B1	o x x o o o o	68 - 6B Hex
	0B2	o o x o o o o	6C - 6F Hex
IFSEL 1	1A1	x x o x o o o	70 - 73 Hex
	1A2	x o o x o o o	74 - 77 Hex
	1B1	o x o x o o o	78 - 7B Hex
	1B2	o o o x o o o	7C - 7F Hex
IFSEL 2	2A1	x x o o x o o	30 - 33 Hex
	2A2	x o o o x o o	34 - 37 Hex
	2B1	o x o o x o o	38 - 3B Hex
	2B2	o o o o x o o	3C - 3F Hex
IFSEL 3	3A1	x x o o o x o	B0 - B3 Hex
	3A2	x o o o o x o	B4 - B7 Hex
	3B1	o x o o o x o	B8 - BB Hex
	3B2	o o o o o x o	BC - BF Hex
IFSEL 4	4A1	x x o o o o x	C0 - C3 Hex
	4A2	x o o o o o x	C4 - C7 Hex
	4B1	o x o o o o x	C8 - CB Hex
	4B2	o o o o o o x	CC - CF Hex
x = closed			o = open

SCC Communication Cartridge

SHARED RAM
(K 233)

PIN ASSIGNMENTS & STRAPPING



a		c
+5V	1	+5V
	2	+12V
RESET/	3	
IWR/	4	IRD/
MWR/	5	MRD/
BD1	6	BD0
BD3	7	BD2
BD5	8	BD4
BD7	9	BD6
	10	
	11	
INTAK/	12	IFSEL4/
	13	DIR/
	14	
PCLK	15	
GRD	16	TRAMD/
BA19	17	BA18
BA17	18	BA16
BA15	19	BA14
BA13	20	BA12
BA11	21	BA10
BA9	22	BA8
BA7	23	BA6
BA5	24	BA4
BA3	25	BA2
BA1	26	BA0
IFSEL3/	27	IFSEL2/
IFSEL 1/	28	IFSEL0/
	29	
	30	
	31	INT/
GRD	32	GRD

Pin assignments P1 (shared RAM)

K 233 SHARED RAM BOARD

Description at a glance:

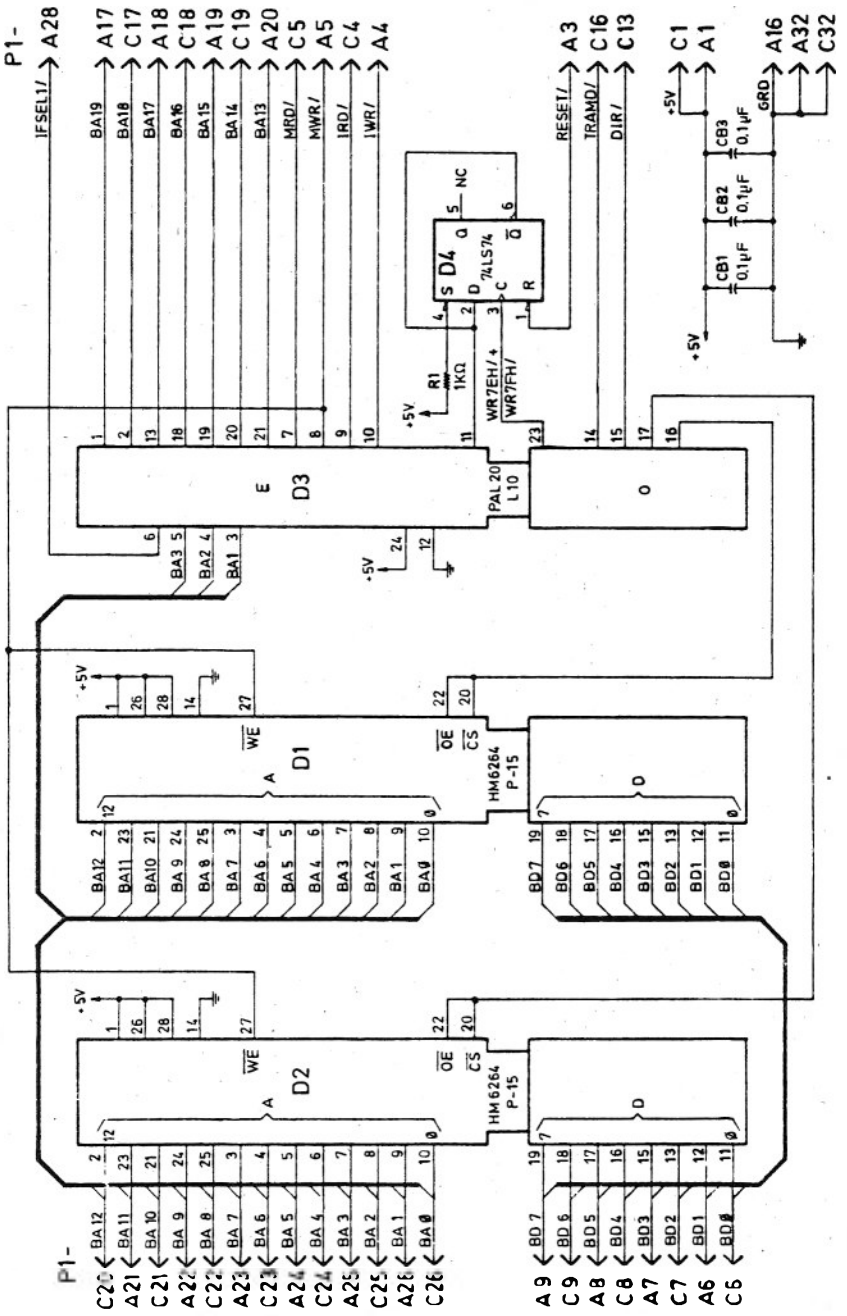
- 16 KByte static RAM with following adressrange: C000 H - FFFF H
- Function of shared RAM switchable by portadress 7F H
- Independent from bankswitching available in all memory banks
- With MP/M operating system the 8-bit DM V with Z 80 can support 256 K RAM
- This function requires the memory expansion K 202 (192 K) or K208 (448 K)

PARTS LIST of K 233 : shared RAM

REQD	PART DESCRIPTION		
1	PC-BOARD		
2	6264	CMOS-RAM 8k*8	D1,2
1	PAL, programmed		D3
1	74LS74	DUAL-D-FF	D4
1	1.0 kOHM	RESISTOR	R1
3	.1 UF/50V	" -C.	CB1,2,3
1	2*32POL	CONNECTOR-PLUG	P1
1	24POL	SOCKET, IC	FOR D3
2	24POL	SOCKET, IC	FOR D1,2

K233

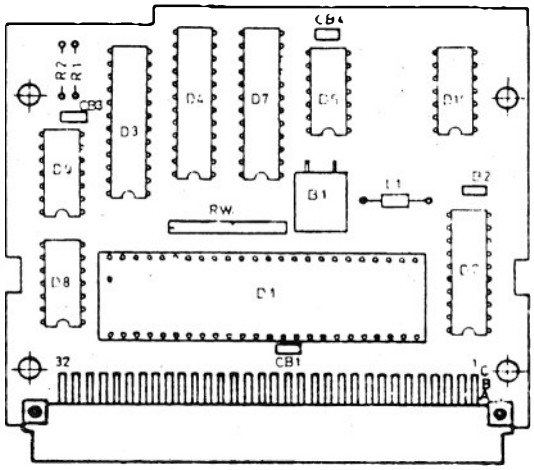
SCHEMATICS



Shared RAM (K233) 017-0033582 Rev. A

Third Party Support Center
NCR GmbH - TSC
Ulmer Straße 160 D-8900 Augsburg
Engineering&Manufacturing
Phone 821/405 1
Telex 5 3749 - Telefax 8211405462
For information only without Guaranty

68008 PROCESSOR BOARD
(K234)



	a	b	c
1	+5V		+5V
2			
3	RESET/		
4	IOW/		IOR/
5	MEMW/		MEMR/
6	D1		D0
7	D3		D2
8	D5	TIMINT	D4
9	D7		D6
10		WAITP/	
11		HOLDDMA/	
12	INTAC/	PROCCH/	
13		HOLDA16	
14	THOLD/		HOLDA
15			
16	GRD		
17	BA19	16 BIT SET	BA18
18	BA17		BA16
19	A15		A14
20	A13	MEMRQ/	A12
21	A11		A10
22	A9		A8
23	A7		A6
24	A5		A4
25	A3		A2
26	A1		A0
27			
28			
29			
30			
31			INT/
32	GRD		GRD

Pin assignments
Processor 68008

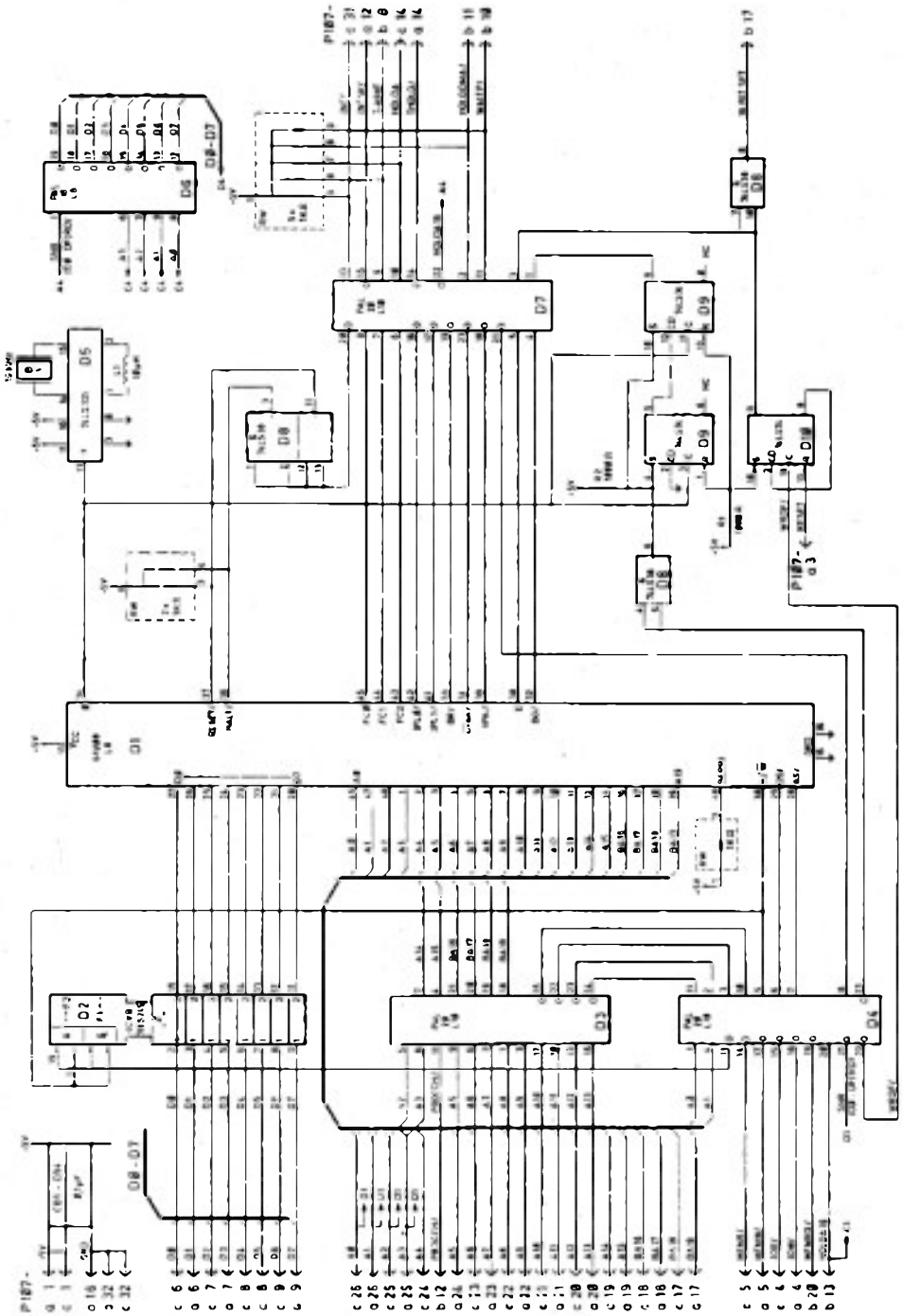
K234 MC 68008 PROCESSORBOARD

Description at a glance:

- The 68008 enables the DM V to handle CP/M 68K incl. C-Compiler
- Using 8-bit databus while providing the benefits of a 32-bit microprocessor architecture
- Code compatible to the 68000
- In the DM V environment as much 512 Kbyte linear adress space
- The 68008 operates with a 8 MHz clock frequency
- Memory mapped I/O

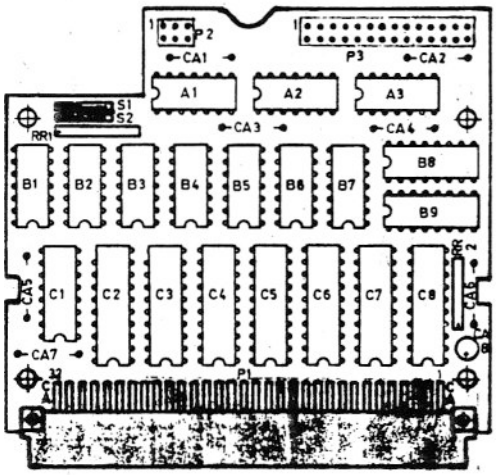
PARTS LIST of K 234 : 68008 processor-board

REQD	PART DESCRIPTION		
1	PC-BOARD		
1	68008	32-BIT PROCESSOR	D1
2	74LS74	DUAL-D-FF	D9,10
1	74LS321	CRISTAL-CONTROLL. OSCILLATOR	D5
1	74LS245	OCTAL-BUS-TRANSCEIVER	D2
1	74LS38	QUAD-NAND-BUFFER	D8
1	PAL, programmed		D4
1	PAL, programmed		D7
1	8*1 kOHM RESISTOR NETWORK		RW
2	1.0 kOHM RESISTOR		R1,2
1	6.0 MHZ	CRISTAL	B1
4	.1 UF/50V	" -C.	CB1,2,3,4
1	COIL		L1
1	3*32POL	CONNECTOR-PLUG	P1
1	40POL	SOCKET, IC	FOR D1
2	24POL	"	FOR D4,7



68008 Processor Board (K234) 017-0033573 Rev. A

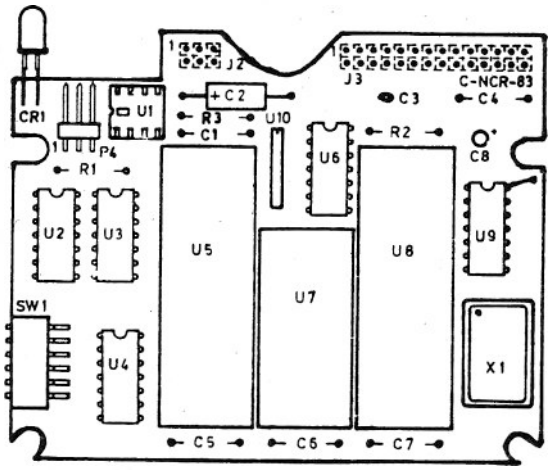
OMNINET ADAPTER
(K 600)



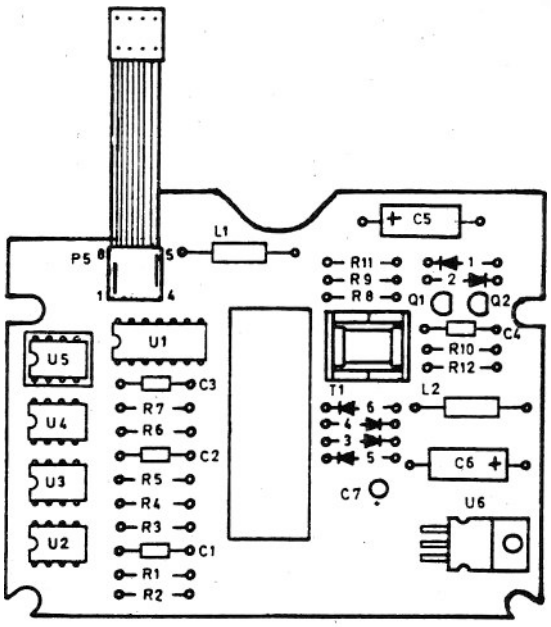
Interface PCB

a		c
+5V	1	+5V
PERC/	2	
RESET/	3	
IOW/	4	IOR/
MEMW/	5	MEMR/
BD1	6	BD0
BD3	7	BD2
BD5	8	BD4
BD7	9	BD6
READYDMA	10	ABTRI/
	11	IDSEL/
	12	
AUTO/	13	DIR/
THOLD/	14	HLDA/
	15	
	16	
BA19	17	BA18
BA17	18	BA16
BA15	19	BA14
BA13	20	BA12
BA11	21	BA10
BA9	22	BA8
BA7	23	BA6
BA5	24	BA4
BA3	25	BA2
BA1	26	BA0
IFSEL3/	27	IFSEL2/
	28	
DRQ1	29	
DACK1/	30	
	31	INT/
LGRD	32	LGRD

Pin assignments P1



Controller PCB



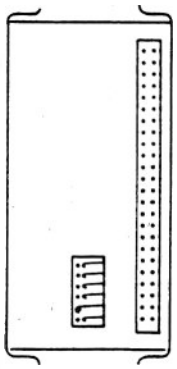
Isolator PCB

K600

PIN ASSIGNMENTS & STRAPPING

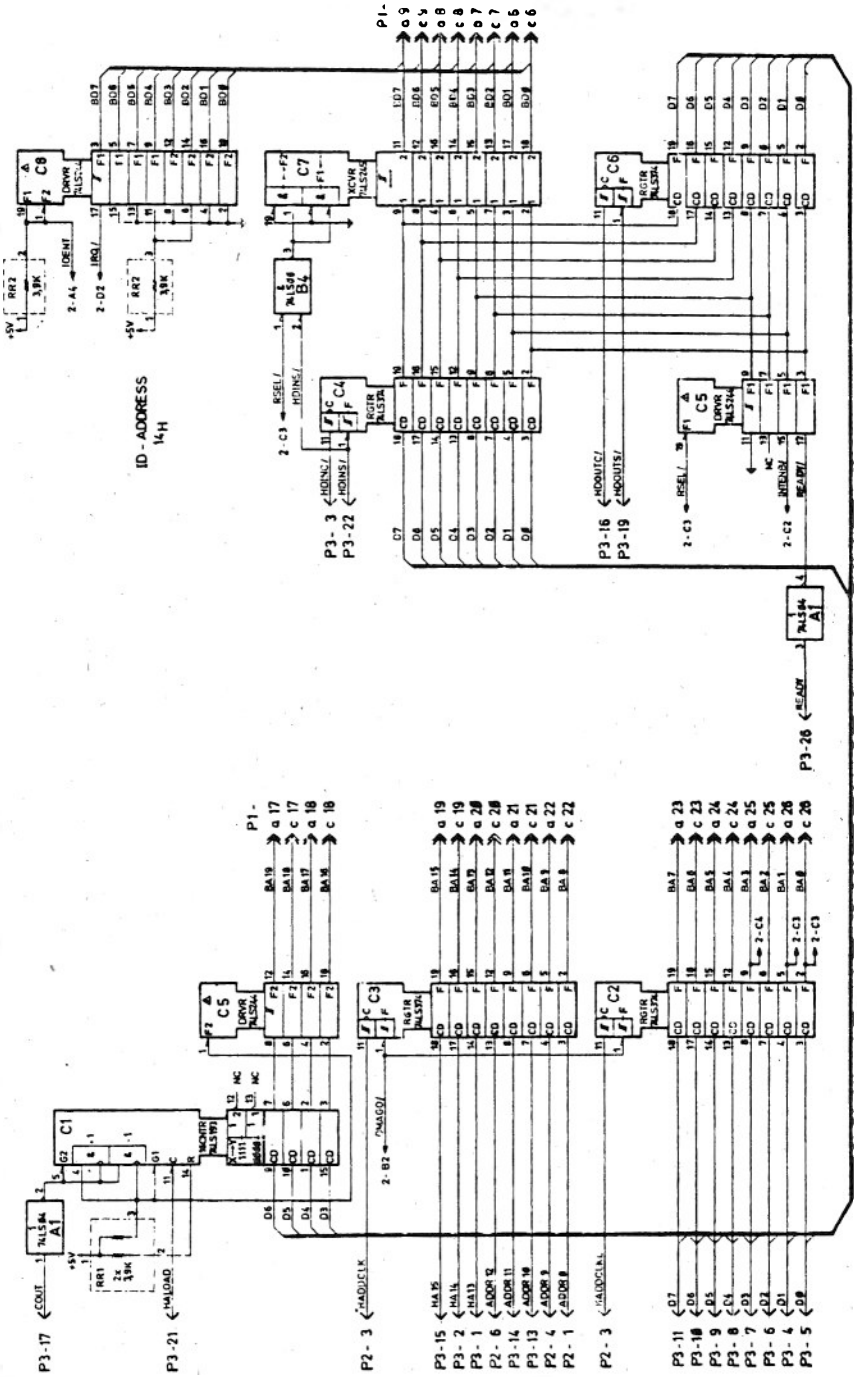
REQUESTER ID		SWITCH SETTING	REQUESTER ID		SWITCH SETTING
HEX	DEC	1 2 3 4 5 6	HEX	DEC	1 2 3 4 5 6
0	0	0 0 0 0 0 0	20	32	0 0 0 0 0 1
1	1	1 0 0 0 0 0	21	33	1 0 0 0 0 1
2	2	0 1 0 0 0 0	22	34	0 1 0 0 0 1
3	3	1 1 0 0 0 0	23	35	1 1 0 0 0 1
4	4	0 0 1 0 0 0	24	36	0 0 1 0 0 1
5	5	1 0 1 0 0 0	25	37	1 0 1 0 0 1
6	6	0 1 1 0 0 0	26	38	0 1 1 0 0 1
7	7	1 1 1 0 0 0	27	39	1 1 1 0 0 1
8	8	0 0 0 1 0 0	28	40	0 0 0 1 0 1
9	9	1 0 0 1 0 0	29	41	1 0 0 1 0 1
A	10	0 1 0 1 0 0	2A	42	0 1 0 1 0 1
B	11	1 1 0 1 0 0	2B	43	1 1 0 1 0 1
C	12	0 0 1 1 0 0	2C	44	0 0 1 1 0 1
D	13	1 0 1 1 0 0	2D	45	1 0 1 1 0 1
E	14	0 1 1 1 0 0	2E	46	0 1 1 1 0 1
F	15	1 1 1 1 0 0	2F	47	1 1 1 1 0 1
10	16	0 0 0 0 1 0	30	48	0 0 0 0 1 1
11	17	1 0 0 0 1 0	31	49	1 0 0 0 1 1
12	18	0 1 0 0 1 0	32	50	0 1 0 0 1 1
13	19	1 1 0 0 1 0	33	51	1 1 0 0 1 1
14	20	0 0 1 0 1 0	34	52	0 0 1 0 1 1
15	21	1 0 1 0 1 0	35	53	1 0 1 0 1 1
16	22	0 1 1 0 1 0	36	54	0 1 1 0 1 1
17	23	1 1 1 0 1 0	37	55	1 1 1 0 1 1
18	24	0 0 0 1 1 0	38	56	0 0 0 1 1 1
19	25	1 0 0 1 1 0	39	57	1 0 0 1 1 1
1A	26	0 1 0 1 1 0	3A	58	0 1 0 1 1 1
1B	27	1 1 0 1 1 0	3B	49	1 1 0 1 1 1
1C	28	0 0 1 1 1 0	3C	60	0 0 1 1 1 1
1D	29	1 0 1 1 1 0	3D	61	1 0 1 1 1 1
1E	30	0 1 1 1 1 0	3E	62	0 1 1 1 1 1
1F	31	1 1 1 1 1 0	3F	63	1 1 1 1 1 1

Requester switch settings



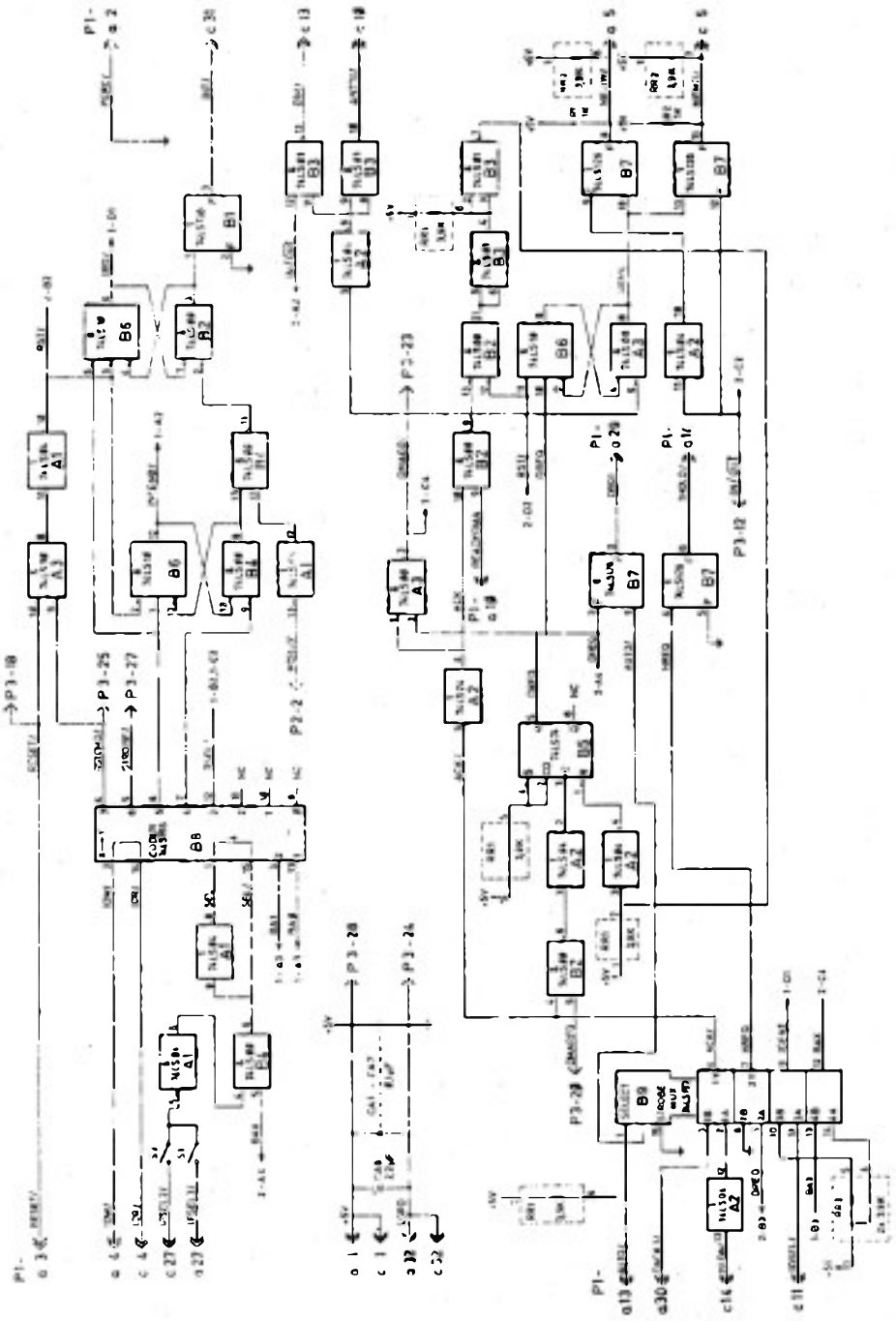
Transporter Switches
(Unit Identification Switches)

K600



Omninet Interface (K600) 017-0032433 Rev. C
1 of 2

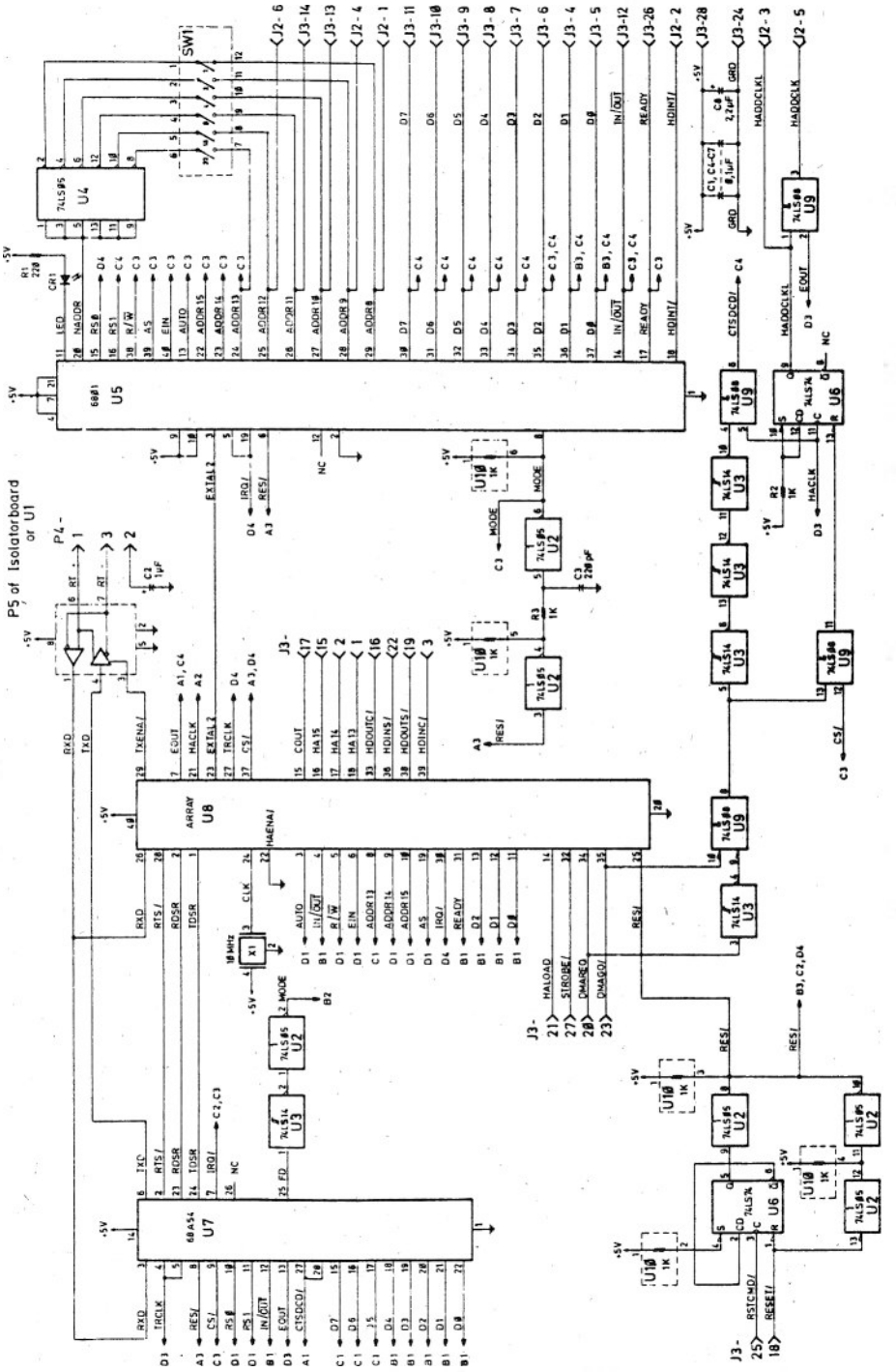
K600



2 of 2

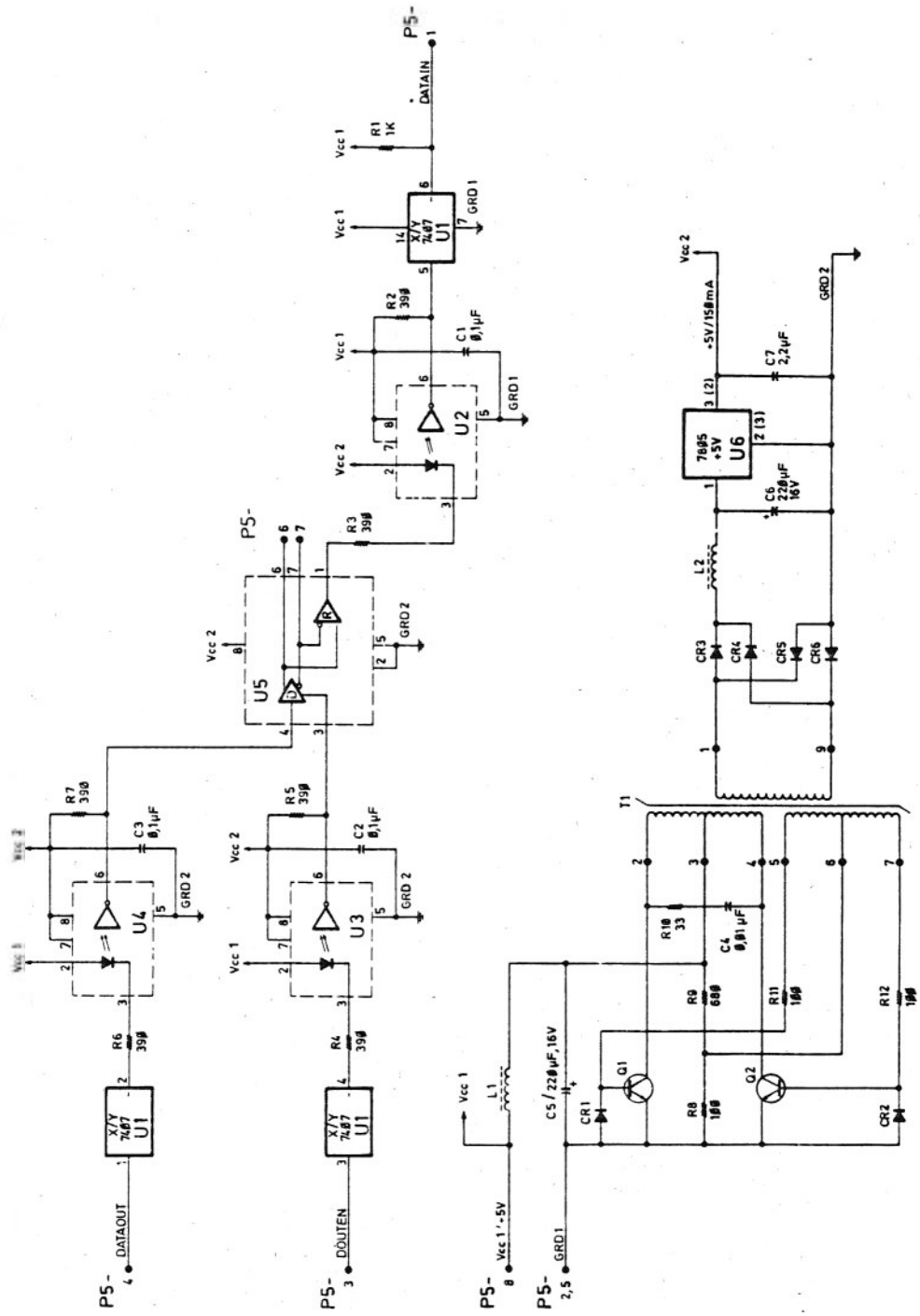
Interface K600

SCHEMATICS
Section 8



Omninet Controller (K600) 017-0032434 Rev. B

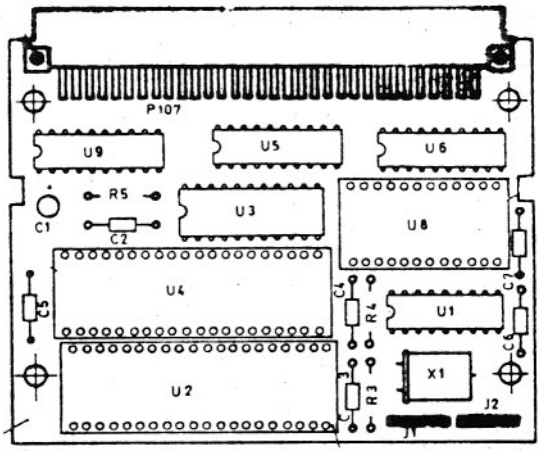
K600



Omninet Isolator (K600) 017-0032222 Rev. A

K600

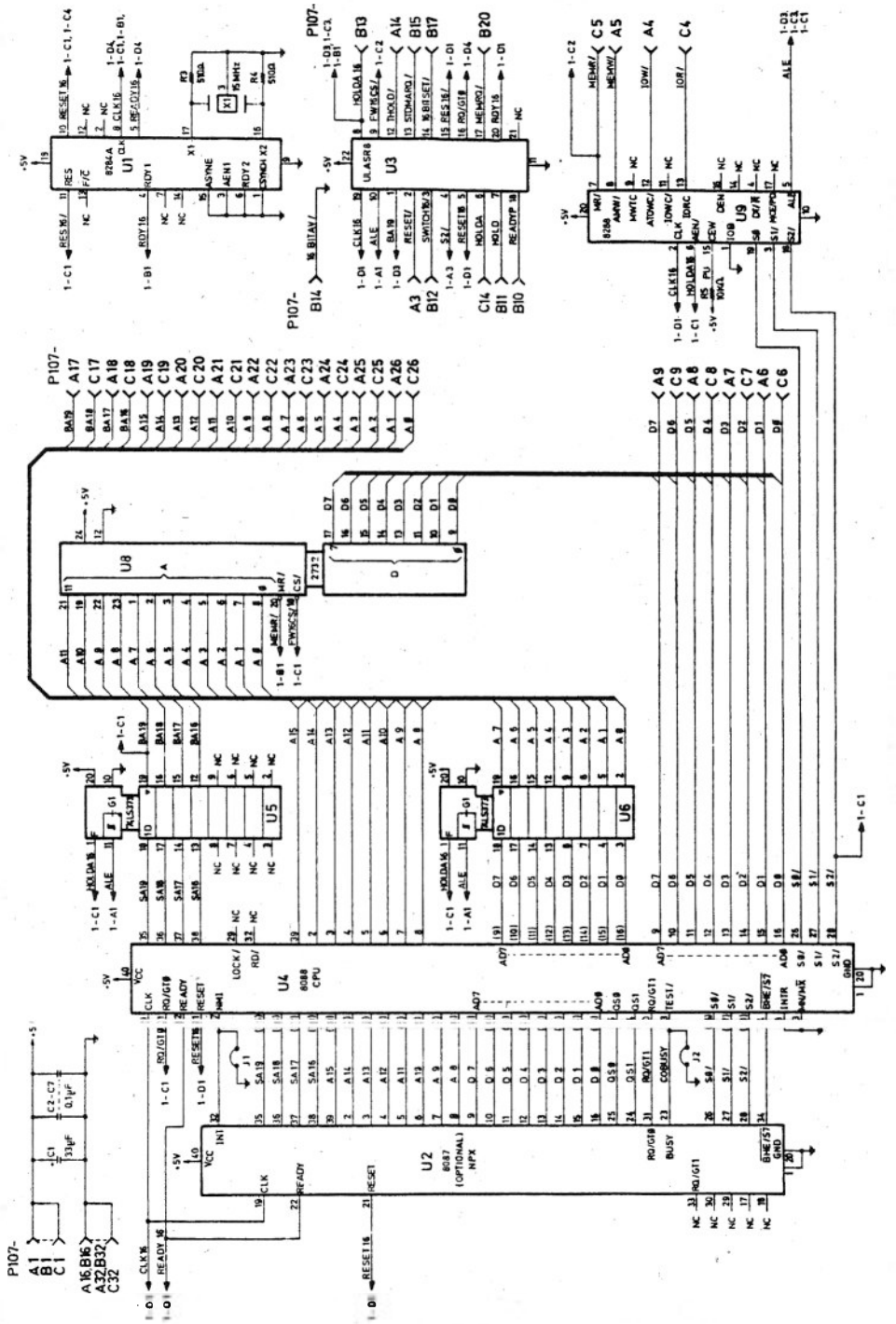
EXTERNAL 16-BIT PROCESSOR (K 231)



J1	closed
J2	closed

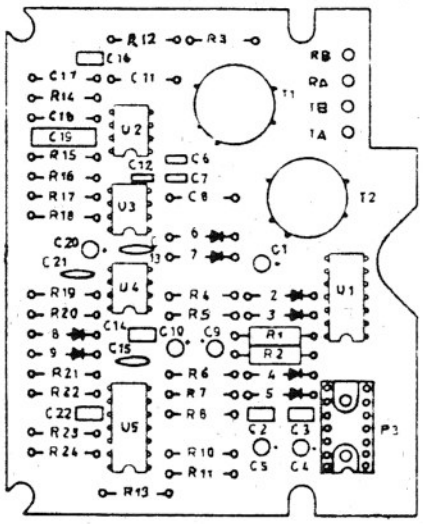
	a	b	c
1	+5 V	+5 V	+5 V
2	OPT 2		+12 V
3	RESET/		RESETIN/
4	IOW/		IOR/
5	MEMW/		MEMR/
6	D1		D0
7	D3		D2
8	D5		D4
9	D7		D6
10	READYDMA	READYP	ABTRI/
11	EOP/	HOLD	
12	INTACK/	SWITCH 16/	IFSEL4/
13	DBTRI/	HLDA 16	DIR/
14	THOLD/	16 BITAV/	HLDA
15	PCLK/	STDMARQ/	CLK1
16	LGRD	LGRD	TRAMD/
17	BA19	16 BITSET/	BA18
18	BA17		BA16
19	A15		A14
20	A13	MEMRQ/	A12
21	A11		A10
22	A9		A8
23	A7		A6
24	A5		A4
25	A3		A2
26	A1		A0
27	IFSEL3/		IFSEL2/
28	IFSEL1/		IFSEL0/
29	DRQ1		DRQ0
30	DACK1/		DACK0/
31	WAIT/		INT/
32	LGRD	LGRD	LGRD

Pin assignments J107/107A (diagnostics/16-bit processor)

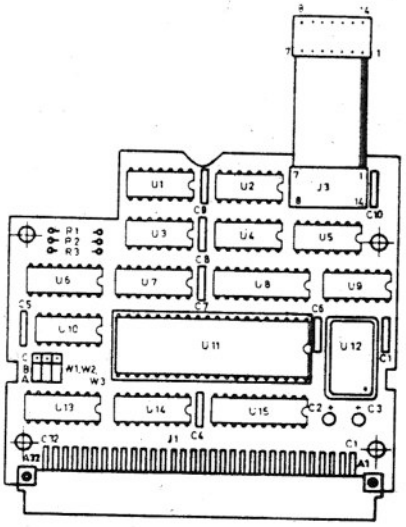


External 16-bit Processor (K231) 017-0032548 Rev. A

DLC INHOUSE INTERFACE (K223)

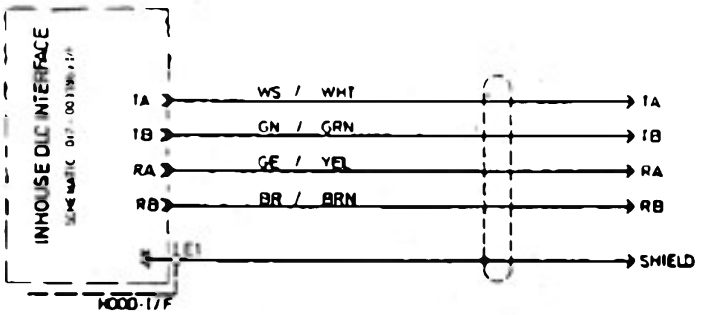


DLC Inhouse I/F, board assy



DLC inhouse controller, board assy

K223



Header	W1	W2	W3	Port Address
IFSEL 2 IFSEL 3	A - B A - B	A - B A - B	A - B B - C	30 - 3F (Hex) 80 - BF (Hex)

DLC inhouse controller

1	+5V
2	TSTART/
3	4MSINT
4	LED
5	GND
6	GND
7	GND
8	RDM
9	4MSSTOP/
10	+12V
11	+12V
12	TDM/
13	TDM
14	+5V

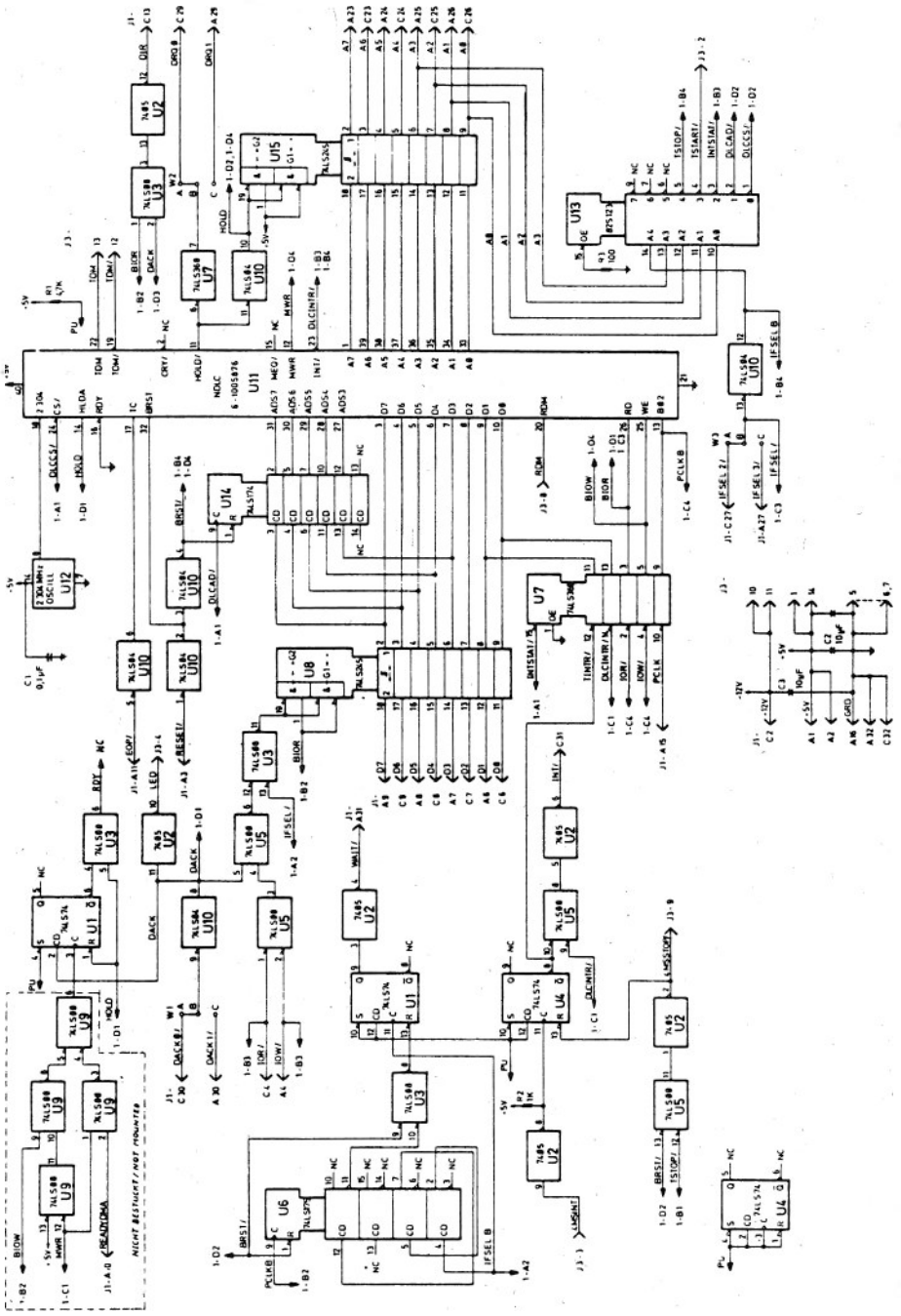
Pin assignments J3
(DLC-Inhouse Controller)

K223

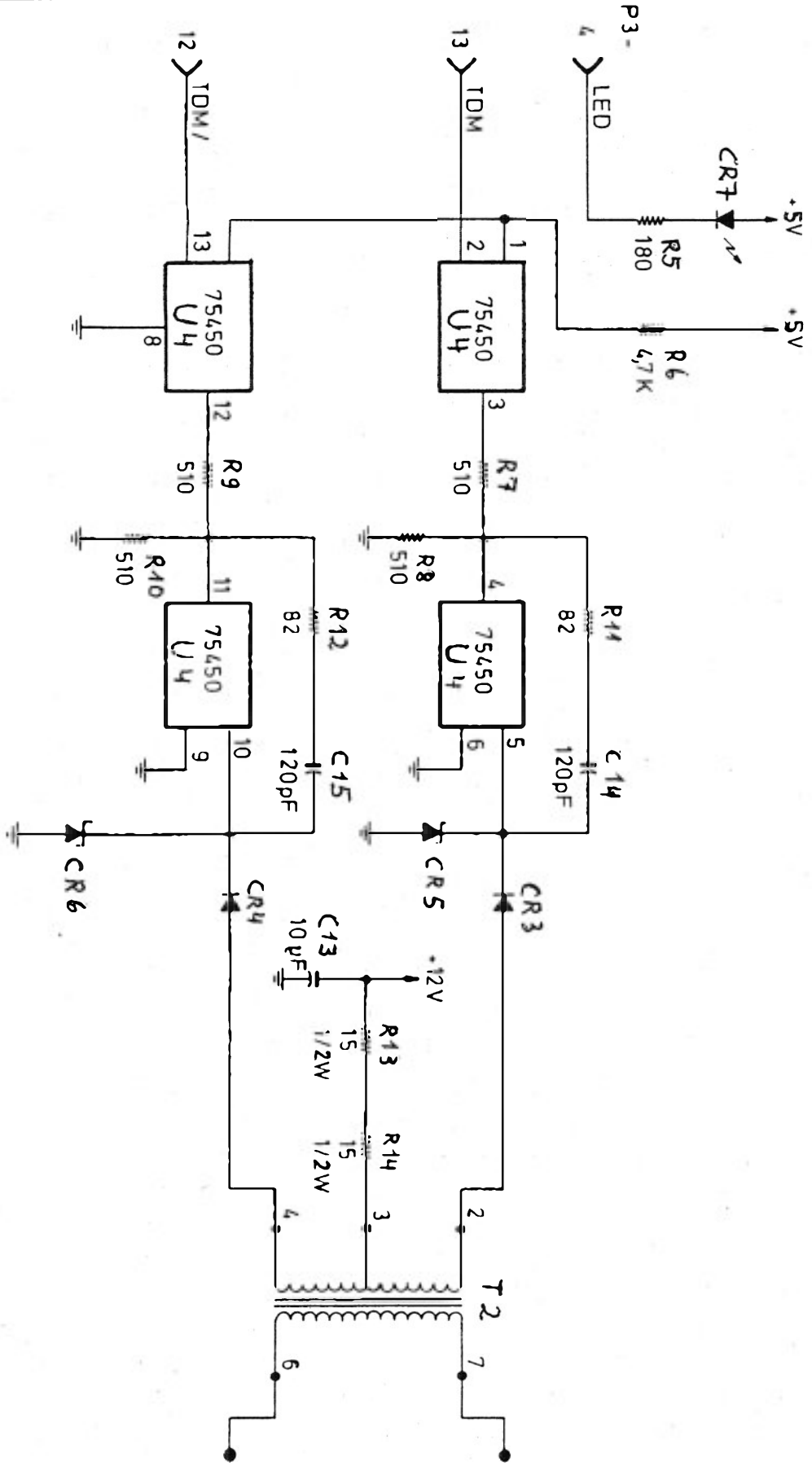
PIN ASSIGNMENTS & STRAPPING

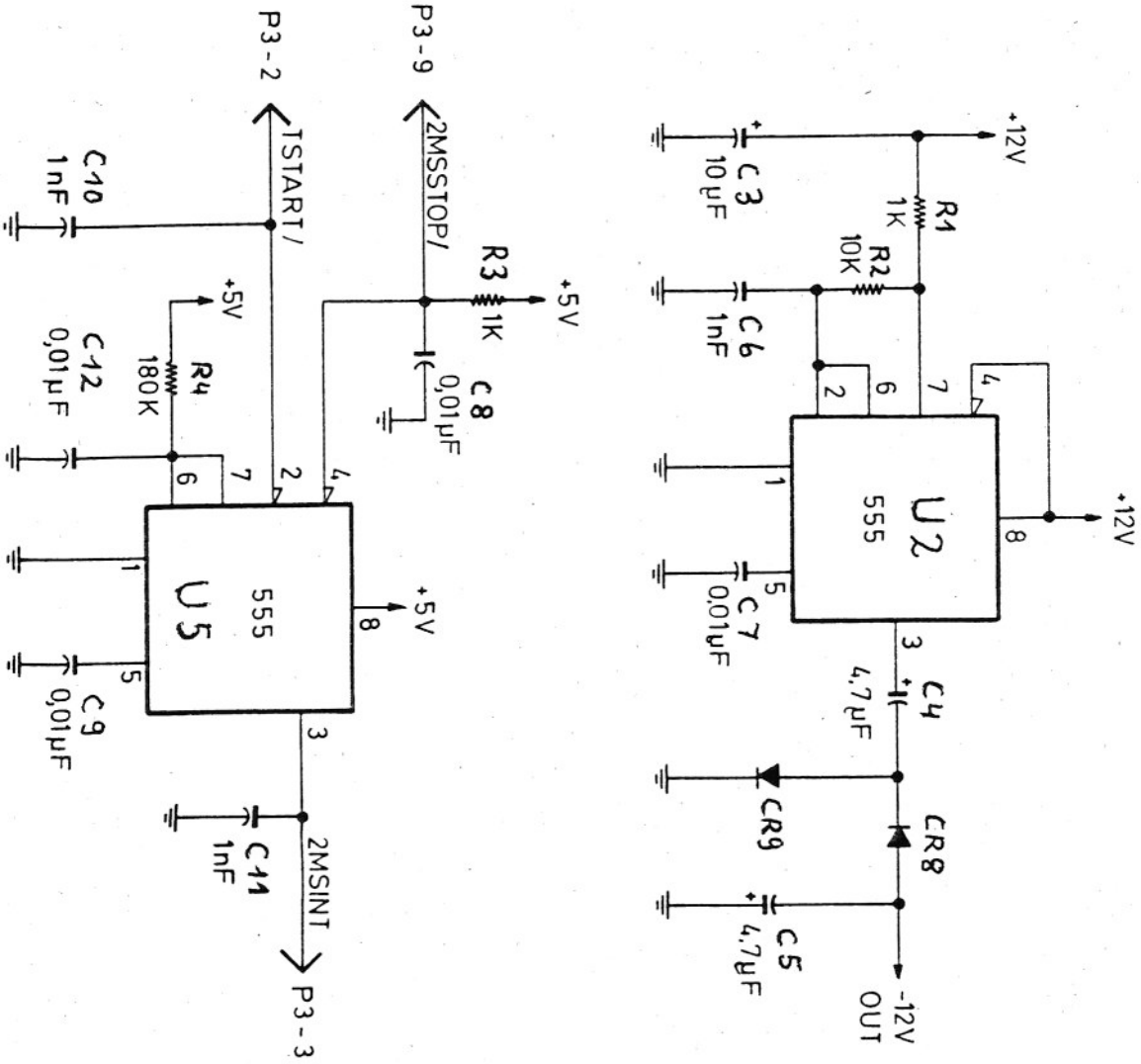
a		c
+5V	1	
+5V	2	+12V
RESET/	3	IFSEL/
IOW/	4	IOR/
	5	
D1	6	D0
D3	7	D2
D5	8	D4
D7	9	D6
READYDMA	10	
EOP/	11	
	12	
	13	DIR
	14	
PCLK	15	
GND	16	
	17	
	18	
	19	
	20	
	21	
	22	
A7	23	A6
A5	24	A4
A3	25	A2
A1	26	A0
IFSEL3/	27	IFSEL2/
	28	
DRQ1	29	DRQ0
DACK1/	30	DACK0/
WAIT	31	
GND	32	GND

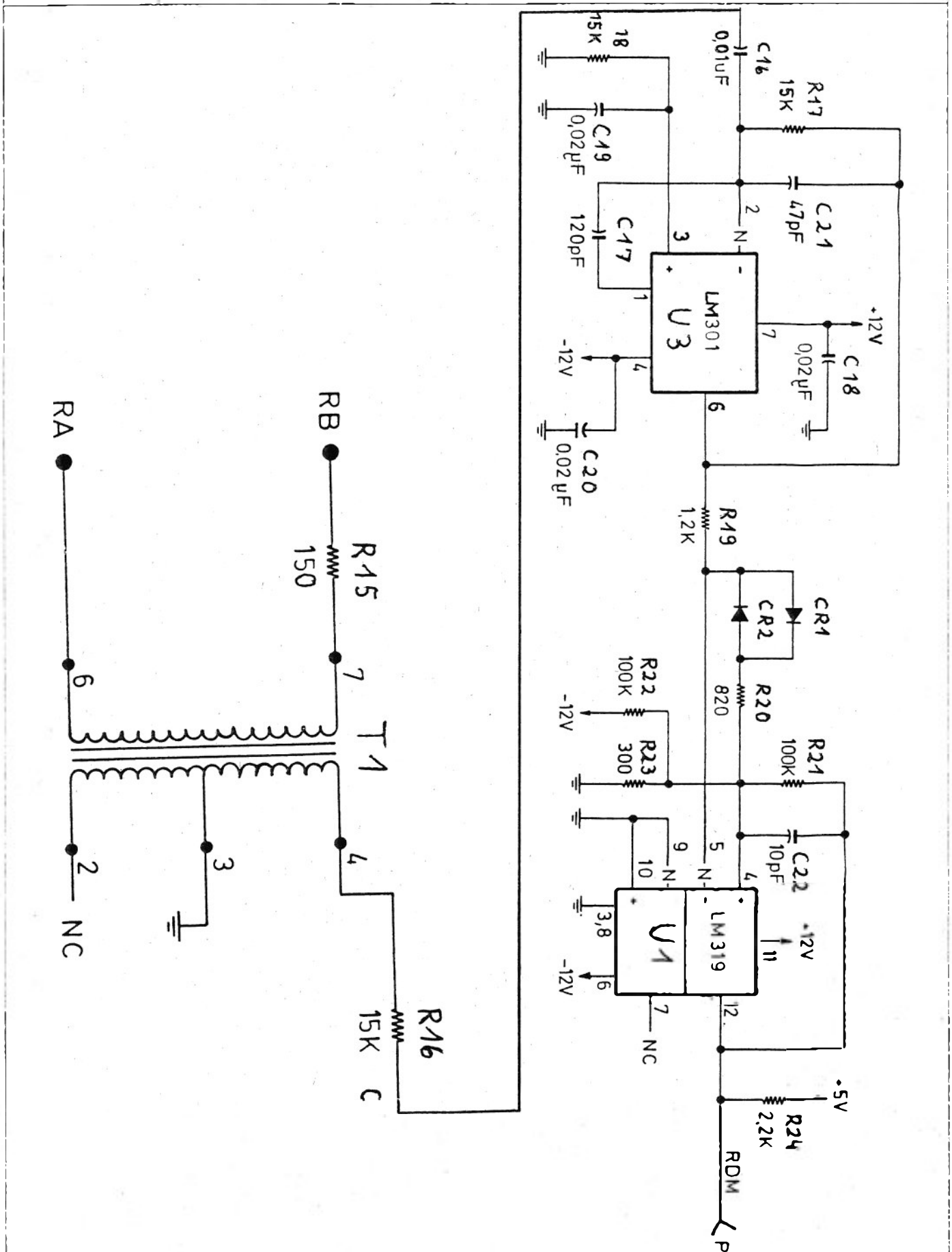
Pin assignments J1
(DLC-Inhouse Controller)



DLC Inhouse Interface (K223) 017-0033972 Rev. C







INTERFACE SELECTION

NCR-DECISION MATE V Interface Configuration

Interface Selects — Family:

IFSEL/ PIN's on DM V BUS	0A 0B IFSEL 0/ c28	1A 1B IFSEL 1/ a28	2A 2B IFSEL 2/ c27	3A 3B IFSEL 3/ a27	4A 4B IFSEL 4/ c12	SLOT	DMA CHANNEL 0 or 1
PRINTER 1 serial/parallel K212 K210	■					2-8	
PLOTTER (See Note 2) K213	■					2-8	
COMMUNICATION K211		■				2-8	
SWITCHABLE RS-232C K801	■	■	■ (See Note 1)	■	■	2-6	
PLOTTER (See Note 2) K801	■		(WITH CP/M or MS-DOS)			2-6	
PLOTTER (See Note 2) K801			(WITH p-SYSTEM) ■			2-8	
BUFFERED SYNC/ ASYNC (See Note 3) K215		■				2-8	
REAL-TIME-CLOCK (See Note 3) K803					■	2-6	
IEEE 488 (See Note 3) K804			■			2-6	
MOUSE INTERFACE (See Note 3) K806			■			2-6	
DECISION NET K600				■		2-6	1
HARD DISK ext. (NCR 3282)					■	2-6	
HARD DISK int.					■	2-A	
PORT ADDRESS	60H 67H / 68H 6FH	70H 77H / 78H 7FH	30H 37H / 38H 3FH	80H 87H / 88H 8FH	C0H C7H / C8H CFH		

No IFSEL for 16-Bit – Ext., Diagnost., Memory 64K, 192K, 448K.

■ - Software and Hardware, ■ - Hardware.

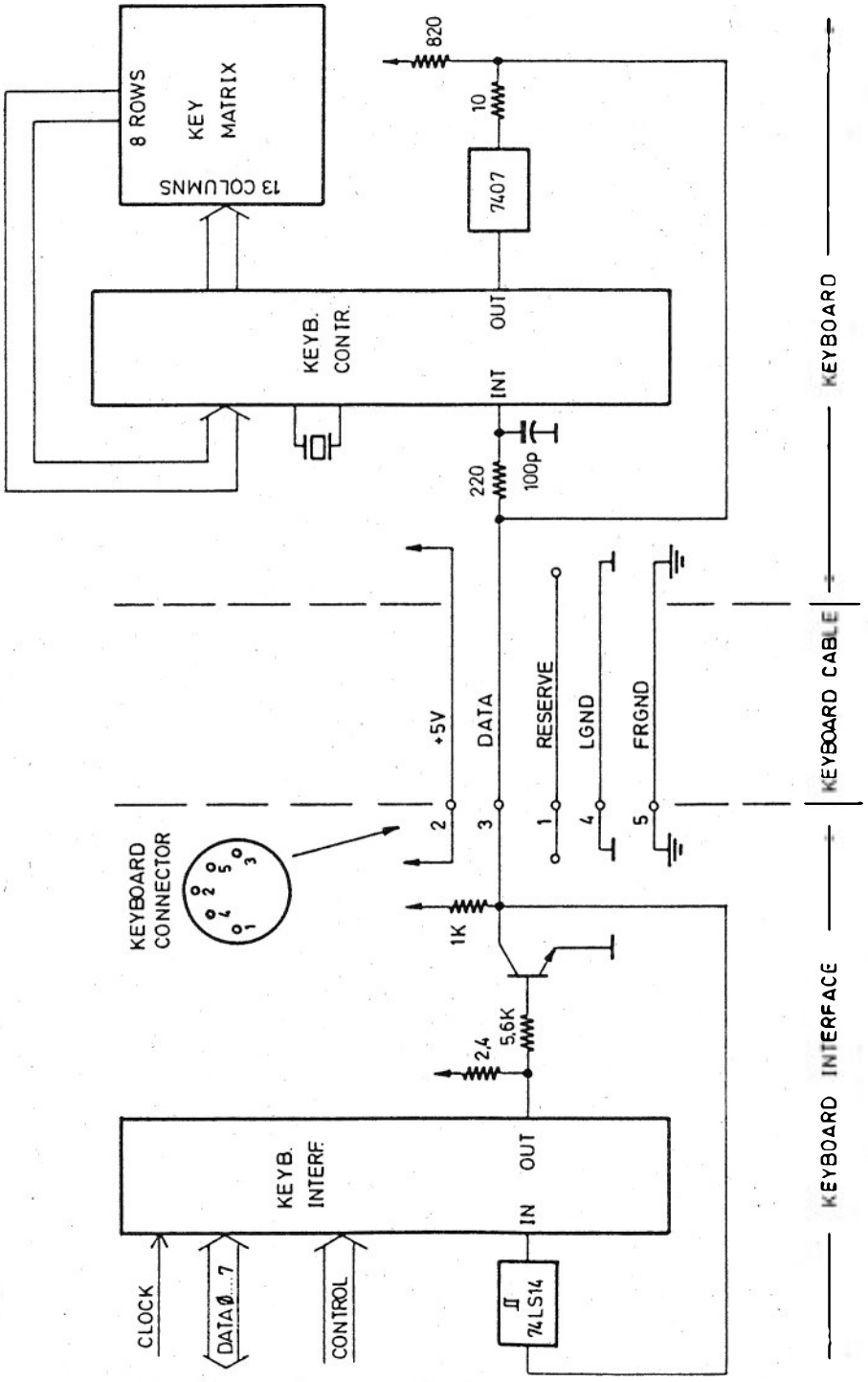
IPORTANT!

