Motorola MVME 143S-2

# CPU Controller

**In Stock**

**Used and in Excellent Condition**

**Open Web Page**

https://www.artisantg.com/97359-15



ARTISAN® TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

**Artisan Technology Group**
(217) 352-9330 | sales@artisantg.com | artisantg.com

- Critical and expedited services
- In stock / Ready-to-ship
- We buy your excess, underutilized, and idle equipment
- Full-service, independent repair center

# MVME143S
# MPU VMEmodule™
# User's Manual
## (MVME143S/D1)

# APPENDIX A
# EIA-232-D INTERCONNECTIONS

The EIA-232-D standard is the most widely used interface between terminals and computers or modems, and yet it is not fully understood. This is because all the lines are not clearly defined, and many users do not see the need to conform for their applications. A system should easily connect to any other. Many times designers think only of their own equipment, but the state-of-the-art is computer-to-computer or computer-to-modem operation.

The EIA-232-D standard was originally developed by the Bell System to connect terminals via modems. Therefore, several handshaking lines were included. In many applications these are not needed, but since they permit diagnosis of problems, they are included in many applications.

The standard EIA-232-D interconnections are listed in Table A-1. To interpret this information correctly it is necessary to know that EIA-232-D is intended to connect a terminal to a modem. When computers are connected to computers without modems, one of them must be configured as a terminal and the other as a modem. Because computers are normally configured to work with terminals, they are said to be configured as a modem. Also, the signal levels must be between +3 and +15 volts for a high level, and between -3 and -15 volts for a low level. Any attempt to connect units in parallel may result in out of range voltages and is not allowed by the EIA-232-D specification.

**Table A-1.** EIA-232-D Interconnections

| Pin Number | Signal Mnemonic | Signal Name and Description |
|---|---|---|
| 1 | | Not used. |
| 2 | TxD | TRANSMIT DATA — data to be transmitted is furnished on this line to the modem from the terminal. |
| 3 | RxD | RECEIVE DATA — data which is demodulated from the receive line is presented to the terminal by the modem. |

**Table A-1.** EIA-232-D Interconnections (cont'd)

| Pin Number | Signal Mnemonic | Signal Name and Description |
|---|---|---|
| 4 | RTS | REQUEST TO SEND — RTS is supplied by the terminal to the modem when required to transmit a message. With RTS off, the modem carrier remains off. When RTS is turned on, the modem immediately turns on the carrier. |
| 5 | CTS | CLEAR TO SEND — CTS is a function supplied to the terminal by the modem which indicates that it is permissible to begin transmission of a message. When using a modem, CTS follows the off-to-on transition of RTS after a time delay. |
| 6 | DSR | DATA SET READY — data set ready is a function supplied by the modem to the terminal to indicate that the modem is ready to transmit data. |
| 7 | SIG-GND | SIGNAL GROUND — common return line for all signals at the modem interface. |
| 8 | DCD | DATA CARRIER DETECT — sent by the modem to the terminal to indicate that a valid carrier is being received. |
| 09-14 | | Not used. |
| 15 | TxC | TRANSMIT CLOCK — this line clocks output data to the modem from the terminal. |
| 16 | | Not used. |
| 17 | RxC | RECEIVE CLOCK — this line clocks input data from a terminal to a modem. |
| 18,19 | | Not used. |
| 20 | DTR | DATA TERMINAL READY — a signal from the terminal to the modem indicating that the terminal is ready to send or receive data. |

**Table A-1.** EIA-232-D Interconnections (cont'd)

| Pin Number | Signal Mnemonic | Signal Name and Description |
|---|---|---|
| 21 | | Not used. |
| 22 | RI | RING INDICATOR — RI is sent by the modem to the terminal. This line indicates to the terminal that an incoming call is present. The terminal causes the modem to answer the phone by carrying DTR true while RI is active. |
| 23 | | Not used. |
| 24 | TxC | TRANSMIT CLOCK — same as TxC on pin 15. |
| 25 | BSY | BUSY — a positive EIA signal applied to this pin causes the modem to go off-hook and make the associated phone busy. |

**NOTES:** 1. High level = +3 to +15 volts. Low level = -3 to -15 volts.

2. EIA-232-D is intended to connect a terminal to a modem. When computers are connected to computers without modems, one of the computers must be configured as a modem and the other as a terminal.

There are several levels of conformance that are appropriate for typical EIA-232-D interconnections. The bare minimum requirement is the two data lines and a ground. The full version of EIA-232-D requires 12 lines and accommodates automatic dialing, automatic answering, and synchronous transmission. A middle-of-the-road approach is illustrated in Figure A-1.

One set of handshaking signals frequently implemented are RTS and CTS. CTS is used in many systems to inhibit transmission until the signal is high. In the modem application, RTS is turned around and returned as CTS after 150 microseconds. RTS is programmable in some systems to work with the older type 202 modem (half duplex). CTS is used in some systems to provide flow control to avoid buffer overflow. This is not possible if modems are used. It is usually necessary to make CTS high by connecting it to RTS or to some source of +12 volts such as the resistors shown in Figure A-2. It is also frequently jumpered to an MC1488 gate which has its inputs grounded (the gate is provided for this purpose). Another signal used in many systems is DCD. The original purpose of this signal was to tell the system that the carrier tone from the distant modem

was being received. This signal is frequently used by the software to display a message like CARRIER NOT PRESENT to help the user to diagnose failure to communicate. Obviously, if the system is designed properly to use this signal, and it is not connected to a modem, the signal must be provided by a pullup resistor or gate as described before (see Figure A-1). Many modems expect a DTR high signal and issue a DSR. These signals are used by software to help prompt the operator about possible causes of trouble. The DTR signal is used sometimes to disconnect the phone circuit in preparation for another automatic call. It is necessary to provide these signals in order to talk to all possible modems (see Figure A-2). Figure A-1 is a good minimum configuration that almost always works. If the CTS and DCD signals are not received from the modem, the jumpers can be moved to artificially provide the needed signal. A way that an EIA-232-D connector can be wired to enable a computer to connect to a basic terminal with only three wires is shown in Figure A-2. This is because most terminals have a DTR signal that is ON, and that can be used to pullup the CTS, DCD, and DSR signals. Two of these connectors wired back-to-back can be used. It must be realized that all the handshaking has been bypassed and possible diagnostic messages do not occur. Also the Tx and Rx lines may have to be crossed since Tx from a terminal is outgoing but the Tx line on a modem is an incoming signal.

Another subject that needs to be considered is the use of ground pins. There are two pins labeled GND. Pin 7 is the SIGNAL GROUND and must be connected to the distant device to complete the circuit. Pin 1 is the CHASSIS GROUND, but it must be used with care. The chassis is connected to the power ground through the green wire in the power cord and must be connected to the chassis to be in compliance with the electrical code. The problem is that when units are connected to different electrical outlets, there may be several volts difference in ground potential. If pin 1 of the devices are interconnected with a cable, several amperes of current could result. This not only may be dangerous for the small wires in a typical cable, but could result in electrical noise that could cause errors. That is the reason that Figure A-1 shows no connection for pin 1. Normally, pin 7 should only be connected to the CHASSIS GROUND at one point, and if several terminals are used with one computer, the logical place for that point is at the computer. The terminals should not have a connection between the logic ground return and the chassis.

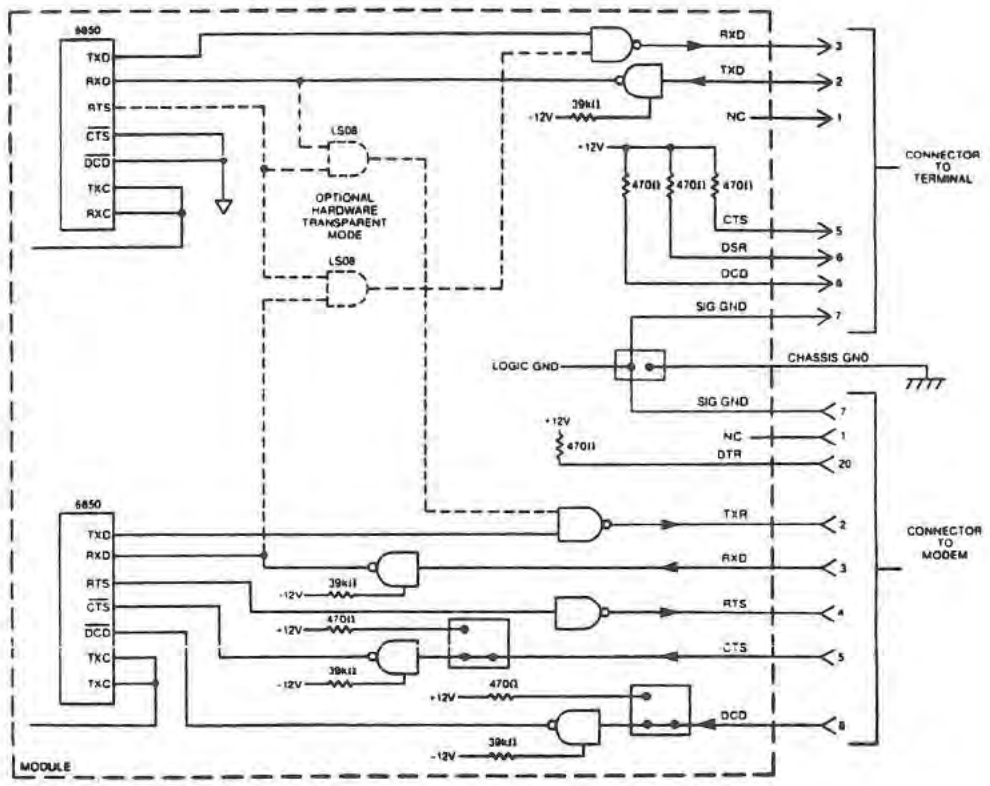



**Figure A-1.** Middle-of-the-Road EIA-232-D Configuration

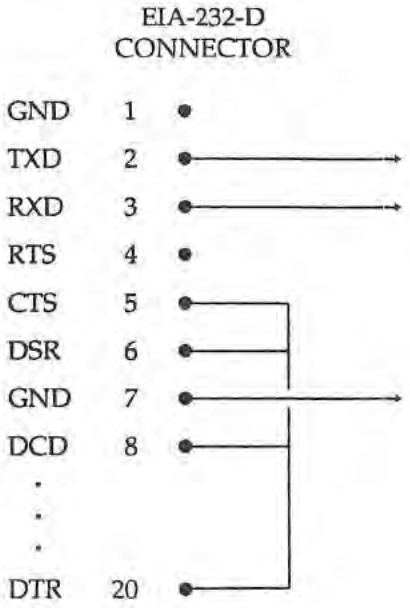



**Figure A-2.** Minimum EIA-232-D Connection

# APPENDIX B
# Z8530 SCC SERIAL PORT B SETUP EXAMPLE

B

This example sets up port B (the EIA-232-D port) of the Z8530 SCC as follows:

- 9600 baud, asynchronous only
- Interrupt on received character, transmitter buffer ready, and external status change with common interrupt vector.

SETUP:

| | |
|---|---|
| Move #$30 into SCCB_WR0 ($FFFA0000) | Clear receiver error status. |
| Move #$10 into SCCB_WR0 ($FFFA0000) | Clear external status interrupts. |
| Move #$09 into SCCB_WR0 ($FFFA0000)<br>Move #$40 into SCCB_WR0 ($FFFA0000) | Select register 9.<br>Reset channel B. |
| Move #$0A into SCCB_WR0 ($FFFA0000)<br>Move #$00 into SCCB_WR0 ($FFFA0000) | Select register 10.<br>Make sure NRZ format is set. |
| Move #$0E into SCCB_WR0 ($FFFA0000)<br>Move #$82 into SCCB_WR0 ($FFFA0000) | Select register 14.<br>Disable baud rate generator. |
| Move #$04 into SCCB_WR0 ($FFFA0000)<br>Move #$44 into SCCB_WR0 ($FFFA0000) | Select register 4.<br>Divide by 16, no parity, one stop bit. |
| Move #$03 into SCCB_WR0 ($FFFA0000)<br>Move #$C1 into SCCB_WR0 ($FFFA0000) | Select register 3.<br>Receiver: eight bits, receiver enabled. |
| Move #$05 into SCCB_WR0 ($FFFA0000)<br>Move #$EA into SCCB_WR0 ($FFFA0000) | Select register 5.<br>Transmitter: eight bits, transmitter enabled, RTS on, DTR on. |
| Move #$0C into SCCB_WR0 ($FFFA0000)<br>Move #$02 into SCCB_WR0 ($FFFA0000) | Select register 12.<br>Lower byte of time constant. |
| Move #$0D into SCCB_WR0 ($FFFA0000)<br>Move #$00 into SCCB_WR0 ($FFFA0000) | Select register 13.<br>Higher byte of time constant. |

| | |
|---|---|
| Move #$0B into SCCB_WR0 ($FFFA0000)<br>Move #$56 into SCCB_WR0 ($FFFA0000) | Select register 11.<br>RX clock = BR Generator output, TX clock = BR Generator output, TRXC = output = BR Generator output. |
| Move #$0E into SCCB_WR0 ($FFFA0000)<br>Move #$81 into SCCB_WR0 ($FFFA0000) | Select register 14.<br>BR Generator clock source = RTXC pin. |
| Move #$01 into SCCB_WR0 ($FFFA0000)<br>Move #$11 into SCCB_WR0 ($FFFA0000) | Select register 1.<br>Interrupt on all Received Character or Special Condition. Also enable external interrupts. |
| Move #$0F into SCCB_WR0 ($FFFA0000)<br>Move #$80 into SCCB_WR0 ($FFFA0000) | Select register 15.<br>Enable Break/Abort interrupts. |
| Move #$02 into SCCB_WR0 ($FFFA0000)<br>Move #$80 into SCCB_WR0 ($FFFA0000) | Select register 2.<br>Interrupt vector number. ($80 => vector offset = $200.) |
| Move #$09 into SCCB_WR0 ($FFFA0000)<br>Move #$08 into SCCB_WR0 ($FFFA0000) | Select register 9.<br>Master interrupt enable. Status information NOT to be included in the vector passed to the MPU. |

To minimize overhead in the interrupt handling routine, status information may be selected to be included in the vector(s). The vector, then, points directly at the appropriate handling routine according to the interrupt cause. If the Vector-Status-Include (VSI) is set and the content in the vector register is $80, then the vector passed to the MPU is:

$80 (vector offset = $200) for Channel B Transmitter Buffer Empty or
$82 (vector offset = $208) for Channel B External Status Change or
$84 (vector offset = $210) for Channel B Received Character Available or
$86 (vector offset = $218) for Channel B Special Received Character.

For this example, place the address of the common interrupt handler at offset $200 in the vector table.

INTERRUPT HANDLER:

| | | |
|---|---|---|
| Move #$03 into SCCB_WR0 | ($FFFA0000) | Select register 3. |
| Read from SCCB_RR0 | ($FFFA0000) | Read the Read Register 3 for interrupt cause. |

Investigate the interrupt pending bits to determine the cause.
Branch to the appropriate handling routine.

TRANSMIT A CHARACTER:

If Transmitter Buffer Empty interrupt is desired, it must be enabled before outputting a character or else the interrupt will not occur.

| | | |
|---|---|---|
| Move #$01 into SCCB_WR0 | ($FFFA0000) | Select register 1. |
| Move #$13 into SCCB_WR0 | ($FFFA0000) | Enable transmitter interrupt. |
| Move output character into SCCB_TDR | ($FFFA0001) | Transmit a character. |

TRANSMITTER BUFFER EMPTY INTERRUPT HANDLER:

| | | |
|---|---|---|
| Move #$01 into SCCB_WR0 | ($FFFA0000) | Select register 1. |
| Move #$11 into SCCB_WR0 | ($FFFA0000) | Disable transmitter interrupt. |
| Move #$38 into SCCB_WR0 | ($FFFA0000) | Reset highest Interrupt-Under-Service (IUS). |

Are there more characters to output?
If Yes, go do TRANSMIT A CHARACTER.
If No, return from exception.

RECEIVED CHARACTER INTERRUPT HANDLER:

| | | |
|---|---|---|
| Move #$01 into SCCB_WR0 | ($FFFA0000) | Select register 1. |
| Read from SCCB_RR0 | ($FFFA0000) | Read the Read Register 1 to check for status. |

Check for framing error, receiver overrun, and parity errors.

| | | |
|---|---|---|
| Read from SCCB_RDR | ($FFFA0001) | Read received character. |
| Move #$38 into SCCB_WR0 | ($FFFA0000) | Reset highest IUS. |

EXTERNAL STATUS CHANGE INTERRUPT HANDLER:

| | | |
|---|---|---|
| Break | — | either start of break or end of break. |
| CTS | — | a transition has occurred on the CTS input pin. |
| DCD | — | a transition has occurred on the DCD input pin. |

| | | |
|---|---|---|
| Move #$00 into SCCB_WR0 | ($FFFA0000) | Select register 0. |
| Read from SCCB_RR0 | ($FFFA0000) | Read the Read Register 0 for status. |
| Move #$10 into SCCB_WR0 | ($FFFA0000) | Reset external status interrupt. |

Take actions as necessary.

If break bit is low, which is the end of a break, a null character is still in the receive buffer. It should be read and discarded.

| | | |
|---|---|---|
| Read data from SCCB_RDR | ($FFFA0001) | Read null character. |

Return from exception.

# APPENDIX C
# MC68901 MFP TIMER A SETUP EXAMPLE

C

The following example sets up the MC68901 MFP timer A (software tick timer) to interrupt the MPU periodically every 10 milliseconds.

SETUP:

| | | |
|---|---|---|
| Clear bit #5 of MFP_IERA | ($FFF80007) | Disable timer A interrupts. |
| Move #$10 into MFP_TACR | ($FFF80019) | Reset and stop timer A. |
| Move #$7B into MFP_TADR | ($FFF8001F) | Load count down value. (Refer to Table 1 in this Appendix.) |
| Move #$06 into MFP_TACR | ($FFF80019) | Delay mode, prescaler = 100. |
| Move #$68 into MFP_VR | ($FFF80017) | Set starting vector at $60. Set software interrupt mode. |

NOTE

The vector passed to the MPU for the timer A interrupt is $6D => vector offset = 4 x $6D = $1B4.

| | | |
|---|---|---|
| Move #$DF into MFP_IPRA | ($FFF8000B) | Clear timer A interrupt pending bit (bit #5 of IPRA). |
| Move #$DF into MFP_ISRA | ($FFF8000F) | Clear timer A interrupt-in-service bit (bit #5 of ISRA). |
| Set bit #5 of MFP_IMRA | ($FFF80013) | Unmask timer A interrupts. |
| Set bit #5 of MFP_IERA | ($FFF80007) | Enable timer A interrupts. |

TIMER A INTERRUPT HANDLER:

Read MFP_ISRA ($FFF8000F) Read interrupt-in-service register A.

C

Investigate MFP_ISRA to determine if it was actually from timer A.

Take actions as necessary.

Move **#$DF** into MFP_ISRA ($FFF8000F) Clear timer A interrupt-in-service bit (bit #5 of ISRA).

Return from exception.

COUNTDOWN CALCULATION:

The countdown value used during setup may be calculated using the following equation:

$$CD = (TI \times TO) / PS$$

where: CD = countdown value to be loaded into timer data register.

TI = timer input frequency in Hz = 1,230,769 Hz.

TO = tick timer interrupts interval in seconds.

PS = prescaler value (4, 10, 16, 50, 64, 100, or 200).

Table D-1 contains the values for PS and CD for some selected interrupts intervals.

**Table C-1.** Prescaler and Countdown Values

| TO | | | CD | |
|---|---|---|---|---|
| MSEC | SEC | PS | HEX | DECIMAL |
| 1.0 | 0.0010 | 10 | $7B | 123 |
| 5.0 | 0.0050 | 50 | $7B | 123 |
| 10.0 | 0.0100 | 100 | $7B | 123 |
| 20.0 | 0.0200 | 100 | $F6 | 246 |
| 40.0 | 0.0400 | 200 | $F6 | 246 |
| 41.6 | 0.0416 | 200 | $00 | 256 |

# INDEX

When using this index, keep in mind that a page number indicates only where referenced material begins; it may extend to the following page or pages.

## D



## E



## F



## G



## H



## I

## L



## M



## N



## O



## P

## R

## S



## T



## U



## V

## W



## Z

# Notes

# Notes

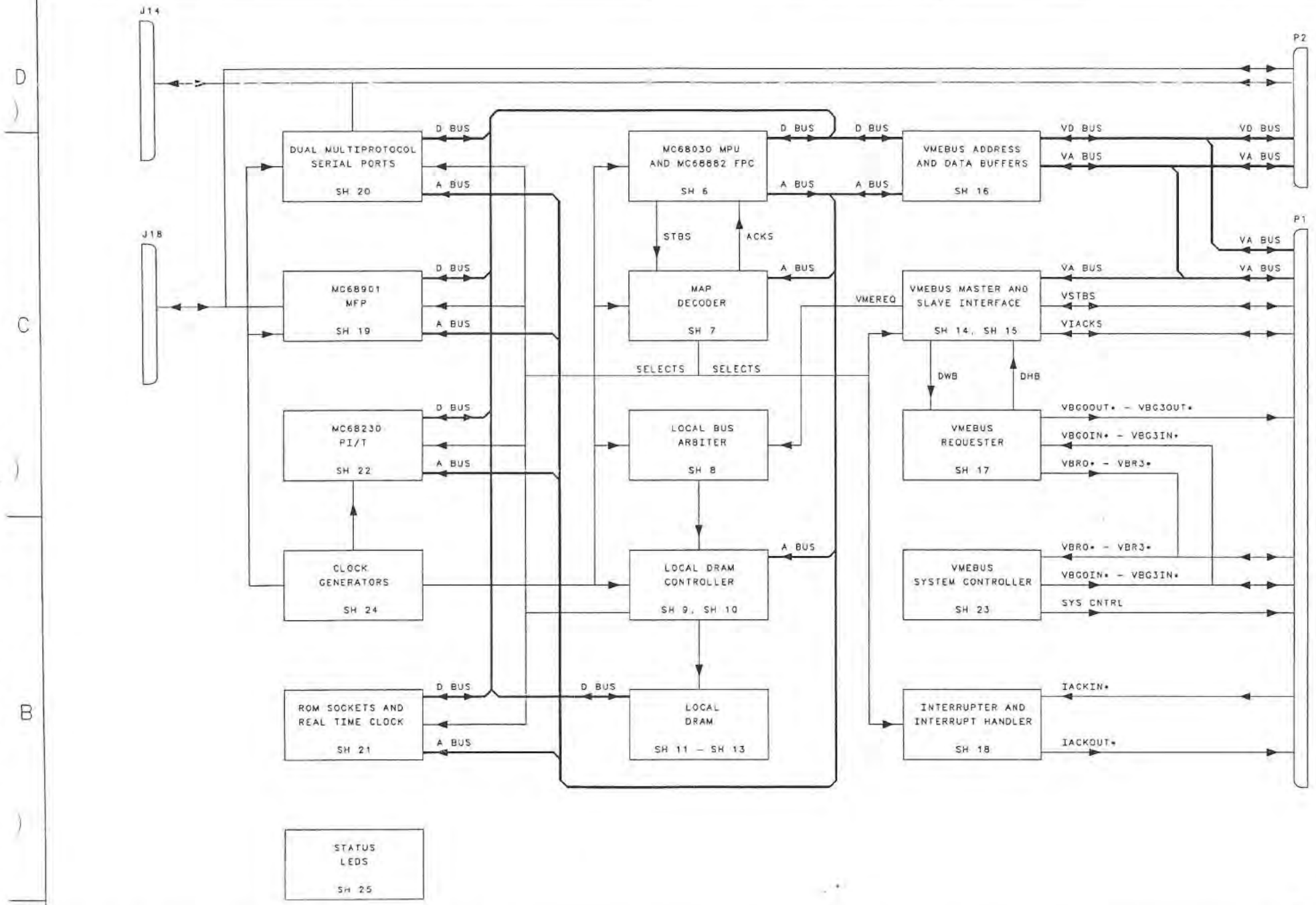



**Figure 4-4.** MVME143S Bloc

VID3-VID0 VID3-VID0 defines the lower four bits of the status ID of the MVME143S VMEbus interrupter. The upper four bits are always high. The status ID of the MVME143S VMEbus interrupter is affected by these control bits as follows:

| VID3 | VID2 | VID1 | VID0 | Status ID |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $F0 |
| 0 | 0 | 0 | 1 | $F1 |
| 0 | 0 | 1 | 0 | $F2 |
| 0 | 0 | 1 | 1 | $F3 |
| 0 | 1 | 0 | 0 | $F4 |
| 0 | 1 | 0 | 1 | $F5 |
| 0 | 1 | 1 | 0 | $F6 |
| 0 | 1 | 1 | 1 | $F7 |
| 1 | 0 | 0 | 0 | $F8 |
| 1 | 0 | 0 | 1 | $F9 |
| 1 | 0 | 1 | 0 | $FA |
| 1 | 0 | 1 | 1 | $FB |
| 1 | 1 | 0 | 0 | $FC |
| 1 | 1 | 0 | 1 | $FD |
| 1 | 1 | 1 | 0 | $FE |
| 1 | 1 | 1 | 1 | $FF |



### Control/Status Register 4: Port C Pins

The MVME143S utilizes the PI/T port C pins to form Control/Status Register 4 (CSR4). Five port C pins are used as control outputs and three are used for interrupt request and acknowledge functions. PC6, PC5, and PC3 are assigned as port interrupt acknowledge pin, port interrupt request pin, and timer interrupt request pin, respectively. After a reset, the PI/T makes all parallel pins inputs. Therefore, software must initialize the control bits and make them outputs after each reset. All port C pins default to high when they are not programmed as outputs. The assignment of PC7-PC0 is as follows:

| Address | PC7 | PC6 | PC5 | PC4 | PC3 | PC2 | PC1 | PC0 |
|---|---|---|---|---|---|---|---|---|
| FFF90019 | LTOEN | PIACK* | PIRQ* | GTOEN | TEIRQ* | WWPAR* | PAREN* | VPAREN* |
| | R/W | | | R/W | R/W | | R/W | R/W |

LTOEN When LTOEN is low, the local timeout is disabled and the MVME143S hangs if a non-mapped location is accessed or if its requester is prevented from obtaining the VMEbus mastership. When LTOEN is high, the local bus timer is enabled. The timeout period is selectable through header J11.

PIACK* PI/T port interrupt acknowledge. This pin is asserted low by the MVME143S interrupt handler in response to a PI/T port interrupt request (PIRQ* = 0). When the MPU performs an interrupt acknowledge cycle to a PIRQ*, the PI/T places the programmed vector with the source encoded to the data bus.

PIRQ* PI/T port interrupt request. The PI/T may be programmed to assert PIRQ* low upon detecting the assertion of the ABORT switch or an asserted edge on parity error, VMEbus error, or local timeout. Each one of these four sources may be enabled/disabled individually in the PI/T.

GTOEN When GTOEN is high, the MVME143S asserts VBERR* to the VMEbus if VDS0* and/or VDS1* is low for more than timeout period if the MVME143S is the system controller. Also, the arbiter rearbitrates if VBBSY* remains high after a grant has been issued for more than the timeout period. When GTOEN is low, global timeout and arbitration timeout are disabled. The global timeout period is selectable through header J5.

TEIRQ* PI/T timer interrupt request. This pin can be cleared by writing a 1 to the ZDS bit in Timer Status Register (TSR). Only auto-vectored interrupt is supported for the PI/T timer. After a reset, software must initialize the PI/T for the desired timer operation.

WWPAR* This control bit may be used to force wrong parity to be written to the onboard DRAM for testing purposes. Special care must be taken in using this bit because both the MPU and the VMEbus may write wrong parity to the DRAM when WWPAR* is low.

PAREN* When this bit is low, parity checking is enabled and the MVME143S asserts BERR* to the MPU if a parity error is detected on a DRAM read cycle. DRAM access time is increased by 1 clock for all read accesses by the MPU when PAREN* is low.

VPAREN* When this bit is low, parity checking is enabled and the MVME143S asserts VBERR* to the VMEbus if a parity error is detected on a read cycle. DRAM access time is increased by 1 clock for all read accesses by the VMEbus when VPAREN* is low.

### MC68230 PI/T Registers

The MVME143S requires that the PI/T is programmed for Mode 00 and Submode 1X for proper operation. Some of the PI/T registers are described in the following paragraphs. Refer to the *MC68230 Data Book* for further details.

### Port General Control Register (PGCR)



PGCR is used to select mode 00 and to define ABORT, PARERR, VMEBERR, and LTOBERR signals as high true status inputs. Only bits 5 and 4 are user-programmable. Bit 5 must be set if either software abort or PARERR interrupt is to be enabled. Bit 4 must be set to allow interrupt functions for VMEBERR and/or LTOBERR.

| Address | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|
| FFF90001 | MODE1 | MODE0 | H34EN | H12EN | H4SNS | H3SNS | H2SNS | H1SNS |
| VALUE | 0 | 0 | 0/1 | 0/1 | 1 | 1 | 1 | 1 |

### Port A Control Register (PACR)

PACR is used to select submode 1X and to define VMEBERR and LTOBERR as edge-sensitive status inputs. Only bits 2 and 1 are user-programmable to enable/disable VMEBERR interrupt and LTOBERR interrupt, respectively.

| Address | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|
| FFF9000D | ASUB1 | ASUB0 | H2C2 | H2C1 | H2C0 | VBEIEN | LTOIEN | H1SC |
| VALUE | 1 | 0 | 0 | 0 | 0 | 0/1 | 0/1 | 0 |

VBEIEN When VBEIEN is set, an occurrence of VMEbus error generates a level 7 interrupt via the PI/T. VMEbus error interrupt is disabled if either VBEIEN or H12EN is cleared.

LTOIEN When LTOIEN is set, an occurrence of local bus timeout generates a level 7 interrupt via the PI/T. Local bus timeout interrupt is disabled if either LTOIEN or H12EN is cleared.

### Port B Control Register (PBCR)

PBCR is used to select submode 1X and to define PARERR and ABORT as edge-sensitive status inputs. Only bits 2 and 1 are user-programmable to enable/disable PARERR interrupt and ABORT interrupt, respectively.

| Address | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|
| FFF9000F | BSUB1 | BSUB0 | H4C2 | H4C1 | H4C0 | PARIEN | ABRTEN | H3SC |
| VALUE | 1 | 0 | 0 | 0 | 0 | 0/1 | 0/1 | 0 |

PARIEN When PARIEN is set, an occurrence of MPU-to-DRAM parity error generates a level 7 interrupt via the PI/T. Parity error interrupt is disabled if either PARIEN or H34EN is cleared.

ABRTEN When ABRTEN is set, a level 7 interrupt via the PI/T is generated upon an activation of the MVME143S front panel ABORT switch. Software abort interrupt is disabled if either ABRTEN or H34EN is cleared.

### Port Interrupt Vector Register (PIVR)

| Address | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|
| FFF9000B | PIVR7 | PIVR6 | PIVR5 | PIVR4 | PIVR3 | PIVR2 | PIVR1 | PIVR0 |

PIVR contains the upper order six bits of the four port interrupt vectors. The contents of this register may be read in two ways: by an ordinary read cycle, or by a port interrupt acknowledge bus cycle. When reading this register with an ordinary read cycle, the lower two bits are always forced to zero. During a port interrupt acknowledge cycle, bits 1 and 0 encode the interrupt sources as follows:

| SOURCE | PIVR1 | PIVR0 |
|---|---|---|
| PARERR | 1 | 1 |
| ABORT | 1 | 0 |
| VMEBERR | 0 | 1 |
| LTOBERR | 0 | 0 |

### Port Status Register (PSR)

| Address | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---|---|---|---|---|---|---|---|---|
| FFF9001B | PARERR Level | ABORT Level | VMEBERR Level | LTOBERR Level | PARERR Status | ABORT Status | VMEBERR Status | LTOBERR Status |
| | R | R | R | R | R/C | R/C | R/C | R/C |



The PSR contains information about software abort and bus error signals activity. Bits 7, 5, and 4 are always read as zero; while bit 6 reflects the instantaneous level of the ABORT signal. Bits 3-0 contain the latched status of PARERR, ABORT, VMEBERR, and LTOBERR, respectively. These bits are set by the assertion edge of the input signals. When set, they can only be cleared by a board reset or by writing a 1 to the set bits.

### Programmed Values for PI/T Registers

The values of the PI/T registers after reset and the recommended programmed values for proper, intended operation are shown in Table 4-5.

**Table 4-5.** Programmed Values for PI/T Registers

| Register Address | Register Name | Reset Value | Programmed Value | Notes |
|---|---|---|---|---|
| $FFF90001 | PGCR | $00 | $3F | Mode 00 and H4-H1 = high true. |
| $FFF90003 | PSRR | $00 | $1F | PC4 = PC4, PC5 = PIRQ*, PC6 = PIACK*. |
| $FFF90005 | PADDR | $00 | $FF | PA7-PA0 = outputs. |
| $FFF90007 | PBDDR | $00 | $7F | PB6-PB0 = outputs, PB7 = input. |
| $FFF90009 | PCDDR | $00 | $BF | PC7 and PC5-PC0 = outputs, PC6 = input. |
| $FFF9000B | PIVR | $0F | User defined | Example: if PIVR = $80 then the vectors for PARERR, ABORT, VMEBERR, and LTOBERR are $83, $82, $81, and $80, respectively. |

**Table 4-5.** Programmed Values for PI/T Registers (cont'd)

| Register Address | Register Name | Reset Value | Programmed Value | Notes |
|---|---|---|---|---|
| $FFF9000D | PACR | $00 | $80 | Sub mode 1X, H2S = latched VMEBERR, H1S = latched LTOBERR, VMEBERR interrupt is disabled. LTOBERR interrupt is disabled. |
| $FFF9000F | PBCR | $00 | $82 | Sub mode 1X, H4S = latched PARERR, H3S = latched ABORT, PARERR interrupt is disabled. ABORT interrupt is enabled. |
| $FFF90011 | PADR | No change | $01 | VMED16* = 1, MBXIEN* = 0, VMEA24* = 0, SLAVEN* = 0, SLVA3-SLVA0 = %0000. |
| $FFF90013 | PBDR | No change | $00 | Interrupter is disabled, interrupter status ID = $F0. |
| $FFF90015 | PAAR | N/A | N/A | Reflects current states of pins. |
| $FFF90017 | PBAR | N/A | N/A | Reflects current states of pins. |
| $FFF90019 | PCDR | Unknown | $FF | LTOEN = 1, GTOEN = 1, WWPAR* = 1, PAREN* = 1, VPAREN* = 1. |
| $FFF9001B | PSR | Unknown | N/A | Write a 1 to the bit to clear it. |
| $FFF90021 | TCR | $00 | $F0 | PC3/TOUT = TIRQ* and timer interrupt is enabled for auto-vectored interrupts, counter rolls over on zero detect then continues counting, PC2/TIN = PC2 = WWPAR*, and timer is disabled. |

**Table 4-5.** Programmed Values for PI/T Registers (cont'd)

| Register Address | Register Name | Reset Value | Programmed Value | Notes |
|---|---|---|---|---|
| $FFF90023 | TIVR | $0F | --- | Not used. |
| $FFF90027 | CPRH | No change | User defined | |
| $FFF90029 | CPRM | No change | User defined | |
| $FFF9002B | CPRL | No change | User defined | |
| $FFF9002F | CNTRH | No change | Read only | Stop counter before reading this register. |
| $FFF90031 | CNTRM | No change | Read only | Stop counter before reading this register. |
| $FFF90033 | CNTRL | No change | Read only | Stop counter before reading this register. |
| $FFF90035 | TSR | $00 | N/A | Write a 1 to clear ZDS bit, if set. |



## Z8530 Serial Communications Controller (SCC)

The MVME143S uses the Z8530 Serial Communications Controller (SCC) to implement its two multiprotocol EIA-232-D serial ports (port A = serial port 3 and port B = serial port 2). Headers J20 and J19 allows port 3 and port 2 to be individually configured as DTE or DCE. Refer to Chapter 2 for jumpers configurations. The SCC occupies 128Kb in the MVME143S memory map and is located at a physical base address of $FFFA0000 and interrupts the MPU on level 5. The address map for the SCC is shown in Table 4-6.