Note 1: Must be swappable by hardware and software
Note 2: Use either K213 or K801
Note 3: Any software must be able to select all 10 port addresses
Note 4: Any port address cannot be used more than once in a system

Interface selection

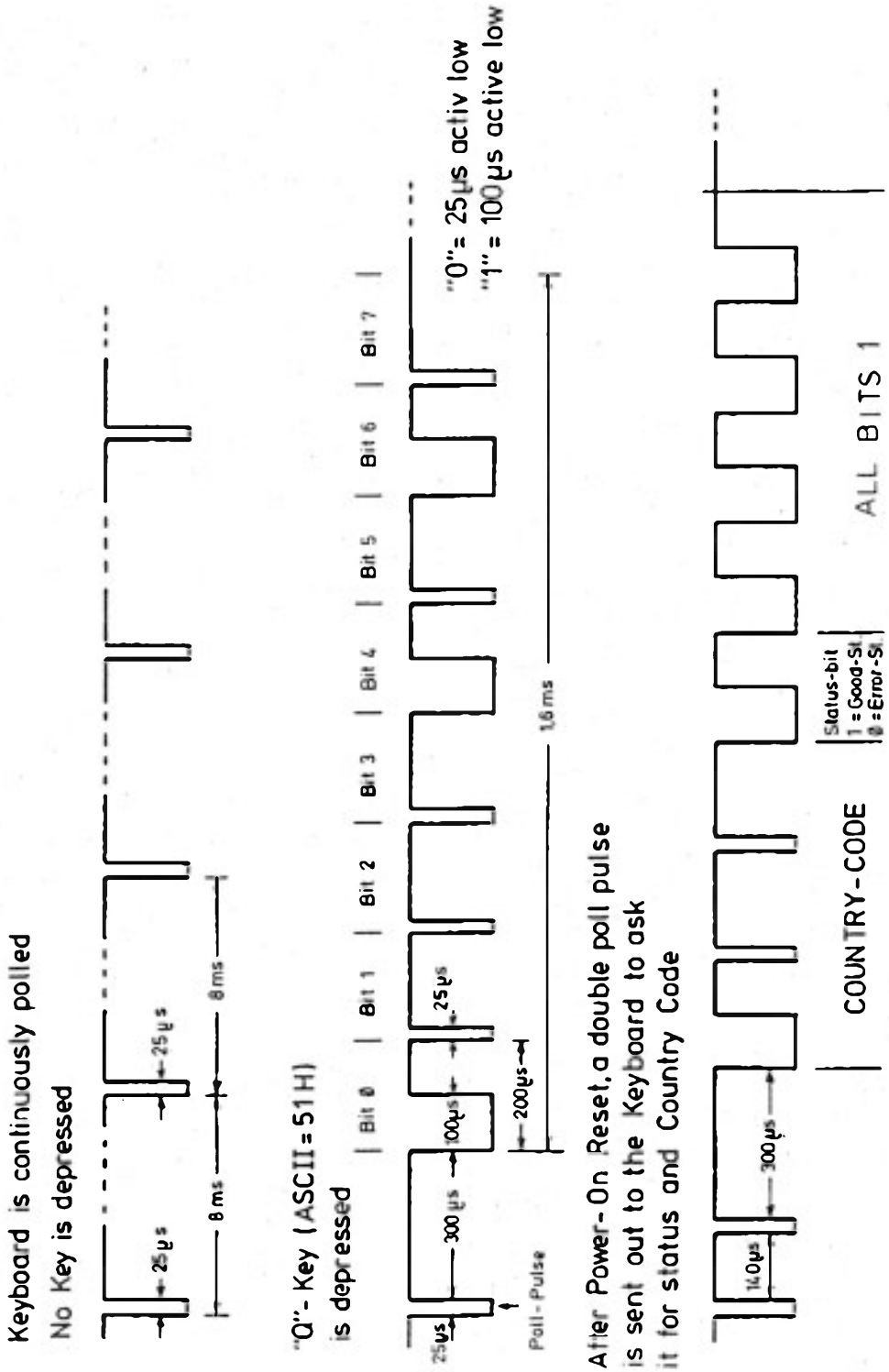
You can also use the IFSEL5 Pin c11
Port address F0 to FF
But: Only by multilayer mainboard

KEYBOARD INTERF AND KEYBOARD

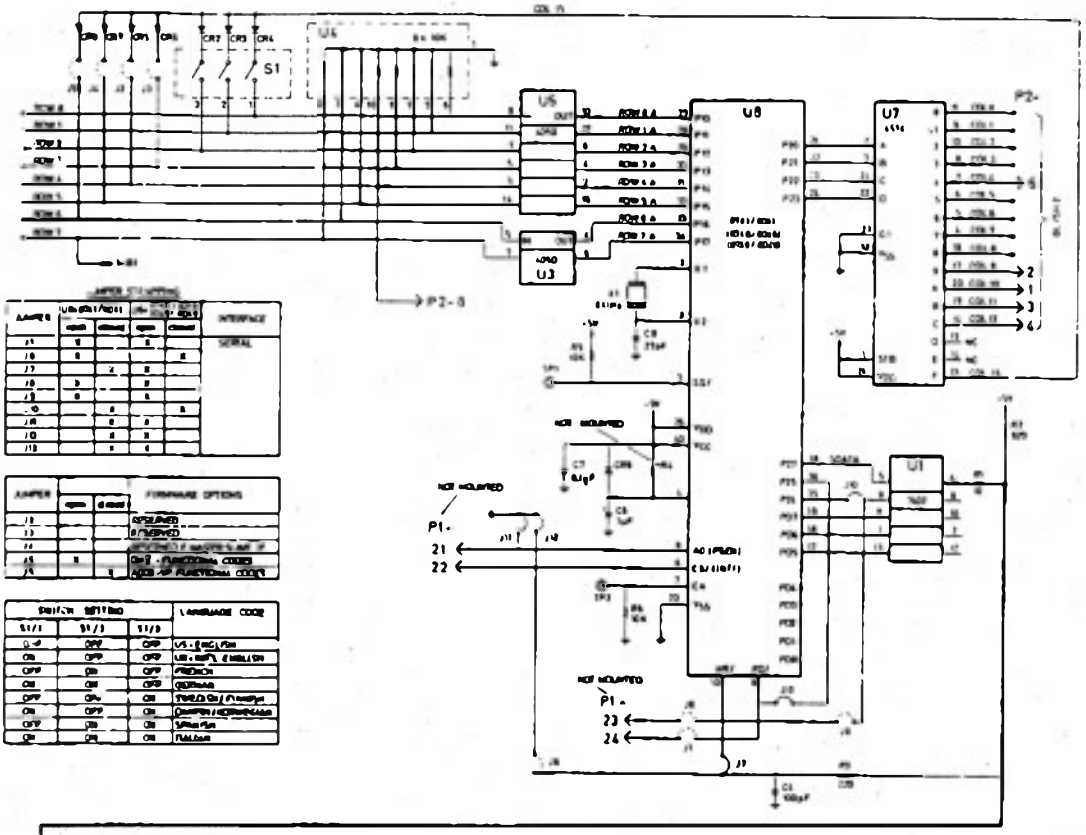


TIMING DIAGRAM

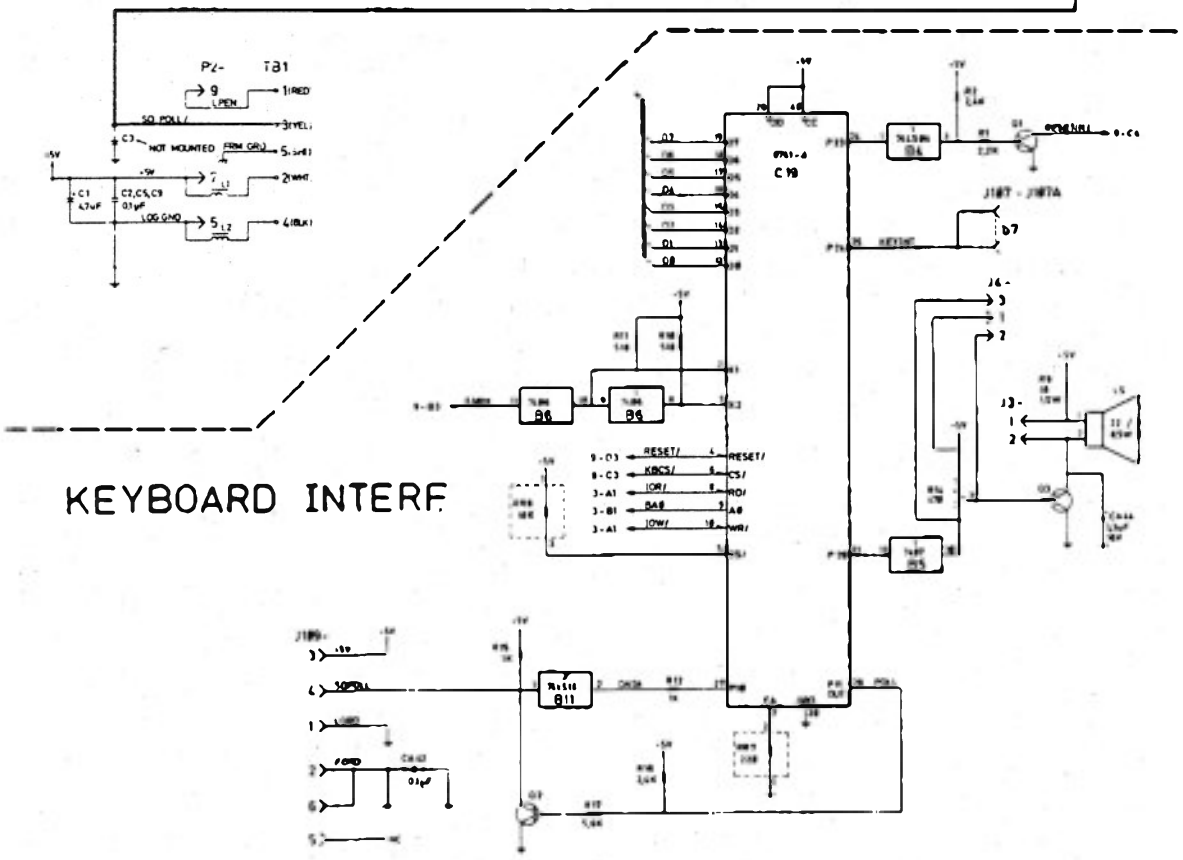
KEYBOARD POLLING AND DATA TRANSFER



KEYBOARD



KEYBOARD INTERF



PAGE 1

```
1 $ MOD41 DEBUG MACRODFILE PAGEWIDTH(106) PAGELENGTH(72) PRINT(:LP:) XREF
2 ;
3 ;
4 $       TITLE ('KEYBOARD-INTERFACE AMF-DMINTF0102-00')
5 ;
6 ;
7 ;       8741-FIRMWARE FOR KEYBOARD-INTERFACE
8 ;       ************************************
9 ;                SOUND-GENERATOR
10 ;               ***************
11 ;
12 ;
13 ;
14 ;
15 ;
16 ;      DATE : 08.07.83
17 ;
18 ;      VERSION: 03.00
19 ;
20 ;
21 ;
22 ;      COPYRIGTH BY NCR 1983
23 ;
```

```
25 ;
26 ;
27 ;                *** FIRMWARE MODIFICATION ***
28 ;                    ----------------------
29 ;
30 ;
31 ;       V 2.1 / 11.03.83          :RESET + FIFO
32 ;
33 ;
34 $       EJECT
```

```
                35 ;
                3A ;
                37 ;        *** REGISTER-DEFINITIONS ***
                38 ;            ------------------------
                39 ;
                40 ;        R0      :  POINTER-REGISTER
                41 ;        R1      :  WAIT COUNTER
                42 ;        R2      :  BIT COUNTER
                43 ;        R3      :  CONVERTED KEYINFO AND TONE FREQ.
                44 ;        R4      :  TONE FREQUENCY , SAVED
                45 ;        R5      :  TONE FREQUENCY , DELAY COUNTER
                46 ;        R6      :  TONE FREQUENCY
                47 ;        R7      :  TONE LENGTH,POINTER FOR LVAR
                48 ;
                49 ;        R0'     :  FIFO-OUT-POINTER
                50 ;        R1'     :  FIFO-IN -POINTER
                51 ;
                52 ;
                53 ;        ***FLAGS***
                54 ;           ------
                55 ;
                56 ;        F0      :  TONE3-FLAG
                57 ;
                58 ;
                59 ;
                60 ;
                61 ;        *** PORTS ***
                62 ;            -------
                63 ;
                64 ;        P10     :  SERIAL DATA IN
                65 ;        P11     :  POLL OUT
                66 ;        P20     :  TONE OUT
                67 ;        P24     :  KEYINT
                68 ;        P23     :  RESET
                69 ;
                70 ;
                71 ;
                72 ;        *** EQU-TABLE ***
                73 ;            ------------
                74 ;
003F            75          MEND    EQU     63      ;END OF RAM-AREA
00C0            76          ROMER   EQU     0C0H    ;STATUS FOR ROM-ERROR
0080            77          RAMER   EQU     80H     ;STATUS FOR RAM-ERROR
0055            78          OUT     EQU     55H     ;RAM-/ROM-TEST OK
                79 ;
                80 $        EJECT
```

```
                        81 ;
0000 0485               82 START:  JMP      INIT
                        83 ;
0007                    84         ORG      7H
0007 04EC               85         JMP      ISRTIN
                        86 ;
                        87 ;
0008                    88         ORG      8H
                        89 ;
0009 2301               90 INIT:   MOV      A,#00000B01B    ;INITIALISE PORTS
000B 39                 91         OUTL     P1,A              FOR INPUT
000C 23FF               92         MOV      A,#0FFH
000E 3A                 93         OUTL     P2,A
000F F5                 94         EN       FLAGS           ;ENABLE MASTER INTERRUPT
0010 65                 95         CLR      F0
                        96 ;
0011 0611               97 START1: JNIBF    START1          ;WAIT OF MASTER-START
0013 4400               98         JMP      COMMAN          ;START SELFTEST
                        99 ;
                       100 ;*************************************************
                       101 ;
                       102 ;         *** RAM-TEST ***
                       103 ;             ----------
                       104 ;*************************************************
                       105 ;
0015 B83F              106 TEST1: MOV      R0,#MEND        ;START RAM-TEST
0017 B93C              107        MOV      R1,#MEND-3
0019 23FF              108 T1:    MOV      A,#0FFH
001B A0                109        MOV      @R0,A
001C F0                110        MOV      A,@R0
001D 17                111        INC      A
001E 962E              112        JNZ      ERR01
0020 C8                113        DEC      R0
0021 E919              114        DJNZ     R1,T1
                       115 ;
0023 27                116        CLR      A
0024 B83F              117        MOV      R0,#MEND
0026 A0                118 T2:    MOV      @R0,A           ;CLEAR RAM
0027 F0                119        MOV      A,@R0
0028 962E              120        JNZ      ERR01
002A E626              121        DJNZ     R0,T2
002C 040C              122        JMP      ROTST           ;RAM-TEST OK
                       123 ;                                 JUMP TO ROM-TEST
002E 2380              124 ERR01: MOV      A,#RAMER        ;RAM-ERROR DEDECTED
0030 0434              125        JMP      OK+2
                       126 ;
0032 2355              127 OK:    MOV      A,#OUT
0034 02                128        OUT      DBB,A           ;SEND STATUS TO MASTER
0035 0635              129 W1:    JOBF     W1
                       130 ;
                       131 $      EJECT
```

```
                    133 ;
0037 BF3F           134         MOV     R7,#MEND        ;SET ADDRESS-POINTER FOR
                    135 ;                                   "LAENDER-VARIANTE"
0039 BA0B           136 CONV:   MOV     R2,#0BH
003B BB2B           137         MOV     R0,#40          ;ENDADDRESS OF TABLE
003D 2380           138         MOV     A,#80H
003F 97             139         CLR     C
0040 A8             140 CONV1:  MOV     @R0,A
0041 C8             141         DEC     R0
0042 67             142         RRC     A
0043 E640           143         JNC     CONV1
                    144 ;
0045 D5             145         SEL     RB1             ;INIT FIFO-POINTER
0046 B82A           146         MOV     R0,#42
0048 B92A           147         MOV     R1,#42
004A C5             148         SEL     RB0
                    149 ;
004B 4400           150         JMP     OPOLL           ;SEND BEEP FOR TEST PASSED
                    151 ;                                   AND GET LVAR
                    152 LVAR:
004D A8             153         MOV     R0,A
004E FB             154         MOV     A,R3            ;(R3) = LAENDER-VARIANTE
004F 37             155         CPL     A
0050 7254           156         JB3     LVAR1           ;JUMP IF KEYBOARD ERROR
0052 37             157         CPL     A
0053 A0             158         MOV     @R0,A           ;STORE COUNTRY TYPE
0054 FB             159 LVAR1:  MOV     A,R3
0055 02             160         OUT     DBB,A           ;SEND STATUS
                    161 ;LVAR2: JOBF    LVAR2           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                    162 ;
0056 BB6F           163 BELL1:  MOV     R3,#6FH         ;PREPARE REGISTER FOR 880 HZ
0058 BC6F           164         MOV     R4,#6FH
005A B800           165         MOV     R5,#0                       BEEP
005C BF1E           166         MOV     R7,#30          ;LENGTH = 500MS
005E 4464           167         JMP     TONE3
                    168
                    169 ;
                    170 ;*****************************************************************
                    171 ;
                    172 ;
                    173 ;       *** SEND POLL ***
                    174 ;
                    175 ;*****************************************************************
                    176 ;
                    177 ;
0060 B821           178 POLL:   MOV     R0,#21H         ;SET POINTER-REGISTER TO START
0062 BA08           179         MOV     R2,#08H         ;SET BIT-COUNTER
0064 BB00           180         MOV     R3,#00H
0066 B902           181         MOV     R1,#02
0068 3903           182         ORL     P1,#000000118   ;POLL-OUT
006A 14E9           183         CALL    WAIT            ;WAIT 25US
006C 9901           184         ANL     P1,#000000018
                    185 ;
                    186 ;
```

```
                        *** WAIT FOR CHARACTER-BIT ***

006E 2398          196 TIME1:  MOV     A,#098H
0070 62            197         MOV     T,A             ;SET TIMER TO 6MS
0071 25            198         EN      TCNTI           ;ENABLE TIMER
0072 55            199         STRT    T
                   200 ;
0073 09            201 TI2:    IN      A,P1
0074 1278          202         JB0     TI3             ;JUMP IF DATA-BIT
0076 0473          203         JMP     TI2
                   204 ;
0078 35            205 TI3:    DIS     TCNTI           ;DISABLE TIMER
0079 65            206         STOP    TCNT
007A B905          207         MOV     R1,#05
007C 14E9          208 DIS:    CALL    WAIT            ;WAIT 60US
007E 09            209         IN      A,P1            ;READ DATA-BIT
007F 1283          210         JB0     BIT1            ;JUMP IF DATA-BIT=1
0081 0486          211         JMP     BIT0            ;JUMP IF DATA-BIT=0
0083 FB            212 BIT1:   MOV     A,R3            ;READ BIT-MASK
0084 48            213         ORL     A,R3
0085 2B            214         XCH     A,R3
0086 18            215 BIT0:   INC     R0              ;POINTER + 1
0087 B909          216         MOV     R1,#09H
0089 14E9          217         CALL    WAIT            ;WAIT 65US
008B EA73          218         DJNZ    R2,TI2
                   219 ;
008D FF            220         MOV     A,R7
008E 9640          221         JNZ     LVAR            ;JUMP AND STORE "LAENGER
0090 FB            222         MOV     A,R3                    -VARIANTE"
0091 D3B3          223         XRL     A,#0B3H         ;IF "F20 THEN RESET DMS
0093 C6DB          224         JZ      RESET
                   225 ;
0095 D5            226         SEL     RB1             ;TEST IF FIFO-FULL
0096 F9            227         MOV     A,R1
0097 D33E          228         XRL     A,#62
0099 C6A3          229         JZ      FIFUL
009B C5            230         SEL     RB0
009C FB            231         MOV     A,R3
009D D5            232         SEL     RB1
009E A1            233         MOV     @R1,A           ;STORE INTO FIFO
009F 19            234         INC     R1
00A0 C5            235         SEL     RB0
00A1 04BE          236         JMP     SENFI           ;OUT FIFO-DATA
                   237 ;
00A3 C5            238 FIFUL:  SEL     RB0
00A4 0456          239         JMP     BELL1
                   240 ;
00A6 B5            241 NEXT1:  CLR     F0              ;RESET TONE2-FLAG
00A7 D660          242         JNIBF   POLL            ;JUMP TO NEW POLL IF NO
00A9 76AE          243         JF1     COMM
00AB 22            244         IN      A,DBB               COMMAND FROM MASTER
00AC 0460          245         JMP     POLL            ;IF NO COMMAND
00AE 4400          246 COMM:   JMP     COMMAN
```

```
                        250 ;
00B0 FF                 251 NODAT:  MOV     A,R7
0081 C687               252         JZ      NODAT2
0083 27                 253         CLR     A
0084 02                 254         OUT     DBB,A           ;SEND STATUS "NO KEYB.
0085 0456               255         JMP     BELL1             CONNECTED"
0087 D5                 256 NODAT2: SEL     RB1
0088 F9                 257         MOV     A,R1            ;TEST IF FIFO-EMPTY
0089 D32A               258         XRL     A,#42
008B C5                 259         SEL     RB0
008C C6A6               260         JZ      NEXT1           ;JUMP IF FULL
008E 66A6               261 SENFT:  JOBF    NEXT1
0090 D5                 262         SEL     RB1
0091 F0                 263         MOV     A,@R0           ;GET FIFO-DATA
0092 02                 264         OUT     DBB,A
0093 18                 265         INC     R0
0094 F8                 266         MOV     A,R0
0095 D9                 267         XRL     A,R1
0096 969C               268         JNZ     NODAT1
0098 B82A               269         MOV     R0,#42          ;REINIT FIFO-POINTER
009A B92A               270         MOV     R1,#42
009C C5                 271 NODAT1: SEL     RB0
009D BB6F               272         MOV     R3,#6FH         ;PREPARE REGISTER FOR
009F BC6F               273         MOV     R4,#6FH               BEEP
00A1 B800               274         MOV     R5,#0
00A3 BF02               275         MOV     R7,#02          ;LENGTH = 40MS
00A5 95                 276         CPL     F0              ;SET TONE3-FLAG
00A6 4464               277         JMP     TONE3
                        278 ;
                        279 ;
                        280 ;
                        281 ;       *** RESET ***
                        282 ;           *****
                        283 ;
00A8 9AF7               284 RESET:  ANL     P2,#11110111B   ;RESET DM5
00AA 04A8               285         JMP     RESET
                        286 ;
                        287 ;       *** ROM-TEST ***
                        288 ;           ------------
                        289 ;
                        290 ;*****************************************************
                        291 ;
                        292 ;
00AC 27                 293 ROMTST: CLR     A
00AD AA                 294         MOV     R2,A
00AE A8                 295         MOV     R0,A
00AF A9                 296 T3:     MOV     R1,A                ;ROM-TEST PAGE 3
00B0 A3                 297         MOVP    A,@A
00B1 68                 298         ADD     A,R0
00B2 A8                 299         MOV     R0,A
00B3 F9                 300         MOV     A,R1
00B4 17                 301         INC     A
00B5 EAAF               302         DJNZ    R2,T3
00B7 2400               303         JMP     T4
                        304 ;
```

```
                        316 ;
                        317 ;       *** WAIT (R1)*SUB + SUB  (IF CLOCK = 6 MHZ)
                        318 ;
00E9 E9E9               323 WAIT:   DJNZ    R1,WAIT
00EB 83                 324         RET
                        325 ;

                        331 ;       *** ISR FOR TIMER ***
                        332 ;           -------------
                        333 ;
00EC 65                 339 ISRTIM: STOP    TCNT
00ED 35                 340         DIS     TCNTI
                        341 ;
00EE B908               342 ISRT1:  MOV     R1,#08
00F0 2380               343         MOV     A,#LOW NODAT
00F2 A1                 344         MOV     @R1,A
00F3 19                 345         INC     R1
00F4 C7                 346         MOV     A,PSW
00F5 53F0               347         ANL     A,#0F0H
00F7 4308               348         ORL     A,#HIGH NODAT
00F9 A1                 349         MOV     @R1,A
00FA 93                 350         RETR
                        351 ;
                        352 ;
0100                    356         ORG     100H
                        357 ;
0100 A9                 358 T4:     MOV     R1,A                    ;ROM-TEST PAGE 1
0101 A3                 359         MOVP    A,@A
0102 68                 360         ADD     A,R0
0103 A8                 361         MOV     R0,A
0104 F9                 362         MOV     A,R1
0105 17                 363         INC     A
0106 EA00               364         DJNZ    R2,T4
0108 4420               365         JMP     TEST2
                        366 ;
                        367 ;
                        368 ;
                        369 ;
0200                    370         ORG     200H
                        371 ;
0200 22                 372 COMMAN: IN      A,DBB               ;GET COMMAND
0201 0304               373         ADD     A,#LOW CTAB
0203 B3                 374         JMPP    @A                  ;JUMP TO COMMAND-TABLE
                        375 ;
                        376 ;
                        377 ;       *** COMMAND-TABLE ***
                        378 ;           -------------
                        379 ;
0204 2B                 380 CTAB:   DB      LOW TEST
0205 1C                 381         DB      LOW LNDVAR
0206 00                 382         NOP
0207 00                 383         NOP
0208 00                 384         NOP
0209 00                 385         NOP
020A 46                 386         DB      LOW TONE
020B 1A                 387         DB      LOW BELL
```

```
LOC  OBJ          LINE        SOURCE STATEMENT

                   390 ;
                   391 ;*******************************************************
                   392 ;
                   393 ;
                   394 ;        *** COMMAND'S ***
                   395 ;            ---------
                   396 ;
                   397 ;*******************************************************
                   398 ;
020C B902          399 DPOLL:   MOV     R1,#02H
020E 8903          400          ORL     P1,#00000011B          ;SEND POLL
0210 14E9          401          CALL    WAIT                   ;WAIT 35US
0212 9901          402          ANL     P1,#00000001B
0214 B916          403          MOV     R1,#22                 ;WAIT 140US
0216 14E9          404          CALL    WAIT
0218 0460          405          JMP     POLL
                   406 ;
                   407 ;
021A 0456          408 BELL:    JMP     BELL1                  ;SEND BELL-TONE
                   409 ;
                   410 ;
021C 861C          411 LNGVAR:  JOBF    LNGVAR   ;WAIT TILL OUTPUT-BUFFER IS CLEARD
021E BA3F          412          MOV     R0,#MEND ;MEND = ADR. FOR "LAENDER-VARIANTE"
0220 2380          413          MOV     A,#80H
0222 90            414          MOV     STS,A                  ;SET USER-FLAG
0223 F0            415          MOV     A,@R0
0224 02            416          OUT     DBB,A    ;SEND "LAENDER-VARIANTE"
0225 8625          417 VAR1:    JOBF    VAR1
0227 27            418          CLR     A
0228 90            419          MOV     STS,A                  ;CLEAR USER-FLAG
0229 0460          420          JMP     POLL
                   421 ;
                   422 ;
022B 0415          423 TEST:    JMP     TEST1
                   424 ;
022D A9            425 TEST2:   MOV     R1,A                   ;ROM-TEST PAGE 2
022E A3            426          MOVP    A,@A
022F 68            427          ADD     A,R0
0230 A8            428          MOV     R0,A
0231 F9            429          MOV     A,R1
0232 17            430          INC     A
0233 EA2D          431          DJNZ    R2,TEST2
                   432 ;
0235 A9            433 T6:      MOV     R1,A                   ;ROM-TEST PAGE 3
0236 E3            434          MOVP3   A,@A
0237 68            435          ADD     A,R0
0238 A8            436          MOV     R0,A
0239 F9            437          MOV     A,R1
023A 17            438          INC     A
023B EA35          439          DJNZ    R2,T6
023D F8            440          MOV     A,R0
023E 9642          441          JNZ     ERR02                  ;JUMP IF ROM-ERROR
```

```
0240 0432          442          JMP     OK

                   443 ;
0242 23C0          444 ERR02:   MOV     A,#NUMER
0244 0434          445          JMP     OK+2
                   446 ;
                   451 ;
0246 D646          452 TONE:    JNIBF   TONE
0248 22            453          IN      A,DBB          ;GET TONE NUMBER
0249 03E0          454          ADD     A,#0E0H
024B AB            455          MOV     R3,A
024C 03CD          456          ADD     A,#0CDH
024E F678          457          JC      OUTSD
0250 FB            458          MOV     A,R3
0251 E7            459          RL      A
0252 0300          460 CALAD:   ADD     A,#LOW FRTAB   ;CALCULATE ADDRESS FOR TABLE
0254 A8            461          MOV     R0,A
0255 E3            462          MOVP3   A,@A
0256 AB            463          MOV     R3,A
0257 AC            464          MOV     R4,A
0258 18            465          INC     R0
0259 F8            466          MOV     A,R0
025A E3            467          MOVP3   A,@A
025B AD            468          MOV     R5,A
025C AE            469          MOV     R6,A
                   470 ;
                   471 ;
                   472 ;
025D D65D          473 TONE2:   JNIBF   TONE2
025F 22            474          IN      A,DBB          ;GET TONE LENGTH
0260 03E0          475          ADD     A,#0E0H
0262 17            476          INC     A
0263 AF            477          MOV     R7,A
0264 2300          478 TONE3:   MOV     A,#00
0266 62            479          MOV     T,A
0267 35            480          DIS     TCNTI
0268 55            481          STRT    T              ;START 20MS TIMER
0269 FD            482          MOV     A,R5
026A 9696          483          JNZ     TGENR
026C FB            484          MOV     A,R3
026D 9676          485          JNZ     TGEN
                   486 ;
026F 1673          487 PAUSE:   JTF     PAUSE1
0271 446F          488          JMP     PAUSE
0273 EF6F          489 PAUSE1:  DJNZ    R7,PAUSE
0275 65            490          STOP    TCNT
0276 0460          491          JMP     POLL
                   492 ;
                   493 ;
0278 27            494 OUTSO:   CLR     A
0279 4452          495          JMP     CALAD
                   496 ;
027B 9AFE          497 TGEN:    ANL     P2,#11111110B  ;SPEAKER ON
027D 1680          498          JTF     CHKLT
027F 00            499          NOP
0280 00            500          NOP
```

```
                        501 ;
0281 EB81               502 LOOP1:  DJNZ    R3,LOOP1        ;"ON"-TIME
0283 AB                 503         MOV     R3,A
0284 8A01               504         ORL     P2,#00000001B   ;SPEAKER OFF
0286 00                 505         NOP
0287 00                 506         NOP
                        507 ;
0288 EB88               508 LOOP2:  DJNZ    R3,LOOP2        ;"OFF"-TIME
028A AB                 509         MOV     R3,A            ;RELOAD TONE FREQUENCY
028B 4476               510         JMP     TGEN
```



```
LOC  OBJ            LINE        SOURCE STATEMENT

                     511 ;
                     512 ;
028D EF91            513 CHKLT:  DJNZ    R7,LOOP1        ;CHECK TONE LENGTH
028F 8A01            514         ORL     P2,#00000001B
0291 65              515         STOP    TCNT            ;READY
0292 B689            516         JF0     NEXT11          ;JUMP IF CHARACTER
0294 0460            517         JMP     POLL                --BEEP
                     518 ;
                     519 ;
0296 9AFE            520 TGENH:  ANL     P2,#11111110B   ;SPEAKER ON
0298 16B2            521         JTF     CHKL1
029A 00              522         NOP
029B 00              523         NOP
029C EB9C            524 LOOP3:  DJNZ    R3,LOOP3        ;"ON"-TIME
029E ED9C            525         DJNZ    R5,LOOP3
02A0 FC              526         MOV     A,R4
02A1 AB              527         MOV     R3,A
02A2 FE              528         MOV     A,R6
02A3 AD              529         MOV     R5,A
02A4 8A01            530         ORL     P2,#00000001B   ;SPEAKER OFF
02A6 00              531         NOP
02A7 00              532         NOP
                     533 ;
02A8 EBA8            534 LOOP4:  DJNZ    R3,LOOP4        ;"OFF"-TIME
02AA EDA8            535         DJNZ    R5,LOOP4
02AC FC              536         MOV     A,R4
02AD AB              537         MOV     R3,A
02AE FE              538         MOV     A,R6
02AF AD              539         MOV     R5,A
02B0 4496            540         JMP     TGENH
                     541 ;
                     542 ;
02B2 EF9C            543 CHKL1:  DJNZ    R7,LOOP3        ;CHECK TONE LENGTH
02B4 8A01            544         ORL     P2,#00000001B
02B6 65              545         STOP    TCNT
02B7 0460            546         JMP     POLL
                     547 ;
02B9 04A6            548 NEXT11: JMP     NEXT1
                     549 ;
```

```
                        550 ;
0300                    551             ORG     300H
0300 00                 552 FRTAB:      DB      00,00
0301 00
0302 88                 553             DB      088H,04         ;A              110HZ
0303 04
0304 55                 554             DB      055H,04
0305 04
0306 24                 555             DB      024H,04         ;H
0307 04
0308 F6                 556             DB      0F6H,03         ;  C
0309 03
030A CC                 557             DB      0CCH,03
030B 03
030C A4                 558             DB      0A4H,03         ;  D
030D 03
030E 7C                 559             DB      07CH,03
030F 03
0310 59                 560             DB      059H,03         ;  F
0311 03
0312 37                 561             DB      037H,03         ;  F
0313 03



0314 17                 562             DB      017H,03
0315 03
0316 F9                 563             DB      0F9H,02         ;  G
0317 02
0318 DB                 564             DB      0DBH,02
0319 02
031A C1                 565             DB      0C1H,02         ;  A            220HZ
031B 02
031C A8                 566             DB      0A8H,02
031D 02
031E 90                 567             DB      090H,02         ;  H
031F 02
0320 79                 568             DB      079H,02         ;     C1
0321 02
0322 63                 569             DB      063H,02
0323 02
0324 50                 570             DB      050H,02         ;     D1
0325 02
0326 3C                 571             DB      03CH,02
0327 02
0328 29                 572             DB      029H,02         ;     E1
0329 02
032A 19                 573             DB      019H,02         ;     F1
032B 02
032C 09                 574             DB      009H,02
032D 02
032E FC                 575             DB      0FCH,00         ;     G1
032F 00
0330 ED                 576             DB      0EDH,00
0331 00
0332 E0                 577             DB      0E0H,00         ;     A1        440HZ
0333 00
```

```
0334 04          576          DB      004H,00
0335 00
0336 C7          579          DB      0C7H,00          ;     H1
0337 00
0338 BC          580          DB      0BCH,00          ;       C2
0339 00
033A B1          581          DB      0B1H,00
033B 00
033C A7          582          DB      0A7H,00          ;     D2
033D 00
033E 90          583          DB      090H,00
033F 00
0340 94          584          DB      094H,00          ;     E2
0341 00
0342 8C          585          DB      08CH,00          ;     F2
0343 00
0344 84          586          DB      084H,00
0345 00
0346 7C          587          DB      07CH,00          ;     G2
0347 00
0348 75          588          DB      075H,00
0349 00
034A 6E          589          DB      06EH,00          ;     A2        880HZ
034B 00
034C 68          590          DB      068H,00
034D 00
034E 62          591          DB      062H,00          ;     H2
034F 00
0350 5C          592          DB      05CH,00          ;       C3

0351 00
0352 57          593          DB      057H,00
0353 00
0354 52          594          DB      052H,00          ;       D3
0355 00

                 595 ;
                 596 ;
                 597          END
```

USER SYMBOLS

BELL	021A	BELL1	0056	BIT0	0086	BIT1	0083
CALAD	0252	COMM	00AE	COMMAN	0200	CONV	0039
CONV1	0040	CTAB	0204	ERRO2	0242	ERTAB	0300
OUT	0055	INIT	0009	LNDVAR	021C	LOOP1	0281
LOOP2	0288	LOOP3	029C	LOOP4	02A8	MEND	003F
NEXT1	00A6	NEXT11	0289	NODAT	0080	NODAT1	00CC
OUTSD	0278	PAUSE	026F	PAUSE1	0273	POLL	0060
RAMER	0080	ROTST	00DC	SENFI	00BE	START	0000
START1	0011	T1	0019	T4	0100	T6	0235
TEST	022B	TEST1	0015	TEST2	022D	TI2	0073
TI3	0078	TI5	007C	TIME1	006E	TONE	0246
VAR1	0225	W1	0035	WAIT	00E9		

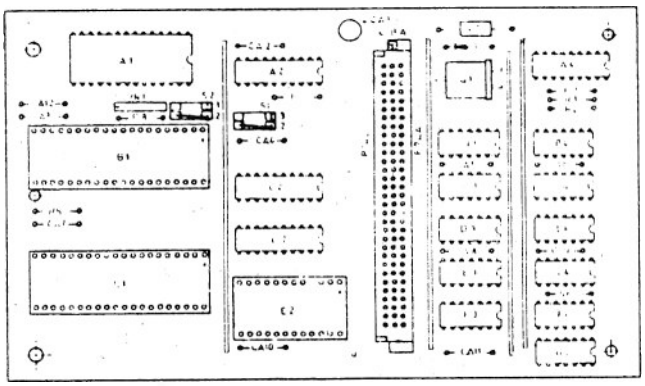
CHKL1	0282	RESET	0008	CHKLT	0280	OK	0032
OPOLL	020C	T2	0026	ERRO1	002F	ROMER	00C0
ISRT1	00EE	TGEN	027B	ISRTIM	00EC	T3	000F
LVAR	004D	TONE2	0250	LVAR1	0054	TGENH	029A
NODAT2	0087					TONE3	0264

F235

16-BIT PROCESSOR WITH PIC
(K235)

(F235)

Interrupt 16-Bit System



S1 - 1 to S1 - 2	IRQ5
S1 - 1 to S1 - 3	IRQ6
S2 - 1 to S2 - 2	IRQ2A
S2 - 1 to S2 - 3	IRQ2

Switches, 16-bit processor with interrupt controller

Install wire jumper in location R5 only when IC (8087) is not mounted in location B1.

Description of 8088 Interrupt System

1. Precondition:
   - Multilayer Controllerboard (stamp on the lower side of slot one ore above keyboard plug
   - F/K230 V003 (603-6091361) with integrated 8259A interruptcontroller

2. Supported Kits:
   - K210 Centronics I/F
   - K211 RS232 printer I/F
   - K212 RS232 communication I/F
   - K801 RS232 switchable I/F
   - K803 Real time clock
   - K806 Mouse I/F
   - K804 IEEE 488 I/F (only HW prepared)

3. Interrupts:
   - 8253 timer 2, 16 bit counter, 2uS to 130mS
   - 8741 keyboard interrupt
   - 8272 (uPD765) flex disk interrupt
   - Peripheral interrupt(bus pin c 31 on slot 2 - 6) for K803, K804 and K806
   - IRQ 3/4 for K211 or K801 on slot 3 and 4
   - IRQ 5 for K210,K212 or K801 on slot 5
   - IRQ 2A for integrated Winchester Disk

4. Description
   - 8259A Interruptcontroller, portadress 90 H
   - Interruptvectors (IBM like)

Address	Interrupt No.	NCR-DM-V	IBM
20-23H	8	8253 timer 2	timer
24-27H	9	8741 keyboard	keyboard
28-2BH	A	bus pin c31 slot 2-6	bus
2C-2FH	B	IRQ 3 RS 232 slot 3	RS232 sec
30-33H	C	IRQ 4 RS 232 slot 4	RS232 prim
34-37H	D	IRQ 2A (intern. Fix Disk)	Fix Disk
38-3BH	E	8272 Flex Disk Contr.	Flex Disk
3C-3FH	F	IRQ 5 Printer	Printer

The interrupts must be enabled individually by an application program. The interrupt priorities must be defined by application program. If K 803, K 804, K 806 are used together, after interrupt, the software has to check the interrupt status,to see which interrupt was set. The interrupts are maskable IBM like. If any interrupt is enabled the peripheral must be inserted in the DM V, to avoid failures (interrupt will be set by pull up).

5. Software

The operating systems will not support interrupts. If an application requires interrupts , the individual interrupt inputs must be enabled. After each application the interrupts must be disabled, or system failures will be occur.

6. 8259A Interruptcontroller

Description : see Intel data book.

7. Diagnostic

In case of interrupt failure the level 0 diagnostic LED 4 lights.

8. Peripheral Interfaces

K210, K211, K212, K801 and K804 can be used without modifications.
K804 must be internally switched to support interrupts.

9. Connections

int-No.	name	from	pin	over pin	to	pin
8	TIMINT	timer 8253	17(tout2)	J107-b8	8259	18(irq0)
9	KEYINT	keyb. 8741	35(P24)	J107-b7	8259	19(irq1)
A	INT/	busint	c31(J2-6)	J107-c31	8259	20(irq2)*
B	IRQ3	bus-Plug 3	a2 (J3)	J107-b3	8259	21(irq3)
C	IRQ4	bus-Plug 4	a2 (J4)	J107-b4	8259	22(irq4)
D	FIXDISK	bus-plug 2a	a2 (J2A)	J107-b31	8259	23(irq5)
E	FLEXINT	7272 (765)	18 (int)	J107-b6	8259	24(irq6)
F	IRQ5	bus-Plug 5	a2 (J5)	J107-b5	8259	25(irq7)

*inverted

all interrupts active high , businterrupt INT/ active low

10. Programming

After the level zero diagnostic following initialization of the 8255 must performed:
interrupt table entry 20 hex
interrupts are level triggered
all interrups inputs are disabled
the 8259 needs an end of interrupt command

use of the interruptcontroller:
mask-unmask interrupt inputs:

Adress 91H data 0= unmask interrups
1= mask interrup

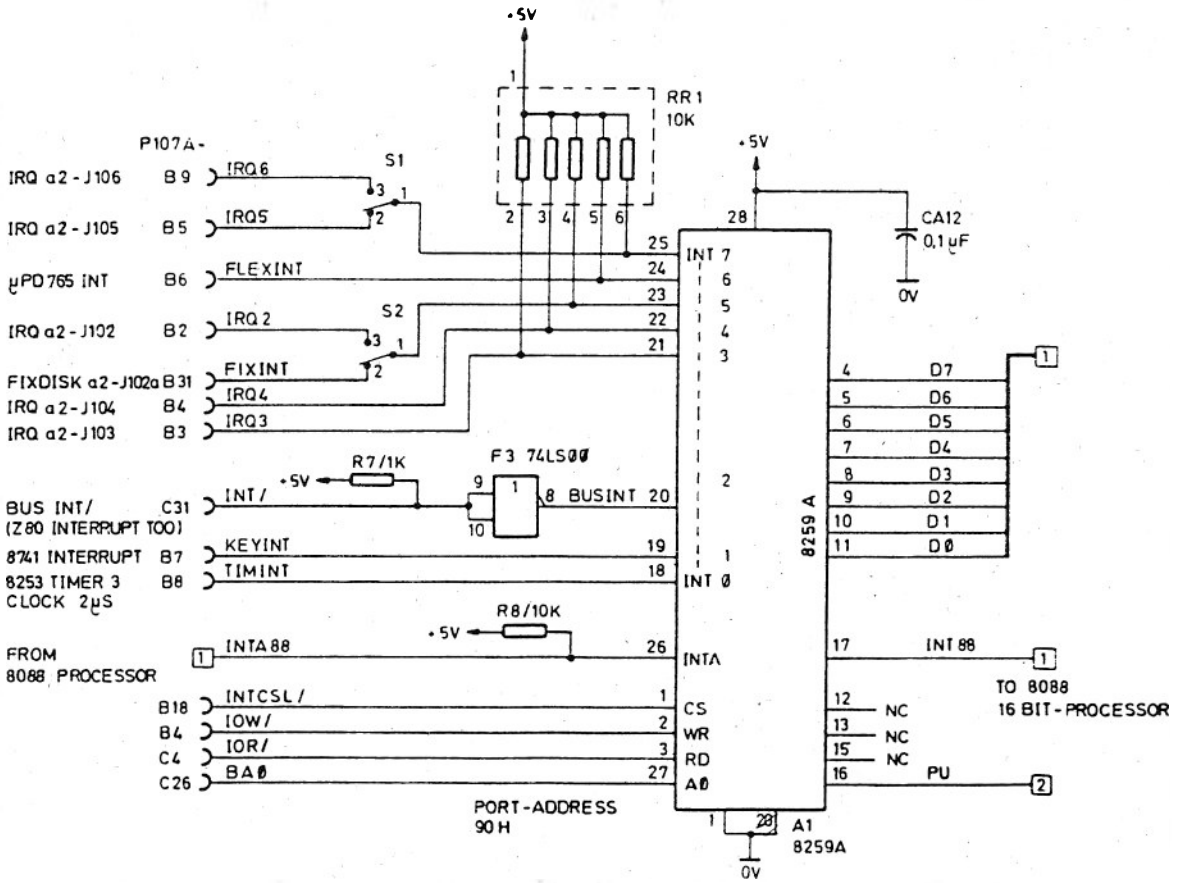
	D7	D6	D5	D4	D3	D2	D1	D0
91H	IR7*	IR6*	IR5*	IR4*	IR3*	IR2*	IR1*	IR0*

End of interrupt command (EOI)

Address 90H Data 20H

This command 20H is a NON Specific - EOI Command
that means the current interrupt will reset.

11. Schematic

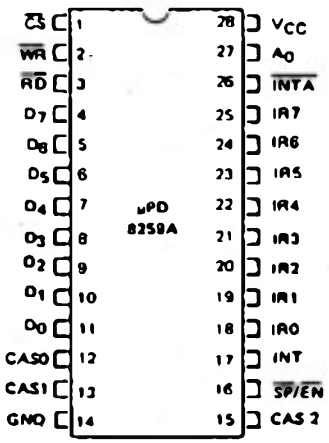


	a	b	c
1	+5V	+5V	+5V
2	OPT2	IRQ2	+12V
3	RESET/	IRQ3	RESETIN/
4	IOW/	IRQ4	IOR/
5	MEMW/	IRQ5	MEMR/
6	D1	FLEXINT	D0
7	D3	KEYINT	D2
8	D5	TIMINT	D4
9	D7	IRQ6	D6
10	READYDMA	READYP	ABTRI/
11	EOP/	HOLD	
12	INTACK/	SWITCH16/	IFSEL4/
13	DBTRI/	HOLDA16	DIR
14	THOLD/	16 BITAV/	HLDA
15	PCLK/	STDMARQ/	CLK1
16	LGRD	LGRD	TRAMD/
17	BA19	16 BIT SET /	BA18
18	BA17	INTCSL /	BA16
19	A15		A14
20	A13	MEMRQ/	A12
21	A11		A10
22	A9		A8
23	A7		A6
24	A5		A4
25	A3		A2
26	A1		A0
27	IFSEL3/		IFSEL2/
28	IFSEL1/		IFSEL0/
29	DRQ1		DRQ0/
30	DACK1/		DACK0/
31	WAIT/	IRQ2A	INT/
32	LGRD	LGRD	LGRD

Pin assignments P7AA to P7AC
(16-bit processor 8088, interrupt controller 8259A)

PROGRAMMABLE INTERRUPT CONTROLLER

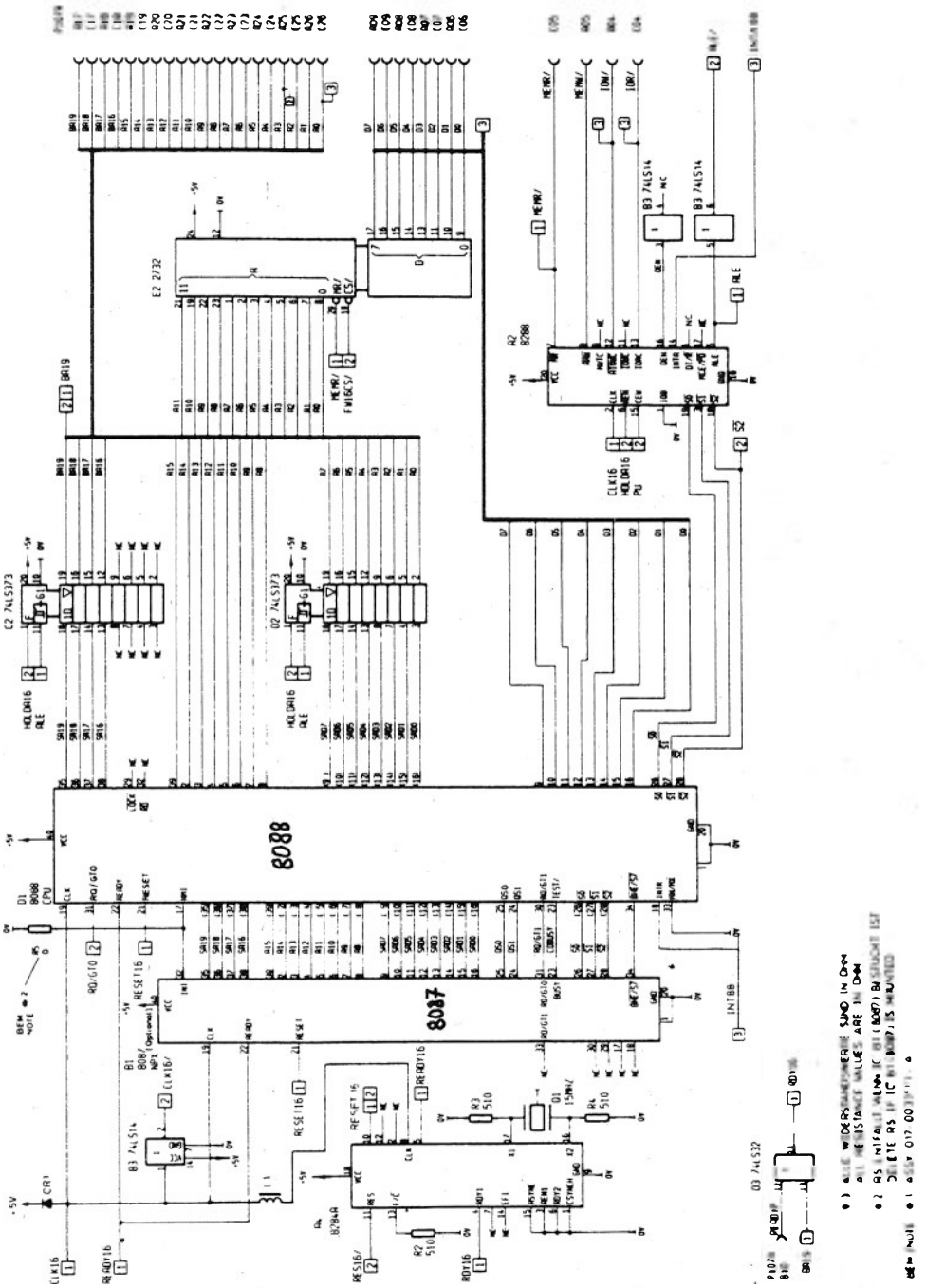
PIN CONFIGURATION



PIN NAMES

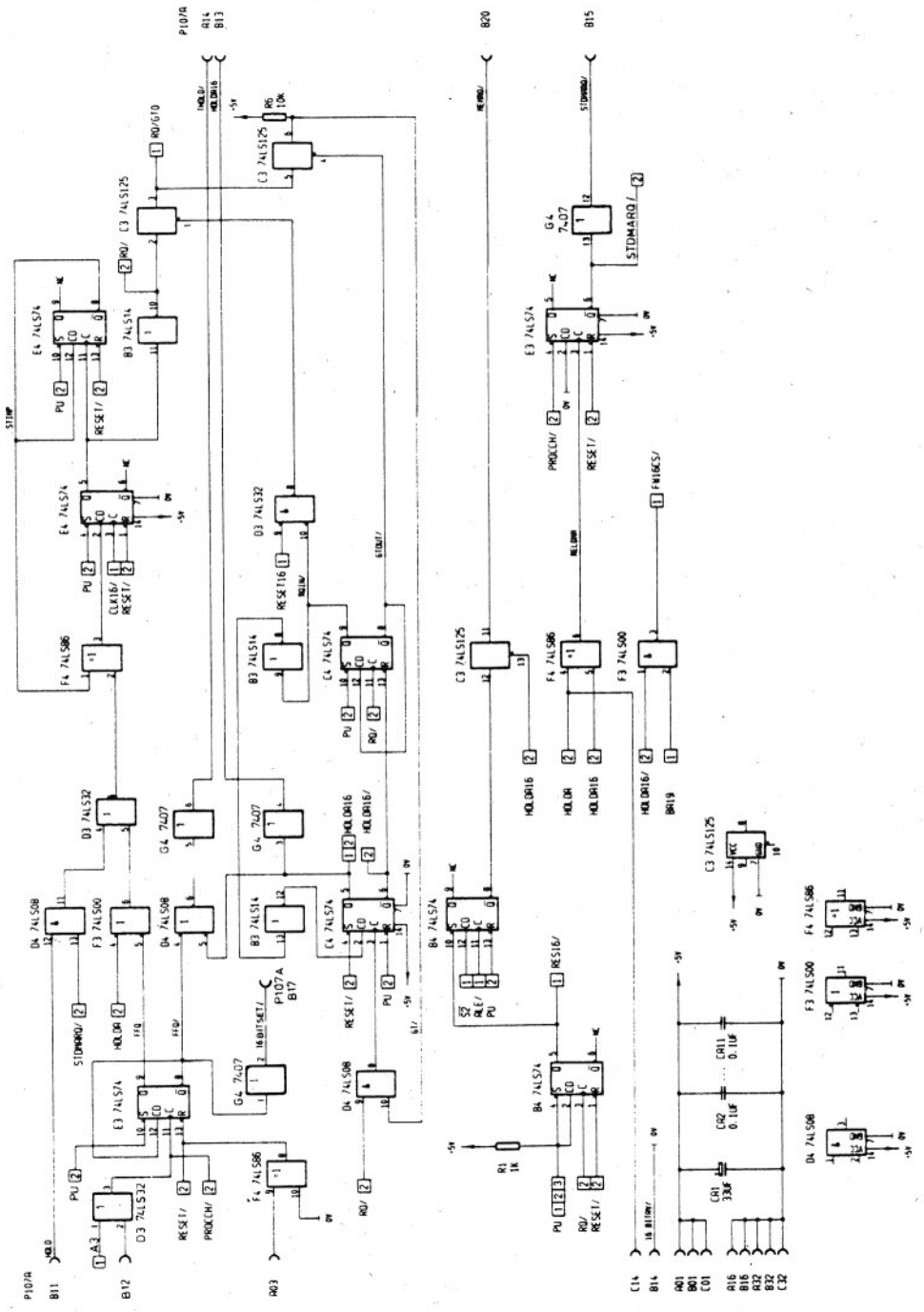
$D_7 - D_0$	Data Bus (Bi-Directional)
$\overline{RD}$	Read Input
$\overline{WR}$	Write Input
A_0	Command Select Address
CAS2 – CAS0	Cascade Lines
$\overline{SP/EN}$	Slave Program Input / Enable Buffer
INT	Interrupt Output
$\overline{INTA}$	Interrupt Acknowledge Input
IR0 – IR7	Interrupt Request Inputs
$\overline{CS}$	Chip Select

SCHEMATICS

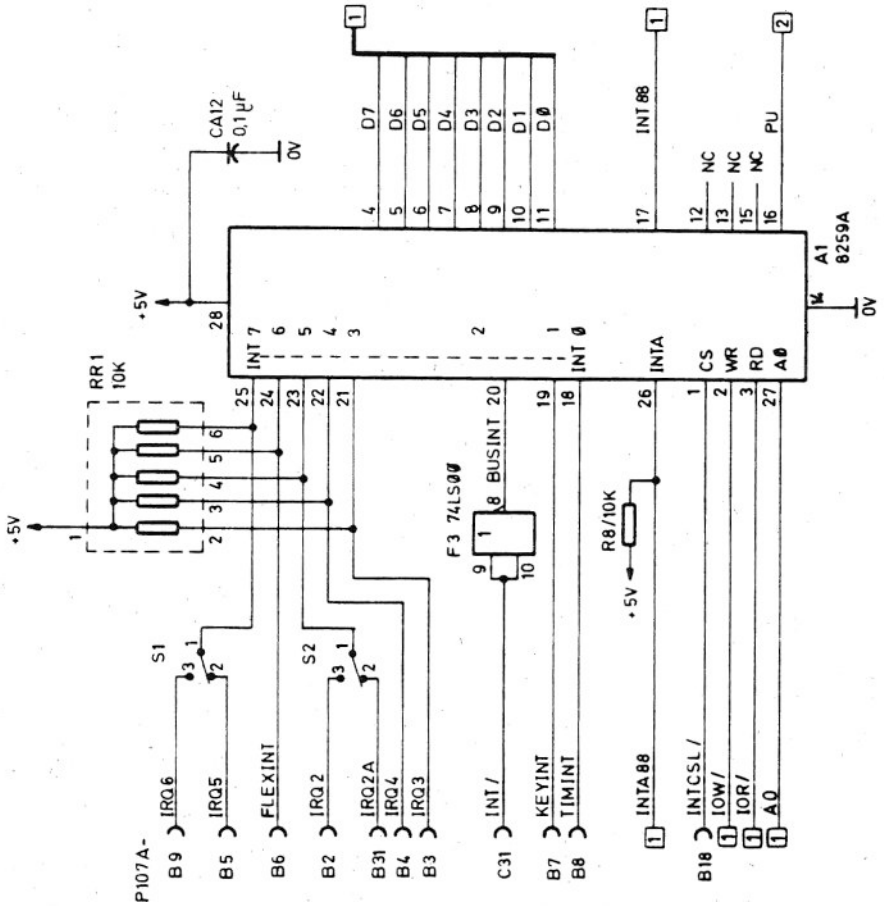


16-Bit Processor with PIC (K235) 017-0033502 Rev. A

1 of 3



2 of 3

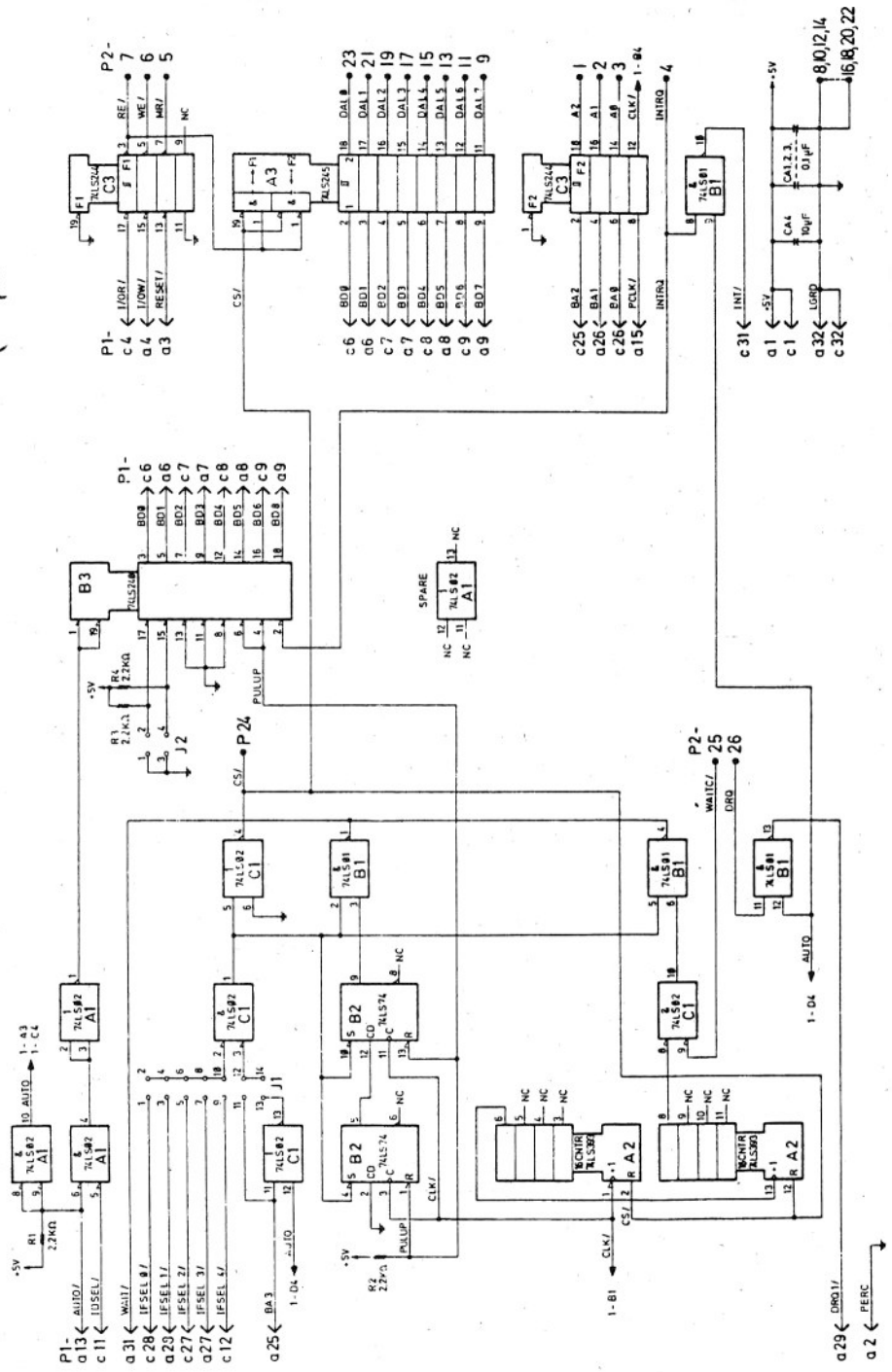


3 of 3

2 74LS02
1 74LS01
1 74LS74
1 74LS393
1 74LS240
1 74LS244
1 74LS245
4 22kΩ

5 16 pol Sockel
3 20 pol Sockel

C3282 Fix Disk



NCR 3282 Fixed Disk Interface 017-0027022 Rev. D

SOFTWARE

SOFTWARE

SW2-1. COLORGRAPHIC WITH CP/M 80
SW2-2. GRAPHICS CONTROLLER BOARD
SW2-3. HOW TO GENERATE A DBASE FILE THAT CAN BE USED WITH DR-GRAPH.

SW3-1. HOW TO CHANGE CPI ON THE NCR6455 UNDER MS-DOS
SW3-2. HOW TO CHANGE CPI ON THE TCR6411 UNDER MS-DOS
SW3-3. HOW TO CHECK PRINTER ERRORS WITH CENTRONICS INTERFACE FOR THE FX80/100 OR NCR 6411.

SW5-1. DM 5 CONTROL CODES FOR INSTALLING MULTIPLAN

SW6-1. HOW TO FIND OUT WHETHER A SYSTEM HAS A MULTI-LAYER MAIN CONTROL BOARD.
SW6-2. THE PROGRAMMABLE INTERVAL TIMER

COMMUNICATION

...

C O L O R G R A P H I C with CP/M-80

NCRGRAF color patch with MBASIC - POKE

With the MBASIC statement POKE it is possible to patch NCRGRAF 1.1 (for interpreted BASIC, not for compiled BASIC) to programm color graphics.

```
POKE  &HB88D,0      green plane

POKE  &HB88D,64     red plane

POKE  &HB88D,128    blue plane
```

The next GPOINT all sets the cursor in the color plane selected by the poke statement, and every color command will be drawn in this plane until the next POKE &HB88D-GPOINT-sequence.