**Table 4-6.** SCC Registers Map

| Physical Address | Register Name | Register Description |
|---|---|---|
| $FFFA0000 | SCCB-RR0<br>SCCB-WR0 | Port B read register 0<br>Port B write register 0 |
| $FFFA0001 | SCCB-RDR<br>SCCB-TDR | Port B received data register<br>Port B transmitted data register |
| $FFFA0002 | SCCA-RR0<br>SCCA-WR0 | Port A read register 0<br>Port A write register 0 |
| $FFFA0003 | SCCA-RDR<br>SCCA-TDR | Port A received data register<br>Port A transmitted data register |

In the SCC, register addressing is direct for the data registers only. In all other cases (with the exception of SCCx-WR0 and SCCx-RR0), accessing the internal SCC read and write registers requires a sequence of two operations. The first operation is a write to SCCx-WR0 with the four least significant bits that point to the selected register. If the second operation is a write, then the selected write register is accessed. On the other hand, if the second operation is a read, then the selected read register is selected. The pointer bits are automatically cleared after the second read or write operation so that SCCx-WR0 (or SCCx-RR0) is addressed again on the next access. Refer to the *Z8530 Serial Communications Controller Data Sheet* for details on programming and using the SCC.

Because of its internal structure, there are several means of obtaining the baud rate clocks for each of the two serial channels. Each channel within the SCC has a programmable baud rate generator. The Baud Rate Generator (BRG) input can be from the RTXC input or from PCLK. The hardware on the MVME143S allows the RTXC pin for each channel to be connected to an external clock source or to the onboard 1.230769 MHz or 2.451538 MHz clock (selectable through header J10). Also, the SCC DPLL input can be either the BRG output or the RTXC pin. The DPLL operates at 32 times the data rate for NRZI and at 16 times the data rate for FM. The values in the SCC time constant register that are required to create some common baud rates are shown in Table 4-7.

**Table 4-7.** Port 2/3 Baud Rates Available with BRG Clock = RTXC

| 2.461538 MHz Baud Rate | 1.230769 MHz Baud Rate | Clock Mode | Time Constant Register Value | Percent Error |
|---|---|---|---|---|
| 38400 | 19200 | x16 | 0 | 0.16 |
| 19200 | 9600 | x16 | 2 | 0.16 |
| 9600 | 4800 | x16 | 6 | 0.16 |
| 4800 | 2400 | x16 | $E | 0.16 |
| 2400 | 1200 | x16 | $1E | 0.16 |
| 1200 | 600 | x16 | $3E | 0.16 |
| 600 | 300 | x16 | $7E | 0.16 |
| 220 | 110 | x16 | $15E | 0.67 |
| 128000 | 64000 | x1 | 8 | 3.85 |
| 112000 | 56000 | x1 | 9 | 0.10 |
| 96000 | 48000 | x1 | $B | 1.38 |
| 76800 | 38400 | x1 | $E | 0.16 |

A few of the data rates that are achievable with PCLK = 4.00 are given in Table 4-8.

**Table 4-8.** Port 2/3 Baud Rates Available with BRG Clock = PCLK = 4.00 MHz

| Baud rate | Clock Mode | Time Constant Register Value | Actual Baud Rate | Percent Error |
|---|---|---|---|---|
| N/A | x32 | 0 | 31250 | N/A |
| N/A | x32 | 1 | 20833 | N/A |
| N/A | x32 | 2 | 15625 | N/A |
| 64000 | x16 | 0 | 62500 | 2.3 |
| N/A | x16 | 1 | 41667 | N/A |
| 32000 | x16 | 2 | 31250 | 2.3 |

### Serial Ports Configurations

All of the buffers and the configuration headers are on the MVME143S. Port 3 (SCC port A) and port 2 (SCC port B) may be individually configured using onboard headers J20 and J19, respectively, to be either DTE (to modem) or DCE (to terminal). The MVME143S also provides an external ability to disable the transmit data with the appropriate DTR pins from the SCC. For example, when the DTRB pin of the SCC is high, the TXDB pin is disabled to the EIA-232-D port. When the DTRB pin is low, the TXDB pin is enabled to the EIA-232-D port. Note that the DTRA and DTRB pins also drive DTR3 and DTR2 EIA-232-D signal lines. DTRA and DTRB are set high by a reset to the SCC.



Port 2 and port 3 can be accessed via connector P2 rows A and C or connector J14 through the front panel. An external cable may be connected to P2 and a DB-25 connector crimped directly onto it for each of the serial ports. For shielding purposes, the DB-25 should be mounted to a back panel that is grounded to the chassis, and connection should be made between the back panel and the shielding metal of the DB-25.

## ROM/PROM/EPROM/EEPROM Sockets

The MVME143S has four 32-pin ROM/PROM/EPROM/ EEPROM sockets that are organized as two banks with two sockets per bank. They are arranged as follows:

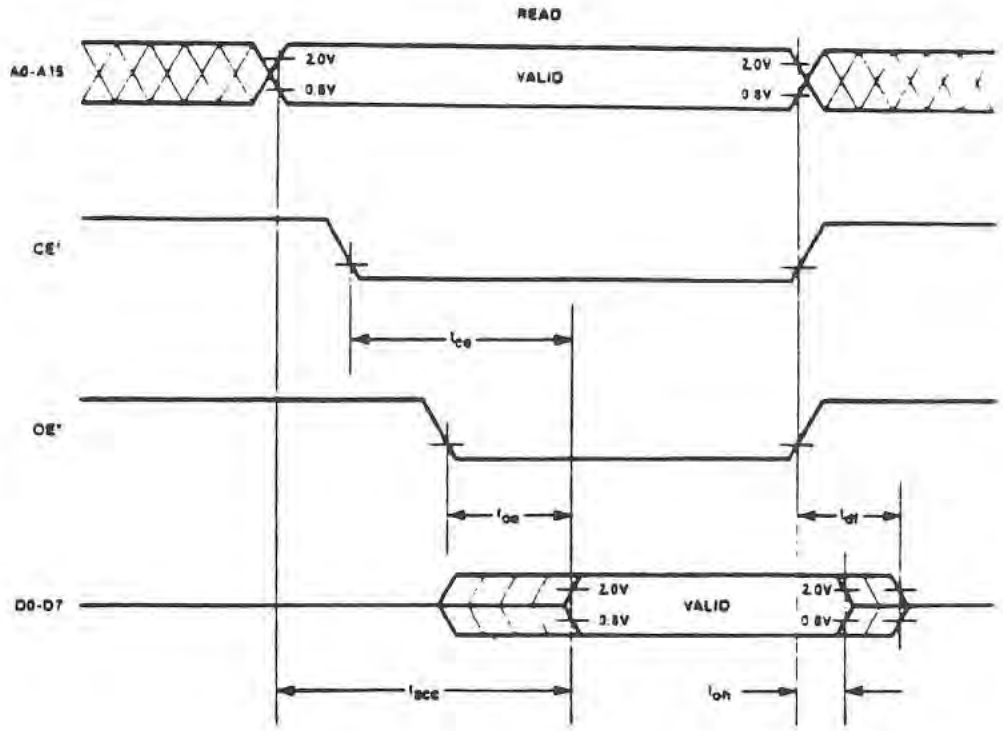
Bank 1: U3 = even, U5 = odd Bank 2: U1 = even, U2 = odd

Each bank appears as a 16-bit word port to the MPU and can be separately configured for 8K x 8, 16K x 8, 32K x 8, 64K x 8, 128K x 8, 256K x 8, 512K x 8, or 1 M x 8 ROM/PROM/EPROM devices; or for 2K x 8, 8K x 8, or 32K x 8 EEPROM devices.

Bank 1 address base is at $FF800000 and bank 2 address base is at $FFA00000. If RAM devices are 64K bit or smaller, the $FFF00000 to $FFF3FFFF range may be used. This allows backward compatibility with the MVME143. Base addresses for bank 1 and bank 2 of 64k bit or smaller devices is $FFF00000 and $FFF20000, respectively.

When 64K bit or smaller devices are used, the address range of $FF800000 to $FFBFFFFF can be mapped as VMEbus A32/D16 (jumper installed on header J13).

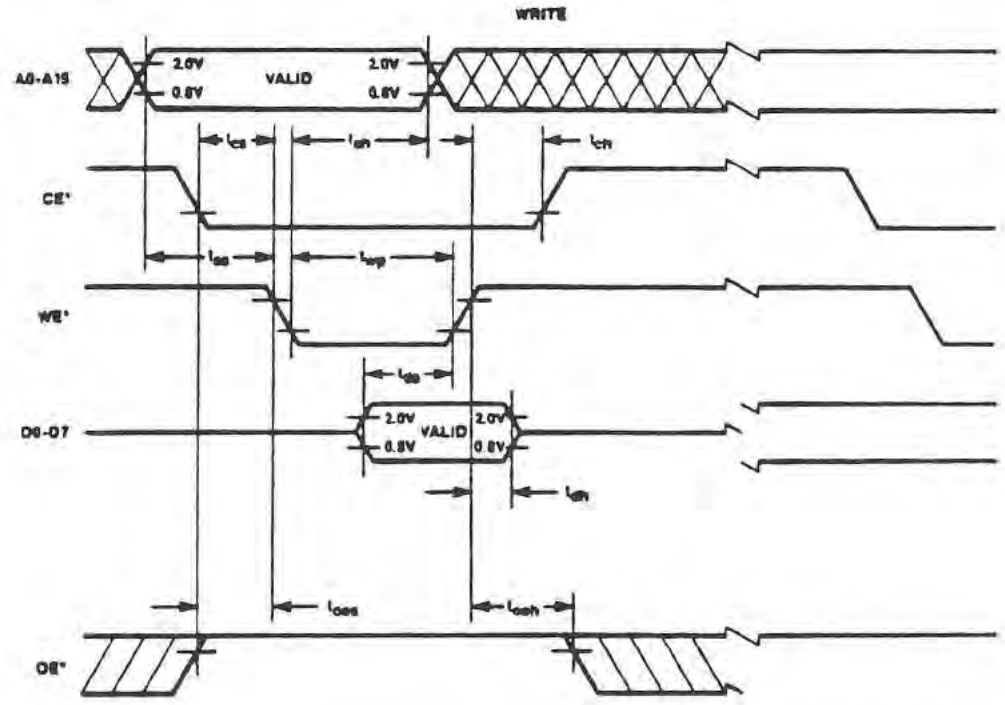
The ROM/PROM/EPROM/EEPROM devices must meet the read timings shown in Figure 4-2. They are guaranteed the write timings shown in Figure 4-3.




| Symbol | Description | Min. | Max. | Unit |
|---|---|---|---|---|
| tacc | Address valid to data valid. | | 230 | ns |
| tce | CE* low to data valid. | | 215 | ns |
| toe | OE* low to data valid. | | 175 | ns |
| toh | Address invalid, CE* or OE* high to data not valid. | 0 | | ns |
| tdf | CE* or OE* high to data high impedance. | | 40 | ns |

**NOTE:** The MVME143S does not guarantee a maximum transition time on address and data lines during the time that CE* is high.

**Figure 4-2.** Read Timings Required by the MVME143S




| Symbol | Description | Min. | Max. | Unit |
|---|---|---|---|---|
| tas | Address valid to WE* low. | 45 | | ns |
| tcs | CE* low to WE* low. | 30 | | ns |
| toes | OE* high to WE* low. | 55 | | ns |
| tah | Address valid after WE* low. | 175 | | ns |
| twp | WE* low pulse width. | 120 | | ns |
| tds | Data valid to WE* high. | 160 | | ns |
| tdh | WE* high to data not valid. | 50 | | ns |
| toeh | WE* high to OE* low. | 130 | | ns |
| tch | WE* high to CE* high. | 30 | | ns |

**NOTE:** The MVME143S does not guarantee a maximum transition time on address and data lines during the time that CE* is high.

**Figure 4-3.** Write Timings Guaranteed by MVME143S

### EEPROM Considerations

There are several considerations to be taken when EEPROMs are to be used on the MVME143S:

1. The MVME143S provides no protection against inadvertent writes to EEPROM that might happen during power on/off transitions. Most devices provide some level of internal protection. In order to gain "absolute protection", devices with additional "software protection" are recommended.
2. When a bank is configured for EEPROM, writes to that bank must always be 16-bit wide. This is because any access to one byte of the bank also causes an access to the other byte; thus, byte-wide write access causes unintended data to be written to the other byte.
3. There are several different algorithms for erasing/writing to EEPROM, depending on the manufacturer. The MVME143S supports only those devices which have a "static RAM" compatible erase/write mechanism.
4. Note that the MVME143S requires that the EEPROM must allow wired-OR on the RDY/BSY* pin (for 2K x 8 and 8K x 8 devices). The MVME143S, however, does not monitor the status of the RDY/BSY* pins.



## Interrupt Handler

The interrupt handler gives the onboard MPU the ability to sense and respond to all onboard interrupts, all seven VMEbus interrupts, VMEbus ACFAIL*, VMEbus SYSFAIL*, and the ABORT switch.

All VMEbus interrupts are enabled/disabled using header J12. All onboard interrupts and VMEbus SYSFAIL* may be enabled/disabled under software control. VMEbus ACFAIL* interrupt is always enabled when MIE* = 0. All interrupts are disabled when the MIE* control bit is high (logic 1).

When the MPU initiates an interrupt acknowledge cycle, the interrupt handler determines the acknowledge level by examining A01-A03. Finally, it asserts AVEC* to indicate to the MPU to generate the interrupt vector internally if the acknowledge cycle was for VMEbus ACFAIL*, PI/T timer, or the mail-box interrupt. If the acknowledge cycle is for the SCC, the MFP, PI/T port interrupt, or the VMEbus, then it initiates a vector fetch cycle to the appropriate device.

If both onboard and VMEbus interrupts are asserted on the same acknowledge level, the interrupt handler acknowledges the onboard interrupt. All the interrupt sources on the MVME143S (in descending order of priority) and the associated interrupt vectors are summarized in Table 4-9.

**Table 4-9.** Interrupt Sources and Vectors

| Interrupt Source | Level | Vector Source | Vector Number | Vector Offset |
|---|---|---|---|---|
| VMEbus ACFAIL* | 7 | Auto-vector 7 | $1F | $7C |
| PARERR | 7 | From PI/T | Programmable | 4 x vector |
| ABORT | 7 | From PI/T | Programmable | 4 x vector |
| QVBERR | 7 | From PI/T | Programmable | 4 x vector |
| LTOBERR | 7 | From PI/T | Programmable | 4 x vector |
| VMEbus IRQ7* | 7 | From VMEbus | Supplied | 4 x vector |
| MC68901 MFP | 6 | From MFP | Programmable | 4 x vector |
| VMEbus IRQ6* | 6 | From VMEbus | Supplied | 4 x vector |
| Z8530 SCC | 5 | From SCC | Programmable | 4 x vector |
| VMEbus IRQ5* | 5 | From VMEbus | Supplied | 4 x vector |
| PI/T timer interrupt | 4 | Auto-vector 4 | $1C | $70 |
| VMEbus IRQ4* | 4 | From VMEbus | Supplied | 4 x vector |
| Mail-box interrupt | 3 | Auto-vector 3 | $1B | $6C |
| VMEbus IRQ3* | 3 | From VMEbus | Supplied | 4 x vector |
| VMEbus IRQ2* | 2 | From VMEbus | Supplied | 4 x vector |
| VMEbus IRQ1* | 1 | From VMEbus | Supplied | 4 x vector |

## Reset

There a total of six sources of reset on the MVME143S. They are as follows:

- VMEbus SYSRESET* — A low level on VMEbus SYSRESET* resets all onboard devices on the MVME143S.
- Power up Reset — Upon power up, the MVME143S resets all onboard devices. It also asserts VMEbus SYSRESET* low if it is system controller.
- RESET Switch — If header J9 pins 1 and 2 are jumpered, the MVME143S resets all onboard devices when the front panel RESET switch is depressed. It also asserts VMEbus SYSRESET* if this module is system controller.
- Watchdog Reset — A high level at the output of MFP timer B causes the MVME143S to reset all onboard devices. VMEbus SYSRESET* is also driven low if this module is system controller.

- Remote Reset — A low level on RRESET* causes the MVME143S to reset all onboard devices. VMEbus SYSRESET* is also driven low if this module is system controller.
- Reset Instruction — When the MC68030 MPU executes a reset instruction, the MVME143S only resets the SCC and the MFP.



### Abort

The ABORT switch is debounced and brought into the PI/T pin H3 which may be programmed to interrupt the MPU on level 7. Refer to the *Interrupt Handler* section in this chapter.

### Local Bus timeout

The local timeout generator aborts any cycle that does not complete within the local timeout period by asserting BERR* to the MPU. The timeout period is selectable to be 2 milliseconds, 64 milliseconds, or infinity through header J11. The local bus timer may be disabled by clearing the LTOEN control bit in CSR4. Refer to the *Bus Error Processing* section in Chapter 3.

## Performance

The following sections discuss performance timing of various functions.

### VMEbus Arbitration Time

When the MVME143S is configured as the system controller and is not requesting VMEbus mastership, the delay from VBBSY* high and VBRx* low to VBGxOUT* low is 110 ns typical and 150 ns maximum. When the MVME143S is not the system controller and is not requesting VMEbus mastership, the delay from VBGxIN* low to VBGxOUT* low is 20 ns typical and 150 ns maximum.

### DRAM Cycle Times

The onboard DRAM always appears as a 32-bit port to the local MPU. Writes to DRAM are always completed in 4 MPU clocks. With parity disabled, DRAM read access time is 4 MPU clocks. For cache burst fill, the first read takes 4 MPU clocks and each subsequent cycle requires 2 clocks. Therefore, a burst fill of 4 longwords would be completed in 10 clocks (2.5 clocks per longword on the average).

When parity checking is enabled, read access cycle to DRAM takes 5 MPU clocks. For cache burst fill, the first read takes 5 MPU clocks and each subsequent cycle requires 3 MPU clocks. For some applications, parity interrupt mode may be used so that DRAM access time is the same as if parity is disabled.

Multiple address RMW cycles to onboard DRAM may take more time because the DRAM sequencer requires that the MVME143S has the VMEbus mastership before the first access of the multiple address RMW cycle can begin.



## VMEbus Access Time to Onboard DRAM

The onboard DRAM access time from the VMEbus (assertion of VDS0* and/or VDS1* to assertion of VDTACK*) is typically 8 MPU clock periods for writes and 9 MPU clock periods for reads including local bus arbitration overhead. Read access time is typically 10 MPU clock periods when parity checking is enabled for the VMEbus (VPAREN* = 0). The MVME143S performs local bus arbitration for every DRAM access from the VMEbus.

## ROM/PROM/EPROM/EEPROM Cycle Times

All ROM/PROM/EPROM/EEPROM accesses require 7 MPU clock cycles to complete. See Figures 4-2 and 4-3.

## VMEbus Cycle Times

The following formula assumes that the MVME143S is the current VMEbus master and that all slaves have released VDTACK* and VBERR*. The time from the assertion of VDS0*/VDS1* to the assertion of VDTACK* is Tac in nanoseconds, T is the MPU clock period in nanoseconds, and N is the total number of MPU clock periods required to complete a VMEbus cycle. N must always be rounded up to the next integer.

N = 5 + [Tac / T] typical for read accesses
N = 6 + [Tac / T] typical for write accesses

The following formula assumes that the MVME143S is not the current VMEbus master but that it is the system controller. Also, it assumes that all previous slaves have released VDTACK* and/or VBERR* when the MVME143S receives VMEbus mastership. The delay from VBRx* low (driven by MVME143S) to VBBSY* high and VAS* high is Tr. The time from the assertion of VDS0*/VDS1* to the assertion of VDTACK* is Tac in nanoseconds, T is the MPU clock period in nanoseconds, and N is the total number of MPU clock periods required to complete a VMEbus cycle. N must always be rounded up to the next integer.

N = 8 + [(Tac + Tr) / T] typical for read accesses
N = 9 + [(Tac + Tr) / T] typical for write accesses

The following formula assumes that the MVME143S is not the current VMEbus master and it is not the system controller. Also, it assumes that all previous slaves have released VDTACK* and/or VBERR* when the MVME143S receives VMEbus mastership. The delay from VBRx* low (driven by MVME143S) to VBGxIN* low and VAS* high is Tg. The time from the assertion of VDS0*/VDS1* to the assertion of VDTACK* is Tac in nanoseconds, T is the MPU clock period in nanoseconds, and N is the total number of MPU clock periods required to complete a VMEbus cycle. N must always be rounded up to the next integer.



N = 8 + [(Tac + Tg) / T] typical for read accesses
N = 9 + [(Tac + Tg) / T] typical for write accesses

## Shared DRAM Address Map on the VMEbus

The onboard shared DRAM base address is controlled by control bits SLVA3-SLVA0. Note that SLVA3-SLVA0 also define the mailbox interrupt address.

Shared DRAM Address Map on VMEbus

| SLVA3 | SLVA2 | SLVA1 | SLVA0 | Offboard Address | Mailbox Address |
|---|---|---|---|---|---|
| | | | | | Short I/O Space |
| 0 | 0 | 0 | 0 | $00000000 | $xxxxFF00 |
| 0 | 0 | 0 | 1 | $00400000 | $xxxxFF10 |
| 0 | 0 | 1 | 0 | $00800000 | $xxxxFF20 |
| 0 | 0 | 1 | 1 | $00C00000 | $xxxxFF30 |
| 0 | 1 | 0 | 0 | $01000000 | $xxxxFF40 |
| 0 | 1 | 0 | 1 | $01400000 | $xxxxFF50 |
| 0 | 1 | 1 | 0 | $01800000 | $xxxxFF60 |
| 0 | 1 | 1 | 1 | $01C00000 | $xxxxFF70 |
| 1 | 0 | 0 | 0 | $02000000 | $xxxxFF80 |
| 1 | 0 | 0 | 1 | $02400000 | $xxxxFF90 |
| 1 | 0 | 1 | 0 | $02800000 | $xxxxFFA0 |
| 1 | 0 | 1 | 1 | $02C00000 | $xxxxFFB0 |
| 1 | 1 | 0 | 0 | $03000000 | $xxxxFFC0 |
| 1 | 1 | 0 | 1 | $03400000 | $xxxxFFD0 |
| 1 | 1 | 1 | 0 | $03800000 | $xxxxFFE0 |
| 1 | 1 | 1 | 1 | $03C00000 | $xxxxFFF0 |

After reset, SLVA3-SLVA0 = %1111. Therefore, software must initialize them if a different value is desired for proper system operation.

The onboard DRAM may be selected to respond to either 32-bit address accesses only or 24-bit and 32-bit address accesses by the VMEbus. Control bit VMEA24* defines the address size for the VMEbus slave interface. The MVME143S onboard DRAM responds to the VMEbus accesses only when the addresses match and the address modifiers (AM0-AM5) indicate privileged or non-privileged, data or program space. Also, an MVME143S may not access its own onboard memory via the VMEbus.

## Software Initialization of the MVME143S



Motorola provides three operating systems that run on the MVME143S module, VERSAdos, VMEexec, and SYSTEM V/68, as well as a debugging package with diagnostics, MVME143BUG.

Upon reset, the MVME143S module tries to fetch the initial stack pointer from the first four bytes of ROM/PROM/EPROM/EEPROM installed in bank 1. Therefore, the first two longwords of the ROM/PROM/EPROM/EEPROM in bank 1 must contain the desired values for the stack pointer and the program counter.

Use the following sequence to initialize the MVME143S from a reset:

1. Initialize all necessary exception vectors.
2. Initialize the PI/T to set up all control and status pins, port interrupts, and timer operation.
3. Initialize the MFP GPIO pins for proper input/output direction, and to required inputs.
4. Set up all timers in the MFP. Note that timer C is used for the debug port baud rate generator.
5. Set up debug serial port.
6. Set up two serial ports of the SCC.
7. Initialize the real-time clock if its oscillator has been turned off.
8. Enable master interrupt enable control bit (MIE*).

Refer to the *MC68901 MFP, MC68230 PI/T, Z8530 SCC,* and *MK48T02 Real-Time Clock Data Sheets* for instructions on programming these devices.

## System Considerations

The MVME143S needs to draw power from both P1 and P2 of the VMEbus backplane. P2 is also used for the upper 16 bits of data for 32-bit transfers, and for the upper 8 address lines for extended addressing mode. The MVME143S may not operate properly without both P1 and P2 of the VMEbus backplane.

The MVME143S contains 4Mb of DRAM whose offboard address may be programmed with control bits SLVA3-SLVA0. However, the onboard MPU always sees the local DRAM at physical address **$00000000-003FFFFF**.

D16 devices must be located in the D16 address range. Otherwise, they must only be accessed with 16-bit and 8-bit data transfers only. Refer to the *VMEbus Data Width* section in Chapter 4.

If the MVME143S tries to access offboard resources in a non-existent location, and if the system does not have a global bus timeout, the MVME143S waits forever for the VMEbus cycle to complete because the local bus timer on the MVME143S is not to terminate a VMEbus access. This would cause the system to hang up. There are two situations in which the system might lack this global bus timeout: the MVME143S is system controller but its onboard global bus timeout is disabled or the MVME143S is not the system controller, and there is no global bus timeout elsewhere in the system.

Note that the MVME143S contains no parallel ports. To use a parallel device, such as a printer, with the MVME143S, it is necessary to add a module such as the MVME050 system controller module to the system.

## Bus Error Processing



There are three sources of bus error exceptions on the MVME143S. They are: local bus timeout error (LTOBERR), VMEbus bus error (VMEBERR), and parity error (PARERR). These sources are described in the following paragraphs.

LTOBERR — Local bus timeout error occurs whenever an MPU access does not complete within the local timeout period (2 ms or 64 ms). If the system is configured properly, this should only happen if software accesses a non- existent location within the onboard range, or something prevents the board from becoming the VMEbus master. LTOBERR status is latched as the H1S bit in PI/T-PSR. The PI/T can also be programmed to interrupt the MPU on level 7 on an occurrence of LTOBERR.

VMEBERR — VMEbus bus error occurs when the BERR* signal line is asserted on the VMEbus while the MVME143S is the VMEbus master performing a VMEbus access. VMEbus bus error should only occur if:

- An initialization routine samples to see if a device is present on the VMEbus and it is not.
- Software accesses a non-existent device within the VMEbus range.
- Software tries to access a device on the VMEbus incorrectly (such as driving VLWORD* low to a 16-bit board).
- A hardware error occurs on the VMEbus.
- A VMEbus slave reports an access error (such as parity error).

VMEBERR status is latched as the H2S bit in PI/T-PSR. The PI/T can also be programmed to interrupt the MPU on level 7 on an occurrence of VMEBERR.

PARERR — Parity error occurs when the MVME143S detects a wrong parity in any or all of the four bytes of data during a read access to onboard DRAM by the local MPU. Parity checking is always performed regardless of the state of the control bit PAREN*. However, the BERR* is asserted low to the MPU only if PAREN* is asserted low. PARERR status is latched as the H4S bit in PI/T- PSR. The PI/T can also be programmed to interrupt the MPU on level 7 on an occurrence of PARERR regardless of the state of the PAREN* control bit.

Because these status flags are latched, more than one bit may be set when the bus error handler interrogates them. This condition can occur in the following situations:

- An interrupt may occur before the software has a chance to read the bus error flags. This interrupt service routine may encounter another bus error. If the bus error source is different, then the flags have two flags set when the bus error handler handles the second bus error. However, if the two bus errors had the same source then the second bus error is serviced normally, while the first bus error would lose its status flag because it would be cleared by the software that handles the latter bus error.
- A bus error can occur while the MPU is prefetching an instruction that it does not use (i.e., due to a branch instruction).

Therefore, the bus error handler must be aware of these conditions and be able to handle them appropriately.

## Use of RMW Instructions

The MC68030 RMW instructions are TAS, CAS, and CAS2. These instructions cause indivisible cycles sequences to occur on the MC68030 local bus. TAS and single address CAS perform 1 read and then 1 write to the same address. Multiple address CAS and CAS2 perform reads and writes to multiple addresses. The VMEbus defines single address indivisible cycles as Read-Modify-Write (RMW) cycles. The VMEbus does not define multiple address indivisible cycles. A scheme has been devised to allow indivisible multiple address cycles on the VMEbus. It is not part of the VMEbus specification. The scheme has the following rules:

1. Locations that are accessed by multiple address indivisible cycles are called Multiple Address Interlock (MAI) locations.
2. All devices that access MAI locations must use indivisible instructions; i.e., CAS2 of MC68030.
3. Any device that executes an indivisible cycle instruction must obtain VMEbus mastership before executing the first cycle of the instruction. In addition, it must retain VMEbus mastership until it has completed the last cycle of the instruction.

Rule 1 is a definition, rule 2 is a software requirement, and rule 3 is taken care of automatically by the MVME143S requester.

# CHAPTER 4
# FUNCTIONAL DESCRIPTION



## Introduction

This chapter provides the functional description of the MVME143S at block level. The functional description provides an overview of the module, followed by a detailed description of the module. A block diagram of the MVME143S is shown in Figure 4-4.

## Functional Description

The MVME143S is a VMEbus MPU module. The MVME143S has an MC68030 MPU, an MC68882 Floating Point Coprocessor (FPC), 4Mb of shared DRAM (accessible from the VMEbus), a battery backup real-time clock, 2Kb of battery backup SRAM, an EIA-232-D serial debug port, two multiprotocol EIA-232-D serial ports, three 8-bit timers, one 24-bit timer, four 32-pin ROM/PROM/EPROM/ EEPROM sockets, an A32/D32 VMEbus interface, a simple VMEbus interrupter, a seven-level VMEbus interrupt handler, and the VMEbus system controller functions.

### MC68030 Microprocessor (MPU)

The MPU on the MVME143S is the MC68030 enhanced 32-bit microprocessor. Some of the features offered by the MC68030 MPU are:

- Object code compatible with the MC68020 and earlier M68000 microprocessors.
- Complete 32-bit non-multiplexed address and data buses.
- Sixteen 32-bit general purpose data and address registers.
- Two 32-bit supervisor stack pointers and ten special purpose control registers.
- 256-byte instruction cache and 256-byte data cache that can be accessed simultaneously.
- Paged memory management unit that translates addresses in parallel with instruction execution and internal cache accesses.
- Two transparent segments allow untranslated access to physical memory to be defined for systems that transfer large blocks of data between predefined physical addresses; e.g., graphics applications.

- Pipelined architecture with increased parallelism allows accesses to internal caches to occur in parallel with bus transfers and instruction execution to be overlapped.
- Enhanced bus controller supports asynchronous bus cycles, synchronous bus cycles, and burst data transfers all to the physical address space.
- Dynamic bus sizing supports 8-bit, 16-bit, and 32-bit memories and peripherals.
- Support for coprocessors with M68000 coprocessor interface.
- 4Gb logical and physical addressing range.
- Implemented in HCMOS technology for maximum speed, lower power, and optimum die size.

Refer to the *MC68030 Enhanced 32-bit Microprocessor User's Manual* for details on utilizing the MPU. The MVME143S, MVME143S-2 MPU operates at 16 and 25 MHz, respectively. MPU clock frequency is controlled by oscillator Y1 which provides 2x frequency.

## MC68882 Floating Point Coprocessor (FPC)

Full IEEE floating point support is provided with the MC68882 Enhanced Floating Point Coprocessor (FPC) which operates at the same clock frequency as the onboard MPU. The major features of the MC68882 FPC are:

- Eight general purpose floating point data registers, each supporting a full 80-bit extended precision real data format.
- A 67-bit arithmetic unit to allow very fast calculations, with intermediate precision greater than the extended precision format.
- A 67-bit barrel shifter for high speed shifting operations.
- Special purpose hardware for high speed conversion of binary real memory operands to and from internal extended format.
- Reduced coprocessor interface overhead to increase throughput.
- Forty-six instructions including 35 arithmetic operations.
- Full conformance to the ANSI-IEEE 754 standard, including all requirements and suggestions.
- Support of functions not defined by the IEEE standard, including a full set of trigonometric and transcendental functions.

- Seven data types: byte, word, and longword integers; single, double, and extended precision real numbers; and packed binary coded decimal string real numbers.
- Twenty-two constants available in the onchip ROM, including pi, e, and powers of 10.
- Virtual memory/machine operations.
- Efficient mechanisms for procedure calls, context switches, and interrupt handling.
- Concurrent instruction execution with the main processor.
- Concurrent instruction execution of multiple floating point instructions.

Refer to the *MC68882 Enhanced Floating Point Coprocessor User's Manual* for details on utilizing the FPC.

## MK48T02 Battery Backup Real-Time Clock with SRAM

The Thompson Components Mostek MK48T02 is utilized by the MVME143S to provide 2040 bytes of battery backup SRAM and a battery backup real-time clock. The MK48T02 is mapped at a physical base address of $FFFC0000. Its 2Kb appears redundantly in a 128Kb block from $FFFC0000 through $FFFDFFFF. Some of the features of the MK48T02 are:

- Integrated ultra-low power SRAM, real-time clock, crystal, power-fail control circuit and battery.
- Byte-wide RAM-like clock access.
- BCD-coded year, month, date, day, hours, minutes, and seconds.
- Software-controlled clock calibration for high accuracy applications.
- Automatic power-fail protection.

The onchip battery on the MK48T02 has a predicted storage life of better than ten years at storage temperature below 70 degrees C. The backup system life is better than six years at 50 percent Vcc and better than three years at zero percent Vcc. Refer to the *MK48T02 Data Sheet* for more details on estimating the storage life and backup system life.

### Battery Backup RTC

The Real-Time Clock (RTC) is provided by the MK48T02. Accessing the RTC is as simple as conventional byte-wide SRAM access via the eight RTC registers located in the upper eight locations of the MK48T02 ($FFFC07F8 through $FFFC07FF). These RTC registers contain year, month, date, day, hours, minutes, and seconds data in 24-hour BCD format. Corrections for 28, 29 (leap year), 30, and 31 day months are made automatically. The eighth location is a control register. These RTC registers are not the actual clock counters; instead, they are bi-port read/write SRAM memory locations. The clock control circuit dumps the counters into these bi-port locations once every second. Note that no interrupts are generated by the RTC.

The MVME143S is shipped with the RTC oscillator stopped to minimize current drain from the onchip battery on the MK48T02. Before the RTC can be used, its oscillator requires a "kick start" to begin oscillation. Refer to the *MK48T02 Data Sheet* (listed in Chapter 1 herein) for programming and utilizing the RTC.