A GINIT or a GCLEAR statement will not reset the color.

In a sequence of GLINE commands the end of the line is the new cursor and must not be set with GPOINT. To change the color during such a sequence a GPOINT statement must be inserted after the poke statement.

Graphic figures which should be drawn in mixed colors (cyan, magenta, yellow and white) must be drawn two (or 3) times at the same position with the same parameters in the corresponding color planes:

```
cyan      = blue and green
magenta   = blue and red
yellow    = red and green
white     = red and blue and green
```

The following example shows the drawing of a cyan circle.

```
EXAMPLE:    30....
            40 POKE &HB88D,0          'SELECT GREEN PLANE
            50 X=100  : Y=200
            60 CALL GPOINT(X,Y)
            70 R=50
            80 CALL GCIRCL(R)         'DRAW A GREEN CIRCLE
            90 POKE &HB88D,128        'SELECT BLUE PLANE
           100 CALL GPOINT(X,Y)
           110 CALL GCIRCL(R)         'DRAW A BLUE CIRCLE
```

On monochrome DM V red or blue figures will be drawn in green. If a figure is drawn in complement mode (GMODE = 1) and two color planes (cyan, magenta, yellow) then this figure will be invisible on a monochrome CRT, because the drawing of the second plane will delete the first one.

GRAPHICS CONTROLLER BOARD

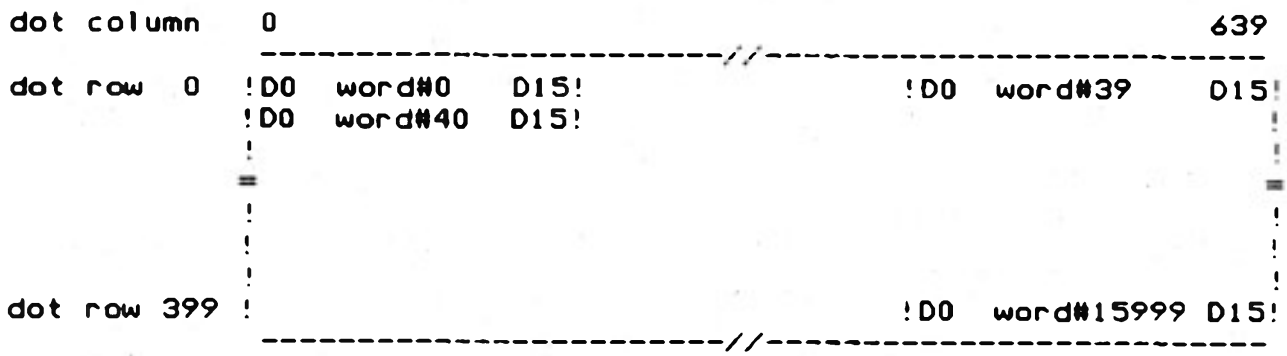
-GENERAL

The Graphic Controller Board is built up with the NEC 7220 GDC (Graphic Display Controller) which operates in Mixed Mode. The GDC controls a 16K*16bit video memory for a monochrome display, in color there are three memory planes of this size.

-MIXED MODE

*In Character Area the frame is 25 character lines by 80 characters. Upper and lower case characters are displayed in a 6*9 dot pattern in a 8*16 dot matrix which allows for descended lower case characters (see Appendix A for ASCII character table and character set). Characters and background can be defined independently in 8 colors: black, white, red, green, blue, yellow, magenta, and cyan.

*In Graphics Area the dot raster is equidistant in X- and Y-direction with 640H*400V pixels resolution. Pixel distance is about 0.3375mm (=0.0132 inch). In color each dot position can be drawn in any of 8 colors. The bit mapping of the video memory is shown in the following figure:

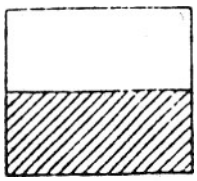


*Switching between both areas can be achieved by modifying bit 7 in parameter RAM byte P3 (and P7).

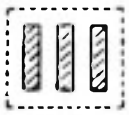
-MEMORY MODIFICATION

Each modification of the video memory lasts 0.8 usec, first the memory is read into the GDC then modified and written back. The modes of modification are:

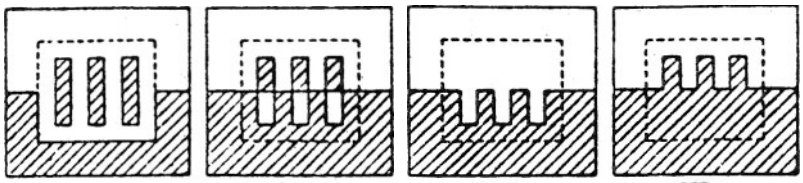
* Zero
* Set
* Replace
* Complement



ACTUAL MEMORY



MODIFICATION



-CHARACTER GENERATOR AND ATTRIBUTE LOGIC

The DMV contains two character generators with each up to 4kByte ROM.

*One is placed on the Graphic Controller Board and is automatically accessed in character area by the lower data byte of the memory (ASCII-code).
The higher data byte is stored into the attribute register which is valid for one character and available only in character area.

*Attributes:		monochrone	color	
Video Memory Bit	8	inverse	half intensity	
	9	blink enable	blink enable	
	10	half intensity	front color red	
	11	---	front color green	
	12	---	front color blue	
	13	---	back color red	b)
	14	---	back color green	b)
	15	---	back color blue	b)

a) standard value in color is E8 hex (back black, front green).
b) back color attribute bits are negative logic levels.

Attributes can also be directly set with escape sequences in the command level of the operating system.
(see table: Terminal Function Codes)

*Another character generator placed on the main board is accessible in the CPU's ROM address area from 1000hex to 1FFFhex. This can be used for character representation in graphic area. Instead of ASCII codes, 16 bytes of raster line information from this generator have to be sent to the video

memory to form one character correctly.

I/O-Address: ROMSEL = 11hex
RAMSEL = 10hex

Note: No attributes are possible in graphic area.

-VIDEO TIMING

*The Video Timing is fully programmable by the GDC. During initialisation of the DMV all timing values are defined for correct operation of the monitor.

CAUTION!

Do not alter any of the video timing parameters!

Redefining any of the parameters succeeding the RESET, SYNC, and VSYNC command may cause damage on the CRT monitor board. Also the surrounding hardware logic will not accept any modification of the video timing.

-GRAPHICS DRAWING

The GDC supports drawing of

* horizontal, vertical lines
* vectors
* rectangles
* arcs, circles
* graphic characters
* line, area pattern
* drawing in 8 directions

For detailed programming information refer to NEC GDC 7220 Product Description.

-INTERFACE TO THE CONTROLLER BOARD

* 8 bit data bus
* address bits BA0, BA1, BA2
* fifo buffer
* read, write control
* dma request GDRQ2 and acknowledge DACK2/

I/O hex address:

A0	read status register
A1	read fifo
A0	write parameter into fifo
A1	write command into fifo
A2	write display zoom factor

Status Register:

Bit 0	data ready
1	fifo full
2	fifo empty
3	drawing in progress
4	dma execute
5	vertical sync active
6	horizontal blank active
7	light pen detect

The graphic controller board is directly connected onto the controller board with two connectors, J/P 113, and J/P 114. Pin assignment:

P114	a	c	P113	a	c
1	+5V	+5V	1	+5V	+5V
2	LGRD	LGRD	2		
3	LPEN	LPENSW	3	D1	D0
4	+12V	+12V	4	D3	D2
5	+12V		5	D5	D4
6			6	D7	D6
7			7		
8			8	HSYNCX	HSYN
9			9	VSYNCX	LGRD
10			10	BA1	BA0
11			11		
12			12	GDCIOW/	GDCIOR/
13			13	DACK2/	GDRQ2
14			14		
15			15		WCLK
16	LGRD	LGRD	16	LGRD	LGRD

-GDC FEATURES

*Split screen: Character and graphic area can be mixed within one image (display partition area 1 and 2 programmable with the PRAM command).

*Scrolling

*Paging: In character area the capacity of the video memory allows storing of 8 video pages.

*Panning: The top left corner of any image cutout is shifted to the top left corner of the screen. Normally used together with zooming.

*Zooming: a) for graphics character writing (factors 1 - 16)
b) for display 1 to 16 (only for DMV with change level > 35)

-VIDEO MEMORY ADDRESS RANGE

Hex address:	monochrome	color
0000-3FFF	green	green
4000-7FFF	-	red
8000-BFFF	-	blue

-GDC COMMAND SUMMARY

* Video Control Commands
  - RESET — resets GDC, specifies video display format
  - SYNC — specifies the video display format
  - VSYNC — selects master or slave mode
  - CCHAR — defines cursor and character row height

* Display Control Commands
  - START — ends idle state and activates the display
  - BCTRL — controls blanking of the screen
  - ZOOM — specifies display or character writing zoomfactor
  - CURS — sets cursor position in display memory
  - PRAM — start address and length of display areas, specifies 8 bytes for graphics character or 2 bytes for drawing pattern
  - PITCH — specifies horizontal width of display

* Drawing Control Commands
  - WDAT — writes data words or bytes into display memory
  - MASK — sets the mask register contents
  - FIGS — specifies parameters for drawing controller
  - FIGD — draws the above specified figure
  - GCHRD — draws graphics characters

* Data Read Commands
  - RDAT — reads data words or bytes from display memory
  - CURD — reads the cursor position
  - LPRD — reads the light pen address

How to generate a dBase file that can be used with DR-GRAPH:

1. Create a dBase file named SDF.DBF

2. Enter in dBase II:

```
Use B: SDF.DBF.
Copy to B: SDF delimited with ",".
```

   This generates the file SDF.TXT on drive B.

3. Invoke "SDI" that is delivered with SuperCalc.

   - Select option B.
   - Enter B:SDF.TXT as source file name.
   - Enter B: SCSD as destination file name.

   This creates the SuperCalc file SCSD.CAL on drive B.

   - Select option C of the SDI menu.
   - Enter B:SCSD.CAL as source file name.
   - Enter V upon "output ALL or VALUES (A or V)".
   - Enter B:SCSD.SDF as destination file name.

4. The file SCSD.SDF can be processed by DR-GRAPH.

How to Change CPI on the NCR 6455 under MS-DOS:

The NCR 6455 can be switched to a different character width (10/12/15 CPI) by escape sequences. In CP/M this can be done via keyboard. In MS-DOS, the ESC-key is not scanned, therefore a file has to be generated for each width. Here an example for 12 CPI (Enter only the underlined characters!):

```
DEBUG (CR)
-E100
0A86:0100 00.1B Sp 00. 5D Sp 00. 4A Sp 00.1A (CR)
-NCP12 (CR)
-RCX (CR)
CX0000
-4 (CR) (length in bytes)
-W
Writing 0004 bytes
-Q

Sp = Space

A> CONTROL P (turn print on)
A>TYPE CP12 (12 characters per inch)
A>DIR (print directory)
```

Other printer settings can be achieved in the same way.

The NCR 6455 can be controlled as follows:

```
ESC J H = 1B 5D 48 = Compressed Pitch (15 CPI)
ESC J J = 1B 5D 4A = Elite (12 CPI)
ESC J L = 1B 5D 4C = Pica (10 CPI)
```

How to Change CPI on the NCR 6411 under MS-DOS:

The NCR 6411 printer can be switched to a different character width (10/12/17 CPI or proportional) by escape sequences. In CP/M this can be done via keyboard. In MS-DOS, the ESC-key is not scanned, therefore a file has to be generated for each width. Here an example for 17 characters per inch. (Enter only the underlined characters!):

```
DEBUG (CR)
-E100
0A86:0100  00.1B Space 00. 51 Space 00. 1A (CR)
-NCP17 (CR)
-RCX (CR)
CX0000
-3 (CR) (length in bytes)
-W
Writing 0003 bytes
-Q
A> CONTROL P (turns on print)
A>TYPE CP17 (17 characters per inch)
A>DIR (print directory)
```

Other printer settings can be achieved in the same way.

The NCR 6411 can be controlled as follows:

```
ESC Q = 1B 51 = Compressed Pitch (17 CPI)
ESC E = 1B 45 = Elite (12 CPI)
ESC N = 1B 4E = Pica (10 CPI)
ESC P = 1B 50 = Proportional
```

How to check the Centronics-Interface for Errors when connected with Epson FX-80/100 or NCR 6411

— K 210 —

When you are printing with the Centronics-Interface (CP/M Driverprogram) and a error (Paper is empty, Printer is off ...) occurs the Printroutine will hang because the Driverprogram only supports Ready-Busy Handshaking. This is done by checking one bit of Port 61H.
But via this 8-bit wide Port the Printer tells the Centronics-Interface also other information about his status:

Port 61H

Bit #:	7	6	5	4	3	2	1	0
Signal:	ERR/	PE	BUSY	SLCT	TEST1	ACK/	OBF	TEST4/

The signals are activated by:

ERR/	Paper empty, Deselected or other Error
PE	Paper empty
BUSY	Data is received or printed, Deselected, ERR/, Buffer full(NCR 6411), Printerinitialisation
SLCT	on Epson always activated; NCR 6411: when ERR/ is not active or SEL-Key is pressed
TEST1	always active (not connected TTL-Input)
ACK/	becomes active for 12us (NCR 6411: 7us) when Printer is ready
OBF	normaly deactivated
TEST4/	always active (Input of TTL-Inverter not connected)

Following States can be detected by combining three signals: (1=active)

	ERR/	PE	BUSY
Ready	0	0	0
Printer is switched off	0	1	1
Paper empty or Initstatus	1	1	1
or	0	1	0
Busy	0	0	1
Deselected or other Error	1	0	1
- must not come -	1	0	0
or	1	1	0

The following two sample programs show the use of the states (MS-BASIC and 8080-Assembler).

```
10 ' Statecontrol on Printing with Centronics-Interface
20 ' for Epson FX-80/100 and NCR 6411 / C. ITOH 8510
40 '
50 DIM M$(5) 'Messages
60 E$=CHR$(27): CI=&H61: CO=&H60
70 P$=E$+"="+CHR$(3+32)+CHR$(0+32)
80 T$="I'm a sentence for testing."+CHR$(13)+CHR$(10)
90 '
100 FOR J=0 TO 5: READ M$(J): M$(J)=M$(J)+CHR$(23): NEXT
110 DATA "Ok - Selected"
120 DATA "Switched off","Paper empty or Init"
130 DATA "Deselected or other Error","Busy"
140 DATA "Damage Status"
150 '
160 PRINT CHR$(26);"I try to print the sentence: ";T$;
170 PRINT "Press any key to stop printing."
180 FOR J=1 TO LEN(T$)
190 S$=MID$(T$,J,1): GOSUB 230 'Print Substring
200 NEXT: GOTO 180
210 '
220 ' Printroutine
230 IF LEN(INKEY$)<>0 THEN END
240 PRINT P$;
250 W=INP(CI) AND 224 'only ERR/,PE,BUSY
260 IF W=128 THEN  PRINT M$(0): GOTO 320
270 IF W=224 THEN PRINT M$(1): GOTO 230
280 IF W=96 OR W=192 THEN PRINT M$(2): GOTO 230
290 IF W=32 THEN PRINT M$(3): GOTO 230
300 IF W=160 THEN PRINT M$(4): GOTO 230
310 PRINT M$(5): GOTO 230
320 OUT CO,ASC(S$)
330 RETURN
```

```
        ; Statecontrol on Printing with Centronics-Interface
        ; for Epson FX-80/100 and NCR 6411 / C. ITOH 8510
        ; 8080-Mnemonics

0005 =          BDOS    EQU 0005H
0061 =          CENIN   EQU 61H         ;IN-Port (Centronics)
0060 =          CENOUT  EQU 60H         ;DATA-OUT-Port (")
0006 =          DIRCON  EQU 6
0009 =          PRTSTR  EQU 9           ;Print String
000A =          LF      EQU 10          ;Line Feed
000D =          CR      EQU 13          ;<CR>
0017 =          EOL     EQU 23          ;Clear End of Line
001A =          HOME    EQU 26
001B =          ESC     EQU 27

0100                    ORG 100H
0100 CD1401             CALL INIT
0103 110702     GETCHAR LXI D,PRTEXT
0106 1A         NXT     LDAX D          ;fetch character
0107 FE24               CPI '$'
0109 CA0301             JZ GETCHAR      ;all char. printed ?
010C CD2C01             CALL PRINT
010F D0                 RNC             ;key pressed ?
0110 13                 INX D           ;next character
0111 C30601             JMP NXT

0114 0E09       INIT    MVI C,PRTSTR    ;Print HTXT0
0116 112502             LXI D,HTXT0
0119 CD0500             CALL BDOS
011C 0E09               MVI C,PRTSTR    ;Print PRTEXT
011E 110702             LXI D,PRTEXT
0121 CD0500             CALL BDOS
0124 0E09               MVI C,PRTSTR    ;Print HTXT1
0126 114402             LXI D,HTXT1
0129 C30500             JMP BDOS

012C D5         PRINT   PUSH D          ;save Registers
012D 47                 MOV B,A         ; "
012E C5                 PUSH B          ; "
012F CD9701     PRINT1  CALL GETKEY
0132 CA9201             JZ PREND
0135 0E09               MVI C,PRTSTR    ;pos. Cursor
0137 110202             LXI D,POS
013A CD0500             CALL BDOS
013D DB61               IN CENIN        ;get status
013F E6E0               ANI 11100000B   ;only ERR/,PE,BUSY
0141 FE80               CPI 128         ;nothing active ?
0143 C25601             JNZ CP1
0146 11A101             LXI D,MESG0
0149 C1                 POP B
014A 78                 MOV A,B
014B D360               OUT CENOUT
```

```
014D 0E09               MVI C,PRTSTR
014F CD0500             CALL BDOS
0152 37                 STC
0153 C39501             JMP PREND2
0156 FEE0       CP1     CPI 224          ;PE,BUSY ?
0158 C26101             JNZ CP2
015B 11B001             LXI D,MESG1
015E C38A01             JMP PRMSG
0161 FE60       CP2     CPI 96           ;ERR/,PE,BUSY ?
0163 C26C01             JNZ CP3
0166 11BE01     CP20    LXI D,MESG2
0169 C38A01             JMP PRMSG        ;print message
016C FEC0       CP3     CPI 192          ;PE ?
016E CA6601             JZ CP20
0171 FE20               CPI 32           ;ERR/,BUSY ?
0173 C27C01             JNZ CP5
0176 11D301             LXI D,MESG3
0179 C38A01             JMP PRMSG
017C FEA0       CP5     CPI 160          ;BUSY ?
017E C28701             JNZ CP6
0181 11ED01             LXI D,MESG4
0184 C38A01             JMP PRMSG
0187 11F301     CP6     LXI D,MESG5
018A 0E09       PRMSG   MVI C,PRTSTR     ;print message
018C CD0500             CALL BDOS
018F C32F01             JMP PRINT1
0192 C1         PREND   POP B
0193 37                 STC
0194 3F                 CMC
0195 D1         PREND2  POP D
0196 C9                 RET

0197 0E06       GETKEY  MVI C,DIRCON     ;get key
0199 1EFF               MVI E,0FFH
019B CD0500             CALL BDOS
019E FE0D               CPI CR
01A0 C9                 RET

01A1 4F6B202D20MESG0    DB 'Ok - Selected',EOL,'$'
01B0 5377697463MESG1    DB 'Switched off',EOL,'$'
01BE 5061706572MESG2    DB 'Paper empty or Init',EOL,'$'
01D3 446573656CMESG3    DB 'Deselected or other Error$'
01ED 4275737917MESG4    DB 'Busy',EOL,'$'
01F3 44616D6167MESG5    DB 'Damage Status',EOL,'$'
0202 1B3D232024POS      DB ESC,'=',3+32,0+32,'$'
0207 49276D2061PRTEXT   DB 'I''m a sentence for testing.'
0222 0D0A24             DB CR,LF,'$'
0225 1A49207472HTXT0    DB HOME,'I try to print the sentence: $'
0244 5072657373HTXT1    DB 'Press <CR> to stop printing.$'
```

DM V CONTROL CODES FOR INSTALLING MULTIPLAN

Question	Answer
Sequentially?	Y
CANCEL	^C
HOME	&E R
END	^Z
RIGHT DIR KEY	^D
UP DIR KEY	^E
LEFT DIR KEY	^S
DOWN DIR KEY	^X
BACKSPACE	^H
DELETE	&X
HELP	?
RETURN	^M
TAB	^I
CHAR LEFT	^K
CHAR RIGHT	^L
WORD LEFT	^O
WORD RIGHT	^P
NEXT WINDOW	^W
PAGE UP	^R^E
PAGE DOWN	^R^X
PAGE LEFT	^R^S
PAGE RIGHT	^R^D
NEXT ULCELL	^F
REFERENCE	@
RECALC	!
UP SCROLL	^U
DOWN SCROLL	^J
LEFT SCROLL	&EH
RIGHT SCROLL	&EK
BEGIN GRAPHICS MODE	N/A
END GRAPHICS MODE	N/A
VERTICAL BAR	|
UPPER RIGHT CORNER	+
LOWER RIGHT CORNER	+
LOWER LEFT CORNER	+
UPPER LEFT CORNER	+
TOP HALF OF +	+
BOTTOM HALF OF +	+
HORIZONTAL BAR	-
CLEAR THE SCREEN	^Z
START CURSOR POSITIONING	&E=
UPPER LEFT CORNER ROW 1, COL 1 ?	N
ROW POSITION FIRST?	Y
ROW NUMBER MODIFIED	5
WHAT VALUE IS ADDED	32
CHARACTERS TO SEPARATE ROW, COL	N/A
HOW IS COLUMN NUMBER MODIFIED	5
WHAT VALUE IS ADDED	32
END CURSOR POSITIONING	N/A

Question	Answer
INITIALIZE TERMINAL	N/A
RESET TERMINAL	N/A
ERASE TO END OF LINE	&Et
CLEAR TO END OF DISPLAY	&Ey
NON-DESTRUCTIVE REVERSE VIDEO	Y
BEGIN REVERSE VIDEO	&EG4
END REVERSE VIDEO	&EG0
TURN OFF REVERSE VIDEO	N
TURN ON KEYBOARD CLICK	N/A
TURN OFF KEYBOARD CLICK	N/A
TURN ON CURSOR	N/A
TURN OFF CURSOR	N/A
SOUND THE BELL	G
NUMBER OF ROWS	24
NUMBER OF CHARACTER COLUMNS	80
NAME OF TERMINAL	NCR Decision Mate V

Note: N/A means that this question is not applicable to the DM V -- leave the answer blank.

^ means that is the CONTROL-KEY.

How to find out whether a system has a multi-layer main control board:

A multi-layer main board is required for:

- Reset via CONTROL-F20.
- Support of 512 K bytes of memory.

As multi-layer board implementation cannot be detected by the tracer or factor number, determination as to whether a system can be upgraded beyond 256 K bytes of memory may create some confusion.

All systems that have a factory installed multi-layer main board have a diamond stamped at the rear between the NCR logo and the FT2 number. With the diamond, you can read QA and a number.

THE PROGRAMMABLE INTERVAL TIMER

Interval timing in the NCR DECISION MATE V is provided by an 8253 Programmable Interval Timer. This integrated circuit can be used as three independent 16-bit counters. The Timer is interfaced to the data bus, so that Timer values can be transmitted and read by the microprocessor. In addition, the Timer can be used to generate interrupts at programmed intervals or a single interrupt after a specific interval. Counting is carried out internally by the Timer, either as a binary or a Binary Coded Decimal (BCD) operation. Counting speed is determined by an external clock signal. Counting is achieved by decrementation of a Counter from the value loaded down to zero.

Communication between Timer and microprocessor is via the Port addresses 80H-83H:

Instruction	Function
OUT 80H	Load Counter 0
OUT 81H	Load Counter 1
OUT 82H	Load Counter 2
OUT 83H	Specify Timer operation
IN 80H	Read Counter 0
IN 81H	Read Counter 1
IN 82H	Read Counter 2
IN 83H	No operation

All three counters have a clock input as follows:

	Multilayer main PCB	Non-multilayer main PCB
CLK 0:	56 Hz	56 Hz
CLK 1:	23.1 KHz	23.1 KHz
CLK 2:	500 KHz	56 Hz

CAUTION: Counters 0 and 1 of the Timer provide synchronization signals required by the video display. These two counters are initialized by the firmware of the NCR DECISION MATE V. Under no circumstances should you disturb the contents of these registers, otherwise damage to your computer may result. If you wish to read these

Counters (for example, in order to derive a random number), you should use a Counter Latch, as described below.

You may wish to use Counter 2 in conjunction with the Programmable Interrupt Controller (8235). An external take-off point for this Timer signal is pin b8 of peripheral slot 7.

Counters can be programmed independently of one another. Before a counter is initialized it is in an undefined state. The following programming steps are required, in order to set up a Timer Counter.

One byte must be transmitted to the Timer's Control Register via Port 83H. The value of this byte is made up as follows:

D7	D6	D5	D4	D3	D2	D1	D0	via Port
COUNTER		R/W SELECT		MODE		BCD		83H

COUNTER is a two-bit binary value 0-2, denoting the number of the Counter to be accessed. Therefore, to access Counter 2, D7 should be set and D6 zero.
R/W SELECT determines the way in which the specified Counter is to be loaded or read. The type of operation to be carried out (read or load) depends on whether an IN or OUT instruction is being used (the Timer has pin connections for /RD and /WR signals). The significance of the binary value contained in these two bits is as follows:

0 Counter Latching (see below)
1 Read/load more significant byte of Counter
2 Read/load less significant byte of Counter
3 Read/load both Counter bytes (less significant first)

BCD: if this bit is set, the 16 bits of the selected Counter are used as a 4-digit BCD counter. If this bit is zero, the Counter represents a 16-bit binary value.
MODE may be a binary value 0-5 in three bits. The following modes can be implemented in the NCR DECISION MATE V hardware:

0 - Following the loading of the Counter, the signal OUTput pin for that Counter goes low, and remains low until the Counter has decremented to zero. The OUT signal then goes high, and remains high until the Counter is next programmed. If you write a new value to the Counter before

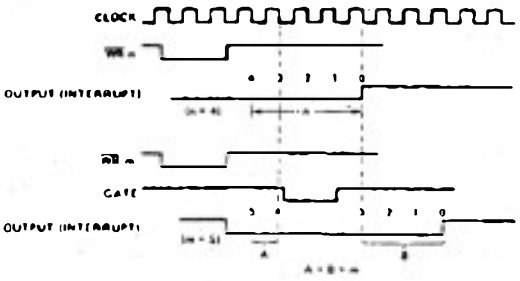
the old count has expired, decrementing resumes from the new value. From the hardware point of view, mode 0 is enabled by a high signal at the Gate for the specified counter. This signal is permanently present at the Gate for Counter 2.

2 - An OUTput low pulse of one clock is issued upon terminal count. As the Gate for Counter 2 is permanently high, this process repeats itself.

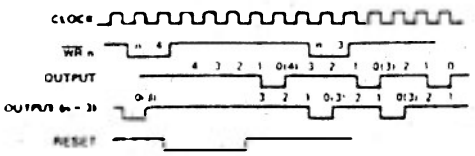
3 - As Mode 2, except that OUTput remains high for the first half of the count and goes low for the second half (achieved by decrementing by 2 at each clock). If the Counter specifies an odd number, the first decrement in the first countdown is by 1, the next countdown starts with a decrement of three.

4 - As soon as this mode has been loaded, the OUT pin for the selected Counter on the Timer goes high. When the Counter has been subsequently loaded, counting begins. As soon as the count has decremented to zero, the OUT pin goes low for one clock period, and then high again. If your software reloads the Counter during decrementing, the new Counter value takes effect at the next clock signal. As in mode 0, operation of mode 4 is dependent on the presence of a high signal at the Gate (decrementing would be suspended if this signal were low).

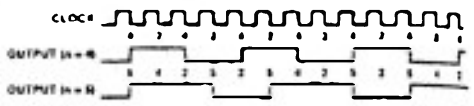
MODE 0: Interrupt on Terminal Count



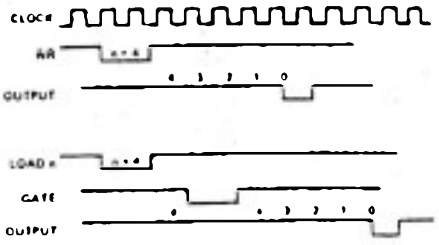
MODE 2: Rate Generator



MODE 3: Square Wave Generator



MODE 4: Software Triggered Strobe



The other modes (1 - programmable one-shot, 5 - hardware triggered strobe) are not available. This is because the Gate pin of Counter 3 is permanently pulled up high via a resistor (see schematics in Appendix A).

Following the Timer Control Register byte detailed above, the Timer expects the number of bytes specified in the two bits R/W to be transmitted by the microprocessor (load operation), or the specified number of bytes to be read. The one or two-byte value is then read from or written to the data bus. Loading all zeros into a Counter results in a maximum count (0FFFFH in binary, 9999 in BCD counting).

Note that it is not necessary to read or write immediately after setting the Timer Control Register. However, the specified number of bytes must be read or written. In mode 0, as soon as the Timer recognizes that the first (or only) byte is being transmitted, the decrementing process is suspended until the new Counter contents have been read.

Reading Counter registers requires some care in order not to disturb the counting process. A Counter can be read directly or via a Counter Latch. The former method requires counting to be inhibited during the reading process. This can be achieved only by controlling the Gate or suspending the clock signal to the Counter which is to be read. For this reason, you should use the Counter Latch method.

To read a Counter Latch, a byte must be written to the Control Register, specifying the Counter and with D5 and D4 zero (this command has no effect on the MODE and BCD settings). Then issue a read Counter byte to the Control Register and read the one or two bytes specified.

EXAMPLE

The following example makes use of the clock signal at the Timer to generate a "random" value in the range 0-0FFH. Routines are used for initializing and reading the timer, and for reading the keyboard (the keyboard reading routine in this example conforms to the Direct Console I/O function of the CP/M-86 operating system). Pressing a key after clock in stops the count in Counter 3. The "random" number is denoted by the LEDs at the back of the computer.

```
;
CONREG   EQU  83H      ;Port to Timer Control Register.
COUNT2   EQU  82H      ;Port to Timer Counter 2.
```

```
C2        EQU  80H      ;Mask for Control Register selecting COUNTER 2.
LATCH     EQU  0        ;Sets R/W SELECT in Control Register to "latch".
LOHI      EQU  30H      ;MODE: both low and high bytes.
;
LEDPORT   EQU  0        ;Port to LEDs.
;
          CSEG
;
;The following routine transmits one byte to Timer Control Register.
;
SETTIM1:  MOV  DX,CONREG
          MOV  AL,C2
          OR   AL,LOHI
          OUT  DX,AL
          RET
;
;This  routine  accepts  the current contents of AX as the  new  value  for
;Counter 2. The low byte is transmitted first.
;
SETTIM2:  MOV  DX,COUNT2
          OUT  DX,AL
          XCHG AH,AL
          OUT  DX,AL
          RET
;
;RDTIM  reads Counter 2 by means of Counter latching.  Both bytes are read.
;The number thus read can be regarded as reasonably random,  as the Counter
;is clocked by a 500 KHz signal.  This gives the Counter a maximum  exhaust
;time  of  approximately 131 ms (0FFFFH decrementing to zero).  The  Counter
;continues counting after the programmed count value is exhausted.  The low
;byte is returned in BL (read first), the high byte in BH.
;
RDTIM:    MOV  DX,CONREG
          MOV  AL,C2       ;OR AL,LATCH superfluous, as LATCH is 0.
          OUT  DX,AL       ;Counter 2 to be latched.
          OR   AL,LOHI
          OUT  DX,AL       ;both bytes of Counter 2 to be read.
          DEC  DX
          IN   AL,DX       ;read low byte.
          MOV  AH,AL
          IN   AL,DX       ;read high byte.
          XCHG AH,AL
          MOV  BX,AX
          RET
;
;Routine to read keyboard (CP/M-86. Adjust for other operating systems)
;
KBSTAT:   PUSH BX
          MOV  CL,6
```

```
        MOV  DL,0FEH
        INT  224
        POP  BX
        RET
;
The main program starts here.
;
SAMPLE: CALL SETTIM1
        XOR  AX,AX      ;set max. count for Counter 2. This is, strictly
                        ;speaking, not necessary, as Counter does not
                        ;terminate at zero.
        CALL SETTIM2
KBWAIT: CALL KBSTAT
        CMP  AL,0
        JE   KBWAIT     ;jump if no key pressed.
BREAK:  CALL RDTIM
        MOV  DX,LEDPORT
        MOV  AL,BL      ;use low byte only, as this yields more random
                        ;sample.
        OUT  DX,AL      ;transmit 8-bit value to diagnostic LEDs.
        NOP
;
        DSEG
        ORG  100H
;
DUMBYTE DB   0
;
        END
;
```

If you require accurate timing for real-time applications, you should make use of the Programmable Interrupt Controller in conjunction with the Timer. An example is included in the appropriate section of this Manual.

Third Party Support Center
NCR GmbH TSC
Ulmer Straße 160 · D-8900 Augsburg
Engineering & Manufacturing
Phone 821/4051
Telex 53749 · Telefax 8211405462
For Information only without Guaranty

NCR DECISION MATE V
Schnittstellen-Beschreibung
(RS 232 - C)

N C R DECISION MATE V, der Personal-Computer mit Herz

DM V SCHNITTSTELLEN - HANDBUCH

Inhaltsverzeichnis Seit

1.	Einführung und Beschreibung der RS232 Schnittstelle	3
1.2	Pinbelegung der V.24 Schnittstelle	4
1.3	Probleme mit der Norm	6
1.4	Übertragungsprotokolle	8
2.	Tips zum Anschluß eines Druckers	8
2.1	Drucker ohne XON/XOFF Protokoll	8
2.2	Checkliste für Druckerinstallation	9
3.	Hardware der V.24 Schnittstelle	10
3.1	Schaltbild des V.24 Interface	11
3.2	V.24 Interface wird interruptfähig	12
4.	Programmierung des V.24 Interfaces	13
4.1	Initialisierung des 2651	15
4.2	Kommunikation zwischen zwei DM V	16
4.2.1	Kommunikation mit PIP	18
5.	Anhang	19

1. Einführung und Beschreibung der RS232 Schnittstelle

Die V.24 Schnittstelle ist im Microcomputerbereich wohl eine der häufigsten Schnittstellen, um sowohl Rechner untereinander als auch Rechner mit Peripheriegeräten zu verbinden.

Die V.24 Schnittstelle,die der amerikanischen RS232-C weitgehend entspricht, ist eine serielle Schnittstelle. Die Ausgabe erfolgt als Einzelbits,die mit einem Startbit beginnen und mit einem oder zwei Stopbits enden.(Bild 1)

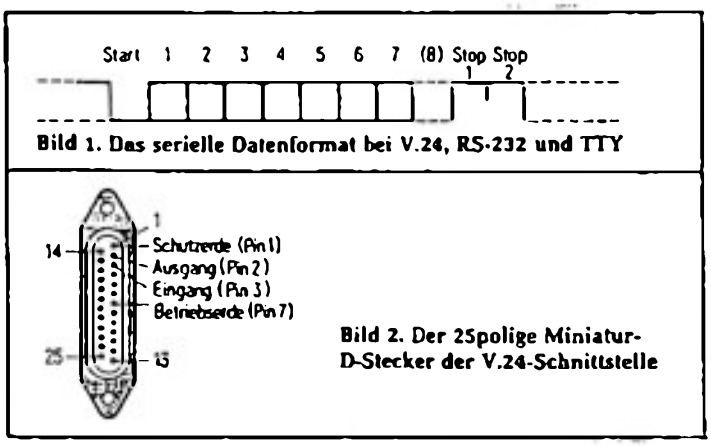


Bild 1. Das serielle Datenformat bei V.24, RS-232 und TTY

Bild 2. Der 25polige Miniatur-D-Stecker der V.24-Schnittstelle

Eine logische Eins entspricht einer Spannung zwischen -3V und -15V, eine logische Null entspricht einer Spannung zwischen +3V und +15V. Der Bereich zwischen -3V und +3V ist undefiniert.(Siehe Bild 3)
Die Übertragungsgeschwindigkeit reicht von 50 Baud bis 19200 Baud. Die maximale Leitungslänge beträgt ca. 30m ist aber von der Baudrate abhängig. Größere Längen sind mit eimem Modem zu erreichen.

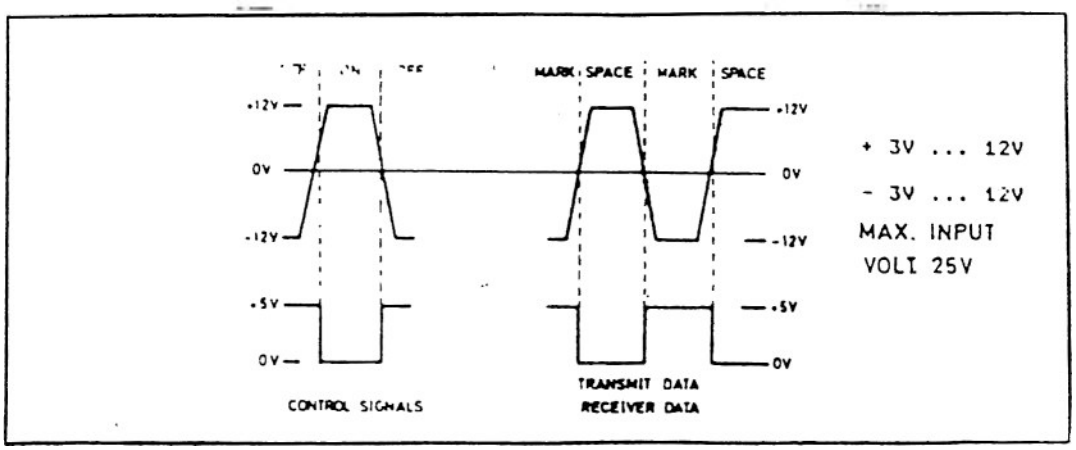
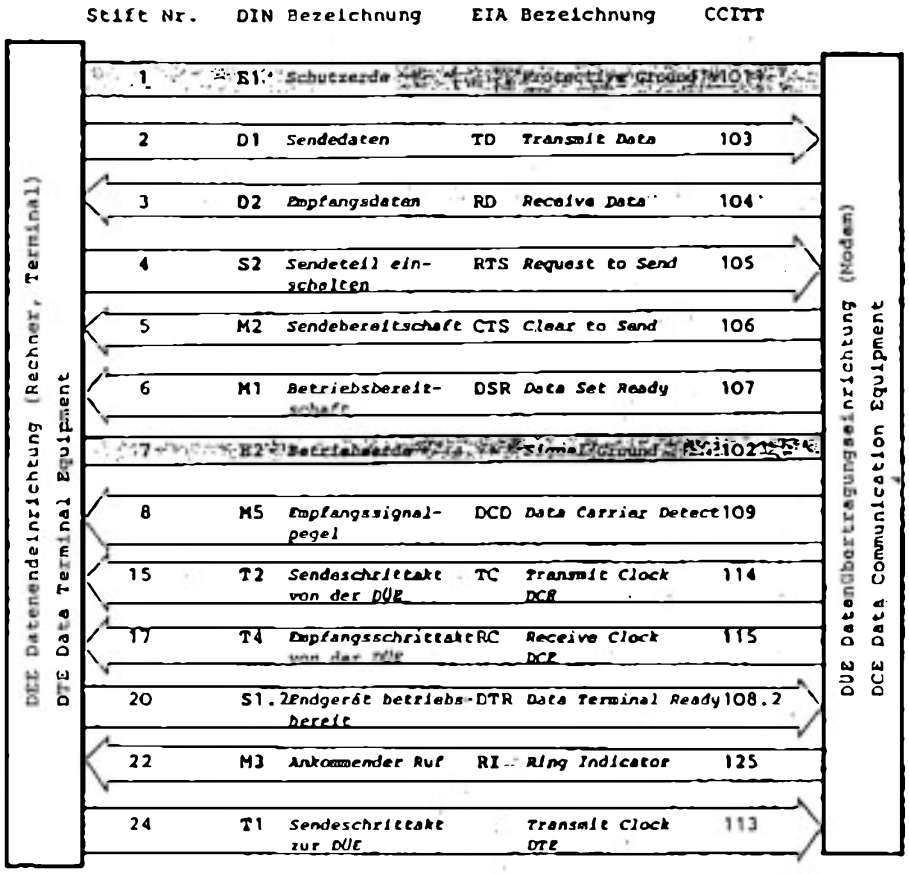


Bild 3: Spannungspegel der V.24 Schnittstelle.

1.2 Pinbelegung der V.24 Schnittstelle

Im folgenden sind die wichtigsten Leitungen schematisch dargestellt:



PIN BESCHREIBUNG

2 TRANSMIT DATA
Serielle Daten werden über diese Leitung vom Terminal gesendet. Logisch "1" ist LOW, logisch "0" ist HIGH. Während der Wartestellung ist die Leitung LOW.

3 RECEIVE DATA
Serielle Daten werden über diese Leitung vom Terminal empfangen. Logisch "1" ist LOW, logisch "0" ist HIGH. Während der Wartestellung ist die Leitung LOW.

PIN BESCHREIBUNG

4 REQUEST TO SEND
Terminal Output. Zeigt an, daß das Terminal sendebereit ist. Im Nicht-Modem Mode ist dieses Signal immer HIGH.
Im Modem und Full-Duplex Mode ist dieses Signal LOW, wenn keine Daten für das Terminal zum Senden vorliegen und HIGH, wenn das Terminal senden will.

5 CLEAR TO SEND
Terminal Input. Muß HIGH sein für eine Datenübertragung im Modem-Mode.
Im Nicht-Modem Mode wird diese Leitung nicht berücksichtigt.

6 DATA SET READY
Terminal Input. Muß HIGH sein für eine Datenübertragung im Modem-Mode.
Im Nicht-Modem Mode wird diese Leitung nicht berücksichtigt.

8 CARRIER DETECT
Terminal Input vom Modem oder ähnlichen Geräten.
Diese Leitung zeigt normalerweise an, daß das Modem den "DATA CARRIER" richtig empfängt.

20 DATA TERMINAL READY
Terminal Output. HIGH, wenn bereit zum Empfang.

1.3 Probleme mit der Norm

Oft ergeben sich beim Anschluß eines Peripheriegerätes mit V.24 Schnittstelle Probleme. Trotz Norm ergeben sich Probleme weil :

- die Verdrathung nicht übereinstimmt.
- falsche Kabel verwendet werden.
- falsche Baudraten eingestellt sind.
- das Übertragungsprotokoll nicht stimmt.

Lösung:Zunächst einmal muß festgestellt werden ob es sich um eine DCE oder DTE handelt

DCE - Data Communication Equipment - Datenübertragungseinrichtung.
DTE - Data Terminal Equipment - Datenendeinrichtung.

Bei einem Drucker handelt es sich um eine DTE, allerdings gibt es auch Drucker, die als DCE verdrahtet sind. Die DM V ist auch eine DTE mit DCE-Schnittstelle.
Aus dem Druckerhanduch ist nun zu entnehmen ob der Drucker eine DTE oder DCE-Schnittstelle hat.
Man kann es aber auch aus der Kennzeichnung der Pinbelegung ablesen:

DCE Pinbelegung	DTE Pinbelegung
2 = Eingang	2 = Ausgang
3 = Ausgang	3 = Eingang
4 = Eingang	4 = Ausgang
5 = Ausgang	5 = Eingang
6 = Ausgang	6 = Eingang
20= Eingang	20= Ausgang

Da man nun weiß ob es sich um eine DCE oder DTE handelt, kann man wie folgt verdrahten.

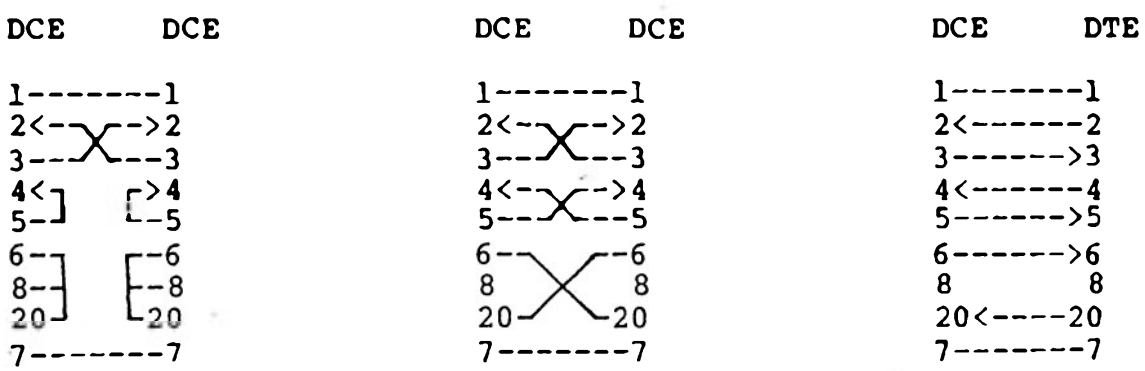


Bild 5: Verdrahtung zwischen DCE und DCE oder DCE und DTE.

1.4 Übertragungsprotokolle

Treten jetzt noch Probleme auf kann es eigentlich nur noch am Übertragungsprotokoll liegen. Bei der V.24 Schnittstelle gibt es drei verschiedene Übertragungsprotokolle.

- RDY/BSY Protokoll

Das Ready/Busy Protokoll ist ein "Hardware-Protokoll" d.h man braucht dazu keine Software. Das Protokoll läuft über Signalleitungen der V.24 Schnittstelle. Im einfachsten Fall genügt hierzu eine Leitung nämlich die DTR-Leitung. Ist diese Leitung positiv so besteht Empfangsbereitschaft. Negative Spannung hingegen zeigt den Busy-Status an.

- ETX/ACK Protokoll

Bei diesem Protokoll werden die ASCII-Zeichen ETX (03H) und ACK (06H)angewandt. Ist das Peripheriegerät bereit so wird DTR positiv und das Peripheriegerät sendet das ACK Zeichen an den Rechner. Dieser sendet die Daten, die mit einem ETX Zeichen abgeschlossen werden. Erkennt das Peripheriegerät das ETX Zeichen so sendet es wieder ACK zum Rechner und zeigt damit, daß das nächste Zeichen gesendet werden kann.
BEACHTE: Der ETX Code muß im Datenfluß des Rechners entsprechend der Pufferkapazität des Peripheriegerätes eingebracht werden.

- XON/XOFF Protokoll

Bei diesem Protokoll kommen die Steuerzeichen XON und XOFF zur Anwendung (ASCII Code DC1 und DC3). Das Peripheriegerät sendet bei Empfangsbereitschaft XON (11H) ansonsten XOFF (13H).

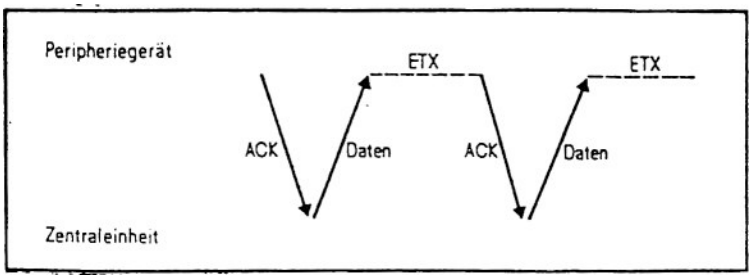


Bild 6: Schematische Darstellung des ACK/ETX Protokolls.

2. Tips zum Anschluß eines Druckers

NCR Drucker und andere Geräte sind voll kompatibel mit der DM V. Auch Drucker anderer Hersteller können ohne Änderungen mit der DM V laufen, oft tun sie es aber nicht d.h es sind teilweise Modifikationen nötig, um sie zum Laufen zu bringen.

Die Systemsoftware der DM V (CP/M und MS-DOS) unterstützt das XON-XOFF Protokoll. Das ETX/ACK Protokoll wird nicht unterstützt. Die CONFIG Utility erlaubt die Änderung folgender Parameter:

- Anzahl der Bits
- Paritäts Auswahl
- Anzahl der Stopbits
- Einstellen der Baud-Rate

2.1 Drucker ohne XON/XOFF Protokoll

Ist ein Drucker nicht in der Lage "Steuerzeichen" zur DM V zu senden,so müssen diese "Steuerzeichen" über Steuerleitungen der V.24 Schnittstelle realisiert werden. Leider sind diese Leitungen nicht genau definiert worden. Bei diesen "Steuerzeichen" handelt es sich meist um Puffer voll o.ä Signale, die auf verschiedenen Leitungen liegen können.Das sind z.B folgende Leitungen: 2,4,11,13,19,20. Schauen Sie in der Druckerinformation nach, welcher Pin gebraucht wird.

Das Druckerinterface (K212) ist für Pin 5 (RTS) als Steuerleitung vorbereitet.

STATUS PIN 5	DECISION MATE V	DRUCKER
ON (+12V)	ÜBERTRAGE DATEN	READY
OFF (-12V)	STOPPE DATENÜBERTRAGUNG	BUSY

Drucker die Pin 2,11,13 oder 19 verwenden müssen entsprechend verdrahtet werden.

2.2 Checkliste für Druckerinstallation

Wenn Drucker nicht druckt,

1. Prüfe Software (CONFIG)
2. Prüfe Schalterstellung von J1,J2 auf dem Interfaceboard
3. Prüfe Druckerverdrahtung
4. Prüfe ob Pin 5 (RTS) +12V

Wenn Drucker mit XON/XOFF Protokoll nicht richtig druckt, prüfe:

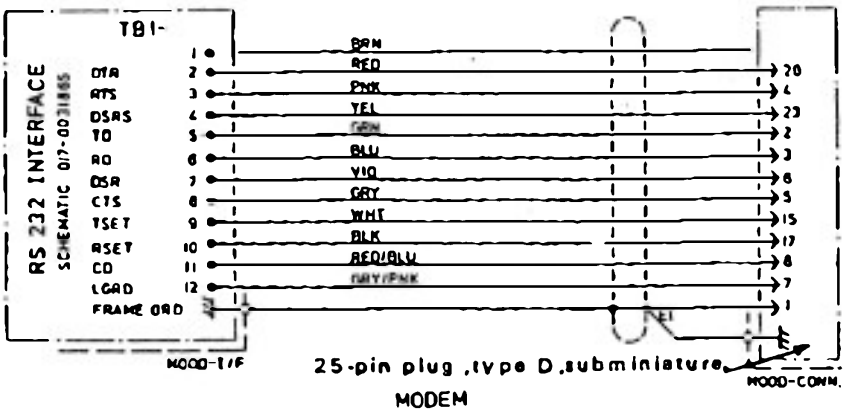
1. Pin 20 (DTR) +12V
2. Alle Verdrahtungen
3. Löte Brücke zwischen TB1-4 und TB1-8 auf dem Interfaceboard

bei anderen Protokollen ist zu prüfen, ob TB1-8 (CTS) seinen Status ändert, wenn der Drucker-Puffer voll ist. Ändert CTS seinen Wert nicht, ist dieser Pin (CTS) mit der Leitung des Druckers zu verbinden, die den Drucker-Puffer-Status anzeigt.

3. Hardware der V.24 Schnittstelle

Bei der DM V gibt es zur Zeit vier serielle Schnittstellen. Ein Drucker(K212), ein Modem(K211), ein gepuffertes(K215) und ein umschaltbares Interface.

K 211 - Anschluß an Modem, Barcodeleser etc.
K 212 - Anschluß an einen Drucker.
K 215 - wie K211 jedoch für höhere Baudraten.
K 801 - Anschluß an Modem,Drucker und Plotter.
Bei Verwendung mehrerer V.24 Interfaces.



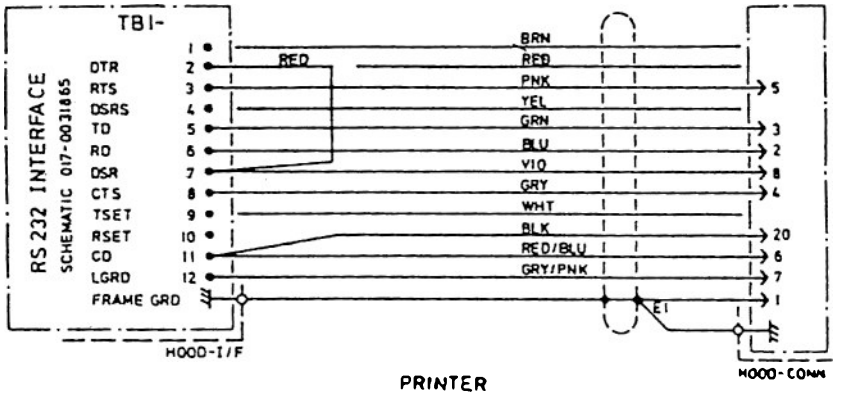


Bild 7 u. 8:Verdrathung von Modem und Druckerinterface.

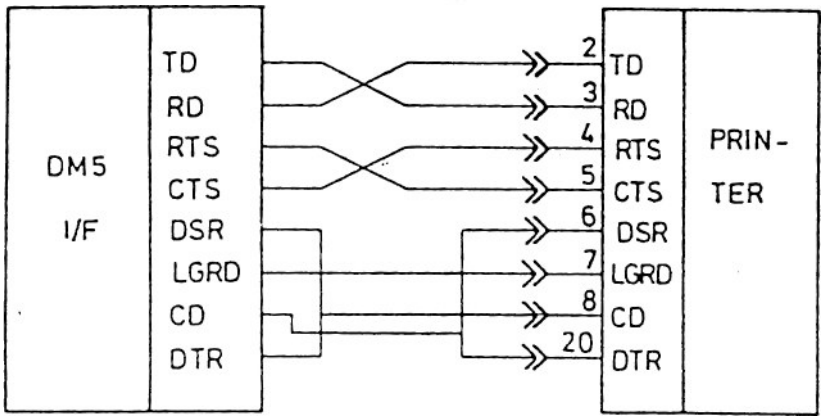
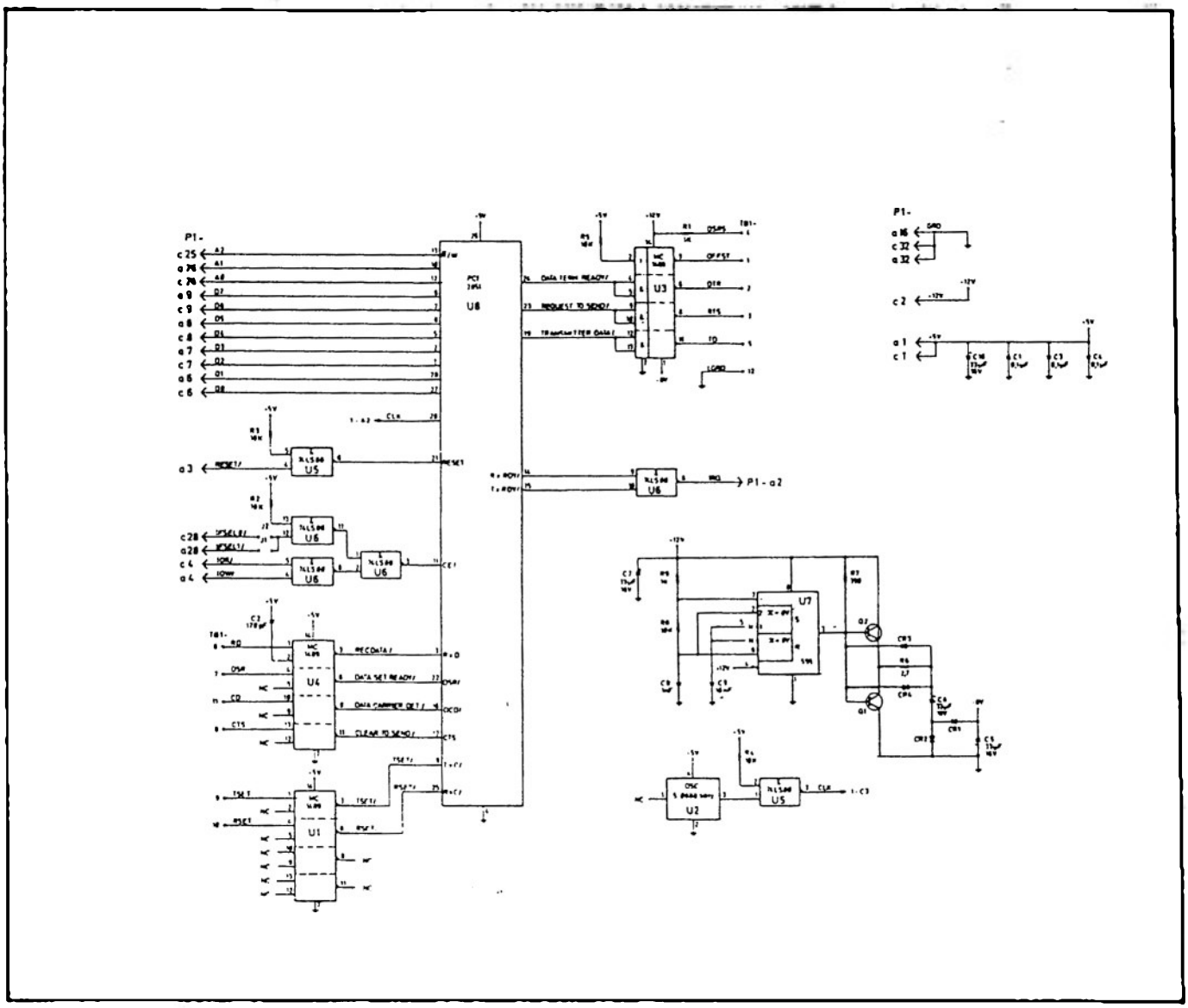


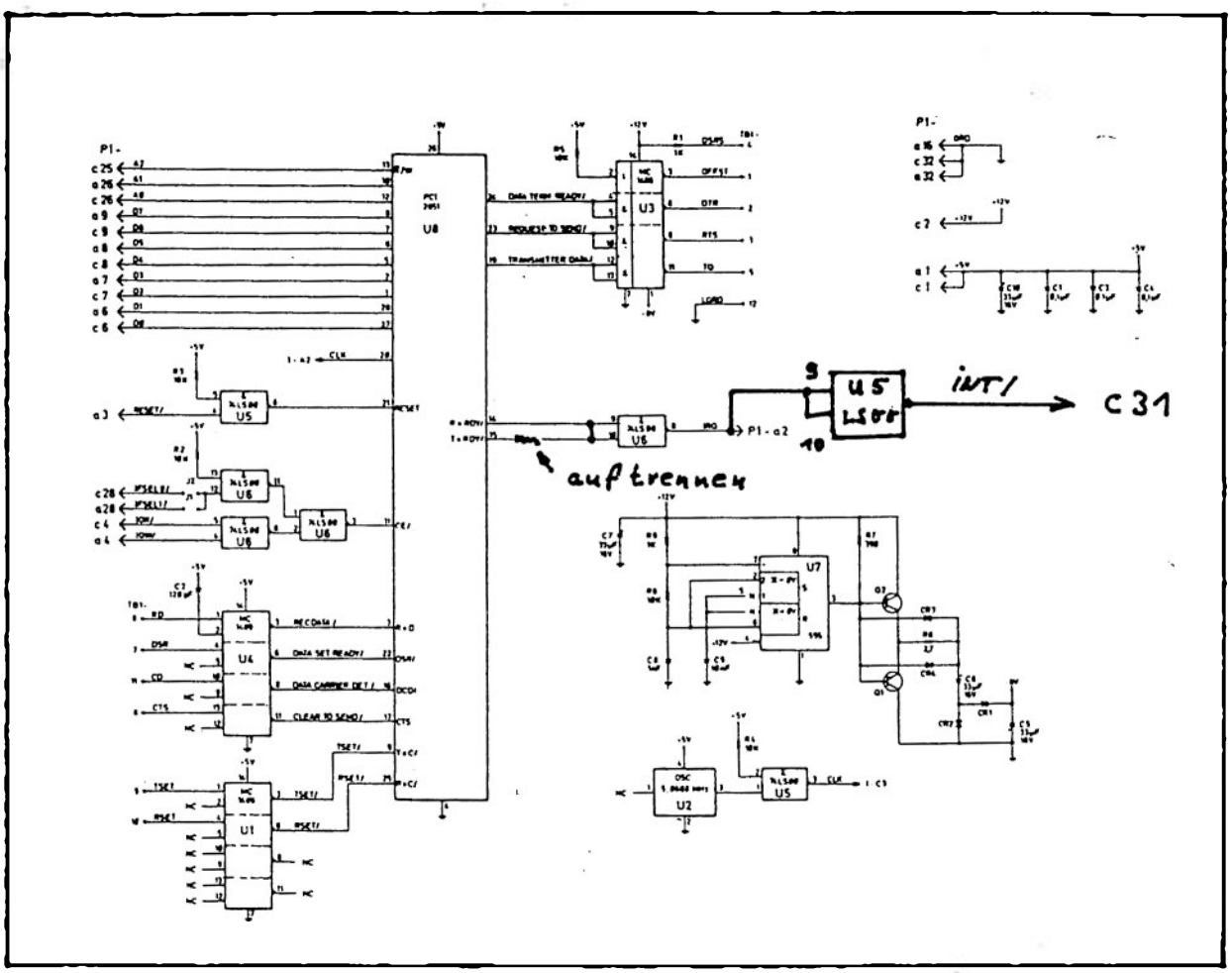
Bild 9:Beispiel für Anschluß eines Druckers an die DM V.

3.1 Schaltbild des V.24 Interface K211/K212

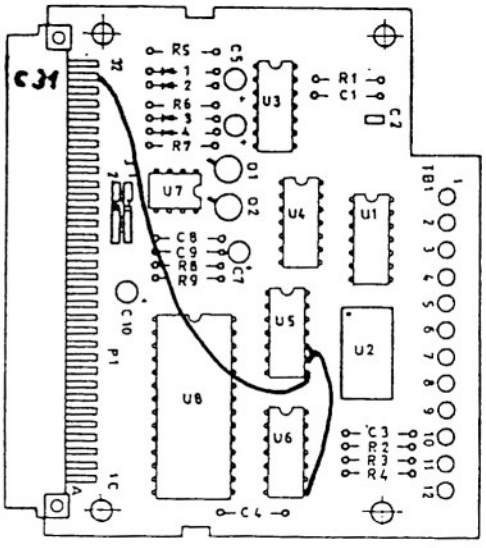


3.2 V.24 Interface wird interruptfähig K211/K212

Oft besteht der Wunsch durch ein empfangenes Zeichen einen Interrupt auszulösen. Dazu muß die Schaltung leicht geändert werden.



RS-232C ADAPTER



4. Programmierung des V.24 Interfaces

Die seriellen Schnittstellen der DM V (K211/K212)sind mit einem 2651 Chip realisiert worden, während das K801 einen 2661 Chip verwendet.Beider Programmierung gibt es aber keine Unterschiede. Da diese Chips von der DM V nur beim Drucken unterstützt werden, muß man sich für andere Anwendungen selbst einen Treiber schreiben.