### Battery Backup SRAM

There are a total of 2040 bytes of non-volatile SRAM available in the MK48T02. The SRAM may be accessed at $FFFC0000 through $FFFC07F7. Because the RTC registers are constructed using bi-port memory cells, access to the rest of the SRAM proceeds unhindered by updates to the RTC registers, even if these RTC registers are being updated at the very moment another location in the memory array is accessed.

## Onboard Dynamic RAM

The onboard dynamic RAM (DRAM) consists of thirty-six 1 megabit x 1 DRAM ZIPs (Zigzag-Inline-Packages), making a total of 4Mb of local DRAM. It is accessible by the local MPU, the refresh circuitry, and the VMEbus.

The onboard DRAM contains byte-wide parity checking that is disabled upon reset. Parity checking is controlled with three control bits: PAREN*, VPAREN*, and WWPAR*. PAREN* is used to enable/disable parity checking for onboard MPU accesses. Control bit VPAREN* is used to enable/disable parity checking for read accesses by the VMEbus. Control bit WWPAR* may be used to force wrong parity to be written into the DRAM.

### Local Multiport Arbiter

Because the local address and data busses are used to access the onboard DRAM, any device that uses the DRAM must become the local bus master first. The MPU arbitration logics are utilized by the multiport arbiter to transfer local bus mastership from the current master to the next. The MPU is the default local bus master and has the lowest priority.



### DRAM Refresh

The DRAM requires that each of their 512 rows be refreshed once every 8 milliseconds. To accomplish this, the refresh timer requests the DRAM sequencer to perform a CAS-before-RAS refresh cycle once every 16 microseconds. The DRAM sequencer waits until the current DRAM cycle is finished before initiating the refresh cycle. If the current local bus master begins a new DRAM access during the refresh cycle, the DRAM sequencer delays the access until the refresh cycle is completed.

Note that a DRAM refresh cycle may be executed in concurrence with other MPU or VMEbus slave activities.

### Local MPU to DRAM Accesses

The local MPU is the default local bus master. Therefore, it has control of the local bus when nothing else is using the bus. The DRAM always appears as a 32-bit port to the local MPU.

Parity checking is always performed by the MVME143S and PARERR is always asserted high when a parity error occurs during a read access by the MPU. However, the MVME143S only asserts BERR* low if control bit PAREN* is low. Therefore, deferred parity checking may be used for MPU read accesses by disabling parity checking with PAREN* while enabling PARERR interrupt via the MC68230 PI/T.

Refer to the *MC68230 Parallel Interface/Timer (PI/T)* section for further details.

NOTE

To support the MC68030 cache, the MVME143S always performs longword access every time the MPU executes a read cycle to the onboard DRAM regardless of the number of bytes requested by the MPU.



### VMEbus to Onboard DRAM Accesses

The onboard DRAM is accessible by other VMEbus masters in the system when control bit SLAVEN* is low. When the VMEbus map decoder detects an onboard DRAM select, it requests local bus mastership from the multiport arbiter. The multiport arbiter then requests the MPU for the local bus. When the local bus is released, the multiport arbiter grants local bus mastership to the VMEbus slave interface. At this time, a DRAM read or write cycle is performed. If the VMEbus master is executing an RMW cycle to the DRAM, then the multiport arbiter does not restore local bus mastership to the MPU until both the read and write cycles are completed.

If the MPU is the current local bus master and is executing a cycle that requires the VMEbus when the VMEbus slave map decoder requests local bus mastership, then a VMEbus deadlock condition occurs. To break this VMEbus deadlock condition, the multiport arbiter signals a retry to the MPU by asserting both BERR* and HALT*. The MPU responds by aborting the current cycle, at which time it relinquishes local bus mastership so that the multiport arbiter can grant it to the VMEbus. When the VMEbus has finished with the DRAM, the multiport arbiter returns local bus mastership to the MPU. The MPU then retries the aborted cycle.

The onboard DRAM appears to the VMEbus as a 16-bit port for transfers with VLWORD* negated, and as a 32-bit port for transfers with VLWORD* asserted. The MVME143S supports misaligned transfers to and from the local DRAM by the VMEbus.

## VMEbus Master Interface

The MVME143S has an A32/D32 VMEbus master interface for buffering of data, address, and control; for word data manipulation to accommodate MC68030 and MC68882 and VMEbus data handling differences; and for interrupt handling and control of misaligned transfers.

### VMEbus Address Size

The MVME143S allows you to select a 32-bit or 24-bit address option for VMEbus references. This option is controlled by control bit VMEA24*. The MVME143S VMEbus memory map is directly affected by the address option. The VMEbus memory map when VMEA24* = 0 is shown in Table 4-1. The VMEbus memory map when VMEA24* = 1 is shown in Table 4-2.



**Table 4-1.** VMEbus Memory Map When VMEA24* = 0

| Address Range | VMEbus Activity Type |
|---|---|
| $00000000-$003FFFFF | Onboard DRAM |
| $00400000-$00FFFFFF | VMEbus standard (24-bit) address space |
| $01000000-$EFFFFFFF | VMEbus extended (32-bit) address space |
| $F0000000-$F0FFFFFF | VMEbus standard (24-bit) address space |
| $F1000000-$FFEFFFFF | VMEbus extended (32-bit) address space |
| $FFF00000-$FFFEFFFF | Onboard resources |
| $FFFF0000-$FFFFFFFF | VMEbus short I/O (16-bit) address space |

**Table 4-2.** VMEbus Memory Map When VMEA24* = 1

| Address Range | VMEbus Activity Type |
|---|---|
| $00000000-$003FFFFF | Onboard DRAM |
| $00400000-$FFEFFFFF | VMEbus extended (32-bit) address space |
| $FFF00000-$FFFEFFFF | Onboard resources |
| $FFFF0000-$FFFFFFFF | VMEbus short I/O (16-bit) address space |

### VMEbus Data Width

As a VMEbus master, the MVME143S performs 32-bit data transfers only on longword-aligned accesses and only if the accessed address falls in the D32 range of the VMEbus. Control bit VMED16* is used to map the D16 and D32 areas for the VMEbus as seen by the MVME143S. When VMED16* = 0, the entire VMEbus range is always considered to be a 16-bit data port by the MVME143S. When VMED16* = 1, then the MVME143S partitions the VMEbus as follows: address ranges $00400000 through $EFFFFFFF are 32-bit data ports and address ranges $F0000000 through $FFEFFFFF and $FFFF0000 through $FFFFFFFF are 16-bit data ports.

### Accessing the VMEbus

Whenever the MVME143S executes a VMEbus cycle (read, write, or interrupt acknowledge) and its VMEbus requester has obtained VMEbus mastership, it drives the VME address bus with its local address bus and the VMEbus address modifiers to indicate proper address space. It also asserts VIACK* if this is an interrupt acknowledge cycle. It asserts VLWORD* on longword-aligned transfers only if the accessed address falls in the 32-bit data range.

When VA01-VA31, VAM0-VAM5, VIACK*, and VLWORD* are driven to their appropriate levels on the VMEbus, the MVME143S asserts VAS*. VWRITE* line is driven low for write accesses and high for read accesses. Then after the data bus is driven according to the access cycle, it asserts VDS0* and/or VDS1* appropriately. If the cycle terminates normally with VDTACK*, then the onboard DSACK generator circuit activates both DSACK1* and DSACK0* if VLWORD* is low or only DSACK1* if VLWORD* is high. If the cycle terminates with VBERR*, then the BERR generator circuit asserts BERR* to the MPU. When the handshake has occurred, the MPU removes AS*, DS*, and the MVME143S completes the cycle by disabling the data bus drivers and removing VDS0*/VDS1* and VAS*.

The above sequence is altered slightly when the local processor executes RMW cycles. When the local processor starts an RMW cycle, the VMEbus master interface checks to see if it is a single or multiple address RMW by examining SIZ1 and SIZ0. If it is a multiple address RMW cycle, then the VMEbus master interface operates normally. If it is a single address RMW cycle, then the VMEbus master interface keeps VAS* active during the entire time from the beginning of the RMW read cycle to the end of the RMW write cycle. This makes a single address RMW cycle from the MPU appear on the VMEbus as a VMEbus-defined read-modify-write cycle.

## VMEbus Requester

The VMEbus requester is used to obtain and relinquish mastership of the VMEbus. It can request VMEbus mastership on any one of the four request levels depending on the configurations of J4 and J6, and it fully supports the bus grant daisy-chain. It requests mastership of the VMEbus any time the MVME143S is not the current VMEbus master and the map decoder or the interrupt handler indicates that the MPU is executing a cycle that requires the VMEbus. It also requests mastership of the VMEbus when the MVME143S is not the current VMEbus master and the local processor is starting to execute a multiple-address RMW sequence to the onboard DRAM.

The VMEbus requester operates in the Release-On-Request (ROR) mode. When the MVME143S has obtained VMEbus mastership, the VMEbus requester

maintains mastership until another VMEbus module requests VMEbus mastership and then only if an RMW sequence is not in process. It releases the VMEbus in one of two different ways, depending on the state of the MVME143S at the time.

1. If the MVME143S is in the middle of a VMEbus cycle (VAS* already asserted) when the VMEbus requester decides to relinquish VMEbus mastership, it releases VBBSY* immediately. The transfer of mastership occurs when the VMEbus master interface negates and releases VAS*.
2. If the MVME143S is not in the middle of a VMEbus cycle when the VMEbus requester decides to relinquish VMEbus mastership, the VMEbus master interface releases all of the VMEbus lines, after which the VMEbus requester releases VBBSY* to complete the transfer of VMEbus mastership.

The VMEbus requester operates in fairness mode (Request On No Request) when the jumper on header J3 is installed.



## Data Bus Structure

The data bus structure on the MVME143S is arranged to accommodate the 8-bit, 16-bit, 32-bit, and 16/32-bit ports that reside on the module. The data bus structure of the MVME143S is shown in Figure 4-1.

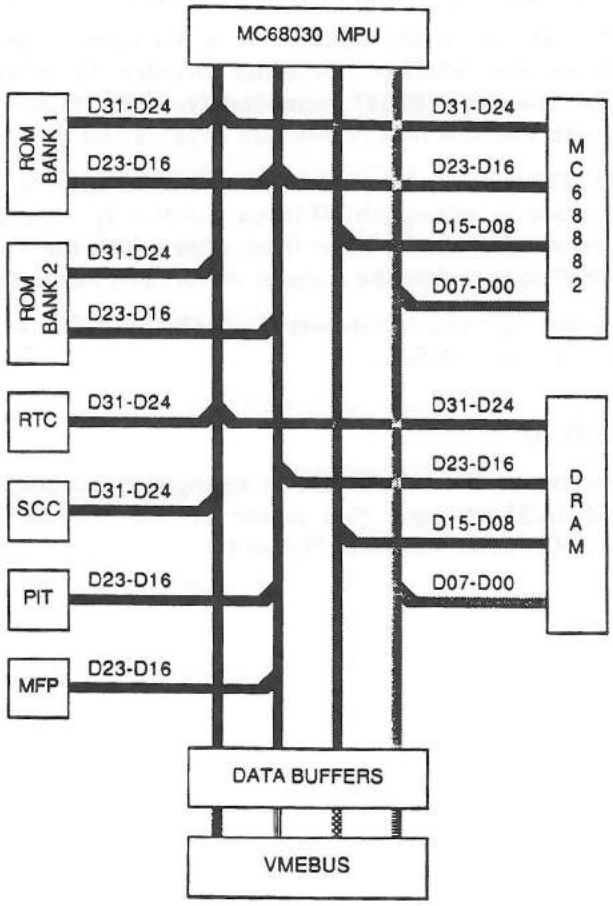



**Figure 4-1.** MVME143S Data Bus Structure

## VMEbus System Controller

All of the MVME143S system controller functions and the SYSRESET* driver are enabled/disabled by header J1. The position of the jumper on J1 appears as status bit SYSCON at the MFP. Also, the SCON LED is lit if the MVME143S is configured as the system controller. The system controller and SYSRESET* driver functions are described in the following sections.



### Global Bus timeout

The global bus timeout circuit starts the timing upon detecting assertion of VDS0* and/or VDS1*. If VDS0* and/or VDS1* are asserted longer than the timeout period, it drives VBERR* low. The global bus timer may be disabled with control bit GTOEN. Note that GTOEN also enables/disables PRI arbiter timeout. Header J5 allows timeout period selections of 64 μs, 128 μs, or infinity.

### SYSCLK Driver

The System Clock (SYSCLK) driver drives a periodic 16 MHz clock onto the SYSCLK line on the VMEbus if the system controller on the MVME143S is enabled.

### VMEbus Arbiter

The bus arbiter can operate in Priority (PRI) mode with the jumper removed from header J2 or in Round-Robin-Select (RRS) mode with the jumper installed on header J2. It is designed to rearbitrate if no VMEbus master responds to a grant within a timeout period. The arbitration timer may be disabled with control bit GTOEN. Note that GTOEN also enables/disables global bus timeout. Header J5 allows timeout period selections of 64 μs, 128 μs, or infinity.

### IACK* Daisy-Chain Driver

The Interrupt Acknowledge (IACK*) daisy-chain driver is designed to meet the VMEbus specification requirements. Note that for the IACK* daisy-chain driver to function properly, the MVME143S must be in the left most slot in the chassis if it is the system controller.

### SYSRESET* Driver

Although System Reset (SYSRESET*) is not a VMEbus system controller function, the MVME143S enables/disables its SYSRESET* function at the same time that it enables/disables its system controller functions. When configured as the system controller, the MVME143S drives the SYSRESET* signal line low when the front panel RESET switch is depressed, when a watchdog timeout occurs, when the RRESET* line is asserted, or when a power up occurs.



**NOTE**

The MVME143S does not fully implement SYSRESET* timing of a VMEbus power monitor.

## Mailbox Interrupt

The MVME143S provides a means for other VMEbus masters in the system to interrupt its MPU via short I/O space. The mailbox interrupt is enabled when control bit MBXIEN* is low. When the MVME143S detects an access to a selected location in the short I/O space, it generates an auto-vectored interrupt to the onboard MPU on level 3. To generate VDTACK* for the VMEbus, the MVME143S actually performs a read or write access to the onboard DRAM. For this reason, only read accesses are recommended because write accesses can alter the onboard DRAM data. The mailbox interrupt address is selected with control bits SLVA3-SLVA0 which also control the offboard address of the onboard DRAM. Refer to the *Shared DRAM Address Map on VMEbus* section in Chapter 3 and the *Interrupt Handler* section in this chapter for more details.

## VMEbus Interrupter

The MVME143S contains a VMEbus interrupter with programmable level and programmable vector. The level is programmed with control bits VI3-VI1 and the vector is programmed with control bits VID3-VID0. The state of the interrupter is reflected as OIRQ status bit. During the interrupt acknowledge cycle, the MVME143S places the state of VID3-VID0 onto VD3-VD0 of the VMEbus. The rest of the data bus is not driven. Therefore, the range of the ID byte (vector) for the bus interrupter on the MVME143S is $F0 through $FF. It is an 8-bit interrupter and consequently responds to all sizes of interrupt acknowledge cycles. The VMEbus interrupter drives the selected interrupt request line low whenever the MPU performs an access to a location within $FFFE0000 through $FFFEFFFF. A typical sequence for interrupting is as follows:

1. Verify that status bit OIRQ is 0.
2. Perform a read access to physical address $FFFE0000.
3. Poll status bit OIRQ until it changes to 0.
4. The VMEbus interrupt has now been acknowledged.
5. Continue with normal processing.



## MC68901 Multi-Functional Peripheral (MFP)

The MVME143S uses the Multi-Functional Peripheral (MFP) for its front panel debug port, tick timers, watchdog timer, and the status and control information. The MFP has the ability to interrupt the MPU on level 6. Its interrupt sources are from the timers, debug port, and from the GPIO (status) bits.

The MFP appears as a 16-bit port to the MPU and is located at a physical base address of $FFF80000. The MFP may be accessed with 8-bit accesses at the odd byte locations or with 16-bit accesses at the even byte locations. When accessed with 16-bit transfers, the MFP appears at the least significant byte of the 16-bit word. The MFP register map is shown in Table 4-3.

**Table 4-3.** MFP Register Map

| Offset | Physical Address | | Register Name | Register Description |
|---|---|---|---|---|
| | 16-Bit Access | 8-Bit Access | | |
| $1 | $FFF80000 | $FFF80001 | GPIP | General purpose I/O |
| $3 | $FFF80002 | $FFF80003 | AER | Active edge register |
| $5 | $FFF80004 | $FFF80005 | DDR | Data direction register |
| $7 | $FFF80006 | $FFF80007 | IERA | Interrupt enable reg A |
| $9 | $FFF80008 | $FFF80009 | IERB | Interrupt enable reg B |
| $B | $FFF8000A | $FFF8000B | IPRA | Interrupt pending reg A |
| $D | $FFF8000C | $FFF8000D | IPRB | Interrupt pending reg B |
| $F | $FFF8000E | $FFF8000F | ISRA | Interrupt in-service reg A |
| $11 | $FFF80010 | $FFF80011 | ISRB | Interrupt in-service reg B |
| $13 | $FFF80012 | $FFF80013 | IMRA | Interrupt mask reg A |
| $15 | $FFF80014 | $FFF80015 | IMRB | Interrupt mask reg B |
| $17 | $FFF80016 | $FFF80017 | VR | Vector register |
| $19 | $FFF80018 | $FFF80019 | TACR | Timer A control register |
| $1B | $FFF8001A | $FFF8001B | TBCR | Timer B control register |
| $1D | $FFF8001C | $FFF8001D | TCDCR | Timer C and D contl reg |
| $1F | $FFF8001E | $FFF8001F | TADR | Timer A data register |
| $21 | $FFF80020 | $FFF80021 | TBDR | Timer B data register |
| $23 | $FFF80022 | $FFF80023 | TCDR | Timer C data register |
| $25 | $FFF80024 | $FFF80025 | TDDR | Timer D data register |
| $27 | $FFF80026 | $FFF80027 | SCR | Sync character register |
| $29 | $FFF80028 | $FFF80029 | UCR | USART control register |
| $2B | $FFF8002A | $FFF8002B | RSR | Receive status register |
| $2D | $FFF8002C | $FFF8002D | TSR | Transmit status register |
| $2F | $FFF8002E | $FFF8002F | UDR | USART data register |

Refer to the *MC68901 Multi-Function Peripheral Data Sheet* for more details in programming and utilizing the MFP.