Beim Programmieren des 2651 muß wie folgt vorgegangen werden:

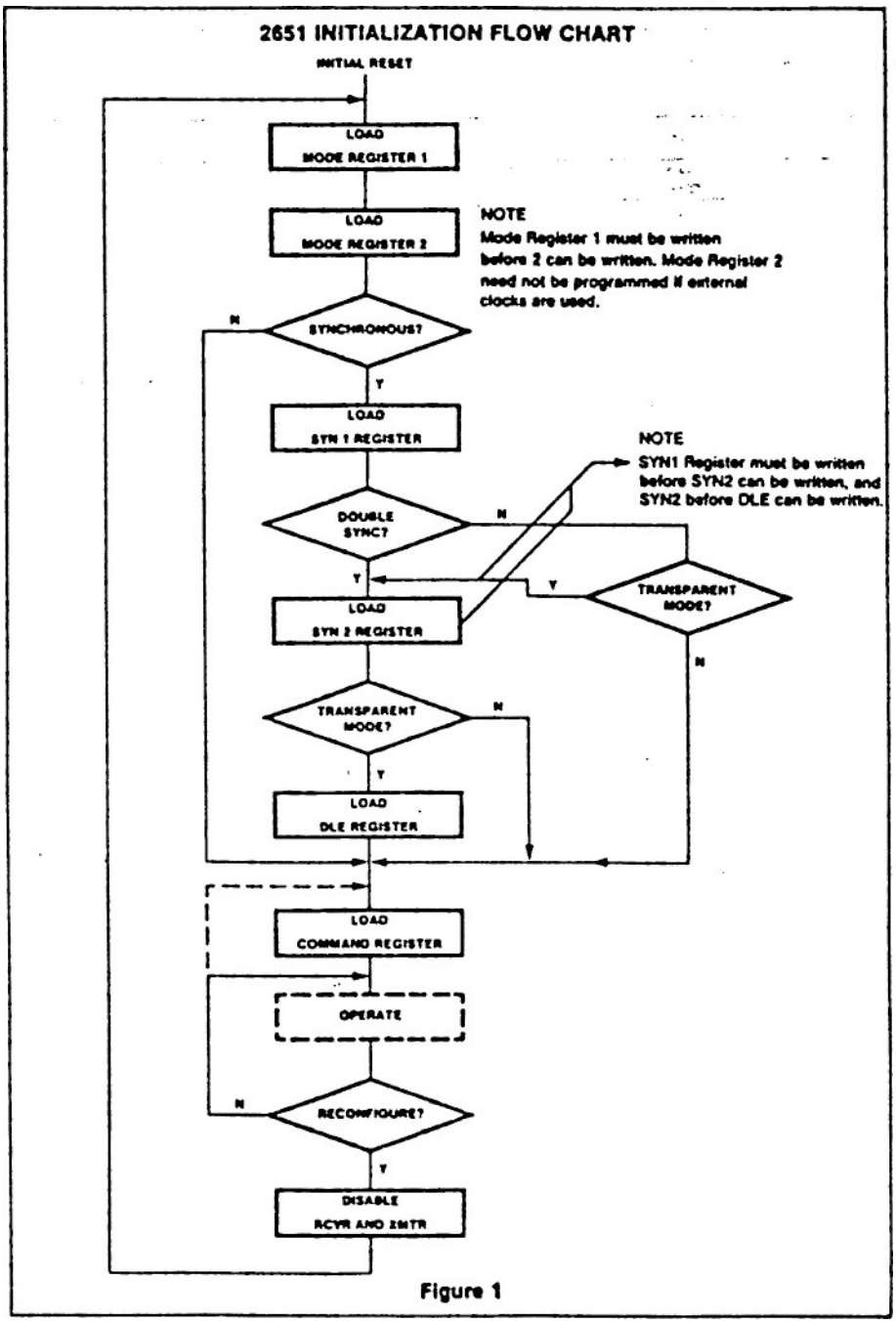


Figure 1

Um den 2651 richtig zu Programmieren ist es wichtig die Portadressen ,Mode und Commandregister zu kennen:

MODE REGISTER 1 (MR1)

MR17	MR16	MR15	MR14	MR13	MR12	MR11	MR10
		Parity Type	Parity Control	Character Length		Mode and Baud Rate Factor	
ASYNCH STOP BIT LENGTH 00 = INVALID 01 = 1 STOP BIT 10 = 1½ STOP BITS 11 = 2 STOP BITS		0 = ODD 1 = EVEN	0 = DISABLED 1 = ENABLED	00 = 5 BITS 01 = 6 BITS 10 = 7 BITS 11 = 8 BITS		00 = SYNCHRONOUS 1X RATE 01 = ASYNCHRONOUS 1X RATE 10 = ASYNCHRONOUS 16X RATE 11 = ASYNCHRONOUS 64X RATE	
SYNCH: NUMBER OF SYN CHAR 0 = DOUBLE SYN 1 = SINGLE SYN	SYNCH: TRANSPARENCY CONTROL 0 = NORMAL 1 = TRANSPARENT						

NOTE

Baud rate factor in asynchronous applies only if external clock is selected. Factor is 16X if internal clock is selected.

MODE REGISTER 2 (MR2)

MR27	MR26	MR25	MR24	MR23	MR22	MR21	MR20
		Transmitter Clock	Receiver Clock	Baud Rate Selection			
Ø	Ø	0 = EXTERNAL 1 = INTERNAL	0 = EXTERNAL 1 = INTERNAL	0000 = 50 BAUD 0001 = 75 0010 = 110 0011 = 134.5 0100 = 150 0101 = 300 0110 = 600 0111 = 1200		1000 = 1800 BAUD 1001 = 2000 1010 = 2400 1011 = 3600 1100 = 4800 1101 = 7200 1110 = 9600 1111 = 19,200	

COMMAND REGISTER (CR)

CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
Operating Mode		Request to Send	Reset Error		Receive Control (RxEN)	Data Terminal Ready	Transmit Control (TxEN)
00 = NORMAL OPERATION 01 = ASYNCH: AUTOMATIC ECHO MODE SYNCH: SYN AND/OR DLE STRIPPING MODE 10 = LOCAL LOOP BACK 11 = REMOTE LOOP BACK		0 = FORCE $\overline{RTS}$ OUTPUT HIGH 1 = FORCE $\overline{RTS}$ OUTPUT LOW	0 = NORMAL 1 = RESET ERROR FLAG IN STATUS REG (FE, OE, PE/DLE DETECT)	ASYNCH: FORCE BREAK 0 = NORMAL 1 = FORCE BREAK SYNCH: SEND DLE 0 = NORMAL 1 = SEND DLE	0 = DISABLE 1 = ENABLE	0 = FORCE $\overline{DTR}$ OUTPUT HIGH 1 = FORCE $\overline{DTR}$ OUTPUT LOW	0 = DISABLE 1 = ENABLE

STATUS REGISTER (SR)

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
Data Set Ready	Data Carrier Detect	FE/SYN Detect	Overrun	PE/DLE Detect	TxEMT/DSCHG	RxRDY	TxRDY
0 = $\overline{DSR}$ INPUT IS HIGH 1 = $\overline{DSR}$ INPUT IS LOW	0 = $\overline{DCD}$ INPUT IS HIGH 1 = $\overline{DCD}$ INPUT IS LOW	ASYNCH: 0 = NORMAL 1 = FRAMING ERROR SYNCH: 0 = NORMAL 1 = SYN CHAR DETECTED	0 = NORMAL 1 = OVERRUN ERROR	ASYNCH: 0 = NORMAL 1 = PARITY ERROR SYNCH: 0 = NORMAL 1 = PARITY ERROR OR DLE CHAR RECEIVED	0 = NORMAL 1 = CHANGE IN $\overline{DSR}$ OR $\overline{DCD}$, OR TRANSMIT SHIFT REGISTER IS EMPTY	0 = RECEIVE HOLDING REG EMPTY 1 = RECEIVE HOLDING REG HAS DATA	0 = TRANSMIT HOLDING REG BUSY 1 = TRANSMIT HOLDING REG EMPTY

2651 REGISTER ADDRESSING

K212	K211	$\overline{CE}$	A_1	A_0	$\overline{R}/\overline{W}$ & A_2	FUNCTION
–	–	1	X	X	X	Tri-state data bus
60H	70H	0	0	0	0	Read receive holding register
64H	74H	0	0	0	1	Write transmit holding register
61H	71H	0	0	1	0	Read status register
65H	75H	0	0	1	1	Write SYN1/SYN2/DLE registers
62H	72H	0	1	0	0	Read mode registers 1/2
66H	76H	0	1	0	1	Write mode registers 1/2
63H	73H	0	1	1	0	Read command register
67H	77H	0	1	1	1	Write command register

Use IN; OUT opcodes by Z80, 8088

4.1 Initialisierung des 2651/2661

folgende Routinen wurden mit der DM V unter CP/M 80
in Assembler erstellt.

EQU-TABELLE:

```
MODREG  EQU     76H     ;ADRESSE DER MODE-REGISTER
COMREG  EQU     77H     ;ADRESSE DES COMMAND-REGISTERS
READC   EQU     73H     ;ADRESSE DES READ-COMMAND-REGISTERS
STATUS  EQU     71H     ;ADRESSE DES STATUS-REGISTERS
READ    EQU     70H     ;ADRESSE DES EMPFANGS-REGISTERS
SENDE   EQU     74H     ;ADRESSE DES SENDE-REGISTERS
```

INITIALISIERUNGS-ROUTINE:

```
INIT:   IN      READC   ;RESET 2651 CHIP
        MVI     A,4DH   ;LADE MODE REGISTER 1: 8 BIT,ASYNC,
        OUT     MODREG  ;NO PARITY,1 STOP-BIT

        MVI     A,3EH   ;LADE MODE REGISTER 2:
        OUT     MODREG  ;9600 BAUD INTERNAL

        MVI     A,27H   ;LADE COMMAND REGISTER: ENABLE TRANSMIT,
        OUT     COMREG  ;RECEIVE, DTR OUTPUT LOW,  RTS OUTPUT LOW
```

Hilfreich ist oft auch noch eine Lese bzw. Sende Routine.

LESE ROUTINE:

```
LESE    IN      STATUS  ;LESE STATUS-REGISTER
WNI     02H     ;IST EIN ZEICHEN EMPFANGEN WORDEN ?

        JZ      LESE    ;NEIN  SPRINGE NACH LESE

        IN      READ    ;JA: LESE EMPFANGENES ZEICHEN
```

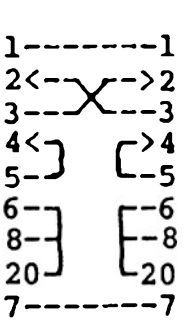
SENDE ROUTINE:

```
SENDEN  IN      STATUS  ;LESE STATUS-REGISTER
BNI     01H     ;IST BAUSTEIN SENDE-BEREIT ?

        JZ      SENDEN  ;NEIN: SPRINGE NACH SENDEN

        OUT     SENDE   ;JA: SENDEN ZEICHEN
```

Mit diesen 3 Routinen sollte es möglich sein ein kleines
Programm zu schreiben, daß 2 DM V miteinander kommunizieren läßt.

4.2 Kommunikation zwischen zwei DM V.

Zuerst müssen die Hardwarevoraussetzungen geschaffen werden. Da es sich bei der DM V um eine DTE mit DCE-Schnittstelle handelt (siehe Seite 6), können die beiden DM V so miteinander verbunden werden:



oftware für DM V - DM V Kommunikatuon

```
;*****************************************************************************
;
; EQU-TABELLE
;
MODRG   EQU     76H             ;ADRESSE DER MODE-REGISTER
COMRG   EQU     77H             ;ADRESSE DES COMMAND-REGISTERS
STATUS  EQU     71H             ;    "      "  STATUS -    "
READC   EQU     73H             ;    "      "  READ-COM.   "
READ    EQU     70H             ;READ-DATA
WRIT    EQU     74H             ;WRITE-DATA
BDOS    EQU     0005H           ;BDOS ENTRY
CPM     EQU     0000H           ;RüCKSPRUNG IN MONITOR
;
;
;
        ORG     100H            ;PROGRAMMIERUNG DES 2651
;  INIT
;  IT:  IN      READC           ;RESET 2651
        MVI     A,04DH          ;MODE 1 01001101 B
        OUT     MODRG           ;ASYNC.,8BITS,1 STOP-BIT,NO PARITY
        MVI     A,35H           ;MODE 2 00110101 B
        OUT     MODRG           ;300 BAUD INTERNAL
        MVI     A,27H
        OUT     COMRG           ;TX - RX ENABLE
        NOP
;
        IN      STATUS          ;STATUSABFRAGE
        ANI     38H
        JNZ     CPM             ;FEHLER BEIM INITIALISIEREN
;                               ;RüCKSPRUNG IN CP/M
;
LOOP    CALL    CONST           ;CONSOLE ABFRAGEN
        JNZ     SENDEN
;
        IN      STATUS          ;ABFRAGE DES STATUS
        ANI     02H
        JNZ     ANZEI           ;ZEICHEN EMPFANGEN ?
        JMP     LOOP
```

Nachfolgend die Sende und Anzeigeroutine:

```
;
;
SENDEN  MOV     E,A             ;DATEN VON CONSOLE SENDEN
READY   IN      STATUS
        ANI     01H
        JZ      READY           ;TX READY ?
;
        MOV     A,E
        OUT     WRIT            ;OUTPUT DATA
        JMP     ANZEIG
;
;
ANZEI   IN      READ            ;DATEN EILESEN
        ANI     7FH
        MOV     E,A
ANZEIG  MVI     C,2
        CALL    BDOS            ;ANZEIGE DER EMPFANGENEN ZEICHEN
        JMP     LOOP
;
;
; CP/M-ROUTINE PRüFT OB EIN TASTE GEDRüCKT WURDE.
; AKKU=0 NO DATA ;AKKU >0 TASTEN-CODE IN AKKU.
;
CONST:  PUSH    H       ;CONSOLE STATUS
        PUSH    D
        PUSH    B
        MVI     C,06H   ;BDOS FUNC. 6
        MVI     E,0FFH  ;NO CRT ECHO
        CALL    BDOS
        POP     B
        POP     D
        POP     H
        ORA     A
        RET
```

Beschreibung des Kommunikation-Programms:

Bei diesem Programm handelt es sich um ein einfaches Terminal-emulationsprogramm, d.h das Programm zeigt die über V.24 empfangenen Zeichen auf dem Bildschirm an. Sind also 2 DM V miteinander verbunden (s.S:13) so wird ein Zeichen von der ersten DM V gesendet (Tastendruck) und von der zweiten auf dem Bildschirm angezeigt und umgekehrt.

Nach der Initialisierung des 2651 Chips läuft das Programm solange in einer Schleife, bis entweder eine Taste gedrückt oder ein Zeichen über die V.24 Schnittstelle empfangen wurde. Nach einem Tastendruck wird das entsprechende ASCII-Zeichen, dieser Taste über V.24 gesendet und anschließend auf dem Bildschirm angezeigt.
Wurde ein Zeichen empfangen so wird es gleich auf dem Bildschirm angezeigt. Das Programm kehrt in die Schleife zurück und wartet erneut auf einen Tastendruck oder auf ein Zeichen.

4.2.1 Kommunikation mit PIP

Möchte man von einem anderen CP/M Rechner Daten zur DM V übertragen so kann man das auch mit Hilfe des PIP Befehls.

Die DM V empfängt mit folgendem Befehl:

```
PIP BEISPIEL.TXT=RDR:
```

Da die DM V nun empfangsbereit ist, kann nun der Fremdrechner seine Daten schicken:

```
PIP LST:=BEISPIEL.TXT
```

ist die Übertragung nun beendet muß der Fremdrechner noch ein EOF senden.

```
PIP LST:=EOF:
```

Voraussetzung das diese Übertragung funktioniert ist eine richtige Verdrahtung (s.S:16). Außerdem ist zu beachten, daß entweder ein K212 oder ein K801 mit Druckerinterface verwendet wird und das der Config richtig eingestellt ist.

```
;
;       Das ist das selbe Programm wie auf S:16/17
;       nur mit dem K215.
;
; PORT-ADRESSEN
;
BASE    EQU     70H
STATUS  EQU     BASE+1
KOMMAD  EQU     BASE+1
;
; FLAGS
;
IBF     EQU     2
OBF     EQU     1
;
; MODE+COMMANDWÖRTER
;
MODE1   EQU     0CDH
MODE2   EQU     35H
COMM    EQU     27H
;
; K215 KOMMANDOS
;
COM     EQU     0
MODL    EQU     1
STAT    EQU     3
SEND    EQU     9
RECEIV  EQU     8
BDOS    EQU     0005H
;
;
        ORG     100H
;
;
INIT    MVI     B,MODE1
        CALL    MODE            ;SCHREIBT MODE1 INS MODREG1
        MVI     B,MODE2
        CALL    MODE            ;SCHREIBT MODE2 INS MODEREG2
        MVI     B,COMM
        CALL    KOMMD           ;SCHREIBT COMMAND INS COMREG
;
;
LOOP    CALL    CONST           ;CONSOLE ABFRAGEN
        CNZ     SENDEN
;
        CALL    STATE           ;ZEICHEN EMPFANGEN ?
        ANI     02H
        CNZ     LESE
        JMP     LOOP
;
;       SCHREIBT MODE INS MODEREGISTER
;
MODE    IN      STATUS                  ;IST K215 BEREIT
        ANI     IBF
        JNZ     MODE
;
        MVI     A,MODL
        OUT     KOMMAD                  ;SCHICKT BEFEHL ZUM K215
;
MOD1    IN      STATUS
        ANI     IBF
```

```
        JNZ     MOD1                    ;IST K215 BEREIT ?
        MOV     A,B
        OUT     BASE
        RET
;
;
;       SCHREIBT COMMAND INS COMMANDREGISTER DES 2661
;
KOMMD   IN      STATUS
        ANI     IBF
        JNZ     KOMMD
        MVI     A,COM
        OUT     KOMMAD
;
KOM1    IN      STATUS
        ANI     IBF
        JNZ     KOM1
        MOV     A,B
        OUT     BASE
        RET
;
;       ROUTINE FRAGT STATUS DES K215 AB
;
STATE   IN      STATUS
        ANI     IBF
        JNZ     STATE
;
        MVI     A,STAT
        OUT     KOMMAD
;
STATE1  IN      STATUS
        ANI     OBF
        JZ      STATE1
        IN      BASE
        RET
;
;       ZEICHEN WIRD GESENDET UND ANGEZEIGT
;
SENDEN  MOV     B,A
SE      CALL    STATE
        ANI     01H
        JZ      SE
SEN     IN      STATUS
        ANI     IBF
        JNZ     SEN
        MVI     A,SEND
        OUT     KOMMAD
;
SEND1   IN      STATUS
        ANI     IBF
        JNZ     SEND1
        MOV     A,B
        OUT     BASE
;
        MOV     E,A
        MVI     C,2
        CALL    BDOS
;
        RET
```

```
;
;       ZEICHEN WIRD EMPFANGEN UND ANGEZEIGT
;
LESE    IN      STATUS
        ANI     IBF
        JNZ     LESE
;
        MVI     A,RECEIV
        OUT     KOMMAD
;
LESE1   IN      STATUS
        ANI     OBF
        JZ      LESE1
        IN      BASE
        ANI     7FH
        MOV     E,A
        MVI     C,2
        CALL    BDOS
        RET
;
;       FRAGT CONSOLE AB OB TASTE GEDRÜCKT,
;       FALLS $-TASTE ABBRUCH
;
CONST   PUSH    H
        PUSH    D
        PUSH    B
        MVI     C,06H
        MVI     E,0FFH
        CALL    BDOS
        POP     B
        POP     D
        POP     H
        CPI     '$'
        JZ      0000
        ORA     A
        RET
```

Quellennachweis:

Lesea,Zaks Mikroprozzesor-Interface Techniken.
Sybex-Verlag.

Sonderheft der Elektronik: Datenkommunikation.
Franzis-Verlag, München.

Handbuch der FX-Drucker.Epson Deutschland GmbH,
Düsseldorf.

Langer S. Die Schnittstelle RS-232.
MC Heft 9/82

Jürgen Plate. Schnittstellen.
MC Heft 7/83

Leonhard Sting. V.24 ganz einfach.
MC Heft 7/83

System Technical Manual Hardware.
NCR Corporation Dayton Ohio.

Programmable Communication Interface (PCI)
Preliminary Specification. VALVO

Erstellt durch:

Produkt Support GP
DM V
pspc-jm-250984

USART 2651 (MODE SET)

MODE REGISTER 1 (MR1)

MR17	MR16	MR15	MR14	MR13 MR12	MR11	MR10
		Parity Type	Parity Control	Character Length	Mode and Baud Rate Factor	
ASYNCH: STOP BIT LENGTH 00 = Invalid 01 = 1 Stop bit 10 = 1 1/2 Stop bits 11 = 2 Stop bits		0 = Odd 1 = Even	0 = Disabled 1 = Enabled	00 = 5 Bits 01 = 6 Bits 10 = 7 Bits 11 = 8 Bits	00 = Synchronous 1X rate 01 = Asynchronous 1X rate 10 = Asynchronous 16X rate 11 = Asynchronous 64X rate	
SYNCH: NUMBER OF SYN CHAR 0 = Double syn 1 = Single syn	SYNCH: TRANSPARENCY CONTROL 0 = Normal 1 = Transparent					
Note Baud rate factor in asynchronous applies only if external clock is selected. Factor is 16X if internal clock is selected.						

MODE REGISTER 2 (MR2)

MR27	MR26	MR25	MR24	MR23	MR22	MR21	MR20
		Transmitter Clock	Receiver Clock	Baud Rate Selection			
Always zero in NCR DMV		0 = External 1 = Internal	0 = External 1 = Internal	0000 = 50 Baud 0001 = 75 0010 = 110 0011 = 134.5 0100 = 150 0101 = 300 0110 = 600 0111 = 1200		1000 = 1800 Baud 1001 = 2000 1010 = 2400 1011 = 3600 1100 = 4800 1101 = 7200 1110 = 9600 1111 = 19.200	

COMMAND REGISTER (CR)

CR7 CR6	CR5	CR4	CR3	CR2	CR1	CR0
Operating Mode	Request to Send	Reset Error		Receive Control (RxEN)	Data Terminal Ready	Transmit Control (TxEN)
00 = Normal operation 01 = Asynch: atutomatic echo mode Synch: SYN and /or DLE stripping mode 10 = Local loop back 11 = Remote loop back	0 = Force $\overline{RTS}$ Output High 1 = Force $\overline{RTS}$ Output low	0 = Normal 1 = Reset error flag in status reg (FE, OE, PE/DLE detect)	ASYNCH: FORCE BREAK 0 = Normal 1 = Force break SYNCH: SEND DLE 0 = Normal 1 = Send DLE	0 = Disable 1 = Enable	0 = Force $\overline{DTR}$ Output High 1 = Force $\overline{DTR}$ Output low	0 = Disable 1 = Enable

STATUS REGISTER (SR)

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
Data Set Ready	Data Carrier Detect	FE/SYN Detect	Overrun	PE/DLE Detect	TxEMT/DSCHG	RxRDY	TxRDY
0 = $\overline{DSR}$ input is high 1 = $\overline{DSR}$ input is low	0 = $\overline{DCD}$ input is high 1 = $\overline{DCD}$ input is low	ASYNCH: 0 = Normal 1 = Framing error SYNCH: 0 = Normal 1 = SYN char detected	0 = Normal 1 = Overrun error	ASYNCH: 0 = Normal 1 = Parity error SYNCH: 0 = Normal 1 = Parity error or DLE char received	0 = Normal 1 = Change in $\overline{DSR}$ or $\overline{DCD}$, or transmit shift register is empty	0 = Receive holding reg empty 1 = Receive holding reg has data	0 = Transmit holding reg busy 1 = Transmit holding reg empty

2651 REGISTER ADDRESSING

K212/K213	K211	$\overline{CE}$	A_1	A_0	$\overline{R/W} = A_2$	FUNCTION
–	–	1	X	X	X	Tri-state data bus
60H	70H	0	0	0	0	Read receive holding register
64H	74H	0	0	0	1	Write transmit holding register
61H	71H	0	0	1	0	Read status register
65H	75H	0	0	1	1	Write SYN1/SYN2/DLE registers
62H	72H	0	1	0	0	Read mode registers 1/2
66H	76H	0	1	0	1	Write mode registers 1/2
63H	73H	0	1	1	0	Read command register
67H	77H	0	1	1	1	Write command register

Use IN; OUT opcodes by Z80, 8088

A N H A N G

Beschreibung aller V.24 Leitungen

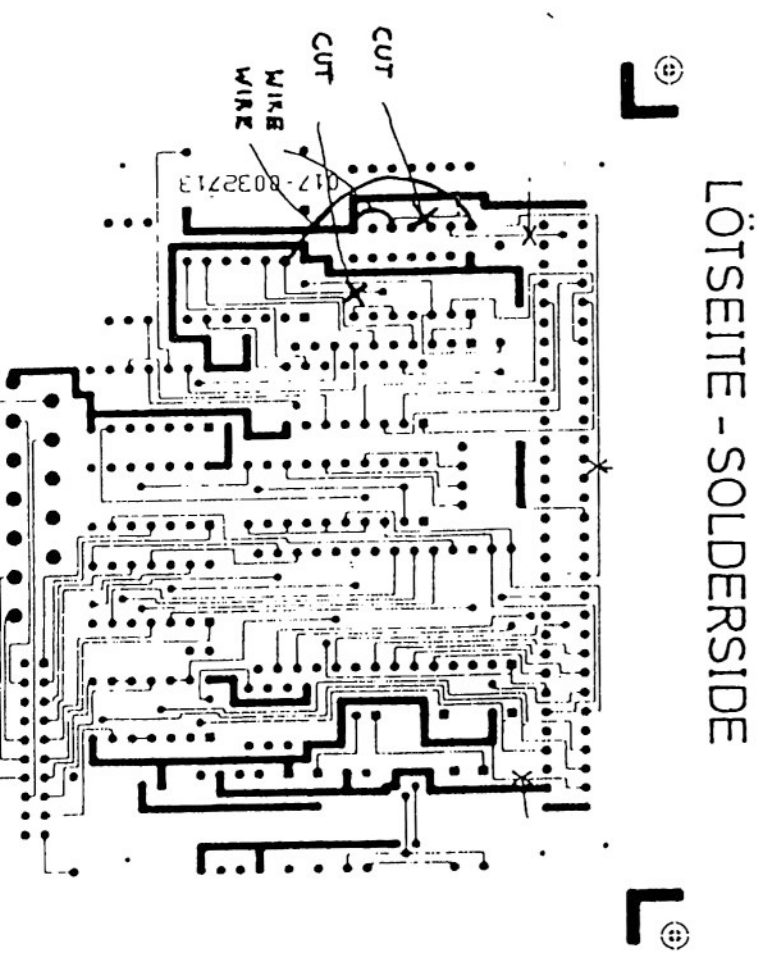
Tabelle 1: V.24-Schnittstellensignale

	Kurzzeichen			Stecker-belegung	Beschreibung		Richtung	
	CCITT V.24	EIA RS 232	DIN 66020		Deutsch	Englisch	Modem (DCE)	Terminal (DTE)
Erde	101	AA	E 1	1	Schutzerde	Protective ground	○—○	
	102	AB	E 2	7	Signalerde/Betriebserde	Signal ground/Common return	○—○	
Daten	103	BA	D 1	2	Sendedaten	Transmittad data (TD)	←—○	
	104	BB	D 2	3	Empfangsdaten	Received data (RD)	○—→	
Steuer- und Meldesignale	105	CA	S 2	4	Sendeteil einschalten	Request to send (RTS)	←—○	
	106	CB	M 2	5	Sendebereitschaft	Clear to send (CTS)	○—→	
	107	CC	M 1	6	Betriebsbereitschaft	Data set ready (DSR)	○—→	
	108.1		S 1.1	20	Übertragungsleitung anschalten	Connect data set to line	←—○	
	108.2	CD	S 1.2	20	Terminal betriebsbereit	Data terminal ready (DTR)	←—○	
	125	CE	M 3	22	Ankommender Ruf	Ring indicator (RI)	○—→	
	109	CF	M 5	8	Empfangssignalpegel	Received line signal detector (DCD) (Carrier detector)	○—→	
	110	CG	M 6	21	Empfangsgüte	Signal quality detector	○—→	
	111	CH	S 4	23	Übertragungsgeschwindigkeit (Wahl vom Terminal)	Data signal rate selector (DTE)	←—○	
	112	CI	M 4	23	Übertragungsgeschwindigkeit (Wahl vom Modem)	Data signal rate selector (DCE)	○—→	
	126	CK	S 5	11	Wahl Sendefrequenz (200 baud Modem)	Select transmit frequency (200 baud modem)	←—○	
Takte	113	DA	T 1	24	Sendeschrittakt von DEE	Transmitter signal element timing (Transmit clock to modem DTE)	←—○	
	114	DB	T 2	15	Sendeschrittakt von DÜE	Transmitter signal element timing (TC) (Transmit clock from modem DCE)	○—→	
	115	DD	T 4	17	Empfangsschrittakt	Receiver signal element timing (RC) Receive clock	○—→	
Zusatzkanal	118	SBA	HD 1	14	Sendedaten Rückkanal	Secondary transmitted data	←—○	
	119	SBB	HD 2	16	Empfangsdaten Rückkanal	Secondary received data	○—→	
	120	SCA	HS 2	19	Rückkanal Sendeteil einschalten	Secondary request to send	←—○	
	121	SCB	HM 2	13	Rückkanal Sendebereitschaft	Secondary clear to send	○—→	
	122	SCF	HM 5	12	Rückkanal Empfangssignalpegel	Secondary Carrier detector	○—→	
Frei				9/10	Zur Verwendung für Prüfgeräte	Reserved for data set testing		
				11/18/25	Nicht belegt	Unasigned		

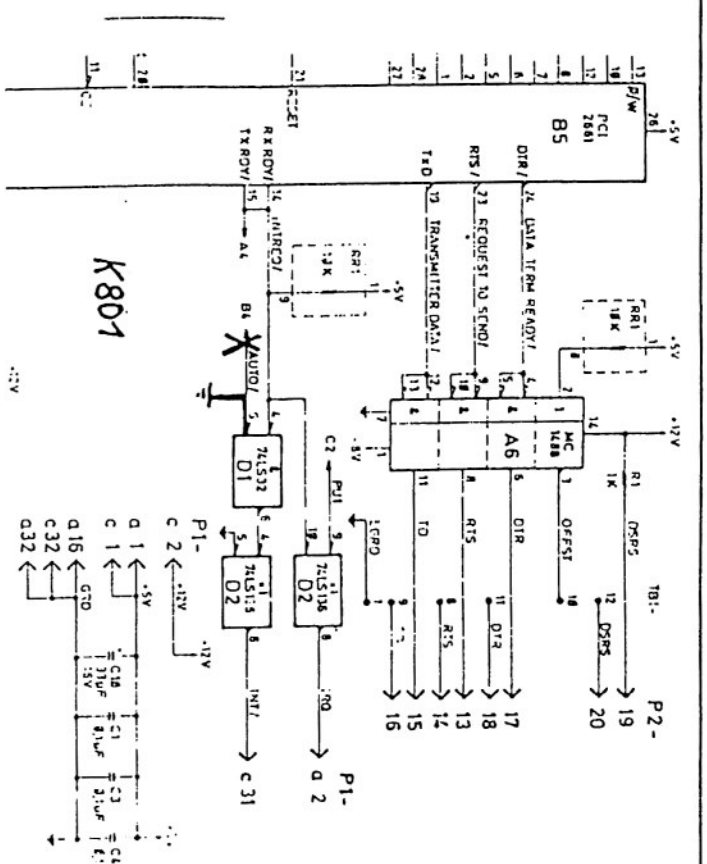
Normübersicht

Empfehlungen (Genf 1976)

V. 1	Äquivalenz zwischen Binärzeichen und den Kennzuständen eines Zwei-Zustand-Codes
V. 2	Leistungspegel für Datenübertragung über Fernsprechleitungen
V. 3	Internationales Alphabet Nr. 5
V. 4	Allgemeine Struktur von Signalen, die nach dem Alphabet Nr. 5 codiert sind
V. 5	Normierung der Übertragungsgeschwindigkeiten für synchrone Datenübertragung über das öffentliche Fernsprechwählnetz
V. 6	Normierung der Übertragungsgeschwindigkeit für synchrone Datenübertragung auf vermieteten (ständig überlassenen, fest geschalteten) Fernsprechleitungen
V. 10	Elektrische Eigenschaften für unsymmetrische Doppelstrom-Schnittstellenleitungen zur allgemeinen Benutzung mit integrierten Schaltkreisen im Bereich der Datenkommunikation (identisch mit X. 26)
V. 11	Elektrische Eigenschaften für symmetrische Doppelstrom-Schnittstellenleitungen zur allgemeinen Benutzung mit integrierten Schaltkreisen im Bereich der Datenkommunikation (identisch mit X. 27)
V. 15	Anwendung von akustischer Kopplung für die Datenübertragung
V. 16	Modems für die Übermittlung analoger medizinischer Daten
V. 19	Modems mit Parallelübertragung unter Verwendung der Fernsprechsignalisierungsfrequenzen
V. 20	Modems mit Parallelübertragung zur allgemeinen Benutzung im öffentlichen Fernsprechwählnetz
V. 21	200-Baud-Modem zur Benutzung im öffentlichen Fernsprechwählnetz
V. 23	600/1200-Baud-Modem zur Benutzung im öffentlichen Fernsprechwählnetz
V. 24	Liste der Definitionen für Schnittstellenleitungen zwischen Datenendeinrichtungen und Datenübertragungseinrichtungen
V. 25	Automatische Wähl- und/oder Anrufbeantwortungseinrichtung im öffentlichen Fernsprechwählnetz und Abschaltung von Echosperren bei handvermittelten Verbindungen
V. 26	Modem mit 2400 bit/s zur Benutzung auf festgeschalteten Vierdraht-Leitungen
V. 26bis	Modem mit 2400/1200 bit/s zur Benutzung im öffentlichen Fernsprechwählnetz
V. 27	Modem für eine Übertragungsgeschwindigkeit von 4800 bit/s auf festgeschalteten Leitungen und manuellem Entzerrer
V. 27bis	Modem für eine Übertragungsgeschwindigkeit von 4800 bit/s auf festgeschalteten Leitungen und automatischem Entzerrer
V. 27ter	Modem mit einer Übertragungsgeschwindigkeit von 4800/2400 bit/s zur Benutzung im öffentlichen Fernsprechwählnetz
V. 28	Elektrische Eigenschaften für unsymmetrische Doppelstrom-Schnittstellenleitungen
V. 29	Modem für eine Übertragungsgeschwindigkeit von 9600 bit/s zur Benutzung auf festgeschalteten Leitungen
V. 31	Elektrische Eigenschaften für Einfachstrom-Schnittstellenleitungen mit Kontakten
V. 35	Datenübertragung mit 48 kbit/s über Primärgruppenleitungen im Bereich von 60 bis 108 kHz
V. 36	Modem zur synchronen Datenübertragung auf Primärgruppenleitungen (60 bis 108 kHz)
V. 40	Fehleranzeige mit elektromechanischen Einrichtungen
V. 41	Vom Code unabhängiges System des Fehlerschutzes
V. 50	Standardgrenzwerte für die Übertragungsgüte von Datenübertragung



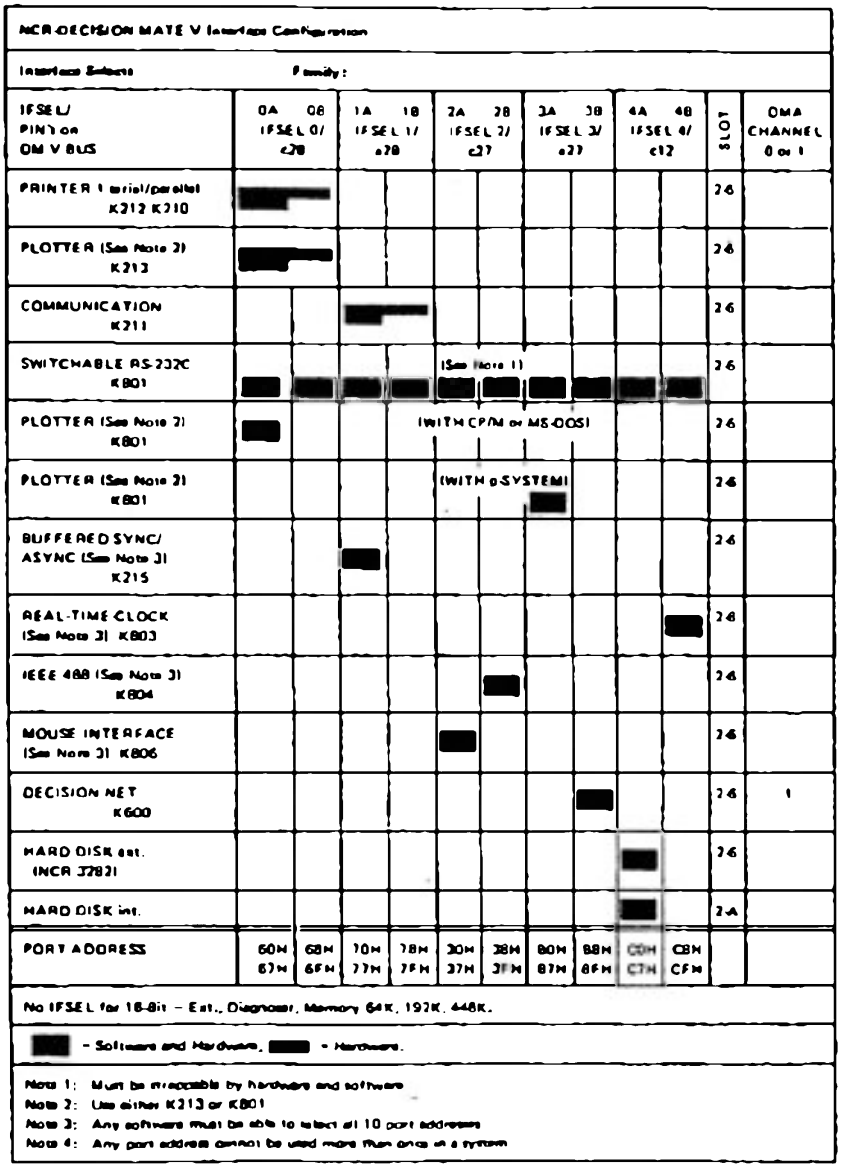
LÖTSEITE - SOLDERSIDE



You can use this interrupt only if you have a special application.
For other applications you can't use this interface.

Do not use with K235

Übersicht der DM V Port-Adressen



NCR DECISION MATE V Interface Configuration

Interface Selects — Family:

IFSEL/ PIN 1 on DM V BUS	0A 0B IFSEL 0/ c28	1A 1B IFSEL 1/ a28	2A 2B IFSEL 2/ c27	3A 3B IFSEL 3/ a27	4A 4B IFSEL 4/ c12	SLOT	DMA CHANNEL 0 or 1
PRINTER (serial/parallel) K212 K210						26	
PLOTTER (See Note 2) K213						26	
COMMUNICATION K211						26	
SWITCHABLE RS-232C K801			(See Note 1)			26	
PLOTTER (See Note 2) K801			(WITH CP/M or MS-DOS)			26	
PLOTTER (See Note 2) K801			(WITH oSYSTEM)			26	
BUFFERED SYNC/ ASYNC (See Note 3) K215						26	
REAL-TIME CLOCK (See Note 3) K803						26	
IEEE 488 (See Note 3) K804						26	
MOUSE INTERFACE (See Note 3) K806						26	
DECISION NET K600						26	1
HARD DISK ext. (NCR 7282)						26	
HARD DISK int.						2A	
PORT ADDRESS	60H 61H / 68H 69H	70H 71H / 78H 79H	30H 31H / 38H 39H	80H 81H / 88H 89H	C0H C1H / C8H C9H		

No IFSEL for 16-Bit – Est., Diagnoser, Memory 64K, 192K, 448K.

■ – Software and Hardware, ■ – Hardware.

Note 1: Must be reappable by hardware and software
Note 2: Use either K213 or K801
Note 3: Any software must be able to select all 10 port addresses
Note 4: Any port address cannot be used more than once in a system

NCR DECISION MATE V

CENTRONICS-ADAPTER
(K210-V001)

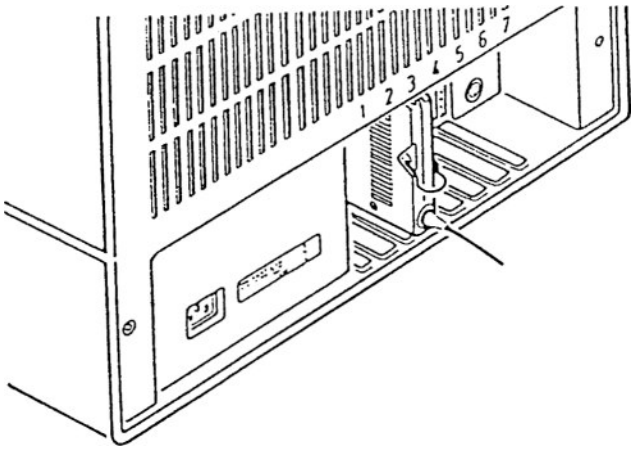
Die beiliegenden Seiten zeigen Ihnen, wie Sie diese Leistungserweiterung an Ihren NCR DECISION MATE V anschließen können. Bitte ordnen Sie diese Beschreibung in Ihre Bedienungsanleitung für den NCR DECISION MATE V ein.

Doc:017-0033059

NCR ist ständig bemüht, die Produkte im Zuge der Entwicklung von Technologie, Bauteilen, Soft- und Firmware dem neuesten Stand anzupassen. NCR behält sich deshalb das Recht vor, Spezifikationen ohne vorherige Ankündigung zu ändern.
Nicht alle hier beschriebenen Leistungen werden von NCR in allen Teilen der Welt vertrieben. Nähere Informationen bezüglich eventueller Einschränkungen oder Erweiterungen sowie den aktuellen Stand erfahren Sie von Ihrem Händler oder der nächstgelegenen NCR-Geschäftsstelle.

CENTRONICS-ADAPTER
(K210-V001)

1. Setzen Sie diesen Adapter in eine der an der Rückseite Ihres NCR DECISION MATE V befindlichen Fassungen (2...6) ein.

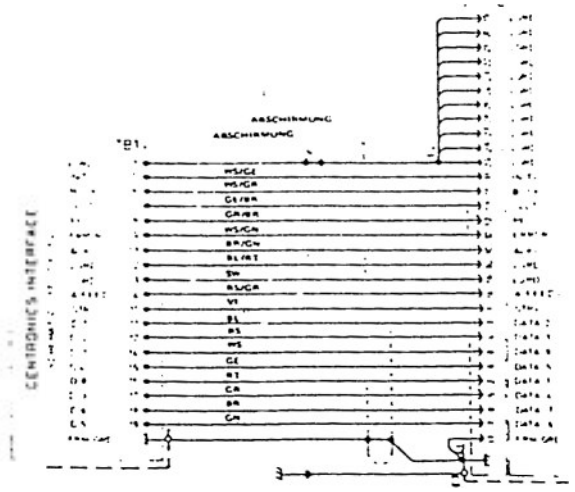


2. Schließen Sie den am Kabelende vorhandenen Stecker an einen Centronics-kompatiblen Drucker an.
3. Das zu Ihrem Drucker gehörende Handbuch beschreibt die Handhabung von Druckpapier und Farbband.
4. Schließen Sie den Drucker an die erforderliche Netzspannung an.

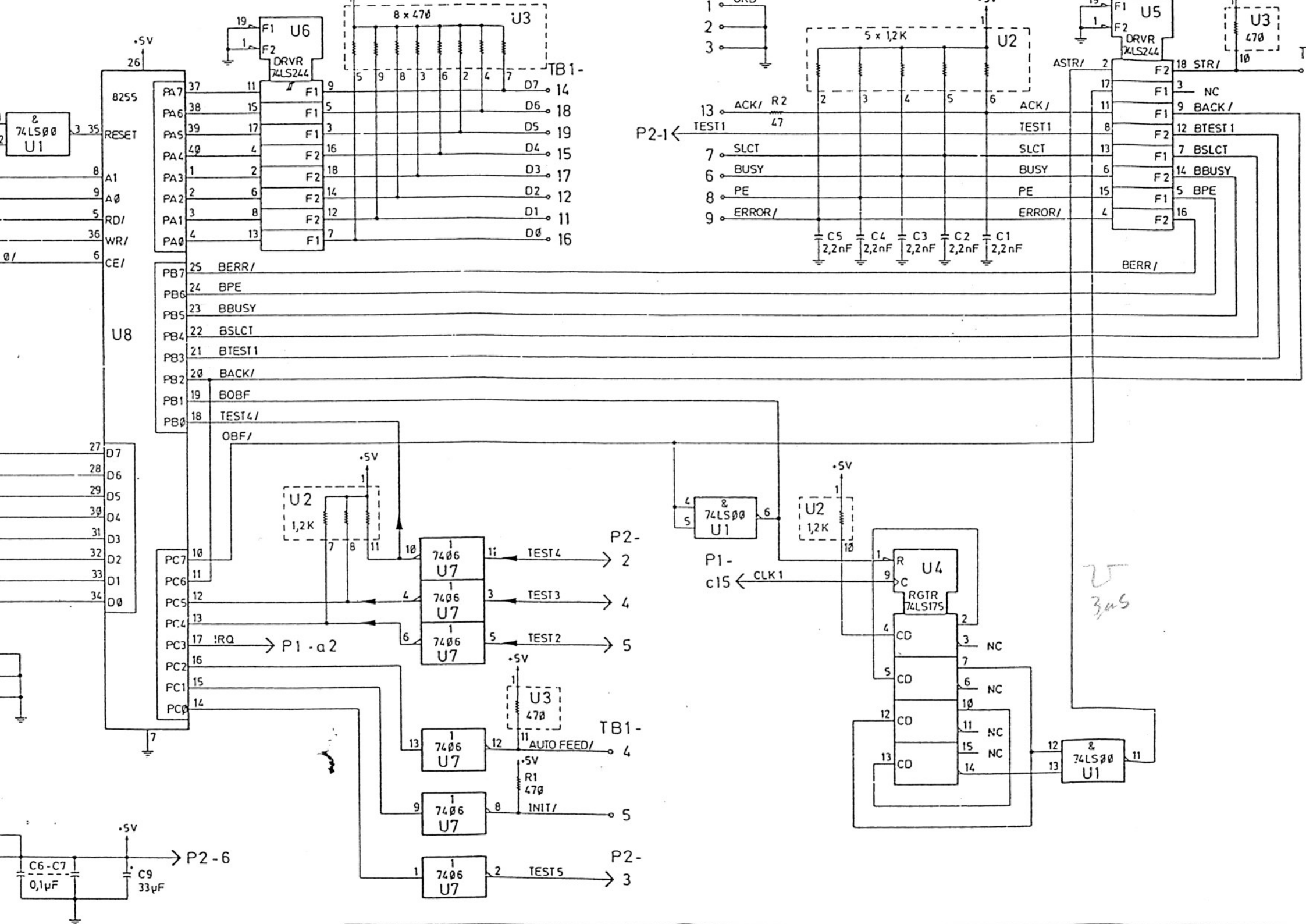
WICHTIG: Der gleichzeitige Betrieb dieses Adapters mit dem Adapter K212 oder K213 ist nicht möglich.

Umseitig finden Sie eine Übersicht der im Rahmen des Übertragungsprotokolls benutzten Leitungen.

STROBE/	Taktfrequenz zur Druckersynchronisierung. Mindesttaktzeit 1 µsec. Empfang erfolgt während Signal-"low".
ACKNLG/	Signal (2.5 bis 10 µsec) zur Datenempfangsbestätigung. Empfang neuer Daten vor Rücksetzung dieses Signals unmöglich.
BUSY	Datenannahme nicht möglich aus einem der folgenden Gründe : - während des Datenempfangs (je nach Drucker) - während des Druckvorgangs - im Off-Line-Zustand - beim Vorliegen eines Fehlerzustandes - während des Zeilenvorschubs
DATA 1-8	Parallele Datenübertragung ("High" = log 1; "Low" = log 0)
PE	"High" zeigt das Papierende an.
SLCT	Signal zeigt Wahlzustand des Druckers an (je nach Drucker)
AUTO FEED/	Automatischer Zeilenvorschub (nicht bei allen Druckern).
INIT/	Computersignal zur Druckerrückstellung (je nach Drucker).
ERROR/	Fehlerzustand des Druckers (z.B. Papierende, Off-Line, o.ä).
TEST 1-4	Eventuell vorhandene Prüfleitungen.



Doc:017-0033059


8255
U8
RESET
A1
A0
RD/
WR/
CE/
PA7
PA6
PA5
PA4
PA3
PA2
PA1
PA0
PB7
PB6
PB5
PB4
PB3
PB2
PB1
PB0
PC7
PC6
PC5
PC4
PC3
PC2
PC1
PC0
D7
D6
D5
D4
D3
D2
D1
D0
BERR/
BPE
BBUSY
BSLCT
BTEST1
BACK/
BOBF
TEST4/
OBF/
IRQ
P1-a2
U6
DRVR
74LS244
8 x 470
U3
TB1-
U2
5 x 1,2K
ACK/
TEST1
SLCT
BUSY
PE
ERROR/
R2
47
P2-1
C5
C4
C3
C2
C1
2,2nF
U5
ASTR/
STR/
NC
BACK/
BTEST1
BSLCT
BBUSY
BPE
BERR/
U1
74LS00
U7
7406
TEST4
TEST3
TEST2
TEST5
P2-2
P2-4
P2-5
P2-3
P2-6
AUTO FEED/
INIT/
R1
470
U4
RGTR
74LS175
CLK1
P1-c15
C6-C7
0,1µF
C9
33µF
+5V

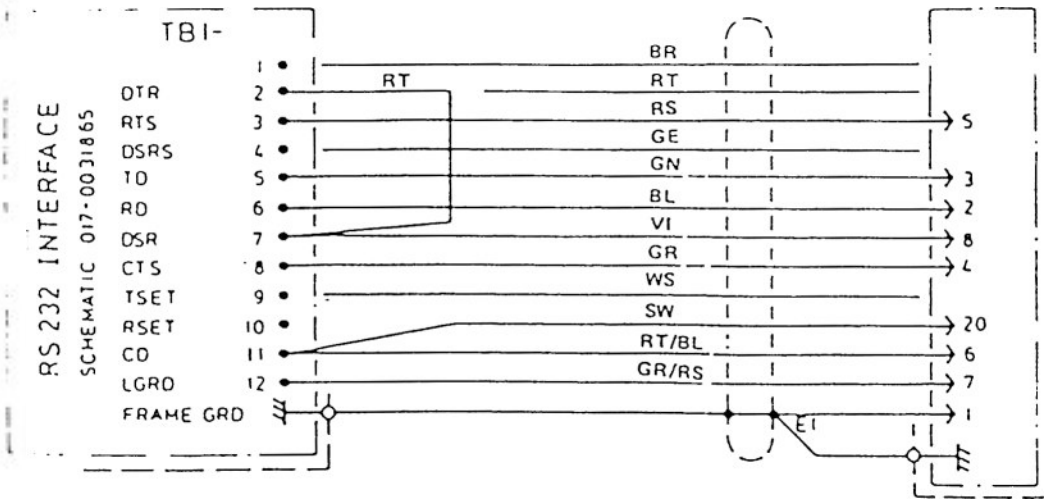
NCR DECISION MATE V

RS-232C-DRUCKER-ADAPTER (K212)

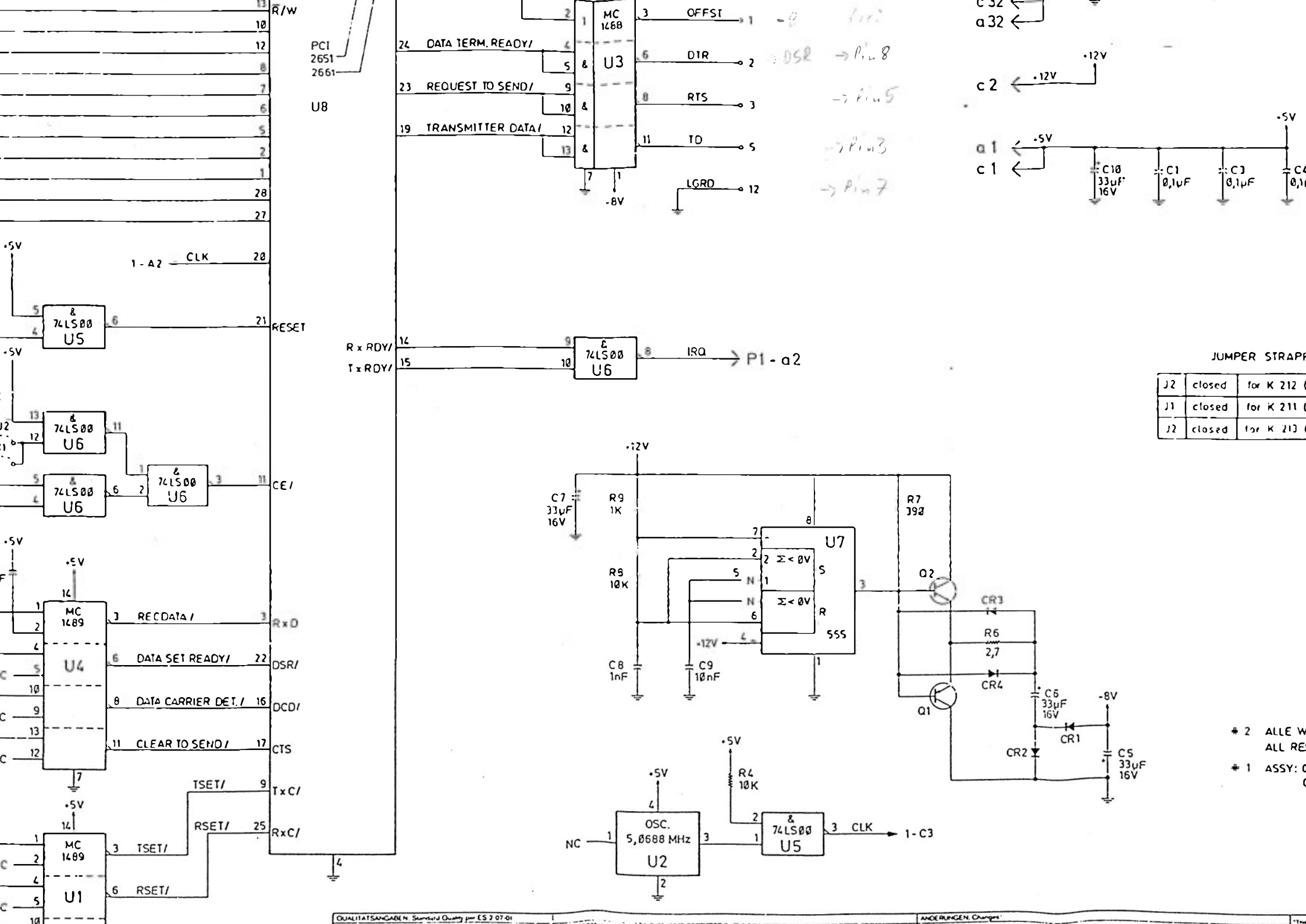
Die beiliegenden Seiten zeigen Ihnen, wie Sie diese Leistungserweiterung an Ihren NCR DECISION MATE V anschließen können. Bitte ordnen Sie diese Beschreibung in Ihre Bedienungsanleitung für den NCR DECISION MATE V ein.

Kit:603-6091230 Doc:017-0033061

TRANSMIT DATA (TD)	Übertragungsleitung des Computers zum Senden von Daten fester Länge (5...8 Bit). Nach der Übertragung jedes Zeichens ist das Signal dieser Leitung "low" (logisch = 1)
RECEIVE DATA (RD)	Über diese Leitung empfängt der Computer Daten. Zeichenlänge und Signalpegel wie TD.
CLEAR TO SEND (CTS)	Eingangssignal an den Computer: Die Übertragung von Daten erfolgt bei "high".
REQUEST TO SEND (RTS)	Sendebereitschaftssignal vom Computer. In der nicht-Modem-Betriebsart ist das Signal immer "high". In der Modem-(Halbduplex-)Betriebsart ist das Signal "high", wenn Daten zur Übertragung bereitstehen, "low" sonst.
DATA SET READY (DSR)	Eingangssignal an den Computer: "high" zeigt dem Computer an, daß Daten zum Empfang bereitstehen.
DATA TERMINAL READY (DTR)	Ausgangssignal vom Computer: Der Computer ist bereit, Daten zu empfangen.
CARRIER DETECT (CD)	Eingangssignal an den Computer: "high" zeigt den ordnungsgemäßen Empfang des Trägersignals des fremden Modems an.



Doc:017-0033061


R/W
PCI 2651 2661
U8
24 DATA TERM. READY/
23 REQUEST TO SEND/
19 TRANSMITTER DATA/
MC 1488
U3
OFFST
DTR
RTS
TD
LGRD
-8V
+12V
+5V
c 32
a 32
c 2
a 1
c 1
C10 33µF 16V
C1 0,1µF
C3 0,1µF
C4
1 - A2
CLK
RESET
74LS00
U5
U6
CE/
R x RDY/
T x RDY/
IRQ
P1 - a2
JUMPER STRAPP
J2 closed for K 212
J1 closed for K 211
J2 closed for K 213
C7 33µF 16V
R9 1K
R8 10K
U7
555
R7 390
Q2
Q1
CR3
R6 2,7
CR4
CR1
CR2
C6 33µF 16V
C5 33µF 16V
MC 1489
U4
U1
REC DATA/
DATA SET READY/
DATA CARRIER DET./
CLEAR TO SEND/
TSET/
RSET/
RxD
DSR/
DCD/
CTS
TxC/
RxC/
C8 1nF
C9 10nF
R4 10K
OSC. 5,0688 MHz
U2
NC
CLK
1-C3
2 ALLE W
ALL RE
1 ASSY: C

NCR DECISION MATE V

CENTRONICS-ADAPTER
(K210-V001)

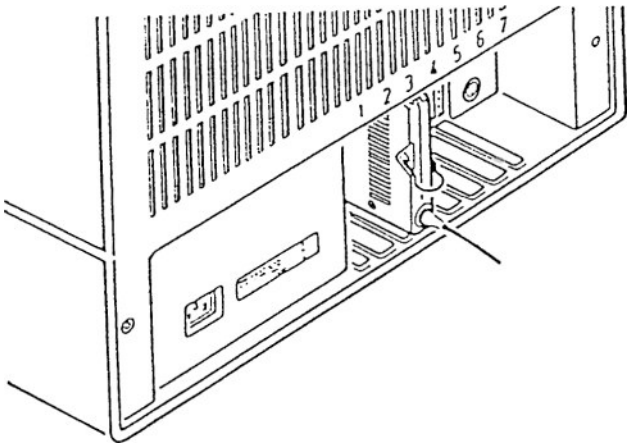
Die beiliegenden Seiten zeigen Ihnen, wie Sie diese Leistungserweiterung an Ihren NCR DECISION MATE V anschließen können. Bitte ordnen Sie diese Beschreibung in Ihre Bedienungsanleitung für den NCR DECISION MATE V ein.

Doc:017-0033059

NCR ist ständig bemüht, die Produkte im Zuge der Entwicklung von Technologie, Bauteilen, Soft- und Firmware dem neuesten Stand anzupassen. NCR behält sich deshalb das Recht vor, Spezifikationen ohne vorherige Ankündigung zu ändern.
Nicht alle hier beschriebenen Leistungen werden von NCR in allen Teilen der Welt vertrieben. Nähere Informationen bezüglich eventueller Einschränkungen oder Erweiterungen sowie den aktuellen Stand erfahren Sie von Ihrem Händler oder der nächstgelegenen NCR-Geschäftsstelle.

CENTRONICS-ADAPTER
(K210-V001)

1. Setzen Sie diesen Adapter in eine der an der Rückseite Ihres NCR DECISION MATE V befindlichen Fassungen (2...6) ein.

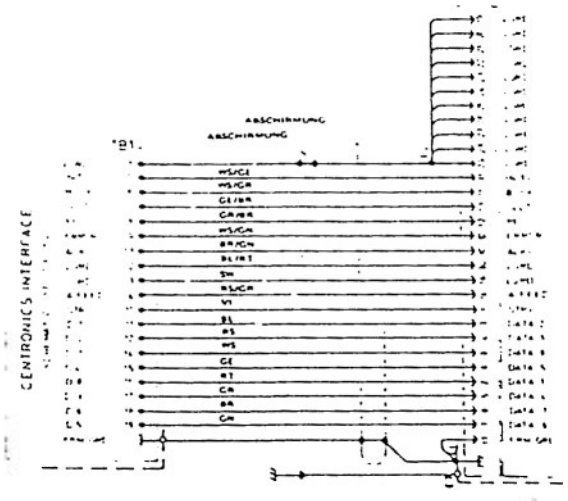


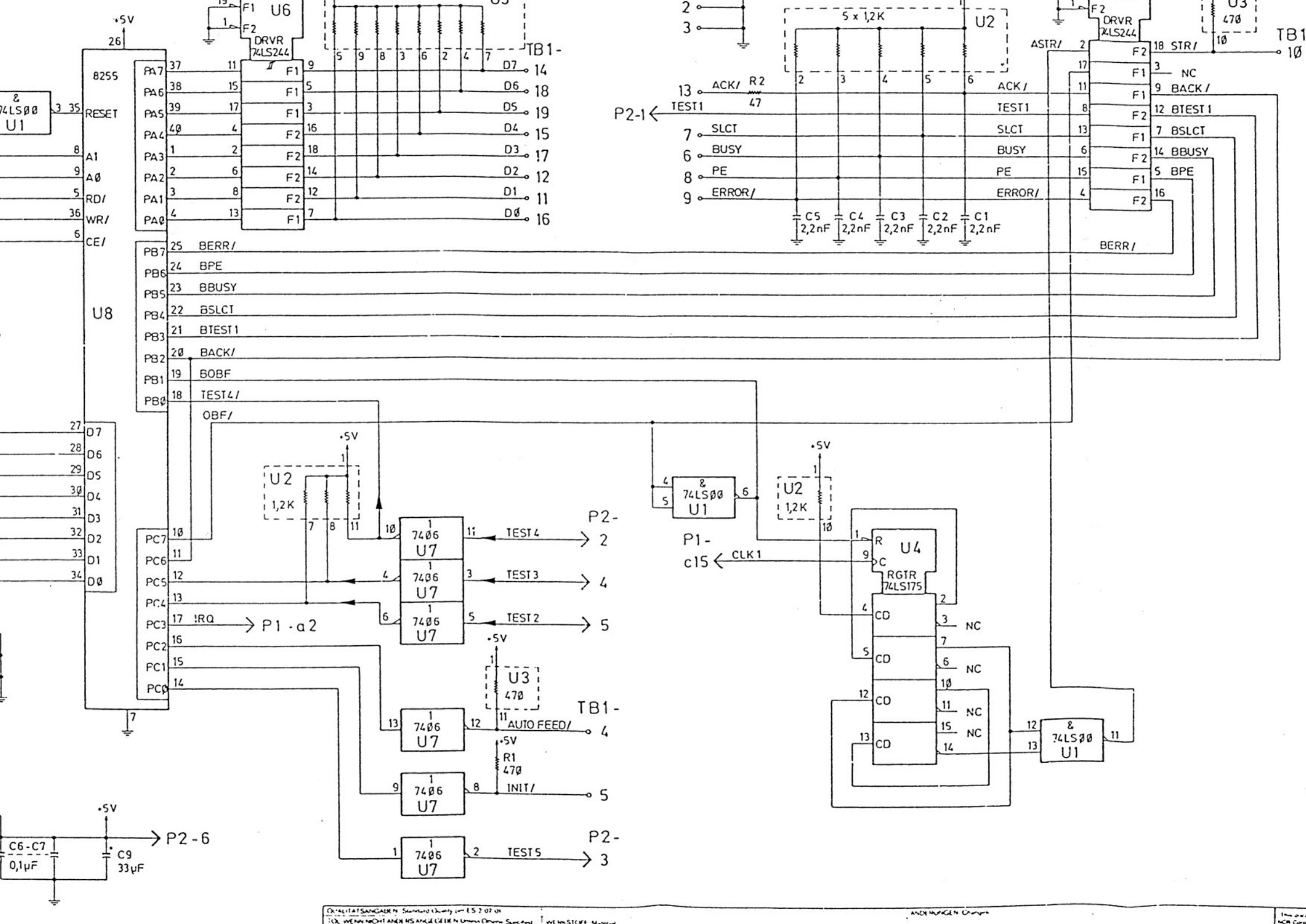
2. Schließen Sie den am Kabelende vorhandenen Stecker an einen Centronics-kompatiblen Drucker an.
3. Das zu Ihrem Drucker gehörende Handbuch beschreibt die Handhabung von Druckpapier und Farbband.
4. Schließen Sie den Drucker an die erforderliche Netzspannung an.

WICHTIG: Der gleichzeitige Betrieb dieses Adapters mit dem Adapter K212 oder K213 ist nicht möglich.

Umseitig finden Sie eine Übersicht der im Rahmen des Übertragungsprotokolls benutzten Leitungen.

STROBE/	Taktfrequenz zur Druckersynchronisierung. Mindesttaktzeit 1 µsec. Empfang erfolgt während Signal-"low".
ACKNLG/	Signal (2.5 bis 10 µsec) zur Datenempfangsbestätigung. Empfang neuer Daten vor Rücksetzung dieses Signals unmöglich.
BUSY	Datenannahme nicht möglich aus einem der folgenden Gründe : - während des Datenempfangs (je nach Drucker) - während des Druckvorgangs - im Off-Line-Zustand - beim Vorliegen eines Fehlerzustandes - während des Zeilenvorschubs
DATA 1-8	Parallele Datenübertragung ("High" = log 1; "Low" = log 0)
PE	"High" zeigt das Papierende an.
SLCT	Signal zeigt Wahlzustand des Druckers an (je nach Drucker)
AUTO FEED/	Automatischer Zeilenvorschub (nicht bei allen Druckern).
INIT/	Computersignal zur Druckerrückstellung (je nach Drucker).
ERROR/	Fehlerzustand des Druckers (z.B. Papierende, Off-Line, o.ä).
TEST 1-4	Eventuell vorhandene Prüfleitungen.




8255
U8
RESET
A1
A0
RD/
WR/
CE/
PA7
PA6
PA5
PA4
PA3
PA2
PA1
PA0
PB7
PB6
PB5
PB4
PB3
PB2
PB1
PB0
BERR/
BPE
BBUSY
BSLCT
BTEST1
BACK/
BOBF
TEST4/
OBF/
D7
D6
D5
D4
D3
D2
D1
D0
PC7
PC6
PC5
PC4
PC3
PC2
PC1
PC0
IRQ
U6
DRVR
74LS244
TB1-
U2
5 x 1,2K
ACK/
TEST1
SLCT
BUSY
PE
ERROR/
R2
47
C5 2,2nF
C4 2,2nF
C3 2,2nF
C2 2,2nF
C1 2,2nF
P2-1
U3
470
STR/
NC
BACK/
BTEST1
BSLCT
BBUSY
BPE
BERR/
ASTR/
U1
74LSØØ
U7
7406
TEST4
TEST3
TEST2
TEST5
AUTO FEED/
INIT/
R1
47Ø
P1 - a2
P2-2
P2-4
P2-5
P2-3
P2-6
P1-
c15
CLK1
U4
RGTR
74LS175
CD
R
C
NC
C6-C7
0,1µF
C9
33µF
+5V

NCR DECISION MATE V

UMSCHALTBARER RS-232C ADAPTER
(K801-V001)

Die beiliegenden Seiten zeigen Ihnen, wie Sie diese Leistungserweiterung an Ihren NCR DECISION MATE V anschließen können. Bitte ordnen Sie diese Beschreibung in Ihre Bedienungsanleitung für den NCR DECISION MATE V ein.

Doc:017-0033070

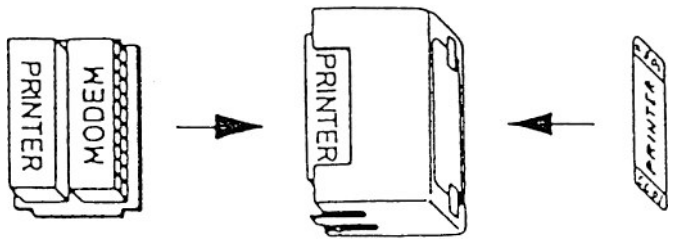
UMSCHALTBARER RS-232C ADAPTER
(K801-V001)

Dieser vielseitig verwendbare RS-232-C Adapter muß vor Gebrauch für den Anschluß eines Druckers, eines Plotters oder eines Modems vorbereitet werden. Gehen Sie bitte nach der folgenden Anleitung vor:

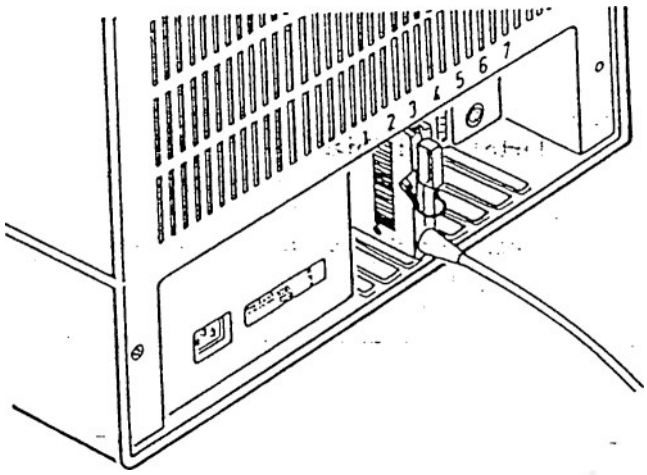
ANPASSUNG DER STECKERBELEGUNG

Ihrem Adapter sind ein kleines Gehäuse und zwei Steckplatten beigefügt. Die Steckplatten tragen beschriftete Kontaktleisten, die die Steckerbelegung für Drucker, Plotter oder Modem definieren. Auf der Steckplatte für Modemanschluß können Sie auf der leeren Seite nach eigenem Bedarf eine Steckerbelegung verdrahten.

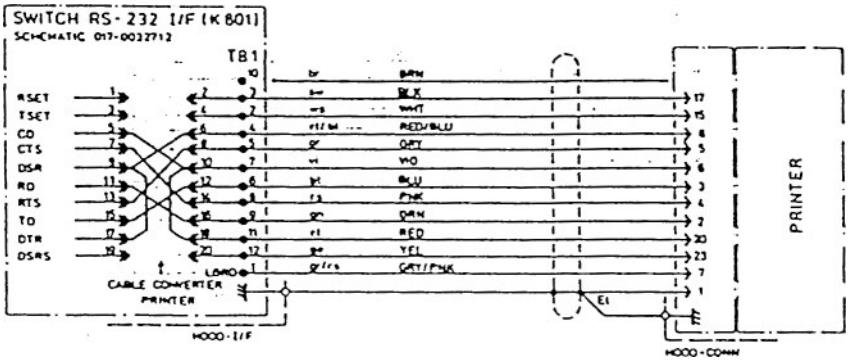
1. Wählen Sie die gewünschte Steckplatte für den Kabeladapter.
2. Stecken Sie die Steckplatte in das kleine Gehäuse. Achten Sie darauf, daß die Beschriftung der gewünschten Kontaktleiste im Ausschnitt des Gehäuses lesbar ist.
3. Kleben Sie den richtigen Identifikationsaufkleber auf die Rückseite des Gehäuses.
4. Stecken Sie das Gehäuse in die Öffnung an der Rückseite des RS-232-C Adapters.

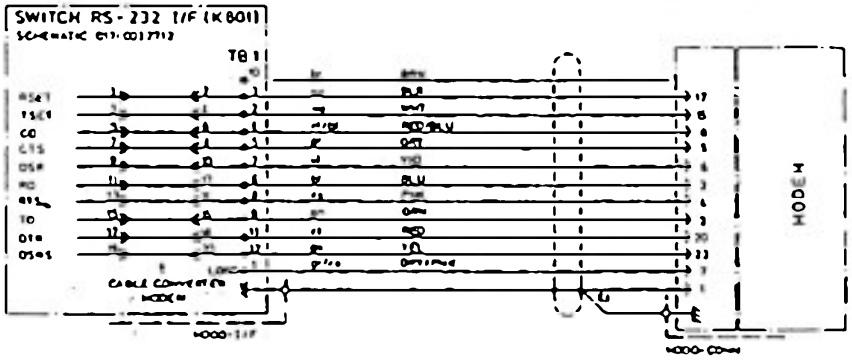


Nachdem Sie den Adapter Ihren Wünschen entsprechend vorbereitet haben, können Sie ihn nun in einen der Steckerplätze 2 bis 6 an der Rückseite Ihres NCR DECISION MATE V einstekken.



Schließen Sie den am Kabelende vorhandenen Stecker an das RS-232C-kompatible Gerät an.
Vergewissern Sie sich bitte, welche Anpassungen an Ihrem Peripheriegerät oder an Ihrer Software erforderlich sind. Entnehmen Sie die hierzu notwendige Information der jeweiligen Dokumentation.
Umseitig finden Sie eine Übersicht über die vom RS-232-C Adapter verwendeten Signale. Die darauf folgenden Seiten geben Ihnen Aufschluß über die verschiedenen Möglichkeiten der Steckerverdrahtung.

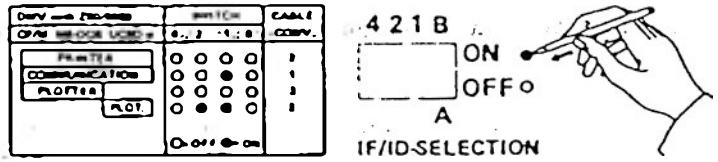

SWITCH RS-232 I/F (K801)
SCHEMATIC 017-0032712
TB1
RSET
TSET
CO
CTS
DSR
RD
RTS
TD
DTR
DSRS
CABLE CONVERTER
PRINTER
HOOD-I/F
HOOD-CONN
PRINTER


SWITCH RS-232 I/F (K801)
TB1
HOOD-I/F
HOOD-CONN
MODEM

TRANSMIT DATA (TD)	Übertragungsleitung des Computers zum Senden von Daten fester Länge (5...8 Bit). Nach der Übertragung jedes Zeichens ist das an dieser Leitung vorhandene Signal "low".
RECEIVE DATA (RD)	Über diese Leitung empfängt der Computer Daten. Zeichenlänge und Signalpegel wie TD.
CLEAR TO SEND (CTS)	Eingangssignal an den Computer: Die Übertragung von Daten erfolgt bei "high".
REQUEST TO SEND (RTS)	Sendebereitschaftssignal vom Computer. In der nicht-Modem-Betriebsart ist das Signal immer "high". In der Modem-(Halbduplex-)Betriebsart ist das Signal "high", wenn Daten zur Übertragung bereitstehen, "low" sonst.
DATA SET READY (DSR)	Eingangssignal an den Computer: "high" zeigt dem Computer an, daß Daten zum Empfang bereitstehen.
DATA TERMINAL READY (DTR)	Ausgangssignal vom Computer: Der Computer ist bereit, Daten zu empfangen.
DATA SIGNAL RATE SELECT (DSRS)	Einige Modems können mit zwei verschiedenen Geschwindigkeiten Daten übertragen: +12V = höhere Geschwindigkeit.
TRANSMITTER SIGNAL ELEMENT TIMING (TSET)	Externe Taktfrequenz für den Sender.
RECEIVER SIGNAL ELEMENT TIMING (RSET)	Externe Taktfrequenz für den Empfänger.
CARRIER DETECT (CD)	Eingangssignal an den Computer: "high" zeigt den ordnungsgemäßen Empfang des Trägersignals des fremden Modems an.

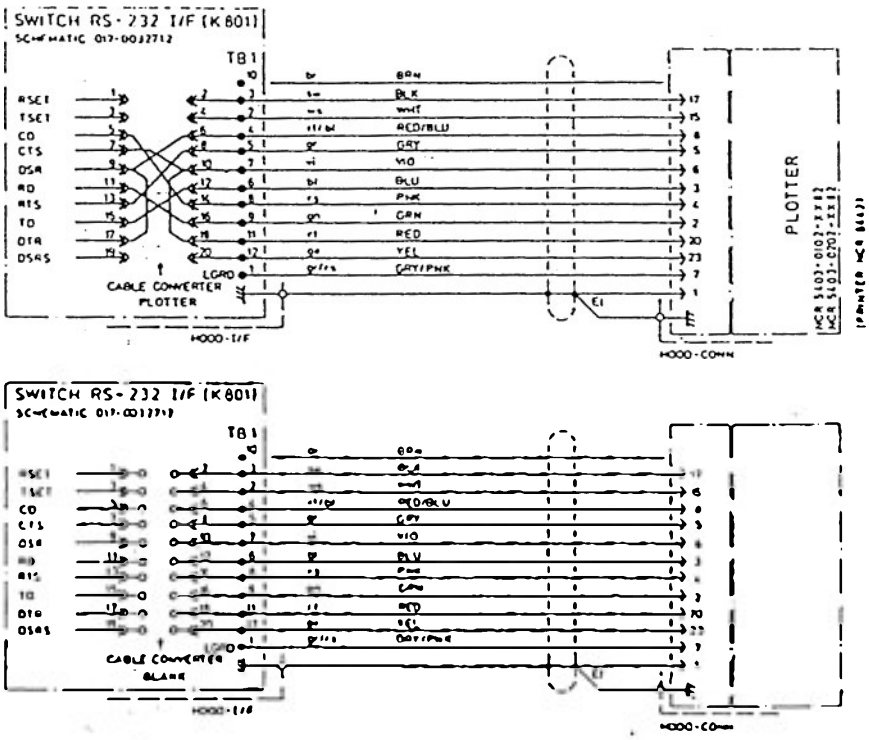
AUSWAHL DER IFSEL - NUMMERN

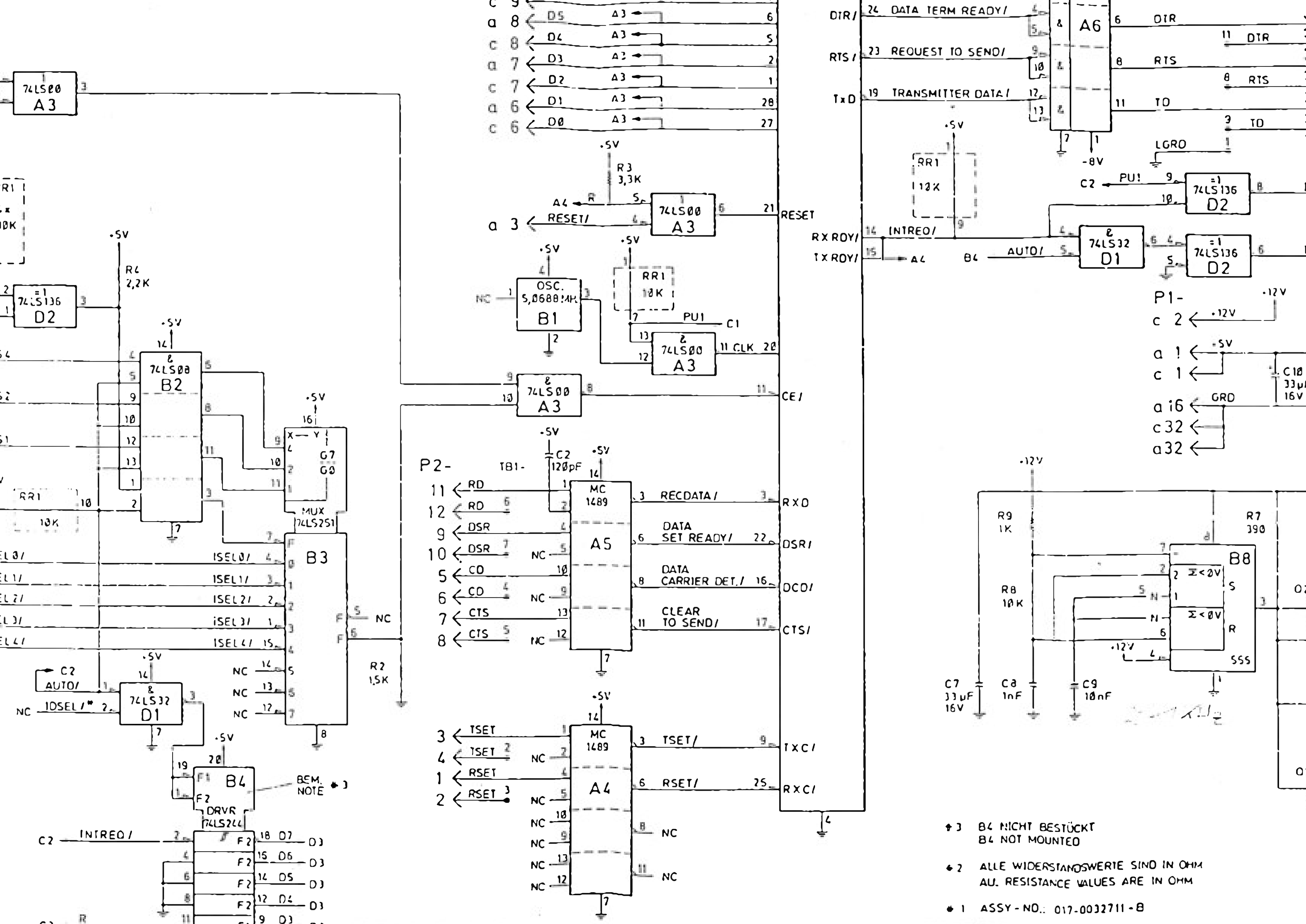
Am Adapter finden Sie vier kleine Schalter, mit deren Hilfe die IFSEL Nummern eingestellt werden. In der Regel werden die Schalter nach folgender Abbildung (Standardeinstellungen) eingestellt:



Falls Ihr System andere IFSEL-Werte verlangt, können Sie nach folgender Übersicht die Schalter einstellen:

IFSEL	SCHALTER 4 2 1 B	PORT-ADRESSE
0A	o o o o	60H - 67H
0B	o o o •	68H - 6FH
1A	o o • o	70H - 77H
1B	o o • •	78H - 7FH
2A	o • o o	30H - 37H
2B	o • o •	38H - 3FH
3A	o • • o	B0H - B7H
3B	o • • •	B8H - BFH
4A	• o o o	C0H - C7H
4B	• o o •	C8H - CFH


SWITCH RS-232 I/F (K801)
TB1
RSET
TSET
CO
CTS
DSR
RD
RTS
TD
DTR
DSRS
LGRD
CABLE CONVERTER
PLOTTER
PLOTTER
SWITCH RS-232 I/F (K801)
TB1
CABLE CONVERTER
BLANK



*3 B4 NICHT BESTÜCKT
B4 NOT MOUNTED

*2 ALLE WIDERSTANDSWERTE SIND IN OHM
ALL RESISTANCE VALUES ARE IN OHM

*1 ASSY-NO.: 017-0032711-8

NCR DECISION MATE V

MAUS-ADAPTER
(K806-V001)

Die beiliegenden Seiten zeigen Ihnen, wie Sie diese Leistungserweiterung an Ihren NCR DECISION MATE V anschließen können. Bitte ordnen Sie diese Beschreibung in Ihre Bedienungsanleitung für den NCR DECISION MATE V ein.

Doc: 017-0033170

MAUS-ADAPTER
(K806-V001)

INHALT

INBETRIEBNAHME

EINLEITUNG.....1
IFSEL- UND MAUS-AUSWAHL.....2

SOFTWARE

EINLEITUNG.....6
BENUTZUNG DER PORT-ADRESSEN.....7
FUNKTIONSPRÜFUNG.....7
DAS STATUS-BYTE.....8
BEFEHLE UND IHRE PARAMETER.....8
INTERRUPT-BEHANDLUNG.....13

INBETRIEBNAHME

EINLEITUNG

Mit Hilfe einer "Maus" lassen sich Handbewegungen auf einem Schreibtisch oder einer ähnlichen Fläche in Bildschirmgrafik umsetzen. Sie können folgende Mäuse in Verbindung mit dem Maus-Adapter benutzen:

- Hawley Mouse MARK II
- Alps Encoder-Mouse
- Logitech LM-P-5

Die Verständigung zwischen einem Adapter und Ihrem NCR DECISION MATE V erfolgt über einen von insgesamt 10 Datenübertragungskanälen. Jeder Datenübertragungskanal (oder IFSEL: engl. InterFace SELect) verfügt über 8 Portadressen. Der Maus-Adapter kann grundsätzlich jede beliebige der 10 IFSEL-Nummern benutzen. Die für diesen Adapter vorhandene p-System-Software betrachtet die IFSEL-Nummer 2A (Portadressen 30H...37H) als Standardwert. Diese IFSEL-Nummer ist bereits bei der Lieferung des Maus-Adapters eingestellt. Die zur IFSEL-Auswahl gehörenden Schalter befinden sich innerhalb des Gehäuses.

Ebenfalls innerhalb des Gehäuses sind Schalter für die Auswahl der von Ihnen benutzten Maus vorhanden. Diese Schalter sind für den Betrieb der folgenden Mäuse werksseitig eingestellt:

- Hawley Mouse Mark II
- Alps Encoder-Mouse

Vorausgesetzt, daß Sie eine dieser Mäuse mit der Standardeinstellung der IFSEL-Nummer betreiben wollen, ist ein

Öffnen des Gehäuses des Maus-Adapters nicht erforderlich. In diesem Fall können Sie ohne weiteres den Adapter in eine der Steckfassungen 2...6 an der Rückseite Ihres NCR DECISION MATE V einsetzen (siehe Abb. 1.1). Anschließend sollten Sie den zweiten Teil dieser Anleitung ("Software") bezüglich der für den Betrieb einer Maus benötigten Software lesen.

Wenn Sie eine Depraz- oder eine Logitech-Maus benutzen, oder im Fall, daß Sie von einer vom Standardwert (2A) abweichenden IFSEL-Nummer Gebrauch machen möchten, sollten Sie gemäß der im folgenden Abschnitt enthaltenen Beschreibung verfahren.

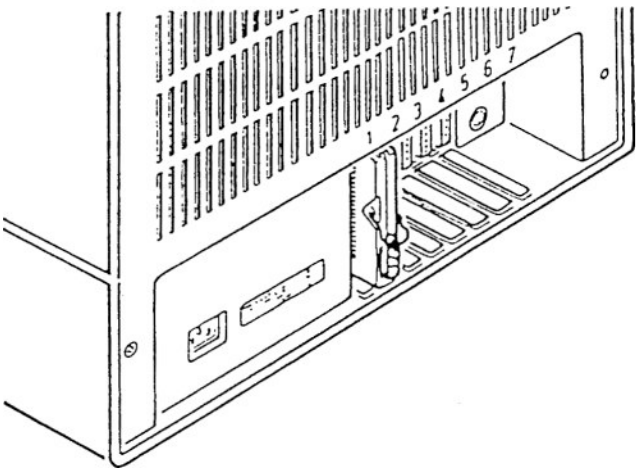


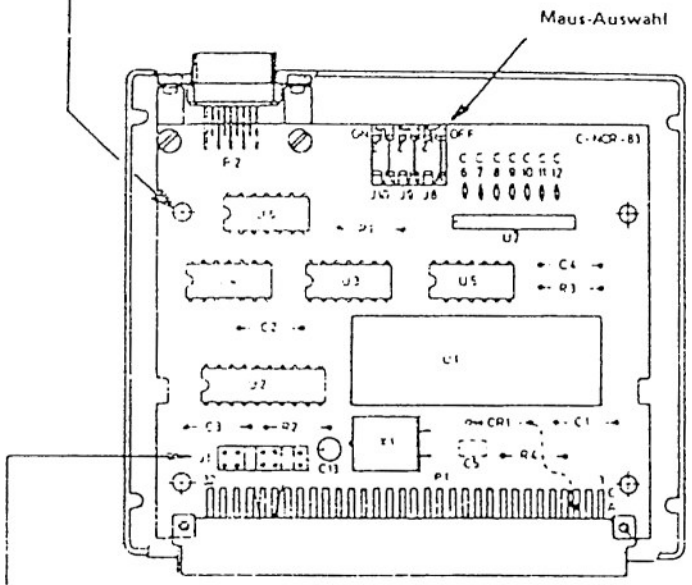
Abbildung 1.1: Der Maus-Adapter

IFSEL- UND MAUS-AUSWAHL

Die nachstehend beschriebenen Arbeitsschritte sind nur dann erforderlich, wenn Sie die IFSEL-Nummer ändern oder eine der folgenden Mäuse benutzen möchten:

WICHTIG:

Die Erdverbindungen zwischen der mit den Leiterbahnen belegten Fläche der Platine und dem Gehäuse.



[illegible]	J1	J6
[illegible]	J1	J7
[illegible]	J2	J6
[illegible]	J2	J7
[illegible]	J3	J6
[illegible]	J3	J7
[illegible]	J4	J6
[illegible]	J4	J7
[illegible]	J5	J6
[illegible]	J5	J7

[illegible] und Maus-Auswahl

- Logitech LM-P-5

1. Entfernen Sie den am Gehäuse befestigten Drahtgriff und die vier Schrauben. Das Gehäuse sollten Sie noch nicht öffnen.
2. Halten Sie die zwei Teile des Gehäuses zusammen, und legen Sie den Adapter auf einen Schreibtisch o.ä., so daß die Schraublöcher nach unten zeigen. Entfernen Sie den oberen Teil des Gehäuses.
3. Auf der Platine des Maus-Adapters sind drei Schaltbrücken mit J10, J9, J8 gekennzeichnet (siehe auch Abbildung 1.2). Die Standardeinstellung 2A hat zur Folge, daß jede dieser drei Schaltbrücken in der OFF-Stellung ist. Wenn Sie eine Depraz- oder eine Logitech-Maus benutzen, müssen Sie alle drei Schalter in die ON-Stellung bringen. Hierzu ist jeder Draht aus seinem Haken zu lösen und in den jeweils unmittelbar links befindlichen Haken einzusetzen.
4. Die Aufschrift J1 auf der Platine (siehe auch Abbildung 1.2) zeigt Ihnen, wo die Schalter für die IFSEL-Auswahl zu finden sind. Die Einstellung einer IFSEL-Nummer erfordert eine Neubelegung der zwei Steckverbindungen auf den paarweise angeordneten Stiften (siehe Abb. 1.2). Die zwei am weitesten links befindlichen Stifte bilden die Schaltbrücke J1, die zwei am weitesten rechts befindlichen Stifte bilden die Schaltbrücke J7. Die zwei Steckverbindungen befinden sich auf den Schaltbrücken J3 bzw. J6, solange die IFSEL-Nummer 2A eingestellt ist.
5. Schrauben Sie das Gehäuse anhand der vier Schrauben zusammen; bringen Sie den Drahtgriff wieder an.

Abbildung 1.3 enthält eine Aufstellung der IFSEL-Nummern mit den jeweils verfügbaren Portadressen.

IFSEL-NR	PORT-ADRESSEN
0A	60-67
0B	68-6F
1A	70-77
1B	78-7F
2A	30-37
2B	38-3F
3A	B0-B7
3B	B8-BF
4A	C0-C7
4B	C8-CF

Abbildung 1.3: IFSEL/Portadressen

SOFTWARE

EINLEITUNG

Der Maus-Adapter beinhaltet einen eigenen Mikroprozessor sowie eigene Firmware. Dies hat zur Folge, daß die zusätzliche Belastung des Prozessors des NCR DECISION MATE V durch den Betrieb des Maus-Adapters sehr gering ist. Der Maus-Adapter ist in der Lage, bis zu 4000 Positionen der angeschlossenen Maus in einer Sekunde an den Computer zu senden. Ihr Programm kann einen Bereich definieren, innerhalb dessen die Maus sich bewegen muß. (Sämtliche Positionen und Bereiche werden als X/Y-Koordinaten angegeben.) Der Maus-Adapter kann nicht nur die Position der Maus sondern auch den Zustand von bis zu drei an der Maus befindlichen Schaltern vermitteln. Ein Status-Byte gibt Aufschluß darüber, ob die internen ROM- und RAM-Speicher des Adapters einwandfrei funktionieren, ob der Adapter bereit ist, Befehle oder Daten zu senden oder zu empfangen, und ob das Interrupt-Signal gesetzt wurde.

Der Maus-Adapter bietet zwei Verfahrensweisen, auf die Ihr Programm die Bewegung der Maus und die Betätigung eines Schalters an der Maus verfolgen kann: Es kann in bestimmten Zeitabständen diese Daten von dem Maus-Adapter lesen. Die zweite Methode besteht darin, auf ein Interruptsignal zu warten, und erst dann die Positions- und Schalterdaten zu lesen. Sofern Sie von der Interrupt-Leitung des Computers Gebrauch machen wollen, sind in der Regel zusätzliche Interrupt-Controller-Schaltungen erforderlich. Wenn Sie aber Ihren NCR DECISION MATE V als 8-Bit-System benutzen, können Sie aufgrund bestimmter Eigenschaften des Z80A-Mikroprozessors Interrupt-Behandlung auch ohne solche Schaltungen verwirklichen (siehe hierzu den Abschnitt "Interrupt-Behandlung").

Die von NCR erhältliche p-System-Software beinhaltet eine Anzahl von BASIC-, FORTRAN- und Pascal-Programme für den Betrieb des Maus-Adapters. Eine Beschreibung dieser Programme in englischer Sprache liegt dieser Anleitung bei.

Sie können den Maus-Adapter selbstverständlich auch mit den Betriebssystemen CP/M und MS-DOS benutzen. Dieser Teil ("Software") beschreibt die Anweisungen für die Programmierung des Maus-Adapters. Die besten Ergebnisse lassen sich mit Programmen in Assemblersprache erzielen.

BENUTZUNG DER PORT-ADRESSEN

Der Maus-Adapter verwendet die erste und die zweite der zur gewählten IFSEL-Nummer gehörenden Port-Adressen. Vorausgesetzt, daß Sie die Standard-IFSEL-Nummer nicht geändert haben, sind die benutzten Port-Adressen 30H (Port 1) und 31H (Port 2). Die Ein- und Ausgabe über die Port-Adressen erfolgt gemäß folgender Einteilung:

IN Port 1 - Das Programm liest Daten vom Maus-Adapter.
OUT Port 1 - Der Maus-Adapter empfängt Daten von Ihrem Programm.
IN Port 2 - Das Program liest das Status-Byte des Maus-Adapters.
OUT Port 2 - Das Programm sendet Befehle an den Maus-Adapter.

FUNKTIONSPRÜFUNG

Wenn Sie Ihren NCR DECISION MATE V bei eingesetztem Maus-Adapter einschalten, setzt das Reset-Signal des Computers auch den Maus-Adapter in seinen Anfangszustand zurück. Der Prozessor des Maus-Adapters liest dann Anweisungen im eigenen ROM: Zunächst wird geprüft, ob ROM und RAM des Maus-Adapters einwandfrei funktionieren. Dann initialisiert der Maus-Adapter seine I/O-Ports.

Sollte ein Fehlerzustand im ROM oder im RAM vorliegen, wird ein entsprechendes Bit im Status-Byte gesetzt. Der folgende Abschnitt erläutert die Bedeutung der einzelnen Bits des Status-Bytes und zeigt Ihnen, wie Ihr Programm dieses Byte lesen kann.

DAS STATUS-BYTE

Ihr Programm kann das Status-Byte mit einer IN-Anweisung an Port 2 (31H bei IFSEL 2A) lesen. Abbildung 2.1 zeigt die Bedeutung der einzelnen Bits dieses Bytes.

Bit:	7	6	5	4	3	2	1	0
	X	INT	RAM	ROM	X	X	IBF	OBF

Abbildung 2.1: Das Status-Byte

X Dieses Bit wird nicht benutzt.

INT Sobald der Maus-Adapter ein Interruptsignal ausgibt, wird dieses Bit gesetzt (logisch 1). Es bleibt in diesem Zustand, während die Interruptleitung aktiviert ist (active "low").

ROM Ein Fehlerzustand im ROM des Maus-Adapters führt dazu, daß dieses Bit gesetzt wird. Sonst ist es zurückgesetzt.

RAM Ein Fehlerzustand im RAM des Maus-Adapters führt dazu, daß dieses Bit gesetzt wird. Sonst ist es zurückgesetzt.

IBF Input Buffer Full: Solange dieses Bit gesetzt ist, kann der Maus-Adapter weder Befehle noch Daten von Ihrem Programm empfangen. Vor der Ausgabe von Befehlen oder Daten an den Maus-Adapter sollte ein Programm den Zustand dieses Bits abfragen und die Ausgabe erst dann ausführen, wenn das Bit zurückgesetzt ist.

OBF Sobald dieses Bit gesetzt wird, kann Ihr Programm ein Datenbyte über Port 1 des Maus-Adapters lesen. Das Programm sollte keinen Lesevorgang versuchen, während dieses Bit zurückgesetzt ist.

BEFEHLE UND IHRE PARAMETER

Dieser Abschnitt befaßt sich mit den vom Maus-Adapter anerkannten Befehlen. Einige dieser Befehle leiten die Übertragung einer Reihe von Parametern (Daten) ein.

WICHTIG: Nachdem ein zum Senden oder zum Empfangen von Parametern gehörender Befehl gesendet worden ist, müssen alle zu diesem Befehl gehörenden Parameter ausgegeben bzw. gelesen werden. Ihr Programm sollte die Übertragung der Parameterliste nicht vorzeitig abbrechen, indem es z.B. einen neuen Befehl an den Maus-Adapter sendet.

Alle Befehle an den Maus-Adapter sind über Port 2 auszugeben; Parameter werden über Port 1 sowohl gesendet als auch empfangen.

Die folgenden Werte werden beim Einschalten Ihres NCR DECISION MATE V mit Maus-Adapter automatisch eingestellt. Sie verlieren ihre Gültigkeit, erst wenn sie von Ihrem Programm bzw. einer Bewegung der Maus aufgehoben werden.

XMAX: Der höchste Wert, den die Maus in der X- (horizontalen) Richtung erreichen kann: 640
YMAX: Der höchste Wert, den die Maus in der Y- (vertikalen) Richtung erreichen kann: 400
XMIN: Der niedrigste Wert, den die Maus in der X-Richtung erreichen kann: 0
YMIN: Der niedrigste Wert, den die Maus in der Y-Richtung erreichen kann: 0
X- und Y-Koordinaten der gegenwärtigen Maus-Position: 0
Alle Interruptsignale sind außer Kraft.

Nachstehend finden Sie eine Beschreibung jedes Befehls und der etwa dazugehörigen Parameter. Für jeden Befehl ist der hexadezimale Wert angegeben, der über Port 2 an den Maus-Adapter zu senden ist. Negative (Minus-) Werte werden immer als Zweierkomplement dargestellt.

00 Die absolute Position der Maus wird gelesen
Die Reihenfolge der zu lesenden Parameter:

X-Koordinate, niederwertiges Byte
X-Koordinate, höherwertiges Byte
Y-Koordinate, niederwertiges Byte
Y-Koordinate, höherwertiges Byte
Schalter-Status-Byte (s. Ende dieses Abschnitts)

Die absolute Position bezieht sich auf die Änderung der Position der Maus (ggf. innerhalb des definierten Bewegungsbereichs) seit dem letzten Setzen der Anfangsposition. (Diese Anfangsposition wird beim Einschalten auf X=0, Y=0 gesetzt. Sie kann ebenfalls anhand des Befehls 04 gesetzt werden.)

01 **Das aufgrund einer Bewegung der Maus erzeugte Interruptsignal wird anerkannt**
Keine Parameter.
Dieser Befehl hat gleichzeitig zur Folge, daß das Interruptsignal für die Mausschalter außer Kraft gesetzt wird.

02 **Interruptsignale werden nicht erkannt**
Keine Parameter.

03 **Das Signal an der Interruptleitung und das INT-Bit im Status-Byte werden zurückgesetzt**
Keine Parameter.

04 **Die logische Position der Maus wird gesetzt**
Die Reihenfolge der Parameter, die an den Maus-Adapter zu senden sind:

X-Koordinate, niederwertiges Byte
X-Koordinate, höherwertiges Byte
Y-Koordinate, niederwertiges Byte
Y-Koordinate, höherwertiges Byte

Wenn diese Position sich außerhalb des für die Bewegung der Maus definierten Bereichs liegt (s. Befehle 05 und 0A), findet die Definition dieses Bereichs keine Anwendung.

05 **Der Höchstwert, den die Maus in der X- bzw. Y-Richtung erreichen kann, wird gesetzt**
Die Reihenfolge der Parameter, die an den Maus-Adapter zu senden sind:

XMAX, niederwertiges Byte
XMAX, höherwertiges Byte

YMAX, niederwertiges Byte
YMAX, höherwertiges Byte

Das Verhältnis der mit der Maus zurückgelegten Entfernung zum Wert, der für diese Entfernung vom Maus-Adapter gesendet wird, ist für die verschiedenen Mäuse unterschiedlich. Die beim Einschalten automatisch eingestellten Werte (XMAX = 640, YMAX = 400) entsprechen einer angenehmen Reichweite bei der Benutzung einer Maus auf einem Schreibtisch.

06 **Das aufgrund der Betätigung eines Maus-Schalters erzeugte Interruptsignal wird anerkannt**
Keine Parameter.
Dieser Befehl hat gleichzeitig zur Folge, daß das bei einer Bewegung der Maus erzeugte Interruptsignal außer Kraft gesetzt wird.

07 **Die Interruptsignale sowohl für die Betätigung eines Schalters als auch für eine Bewegung der Maus werden erkannt**
Keine Parameter.

08 **Der ROM des Maus-Adapters wird überprüft**
Keine Parameter.
Diese Überprüfung erfolgt auch beim Einschalten.

09 **Die relative Position der Maus wird gelesen**
Parameter: Siehe 00.
Die relative Position der Maus bezieht sich auf den Positionsunterschied zwischen der gegenwärtigen Position und der zuletzt gelesenen Position.

0A **Der Niedrigstwert, den die Maus in der X- bzw. Y-Richtung erreichen kann**
Die Reihenfolge der Parameter, die an den Maus-Adapter zu senden sind:

XMIN, niederwertiges Byte
XMIN, höherwertiges Byte
YMIN, niederwertiges Byte
YMIN, höherwertiges Byte

WICHTIG: Absolute und relative Position der Maus werden in denselben Registern des Maus-Adapters gespeichert. Infolgedessen ist bei der Benutzung beider Positionsformate in einem Programm Vorsicht geboten. Im Fall, daß sowohl die absolute als auch die relative Position der Maus in Ihrem Programm benötigt wird, empfiehlt es sich, die absolute Position als X/Y-Koordinaten in Programmvariablen festzuhalten. Sie können mit Hilfe des Befehls 04 diese Werte in die Positionsregister wieder zurückschreiben.

Der Maus-Adapter berücksichtigt die am Gehäuse der Maus vorhandenen Schalter. Der am weitesten links befindliche Schalter wird als S1, der am weitesten rechts befindiche Schalter als S2 bezeichnet. Ein etwa in der Mitte vorhandener Schalter wird als S3 bezeichnet. Jedem Schalter ist ein Bit im Schalter-Status-Byte (Befehl 00) zugeordnet:

S1 - Bit 0
S2 - Bit 1
S3 - Bit 2

(Die verbleibenden fünf Bits dieses Bytes werden nicht benutzt.) Das Bit ist gesetzt, während der entsprechende Schalter sich in gedrückter Stellung befindet.

INTERRUPT-BEHANDLUNG

Dieser Abschnitt befaßt sich mit den Interruptsignalen, die von dem Maus-Adapter ausgegeben werden. Vorausgesetzt, daß die Interruptsignale nicht außer Kraft sind, wird ein solches Signal bei jeder Bewegung der Maus bzw. bei jedem Öffnen und jedem Schließen der Kontakte eines Schalters.

WICHTIG: Die Benutzung der Interrupt-Leistungen des Maus-Adapters (über die Abfrage des INT-Bits des Status-Bytes hinaus) setzt Erfahrung im Umgang mit Halbleiter-Bauteilen voraus. Wichtige Informationen über die Hardware Ihres NCR DECISION MATE V finden Sie im Band "Hardware" des von NCR herausgegebenen System Technical Manual.

Wenn Sie Ihren NCR DECISION MATE V als 16-Bit-System benutzen, sind für die Behandlung von Interruptsignalen zusätzliche Interrupt-Controller unerläßlich.

Bestimmte Eigenschaften des Z80A-Mikroprozessors ermöglichen in einem 8-Bit-System eine Interrupt-Behandlung ohne Interrupt-Controller: Beim Interrupt-Modus 1 (IM 1 / Opcode: ED 56) wird beim Erzeugen eines Signals auf der Interruptleitung die Programmsteuerung an die hexadezimale Adresse 38 übergeben. Das Betriebssystem CP/M-80 enthält an dieser Adresse einen Sprungbefehl (JP) an das zu Testzwecken eingesetzte DDT-Programm. Vorausgesetzt, daß DDT für Ihre Anwendung nicht benötigt wird, können Sie einen Sprungbefehl an dieser Adresse ablegen. Dieser könnte dann auf Ihr eigenes Interrupt-Behandlungsprogramm verweisen.

CHAPTER 8

MOUSE

Introduction

The following is a description of the procedures you can use with the p-System™ to work with your mouse.

The software you need for using the Mouse interface and working with the Mouse in the p-System™ is contained in the MOUSE.CODE file. If you have a Runtime System MOUSE.CODE is incorporated in your SYSTEM.LIBRARY. If you have a Plus System, you have to insert this code in your SYSTEM.LIBRARY, in any of slots 0 through 15. Chapter 6, utility 'LIBRARY' in the 'UCSD p-System, Programming' Manual describes how this is done.

The interface part of the Mouse unit affords access to the following procedures:

GMOUS:	gives the status of the Mouse switches and Mouse position. This procedure supplies the absolute position (x and y coordinates) of the Mouse and the status of the switch(es) (up to 3). Use all three switch variables, even if your Mouse has only one or two switches.
X,Y:	absolute x and y coordinates
SW1,SW2,SW3:	status of the Mouse switches (1 = pressed, 0 = not pressed or not existent)
SETPOS:	sets the logical position of the Mouse.
X,Y:	New x and y coordinates for the Mouse position.

Note: Use this option to set a new position for the Mouse. If you require no direct correlation between the physical position of the Mouse and its program position, you can reset

(This is the description of the Mouse unit of the p-System™. Please insert it in the manual UCSD p-System, Programming, chapter 8)

p-System™ is a trademark of Softech Microsystems, Inc., and is used pursued to a license granted by Softech Microsystems, Inc.

the logical Mouse position by means of SETPOS.

SETMAX: sets maximum values for the Mouse position. Sets the maximum x and y values the Mouse can attain (default: x=640, y=400).

XMAX,YMAX: are maximum values for the x and y coordinates

For SETPOS note that the initial Mouse position is x=0, y=0. The Mouse can move within the area delimitated by x=0, y=0 and the values set in SETMAX.

Pascal Procedures

To use the Mouse procedures in a Pascal program, declare
USES MOUSE;

The first lines of the respective procedures have the following forms:

GMOUS

```
PROCEDURE GMOUS
   (VAR SW3, SW2, SW1, Y, X: INTEGER)
```

SETPOS

```
PROCEDURE SETPOS (X,Y: INTEGER)
```

SETMAX

PROCEDURE SETMAX (XMAX,YMAX: INTEGER)

Example

Here is an example for the use of the Mouse procedures in Pascal.

In the first part of the main program, the maximum Mouse position is set by means of the X and Y coordinates; and the present Mouse position is set to zero. In the second part of the main program, the variables are collected and displayed on the screen. SW1, SW2, and SW3 designate the switches on the Mouse (here the HAWLEY MARK II Mouse), which can be on or off. X is the Mouse position on the X coordinate, Y the Mouse position on the Y coordinate.

```
{ EXAMPLE MOUSE PROGRAM PASCAL }

 PROGRAM MOUSETP;

   USES MOUSE;

   VAR SW1,SW2,SW3,X,Y,X0,Y0,XM,YM:INTEGER; C:CHAR;

   PROCEDURE FIN(VAR C:CHAR);
{ PROMPT FOR FINISH }
     BEGIN
       GOTOXY(0,18);
       WRITE('    DO YOU WANT TO FINISH?   Y/N ');
       READLN(C);
     END;

   BEGIN
{ CLEAR SCREEN, CURSOR INVISIBLE }
     WRITELN(CHR(27),'@0',CHR(27),CHR(69));

     WRITELN('    ******   M O U S E  EXAMPLE  P A S C A L   ******');
     GOTOXY(0,16);
     WRITELN('    TO STOP PRESS SWITCH 1 ');
```

```
{ SET MAXIMUM }
     XM:=10000;
     YM:=10000;
     SETMAX(XM,YM);

{ SET START POSITION }
     X0:=0;
     Y0:=0;
     SETPOS(X0,Y0);

     C:='N';

{ LOOP MOUSE ACTIONS }
     REPEAT
       GMOUS(SW3,SW2,SW1,Y,X);
       GOTOXY(0,4);
       WRITELN('     X=',X:6,'      Y=',Y:6);
       GOTOXY(0,6);
       WRITELN('     SWITCH 1:',SW1,'  SWITCH 3:',SW3,'  SWITCH 2:',SW2);
       GOTOXY(0,10);

{ SET OR CLEAR PROMPT FOR FINISH DEPENDING ON SW1 }


       IF SW1=1 THEN FIN(C)
                ELSE WRITE(CHR(27),'L');

     UNTIL C in ['y','Y'];

{ CURSOR VISIBLE }
     WRITE(CHR(27),'@1')
   END.
```

Fortran Procedures
To use the Mouse procedures in a Fortran program, declare

```
$USES MOUSE
```

The subroutine and parameter definition can be found below:

GMOUS

```
SUBROUTINE GMOUS(SW3,SW2,SW1,Y,X)
   INTEGER SW1, SW2, SW3, Y, X
```

SETPOS

```
SUBROUTINE SETPOS (X,Y)
   INTEGER X,Y
```

SETMAX

```
SUBROUTINE SETMAX (XMAX,YMAX)
   INTEGER XMAX,YMAX
```

Example

Here is an example for the use of the Mouse procedures in FORTRAN.

In the line starting with 1, the maximum Mouse position is set by means of the X and Y coordinates. In the line starting with 2, the present Mouse position is set to zero. In the line starting with 10, SW1, SW2, and SW3 designate the switches on the Mouse (here the HAWLEY MARK II Mouse), which can be on or off. X is the Mouse position

on the X coordinate; y the Mouse position on the Y coordinate.

Then the program collects the values and displays them on the screen.

```
C
C       MOUSE EXAMPLE PROGRAM FOR FORTRAN
C
$USES MOUSE

        PROGRAM MOUSTF

        INTEGER SW1,SW2,SW3,X,Y,X0,Y0,XM,YM
        CHARACTER C

C       CLEAR SCREEN, CURSOR INVISIBLE
        WRITE(*,'(AAAA)') CHAR(27),'00',CHAR(27),'E'

        WRITE(*,200)
200     FORMAT(4X,'******   M O U S E  EXAMPLE  F O R T R A N   *******')
C       SCREEN POSITION LINE 16 COLUMN 0
        WRITE(*,'(AA\)') CHAR(27),'Y0 '
        WRITE(*,'(A)') '     TO STOP PRESS SWITCH 1 '

C       SET MOUSE MAX VAL
        XM=10000
        YM=10000
1       CALL SETMAX(XM,YM)

C       SET MOUSE POSITION
        X=0
        Y=0
2       CALL SETPOS(X,Y)

        C='N'
```

```
C       LOOP MOUSE ACTIONS
        DO 1000 I=0,0,0
10      CALL GMOUS(SW3,SW2,SW1,Y,X)
        WRITE(*,'(AA\)') CHAR(27),'Y5 '
C       SCREEN POSITION LINE 4, COLUMN 0
        WRITE(*,300) X,Y
300     FORMAT ('     X=',I6,'      Y=',I6//)
        WRITE(*,100) SW1,SW3,SW2
100     FORMAT('    SWITCH1:',I1,'   SWITCH3:',I1,'   SWITCH2:',I1)
C       SCREEN POSITION LINE 18, COLUMN 0
        WRITE(*,'(AA\)') CHAR(27),'Y2 '

C       SET OR RESET PROMPT FOR FINISH DEPENDING ON SW1
        IF (SW1 .EQ. 1) THEN
        CALL FIN(C)
        ELSE
        WRITE(*,'(AA\)') CHAR(27),'L'
        ENDIF

        IF ((C .EQ. 'Y') .OR. (C .EQ. 'y')) THEN
        I=1
        ENDIF
1000    CONTINUE

C       CURSOR VISIBLE
        WRITE(*,'(AA)') CHAR(27),'@1'
        END

        SUBROUTINE FIN (C)
C       PROMPT FOR FINISH
        CHARACTER C
        WRITE(*,'(A\)') '    DO YOU WANT TO FINISH?   Y/N '
        READ(*,'(A1)') C
        RETURN
        END
```

BASIC Interface

To use the Mouse procedures in a BASIC program, declare

```
USES MOUSE
```

The subroutine and parameter definition can be found below:

GMOUS

```
SUB GMOUS(SW3,SW2,SW1,Y,X)
   INTEGER SW1, SW2, SW3, Y, X
```

SETPOS

```
SUB SETPOS (X,Y)
   INTEGER X,Y
```

SETMAX

```
SUB SETMAX (XMAX,YMAX)
   INTEGER XMAX,YMAX
```

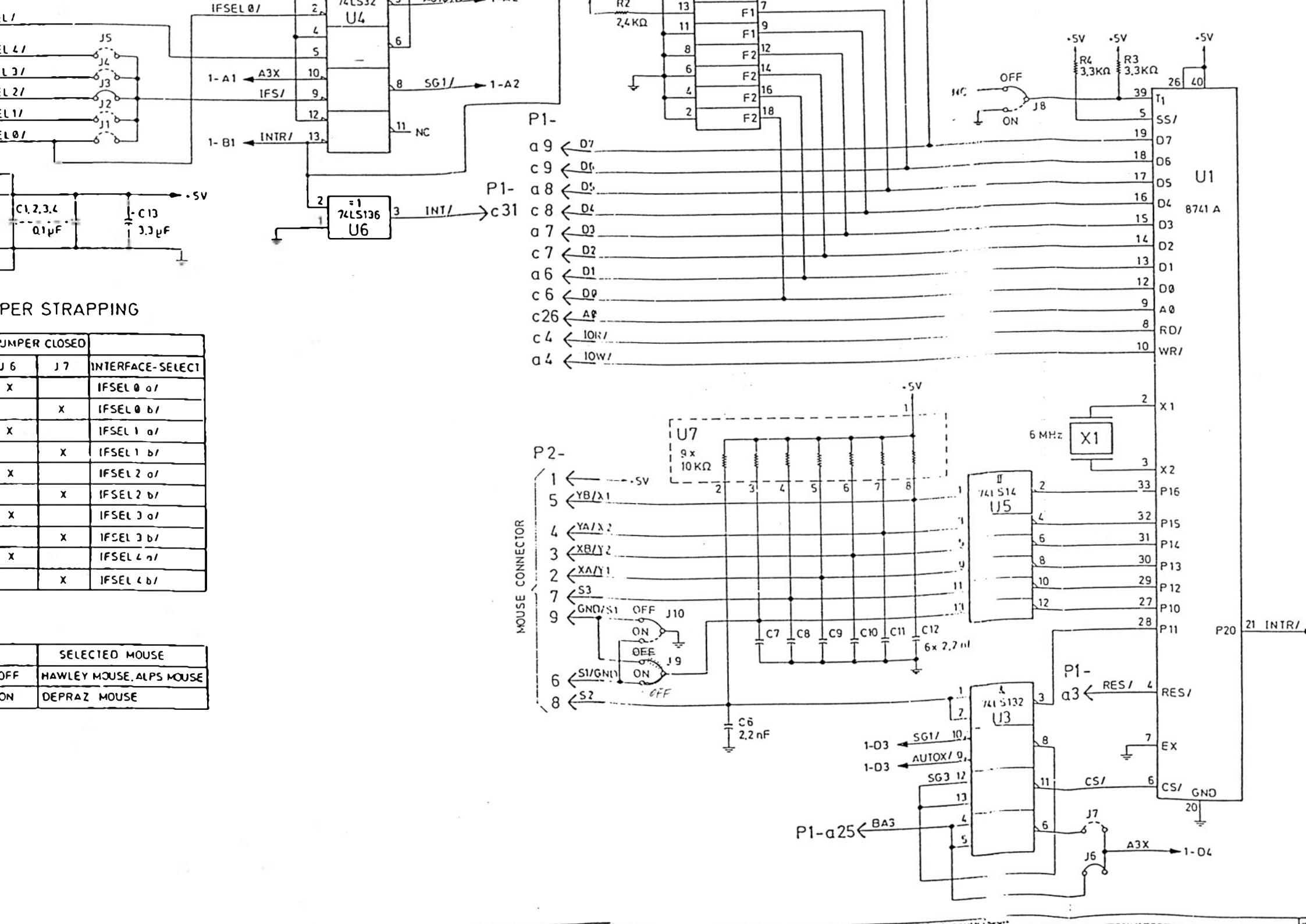
Example

Here is an example for the use of the Mouse procedures in BASIC.

In the line starting with 10, the maximum Mouse position is set by means of the X and Y coordinates. In the line starting with 20, the present Mouse position is set to zero. In the line of the main program starting with 100, the variables are collected and displayed on the screen.

SW1, SW2, and SW3 designate the switches on the Mouse (here the HAWLEY MARK II Mouse), which can be on or off. X is the Mouse position on the X coordinate; y the Mouse position on the Y coordinate.

```
        REM
        REM   MOUSE TEST PROGRAM  B A S I C
        REM

        USES MOUSE

        INTEGER SW1,SW2,SW3,X,Y,X0,Y0,XM,YM
        DIM CS*1

        SUB FIN (CS)
        REM PROMPT FOR FINISH
        INPUT AT (19,1)"    DO YOU WANT TO FINISH?   Y/N ":CS
        SUBEND

        REM CLEAR SCREEN, CURSOR INVISIBLE
        DISPLAY ERASE ALL:CHR$(27);"@0"

        DISPLAY AT (2,1): "    ******   M O U S E  EXAMPLE  B A S I C   *******
        DISPLAY AT (17,1):"    TO STOP PRESS SWITCH 1 "

        REM  SET MOUSE MAX VAL
        XM=10000
        YM=10000
10      CALL SETMAX(XM,YM)

        REM   SET MOUSE POSITION
        X0=0
        Y0=0
20      CALL SETPOS(X0,Y0)

        CS="N"

        REM LOOP MOUSE ACTION
        FOR I=0 TO 0 STEP 0
100     CALL GMOUS(SW3,SW2,SW1,Y,X)
        DISPLAY AT(5,1) USING "    X=######    Y=######":X,Y
        DISPLAY AT(7,1):"    SWITCH 1:";SW1;"  SWITCH 3:";SW3;"  SWITCH 2:";SW2

        REM    SET OR CLEAR PROMPT FOR FINISH DEPENDING ON SW1
        IF SW1=1 THEN CALL FIN(CS) ELSE DISPLAY AT (19,1):CHR$(27);"L";

        IF CS="Y" OR CS="y" THEN I=1
        NEXT I

1002    REM CURSOR VISIBLE
        DISPLAY AT (1,1):CHR$(27);"@1"
        END
```



PER STRAPPING

UMPER CLOSED		
J 6	J 7	INTERFACE-SELECT
X		IFSEL 0 a/
	X	IFSEL 0 b/
X		IFSEL 1 a/
	X	IFSEL 1 b/
X		IFSEL 2 a/
	X	IFSEL 2 b/
X		IFSEL 3 a/
	X	IFSEL 3 b/
X		IFSEL 4 a/
	X	IFSEL 4 b/

	SELECTED MOUSE
OFF	HAWLEY MOUSE, ALPS MOUSE
ON	DEPRAZ MOUSE

K i t S p e c i f i c a t i o n K806

Mouse-Interface

1.0 Scope

This specification defines the requirements for an interface between the DMV and most today known "mice".
A MOUSE is a user friendly device which is used to control the position of a cursor on the CRT-screen. This position capability combined with up to three switch options on the MOUSE enables this unit to be used to construct graphic displays or to select individual commands from a menu of commands displayed on the screen.
This specification describes only the mouse-interface and not the MOUSE itself.

The mouse-interface has its own microcomputer with firmware and is able to send at least 4000 positions per second. (see 3.6). It offloads the mainprocessor considerable by its own intelligence.

Major features of the mouse-interface are:
- compute absolute or relative mouse-position and sample switch-status of the mouse-switches
- transfer this information to the DMV in a defined format
- accept control-codes from the DMV for the mouse (interrupthandling,range-check)
- range-check for the mouse-position
- mouse-interrupt generation
- auto-config-capability

2.0 Reference Documents

2.1 Unit Dependent Documents

017-002473 FS Professional Desk Top Computer
NCR Decision Mate V
008-0072105 FS MOUSE
006-1004082 Circuit Integrated NMOS CPU with EPROM

2.2 NCR Standards

CES 2-11-01 Environment,Humidity,Temperature,Pressure
CES 2-11-08 El. Magn. Interference Emission
CES 2-11-09 El. Magn. Interference Susceptibility
CES 2-11-10 Electrostatic Discharge Requirements
CES 3-02-11 Product Safety,Design and Certification

3.0 Requirements

3.1 General Description

The mouse-interface described here is to be used with a first or second generation Desk Top Computer DMV and one mechanical mouse. A list of connectable types of mice is given by table 4.
The interface has to be connected to one interface-connector on the backside of the DMV. A similar housing to that of the RS-232- or the IEEE-488-interface is used.
The interface has a second connector to which a MOUSE must be connected. Some types of mice can be used.
The mouse-type is selected by strapping the jumpers on the interface. For first generation units of the DMV (Z80\8088) the I\O-adress of the mouse-interface can be selected by jumpers on the interface.
For second generation units of the DMV the AUTO/-signal is used. This signal tells the mouse-interface if this is a first or a second generation DMV.
No straps are neccessary for this distinction.
The electronic circuity works in conjunction with a single-chip-microcomputer to compute the mouse-position, get switch-status and transfer this information to the host-system. This circuit handles also the generation of interrupts to the host-system. An additional [illegible] for the mouse-position is performed.

3.2 Electrical Requirements

Electrical specification for the interface circuits is [illegible] based on the use of TTL-technology. Power supply should not exceed 5.25 volts and is referenced to logic ground.

Supply voltage :	5 V DC at 150 mA max.
Data line levels :	"1" 2.4 Volts min , 5.25 Volts max
	"0" 0 Volts min, 0.45 Volts max
	at 2.0 mA

3.2.1 Connector Pin Assignment for DMV
The mouse-IF is prepared to serve the DMV-bus with/without autoconfig and with/without interrupt.
(first/second generation)

3.2.1.1 First Generation

a		c
+5V	1	+5V
RESERVED	2	+12V
RESET/	3	RESET IN/
IOW/	4	IOR/
MEMW/	5	MEMR/
BD1	6	BD0
BD2	7	BD2
BD4	8	BD4
BD7	9	BD6
READY DMA	10	ABTRI/
EOP/	11	RESERVED
ESTACK/	12	IFSEL 4/
AUTO/	13	DIR/
THOLD/	14	HLDA
PCLK/	15	CLK1
LGRD	16	TRAMO/
BA19	17	BA15
BA17	18	BA16
BA15	19	BA14
BA13	20	BA12
BA11	21	BA10
BA9	22	BA8
BA7	23	BA6
BA5	24	BA4
BA3	25	BA2
BA1	26	BA0
IFSEL3/	27	IFSEL2/
IFSEL1/	28	IFSEL0/
DRQ1	29	DRQ0
DACK1/	30	DACK0/
WAIT/	31	INT/
LGRD	32	LGRD

3.2.1.2 Second Generation

a		c
+5V	1	+5V
PERC/	2	+12V
RESET/	3	RESET IN/
IOW/	4	IOR/
MEMW/	5	MEMR/
BD1	6	BD0
BD3	7	BD2
BD5	8	BD4
BD7	9	BD6
READY DMA	10	RESERVED
EOP/	11	IDSEL/
RESERVED	12	IFSEL 4/
AUTO/	13	DIR/
THOLD/	14	HLDA
PCLK/	15	CLK1
LGRD	16	TRAMO/
BA19	17	BA18
BA17	18	BA16
BA15	19	BA14
BA13	20	BA12
BA11	21	BA10
BA9	22	BA8
BA7	23	BA6
BA5	24	BA4
BA3	25	BA2
BA1	26	BA0
IFSEL 3/	27	IFSEL 2/
IFSEL 1/	28	IFSEL 0/
DRQ1/	29	DRQ0/
RESERVED	30	RESERVED
WAIT/	31	INT/
LGRD	32	LGRD

3.2.1.3 Signals used by the mouse-interface

Pin#

1	a	+5V	c	+5V
2	a	PERC/		
3	a	RESET/		
4	a	IOW/	c	IOR/
6	a	BD1	c	BD0
7	a	BD3	c	BD2
8	a	BD5	c	BD4
9	a	BD7	c	BD6
11			c	IDSEL/
12			c	IFSEL4/
13	a	AUTO/		
25	a	BA3		
26			c	BA0
27	a	IFSEL3/	c	IFSEL2/
28	a	IFSEL1/	c	IFSEL0/
31			c	INT/
32	a	LGRD	c	LGRD

3.2.2 Connector Pin Assignment for Mouse-connector

Pin#	Mouse-type	
	HAWLEY Mark II ALPS Encoder Mouse Mouse Systems Quad Mouse	DEFRAZ Souris P4 LOGITECH P4 LOGITECH LM-P-5
	Assignment	
1	+5V	+5V
2	XA	Y1
3	XB	Y2
4	YA	X2
5	YB	X1
6	S1	GND
7	S3	S3
8	S2	S2
9	GND	S1

3.3 Mechanical Requirements

Weight less then 0.400 kg
Dimensions Widht 114 mm (4.5 ")
Depht 107 mm (4.2 ")
Height 22 mm (0.9 ")

The PC-board must meet the requirements of UL 94 V2 or better.

3.4 Power-Up/Level-0-Diagnostic

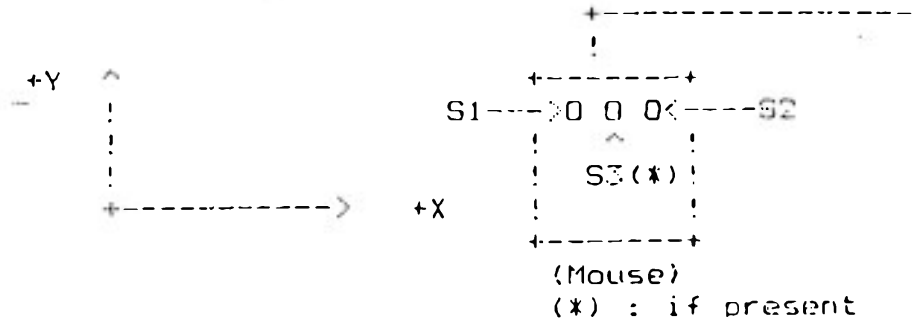
A low level on the RESET-pin of the 8741A which is connected to SYSTEM-RESET (pin 3a) initiates the microcomputer-chip to start firmware at location zero. The RESET-signal must have a minimal lenght of 20 us after power-up.
At this point the unit begins to examine the level-0-diagnostic-routine. A RAM- and a ROM-check are performed. After completion of these checks the following bits of the status-register contain level-0-diagnostic informa

bit 5 is set ---> RAM of 8741A is defective
bit 4 is set ---> ROM of 8741A is defective
Otherwise these bits are cleared.

Now the ports are set to start condition and the processor begins to count the pulses coming from the MOUSE.
The exchanging of the pins of the various mouse-types is done by the firmware in the microcomputer-chip.
Only the GND-signal is exchanged by mechanical switches.

3.5 Direction and Switch Definitions

Following definitions of the direction of the mouse-moves and the switches on the mouse are made:



3.6 Mouse-Firmware/Software-Interfacing

To access the mouse-position and the switch-status of the mouse-switches it is important to know the protocol which must be used when communicating with the mouse-interface.
There are 2 I\O-adresses used to send information to or receive information from the mouse-interface.
These I\O-adresses can be changed with jumpers J1..J5 for first generation units. (See Table 1).

Normal adjustment for first generation units:

IFSEL 2A\ I\O-adresses 30H and 31H
Command: Write Commands to\
Get Status from Mouse-interface:31H
Data: Write Data to\Get Data from Mouse-interface :30H

The other adresses selected by IFSEL 2A\ are not used.

Second Generation:
By activating the IDSEL\-line the host-system gets the ID-number of the mouse-interface.
This ID-number is 20H.
The host system knows now that there is a mouse-interface in this slot which can be adressed by activating the IFSEL-line which is private to the slot.

3.6.1 Principal Communication with the Interface

After checking the status of the mouse-interface a command byte is outputted to the command-adress (normal I\O-adress 31H). If there are any parameters which belong to the command then these parameters can be read from or written to the mouse-interface. Before each read or write a status check has to be performed. It is important that the correct number of parameters is used. A writing of a new command while the interface expects parameters of the preceding command will terminate the preceding command.

Statusbyte:	bit #	7	6	5	4	3	2	1	0
	meaning	X	INT	RAM	ROM	X	X	IBF	OBF

For a detailed description of the meaning of the statusbyte-bits see table 3.

Before writing anything to the interface it must be checked, that IBF is "0". Before reading parameters from the interface it must be checked, that OBF is "1".