### Serial Port 1: Front Panel Debug Port

The front panel debug port is a minimal implementation of an EIA-232-D serial port. It may be configured to be either DTE or DCE by properly jumpering header J17. It supports TXD1, RXD1, RTS1, and CTS1. When configured as DCE, it drives data with RXD1, receives data with TXD1, controls CTS1 with GPIO3, and monitors RTS1 with GPIO0. Also, DSR1 and DCD1 are pulled up to +12 Vdc in the DCE configuration. When configured as DTE, it drives data with TXD1, receives data with RXD1, controls RTS1 with GPIO3, and monitors CTS1 with GPIO0. DTR1 is also pulled up to +12 Vdc in the DTE configuration. Port configurations for port 1 are shown in the figures under the *Serial Port 1 Configuration Select Header* section in Chapter 2. The baud rate generator for the serial port is timer C of the MFP. The XTAL input to the MFP can be either 1.230769 MHz (J10 pins 1-2) or 2.461538 MHz (J10 pins 2-3). The baud rates supported are programmed as shown in the following table.

Port 1 Baud Rates Available

| J10 (1-2) Baud Rate | J10 (2-3) Baud Rate | Clock Mode | Prescale Value | Timer C Count | Percent Error |
|---|---|---|---|---|---|
| 9600 | 19200 | X16 | 4 | 1 | .16 |
| 4800 | 9600 | X16 | 4 | 2 | .16 |
| 2400 | 4800 | X16 | 4 | 4 | .16 |
| 1200 | 2400 | X16 | 4 | 8 | .16 |
| 600 | 1200 | X16 | 4 | $10 | .16 |
| 300 | 600 | X16 | 4 | $20 | .16 |
| 110 | 220 | X16 | 4 | $57 | .47 |

### MFP Timers

The MFP provides the MVME143S with four 8-bit timers. They are assigned as follows:

- Timer C - Baud rate generator for the debug serial port.
- Timer A - Software tick timer.
- Timer B - Tick timer overflow/watchdog timeout.
- Timer D - Delay mode only. Unassigned by hardware.

**NOTE**

The watchdog timeout resets the MPU module when timer B output is high.



### Control/Status Register 1: GPIO Pins

The MC68901 MFP has eight General Purpose I/O (GPIO) pins. The MVME143S uses five of these pins as status inputs and three of them as control outputs to form Control/Status Register 1 (CSR1). After a reset, the MFP makes all of the GPIO pins inputs. Therefore, after each reset, the software should initialize the control bits and make them outputs. MVME143S hardware defaults the control lines to high when they are not programmed as outputs. GPIO0- GPIO7 pins are assigned as follows:

| Address | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
|---|---|---|---|---|---|---|---|---|
| FFF80001 | SYSFAIL | ACFAIL | BRDFAIL | MIE* | DCTS* | PWRUP | SYSCON | DRTS* |
| | R | R | R/W | R/W | R/W | R | R | R |

DRTS* This bit reflects the state of RTS1 when port 1 is configured as DCE. When port 1 configured as DTE, DRTS* monitors the state of CTS1 instead. DRTS* is 0 when the corresponding EIA-232-D signal line is high and is 1 when it is low. Bit 0 of the IPRB may be initialized by software to detect the transitions of DRTS*.

SYSCON A logical 1 at this status bit indicates that the MVME143S is the system controller.

PWRUP A power up reset sets this status bit to a logical 1. PWRUP bit is cleared when software clears control bit BRDFAIL.

DCTS* This bit controls the state of CTS1 when port 1 is configured as DCE. When port 1 configured as DTE, DCTS* controls the state of RTS1 instead. The corresponding EIA-232-D signal line is high when DCTS* is 0 and is low when it is 1.

MIE* When MIE* = 1, no interrupt requests reach the MPU. When MIE* = 0, interrupts are enabled to reach the MPU. Note that after a reset this pin becomes input and defaults to a logical 1 by the MVME143S hardware.

BRDFAIL A high level on BRDFAIL lights the front panel FAIL indicator. VMEbus SYSFAIL* signal is also driven low if the MVME143S is not the system controller. Note that after a reset this pin becomes input and defaults to a logical 1 by the MVME143S hardware.

ACFAIL This bit reflects the state of the ACFAIL* signal line on the VMEbus. A logical 1 indicates that ACFAIL* is asserted and a logical 0 indicates that ACFAIL* is negated. An active level on ACFAIL also causes a level 7 interrupt to the onboard MPU if MIE* = 0. Transitions on ACFAIL* signal may be detected and latched in IPRA bit 6.

SYSFAIL This bit reflects the state of the SYSFAIL* signal line on the VMEbus. A logical 1 indicates that SYSFAIL* is asserted and logical 0 indicates that SYSFAIL* is negated. Transitions on SYSFAIL* may be detected and latched in IPRA bit 7.



## MC68230 Parallel Interface/Timer (PI/T)

The MVME143S uses the MC68230 Parallel Interface/Timer (PI/T) to provide 20 control bits, 5 status bits, one 24-bit timer, and interrupt handling capabilities for the ABORT switch and bus errors. Software abort (ABORT switch), parity error, VMEbus access error, and local timeout bus error may be individually enabled under software control to interrupt on level 7 with programmable vector. The PI/T also provides the MVME143S with a 24-bit timer which can generate an auto-vectored interrupt to the local MPU on level 4.

The PI/T appears as a 16-bit port to the MPU and is located at a physical base address of $FFF90000. The PI/T may be accessed with 8-bit accesses at the odd byte locations or with 16-bit accesses at the even byte locations. When accessed with 16-bit transfers, the PI/T appears at the least significant byte of the 16-bit word. The register map of the PI/T is shown in Table 4-4.

**Table 4-4.** PI/T Register Map

| Offset | Physical Address | | Register | Register |
|---|---|---|---|---|
| | 16-Bit Access | 8-Bit Access | Name | Description |
| $1 | $FFF90000 | $FFF90001 | PGCR | Port general control reg |
| $3 | $FFF90002 | $FFF90003 | PSRR | Port service request reg |
| $5 | $FFF90004 | $FFF90005 | PADDR | Port A data direction reg |
| $7 | $FFF90006 | $FFF90007 | PBDDR | Port B data direction reg |
| $9 | $FFF90008 | $FFF90009 | PCDDR | Port C data direction reg |
| $B | $FFF9000A | $FFF9000B | PIVR | Port interrupt vector reg |
| $D | $FFF9000C | $FFF9000D | PACR | Port A control register |
| $F | $FFF9000E | $FFF9000F | PBCR | Port B control register |
| $11 | $FFF90010 | $FFF90011 | PADR | Port A data register |
| $13 | $FFF90012 | $FFF90013 | PBDR | Port B data register |
| $15 | $FFF90014 | $FFF90015 | PAAR | Port A alternate register |
| $17 | $FFF90016 | $FFF90017 | PBAR | Port B alternate register |
| $19 | $FFF90018 | $FFF90019 | PCDR | Port C data register |
| $1B | $FFF9001A | $FFF9001B | PSR | Port status register |
| $21 | $FFF90020 | $FFF90021 | TCR | Timer control register |
| $23 | $FFF90022 | $FFF90023 | TIVR | Timer interrupt vector reg |
| $27 | $FFF90026 | $FFF90027 | CPRH | Counter preload reg high |
| $29 | $FFF90028 | $FFF90029 | CPRM | Counter preload reg mid |
| $2B | $FFF9002A | $FFF9002B | CPRL | Counter preload reg low |
| $2F | $FFF9002E | $FFF9002F | CNTRH | Counter register high |
| $31 | $FFF90030 | $FFF90031 | CNTRM | Counter register middle |
| $33 | $FFF90032 | $FFF90033 | CNTRL | Counter register low |
| $35 | $FFF90034 | $FFF90035 | TSR | Transmit status register |

### Control/Status Register 2: Port A Pins

The MVME143S utilizes the PI/T port A pins to form Control/Status Register 2 (CSR2). All eight port A pins are used as control outputs. After a reset, the PI/T makes all parallel pins inputs. Therefore, software must initialize the PI/T to make all port A pins outputs after each reset. All port A pins default to high when they are not programmed as outputs. The assignment of PA7-PA0 is as follows:



| Address | PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
|---|---|---|---|---|---|---|---|---|
| FFF90011 | SLVA3 | SLVA2 | SLVA1 | SLVA0 | SLAVEN* | VMEA24* | MBXIEN* | VMED16* |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

SLVA3-SLVA0 — SLVA3-SLVA0 define the offboard address of the onboard shared DRAM for accesses from the VMEbus. They also define the mailbox interrupt address in the short I/O space for the MVME143S. Refer to shared DRAM address map on VMEbus for more details.

SLAVEN* — This control bit is used to enable/disable the VMEbus slave interface to the onboard shared DRAM. When SLAVEN is high, the slave interface is disabled.

VMEA24* — This control bit defines the address size of the VMEbus. When this bit is low, the VMEbus is treated as a 24-bit address port (A24) for address range $00400000 through $00FFFFFF and $F0000000 through $F0FFFFFF and as a 32-bit address port (A32) for address range $01000000 through $EFFFFFFF and $F1000000 through $FFFEFFFF. Also, the MVME143S slave interface responds to both A24 and A32 accesses to the onboard shared DRAM. When this control bit is high, the VMEbus is always treated as an A32 port except for the short I/O space. A logical high on VMEA24* also restricts the MVME143S shared DRAM to respond to A32 accesses only.

MBXIEN* — This control bit is used to enable/disable the mailbox interrupt on the MVME143S. When this pin is low, a level 3 interrupt is issued to the onboard MPU every time a selected address range in short I/O space is accessed (provided the master interrupt bit MIE* is enabled low).

VMED16* The state of this pin defines the VMEbus data size. When VMED16* is low, the MVME143S always treats VMEbus as a 16-bit data port for all data transfers. When it is high, VMEbus is treated as a 32-bit data port for address range $40000000 through $EFFFFFFF and as a 16-bit data port for $F0000000 through $FFEFFFFF and $FFFF0000 through $FFFFFFFF.

4

### Control/Status Register 3: Port B Pins

The MVME143S utilizes the PI/T port B pins to form Control/Status Register 3 (CSR3). Seven port B pins are used as control outputs and one as a status input. After a reset, the PI/T makes all parallel pins inputs. Therefore, software must initialize the control bits and make them outputs after each reset. All port B pins default to high when they are not programmed as outputs. CSR3 should only be changed when OIRQ status bit is low. The assignment of PB7-PB0 is as follows:

| Address | PB7 | PB6 | PB5 | PB4 | PB3 | PB2 | PB1 | PB0 |
|---|---|---|---|---|---|---|---|---|
| FFF90013 | OIRQ | VI3 | VI2 | VI1 | VID3 | VID2 | VID1 | VID0 |
| | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

OIRQ This status pin monitors the state of the MVME143S VMEbus interrupter. When it is high, a VMEbus interrupt request is driven at the level selected by VI3-VI1.

VI3-VI1 VI3-VI1 select the interrupt level for the MVME143S VMEbus interrupter. They are assigned as follows:

| VI3 | VI2 | VI1 | Interrupt Level |
|---|---|---|---|
| 0 | 0 | 0 | No interrupt |
| 0 | 0 | 1 | Level 1 |
| 0 | 1 | 0 | Level 2 |
| 0 | 1 | 1 | Level 3 |
| 1 | 0 | 0 | Level 4 |
| 1 | 0 | 1 | Level 5 |
| 1 | 1 | 0 | Level 6 |
| 1 | 1 | 1 | Level 7 |

| Configuration 5 | Configuration 6 | Configuration 7 | Configuration 8 | | | | | Configuration 8 | Configuration 7 | Configuration 6 | Configuration 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| +5V | +5V | +5V | A20 | 1 | | | 32 | +5V | +5V | +5V | +5V |
| A17 | A17 | A17 | A17 | 2 | | | 31 | A19 | A19 | +5V | +5V |
| A16 | A16 | A16 | A16 | 3 | 1 | 28 | 30 | A18 | A18 | A18 | A18 |
| A13 | A13 | A13 | A13 | 4 | 2 | 27 | 29 | A15 | A15 | A15 | A15 |
| A8 | A8 | A8 | A8 | 5 | 3 | 26 | 28 | A14 | A14 | A14 | A14 |
| A7 | A7 | A7 | A7 | 6 | 4 | 25 | 27 | A9 | A9 | A9 | A9 |
| A6 | A6 | A6 | A6 | 7 | 5 | 24 | 26 | A10 | A10 | A10 | A10 |
| A5 | A5 | A5 | A5 | 8 | 6 | 23 | 25 | A12 | A12 | A12 | A12 |
| A4 | A4 | A4 | A4 | 9 | 7 | 22 | 24 | OE* | OE* | OE* | OE* |
| A3 | A3 | A3 | A3 | 10 | 8 | 21 | 23 | A11 | A11 | A11 | A11 |
| A2 | A2 | A2 | A2 | 11 | 9 | 20 | 22 | CE* | CE* | CE* | CE* |
| A1 | A1 | A1 | A1 | 12 | 10 | 19 | 21 | D7 | D7 | D7 | D7 |
| D0 | D0 | D0 | D0 | 13 | 11 | 18 | 20 | D6 | D6 | D6 | D6 |
| D1 | D1 | D1 | D1 | 14 | 12 | 17 | 19 | D5 | D5 | D5 | D5 |
| D2 | D2 | D2 | D2 | 15 | 13 | 16 | 18 | D4 | D4 | D4 | D4 |
| GND | GND | GND | GND | 16 | 14 | 15 | 17 | D3 | D3 | D3 | D3 |

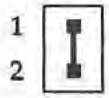
| Configuration 9 | | | | | Configuration 9 |
|---|---|---|---|---|---|
| | 1 | | | 32 | |
| | 2 | | | 31 | |
| A15 | 3 | 1 | 28 | 30 | +5V |
| A13 | 4 | 2 | 27 | 29 | WE* |
| A8 | 5 | 3 | 26 | 28 | A14 |
| A7 | 6 | 4 | 25 | 27 | A9 |
| A6 | 7 | 5 | 24 | 26 | A10 |
| A5 | 8 | 6 | 23 | 25 | A12 |
| A4 | 9 | 7 | 22 | 24 | OE* |
| A3 | 10 | 8 | 21 | 23 | A11 |
| A2 | 11 | 9 | 20 | 22 | CE* |
| A1 | 12 | 10 | 19 | 21 | D7 |
| D0 | 13 | 11 | 18 | 20 | D6 |
| D1 | 14 | 12 | 17 | 19 | D5 |
| D2 | 15 | 13 | 16 | 18 | D4 |
| GND | 16 | 14 | 15 | 17 | D3 |

2

## RESET Switch Enable/Disable Select Header (J9)

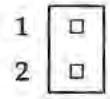
Header J9 allows you to enable or disable the front panel RESET switch. If the MVME143S is system controller, header J9 also asserts VMEbus SYSRESET*. The module is shipped with factory configuration of RESET switch enabled.

J9




Reset Switch Enabled

J9



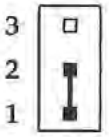

Reset Switch Disabled

## Serial Clock Select Header (J10)

Header J10 provides a choice of XTAL input to the MC68901 Multifunction Peripheral (MFP). Timer C of the MFP is the baud rate generator for the front panel debug port (serial port 1). The XTAL input to the MFP may be either 1.230769 MHz or 2.461538 MHz. The table below shows the baud rates for the two inputs. The as-shipped factory is with XTAL input of 1.230769.

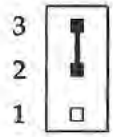
| J10 (1-2) Baud Rate | J10 (2-3)Clock Baud Rate | Prescale Mode | Timer C Value | Percent Count | Error |
|---|---|---|---|---|---|
| 9600 | 19200 | X16 | 4 | 1 | .16 |
| 4800 | 9600 | X16 | 4 | 2 | .16 |
| 2400 | 4800 | X16 | 4 | 4 | .16 |
| 1200 | 2400 | X16 | 4 | 8 | .16 |
| 600 | 1200 | X16 | 4 | $10 | .16 |
| 300 | 600 | X16 | 4 | $20 | .16 |
| 110 | 220 | X16 | 4 | $57 | .47 |

J10




Serial clock is 1.230769 MHz

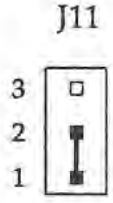
J10



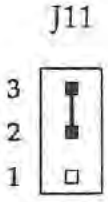

Serial clock is 2.461538 MHz

## Local Timeout Select Header (J11)

Header J11 provides a choice of the local timeout period to be 2 milliseconds or 64 milliseconds. When the PI/T Local Timeout Enable (LTOEN) bit is high, the local bus timer is enabled and the selected timeout period is accepted. The as-shipped factory configuration is 2 milliseconds.



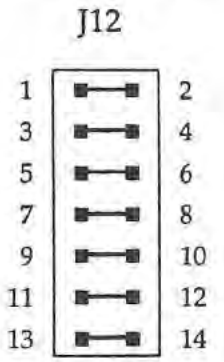

Timeout Period is 2 ms



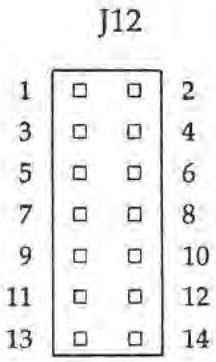

Timeout Period is 64 ms

## VMEbus Interrupt Handler Select Header (J12)

Header J12 is used to enable or disable each individual interrupt level for the interrupt handler. With a jumper installed that interrupt level is enabled. Any or all interrupt levels may be enabled. The factory configuration is that all seven levels are enabled for the handler.