3.6.2 Mouse-interface-functions

The function of the various mouse-commands and the correct number and meaning of the belonging parameters are defined in this section.

Command-code	Meaning	parameters to interface (this sequence)	parameters from interface (this sequence)
00	get absolute mouseposition	none	X-coord. low byte X-coord. high byte Y-coord. low byte Y-coord. high byte Switch-status-byte bit 7 6 5 4 3 2 1 0 X X X X X S1 S0 S1,S0: "1": switch pressed "0": switch released
01	enable interrupt by mouse-moving	none	none
02	disable interrupt	none	none
03	reset interrupt line and INT-bit	none	none
04	set logically mouseposition	X low byte X high byte Y low byte Y high byte	none
05	set maximal value for range-check of mouse-position in X- and Y-direction	XMAX low byte XMAX high byte YMAX low byte YMAX high byte	none

Command-byte	meaning	parameter to mouse-interface	parameter from mouse-interfac
06	enable interrupt for changing switch-status (OFF to ON and ON to OFF)	none	none
07	enable interrupt for mousemoving or changing status of mouse-switch	none	none
08	perform ROM-check per software-command	none	none
09	get relative mouseposition	none	see command 09
0A	set minimal value for range-check of mouseposition in X- and Y-direction	XMIN low byte XMIN high byte YMIN low byte YMIN high byte	none

Default adjustments are:
all interrupts disabled
XMAX = 640 XMIN = 0
YMAX = 400 YMIN = 0
X-coord = 0
Y-coord = 0

3.6.3 Interrupts

For both first and second generation units of the MMI interrupts from the mouseinterface are possible. Interrupts must be enabled by issueing one of the command-bytes 01,06 or 07 to enable the desired interruptmode. If the interrupt condition occurs the interface pulls the interrupt-line down to ground. To indicate that the mouseinterface has interrupted the hostprocessor bit #6 of the interface-status-byte can be used. This bit is reset by issueing the reset-interrupt-line-command (command byte 03).

Pin#
31 c INT\

3.7 Auto-config capability

For second-generation units an auto-config capability is installed on the interface. When both signals AUTO\ and IDSEL\ are driven low then the ID-number of the mouse-interface is given on the databus.
ID-number of the mouse-interface: 20H

pin#
11 c IDSEL\

3.8 Data-Transfer-Rate

The transmission of a complete block which contains mouse position and switch status needs less then 250 us. This time is measured with an assembler-program for the 8088. The other commands for the mouse-interface need a shorter time, because there are fewer bytes to be tranfered.

4.0 Environmental Requirements

The operating, shipping and storage requirements are according CES 2-11-01 range 3

EMC requirements must met CES 2-11-05
EMS " " " " 2-11-09
ESD " " " " 2-11-10

5.0 Reliability

5.1 Workload

The average workload of the unit will be:
3.2 hours activity per working day
5 working days per week
52 weeks per year

5.2 Critical Failures

Any failure or combination of failures that prohibit further use of the mouse without a service call is considered critical.

5.3 Mean Time between Failures

MTBF = 574052 h
MTBSC = 396096 h
Service Calls per Year : 0.003

Table 2. Switch-status-byte

7	6	5	4	3	2	1	0
X	X	X	X	X	S3	S2	S1

S1,S2,S3 : "1" means switch pressed
"0" means switch released

Table 3. Interface-Status-Byte

7	6	5	4	3	2	1	0
X	INT	RAM	ROM	X	X	IBF	OBF

IBF: "1" means Interface is busy,wait till ready
"0" means Interface is ready to receive data or command
OBF: "1" means Interface has one data-byte ready to read
"0" means Interface has no data-byte ready to read
ROM: "1" means ROM of 8741A is defective
"0" means ROM is ok
RAM: "1" means RAM of 8741A is defective
"0" means RAM is ok
INT: "1" means mouseinterface has issued an interrupt or the interruptline is still active (low).
"0" means mouseinterface has not issued an interrupt

Table 4. Connectable Mice

Company	Mouse-type
Alps	Encoder-mouse
Depraz	Souris P4
Hawley	Mark II
Logitech	P4 (same as Depraz Souris P4)
Logitech	LM-P-S
Mouse Systems	Quad Mouse

MS 017-0005153 Mouse-Interface Kit Nr. K806

Fig. 1. Interface-housing

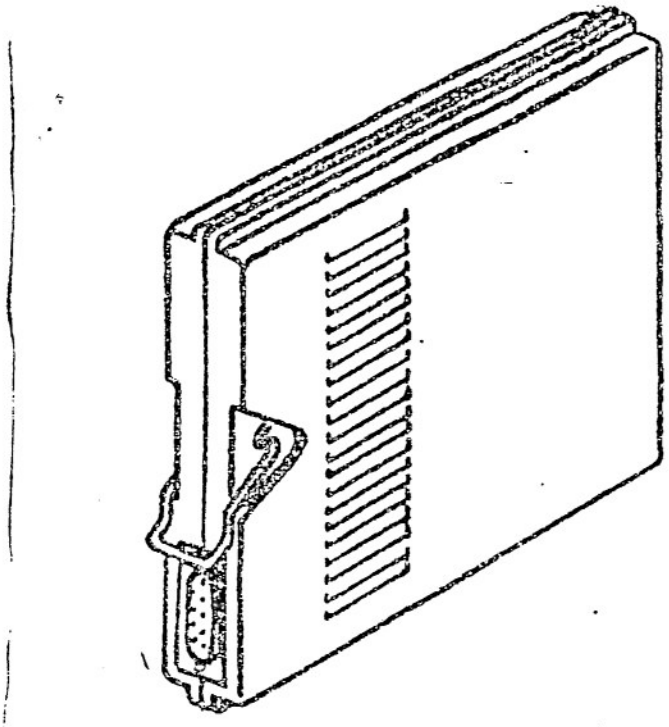


Table 1. Jumper strapping

IFSEL	JUMPER 1 2 3 4 5 6 7	PORT ADRESSES
0A	X O O O O X O	60-67 HEX
0B	X O O O O O X	68-6F HEX
1A	O X O O O X O	70-77 HEX
1B	O X O O O O X	78-7F HEX
2A	O O X O O X O	30-37 HEX
2B	O O X O O O X	38-3F HEX
3A	O O O X O X O	B0-B7 HEX
3B	O O O X O O X	B8-BF HEX
4A	O O O O X X O	C0-C7 HEX
4B	O O O O X O X	C8-CF HEX

O=open X=closed

DIP-FIX-switches
(Jumpers J8,J9,J10)

Jumpers		Selected Mouse-types	J10 J9 J8
J8,J9,J10	OFF	Hawley mouse MARK II Alps encoder-mouse Mouse Systems Quad Mouse	+-+-+-+-+-+ ! !X! !X! !X! +-+-+-+-+-+
J8,J9,J10	ON	Depraz Souris P4 Logitech LM-P-5 Logitech P4	+-+-+-+-+-+ !X! !X! !X! ! +-+-+-+-+-+

Interrupt-Service-Routine

- Registerbank 1 anwählen
- Akku-Inhalt zwischenspeichern
- Datenbusbuffer in Akku einlesen
- F1 - Flag prüfen (Command-Flag)
  falls F1 nicht aktiv, Akkuinhalt zurückholen und return
- F1 löschen
- Kommando identifizieren

  00 - Mausposition und Schalterstatus ausgeben

  01 - Enable Interrupt bei Mausbewegung

  02 - Disable alle Interrupts

  03 - Interrupt zurücksetzen

  04 - Mausposition setzen

  05 - Maximalwerte laden

  06 - Enable Interrupt bei Schalterbetätigung

  07 - Enable Interrupts bei Mausbewegung oder Schalterbetätigung

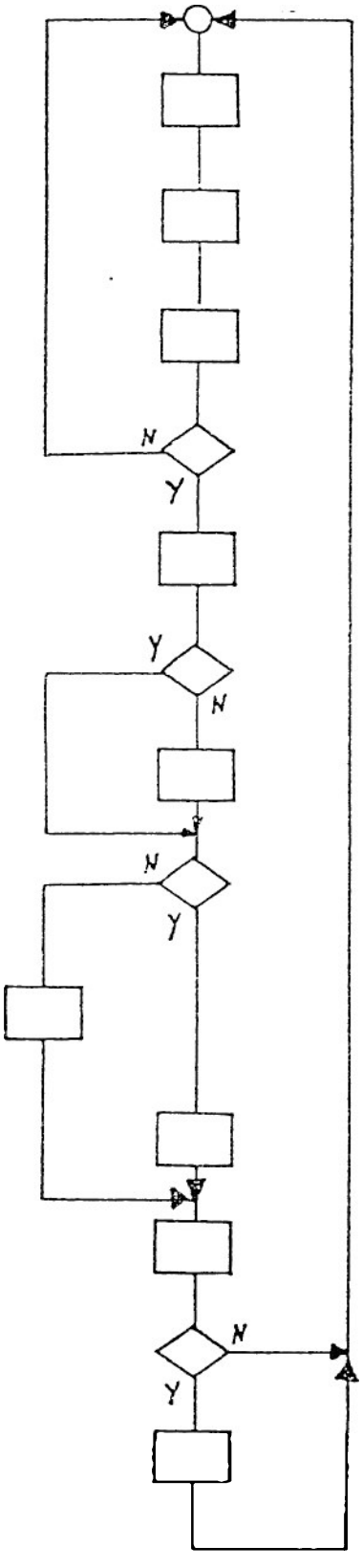
  08 - ROM-Test ausführen

  09 - Mausposition und Schalterstatus ausgeben, anschließend Mausposition auf Ø setzen

  0A - Minimalwerte laden

- Kommando ausführen
- Akkuinhalt zurückholen
- Return

IDLE - Routine



Interrupt enable

Mouse-Status einlesen

Eingelesenen Status mit bisherigem Status vergleichen

Statusänderung?

Interrupt disable

HAWLEY oder ALPS-Mouse?

Portbits vertauschen

Schalterbits verändert?

Neue Mouse-Position im Rahmen der Min. und Max-Werte berechnen

Neuen Schalterstatus abspeichern

F0 setzen

Interrupts enabled?

INTR/-Leitung aktivieren und Statusbit 6 setzen

Mouse - Firmware (K806)

Initialisierungsroutine:

- Stackpointer initialisieren
- Interrupt disablen
- RBØ selektieren

} falls Software-Reset

- RAM-Test (64 Byte)
  - alle Bits auf 1 setzen und lesen
  - Adressentest (0, 1, 2, 3 ....)
  - alle Bits auf 0 setzen und lesen
  - im Fehlerfall: Statusbit 5 setzen

- ROM-Test (Subroutine, 2-fach, verschachtelt)
  - R2-Register retten
  - ROM-Test Page Ø
  - ROM-Test Page 1
  - ROM-Test Page 2
  - ROM-Test Page 3
  - Im Fehlerfall: Statusbis 4 setzen
  - R2-Register mit Ursprungswert laden

- Minimal- und Maximalwerte der X- und Y-Koordinaten von ROM ins RAM übernehmen.

  X : 0 , 640

  Y : 0 , 400

- Mouse-Status lesen und speichern

PRELIMINARY / NOT RELEASED

FUNCTIONAL SPECIFICATION

C 3 2 7 3

K - 8 0 4 I E E E - 4 8 8 - 1 F

TABLE OF CONTENTS:

1 INTRODUCTION

1.1 GENERAL DESCRIPTION

1.2 REFERENCE DOCUMENTS

2 FUNCTIONAL DESCRIPTION

2.1 GENERAL DESCRIPTION

2.2 SUMMARY OF IEEE-488 FEATURES

2.3 TALKER/LISTENER ADDRESS

2.4 IEEE-488 INTERFACE BUS

2.4.1 SIGNAL LINES

2.4.2 SIGNAL LINE TIMING SEQUENCE

2.4.3 SIGNAL LEVEL

2.4.4 CABLE

2.4.5 CONNECTOR

2.4.6 IEC-625 CONNECTOR

2.5 DM-V INTERFACE BUS

2.5.1 DM-V INTERFACE BUS (1. GENERATION)

2.5.1.1 IF-SELECTION

2.5.1.2 INTERRUPT

2.5.1.3 DATA TRANSMISSION

2.5.1.4 DMA

2.5.1.5 TIMING

2.5.2 DM-V INTERFACE BUS (2. GENERATION)

2.5.2.1 IF-IDENTIFICATION

2.5.2.2 IF-SELECTION

2.5.2.3 INTERRUPT

2.5.2.4 DATA TRANSMISSION

2.5.2.5 DMA

2.5.2.6 TIMING

3 ENVIRONMENTAL CHARACTERISTICS

4 STANDARD REQUIREMENTS

5 RELIABILITY

6 MAINTAINABILITY

APPENDIX A: K804 IEEE-488 - IF
INSTALLATION, FIGURES, TABLES
APPENDIX B: K804 IEEE-488 - IF
NEC uPD 7210 INTELLIGENT GPIB INTERFACE CONTROLLER
DATA SHEETS

C - 3 2 7 3

K 8 0 4 I E E E - 4 8 8 I F

1 INTRODUCTION

1.1 GENERAL DESCRIPTION

K804 is a connection module between two buses:

(A) GENERAL PURPOSE INTERFACE BUS (GPIB)

(B) DM-V BUS

(A) GENERAL PURPOSE INTERFACE BUS (GPIB)

K804 is the DM-V interface to the Genereal Purpose Interface Bus (GPIB) for asynchronous communication. For data transmission on this bus a byte-serial, 8 bit parallel data transfer is used.

Up to 15 devices can be interconnected to a distance up to 60 feet (20 m) using this concept.

The maximal data transfer rate is 250 kByte per second using DMA.

One active controller is permitted in a GPIB system configuration.

The NCR-GPIB Interface K804 is designed according to the following specifications:

- ANSI/IEEE STD 488-1975
- ANSI/IEEE STD 488-1978
- ANSI/IEEE STD 488A-1980
- IEC-625/1 STANDARD

(ANSI - AMERICAN NATIONAL STANDARD INSTITUTE)
(IEEE - INSTITUTE OF ELECTRICAL AND ELECTRONICS ENG.)
(IEC - INTERNATIONAL ELECTROTECHNICAL COMMISSION)

Interface function, message coding, driver and receive, and the interface bus are defined in the above standards.

A bus converter is required to adapt the IEEE-488 IF to the IEC-625 bus (see 2.4.6).

(B) DM-V BUS

(1) K804 is prepared to serve the DM-V Bus without AUTOCONFIG. (1. Generation) and with AUTOCONFIG. (2.Generation).

(2) K804 is prepared to serve Z80/8088 based DM-V's without interrupt and 80186 based systems with interrupt.

(3) K804 is selectable by

a) IFSEL X and BA3 (1. Generation DM-V Bus)

via switches located inside the adapter housing.

b) ID-Number (2. Generation DM-V Bus)

1.2 REFERENCE DOCUMENTS

(1)	017-0024673	FS Professional Desktop Computer DECISION MATE V.
(2)		NEC Product Description: GPIB Controller uPD 7210
(3)		Texas Intruments Interface Circuits: SN 75160A / SN 75161A Bulletin No. DL-S 12786 Oct. 1980 DL-S 12787 Oct. 1980

NCR Standards

(1) CES 2-11-01 Environment, Humidity, Temperature, Presure
(2) CES 2-11-08 Electromagnetic Interference (Emission)
(3) CES 2-11-09 Electromagnetic Interference Suspectibility
(4) CES 2-11-10 Electrostatic Discharge
(5) CES 2-11-11 Product Safety, Design and Certification

NCR external Standards:

(A)		Radio Protection:
(1)	USA	FCC Docket-# 20780, Class B
(2)	GERMANY	VDE 0871, Class A Certification by GERMAN FEDERAL POST (FTZ)
(B)		GPIB Standards:
(1)	USA	ANSI/IEEE STD 488-1978 (Revision of ANSI/IEEE STD 488-1975) (Includes Supplement IEEE STD 488A-1980)
(2)	GERMANY	DIN IEC 625 Teil 1 DIN DEUTSCHES INSTITUT FUER NORMUNG E.V. DK 621.317.7.037.37:681.3.06 MAI 1981

2 FUNCTIONAL DESCRIPTION

2.1 GENERAL DESCRIPTION

K804 is based on the NEC 7210 Intelligent GPIB Interface Controller.
(see Appendix B, 7210 GPIB Interface Controller Data Sheets

Integrated GPIB bus driver/receiver SN75160/161 are used.

2.2 SUMMARY OF IEEE-488 IF (K804) FEATURES

This is a list of the general features the IEEE-488 IF will provide:

- SH1 (Source Handshake)
  Capability to guarantee proper transfer of multiline messages.

- AH1 (Acceptor Handshake)
  Capability to guarantee proper reception of remote multiline messages.

- T5 (Talker)
  Capability to send device-dependent data via the interface to other devices.

- TE5 (Extended Talker)
  Talker with extended (2 byte) address.

- L3 (Listener)
  Capability to receive device-dependent data from other devices.

- LE3 (Extended Listener)
  Listener with extended (2 byte) address.

- SR1 (Service Request)
  Capability to request service asynchronously from the controller-in-charge of the interface.

- RL1 (Remote Local)
  Capability to select between two sources of input information.

- PP1 or PP2 (Parallel Poll for Remote or Local Configuration)
  Capability to present a PPR message to the controller-in-charge without being previousl addressed to talk.

- DC1 (Device Clear)
  Capability to be cleared (initialized) eithe individually or as part of a groupb of devices.

- DT1 (Device Trigger)
  Capability to have its basic operation started either individually or as part of a group of devices.

- C1 - C28 (Controller)
  Capability to send device addresses, universal commands and addressed commands.

- Programmable Data Transfer Rate (low / high speed)

- 16 CPU Accessible Registers (8 Read- and 8 Write-Registers)

- 2 Address Registers

  - Detection of MTA (My Talker Address), MLA (My Listener Address, MSA (My Secondary Address). (to set Talker, Listener and Secondary Addresses the Configuration Utility is used. Refer FS IEEE-488 SW

  - 2 Device Addresses.

- EOS Message Automatic Detection.

- Command Automatic Processing and undefind Command read Capability.

- DMA Capability.

For a more detailed description of the interface features see Appendix B: 7210 GPIB INTERFACE CONTROLLER NEC DATA SHEETS.

2.3 TALKER / LISTENER ADDRESS

K804 has no talker/listener address switch. To set these addresses the CONFIGuration utility will be used. The controller default address is zero.

2.4 IEEE-488 INTERFACE BUS

2.4.1 SIGNAL LINES

The IEEE-488 interface bus has 16 signal and data lines. (refer IEEE-488 Standard)

- 8 DATA lines are used to transfer data, addresses an control information. The formats are defined by IEEE-488.

- 5 MANAGEMENT control lines:
  - ATN ATtentioN
    is used (by the controller) to specify how data on the DIO signal lines are to be interpreted and which devices must respond to the data.
  - EOI End Or Identify
    is used (by a talker) to indicate the end of a multiple byte transfer se-

quence
or, in conjunction with ATN (by a controller), to execute a polling sequence

- SRQ Service ReQuest
  requests the controller to take control
- IFC InterFace Clear
  is used (by a controller) to place the interface system (in all interconnected devices) in a known quiescent state.
- REN Remote ENable
  is used (by a controller) in conjunction with other messages to select between two alternate sources of device programming data.

- 3 HANDSHAKE lines:
  - NRFD Not Ready For Data
    is used to indicate the condition of readiness of device(s) to accept data.
  - NDAC Not Data ACcepted
    is used to indicate the condition of acceptance of data by device(s).
  - DAV DAta Valid
    is used to indicate the condition (availibility and validity) of information on the DIO signal lines.

4.4.2 SIGNAL LINE TIMING SEQUENCE

For data transmission the IEEE-488 IF is using the handshake process acc. to the IEEE-488 standard.

4.4.3 SIGNAL LEVEL

Electrical specifications for the interface circuits are based on the use of TTL-technology. Power source does not exceed 5.25 VDC.

GPIB-Driver: (three state)

LOW state Output <+0.5 V
+48 mA Sink Current (cont.

HIGH state Output >+2.4 V
-5.2 mA

GPIB-Receiver:

LOW state Input <0.8 V

HIGH state Input >2.0 V

2.4.5 CABLE

The IEEE-488 IF will have a cable (acc. to IEEE-488 standard) of 1.00 meter length.
Only cables according to the IEEE-488 standard are

allowed to expand the bus.

2.4.5 CONNECTOR

Acc. IEEE-488 Standard. See App. A

2.4.6 IEC-625 CONNECTOR

The IEEE-488 and the IEC-625 Standards do define different connectors to the devices. (see App. A)
Using the K804 to control the IEC-625 bus commercially available converters are required:

(A) IEEE-488 to IEC-625 Converter
(B) IEC-625 to IEEE-488 Converter

Some companies offering these converters are listed in App. A.

2.5 DM-V INTERFACE BUS

The IEE-488 IF (K804) is prepared to serve the DM-V bus without AUTOCONFIG. (1. Generation) and with AUTO-CONFIG. (2.Generation).
(see FS Decision Mate V, 017-0024673, Rev. C, App.A)

2.5.1 DM-V INTERFACE BUS (1. GENERATION)

2.5.1.1 IF-SELECTION

The IEEE-488 IF is addressed by IFSEL X and BA3. This means, that the IF does require one IFSEL-line and 8 of 16 port-addresses.
IFSEL X and BA3 are selectable by DIP-switches located inside the adapter housing.
(see appendix A)

IFSEL X	:	0 , 1 , 2 , 3 , 4
BA3 = low	:	A
BA3 = high	:	B

2.5.1.2 INTERRUPT

The interrupt output signal will be permanent disabled for the Z80/8088 based DM-V.

2.5.1.3 DATA TRANSMISSION

Data transmission is indicated by the Interrupt Status Register bits DI and DO. The DI bit indicates, that a data byte is written into the Data-in-Register from the GPIB. This means the CPU must read the Data-In-Register.The DI bit is reset by reading the Data-In-Register.
The DO bit is reset by writing data to the Data-Out-Register. DO is set when data is accepted by the receiver device.
The maximal data transfer rate may be reached by using DMA.

2.5.1.4 DMA

The default DMA-channel is 0. The PC-board layout does allow a change to DMA-channel 1 by inserting a wire and cutting the channel 0 selection.

DRQ is active HIGH.
DACKX/ (DMA-Acknowledge) is active LOW and sent by the DMA controller on the DM-V mainboard.

2.5.1.5 TIMING

In Z80/8088 environment the K804 does not require any WAIT-States.

2.5.2 DM-V INTERFACE BUS (2. GENERATION)

2.5.2.1 IF-IDENTIFICATION

The ID-number of K804 is

This ID-# is readable by IDSEL (BD0 - BD6).

2.5.2.2 IF-SELECTION

The IF is selected by IFSEL/ (active low).

2.5.2.3 INTERRUPT

The K804 will activate different interrups if enabled by masks.

Bus pin INT/ activates the processor interrupt. To find out which IF-adapter did send the interrupt request, the CPU must scan with IDSEL all IF-adapters. BD7 active LOW indicates the interrupt from the scanned adapter.

2.5.2.4 DATA TRANSMISSION

See 2.5.1.4

2.5.2.5 DMA

DRQX/ is active LOW.
DACKX is generated by the IEEE-488 IF logic.

2.5.2.6 TIMING

WAIT - States have to be programmed for K804 IN/OUT commands.

3 ENVIRONMENTAL CHARACTERISTICS

See FS 017-0024673 Par. 5 for general environmental requirements.

Electrical Requirements:

+ 5.1 VDC +/- 3% XX.XX ADC

4 STANDARD REQUIREMENTS

See FS 017-0024673 Par. 6 for general Standard requirements.

5 RELIABILITY

5.1 LIFETIME

Operating life of the K804 is 5 years or 10.000 hours

without major overhaul, based on the average application workload.

5.2 AVERAGE WORKLOAD

The average workload of the K804 will be:

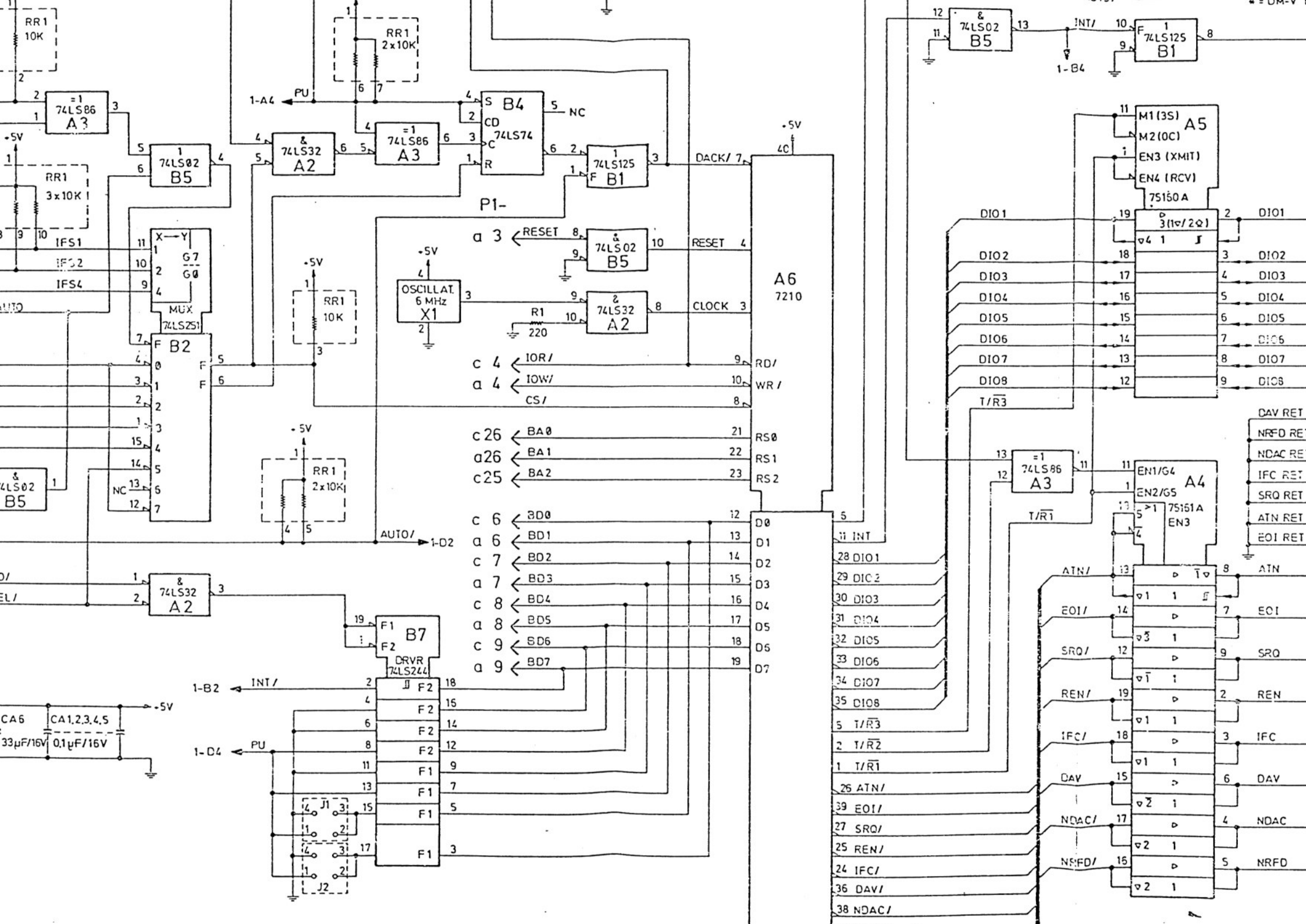
3.2 hours activity per working day
5 working days per week
52 weeks per year

5.3 MTBF, MTBSC AND SERVICE CALLS/YEAR

	MTBF/h	MTBSC/h	SERVICE-CALLS/YEAR
K801	581.395	401.162	0.002

6 MAINTAINABILITY

TBD


RR1 10K
RR1 2x10K
RR1 3x10K
RR1 10K
RR1 2x10K
=1 74LS86 A3
=1 74LS86 A3
& 74LS32 A2
& 74LS32 A2
1 74LS02 B5
& 74LS02 B5
& 74LS02 B5
1 74LS125 F B1
1 74LS125 F B1
S B4 CD 74LS74 C R
NC
PU
1-A4
P1-
DACK/ 7
RESET 4
CLOCK 3
RD/
WR/
CS/
RS0
RS1
RS2
D0
D1
D2
D3
D4
D5
D6
D7
A6 7210
+5V
OSCILLAT. 6 MHz X1
R1 220
RESET
IOR/
IOW/
BA0
BA1
BA2
BD0
BD1
BD2
BD3
BD4
BD5
BD6
BD7
IFS1
IFS2
IFS4
X→Y G7 G0 MUX 74LS251
F B2
AUTO/
1-D2
NC
F1 F2 B7 DRVR 74LS244
1-B2
INT/
1-D4
PU
J1
J2
CA6 33µF/16V
CA1,2,3,4,5 0,1µF/16V
1-B4
INT/
M1(3S) M2(0C) EN3 (XMIT) EN4 (RCV) A5 75150A
DIO1
DIO2
DIO3
DIO4
DIO5
DIO6
DIO7
DIO8
T/R3
T/R1
EN1/G4 EN2/G5 A4 75151A EN3
DAV RET
NRFD RET
NDAC RET
IFC RET
SRQ RET
ATN RET
EOI RET
ATN/
EOI/
SRQ/
REN/
IFC/
DAV
NDAC/
NRFD/
ATN
EOI
SRQ
REN
IFC
DAV
NDAC
NRFD
11 INT
28 DIO1
29 DIO2
30 DIO3
31 DIO4
32 DIO5
33 DIO6
34 DIO7
35 DIO8
5 T/R3
2 T/R2
1 T/R1
26 ATN/
39 EOI/
27 SRQ/
25 REN/
24 IFC/
36 DAV/
38 NDAC/

PRELIMINARY / NOT RELEASED

FUNCTIONAL SPECIFICATION
APPENDIX A

C 3 2 7 3

K - 8 0 4 I E E E - 4 8 8 - I F

TABLE OF CONTENTS:

1 INSTALLATION

2 SOFTWARE STRAPPING

3 DM-V I/O BUS PIN ASSIGNMENT

4 IEEE-488 CONNECTOR

5 IEEE-488 INTERCONNECTION CABLE

6 IEC-625 CONNECTOR

7 IEEE-488 / IEC-625 CONVERTER

1 KIT INSTALLATION

This Kit should be prepared for use with peripheral devices using the IEEE-488 bus according to the IEEE-488 Standard Digital Interface for Programmable Instrumentation ANSI/IEEE STD 488-1978, 488A-1980.

The IF-SELection switches have to be set according to the CPU type:

(A) Z80/8088 see 1.1
(B) 80186 see 1.2

1.1 DM-V WITH Z80/8088.

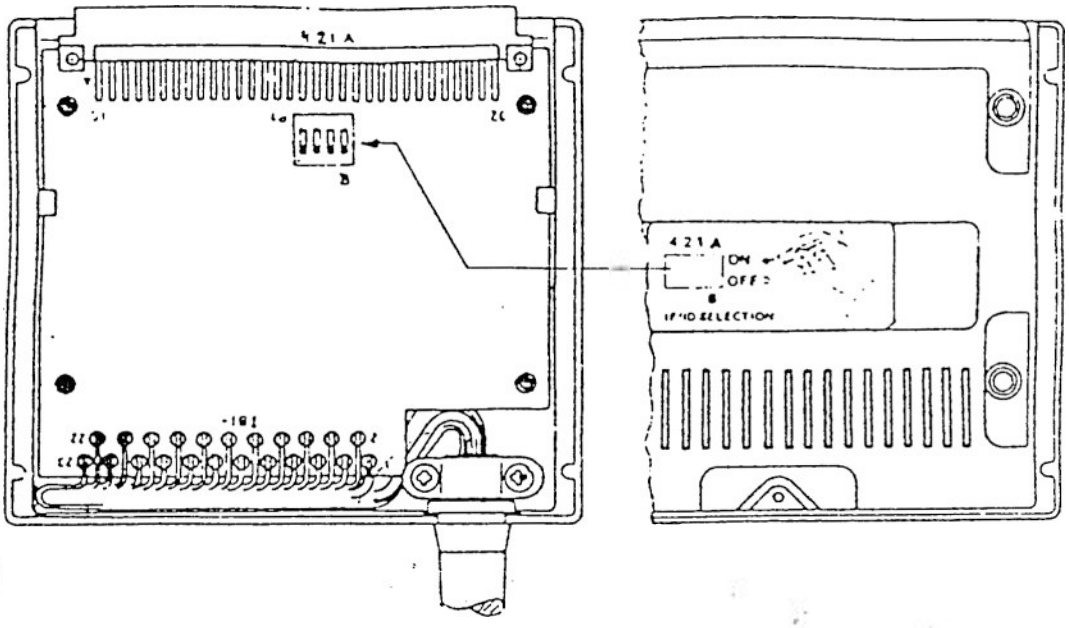
1. The adapter IFSEL switches located inside the adapter housing have to be set according to the software being used.
   Normally these switches are set as shown below (default values) to 28 :

Should your system require a different IFSEL then the switches may be set as shown in the following table:

IFSEL	SWITCH 4 2 1 A	PORT-ADDR.
0A	o o o ●	60-67H
0B	o o o o	68-6FH
1A	o o ● ●	70-77H
1B	o o ● o	78-7FH
2A	o ● o ●	30-37H
2B	o ● o o	38-3FH
3A	o ● ● ●	B0-B7H
3B	o ● ● o	B8-BFH
4A	● o o ●	C0-C7H
4B	● o o o	C8-CFH

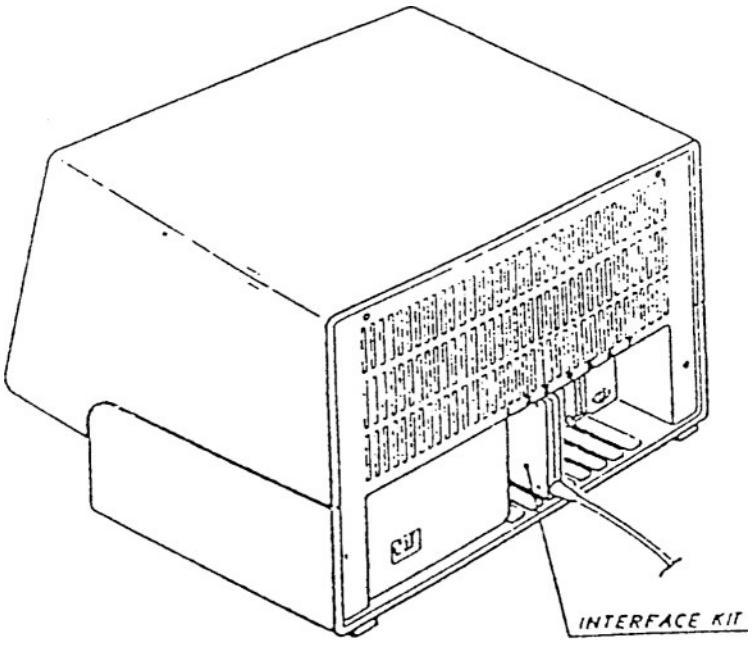
OFF : o ON : ●

To change the switches open the adapter and set the switches according to the label you find inside.



IEEE-488 Adapter InterFace SELect Switch Setting.

2. When the adapter has been prepared fit into a vacant slot at the rear of the NCR DECISION MATE V (slots 2 to 6 may be used).



3. Connect the plug to the IEEE-488 compatible device. If you are using additional cables make shure, that these match the IEEE-488 standards.

4. Check the device documentation for any strapping or switch setting (Talker / Listener Address) requirements.

5. Refer to the IEEE-488 CONFIG desciption of your IEEE-488 Software Support Package.

1.2 DM-V WITH 80186.

The IFSEL switches are not activated in a DM-V /80186 environment.
The K804 has a unique ID-Number : 15H

For installation see 1.1.2 to 1.1.5.

2 SOFTWARE STRAPPING

With a CONFIGURE Routine the user will be able to configure an individual IEEE-488 system. He can assign device addresses to device numbers used in his program, and adjust the transfer mode. The whole configuration data set is stored on disk.

OPERATING SYSTEM	SOFTWARE FS NUMBER
UCSD-P	017-0005182
CP/M	TBD
CP/M86	TBD
MSDOS	TBD

3 DM-V I/O BUS PIN ASSIGNMENT

The pin assignments for the I/O bus connector are shown in Fig. 3.1 for Z80/8088 based systems and in Fig. 3.2 for 80186 based systems.
These signals match those of the I/O bus and are defined in the bus description of Appendix A of FS 017-0024673 Rev.C.

Fig. 3.1: DM-V I/O BUS (1. Generation)

A	PIN	C
+5V	1	+5V
	2	+12V
RESET/	3	
IOW/	4	IOR/
	5	
BD1	6	BD0
BD3	7	BD2
BD5	8	BD4
BD7	9	BD6
	10	
	11	
	12	IFSEL4/
	13	DIR/
	14	
	15	
LGRD	16	
	17	
	18	
	19	
	20	
	21	
	22	
	23	
	24	
BA3	25	BA2
BA1	26	BA0
IFSEL3/	27	IFSEL2/
IFSEL1/	28	IFSEL0/
DRQ1	29	DRQ0
DACK1/	30	DACK0/
	31	
LGRD	32	LGRD

Fig. 3.2: DM-V I/O BUS (2. Generation)

A	PIN	C
+5V	1	+5V
PERC/	2	+12V
RESET/	3	
IOW/	4	IOR/
	5	
BD1	6	BD0
BD3	7	BD2
BD5	8	BD4
BD7	9	BD6
	10	
	11	IOSEL/
	12	IFSEL
AUTO/	13	DIR/
	14	
	15	
LGRD	16	
	17	
	18	
	19	
	20	
	21	
	22	
	23	
	24	
BA3	25	BA2
BA1	26	BA0
IFSEL/	27	IFSEL
IFSEL/	28	IFSEL
DRQ1/	29	DRQ0/
	30	
	31	INT/
LGRD	32	LGRD

4 IEEE-488 CONNECTOR

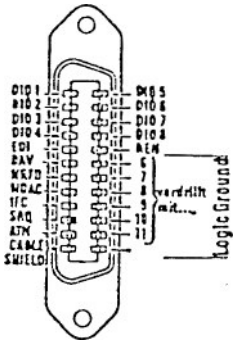


Fig. : IEEE-488 Connector

5 IEEE-488 INTERCONNECTION CABLE

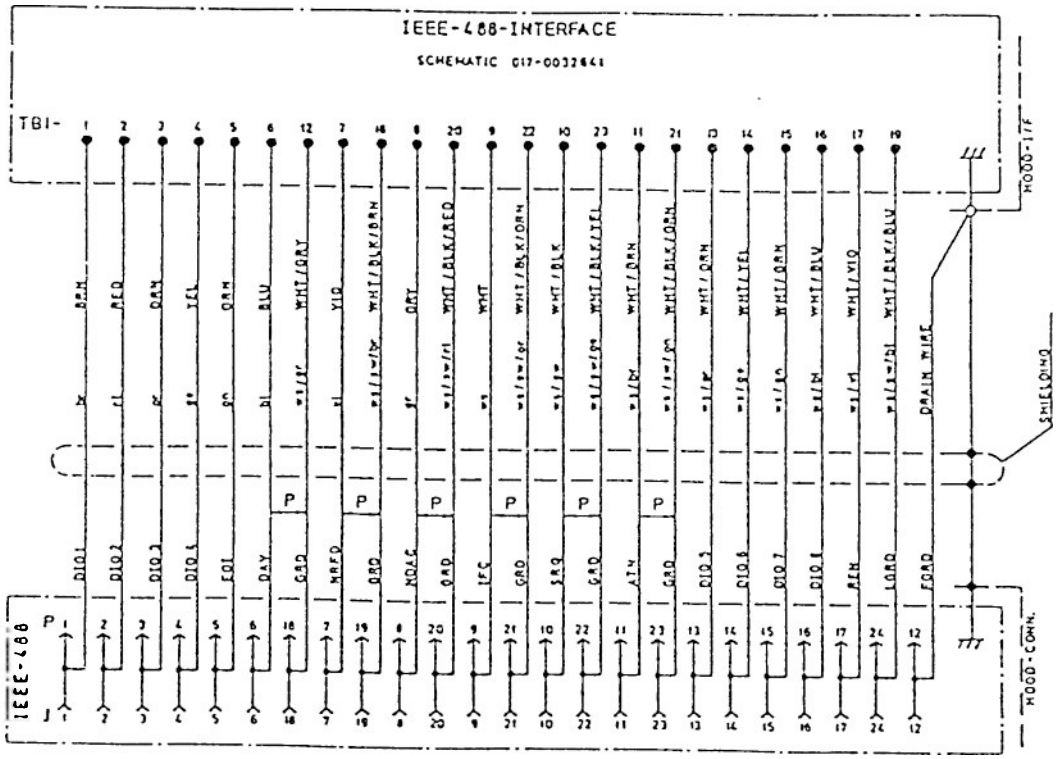


Fig. : Schematic IEEE-488 IF Cable

6 IEC-625 CONNECTOR

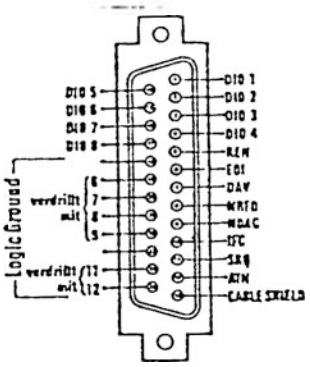


Fig. 6.1: IEC-625 Connector

7 IEEE-488 / IEC-625 CONVERTER

The IEEE-488 and the IEC-625 Standards define different connectors to the devices.
Using the NCR IEEE-488 Adapter K804 to control the IEC-625 bus commercially available converters are required:

(1) IEEE-488 to IEC-625 Converter
(AMPHENOL TUCHEL Type ICC-2)

(2) IEC-625 to IEEE-488 Converter
(AMPHENOL TUCHEL Type ICC-1)

==========
PRELIMINARY / NOT RELEASED
==========

FUNCTIONAL SPECIFICATION
APPENDIX B

C 3 2 7 3

K - 8 0 4 I E E E - 4 8 8 - IF
==========

1 NEC µPD7210 INTELLIGENT GPIB INTERFACE CONTROLLER DATA SHEETS

NEC Electronics (Europe) GmbH

NEC
µPD7210

INTELLIGENT GPIB INTERFACE CONTROLLER

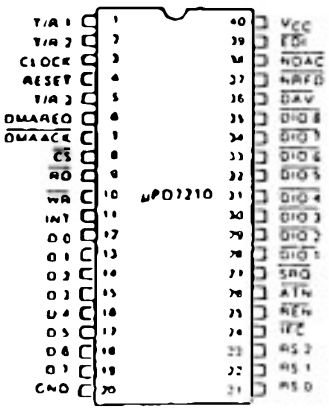
DESCRIPTION

The µPD7210 TLC is an intelligent GPIB Interface Controller designed to meet all of the functional requirements for Talkers, Listeners, and Controllers as specified by the IEEE Standard 488-1978. Connected between a processor bus and the GPIB, the TLC provides high level management of the GPIB to unburden the processor and to simplify both hardware and software design. Fully compatible with most processor architectures, Bus Driver/Receivers are the only additional components required to implement any type of GPIB interface.

FEATURES

- All Functional Interface Capability Meeting IEEE Standard
  - SH 1 (Source Handshake)
  - AH 1 (Acceptor Handshake)
  - T5 or TE5 (Talker or Extended Talker)
  - L3 or LE3 (Listener or Extended Listener)
  - SR1 (Service Request)
  - RL1 (Remote Local)
  - PP1 or PP2 (Parallel Poll [Remote or Local Configuration])
  - DC 1 (Device Clear)
  - DT1 (Device Trigger)
  - C1-5 (Controller [All Functions])
- Programmable Data Transfer Rate
- 16 MPU Accessible Registers – 8 Read/8 Write
- 2 Address Registers
  - Detection of MTA, MLA, MSA (My Talk/Listen/Secondary Address)
  - 2 Device Addresses
- EOS Message Automatic Detection
- Command (IEEE Standard 488-78) Automatic Processing and Undefined Command Read Capability
- DMA Capability
- Programmable Bus Transceiver I/O Specification (Works with T.I./Motorola/Intel)
- 1 to 8 MHz Clock Range
- TTL Compatible
- N Channel MOS
- +5V Single Power Supply
- 40 Pin Plastic DIP
- 8080/85/86 Compatible

PIN CONFIGURATION

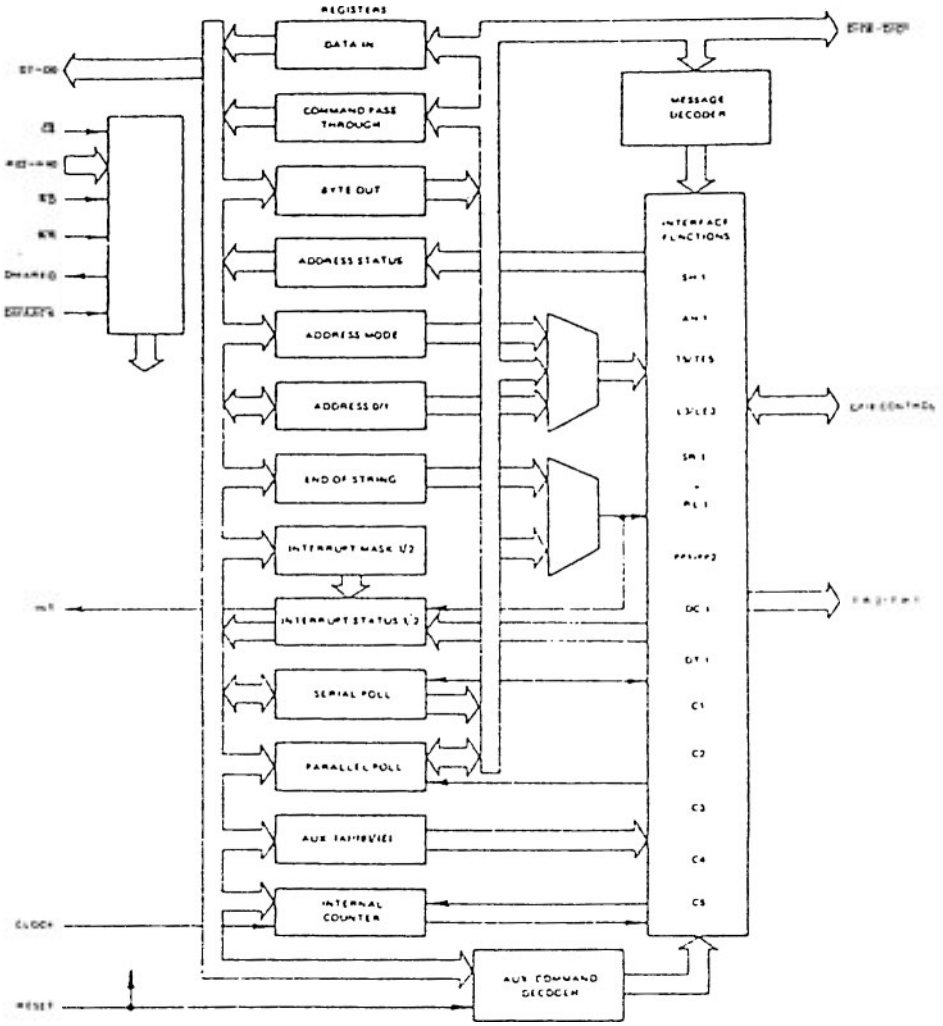


μPD7210

PIN IDENTIFICATION

PIN	NAME	I/O	DESCRIPTION
1	T/R1	O	Transmit/Receive Control – Input/Output Control Signal for the GPIB Bus Transceivers.
2	T/R2	O	Transmit/Receive Control – The function of T/R2, T/R1 are determined by the value of TRM1, TRM0 of the address mode register.
3	CLK	I	Clock – (1-8 MHz) Reference Clock for generating the state change prohibit times T1, T6, T7, T8 specified in IEEE Standard 488 1978
4	RST	I	Reset – Resets 7210 to an idle state when high (active high)
5	T/R3	O	Transmit/Receive Control – Function determined by TRM1 and TRM0 of address mode register (See T/R2)
6	DRQ	O	DMA Request – 7210 requests data transfer to the computer system, becomes low on input of DMA acknowledge signal $\overline{DACK}$.
7	$\overline{DACK}$	I	DMA Acknowledge – (Active Low) Signal connects the computer system data bus to the data register of the 7210
8	$\overline{CS}$	I	Chip Select – (Active Low) Enables access to the registers selected by RS0-2 (read or write operation)
9	$\overline{RD}$	I	Read – (Active Low) Places contents of read register specified by RS0-2 – on D0-7 (Computer Bus)
10	$\overline{WR}$	I	Write – (Active Low) writes data on D0-7 into the write register specified by RS0-2.
11	INT / $\overline{INT}$	O	Interrupt Request – (Active High/Low) Becomes active due to any 1 of 12 internal interrupt factors (unmasked) active state software configurable, active high on chip reset.
12-19	D0-7	I/O	Data Bus – 8 bit bidirectional data bus, for interface to computer system
20	GND		Ground
21-23	RS0-2	I	Register Select – These lines select one of eight read (write) registers during a read (write) operation
24	$\overline{IFC}$	I/O	Interface Clear – Control line used for clearing the interface functions.
25	$\overline{REN}$	I/O	Remote Enable – Control line used to select remote or local control of the devices
26	$\overline{ATN}$	I/O	Attention – Control line which indicates whether data on DIO lines is an interface message or device dependent message
27	$\overline{SRQ}$	I/O	Service Request – Control line used to request the controller for service
28-35	$\overline{DIO1-8}$	I/O	Data Input/Output – 8 bit bidirectional bus for transfer of message on the GPIB.
36	$\overline{DAV}$	I/O	Data Valid – Handshake line indicating that data on DIO lines is valid
37	$\overline{NRFD}$	I/O	Ready for Data – Handshake line indicating that device is ready for data
38	$\overline{NDAC}$	I/O	Data Accepted – Handshake line indicating completion of message reception
39	$\overline{EOI}$	I/O	End or Identify – Control line used to indicate the end of multiple byte transfer sequence or to execute a parallel polling in conjunction with ATN
40	V_{CC}		+5V DC – Technical Specifications: +5V; NMOS; 500 MW; 40 Pins; TTL Compatible; 1-8 MHz.

BLOCK DIAGRAM



μPD7210

INTRODUCTION

The IEEE Standard 488 describes a "Standard Digital Interface for Programmable Instrumentation" which, since its introduction in 1975, has become the most popular means of interconnecting instruments and controllers in laboratory, automatic test and even industrial applications. Refined over several years, the 488-1978 standard, also known as the General Purpose Interface Bus (GPIB), is a highly sophisticated standard providing a high degree of flexibility to meet virtually most all instrumentation requirements. The μPD7210 TLC implements all of the functions that are required to interface to the GPIB. While it is beyond the scope of this document to provide a complete explanation of the IEEE 488 Standard, a basic description follows:

The GPIB interconnects up to 15 devices over a common set of data control lines. Three types of devices are defined by the standard: Talkers, Listeners, and Controllers, although some devices may combine functions such as Talker/Listener or Talker/Controller.

Data on the GPIB is transferred in a bit parallel, byte serial fashion over 8 Data I/O lines (DIO1 – DIO8). A 3 wire handshake is used to ensure synchronization of transmission and reception. In order to permit more than one device to receive data at the same time, these control lines are "Open Collector" so that the slowest device controls the data rate. A number of other control lines perform a variety of functions such as device addressing, interrupt generation, etc.

The μPD7210 TLC implements all functional aspects of Talker, Listener and Controller functions as defined by the 488-1978 Standard, and on a single chip.

GENERAL

The μPD7210 TLC is an intelligent controller designed to provide high level protocol management of the GPIB, freeing the host processor for other tasks. Control of the TLC is accomplished via 16 internal registers. Data may be transferred either under program control or via DMA using the TLC's DMA control facilities to further reduce processor overhead. The processor interface of the TLC is general in nature and may be readily interfaced to most processor lines.

In addition to providing all control and data lines necessary for a complete GPIB implementation, the TLC also provides a unique set of bus transceiver controls permitting the use of a variety of different transceiver configurations for maximum flexibility.

INTERNAL REGISTERS

The TLC has 16 registers, eight of which are read and 8 write.

REGISTER NAME	ADDRESSING					SPECIFICATION							
	R S 2	R S 1	R S 0	WR	CS								
Data In [0R]	0	0	0	WR	CS	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
Interrupt Status 1 [1R]	0	0	1	WR	CS	CPT	APT	DET	END	DEC	ERR	DO	DI
Interrupt Status 2 [2R]	0	1	0	WR	CS	INT	SRQI	LOK	REM	CO	LOKC	REMC	ADSC
Serial Poll Status [3R]	0	1	1	WR	CS	S8	RSV	S6	S5	S4	S3	S2	S1
Address Status [4R]	1	0	0	WR	CS	CIC	ATN	EOI	TPAS	TPAS	LA	TA	MJMN
Command Pass Through [5R]	1	0	1	WR	CS	CPT7	CPT6	CPT5	CPT4	CPT3	CPT2	CPT1	CPT0
Address 0 [6R]	1	1	0	WR	CS	X	DT0	DL0	ADS0	ADM0	ADD0	ADD0	ADT0
Address 1 [7R]	1	1	1	WR	CS	EOI	DT1	DL1	ADS1	ADM1	ADD1	ADD1	ADT1
Data Out [0W]	0	0	0	WR	CS	BO7	BO6	BO5	BO4	BO3	BO2	BO1	BO0
Interrupt Mask 1 [1W]	0	0	1	WR	CS	CPT	APT	DET	END	DEC	ERR	DO	DI
Interrupt Mask 2 [2W]	0	1	0	WR	CS	0	SRQI	DMAO	DMAI	CO	LOKC	REMC	ADSC
Serial Poll Mode [3W]	0	1	1	WR	CS	S8	rsv	S6	S5	S4	S3	S2	S1
Address Mode [4W]	1	0	0	WR	CS	ton	lon	TAM1	TAM0	0	0	ADM1	ADM0
Auxiliary Mode [5W]	1	0	1	WR	CS	CNT2	CNT1	CNT0	COM4	COM3	COM2	COM1	COM0
Address 0/1 [6W]	1	1	0	WR	CS	ARS	DT	DL	ADS	ADM	ADD	ADD	ADT
End of String [7W]	1	1	1	WR	CS	EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0

DATA REGISTERS

The data registers are used for data and command transfers between the GPIB and the microcomputer system.

DATA IN (DR)

DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0

Holds data sent from the GPIB to the computer

BYTE OUT (DW)

DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0

Holds information written into it for transfer to the GPIB

INTERRUPT REGISTERS

The interrupt registers are composed of interrupt status bits, interrupt mask bits, and some other noninterrupt related status bits

READ

INTERRUPT STATUS 1 (IR1)	CPT	APT	DET	END	DEC	ERR	DO	DI
INTERRUPT STATUS 2 (IR2)	INT	SRQI	LOK	REM	CO	LOKC	REMC	ADSC

WRITE

INTERRUPT MASK 1 (IM1)	CPT	APT	DET	END	DEC	ERR	DO	DI
INTERRUPT MASK 2 (IM2)	0	SRQI	DMAO	DMAI	CO	LOKC	REMC	ADSC

There are thirteen factors which can generate an interrupt from the μPD7210, each with their own status bit and mask bit.

The interrupt status bits are always set to one if the interrupt condition is met. The interrupt mask bits decide whether the IIET bit and the interrupt pin will be active for that condition.

Interrupt Status Bits

INT	OR of All Unmasked Interrupt Status Bits
CPT	Command Pass Through
APT	Address Pass Through
DET	Device Trigger
END	End (END or EOS Message Received)
DEC	Device Clear
ERR	Error
DO	Data Out
DI	Data In
SRQI	Service Request Input
LOKC	Lockout Change
RENC	Remote Change
ADSC	Address Status Change
CO	Command Output

Non Interrupt Status Bits

LOK	Lockout
REM	Remote/Local
DMAO	Enable/Disable DMA Out
DMAI	Enable/Disable DMA In

SERIAL POLL REGISTERS

READ

SERIAL POLL STATUS (1R)	S8	PEND	S6	S5	S4	S3	S2	S1

WRITE

SERIAL POLL MODE (1W)	S8	RSV	S6	S5	S4	S3	S2	S1

The Serial Poll Mode register holds the STB (status byte: S8, S6-S1) sent over the GPIB and the local message rSV (request service). The Serial Poll Mode register may be read through the Serial Poll Status register. The PEND is set by rSV = 1, and cleared by NPRS + rSV = 1 (NPRS = Negative Poll Response State).

ADDRESS MODE/STATUS REGISTERS

ADDRESS STATUS (0R)	CIC	ATN	SPMS	LPAS	TPAS	LA	TA	MJMN
ADDRESS MODE (0W)	ton	ton	TRM1	TRM0	0	0	ADM1	ADM0

The Address Mode register selects the address mode of the device and also sets the mode for T/R3 and T/R2 the transceiver control lines.

The TLC is able to automatically detect two types of addresses which are held in address registers 0 and 1. The addressing modes are outlined below.

ADDRESS MODES

ton	ton	ADM1	ADM0	ADDRESS MODE	CONTENTS OF ADDRESS (0) REGISTER	CONTENTS OF ADDRESS (1) REGISTER
1	0	0	0	Talk only mode	Address identification not necessary	
0	1	0	0	Listen only mode	Not Used	
0	0	0	1	Address mode 1	Major talk address or Major listen address	Minor talk address or Minor listen address
0	0	1	0	Address mode 2	Primary address (talk or listen)	Secondary address (talk or listen)
0	0	1	1	Address mode 3	Primary address (major talk or major listen)	Primary address (minor talk or minor listen)
Combinations other than above indicated Prohibited						

Notes: A1 – Either MTA or MLA reception is indicated by coincidence of either address with the received address. Interface function T or L.

A2 – Address register 0 = primary, Address register 1 = secondary, Interface function TC or LC

A3 – CPU must read secondary address via Command Pass Through Register, TE or LC Command

ADDRESS STATUS BITS

$\overline{ATN}$	Data Transfer Cycle (device in CSBS)
LPAS	Listener Primary Addressed State
TPAS	Talker Primary Addressed State
CIC	Controller Active
LA	Listener Addressed
TA	Talker Addressed
MJMN	Sets minor T/L address. Reset = Major T/L address
SPMS	Serial Poll Mode State

ADDRESS REGISTERS

ADDRESS 0 (6A)	X	DY0	DL0	ADS0	AD4.0	AD3.0	AD2.0	AD1.0
ADDRESS 1 (7A)	EOI	DT1	DL1	ADS1	AD4.1	AD3.1	AD2.1	AD1.1
ADDRESS 0/1 (6W)	ARS	DT	DL	ADS	AD4	AD3	AD2	AD1

Address settings are made by writing into the address 0/1 register. The function of each bit is described below.

ADDRESS 0/1 REGISTER BIT SELECTIONS

ARS – Selects which address register 0 or 1
DT – Permits or Prohibits address to be detected at Talk
DL – Permits or Prohibits address to be detected at Listen
ADS – AD1 – Device address value
EOI – Holds the value of EOI line when data is received

COMMAND PASS THROUGH REGISTER

COMMAND PASS THROUGH (3R)	CPT7	CPT6	CPT5	CPT4	CPT3	CPT2	CPT1	CPT0

The CPT register is used such that the CPU may read the DIO lines in the case of undefined command, secondary address, or parallel poll response.

END OF STRING REGISTER

END OF STRING (7W)	EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0

This register holds either a 7 or 8 bit EOS message byte used in the GPIB system to detect the end of a data block. Aux Mode Register A controls the specific use of this register.

AUXILIARY MODE REGISTER

AUXILIARY MODE (5W)	CNT2	CNT1	CNT0	COM4	COM3	COM2	COM1	COM0

This is a multipurpose register. A write to this register generates one of the following operations according to the values of the CNT bits.

CNT 3	CNT 1	CNT 0	COM 4	COM 3	COM 2	COM 1	COM 0	OPERATION
0	0	0	C_4	C_3	C_2	C_1	C_0	Issues an auxiliary command specified by C_4 to C_0.
0	0	1	0	F_3	F_2	F_1	F_0	The reference clock frequency is specified and T_1, T_6, T_7, T_8 are determined as a result.
0	1	1	U	S	P_3	P_2	P_1	Makes write operation to the parallel poll register.
1	0	0	A_4	A_3	A_2	A_1	A_0	Makes write operation to the aux. (A) register.
1	0	1	B_4	B_3	B_2	B_1	B_0	Makes write operation to the aux. (B) register.
1	1	0	0	0	0	E_1	E_0	Makes write operation to the aux. (E) register.

AUXILIARY COMMANDS 0 0 0 C_4 C_3 C_2 C_1 C_0

COM 43210			
00000	iepon	–	Immediate Execute pon – Generate local pon Message
00010	crst	–	Chip Reset – Same as External Reset
00011	rtfd	–	Release RFD
00100	trig	–	Trigger
00101	rtl	–	Return to Local Message Generation
00110	seoi	–	Send EOI Message
00111	nvld	–	Non Valid (OSA reception) – Release DAC Holdoff
01111	vld	–	Valid (MSA reception, CPT, DEC, OET) – Release DAC Holdoff
0X001	rppf	–	Set/Reset Parallel Poll Flag
10000	gts	–	Go To Standby
10001	tca	–	Take Control Asynchronously
10010	tcs	–	Take Control Synchronously
11010	tcse	–	Take Control Synchronously on End
10011	lln	–	Listen
11011	llnc	–	Listen with Continuous Mode
11100	lun	–	Local Unlisten
11101	epp	–	Execute Parallel Poll
1X110	sifc	–	Set/Reset IFC
1X111	sren	–	Set/Reset REN
10100	dsc	–	Disable System Control

INTERNAL COUNTER 0 0 1 0 F_3 F_2 F_1 F_0

The internal counter generates the state change prohibit times (T_1, T_6, T_7, T_8) specified in the IEEE std 488-1978 with reference to the clock frequency.

AUXILIARY A REGISTER 1 0 0 A_4 A_3 A_2 A_1 A_0

Of the 5 bits that may be specified as part of its access word, two bits control the GPIB data receiving modes of the 7210 and 3 bits control how the EOS message is used.

A_1	A_0	DATA RECEIVING MODE
0	0	Normal Handshake Mode
0	1	RFD Holdoff on all Data Mode
1	0	RFD Holdoff on End Mode
1	1	Continuous Mode

BIT NAME			FUNCTION
A_2	0	Prohibit	Permits (prohibits) the setting of the END bit by reception of the EOS message
	1	Permit	
A_3	0	Prohibit	Permits (prohibits) automatic transmission of END message simultaneously with the transmission of EOS message TACS.
	1	Permit	
A_4	0	7 bit EOS	Makes the 8 bits/7 bits of EOS register the valid EOS message.
	1	8 bit EOS	

AUXILIARY B REGISTER 1 0 1 B_4 B_3 B_2 B_1 B_0

The Auxiliary B Register is much like the A Register in that it controls the special operating features of the device.

BIT NAME			FUNCTION
B_0	1	Permit	Permits (prohibits) the detection of undefined command. In other words, it permits (prohibits) the setting of the CPT bit on reception of an undefined command.
	0	Prohibit	
B_1	1	Permit	Permits (prohibits) the transmission of the END message when in serial poll active state (SPAS).
	0	Prohibit	
B_2	1	T_1 (high speed)	T_1 (high speed) as T_1 of handshake after transmission of 2nd byte following data transmission.
	0	T_1 (low speed)	
B_3	1	INT	Specifies the active level of INT pin.
	0	$\overline{INT}$	
B_4	1	1st = SRQS	SRQS indicates the value of 1st level local message (the value of the parallel poll flag is ignored). SRQS = 1 . . . 1st = 1 SRQS = 0 . . . 1st = 0
	0	1st = Parallel Poll Flag	The value of the parallel poll flag is taken as the 1st local message.