All Seven Interrupts Enabled




All Seven Interrupts Disabled

2

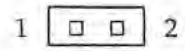
## EPROM Address Map Select Header (J13)

Header J13 allows the EPROM base address to be mapped according to the device size. The four 32-pin sockets are organized in two banks with two sockets per bank. Bank 1 (U3 = even, U5 = odd); bank 2 (U1 = even, U2 = odd). Each bank appears as a 16-bit word.

Bank 1 address base is at $FF800000 and bank 2 address base is at $FFA00000. If RAM devices are 64K bit or smaller, the $FFF00000 to $FFF3FFFF range may be used. This allows backward compatibility with the MVME143. Base addresses for bank 1 and bank 2 of 64k bit or smaller devices is $FFF00000 and $FFF20000, respectively.

When 64K bit or smaller devices are used, the address range of $FF800000 to $FFBFFFFF can be mapped as VMEbus A32/D16 (jumper installed). The as-shipped factory configuration is with mapping at $FFF00000 to $FFF30000.

J13




EPROMs are mapped at FFF00000 to FFF3FFFF and also at FF800000 to FFBFFFFF

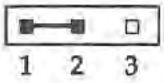
J13


EPROMs are mapped at FFF00000 to FFF3FFFF. FF800000 to FFBFFFFF is mapped as VMEbus A32/D16

## SCC RTXCA Source Select Header (J15)

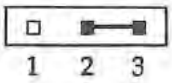
Header J15 is used to connect the SCC RTXCA input to the onboard 1.230769 MHz clock. The module is shipped with factory configuration of RTXCA input connected to the 1.230769 MHz clock.

J15




RTXCA input from onboard 1.23 MHz clock

J15

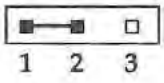



RTXCA input from TXCO3 if port 3 is DCE and from TXCI3 if port 3 is DTE

## SCC RTXCB Source Select Header (J16)

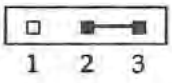
Header J16 is used to connect the SCC RTXCB input to the onboard 1.23 MHz or the 2.46 MHz clock. The module is shipped with factory configuration of RTXCB input connected for input from the clock.

J16




RTXCB input from onboard 1.23 or 2.46 MHz clock

J16

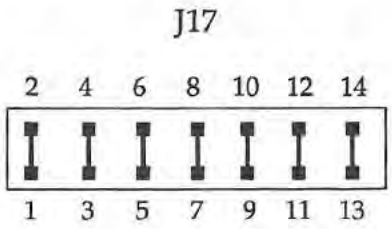


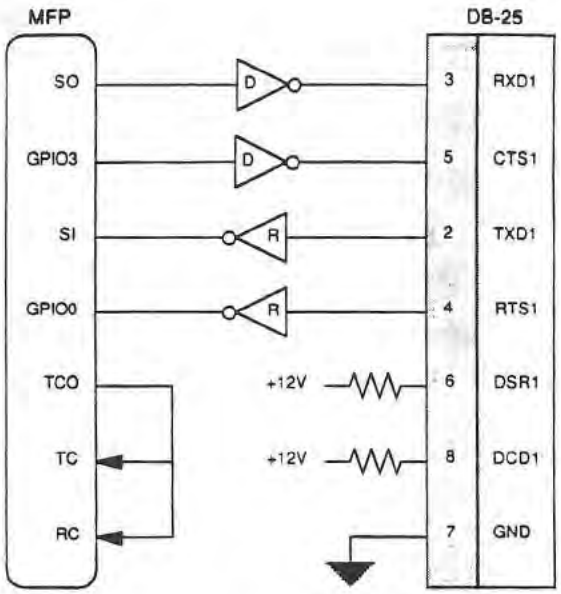

RTXCB input from TXCO2 if port 2 is DCE and from TXCI2 if port 2 is DTE

## Serial Port 1 Configuration Select Header (J17)

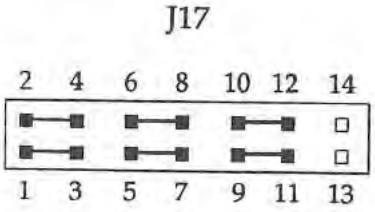
Header J17 configures serial port 1 on the front panel and on connector P2 as a modem (DCE) for connection to a terminal or as a terminal (DTE) for connection to a modem. This debug port is a minimal implementation of an EIA-232-D serial port. The module is shipped factory configured as DCE (to terminal). Configuration is accomplished by installing jumpers as shown below. A schematic representation of each configuration is also shown. Refer to the *Connector J18 Interconnect Signals* section in this chapter for pinouts.

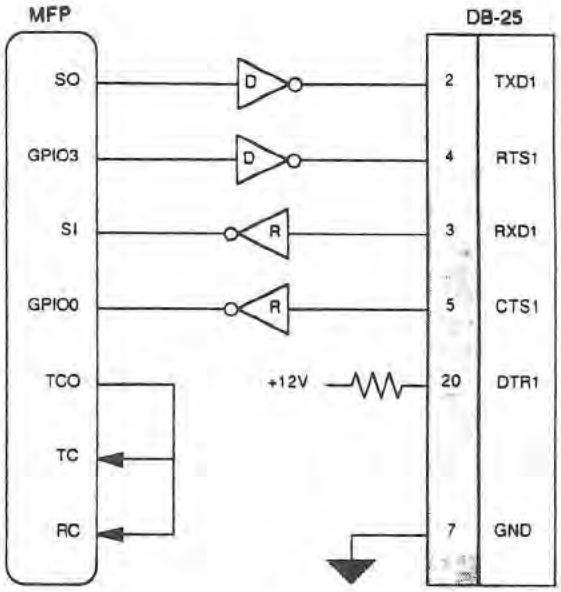
The front panel connector supports TXD, RXD, RTS, and CTS. DSR and DCD are pulled up to +12 Vdc by the module.




Port 1 Configured as DCE



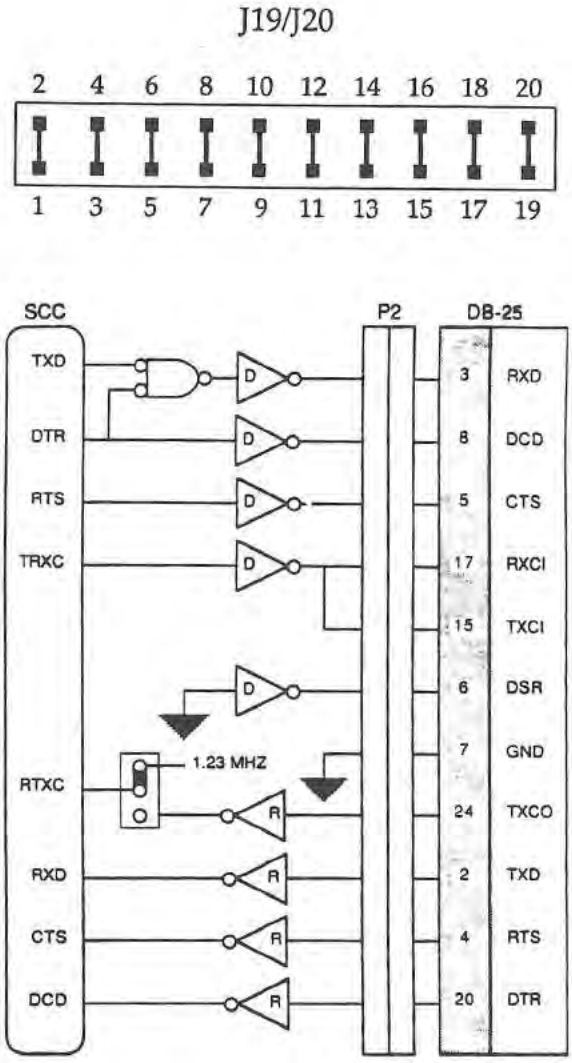




Port 1 Configured as DTE

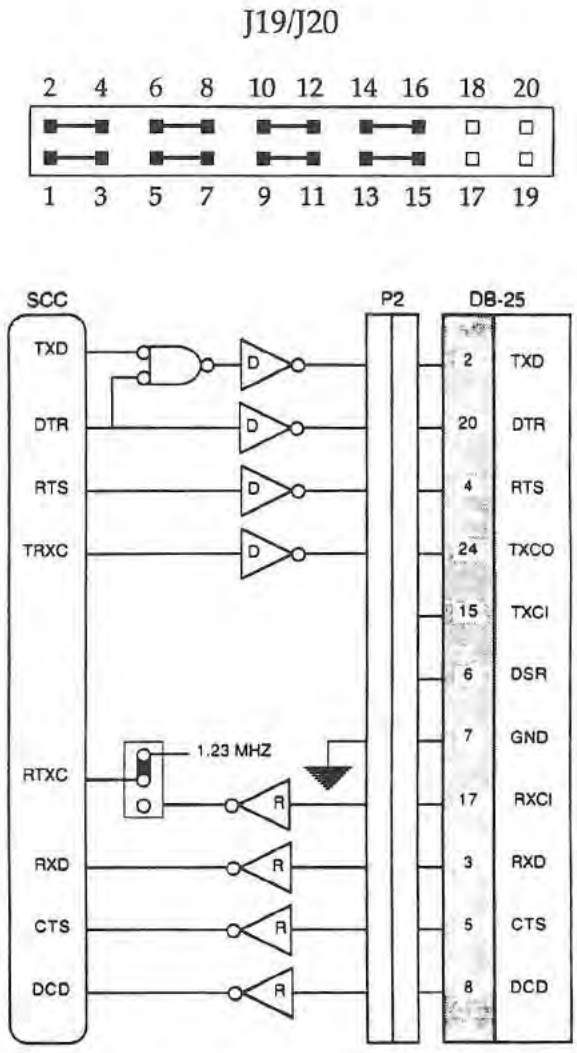
2

## Serial Port 2 (J19) and Port 3 (J20) Configuration Select Headers

Header J19 and header J20 configure serial ports 3 and 2, respectively, on connector P2 and J14 as a modem (DCE) for connection to a terminal or as a terminal (DTE) for connection to a modem. The module is shipped factory configured as DCE (to terminal). Configuration is accomplished by installing jumpers as shown below. The schematic representation of each configuration is also shown. Refer to the *Connector J14 Interconnect Signals* section in this chapter for pinouts.




Ports 2/3 Configured as DCE




Ports 2/3 Configured as DTE

## Installation Instructions

When the MVME143S has been configured as desired, it can be installed in the system as follows:

1. Turn all equipment power OFF and disconnect power cable from ac power source.

> CAUTION
>
> **Connecting Modules While Power is Applied May Result in Damage to Components on the Module.**

> **WARNING**
>
> **DANGEROUS VOLTAGES, CAPABLE OF CAUSING DEATH, ARE PRESENT IN THIS EQUIPMENT. USE EXTREME CAUTION WHEN HANDLING, TESTING, AND ADJUSTING.**

2. Remove chassis cover as instructed in the equipment user's manual.
3. Remove the filler panel from the appropriate card slot at the front of the chassis. If the MVME143S is configured as the system controller, it must be installed in the left most card slot (slot 1) to correctly initiate the bus grant daisy chain and the IACK daisy chain.
4. Insert the MVME143S into the selected card slot. Be sure module is seated properly into the connectors on the backplane. Fasten the module in the chassis with the screws provided.
5. Remove IACK and BG jumpers from header on chassis backplane for card slot the MVME143S is installed in.

6. Connect any desired cables to the MVME143S module at the P2 backplane connector, to mate with optional peripherals at the EIA-232-D serial ports, and optional remote RESET switch. These cables are user supplied.
7. Replace cover and turn equipment power ON.

A suggested cabling arrangement is shown in Figure 2-2. Connector P2 is a standard DIN 41612 96-pin male connector. A 64-pin (rows A and C) DIN 41612 female connector may be used to connect to P2. This cable may be separated and connected to female DB-25 connector for each of the three ports. Refer to the *MVME143S Support Information User's Manual* for the pinouts of connector P2. Note that the optional remote RESET switch may be connected to pin 64 (P2-A32) and pin 63 (P2-C32 GND).

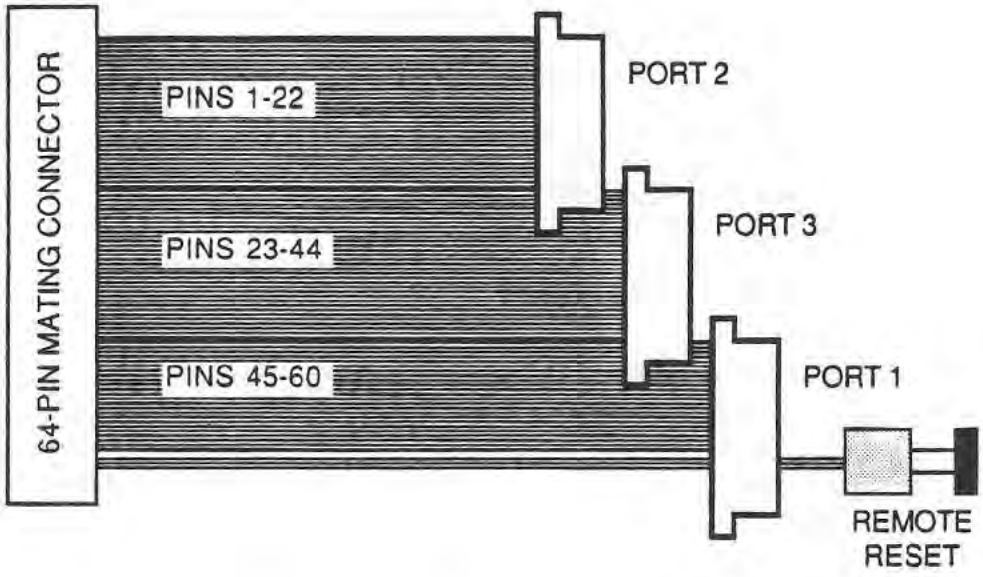



**Figure 2-2.** Cabling Connections to P2

## Connector J14 Interconnect Signals

Connector J14 is a 34-pin male connector providing interconnections for serial port 2 and serial port 3. These two serial ports are also available through connector P2. In addition, J14 provides +5, +12, and -12 Vdc through three separate 1 A fuses. FUSE LED on the front panel lights red when any of these fuses are blown. Each pin connection and signal mnemonic for the connector is listed in Table 2-3. Refer to Table 2-2 for descriptions of the signals.

**Table 2-1.** EIA-232-D Connector J14 Interconnect Signals

| Pin Number | Signal Mnemonic | Pin Number | Signal Mnemonic |
|---|---|---|---|
| 1 | Not used. | 18 | Not used. |
| 2 | TXD3 | 19 | TXD2 |
| 3 | TXCI3 | 20 | TXCI2 |
| 4 | RXD3 | 21 | RXD2 |
| 5 | Not used. | 22 | Not used. |
| 6 | RTS3 | 23 | RTS2 |
| 7 | RSCI3 | 24 | RSCI2 |
| 8 | CTS3 | 25 | CTS2 |
| 9 | Not used. | 26 | +5 V FUSE |
| 10 | DSR3 | 27 | DSR2 |
| 11 | Not used. | 28 | Not used. |
| 12 | GND | 29 | GND |
| 13 | DTR3 | 30 | DTR2 |
| 14 | DCD3 | 31 | DCD2 |
| 15 | Not used. | 32 | +12 V FUSE |
| 16 | Not used. | 33 | -12 V FUSE |
| 17 | Not used. | 34 | Not used. |

2

## Connector J18 Interconnect Signals

The MC68901 MFP debug port (port 1) is brought out through connector J18 and P2. J18 is a female DB-25 connector. It has a metal shell and jack posts that are electrically connected to the MVME143S front panel. If the MVME143S front panel is electrically connected to the chassis ground, then the shell and jack posts on J9 are connected to ground. This allows shielded cabling to be used for effective reduction of EMI and EMC problems. Each pin connection, signal mnemonic, and signal characteristic for the connector is listed in Table 2-4.

**Table 2-2.** EIA-232-D Connector J18 Interconnect Signals

| Pin Number | Signal Mnemonic | Signal Name and Description |
|---|---|---|
| 1 | | Not used. |
| 2 | TXD1 | EIA-232-D TRANSMITTED DATA — input to the SI pin of the MFP if port 1 is configured as DCE, and output from SO pin of the MFP if port 1 is configured as DTE. |
| 3 | RXD1 | EIA-232-D RECEIVED DATA — output from the SO pin of the MFP if port 1 is configured as DCE, and input to the SI pin of the MFP if port 1 is configured as DTE. |
| 4 | RTS1 | EIA-232-D REQUEST TO SEND — input to the GPIO0 pin of the MFP if port 1 is configured as DCE, and output from GPIO3 pin of the MFP if port 1 is configured as DTE. |
| 5 | CTS1 | EIA-232-D CLEAR TO SEND — output from the GPIO3 pin of the MFP if port 1 is configured as DCE, and input to GPIO0 pin of the MFP if port 1 is configured as DTE. |
| 6 | DSR1 | EIA-232-D DATA SET READY — pulled up to +12 Vdc when port 1 is configured as DCE. This pin is not connected in DTE configuration. |

**Table 2-2.** EIA-232-D Connector J18 Interconnect Signals (cont'd)

| Pin Number | Signal Mnemonic | Signal Name and Description |
|---|---|---|
| 7 | GND | EIA-232-D SIGNAL GROUND/COMMON RETURN — connected to the MVME143S ground plane. Not connected to chassis ground by the MVME143S. |
| 8 | DCD1 | EIA-232-D RECEIVED LINE SIGNAL DETECTOR — pulled up to +12 Vdc when port 1 is configured as DCE. This pin is not connected in DTE configuration. |
| 9-19 | | Not used. |
| 20 | DTR1 | EIA-232-D DATA TERMINAL READY — pulled up to +12 Vdc when port 1 is configured as DTE. This pin is not connected in DCE configuration. |
| 21-25 | | Not used. |

# CHAPTER 3
# OPERATING INSTRUCTIONS

3

## Introduction

This chapter provides necessary information to use the MVME143S module in a system configuration. This includes controls and indicators, memory map details, software initialization of the module, and system considerations.