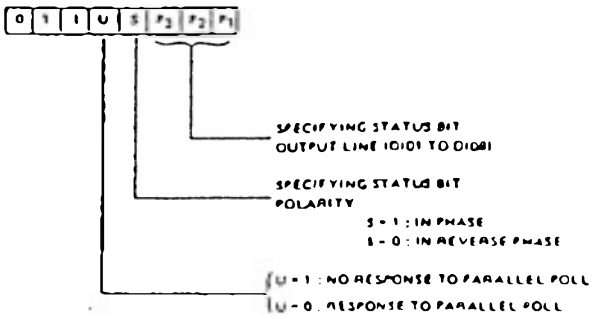
AUXILIARY E REGISTER 1 1 0 0 0 0 E_1 E_0

This register controls the Data Acceptance Modes of the TLC.

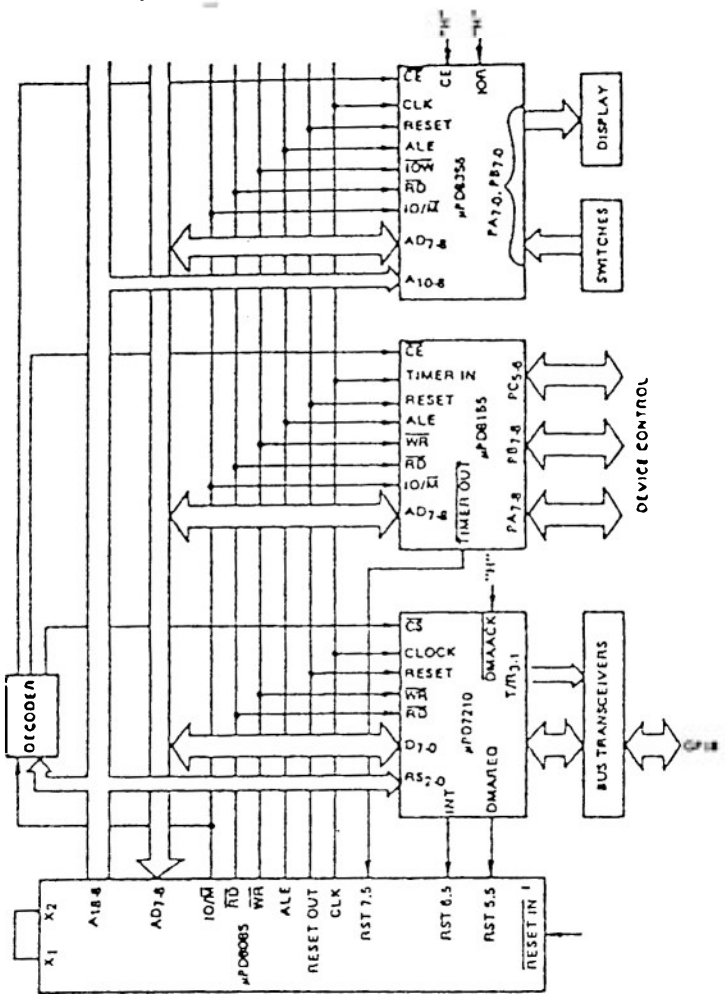
BIT	FUNCTION		
E_0	1	Enable	DAC Holdoff by Initiation of DCAS
	0	Disable	
E_1	1	Enable	DAC Holdoff by Initiation of DTAS
	0	Disable	

Parallel Poll Register 0 1 1 U S P_3 P_2 P_1

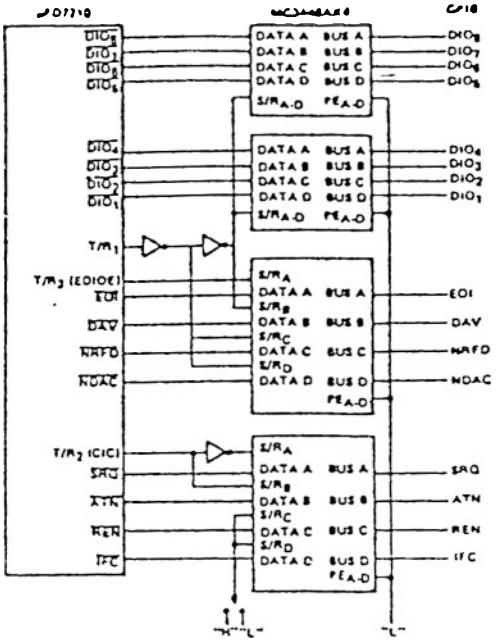
The Parallel Poll Register defines the parallel poll response of the μPD7210.



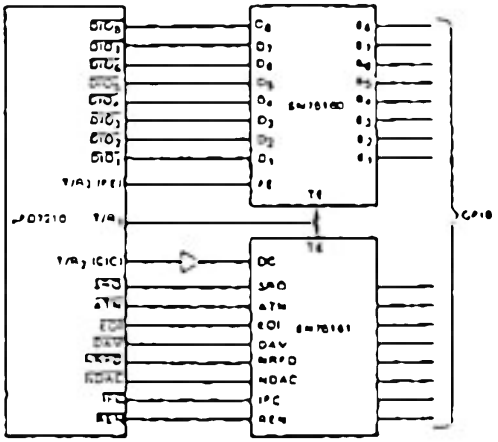
MINIMUM 8085 SYSTEM
WITH μPD7210



MINIMUM 8085 SYSTEM
WITH μPD7210 (CONT.)



Note: In this example, high-speed data transfer cannot be made since the bus transceiver is of the open collector type (Set B_2 = 0).



Note: In the case of low-speed data transfer (B_2 = 0), the T/$\overline{R}_3$ pin can be used as a TRIG output. The PE input of SN75160 should be cleared to "0." Using this connection, pass control operation cannot be used.

ABSOLUTE MAXIMUM RATINGS

$T_a = 25°C$

PARAMETER	SYMBOL	TEST CONDITIONS	RATINGS	UNITS
Supply Voltage	V_{CC}		−0.5 ~ +7.0	V
Input Voltage	V_I		−0.5 ~ +7.0	V
Output Voltage	V_O		−0.5 ~ +7.0	V
Operating Temperature	T_{opt}		0 ~ +70	°C
Storage Temperature	T_{stg}		−65 ~ +150	°C

DC CHARACTERISTICS

$T_a = 0 \sim +70°C$, $V_{CC} = +5V \pm 10\%$

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS
Input Low Voltage	V_{IL}		−0.5		+0.8	V
Input High Voltage	V_{IH}		+2.0		V_{CC} +0.5	V
Low Level Output Voltage	V_{OL}	I_{OL} = 2mA (4mA: T/R1 Pin)			+0.45	V
High Level Output Voltage	V_{OH1}	I_{OH} = −400µA, [illegible]	+2.4			V
	V_{OH2}	I_{OH} = −400µA (T/R1 Pin)	+2.4			V
		I_{OH} = −100µA	+3.5			V
Input Leakage Current	I_{IL}	V_{IN} = 0V ~ V_{CC}	−10		+10	µA
Output Leakage Current	I_{OL}	V_{OUT} = 0.45V ~ V_{CC}	−10		+10	µA
Supply Current	I_{CC}				+180	mA

CAPACITANCE

$T_a = 25°C$, $V_{CC} = GND = 0V$

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS
Input Capacitance	C_{IN}	f = 1 MHz, All Pins Except Pin Under Test Tied to AC Ground			10	pF
Output Capacitance	C_{OUT}				15	pF
I/O Capacitance	$C_{I/O}$				20	pF

AC CHARACTERISTICS

$T_a = 0 \sim +70°C$, $V_{CC} = +5V \pm 10\%$

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS
Address Setup to RD	t_{AR}	RS2-0	85			ns
		CS	0			
Address Hold from RD	t_{RA}		0			ns
RD Pulse Width	t_{RR}		170			ns
Data Delay from Address	t_{AD}				250	ns
Data Delay from RD ↓	t_{RD}				150	ns
Output Float Delay from RD ↑	t_{DF}		0		85	ns
RD Recovery Time	t_{RV}		250			ns

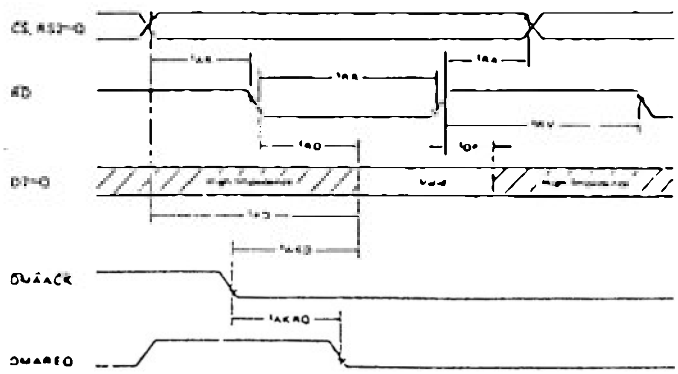
Address Setup to WR	t_{AW}		0			ns
Address Hold from WR	t_{WA}		0			ns
WR Pulse Width	t_{WW}		170			ns
Data Setup to WR	t_{DW}		150			ns
Data Hold from WR	t_{WD}		0			ns
WR Recovery Time	t_{RV}		250			ns

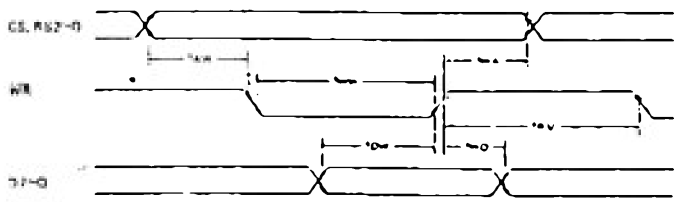
DMAREQ ↓ Delay from DMAACK	t_{AKRQ}				130	ns
Data Delay from DMAACK	t_{AKD}				200	ns

AC CHARACTERISTICS

[illegible]

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS MIN	TYP	MAX	UNITS
[illegible]	[illegible]	[illegible]			250	ns
[illegible]	[illegible]	[illegible]			150	ns
[illegible]	[illegible]	[illegible]			200	ns
[illegible]	[illegible]	[illegible]			150	ns
[illegible]	[illegible]	[illegible]			150	ns
[illegible]	[illegible]	[illegible]			200	ns
[illegible]	[illegible]	[illegible]			400	ns
[illegible]	[illegible]	[illegible]			250	ns
[illegible]	[illegible]	[illegible]			650	ns
[illegible]	[illegible]	[illegible]			350	ns
[illegible]	[illegible]	[illegible]			250	ns
[illegible]	[illegible]	[illegible]			600	ns
[illegible]	[illegible]	[illegible]			400	ns
[illegible]	[illegible]	[illegible]			[illegible]	ns
[illegible]	[illegible]	[illegible]			250	ns
[illegible]	[illegible]	[illegible]			250	ns
[illegible]	[illegible]	[illegible]			650 [illegible]	ns
TRIG Pulse Width	[illegible]		50			ns





NEC cannot assume any responsibility for any circuits shown or represent that they are free from patent infringement.
NEC reserves the right to make changes any time without notice in order to improve design and supply the best product possible.

Dec. 1981

Preliminary Specification

****
****

NCR DECISION MATE V DIAGNOSTICS

****
****

1.0 GENERAL:

Level 0 and Level 1 Diagnostics for Decision Mate V

Level 0: -integrated on Main Board
-checking of basic microprocessor and controllers functions
-error messages on LED row on the back rear
-executed after Power On or Reset

Level 1: -plugable box connected to system bus
-Power Supply function controlled
-tests selectable by switches or keyboard inputs
-Level 1 ROM resitend on the box
-error messages on CRT or two 7-segment displays

1.1 Usage Intention

Installed and used by field enginering and by customer
Easy understandable error messages
Test of modules and boards down to the lowest replaceable part

To perform the diagnostic, the module will be inserted in slot 7 on the rear of the cabinet.

2.0 LEVEL 0 Diagnostics

Started after each power on or reset with a general test of all components:

- Processor
- ROM check
- RAM test
- Keyboard
- GDC Controller
- DMA Controller
- Flex Disk Controller

2.1 Description of Level 0 Tests

Level 0 Diagnostics

1. Processor Test

2. Firmware Sum Check Test

3. Memory Test
   Write/Read test of RAM locations 0000H - FFFFH with pattern 55/AA.

4. Keyboard Processor (8041) Test
   Self test of processor and checking of possible country code of keyboard.

5. CRT Controller Test

6. DMA Controller (8237) Test
   Write/Read test of registers is performed.

7. Flex Disk Controller (8272) Test
   Read Main Status of flex Disk Controller and if status is no 80H test failed

2.2 Level 0 Error Codes

If an error is detected the program stops and the LED row on the rear side shows the error status:

"x" indicates a burning LED

LED number:	8	7	6	5	4	3	2	1	
	x	x	x	x	x	x	x	x	Processor
	x							x	ROM Sum Check Error
	x						x		CRT Controller
	x					x			Flex Disk Controller
	x				x				not used
	x			x					Keyboard Error
	x		x						DMA Controller Error
	x	x							Memory Error

3.0 LEVEL 1 Diagnostics:

3.1 Hardware Scope

- 8 k ROM
- 2 k RAM
- Timer
- port for switches and 7 segment display
- Memory select logic
- LEDs for display of running
- reset switch

3.1.1 Running LED's

- Voltage indicator
  5 Volts over under and correct voltage
  12 Volts over under and correct voltage

- MEMR/ Memory Read indicator - when it is "on" something going on
- PCLK Processor Clock - when it is "on" the processor clock runs
  it is no indication of right clock frequ
- HOLDA Holdacknowledge indicator - when it is "on" the processor
  is not in HOLD, it can work

All these green LED's must burn when the board is running.

3.1.2. Memory Select Logic:

As the entire 64k Memory is occupied by the user Ram, a select logic must share areas which are also used by the diagnostic firmware

Switch logic of shared memory with two port lines:

PC 1	PC 0		
0	0	disable diag ROM	disable diag RAM
0	1	enable diag ROM	disable diag RAM
1	0	disable diag ROM	enable diag RAM
1	1	enable diag ROM	enable diag RAM

3.1.3 ROM/RAM

8k ROM 2 * 2732
2k RAM 1 * 6116

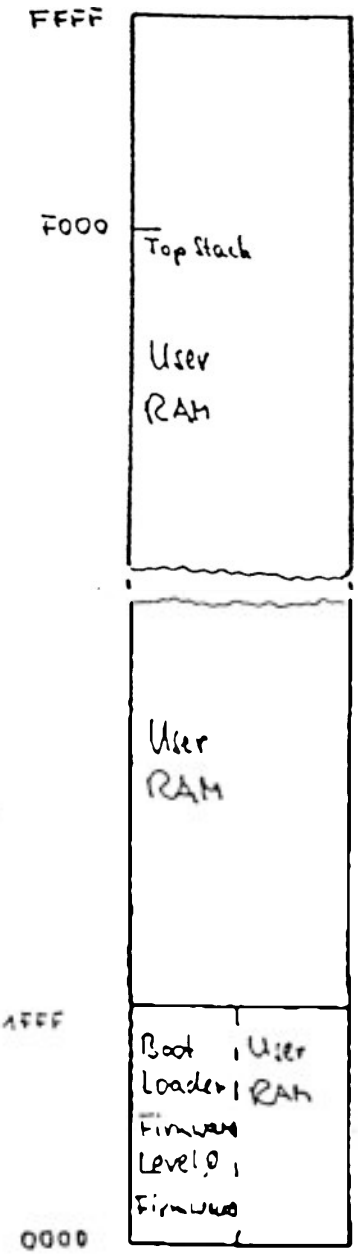
3.1.4. Timer (8253)

For interrupts (running into an endless loop)
Measuring of timing

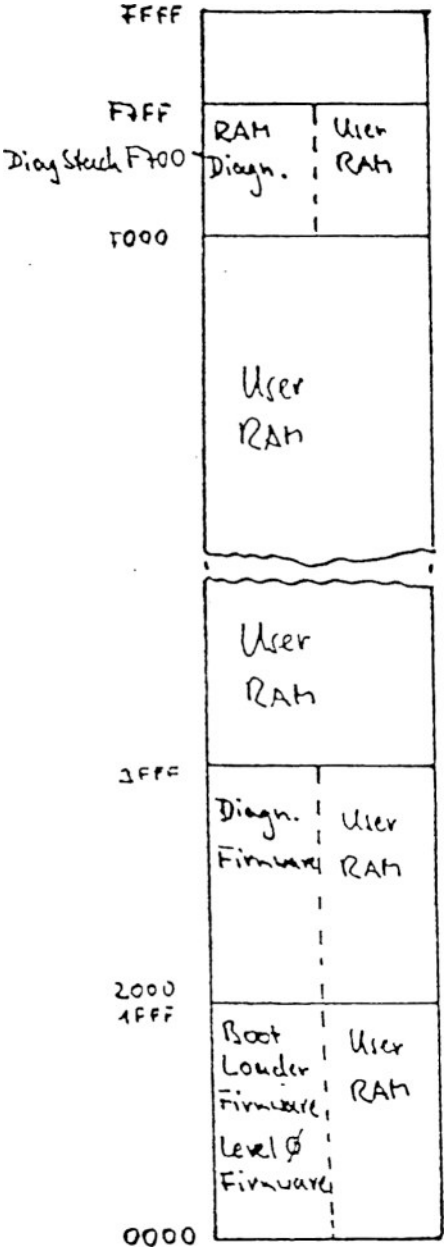
3.1.5.Ports Using (8255)

8 test selection switches
7 segment displays
Memory select logic

3.1.6. Memory Sharing in Diagnostic



Memory Sharing DMV

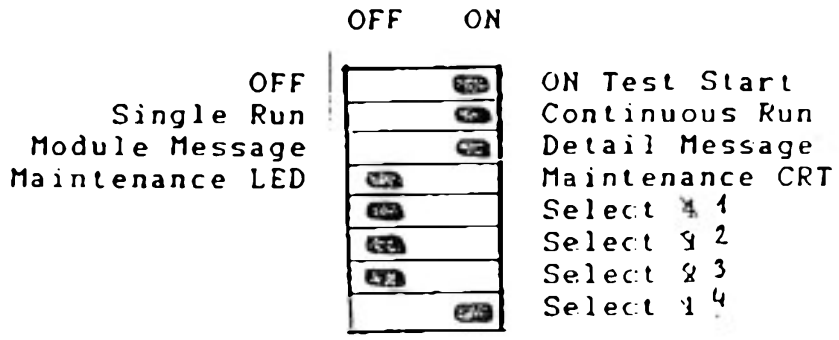


Memory Sharing Diagnostic DMV

3.2. Functions of Diagnostic Box

Requirement for a sucessful test start:
No fault in the Processor and Address-Data bus

3.2.1.1. Function and test switch select



3.2.1.2. Switch Setting of Select 1 to Select 4

Test # in Maintenance Mode	Select Switch 4	3	2	1	Test Name
-	0	0	0	0	Self Configuration Test
1	0	0	0	1	DMA Controller Test
2	0	0	1	0	CRT Controller Test
3	0	0	1	1	Disk Controller Test
4	0	1	0	0	Keyboard Controller Test
5	0	1	0	1	CRT Test
6	0	1	1	0	Disk Drive A Test
7	0	1	1	1	Disk Drive B Test
8	1	0	0	0	Keyboard Test
9	1	0	0	1	Memory Test
A	1	0	1	0	Main Board Test
B	1	0	1	1	Disk Drive Alignment
C	1	1	0	0	not used
D	1	1	0	1	not used
E	1	1	1	0	not used
F	1	1	1	1	not used

- ON/OFF Switch

  OFF - diagnostic box not activ

  ON - run diagnostic

- Continous Run

  OFF - selected diagnostic test passes only one time then stops and displays the error code or 99 for no errors

  ON - the test is running as long as this switch is on or an error is detected

  Switch is only activ if Maintenance is OFF

- Detail Message

  OFF - the shown error code on the 7 segment display is only a general error code, pointing to a failed module

  ON - detailed error code will enable an educated user to isolate the trouble to the lowest field replaceable part

  Switch is only activ if Maintenance is OFF.

- Maintenance Switch

  OFF - test are selected by switches
  error messages shown on 7 segment displays
  only the selected test displayed on CRT

  ON - a test menue is shown on CRT, and the tests are selectable by keyboard
  error messages on CRT the 7 segment display shows 00.

- Select switches 1..4

  these switches select the specified module test in Maintenance Off they are binary coded

4.0 Description of Level 1 Tests

After entering Level 1 diagnostic a Sum Check of Diagnostic ROM's is done.

0. Self Configuration Test
   The Self Configuration Test execution enclose several tests:
   - Main Board Test
   - Keyboard Processor Test
   - CRT Test
   In a later version it shall also test the entire system configuration with connected interfaces, RAM extensions or 16-Bit extension.

1. DMA Controller Test
   A register Read/Write test with different bit pattern is performed and if the bit pattern does not match, an error code is displayed.

2. CRT Controller Test
   Write/Read of the Graphic RAM with pattern 55/AA,AA/55,00/FF and FF/00. The pattern is displayed on the screen.

3. Disk Controller Test
   An invalid command is sent to the Disk Controller and the status register is checked.

4. Keyboard Controller Test
   A self check command is send to the keyboard controller on the main board and the return status is checked for error.

5. CRT Test
   Display some pictures on the screen
   - Cursor Movement draws a square on the screen
   - Full screen display with character "E"
   - Full screen display with the whole character set (00-7F Hex). each picture is seperated by a key input

6. Disk Drive A Test
   To check the drive a scratch disk must be inserted.
   - Restore function (position to track #0)
   - Format track (last track #27 Hex is formatted)
   - Seek function (seek track #27 Hex)
   - Write data to disk (track #27 Hex is written)
   - Read data from disk and compare (read track #27 Hex)

7. Disk Drive B Test
   The same as for Drive A.

8. Keyboard Test
Performs first the Keyboard Controller Test (Test #4)
Next the language code is read and displayed.
In the Maintenance On Mode you can additionally test each key typed in, which is displayed on the CRT (including sound of tone).

9. Memory Test
The Memory is checked with 55/AA,AA/55,00/FF and FF/00.
Directly folowing a Memory Address Decode Test is performed.
The memory address is written into the addressed memory location.
Processor is set into HALT to wait for automatic refresh from dynamic RAM controller.
After one second, all memory locations are read and verified with the written values.

A. Main Board Test
It runs the entire set of component tests on the Main Board
- Run Level 0 Diagnostic
- Main Board LED Test
- Several Tests described before
  DMA Controller
  CRT controller
  Disk Controller
  Keyboard Controller
  Memory Test

B. Disk Alignment
After a Restore on the selected Drive, a continuous Read of Track 16 is performed. Stop the test by entering any key on the keyboard.
A special alignment Disk is necessary for this test.

5.0 Error Codes and Error Messages

5.1. Error codes

10 - Main Board

11 - Level 0 Diagnostic Error
12 - DMA Controller Error
13 - Disk Controller Error
14 - Keyboard Controller Error
15 - CRT Controller Error (GDC Graphic Display Controller)

20 - Memory Address Error
21 - Memory Bit 1 Error
to
28 - Memory Bit 8 Error

30 - Disk Drive Error

31 - Recalibrate Error
32 - Disk Format Error
33 - Read ID Error
34 - Write Data Error
35 - Read Data Error
36 - Write/Read Data Compare Error

50 - Keyboard Error

51 - Keyboard not connected
52 - Keyboard Processor Error

90 - Diagnostic Box Sum Check Error

99 - Test passed OK Processor stops

5.2. Error Messages on CRT Screen

In the Maintenance On Mode all error messages and codes are displayed on the screen. This is a listing and short description of these messages.

0. General Messages

Level 0 failed

```
ERROR CODE = 11
LEVEL 0 DIAGNOSTICS ERROR STATUS = x x x x x x x x
```

x = 0 or 1 bit pattern of Level 0 LED's (see 2.2.)

Level 1 ROM check error

```
ERROR CODE = 90
ROM SUM CHECK = xx
```

xx = Another Sumcheck than 00 is a error

1. DMA Controller Test

```
ERROR CODE = 12
DMA CONTROLLER ERROR ON CHANNEL  n
PORT     EXP  OBS
ADDR      VALUE
 ii       xx   xx
```

n = 0..3
ii = Portaddress 20...27
xx = Data

2. CRT Controller Test

```
ERROR CODE = 18
GDC RAM WRITE/READ ERRROR
Graph.RAM ADDR. = iiii
EXP.VALUE = xx
OBS.VALUE = xx
```

iiii = Graphic RAM address 0...3840 Hex
xx = Data

3. Disk Controller Test

```
ERROR CODE = 13
FLEX DISK CONTROLLER ERROR
STATUS 0 = xx
```

xx = Another Status 0 than 80H are errors

4. Keyboard Controller Test

```
ERROR CODE = 14
KEYBORD CONTROLLER ERROR
SELFCHECK STATUS = xx
```

xx = Another status than 55H is an error

5. CRT Test

This phase has no messages. You see only the drawn pictures.

6. Disk Drive A Test

1. Recalibrate Test

```
ERROR CODE = 31
 RECALIBRATE ERROR
 STATUS 0 = aa
 PRESENT CYL.(HEX) = nn
```

aa =

nn = Present Track

2.Format Error

```
ERROR CODE = 32
 FORMAT ERROR
 DISK STATUS VALUES
 ST0 ST1 ST2 C   H   R   N
 xx  xx  xx  xx  xx  xx  xx
```

ST0 = Status Register 0
ST1 = Status Register 1
ST2 = Status Register 2
C = Current selected track number 0-27 Hex
= Head number 0 or 1
R = Sector number which is read or written
N = Number of data byte written on a sector

3.Read ID Error

```
ERROR CODE = 33
 READ ID ERROR
 DISK STATUS VALUES:
 ST0 ST1 ST2 C   H   R   N
 xx  xx  xx  xx  xx  xx  xx
```

```
ERROR CODE = 34
 URITE ERROR
 DISK STATUS VALUES:
 ST0 ST1 ST2 C   H   R   N
 xx  xx  xx  xx  xx  xx  xx
```

5.Read Data and Compare Error

```
ERROR CODE = 35
 READ ERROR
 DISK STATUS VALUES:
 ST0 ST1 ST2 C   H   R   N
 xx  xx  xx  xx  xx  xx  xx


ERROR CODE = 36
 READ DATA COMPARE ERROR
```

Disk Drive B Test

The error messages are the same as for drive A.

8. Keyboard Test

At this time can also appear an error from Keyboard Controller

```
ERROR CODE = 51
KEYBOARD NOT CONNECTED


ERROR CODE = 52
KEYBOARD PROCESSOR ERROR
```

Memory Test

```
ERROR CODE = 20
 MEMORY ADDRESS ERROR
 EXP. ADDR. = iiii
 OBS. ADDR. = iiii


ERROR CODE = aa
MEMORY ERROR ON BIT n
 ADDR.    EXP/OBS VALUE
 iiii      xx  xx
```

aa = 21...28 corresponds with the error bit position
n = Bit position 1..8
iiii = RAM Address from 0 to FFFFH
xx = Data

8. Main Board Test

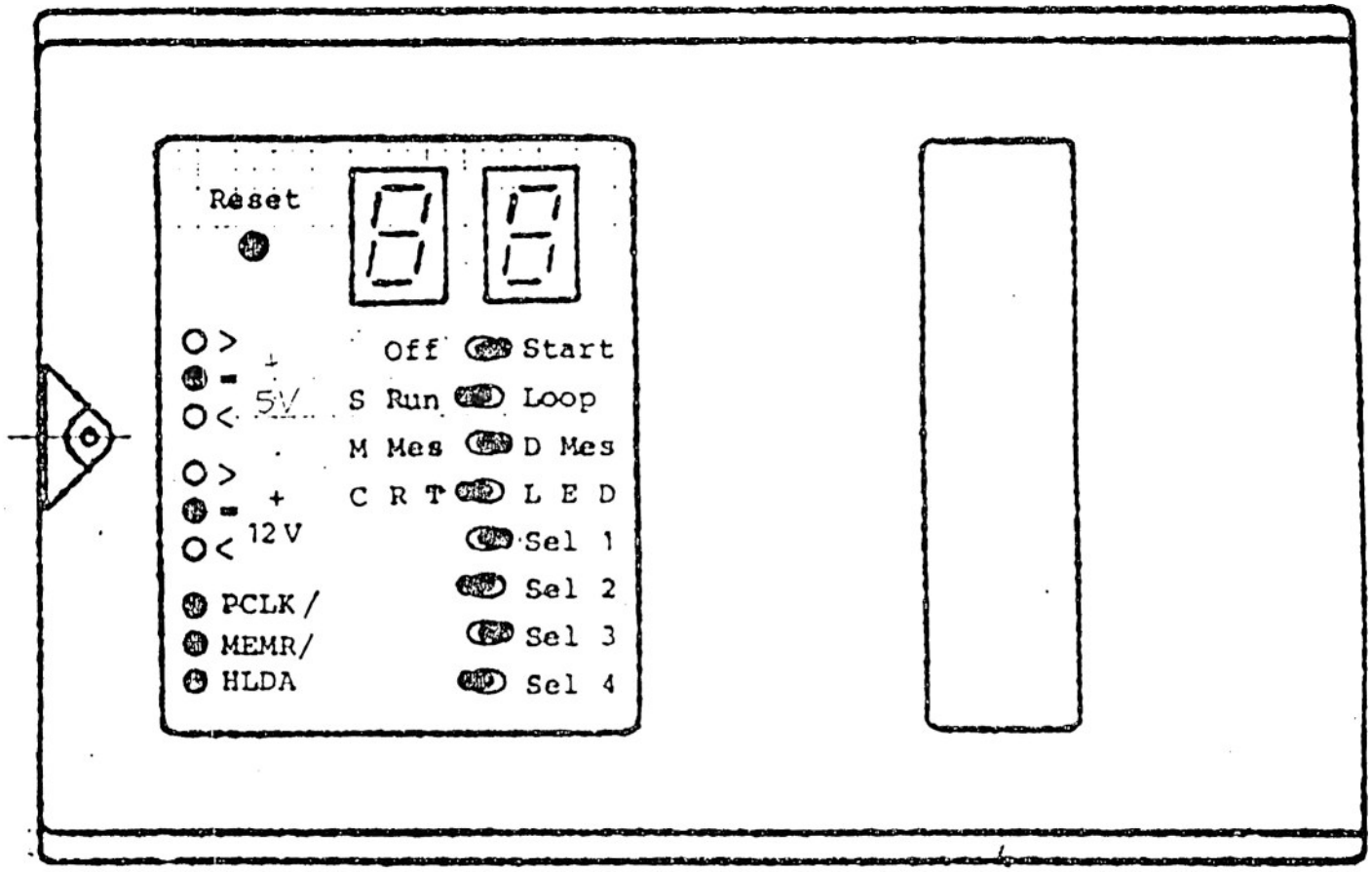
In this phase can appear all messages from
- Level 0
- Memory
- DMA controller
- Disk Controller
- Keyboard
- GDC RAM Test

additionally a Lamp Test of the level 0 LED's is performed
- all Lamps on
- all Lamps off
- each Lamp seperately on

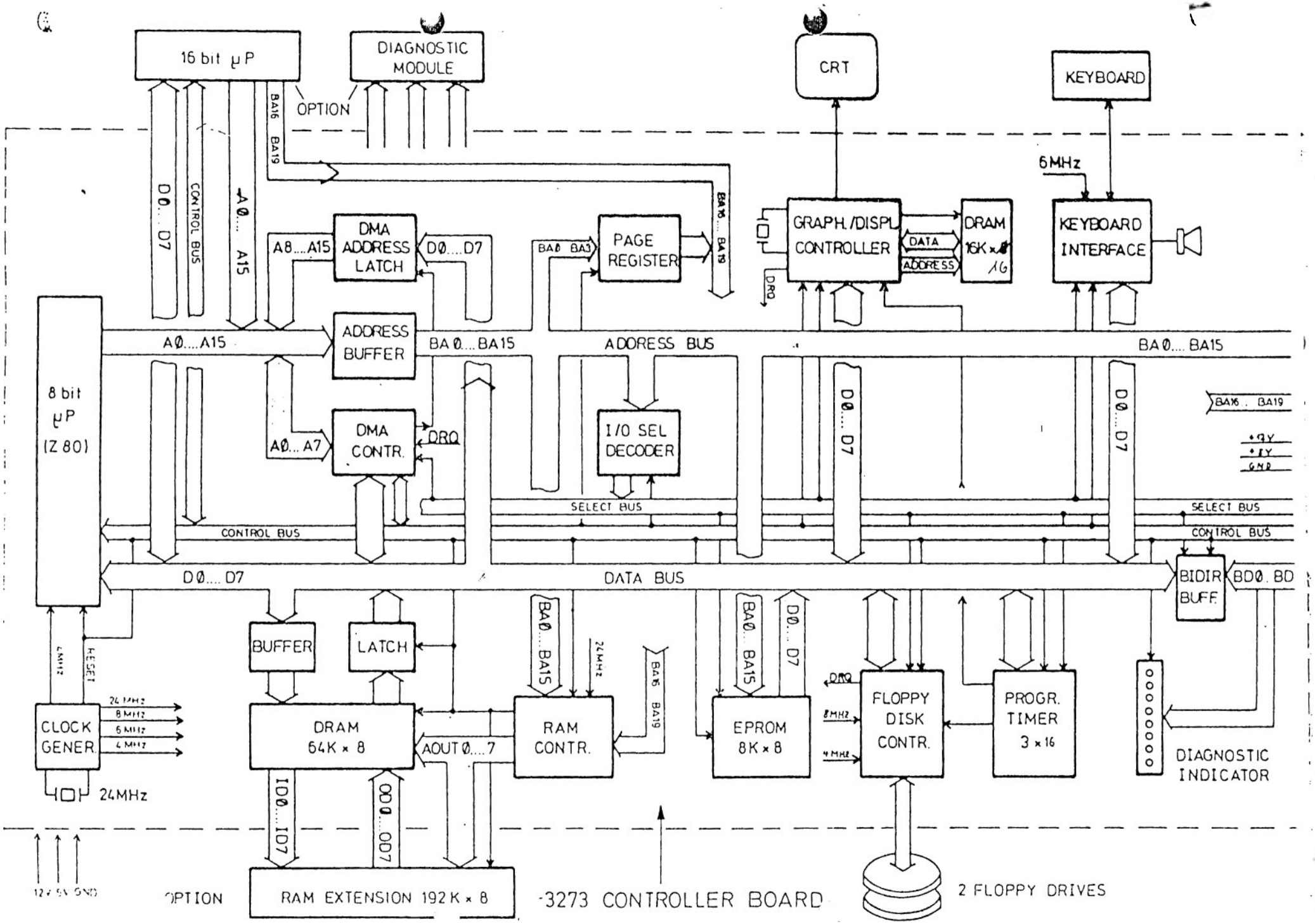
between these phases is always 0,5 second delay

9. Disk Alignment Test

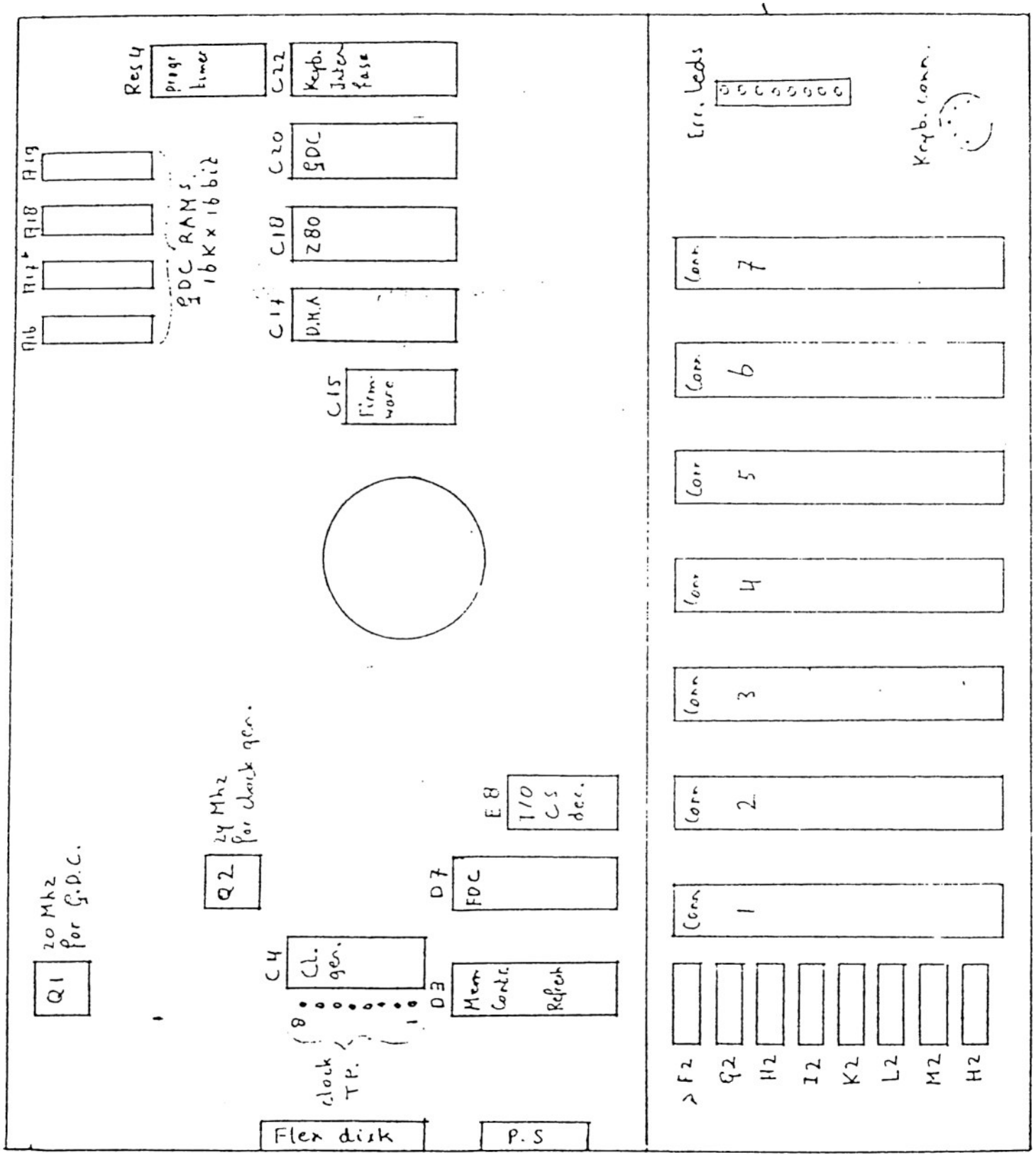
No Error Message is displayed.


Reset
+
5V
-
+
12V
-
PCLK /
MEMR/
HLDA
Off
Start
S Run
Loop
M Mes
D Mes
C R T
L E D
Sel 1
Sel 2
Sel 3
Sel 4

13.10.82


16 bit µP
DIAGNOSTIC MODULE
OPTION
CRT
KEYBOARD
6MHz
D0.... D7
CONTROL BUS
A0.... A15
BA16 BA19
DMA ADDRESS LATCH
A8....A15
D0....D7
PAGE REGISTER
BA8 BA3
BA16.... BA19
GRAPH./DISPL. CONTROLLER
DATA
ADDRESS
DRQ
DRAM 16K x 8
16
KEYBOARD INTERFACE
8 bit µP (Z 80)
A0....A15
ADDRESS BUFFER
BA0....BA15
ADDRESS BUS
BA0.... BA15
DMA CONTR.
A0... A7
DRQ
I/O SEL DECODER
SELECT BUS
CONTROL BUS
D0.... D7
DATA BUS
BA16 . BA19
+5V
+12V
GND
BIDIR BUFF.
BD0. BD
BUFFER
LATCH
BA0....BA15
24MHz
D0.... D7
DRQ
8MHz
4MHz
FLOPPY DISK CONTR.
PROGR. TIMER 3 x 16
DIAGNOSTIC INDICATOR
CLOCK GENER.
24MHz
8MHz
6MHz
4MHz
RESET
DRAM 64K x 8
AOUT0....7
RAM CONTR.
EPROM 8K x 8
ID0....ID7
OD0....OD7
12V 5V GND
OPTION
RAM EXTENSION 192K x 8
3273 CONTROLLER BOARD
2 FLOPPY DRIVES

CONTROLLER BOARD.



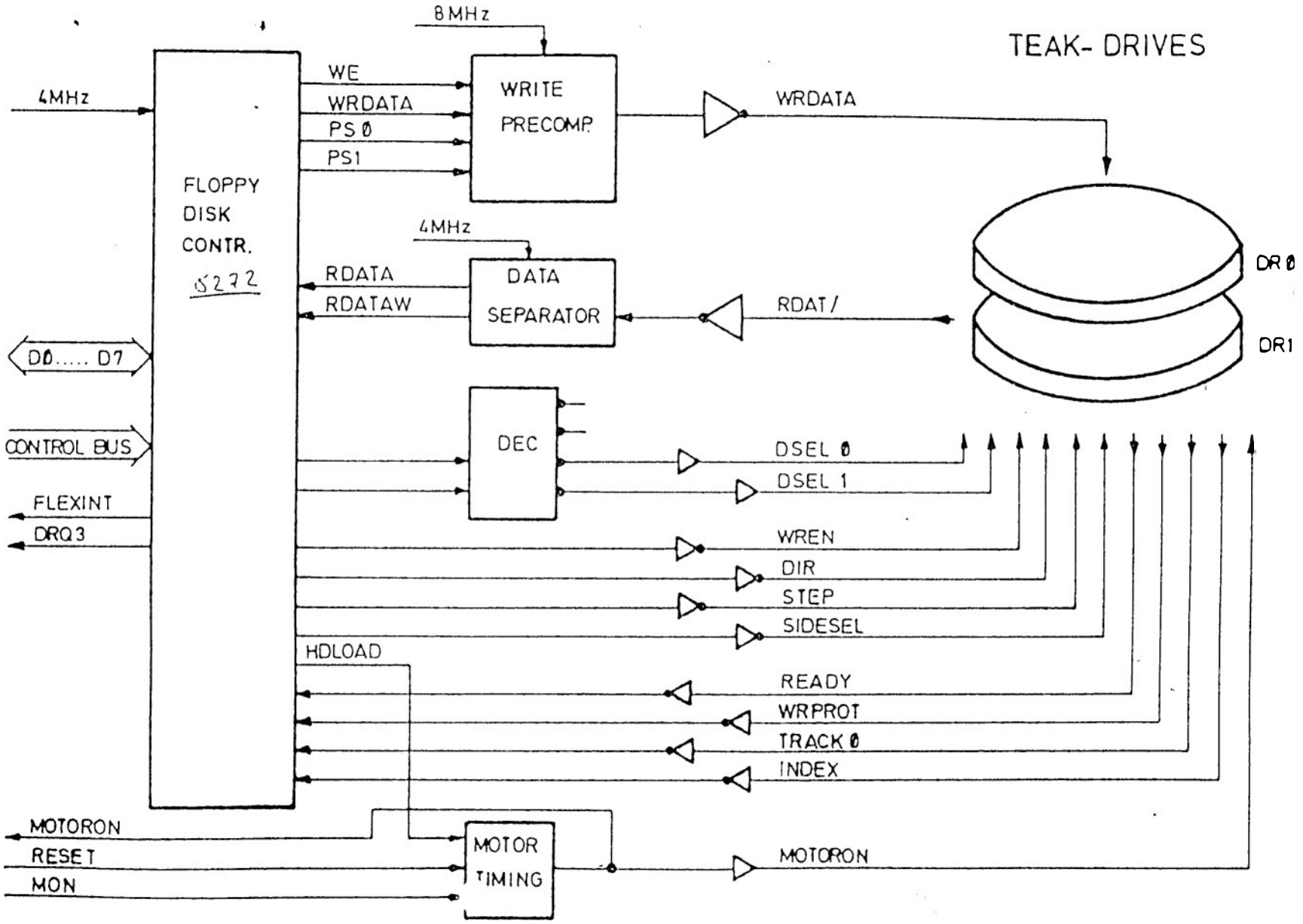
1 Kom. 16 K x 4 bit.

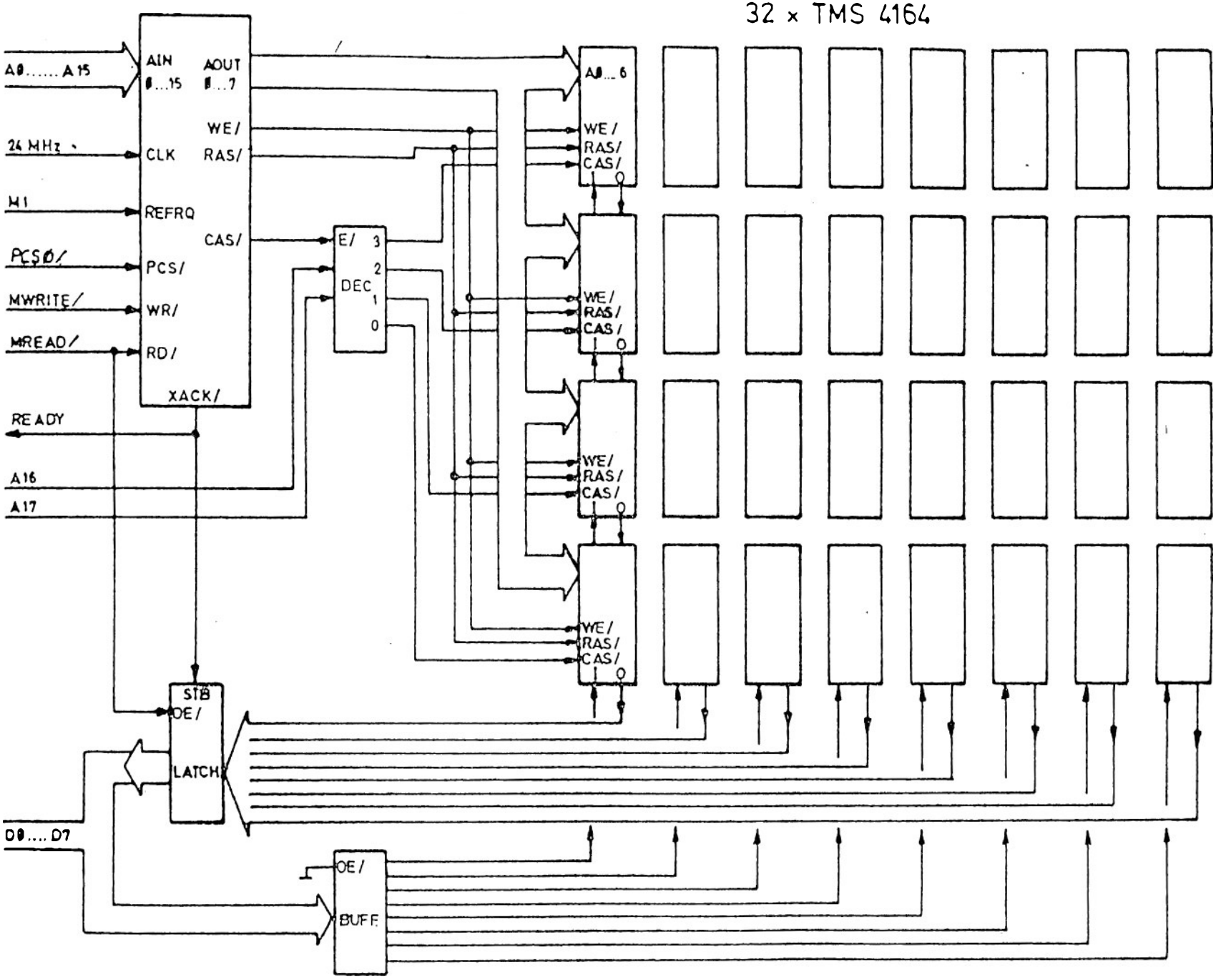
16K x 1 bit

1 K byte.

BLOCKDIAGRAM

FLOPPY DISK CONTROLLER



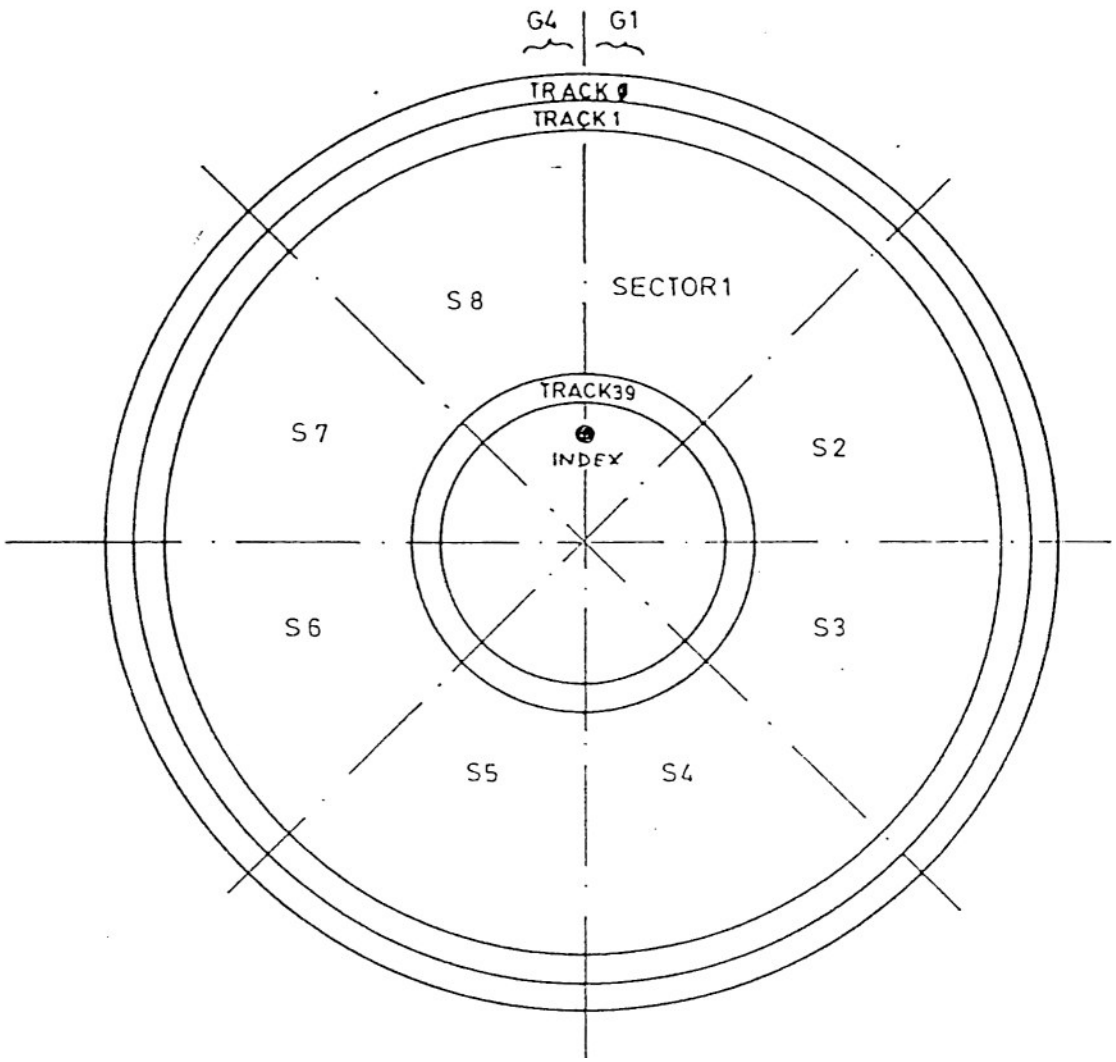


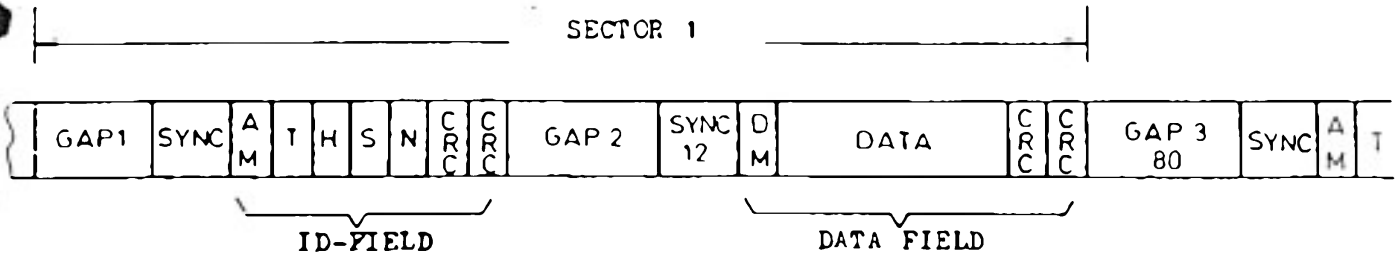
BLOCKDIAGRAM

256K x 8 DYNAMIC RAM

64 k x 8 B

DOUBLE DENSITY FLOPPY DISK FORMAT (IBM SYSTEM 34)





T - TRACK NR. ØØ 27H
H - HEAD NR. Ø or 1
S - SECTOR NR. 1 8
N - NR. OF DBYTES 02 - 512

} WRITTEN BY COMMAND : "FORMAT TRACK"

GAP'S
SYNC'S
AM (ADDRESS MARK)
DM (DATA MARK)

} Supplied automatically by FD-Controller

DM5 KEYBOARD

DM5 C-3273
Serial interface polling technique:
The KB-IF within C-3273 sends every 8 ms an active low signal on data line with 25 µs duration. If KB has a datacode in its FIFO it will send back 8 bit KB data code LSB first ,MSB last.

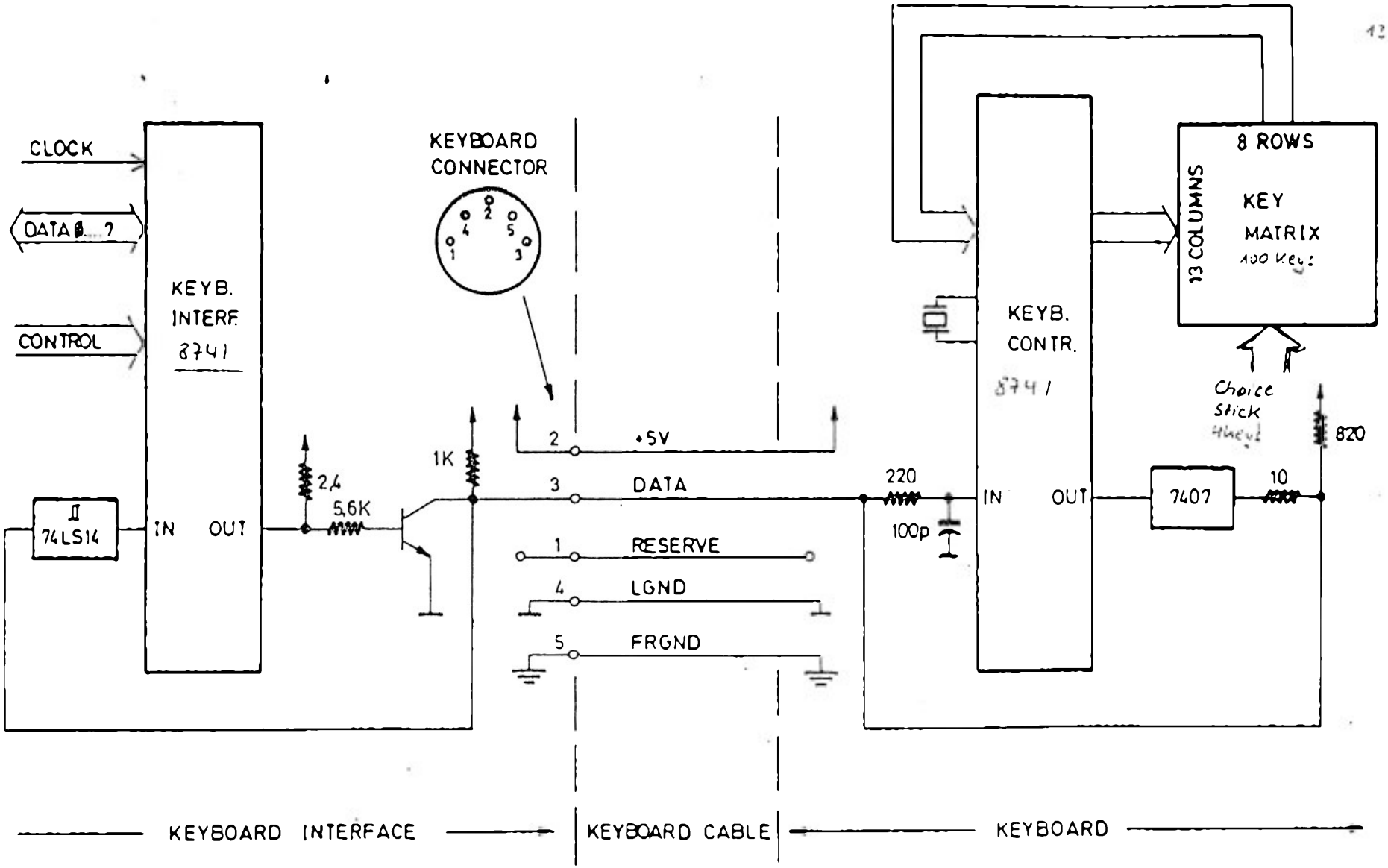
Each bit cycle has 200 µs duration.
A "0" data bit holds dataline for 25 µs active low,
A "1" data bit holds dataline for 100 µs active low.
Remaining time of bit cycle time the line is pulled up.
Send procedure starts with 300 µs delay after polling cycle.
Receiving a 2nd polling cycle within this delay time means that the host processor requests from KH the KB-status and language code strapping info.
The KB-FIFO is not matched with this action.

Switch 1 Language Code Setting

Country	Hex	S1/1	S1/2	S1/3
US-English	0	off	off	off
UK-English	1	on	off	off
French	2	off	on	off
German	3	on	on	off
Swedish/Finnish	4	off	off	on
Danish/Norwegian	5	on	off	on
Spanish	6	off	on	on
Italian	7	on	on	on

BLOCKDIAGRAM

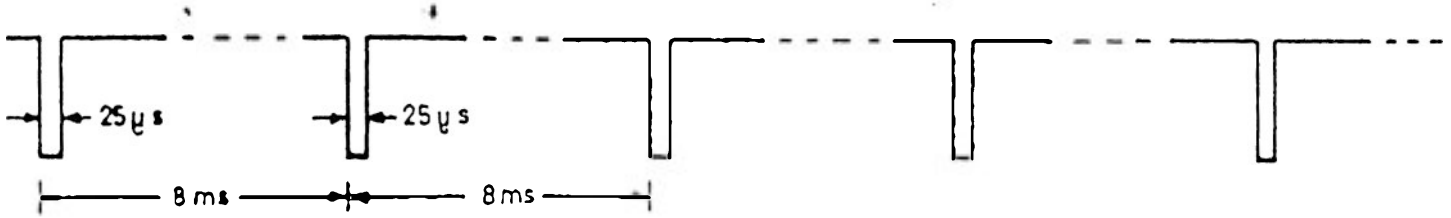
KEYBOARD INTERF. AND KEYBOARD



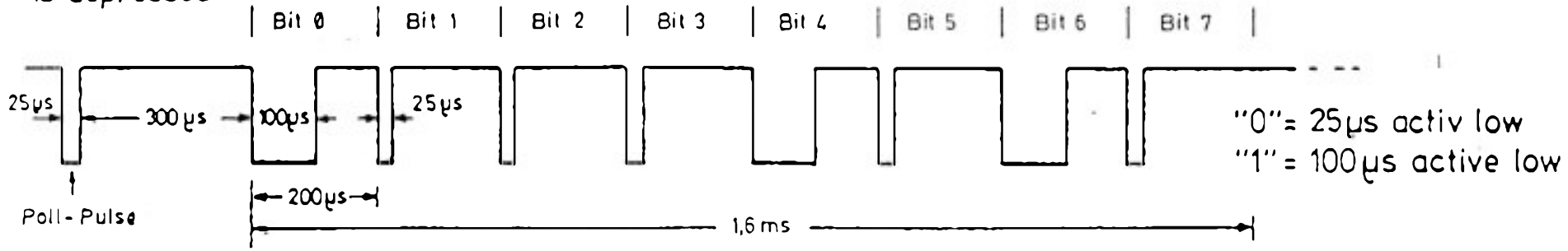
TIMING DIAGRAM

KEYBOARD POLLING AND DATA TRANSFER

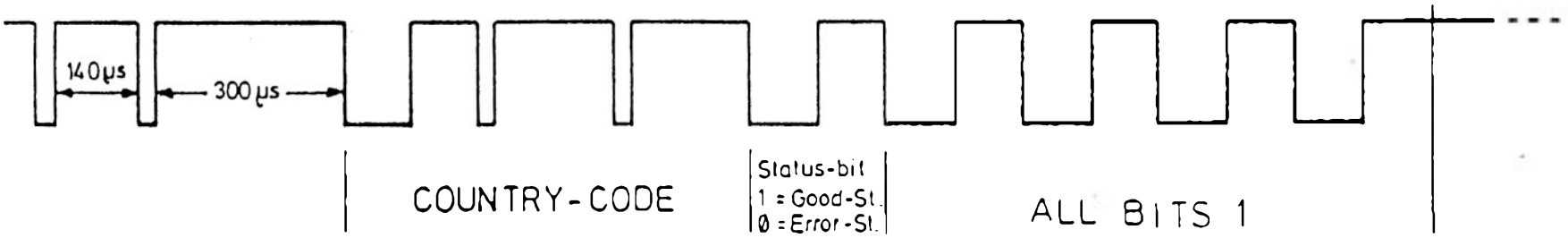
Keyboard is continuously polled
No Key is depressed



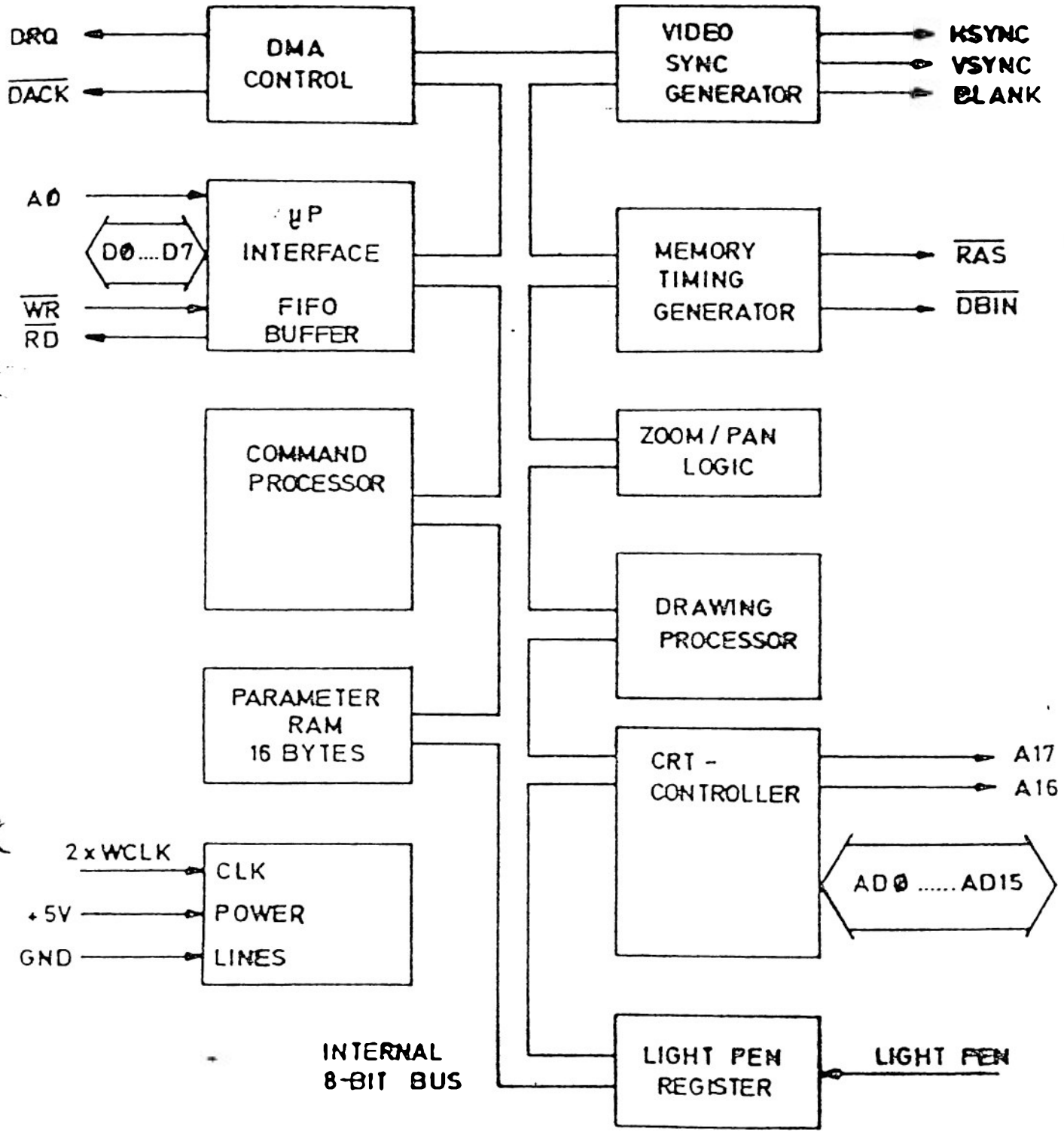
"Q"- Key (ASCII = 51H)
is depressed



After Power-On Reset, a double poll pulse
is sent out to the Keyboard to ask
it for status and Country Code



BLOCK DIAGRAM OF THE GDC 7220:



HARDWARE FEATURES OF THE GDC 7220:

- INTERFACE:
  - 8 BIT DATA, VERY SIMPLE STRUCTURE ?
  - ONLY TWO ADDRESSES NECESSARY FOR PROGRAMMING
  - COMMAND PROCESSOR
  - DMA-MODE

- CRT CONTROLLER:
  - MEMORY TIMING LOGIC FOR DRAMS
  - VIDEO SYNC GENERATOR
  - CURSOR MOVING, BLINKING
  - LIGHT PEN REGISTER
  - SCROLLING

- GRAPHIC CONTROLLER:
  - HIGH SPEED DRAWING PROCESSOR FOR CALCULATION OF PIXEL ADDRESSES/DATA DURING DRAWING PROGRESS
  - ZOOMING/PANNING LOGIC

GRAPHIC CAPABILITIES

- OPERATION MODES:
  - GRAPHIC
  - CHARACTER MODE
  - MIXED

- READ/WRITE MODIFICATION OF DISPLAY MEMORY IN 1.6 µSEC.
  - ZERO
  - SET
  - REPLACE
  - COMPLEMENT

- FIGURE DRAWING:
  - LINES HORIZONTAL AND VERTICAL
  - VECTORS
  - RECTANGLES
  - ARCUS/CIRCLES
  - GRAPHIC CHARACTER SYMBOLS
  - LINE PATTERN OR AREA PATTERN PROGRAMMABLE
  - SLANTED FIGURES ($\pm$ N·45°)

NCR DECISION MATE V

ECHTZEITUHR
(K803-V001)

Die beiliegenden Seiten zeigen Ihnen, wie Sie diese Leistungserweiterung an Ihren NCR DECISION MATE V anschließen können. Bitte ordnen Sie diese Beschreibung in Ihre Bedienungsanleitung für den NCR DECISION MATE V ein.

Doc:017-0033071

ECHTZEITUHR
(K803-V001)

INHALTSVERZEICHNIS

Teil 1
INSTALLATION

Allgemeine Hinweise..... 1
Einstellung der IFSEL-Nummer..... 2
Interrupt-Signal im Bereitschaftsbetrieb..... 4
Stromversorgung..... 5

Teil 2
SOFTWARE

Einleitung..... 6
Formatumwandlung..... 7
Adressierung von Registern..... 8
Die Struktur der Register..... 10
Programmierung der Register..... 10
Rücksetzen der Register..... 12
Programmierung der Interrupts..... 12
Der Interrupt im Bereitschaftsbetrieb..... 13
Umstellungskontrollbit..... 14
Der "GO"-Befehl..... 14
Softwarebeispiele..... 15
Beispiel 1..... 16
Beispiel 2..... 21

ALLGEMEINE HINWEISE

ACHTUNG: Diese Leistungserweiterung enthält CMOS Bauelemente. Bitte berücksichtigen Sie die üblichen Vorsichtsmaßnahmen für die Behandlung solcher Bausteine:

* Vermeiden Sie die Berührung der Kontaktleiste.
* Stellen Sie sicher, daß Ihr NCR DECISION MATE V ausgeschaltet ist, bevor der Adapter eingesetzt oder entfernt wird.
* Falls Sie das Gehäuse öffnen (siehe unten), sollten Sie die Bauelemente oder die Lötseite der Platine nicht berühren.

Dieser Adapter verwendet (wie alle anderen) eine IFSEL-Nummer (InterFace SELect), um mit dem NCR DECISION MATE V Daten auszutauschen. Insgesamt stehen 10 IFSEL-Nummern zur Verfügung, denen wiederum jeweils acht E/A Ports zugeordnet sind. Die Software zur Echtzeituhr, die unter dem Betriebssystem p-UCSD zur Verfügung steht, verwendet die IFSEL-Nummer 4B (Ports C8 bis CF) als Standardvoreinstellung. Zum Zeitpunkt der Auslieferung sind die Wahlschalter des Adapters auf diesen Wert eingestellt. Falls Sie diesen voreingestellten Wert verwenden möchten, können Sie den Adapter - ohne weitere Vorbereitungen - in einen der Steckplätze 2 bis 6 Ihres NCR DECISION MATE V einstecken (Siehe Abb. 1.1).

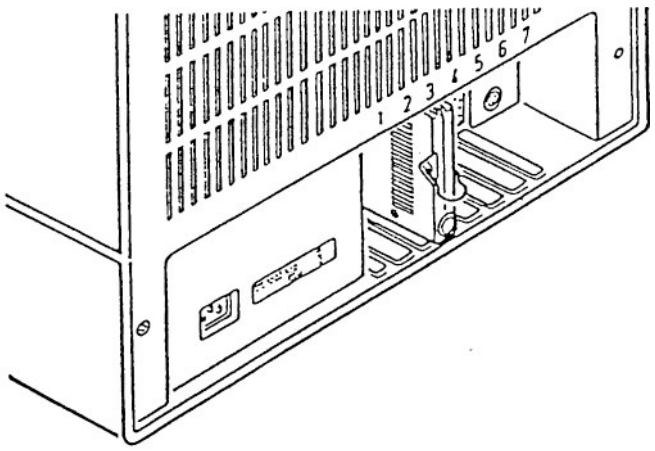


Abbildung 1.1 Anbringung des Adapters

Lesen Sie in diesem Fall bitte den zweiten Teil (Software) dieser Dokumentation.

Falls Sie die eingestellte IFSEL-Nummer verändern möchten, sollten Sie folgendermaßen vorgehen:

EINSTELLUNG DER IFSEL-NUMMER

1. Entfernen Sie den Drahtbügel und die vier Kreuzschlitzschrauben, ohne die Gehäusehälften zu trennen.
2. Drehen Sie den Adapter um (Schraublöcher nach unten) und legen Sie ihn auf eine flache Unterlage. Heben Sie die obere Gehäusehälfte ab, so daß Sie die Bestückungsseite (nicht die Lötseite) der Platine vor sich sehen. Außer den üblichen Vorsichtsmaßnahmen bei der Behandlung von CMOS-Bauelementen sollten Sie beachten, daß der Adapter einen kleinen Akkumulator enthält, um die Uhr mit Strom zu versorgen. Vermeiden Sie daher die Berührung der Platine oder von einzelnen Elementen mit leitfähigen Werkzeugen.
3. Bringen Sie die IFSEL-Wahlschalter (Abb. 1.2) in die gewünschte Stellung. Die Tabelle (Abb. 1.3) zeigt die möglichen Schaltereinstellungen. Um den Adapter in die selbe Lage zu bringen wie die Abbildungen in diesem Text, sollten Sie ihn so drehen, daß der Steckkontakt von Ihnen wegzeigt. Der Akkumulator auf der Platine ist Ihnen dann zugewandt. Der Schalter auf der linken Seite ist der Schalter Nummer 4, der Schalter auf der rechten Seite ist der Schalter 8. Falls abweichende Beschriftungen auf dem Schalterblock vorhanden sind, sollten Sie diese nicht beachten. Ein Punkt (•) in der Abbildung 1.3 zeigt, daß der betreffende Schalter an der Seite der Steckerleiste niedergedrückt werden muß (sollte Ihr Adapter mit Schiebeschalter ausgestattet sein, müssen Sie den Schalter in Richtung auf den Steckkontakt schieben). Ein Kreis zeigt, daß der Schalter an der dem Akkumulator zugewandten Seite niedergedrückt bzw. in diese Stellung geschoben werden muß. Die Abbildung 1.4 zeigt die Schalterstellung für die IFSEL-Nummer 4B.
4. Bauen Sie den Adapter wieder zusammen und setzen Sie ihn in eine der Steckfassungen 2 bis 6 Ihres (ausgeschalteten) NCR DECISION MATE V ein (Siehe Abb. 1.1).

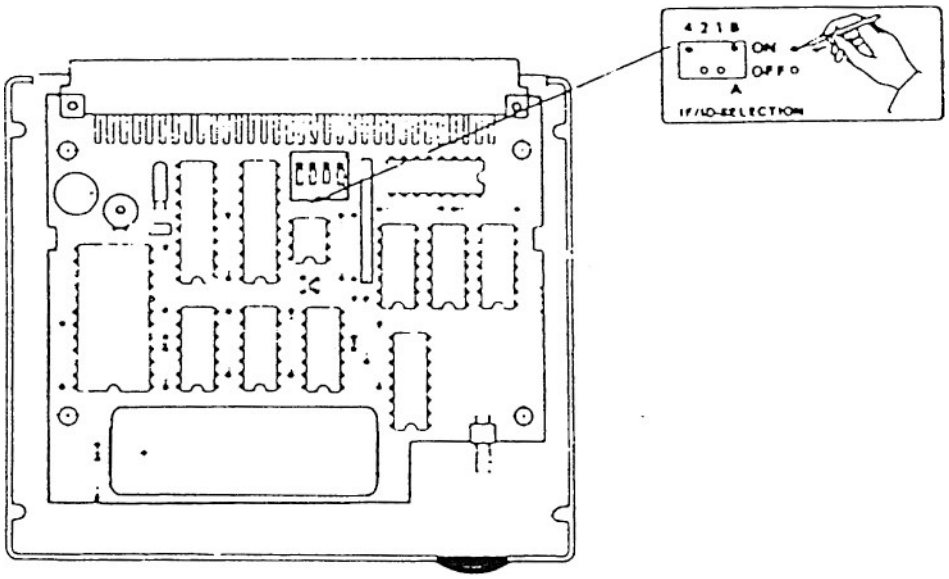


Abbildung 1.2 IFSEL-Schalter

IFSEL	SCHALTER 4 2 1 B	PORT-ADRESSE HEX	DEZ
0A	o o o o	60H - 67H	96 - 103
0S	o o o •	68H - 6FH	104 - 111
1A	o o • o	70H - 77H	112 - 119
1B	o o • •	78H - 7FH	120 - 127
2A	o • o o	30H - 37H	48 - 55
2B	o • o •	38H - 3FH	56 - 63
3A	o • • o	B0H - B7H	176 - 183
3B	o • • •	B8H - BFH	184 - 191
4A	• o o o	C0H - C7H	192 - 199
4B	• o o •	C8H - CFH	200 - 207

Abbildung 1.3 Mögliche IFSEL-Einstellungen

Abbildung 1.4 Beispiel: IFSEL-Nummer 4B

INTERRUPT-SIGNAL IM BEREITSCHAFTSBETRIEB

Die Echtzeituhr läuft auch nach dem Ausschalten Ihres Computers oder nach der Entfernung aus dem Computer weiter. Es besteht die Möglichkeit, die Echtzeituhr auf das Setzen eines Interrupt-Signals in diesem Bereitschaftsbetrieb zu programmieren. Das Interrupt-Signal wird in dem Augenblick erzeugt, in dem die Werte der laufenden Zeitzähler den vorab gespeicherten Werten entsprechen. Dieses Interrupt-Signal wird jedoch nur einmal gesetzt und verbleibt bis zu einer Rücksetzung in diesem Zustand.

Das Interrupt-Signal kann an zwei Kontakten (P2) auf der Platine abgegriffen werden (Siehe Abbildung 1.5):

Anschluß 1: INTERRUPT-SIGNAL (gesetzt=LOW)
Anschluß 2: MASSE

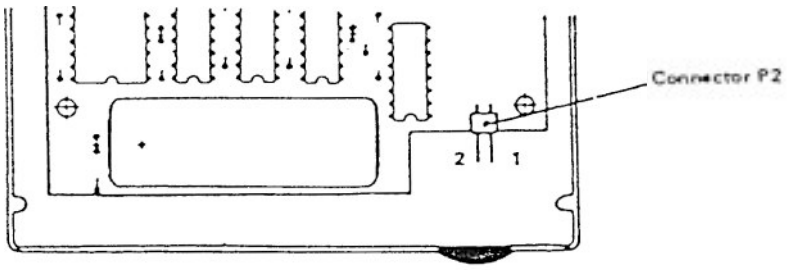


Abbildung 1.5 2-Pol Stiftleiste P2

Diese Stiftleiste ist nach Öffnung des Adapters (Siehe Beschreibung auf Seite 2) zugänglich. Den Anschlußdraht können Sie nach Entfernung des Blindstopfens durch die Gehäuseöffnung herausführen.

Der Ausgang kann z.B. zum Einschalten eines Gerätes durch eine Steuerschaltung verwendet werden. Er ist für die Belastung durch eine TTL-Schaltung entworfen. Es muß ein Abschlußwiderstand (Pull-Up- Widerstand) von 3.6KOhm verwendet werden.

Die softwareseitige Beschreibung dieses Interrupts finden Sie im zweiten Teil dieser Dokumentation.

STROMVERSORGUNG

Der Adapter enthält einen kleinen Akkumulator (3.6V; 60mAh), der die Stromversorgung der Echtzeituhr sicherstellt, wenn der Computer ausgeschaltet oder der Adapter nicht in Ihren NCR DECISION MATE V eingesteckt ist. Der Akkumulator wird automatisch während dem Betrieb des Computers geladen. Die mögliche Zeitdauer der Batterieunterstützung ist stark von der Umgebungstemperatur abhängig, bei konstant 20 Grad Celsius kann von einer Dauer von etwa 180 Tagen ausgegangen werden.

Sie sollten unter keinen Umständen versuchen, den Akkumulator von der Platine zu entfernen.

SOFTWARE

EINLEITUNG

Die Echtzeituhr ist ein Zeitzähler mit programmierbarem Alarmsignal. Die Zeiteinheiten, die von der Echtzeituhr gezählt werden sind Monate, Monatstage, Wochentage, Stunden, Minuten, Sekunden und Sekundenbruchteile.

Der Zeitpunkt für ein Alarmsignal wird in einem Satz von Speichern ("Latches") festgehalten. Durch das Besetzen einzelner oder aller Speicher mit einem "Ignoriere"-Wert ist es möglich, das Alarmsignal in wiederkehrenden Intervallen zu programmieren. Werden z.B. die Speicher für Monat, Tag des Monats und Wochentag auf "Ignoriere" gesetzt und die restlichen Speicher auf 12 Uhr Mittags programmiert, wird jeden Tag um 12 Uhr Mittags das Alarmsignal gegeben.

Außer der Zeitmessung und der Alarmfunktion bietet die Echtzeituhr folgende Leistungen:

Für den Zweck der Synchronisierung der Echtzeituhr steht ein eigenes "GO"-Befehlsregister zur Verfügung. Ein spezielles Bit ermöglicht Assemblerprogrammen eine Kontrolle, ob sich einer der Zähler während eines Lesezugriffes in Umstellung befand. Ein Lesen der Zeitzähler während des Umstellungsvorgangs kann zu einer Ungenauigkeit führen.

Die UCSD-p-Betriebssystemsoftware unterstützt die Echtzeituhr durch eine Reihe von Routinen in den Programmiersprachen BASIC, FORTRAN und Pascal. Weitergehende Information zu diesen Routinen können Sie der Dokumentation des UCSD-p-Betriebssystems entnehmen.

Selbstverständlich kann die Echtzeituhr auch unter den Betriebssystemen CP/M und MS-DOS in Assembler oder BASIC programmiert werden. Die folgenden Seiten geben Beispiele und Hinweise zur Implementierung solcher Routinen in diesen Betriebssystemen.

FORMATUMWANDLUNG

Die Speicher und Zähler der Echtzeituhr enthalten Werte im BCD-Format (Binär kodierte Dezimalzahlen). In diesem Format stellt jedes Byte zwei Stellen der Zahl, jeweils in vier Bit, dar. Die Dezimalzahl "25" wird z.B in einem Byte als 0010 0101 abgelegt, wobei die Ziffer mit der höheren Dezimalwertigkeit ("2") die oberen vier Bits belegt. Da in BASIC die Erkennung von BCD-kodierten Zahlen nicht vorgesehen ist, muß zunächst eine Ersatzdarstellung für die BCD-Zahlen errechnet werden. Für das obige Beispiel findet sich die Dezimalzahl 37, die als 00100101 binär kodiert wird.

Die Gleichung für das Umwandeln von Dezimalzahlen in eine Ersatzdarstellung für BCD-kodierte Zahlen lautet in BASIC:

```
BCD = INT(DEZIMAL/10) * 6 + DEZIMAL
```

Die Gleichung für das Umwandeln der Ersatzdarstellung von BCD-kodierten Zahlen in Dezimalzahlen lautet in BASIC:

```
DEZIMAL = INT(BCD/16) * 10 + BCD MOD 16
```

Beispiel 1:
Umwandlung der Dezimalzahl 43 in eine BCD-kodierte Zahl:

```
BCD = INT(43/10) * 6 + 43
    =      4     * 6 + 43
    = 67
```

Die Ausgabe der Zahl 67 durch das BASIC Programm ergibt dasselbe Bitmuster wie die BCD-Kodierung von 43.

Beispiel 2:
Umwandlung der BCD-kodierten Zahl 67 in eine Dezimalzahl:

```
DEZIMAL = INT(67/16) * 10 + 67 MOD 16
        =      4     * 10 +     3
        = 43
```

Dies ist der wahre Dezimalwert des Bytes 0100 0011, das von BASIC aber als "67" gelesen wird.

Jede BCD-Zahl entspricht der Binärkodierung einer hexadezimalen Zahl, bei der keine Ziffer größer als neun ist. Dies ist von besonderer Bedeutung, falls Sie in Assembler programmieren oder hexadezimale Zahlen als Eingabewerte für ein BASIC-Programm verwenden.

ADRESSIERUNG VON REGISTERN

Die Echtzeituhr verwendet 23 Register. Ein Registerzugriff wird durch zwei zusammengehörige Befehle ausgeführt:
Der erste Befehl wählt einen der sechs Adreßspeicher aus, welche jeweils eine Gruppe von vier (Adreßspeicher 5: drei) Register verwalten. Der nachfolgende Befehl wählt eines der zum angewählten Adreßspeicher gehörigen Register und greift auf dieses zu. Die Abbildung 3.1 zeigt tabellarisch die Funktion der einzelnen Register, die Zugehörigkeit zu den Adreßspeichern und die mögliche Art des Zugriffs.
(R=READ=Lesezugriff; W=WRITE=Schreibzugriff)

BADD stellt die Basisadresse des verwendeten IFSELs dar. Unter der Annahme, daß IFSEL 4B verwendet wird, ist das adressierte Port C8H (Siehe Abb. 1.3). Der zuletzt aktivierte Adreßspeicher bleibt gültig bis zur nächsten Ausgabe an das Port BADD. Dies ermöglicht den Austausch von Daten mit Register, die zu einen Adreßspeicher gehören, ohne der Notwendigkeit einer wiederholten Anwahl des Adreßspeichers. Das nachfolgende Beispiel zeigt diesen Vorgang: Zuerst wird eine Gruppe von vier Registern angewählt, dann werden die Werte für die Register der Sekunden- und Minutenzähler auf die jeweiligen Werte in SEC und MIN gesetzt.

```
OUT BADD,0
OUT BADD+6,SEC
OUT BADD+7,MIN
```

FUNKTION	ADR.SPEICHER	REGISTER	ZUGRIFF
ZÄHLER 1/10000 SEKUNDEN	BADD,0	BADD+4	R/W
ZÄHLER 1/100 SEKUNDEN	BADD,0	BADD+5	R/W
ZÄHLER SEKUNDEN	BADD,0	BADD+6	R/W
ZÄHLER MINUTEN	BADD,0	BADD+7	R/W
ZÄHLER STUNDEN	BADD,1	BADD+4	R/W
ZÄHLER WOCHENTAGE	BADD,1	BADD+5	R/W
ZÄHLER TAGE DES MONATS	BADD,1	BADD+6	R/W
ZÄHLER MONATE	BADD,1	BADD+7	R/W
LATCH 1/10000 SEKUNDEN	BADD,2	BADD+4	R/W
LATCH 1/100 SEKUNDEN	BADD,2	BADD+5	R/W
LATCH SEKUNDE	BADD,2	BADD+6	R/W
LATCH MINUTE	BADD,2	BADD+7	R/W
LATCH STUNDE	BADD,3	BADD+4	R/W
LATCH WOCHENTAG	BADD,3	BADD+5	R/W
LATCH TAG DES MONATS	BADD,3	BADD+6	R/W
LATCH MONAT	BADD,3	BADD+7	R/W
INTERRUPT STATUS	BADD,4	BADD+4	R
INTERRUPT BEFEHL	BADD,4	BADD+5	W
RÜCKSETZEN ZÄHLER	BADD,4	BADD+6	W
RÜCKSETZEN LATCHES	BADD,4	BADD+7	W
UMSTELLUNGSKONTROLLBIT	BADD,5	BADD+4	R
"GO" BEFEHL	BADD,5	BADD+5	W
BEREITSCHAFTSINTERRUPT	BADD,5	BADD+6	W

Abbildung 2.1 Register der Echtzeituhr

DIE STRUKTUR DER REGISTER

Es stehen jeweils acht Register für die Zähler und für die Alarm-"Latches" zur Verfügung. Jedes Register enthält acht Bits und kann somit eine ein- oder zweiziffrige BCD-Zahl enthalten. Unbelegte Bits sind in der Abb. 2.2 durch "-" dargestellt, sie werden bei Lesezugriffen als logisch 0 zurückgegeben und bei Schreibzugriffen ignoriert.

ZÄHLER/LATCH	EINER Bit:D0	D1	D2	D3	ZEHNER Bit:D4	D5	D6	D7
1/10000 SEKUNDEN	-	-	-	-	X	X	X	X
1/100 SEKUNDEN	X	X	X	X	X	X	X	X
SEKUNDEN	X	X	X	X	X	X	X	-
MINUTEN	X	X	X	X	X	X	X	-
STUNDEN	X	X	X	X	X	X	-	-
WOCHENTAGE	X	X	X	-	-	-	-	-
TAGE DES MONATS	X	X	X	X	X	X	-	-
MONAT	X	X	X	X	X	-	-	-

Abbildung 2.2 Bitbelegung der Zähler- und Latch-Register

PROGRAMMIERUNG DER REGISTER

Die Abbildungen 2.1 und 2.2 enthalten alle Informationen, die Sie zum Programmieren der Register benötigen. Nachfolgend finden Sie Beispiele für Zugriffe auf die Zählerregister und eine Beschreibung der Alarm-"Latches". Beachten Sie bitte, daß es nicht erforderlich ist, einen Adreßspeicher wiederholt anzuwählen, wenn nacheinander Zugriffe auf die zugeordneten Register erfolgen.

PROGRAMMIERUNG DER ZÄHLERREGISTER

Das folgende Beispiel stellt den Stundenzähler auf den in der Variable HOUR enthaltenen BCD-Wert:

```
OUT BADD,1          (Wahl des Adreßspeichers)
OUT BADD+4,HOUR (Schreibzugriff auf den
                    Stundenzähler)
```

Das folgende Beispiel liest den BCD-Wert des Stundenzählers in die Variable HOUR ein:

```
OUT  BADD,1          (Wahl des Adreßspeichers)
IN   (BADD+4),HOUR   (Lesen des Registerinhalts
                      in die Variable)
```

PROGRAMMIERUNG DER ALARM-"LATCHES"

Die Struktur und die Zugriffsweise auf die Register der Alarm-"Latches" unterscheiden sich nicht von denen der Zählerregister. Die Echtzeituhr vergleicht den Inhalt der Latch-Register mit denen der Zählerregister und löst einen Interrupt aus, sobald alle folgenden Bedingungen erfüllt sind:

* Es wird eine Übereinstimmung von mindestens einem Zähler- / Alarmregister ermittelt.
* Alle Alarmregister, die nicht übereinstimmen, sind auf den Wert "ignoriere" gesetzt.
* Der Alarm-"Latch"-Interrupt ist aktiviert. (Siehe Abschnitt "PROGRAMMIEREN DES INTERRUPTS")

Die Register der Alarm-"Latches" können einzeln auf "ignoriere" gesetzt werden durch die jeweilige Belegung mit der Bitfolge 1100 1100 (Dezimalwert 204). Die Echtzeituhr erkennt diesen besonderen Wert, da keine BCD-kodierte Zahl existiert, bei der die oberen zwei Bits der vier-Bit-Gruppen gesetzt sind. Das folgende Beispiel setzt den Alarm-"Latch" für Minuten auf "ignoriere":

```
OUT BADD,2        (Wahl des Adreßspeichers)
OUT BADD+7,204    (Schreibzugriff auf den
                   Latch:"ignoriere")
```

RÜCKSETZEN DER REGISTER

Um die Zähler- und die Alarmregister einfach auf den niedrigstmöglichen Wert zu setzen, stehen jeweils spezielle Register zur Verfügung. Um die Rücksetzung zu bewirken, müssen die jeweiligen Register (Siehe Abbildung 2.1) mit dem Dezimalwert 255 besetzt werden. Beispiel:

```
OUT BADD,4        (Wahl des Adreßspeichers)
OUT BADD+6,255    (Rücksetzen aller Zählerregister)
```

PROGRAMMIERUNG DER INTERRUPTS

Zusätzlich zum bisher beschriebenen "Alarm"-Interrupt kann die Echtzeituhr auf die Erzeugung von Interrupts in den folgenden Intervallen programmiert werden:

einmal pro 1/10 Sekunde
einmal pro Sekunde
einmal pro Minute
einmal pro Stunde
einmal pro Tag
einmal pro Woche
einmal pro Monat

Die Abbildung 2.3 zeigt die Bedeutung der einzelnen Bits im Interrupt-Befehlsregister. Ein gesetztes Bit zeigt an, daß der jeweilige Interrupt aktiviert ist.

BITPOSITION	DEZIMAL	INTERRUPTQUELLE
7	128	Monat
6	64	Woche
5	32	Tag
4	16	Stunde
3	8	Minute
2	4	Sekunde
1	2	1/10 Sekunde
0	1	Alarm

Abbildung 2.3 Bitpositionen des Interrupt-Befehlsregisters

Zwei Beispiele zeigen die Aktivierung von Interrupts:

```
OUT BADD,4        (Wahl des Adreßspeichers)
OUT BADD+5,1      (Setzen des Bit 0, Alarm)

OUT BADD,4        (Wahl des Adreßspeichers)
OUT BADD+5,24     (Setzen der Bits Stunde
                   und Minute)
```

Ein Interrupt-Statusregister enthält Information, welche Interrupts seit dem letzten Lesen des Statusregisters stattgefunden haben. Die Zuordnung der Bitpositionen bzw. der Dezimalwerte entspricht der Abbildung 2.3. Beispiel:

```
OUT BADD,4            (Wahl des Adreßspeichers)
IN (BADD+4),INTSET    (Einlesen in Variable INTSET)
```

Der Dezimalwert 1 in der Variable INTSET nach dem Lesen des Interrupt-Statusregisters z.B. zeigt an, daß seit dem letzten Lesen ein Alarm-Interrupt ausgelöst worden ist.

DER INTERRUPT IM BEREITSCHAFTSBETRIEB

Dieser Interrupt ist an die Erfüllung folgender Bedingungen geknüpft:

* Die Werte aller Alarm-"Latches", die nicht mit "ignoriere" besetzt sind, stimmen mit den Werten der jeweiligen Zähler überein.
* Der Bereitschafts-Interrupt ist aktiviert.

Der Interrupt ist aktiviert, wenn das niedrigstwertige Bit des zugehörigen Registers (Siehe Abb. 2.1) gesetzt ist. Dieser Interrupt wird vom Interrupt-Steuerregister nicht beeinflußt. Beispiel:

```
OUT BADD,5        (Wahl des Adreßspeichers)
OUT BADD+6,1      (Aktivieren des Ber.Interrupts)
              bzw.
OUT BADD+6,0      (Rücksetzen des Ber.Interrupts)
```

UMSTELLUNGSKONTROLLBIT

Dieses Bit wird in einem eigenen Register (Siehe Abb.2.1) gespeichert und wird von der Echtzeituhr gesetzt, wenn sich eines der Zählerregister gerade in Umstellung befindet. Dieses Bit ist insbesondere von Bedeutung für Assembler-Programme, die ein hohes Maß an Zeit-Einlesegenauigkeit erfordern. Solche Programme sollten dieses Bit abfragen und einen Lesezugriff auf einen Zähler wiederholen, falls dieses Bit gesetzt war.

DER "GO"-BEFEHL

Mithilfe dieses Befehls kann der Ablauf der Echtzeituhr genau gestartet werden. Ein Schreibzugriff auf das Register, das diesen Befehl enthält, bewirkt die Rücksetzung der Zähler 1/10000, 1/100, 1/10 und 1/1 Sekunde. Falls der Wert des Zählers für ganze Sekunden zum Zeitpunkt der Rücksetzung größer als 40 ist, wird der Minutenzähler um eins erhöht. Andernfalls ist der Minutenzähler nicht betroffen.

Der Ablauf der Synchronisierung der Echtzeituhr auf die Zeit 15 Uhr 12 Minuten ist z.B. wie folgt:

1. Setzen der "langsameren" Zähler für Stunden und Minuten auf die Werte 15 bzw. 12.
2. Setzen des Sekundenzählers auf den Wert "0"
3. Wahl des Adreßspeichers: OUT BADD,5
4. Vorausgesetzt, daß der Wert "1" um genau 15 Uhr 12 in den "GO"-Befehlsregister geschrieben wird und seit der Nullsetzung des Sekundenzählers weniger als 40 Sekunden vergangen sind, wird die Echtzeituhr genau auf 15 Uhr 12 synchronisiert: OUT BADD+5,1

Die genaue Einstellung der Echtzeituhr ist nur in Assembler-Programmen möglich, da diese die unmittelbarste Ausführung eines Programmausgabebefehls gewähren. Für Programmierer, die die größtmögliche Synchronisations-Genauigkeit erzielen möchten, ist die Arbeitsfrequenz der Prozessoren des NCR DECISION MATE V von Bedeutung:

8-Bit (Z80A) - 4MHz
16-Bit (8088) - 5MHz

SOFTWAREBEISPIELE

Der folgende Abschnitt zeigt anhand zweier Beispiele die Programmierung der Echtzeituhr in BASIC. Auch Assembler-Programmierern werden diese Beispiele nützlich sein, da die grundsätzlichen Ein-/Ausgabebefehle an die Echtzeituhr auch in der höheren Programmiersprache verständlich sind.

Das erste Beispiel zeigt wie die Uhrzeit und das Tagesdatum eingestellt und abgefragt werden. Außerdem wird ein Test durchgeführt, um die verwendete IFSEL-Nummer zu ermitteln und davon die korrekte Basisadresse abzuleiten und auf der Diskette zu speichern.

Das zweite Beispiel zeigt das setzen der Alarm-Zeit und ermöglicht die Ausgabe eines Alarmtextes. Die Basisadresse wird von der Diskette gelesen.

Die Programme sind unter den Betriebssystemen CP/M und MS-DOS lauffähig.

BEISPIEL 1

ZWECK: STELLEN DER ECHTZEITUHR, ANZEIGEN VON DATUM UND UHRZEIT.

BESCHREIBUNG DES PROGRAMMS:

ZEILENNR.:	KURZBESCHREIBUNG:
50 - 100	ROUTINE ZUR UMWANDLUNG VON DEZIMAL ZU BCD UND ZURÜCK
110 - 130	SPRUNG IN DIE TESTROUTINE DER ECHTZEITUHR (SIEHE 1140)
140 - 210	AUSGABE EINES BILDSCHIRMMENÜS: WAHL ZWISCHEN STELLEN, ABFRAGEN DER ZEIT ODER ENDE DES PROGRAMMS
220 - 380	EINGABE DER ZEIT UND DES DATUMS. EINGABEN AUSSERHALB DES WERTEBEREICHS WERDEN ERKANNT.
390 - 460	DEZIMAL EINGEGEBENE DATEN WERDEN IN BCD VERWANDELT.
470 - 550	DIE UMGEWANDELTEN WERTE WERDEN IN DIE REGISTER DER ECHTZEITUHR EINGETRAGEN.
560 - 700	ZEIT UND DATUM WERDEN VON DER ECHTZEITUHR ABGEFRAGT. EINE UMSTELLUNG DER UHR VON 59 SEKUNDEN AUF 00 SEKUNDEN WIRD ÜBERWACHT. ANSCHLIESSEND WERDEN DIE ERMITTELTEN WERTE VON BCD NACH DEZIMAL UMGEWANDELT. 'FLAG' IST EIN SCHALTER FÜR DIE BILDSCHIRMAUSGABE DER ZEIT UND DES DATUMS BEIM ERSTEN SCHLEIFENDURCHGANG .
710 - 820	AUSGABE DES DATUMS UND DER ZEIT IM KORREKTEN FORMAT. WOCHENTAG- UND MONATWERTE WERDEN MIT NAMEN ANGEZEIGT.
830 - 880	ABFRAGE, OB DER ANWENDER DAS PROGRAMM VERLASSEN MÖCHTE.
890 - 980	UMWANDLUNG VOM WOCHENTAG-WERT IN DEN WOCHENTAG-NAMEN.
990 - 1130	UMWANDLUNG VOM MONATS-WERT IN DEN MONATS-NAMEN.

1140 - 1480 DIESES UNTERPROGRAMM KONTROLLIERT, OB DIE ECHTZEITUHR ZUR VERFÜGUNG STEHT. EIN AUSGEWÄHLTES ALARM-'LATCH'-REGISTER WIRD MIT EINEM WERT BESCHRIEBEN. DIE ECHTZEITUHR IST VERFÜGBAR, WENN EINE ANSCHLIESSENDE ABFRAGE DES REGISTERS DENSELBEN WERT ZURÜCKGIBT, DER VORHER GESCHRIEBEN WURDE. FALLS DIE BASISADRESSE DES PORTS FALSCH IST, WIRD EINE TABELLE ALLER I/O PORTADRESSEN ANGEZEIGT. ES KANN DER KORREKTE WERT EINGEGEBEN WERDEN, EINE ABFRAGE FINDET STATT.
DIE STANDARDVOREINSTELLUNG DER PORTADRESSE IST DEZIMAL 200 (HEXADEZIMAL C8 VON DIPSEL 4B). DIESE PORTADRESSE WIRD AUF DER DISKETTE GESPEICHERT, SIE WIRD VOM BEISPIELPROGRAMM 2 VON DORT ABGELESEN.

1490 - 1530 ROUTINEN ZUR ZEIGERPOSITIONIERUNG UM EINE KORREKT FORMATIERTE AUSGABE ZU ERZIELEN.

```
10 REM                EXAMPLE - 1
20 REM  SET AND GET TIME TO/FROM REAL TIME CLOCK
30 REM                  K 8 0 3
40 REM  ------------------------------------
50 REM DEFINE A FUNCTION TO CONVERT FROM DECIMAL TO BCD
60 DEF FNTOBCD(X)=(X\10)*6+X
70 REM
80 REM DEFINE A FUNCTION TO CONVERT FROM BCD TO DECIMAL
90 DEF FNTODEC(X)=(X\16)*10 + X MOD 16
100 REM
110 REM  GOTO RTC AVAILABLE
120 GOSUB 1140
130 REM
140 PRINT:PRINT " FUNCTION SELECTION : "
150 PRINT " 1 - SET TIME TO RTC "
160 PRINT " 2 - GET TIME FROM RTC "
170 PRINT " 3 - EXIT PROGRAM "
180 SEL$=" "
190 INPUT " ",A
200 ON A GOTO 220,550,1540
210 PRINT " ? ";:GOTO 190
220 REM CLEAR SCREEN AND GET TIME FROM KEYBOARD
230 REM ------------------------------------
240 PRINT CHR$(26);"PLEASE ENTER THE FOLLOWING DATA!"
250 PRINT
```

```
260 INPUT "           MONTH (1-12):";MTH
270  IF MTH<1 OR MTH>12 GOTO 260
280 INPUT "DAY OF WEEK (1-7 SUN=1):";DOW
290  IF DOW<1 OR DOW>7 GOTO 280
300 INPUT "    DAY OF MONTH (1-31):";DOM
310  IF DOM<1 OR DOM>31 GOTO 300
320 PRINT
330 INPUT "            HOUR (0-23):";HR
340  IF HR<0 OR HR>23 GOTO 330
350 INPUT "         MINUTES (0-59):";MIN
360  IF MIN<0 OR MIN>59 GOTO 350
370 INPUT "         SECONDS (0-59):";SEC
380  IF SEC<0 OR SEC>59 GOTO 370
390 REM CONVERT REAL TIME AND DATE FROM DECIMAL TO BCD
400 REM ------------------------------------------
410 MTH=FNTOBCD(MTH)          'CONVERT MONTH
420 DOM=FNTOBCD(DOM)          'CONVERT DATE
430 DOW=FNTOBCD(DOW)          'CONVERT DAY OF WEEK
440 HR=FNTOBCD(HR)            'CONVERT HOUR
450 MIN=FNTOBCD(MIN)          'CONVERT MINUTES
460 SEC=FNTOBCD(SEC)          'CONVERT SECOND
470 REM SET REAL TIME AND DATE AS ENTERED
480 REM ------------------------------------
490  OUT BADD,1: OUT BADD+7,MTH       'SET MONTH
500              OUT BADD+6,DOM       'SET DAY OF MONTH
510              OUT BADD+5,DOW       'SET DAY OF WEEK
520              OUT BADD+4,HR        'SET HOUR
530  OUT BADD,0: OUT BADD+7,MIN       'SET MINUTES
540              OUT BADD+6,SEC       'SET SECONDS
550 REM
560 REM GET TIME AND DATE FROM RTC
570 REM ---------------------------
580 PRINT CHR$(26)          'CLEAR SCREEN AND CURSOR HOME
585 LI=0 : PO=0 : GOSUB 1520 : PRINT "ENTER 'Q' TO QUIT PROGRAM"
590 FLAG=0
600 REM GET TIME AND PRINT ONCE PER SECOND
610  OUT BADD,0: X=INP(BADD+6): SEC=FNTODEC(X)     'READ SECOND
620 IF SEC=59 THEN ADDSEC=0 ELSE ADDSEC=SEC+1      'SET ROLLOVER CHECK
630  OUT BADD,0: X=INP(BADD+6): SEC=FNTODEC(X)     'READ SECONDS
640 IF SEC<>ADDSEC GOTO 630         'IF A SECOND HAS PASSED, GET THE TIME
650  OUT BADD,0: X=INP(BADD+7): MIN=FNTODEC(X)     'GET MINUTE
660  OUT BADD,1: X=INP(BADD+4): HR=FNTODEC(X)      'GET HOURS
```

```
670  X=INP(BADD+5): DOW=FNTODEC(X)      'GET DAY OF WEEK
680  X=INP(BADD+6): DOM=FNTODEC(X)      'GET DAY OF MONTH
690  X=INP(BADD+7): MTH=FNTODEC(X)      'GET MONTH
700 IF ADDSEC<>0 AND FLAG=1 GOTO 800
710 REM PRINT REAL TIME AND DATE
720 REM ------------------------
730 PRINT CHR$(30): PRINT  CHR$(23)   'CURSOR HOME & CLEAR END OF LINE
740 ON DOW GOSUB 910,920,930,940,950,960,970
750 ON MTH GOSUB 1010,1020,1030,1040,1050,1060,1070,1080,1090,1100,1110,
    1120
760 PRINT "DATE: "; DOW$;", ";MTH$;" ";DOM;
770 LI=2 : PO=40 : GOSUB 1520                        'POS. HR,MIN
780 PRINT USING "TIME: ## : ## :";HR, MIN
790 FLAG=1
800 LI=2 :PO=55:  GOSUB 1520: PRINT CHR$ (23)        'CLEAR END OF LINE
810 LI=2 :PO=55:  GOSUB 1520                         'POS. SEC
820 PRINT  SEC;
830 REM QUIT PROGRAM ?
840 REM ---------------
850 SEL$=INKEY$
860 IF SEL$="Q" GOTO 140
870 GOTO 620
880 REM
890 REM CONVERT DAY OF WEEK NUMBERS TO DAY OF WEEK NAMES
900 REM ------------------------------------------------
910 DOW$="Sunday": RETURN
920 DOW$="Monday": RETURN
930 DOW$="Tuesday": RETURN
940 DOW$="Wednesday": RETURN
950 DOW$="Thursday": RETURN
960 DOW$="Friday": RETURN
970 DOW$="Saturday": RETURN
980 REM
990 REM CONVERT MONTH NUMBERS TO MONTH NAMES
1000 REM ------------------------------------
1010 MTH$="January": RETURN
1020 MTH$="February": RETURN
1030 MTH$="March": RETURN
1040 MTH$="April": RETURN
1050 MTH$="May": RETURN
1060 MTH$="June": RETURN
1070 MTH$="July": RETURN
```

```
1080 MTH$="August": RETURN
1090 MTH$="September": RETURN
1100 MTH$="October": RETURN
1110 MTH$="November": RETURN
1120 MTH$="December": RETURN
1130 REM
1140 REM RTC AVAILABLE ? TEST
1150 REM ----------
1160 REM
1170 REM DEFADLT BADD = 200 DEC
1180 BADD=200
1190 OUT BADD,3: X=INP(BADD+7)
1200 OUT BADD+7,11
1210 T=INP(BADD+7)
1220 OUT BADD+7,X
1230 IF T=11 THEN GOTO 1430
1240 PRINT  CHR$(26)
1250 PRINT "TABLE FOR BASE PORT ADDRESS (BADD):"
1260 PRINT
1270 PRINT "IPSEL-   BADD    BADD"
1280 PRINT "SWITCH   (HEX)   (DEC)"
1290 PRINT "-------------------"
1300 PRINT "0-A      60      96"
1310 PRINT "0-B      68      104"
1320 PRINT "1-A      70      112"
1330 PRINT "1-B      78      120"
1340 PRINT "2-A      30      48"
1350 PRINT "2-B      38      56"
1360 PRINT "3-A      B0      176"
1370 PRINT "3-B      B8      184"
1380 PRINT "4-A      C0      192"
1390 PRINT "4-B      C8      200"
1400 PRINT
1410 INPUT "INPUT THE BASE PORT ADDRESS (DEC) OF THE RTC (SEE TABLE):",
     BADD
1420 IF BADD=96 OR BADD=104 OR BADD=112 OR BADD=120 OR BADD=48 OR
     BADD=56 OR BADD=176 OR BADD=184 OR BADD=192 OR BADD=200 THEN
     GOTO 1190
     ELSE GOTO 1410
1430 PRINT CHR$(26)
1440 PRINT "RTC AVAILABLE ON PORT ADDRESS (DEC): " ;BADD
1450 PRINT
```

```
1460 BADD$=STR$(BADD) : OPEN "O",#1,"BADD" : WRITE #1,BADD$ : CLOSE #1
1470 RETURN
1480 REM
1490 REM CURSOR POS
1500 REM ----------
1510 REM
1520 PRINT CHR$(27);CHR$(61);CHR$(32+LI);CHR$(32+PO);
1530 RETURN
1540 END
```

BEISPIEL 2

ZWECK: DIESES PROGRAMM ZEIGT DIE UHRZEIT UND DAS DATUM AN UND ERLAUBT DAS SETZEN DER ALARM-"LATCHES". SOBALD EIN ALARMZEITPUNKT ERREICHT WIRD, ERFOLGT DIE AUSGABE EINER MELDUNG.

BESCHREIBUNG DES PROGRAMMS:

ZEILENNR.:	KURZBESCHREIBUNG:
50 - 90	ROUTINE ZUR UMWANDLUNG VON DEZIMALZAHLEN IN BCD-ZAHLEN UND ZURÜCK.
100 - 110	PORTADRESSE (STANDARDVOREINSTELLUNG) WIRD VON DER DISKETTE GELESEN.
120 - 160	RÜCKSETZUNG DER ALARM-"LATCH"-REGISTER DER ECHTZEITUHR.
170 - 500	ZEIT- UND DATUMANZEIGE DER ECHTZEITUHR WIE IN BEISPIEL 1. ZUERST WIRD UM EINE EINSTELLUNG DER ALARMZEIT GEBETEN. DAS VORLIEGEN EINES ALARMS WIRD ABGEFRAGT. BEIM VORLIEGEN EINES ALARMS WIRD EINE NACHRICHT AUSGEGEBEN. UM EINE KORREKT FORMATIERTE AUSGABE ZU ERZIELEN WIRD "PRINT USING" VERWENDET.
510 - 620	ABFRAGE, OB TASTATUREINGABE VORLIEGT DURCH DIE HAUPTSCHLEIFE, DIE SEKUNDEN ANZEIGT. 'Q'=BEENDEN DES PROGRAMMS, "A"=ALARM. ZUSÄTZLICH WIRD DER ALARM AKTIVIERT.
630 - 720	UMWANDLUNG VOM WOCHENTAG-WERT IN DEN WOCHENTAG-NAMEN.

730 - 870 UMWANDLUNG VOM MONAT-WERT IN DEN MONAT-NAMEN.

880 - 1170 DIESE ROUTINE WIRD AUFGERUFEN, WENN SIE DEN ALARM STELLEN MÖCHTEN. ZUERST WIRD DER ALARM, FALLS GESETZT, UNWIRKSAM GEMACHT UND DIE WERTE DES VORHERIGEN ALARMS GELÖSCHT. EINE BILDSCHIRMANZEIGE ERBITTET DIE EINGABE DES DATUMS, DER ZEIT UND EINER NACHRICHT. UNKORREKTE WERTE WERDEN ABGEFANGEN. LEEREINGABE VON <CR> BEWIRKT DIE BELEGUNG DER ALARM-"LATCH"-REGISTER MIT "IGNORIERE".

1180 - 1300 DARSTELLUNG DES ALARMZEITPUNKTES IM KORREKTEN FORMAT.

1310 - 1390 UMWANDLUNG DES DEZIMALWERTES IN BCD. "IGNORIERE" BELEGT DIE LATCHES MIT 204 DEZIMAL = 11001100 BINÄR.

1400 - 1490 SETZEN DER ALARMLATCHES MIT DEN ERMITTELTEN WERTEN, AUSGABE DER AKTUELLEN ZEIT UND DES DATUMS

1500 - 1610 UNTERROUTINE ZUR AUSGABE DER ALARM-NACHRICHT UND ERZEUGUNG EINES AKUSTISCHEN SIGNALS.

1620 - 1650 POSITIONIERUNG DER SCHREIBMARKE (CURSOR)

1660 - 1670 LÖSCHEN BIS ENDE BILDSCHIRM

```
10 REM                        EXAMPLE - 2
20 REM  SET THE LATCHES ON THE REAL TIME CLOCK AND WAIT FOR ALARM
30 REM                          K 8 0 3
40 REM ------------------------------------------------------------
50 REM DEFINE A FUNCTION TO CONVERT FROM DECIMAL TO BCD
60 DEF FNTOBCD(X)=(X\10)*6+X
70 REM
80 REM DEFINE A FUNCTION TO CONVERT FROM BCD TO DECIMAL
90 DEF FNTODEC(X)=(X\16)*10 + X MOD 16
100 OPEN "I",#1,"BADD" : INPUT #1,BADD$ : BADD=VAL(BADD$) : CLOSE #1
110 REM
120 REM LATCH RESET
130 REM ----------
140 OUT BADD,4
150 OUT BADD+7,255
160 REM
```

```
170 REM GET TIME AND DATE FROM RTC
180 REM ---------------------------
190 REM
200 PRINT CHR$(26)        'CLEAR SCREEN AND CURSOR HOME
210 LI=0 : PO=0 : GOSUB 1610
220 PRINT "ENTER 'A' TO SET ALARM, 'Q' TO QUIT PROGRAM "
230 FLAG=0
240 REM GET TIME AND PRINT ONCE PER SECOND
250  OUT BADD,0: X=INP(BADD+6): SEC=FNTODEC(X)    'READ SECOND
260 IF SEC=59 THEN ADDSEC=0 ELSE ADDSEC=SEC+1     'SET ROLLOVER CHECK
270  OUT BADD,0: X=INP(BADD+6): SEC=FNTODEC(X)    'READ SECONDS
280 REM DETECT FOR ALARM
290  OUT BADD,4          'INT. CONTR./STATUS REG.
300 XX=INP(BADD+4)       'READ INT. STATUS REG.
310 IF XX<>0 THEN GOSUB 1500 'DISPLAY THAT INT. HAS OCCURRED
320 IF SEC<>ADDSEC GOTO 270        'IF A SECOND HAS PASSED, GET THE TIME
330  OUT BADD,0: X=INP(BADD+7): MIN=FNTODEC(X)    'GET MINUTE
340  OUT BADD,1: X=INP(BADD+4): HR=FNTODEC(X)     'GET HOURS
350  X=INP(BADD+5): DOW=FNTODEC(X)     'GET DAY OF WEEK
360  X=INP(BADD+6): DOM=FNTODEC(X)     'GET DAY OF MONTH
370  X=INP(BADD+7): MTH=FNTODEC(X)     'GET MONTH
380 IF ADDSEC<>0 AND FLAG=1 GOTO 470
390 REM PRINT REAL TIME AND DATE
400 PRINT CHR$(30): PRINT  CHR$(23)     'CURSOR HOME & CLEAR LINE
410 ON DOW GOSUB 650,660,670,680,690,700,710
420 ON MTH GOSUB 750,760,770,780,790,800,810,820,830,840,850,860
430 PRINT "DATE: ", DOW$;", ",MTH$,DOM;
440 LI=2 : PO=55 : GOSUB 1610           'POS. HR,MIN
450 PRINT USING "TIME: ## : ## :";HR, MIN
460 FLAG=1
470 LI=2 :PO=70:  GOSUB 1610: PRINT CHR$ (23)   'CLEAR END OF LINE
480 LI=2 :PO=70:  GOSUB 1610                    'POS. SEC
490 PRINT USING " ##"; SEC;
500 REM
510 REM QUIT PROGRAM ? OR SET LATCHES ?
520 REM ---------------------------------
530 SEL$=INKEY$
540 IF SEL$="Q" GOTO 1680
550 IF SEL$="A" THEN GOTO  880     'ENTRY FOR LATCH
560 REM ENABLE INTERRUPT ON LATCH ALARM
570  OUT BADD,4        'INT. CONTR.& STATUS REG.
580  OUT BADD+5,1      'SELECT INT. CONTR. REG & ENABLE ALARM
```

```
590 REM
600 REM
610 GOTO 250          'WAIT FOR NEXT SEC
620 REM
630 REM CONVERT DAY OF WEEK NUMBERS TO DAY OF WEEK NAMES
640 REM ------------------------------------------
650 DOW$="Sunday": RETURN
660 DOW$="Monday": RETURN
670 DOW$="Tuesday": RETURN
680 DOW$="Wednesday": RETURN
690 DOW$="Thursday": RETURN
700 DOW$="Friday": RETURN
710 DOW$="Saturday": RETURN
720 REM
730 REM CONVERT MONTH NUMBERS TO NAMES
740 REM ---------------------------
750 MTH$="January": RETURN
760 MTH$="February": RETURN
770 MTH$="March": RETURN
780 MTH$="April": RETURN
790 MTH$="May": RETURN
800 MTH$="June": RETURN
810 MTH$="July": RETURN
820 MTH$="August": RETURN
830 MTH$="September": RETURN
840 MTH$="October": RETURN
850 MTH$="November": RETURN
860 MTH$="December": RETURN
870 REM
880 REM S E T  L A T C H
890 REM -----------------
900 OUT BADD,4 : OUT BADD+5,0 : YY=INP(BADD+4)   'DISABLE ALARM,
    CLEAR INT.
910 LI=8 : PO=0 : GOSUB 1610 : GOSUB 1660   'CURSOR POS. & CLEAR END OF
    SCREEN
920 PRINT TAB(10); " E N T E R   A L A R M  --  CR = Don`t care"
930 PRINT
940 INPUT "DAY OF WEEK (1-7 SUN=1):";LDOW$:LDOW=VAL(LDOW$)
950   IF LEN(LDOW$)=0 GOTO 970
960   IF LDOW<1 OR LDOW>7 THEN  PRINT CHR$(11) CHR$(23);:GOTO 940
970 INPUT "           MONTH (1-12):";LMTH$:LMTH=VAL(LMTH$)
980   IF LEN(LMTH$)=0 GOTO 1000
```

```
990   IF LMTH<1 OR LMTH>12 THEN PRINT CHR$(11) CHR$(23);:GOTO 970
1000 INPUT "    DAY OF MONTH (1-31):";LDOM$:LDOM=VAL(LDOM$)
1010   IF LEN(LDOM$)=0 GOTO 1030
1020   IF LDOM<1 OR LDOM>31 THEN PRINT CHR$(11) CHR$(23);:GOTO 1000
1030 PRINT
1040 INPUT "           HOUR (0-23):";LHR$:LHR=VAL(LHR$)
1050   IF LEN(LHR$)=0 GOTO 1070
1060   IF LHR<0 OR LHR>23 THEN PRINT CHR$(11) CHR$(23);:GOTO 1040
1070 INPUT "         MINUTES (0-59):";LMIN$:LMIN=VAL(LMIN$)
1080   IF LEN(LMIN$)=0 GOTO 1100
1090   IF LMIN<0 OR LMIN>59 THEN PRINT CHR$(11) CHR$(23);:GOTO 1070
1100 INPUT "         SECONDS (0-59):";LSEC$ :LSEC=VAL(LSEC$)
1110   IF LEN(LSEC$)=0 GOTO 1130
1120   IF LSEC<0 OR LSEC>59 THEN PRINT CHR$(11) CHR$(23);:GOTO 1100
1130 PRINT: INPUT " ALARM MESSAGE (MAX.50):";MESSAGE$
1140 ON LDOW GOSUB 650,660,670,680,690,700,710
1150 ON LMTH GOSUB 750,760,770,780,790,800,810,820,830,840,850,860
1160 GOSUB 1610 : GOSUB 1660    'CURSOR POS. & CLEAR END OF SCREEN
1170 REM
1180 REM DISPLAY ALARM TIME & DATE
1190 REM ---------------------
1200 LI=3 : PO=0
1210 GOSUB 1610 :PRINT CHR$(23)      'CURSORPOS. & CLEAR LINE
1220 PRINT "ALARM:",
1230   IF LEN(LDOW$)=0 THEN PRINT "---,",ELSE PRINT DOW$;",",
1240   IF LEN(LMTH$)=0 THEN PRINT "---",ELSE PRINT MTH$,
1250   IF LEN(LDOM$)=0 THEN PRINT " --",ELSE PRINT " ";LDOM$,
1260 LI=4: PO=61 : GOSUB 1610        'CURSOR POS.
1270   IF LEN(LHR$)=0 THEN PRINT USING "& :";"--",
       ELSE PRINT USING "## :";VAL(LHR$),
1280   IF LEN(LMIN$)=0 THEN PRINT USING " & :";"--",
       ELSE PRINT USING " ## :";VAL(LMIN$),
1290   IF LEN(LSEC$)=0 THEN PRINT USING " &";"--",
       ELSE PRINT USING " ##";VAL(LSEC$);
1300 REM
1310 REM CONVERT DATA TO BCD, DETERMINE (DON'T CARE) LOCATIONS
1320 REM ---------------------------------------------------
1330   IF LEN(LMTH$)=0 THEN LMTH=204 ELSE LMTH=FNTOBCD(LMTH)
1340   IF LEN(LDOW$)=0 THEN LDOW=204 ELSE LDOW=FNTOBCD(LDOW)
1350   IF LEN(LDOM$)=0 THEN LDOM=204 ELSE LDOM=FNTOBCD(LDOM)
1360   IF LEN(LHR$)=0 THEN LHR=204 ELSE LHR=FNTOBCD(LHR)
1370   IF LEN(LMIN$)=0 THEN LMIN=204 ELSE LMIN=FNTOBCD(LMIN)
```

```
1380   IF LEN(LSEC$)=0 THEN LSEC=204 ELSE LSEC=FNTOBCD(LSEC)
1390 REM
1400 REM SET TIMES AND (DON'T CARES) INTO THE RTC LATCHES
1410 REM ------------------------------------------------
1420  OUT BADD,2 : OUT BADD+6,LSEC
1430               OUT BADD+7,LMIN
1440  OUT BADD,3 : OUT BADD+4,LHR
1450               OUT BADD+5,LDOW
1460               OUT BADD+6,LDOM
1470               OUT BADD+7,LMTH
1480 FLAG=0 : GOTO 250     'WAIT FOR NEXT SEC
1490 REM
1500 REM OUTPUT MESSAGE
1510 REM ---------------
1520 LI=20: PO=0
1530 GOSUB 1610 :PRINT ;MESSAGE$;
1540 RESTORE
1550 FOR I=1 TO 4
1560 READ TRE,VOL
1570 PRINT CHR$(27);CHR$(77);CHR$(32+TRE);CHR$(32+VOL);
1580 DATA 20,10,30,10,40,00,30,5
1590 NEXT
1600 RETURN
1610 REM
1620 REM CURSOR POSITION
1630 REM ---------------
1640 PRINT CHR$(27);CHR$(61);CHR$(32+LI);CHR$(32+PO);
1650 RETURN
1660 PRINT CHR$(27);CHR$(121)
1670 RETURN
1680 END
```

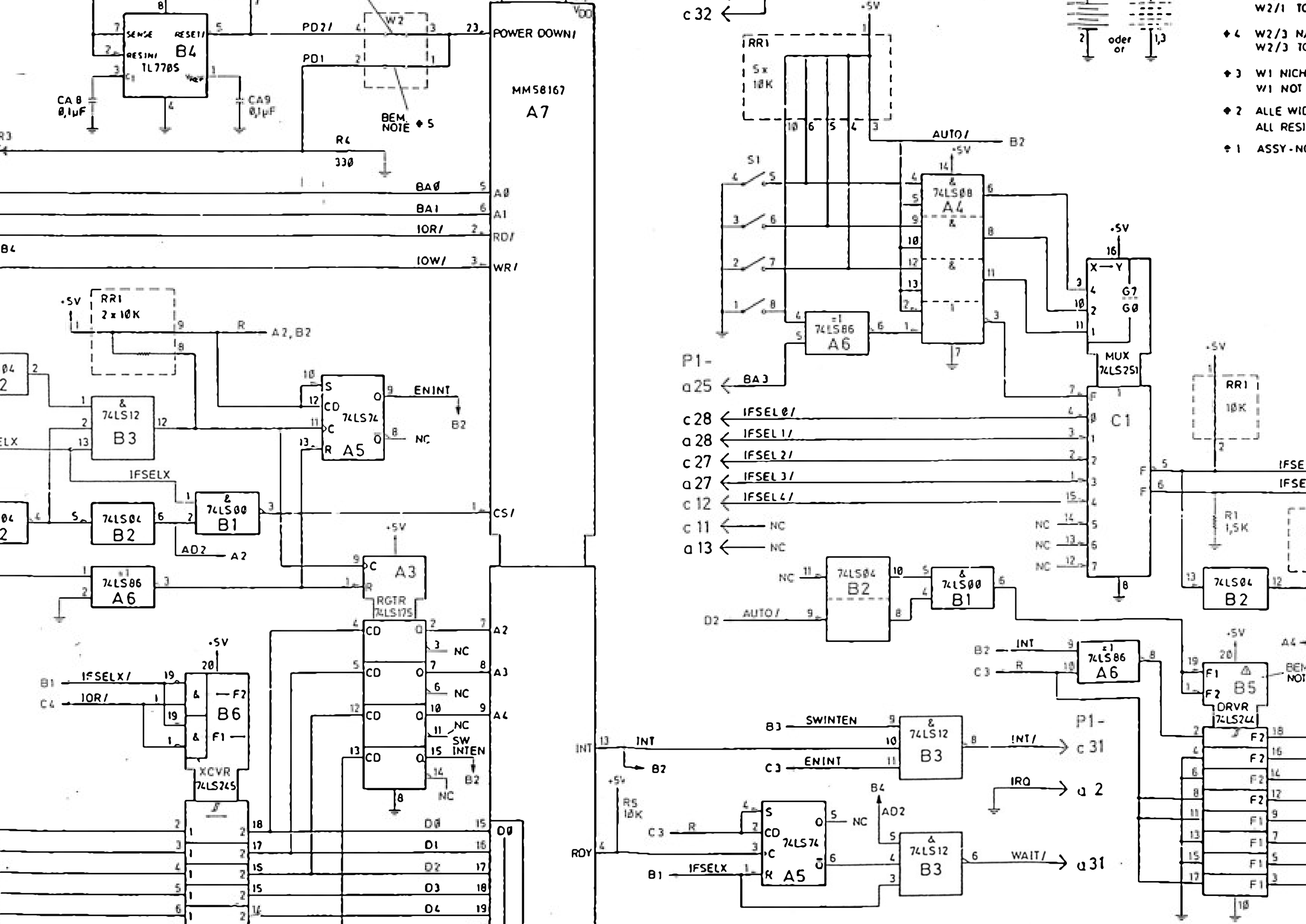
INFORMATION FÜR ASSEMBLER PROGRAMMIERER

Das Lesen des Interrupt-(Unterbrechungs-)Status-Registers löscht dieses Register. Das Interrupt-Status-Register könnte genau dann gelesen werden, wenn ein Interrupt auftritt. Das kann eine Störung verursachen und der Interrupt könnte vom Lesekommando übersehen werden. Vergewissern Sie sich, daß in Assemblerprogrammen das Interrupt-Status-Register dann gelesen wird, wenn kein Interrupt erwartet wird.

Beispiel:

```
INSTAT    LESE ZÄHLER 1/1000 SEK. IN ALPHA
          LESE UMSTELLUNGKONTROLLBIT (ROLLOVER-BIT)
          WENN UMSTELLUNGKONTROLLBIT = 1, SPRUNG NACH INSTAT
LOOP      LESE ZÄHLER 1/1000 SEK. IN BETA
          LESE UMSTELLUNGKONTROLLBIT
          WENN UMSTELLUNGKONTROLLBIT = 1, SPRUNG NACH LOOP
          WENN ALPHA = BETA, SPRUNG NACH LOOP
          WARTE 0,1 MS.  *
          LESE INTERRUPT STATUS REGISTER
```

* = Nächster Interrupt kann nach 0,9 ms auftreten.



B4
TL7705
SENSE
RESET/
RESIN/
CA8 0,1µF
CA9 0,1µF
PD2/
PD1
W2
BEM. NOTE ◆5
R4 330
POWER DOWN/
MM58167
A7
BA0
BA1
IOR/
IOW/
A0
A1
RD/
WR/
CS/
RR1 2 x 10K
R A2, B2
74LS12 B3
74LS74 A5
ENINT
NC
IFSELX
74LS00 B1
74LS04 B2
AD2 A2
74LS86 A6
A3
RGTR 74LS175
CD
SW INTEN
XCVR 74LS245
B6
IFSELX/
IOR/
D0
D1
D2
D3
D4
INT
RDY
c 32
RR1 5x 10K
AUTO/
S1
74LS08 A4
MUX 74LS251
C1
P1-
a 25
BA3
c 28
IFSEL0/
a 28
IFSEL1/
c 27
IFSEL2/
a 27
IFSEL3/
c 12
IFSEL4/
c 11
a 13
R1 1,5K
DRVR 74LS244
B5
SWINTEN
INT/
c 31
IRQ
a 2
R5 10K
WAIT/
a 31
oder or
◆4 W2/3 NA W2/3 TO
◆3 W1 NICH W1 NOT
◆2 ALLE WID ALL RESI
◆1 ASSY-NO
W2/1 TO