## Controls and Indicators

The MVME143S module has ABORT and RESET switches, and FAIL, STATUS, RUN, SCON, and FUSE indicators, all of which are located on the front panel of the module.

### ABORT Switch S1

The ABORT switch is debounced and brought into the PI/T pin H3 which may be programmed to interrupt the MPU on level 7.

### RESET Switch S2

If the front panel RESET switch is enabled (header J4), pressing the switch resets all onboard devices (including the MPU) and asserts SYSRESET* if the MVME143S is the system controller.

### FAIL, STATUS, RUN, SCON, and FUSE Indicators DS1, DS2, DS3, DS4, and DS5

MVME143S has five LEDs: FAIL, STATUS, RUN, SCON, and FUSE. SCON is on (green) when the MVME143S is configured as system controller. RUN is on (green) when MPU address strobe is on, STATUS is on (yellow) when reset is true or when the MPU STATUS* line is low, FAIL is on (red) when the BRDFAIL control bit is high or when VMEbus SYSFAIL* is low while the MVME143S is system controller and FUSE is on (red) when any of the three fuses is blown (+5V, +12V, -12V). The module state is matched with the state of the RUN, STATUS, and FAIL LEDs as shown in Table 3-1.

**Table 3-1.** Front Panel LED Indicators

| FAIL | STATUS | RUN | Description |
|---|---|---|---|
| OFF | OFF | OFF | No power is applied to the module, or the MPU is not the current local bus master. |
| OFF | OFF | ON | Either the MPU is accessing very slow devices or it cannot complete its current external bus cycle. |
| OFF | ON | OFF | MPU is halted or MPU is executing out of onchip cache. |
| OFF | ON | ON | Normal operation. The intensity of the STATUS LED is controlled by MPU STATUS* pin. |
| ON | OFF | OFF | MPU is not current local bus master. Also BRDFAIL has not been cleared since reset or has been set by software. FAIL indicator is also on if MVME143S is system controller and SYSFAIL* is detected low on the VMEbus. |
| ON | OFF | ON | Either the MPU is accessing very slow devices or it cannot complete its current external bus cycle. BRDFAIL has not been cleared since reset or has been set by software. FAIL indicator is also on if MVME143S is system controller and SYSFAIL* is detected low on the VMEbus. |
| ON | ON | OFF | MPU is halted or is executing out of onchip cache and BRDFAIL has not been cleared since reset or has been set by software. FAIL indicator is also on if MVME143S is system controller and SYSFAIL* is detected low on the VMEbus. |
| ON | ON | ON | MPU is running. The intensity of the STATUS LED is controlled by MPU STATUS* pin. Also BRDFAIL has not been cleared since reset or has been set by software. FAIL indicator is also on if MVME143S is system controller and SYSFAIL* is detected low on the VMEbus. |

## Map Decoder

The MC68030 MPU divides the memory map into different address spaces by using the function codes FC2-FC0. The MVME143S has different groups of devices that respond to an MPU bus cycle depending on the address space as shown in Table 3-2.

**Table 3-2.** MVME143S Address Spaces

| FC2 | FC1 | FC0 | Address Space | Devices that Respond |
|---|---|---|---|---|
| 0 | 0 | 0 | Reserved | None (causes local timeout) |
| 0 | 0 | 1 | User Data | All except interrupt handler and MC68882 |
| 0 | 1 | 0 | User Program | All except interrupt handler and MC68882 |
| 0 | 1 | 1 | Reserved | None (causes local timeout) |
| 1 | 0 | 0 | Reserved | None (causes local timeout) |
| 1 | 0 | 1 | Supervisory Data | All except interrupt handler and MC68882 |
| 1 | 1 | 0 | Supervisory Program | All except interrupt handler and MC68882 |
| 1 | 1 | 1 | CPU (IACK) | VMEbus, Z8530, MC68901, MC68230, interrupt handler |
| 1 | 1 | 1 | CPU (Coprocessor) | MC68882 FPC |

### Program and Data Space Memory Map

The memory map of devices that respond in user data, user program, supervisory data, and supervisory program spaces is shown in Table 3-3.

**Table 3-3.** MVME143S Program and Data Space Memory Map

| Physical Address | Devices Accessed | Port Size | Size | Notes |
|---|---|---|---|---|
| $00000000-003FFFFF | Onboard DRAM | D32 | 4Mb | 1 |
| $00400000-00FFFFFF | VMEbus A32/A24 | D32/D16 | 12Mb | 2,3 |
| $01000000-7FFFFFFF | VMEbus A32 | D32/D16 | 2032Mb | 3 |
| $80000000-EFFFFFFF | VMEbus A32 | D32/D16 | 1792Mb | 3,6 |

**Table 3-3.** MVME143S Program and Data Space Memory Map (cont'd)

| Physical Address | Devices Accessed | Port Size | Size | Notes |
|---|---|---|---|---|
| $F0000000-F0FFFFFF | VMEbus A32/A24 | D16 | 16Mb | 2,6 |
| $F1000000-FF7FFFFF | VMEbus A32 | D16 | 2Mb | 6 |
| $FF800000-FF9FFFFF | ROM/EEPROM bank 1 | D16 | 2Mb | 4,8 |
| $FFA00000-FFBFFFFF | ROM/EEPROM bank 2 | D16 | 2Mb | 4,8 |
| $FFC00000-FFEFFFFF | VMEbus A32 | D16 | 3Mb | 6 |
| $FFF00000-FFF1FFFF | ROM/EEPROM bank 1 | D16 | 128Kb | 4,9 |
| $FFF20000-FFF3FFFF | ROM/EEPROM bank 2 | D16 | 128Kb | 4,9 |
| $FFF40000-FFF7FFFF | Reserved | N/A | 256Kb | |
| $FFF80000-FFF8FFFF | MC68901 MFP | D16 | 64Kb | 5,6 |
| $FFF90000-FFF9FFFF | MC68230 PI/T | D16 | 64Kb | 5,6 |
| $FFFA0000-FFFBFFFF | Z8530 SCC | D08 | 128Kb | 6 |
| $FFFC0000-FFFDFFFF | MK48T02 RTC and SRAM | D08 | 128Kb | 6 |
| $FFFE0000-FFFEFFFF | VMEbus interrupter | N/A | 64Kb | 6,7 |
| $FFFF0000-FFFFFFFF | VMEbus short I/O space | D16 | 64Kb | 6 |

**NOTES:** 1. Onboard ROM/PROM/EPROM/EEPROM bank 1 for first four cycles after a reset, onboard DRAM thereafter.

2. VMEbus address size is selectable with VMEA24* control bit.

3. VMEbus data width is selectable with VMED16* control bit.

4. Writes to EEPROM must always be 16-bit wide.

5. MC68901 Multifunction Peripheral (MFP) and MC68230 Parallel Interface Timer (PI/T) appear at the lower bytes.

6. The MVME143S inhibits the MC68030 MPU from caching these locations.

7. A VMEbus interrupt is generated on the selected level every time the MPU performs an access to a location within $FFFE0000-FFFEFFFF.
8. Address range of $FF800000-FFBFFFFF is mapped for VMEbus A32/D16 when jumper on header J13 is installed.
9. These address ranges should only be used if the ROM devices are 64K bit or smaller.



## Local Processor CPU Space Memory Map

Two types of CPU space cycles are supported by the MVME143S: coprocessor and interrupt acknowledge (IACK). All other types of CPU space cycles generated by the MPU are ignored and are terminated with BERR*.

### Coprocessor Interface Map

The only coprocessor on the MVME143S is the MC68882 FPC. The map decoder selects the FPC any time the MPU executes a coprocessor cycle (FC2-FC0 = %111 and A19-A16 = %0010). The MC68882 FPC coprocessor interface register locations in the CPU space that are used for communications between the MPU and the FPC are identified in Table 3-4.

**Table 3-4.** MC68882 Coprocessor Interface Register Map

| Register | A04-A00 | Offset | Data Width | R/W |
|---|---|---|---|---|
| Response | %0000x | $00 | 16-bit | R |
| Control | %0001x | $02 | 16-bit | W |
| Save | %0010x | $04 | 16-bit | R |
| Restore | %0011x | $06 | 16-bit | R/W |
| Reserved | %0100x | $08 | 16-bit | - |
| Command | %0101x | $0A | 16-bit | W |
| Reserved | %0110x | $0C | 16-bit | - |
| Condition | %0111x | $0E | 16-bit | W |
| Operand | %100xx | $10 | 32-bit | R/W |
| Register Select | %1010x | $14 | 16-bit | R |
| Reserved | %1011x | $16 | 16-bit | - |
| Instruction Address | %110xx | $18 | 32-bit | W |
| Operand Address | %111xx | $1C | 32-bit | R/W |

**NOTE:** Read accesses to write-only locations return with all ones; write accesses to read-only locations are ignored. In all cases, the MC68882 terminates the cycle properly with DSACK0*/DSACK1*. If the MC68882 is not present on the MVME143S, then accesses to the FPC result in F-line exceptions.

### Interrupt Acknowledge Map

The MC68030 distinguishes IACK cycles from other CPU space cycles by placing the binary value %1111 on A19-A16. The interrupt handler is selected when FC2-FC0 = $7 and A19-A16 = %1111. The MPU also indicates the level that is being acknowledged with address lines A03-A01. Refer to the *Interrupt Handler* section in Chapter 4 for further details.

NOTICE

While reasonable efforts have been made to assure the accuracy of this document, Motorola, Inc. assumes no liability resulting from any omissions in this document, or from the use of the information obtained therein. Motorola reserves the right to revise this document and to make changes from time to time in the content hereof without obligation of Motorola to notify any person of such revision or changes.



RESTRICTED RIGHTS LEGEND

If the documentation contained herein is supplied, directly or indirectly, to the U.S. Government, the following notice shall apply unless otherwise agreed to in writing by Motorola, Inc.

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Motorola, Inc.
Computer Group
2900 South Diablo Way
Tempe, Arizona 85282

# PREFACE

This manual provides general information, hardware preparation, installation instructions, and functional description for the MVME143S MPU VMEmodule.

This manual is intended for anyone who wants to design OEM systems, supply additional capability to an existing compatible system, or in a lab environment for experimental purposes.

A basic knowledge of computers, and digital logic is assumed.

To use this manual, you should be familiar with the publications listed in the *Related Documentation* paragraph in Chapter 1 of this manual.

The computer programs stored in the read only memories of this device contain material copyrighted by Motorola, Inc., first published 1989, and may be used only under a license such as the License for Computer Programs (Article 14) contained in Motorola's Terms and Conditions of Sale, Rev. 1/79.

**WARNING**

**THIS EQUIPMENT GENERATES, USES, AND CAN RADIATE RADIO FREQUENCY ENERGY AND IF NOT INSTALLED AND USED IN ACCORDANCE WITH THE INSTRUCTIONS MANUAL, MAY CAUSE INTERFERENCE TO RADIO COMMUNICATIONS. IT HAS BEEN TESTED AND FOUND TO COMPLY WITH THE LIMITS FOR A CLASS A COMPUTING DEVICE PURSUANT TO SUBPART J OF PART 15 OF FCC RULES, WHICH ARE DESIGNED TO PROVIDE REASONABLE PROTECTION AGAINST SUCH INTERFERENCE WHEN OPERATED IN A COMMERCIAL ENVIRONMENT. OPERATION OF THIS EQUIPMENT IN A RESIDENTIAL AREA IS LIKELY TO CAUSE INTERFERENCE IN WHICH CASE THE USER, AT HIS OWN EXPENSE, WILL BE REQUIRED TO TAKE WHATEVER MEASURES NECESSARY TO CORRECT THE INTERFERENCE.**

# SAFETY SUMMARY
# SAFETY DEPENDS ON YOU

***The following general safety precautions must be observed during all phases of operation, service, and repair of this equipment. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the equipment. Motorola Inc. assumes no liability for the customer's failure to comply with these requirements. The safety precautions listed below represent warnings of certain dangers of which we are aware. You, as the user of the product, should follow these warnings and all other safety precautions necessary for the safe operation of the equipment in your operating environment.***

**GROUND THE INSTRUMENT.**
To minimize shock hazard, the equipment chassis and enclosure must be connected to an electrical ground. The equipment is supplied with a three-conductor ac power cable. The power cable must either be plugged into an approved three-contact electrical outlet or used with a three-contact to two-contact adapter, with the grounding wire (green) firmly connected to an electrical ground (safety ground) at the power outlet. The power jack and mating plug of the power cable meet International Electrotechnical Commission (IEC) safety standards.

**DO NOT OPERATE IN AN EXPLOSIVE ATMOSPHERE.**
Do not operate the equipment in the presence of flammable gases or fumes. Operation of any electrical equipment in such an environment constitutes a definite safety hazard.

**KEEP AWAY FROM LIVE CIRCUITS.**
Operating personnel must not remove equipment covers. Only Factory Authorized Service Personnel or other qualified maintenance personnel may remove equipment covers for internal subassembly or component replacement or any internal adjustment. Do not replace components with power cable connected. Under certain conditions, dangerous voltages may exist even with the power cable removed. To avoid injuries, always disconnect power and discharge circuits before touching them.

**DO NOT SERVICE OR ADJUST ALONE.**
Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

**USE CAUTION WHEN EXPOSING OR HANDLING THE CRT.**
Breakage of the Cathode-Ray Tube (CRT) causes a high-velocity scattering of glass fragments (implosion). To prevent CRT implosion, avoid rough handling or jarring of the equipment. Handling of the CRT should be done only by qualified maintenance personnel using approved safety mask and gloves.

**DO NOT SUBSTITUTE PARTS OR MODIFY EQUIPMENT.**
Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification of the equipment. Contact your local Motorola representative for service and repair to ensure that safety features are maintained.

**DANGEROUS PROCEDURE WARNINGS.**
Warnings, such as the example below, precede potentially dangerous procedures throughout this manual. Instructions contained in the warnings must be followed. You should also employ all other safety precautions which you deem necessary for the operation of the equipment in your operating environment.

**WARNING**

**Dangerous voltages, capable of causing death, are present in this equipment.
Use extreme caution when handling, testing, and adjusting.**

(3/91)

# CONTENTS

# APPENDICES



# INDEX



# FIGURES

# TABLES

# CHAPTER 1
# GENERAL INFORMATION

## Introduction

This manual provides general information, preparation for use and installation instructions, operating instructions, and functional description for the MVME143S MPU VMEmodule (referred to as the MVME143S throughout this manual).

## Model Designations

The MVME143S is available in several configurations which are summarized in the following table. The main differences between the versions is processor speed.

MVME143S Model Designations

| Model Number | Clock Speed | PCC | Memory | Parity |
|---|---|---|---|---|
| MVME143S | 16 MHz | MC68882 | 4Mb | Yes |
| MVME143S-2 | 25 MHz | MC68882 | 4Mb | Yes |

## Features

The features of the MVME143S include:

- Double-high/single-wide VMEboard
- Address 32/Data 32 (A32/D32) VMEbus master (A32/D16, A24/D32, A24/D16 compatible) interface
- MC68030 Enhanced 32-bit microprocessor
- MC68882 Enhanced Floating Point Coprocessor (FPC)
- 4Mb of shared local DRAM, 32-bit wide with byte-wide parity
- Programmable VMEbus base address for shared DRAM
- Four 32-pin sockets for ROM/PROM/EPROM/EEPROM (two 16-bit banks)

- Three 8-bit programmable timers for tick and watchdog functions
- One 24-bit timer
- Battery backup real-time clock
- 2Kb of battery backup SRAM
- A32/D32 VMEbus master interface
- Front panel asynchronous serial EIA-232-D debug port
- Front panel accessible EIA-232-D multiprotocol serial port (dual)
- Two multiprotocol (synchronous/asynchronous) serial ports plus debug port on connector P2
- VMEbus system controller functions with SGL/PRI/RRS arbiter
- Four-level bus requester (jumper selectable level)
- VMEbus interrupter (programmable level (1 through 7) and vector ($F0 through FF))
- VMEbus interrupt handler for all local and VMEbus interrupts
- Mailbox interrupt in short I/O space
- Front panel FAIL, STATUS, RUN, SCON, and FUSE LED indicators
- Front panel ABORT and RESET switches
- Remote reset through edge connector P2

## Specifications

Specifications for the MVME143S are shown in Table 1-1.

**Table 1-1.** MVME143S Specifications

| Characteristics | Specifications |
|---|---|
| Power requirements (with full set of ROM/PROM/ EPROM/EEPROM) | +5 Vdc @ 5 A (typical), 7 A (maximum)<br>+12 Vdc @ 100 mA (typical), 250 mA (maximum)<br>-12 Vdc @ 100 mA (typical), 250 mA (maximum) |
| Microprocessor | MC68030 (MPU) |
| Coprocessor | MC68882 (FPC) |
| Clock signal to MPU and FPC | 16 MHz (MVME143S)<br>25 MHz (MVME143S-2) |
| Addressing | 4 gigabytes (on and offboard) |
| ROM/PROM/EPROM/EEPROM | 256Kb maximum: four sockets (two banks of two each, 16 bits wide) for 8K x 8, 16K x 8, 32K x 8, 64K x 8, 128K x 8, 256K x 8, 512K x 8, or 1M x 8 ROM/PROM/EPROM devices or 2K x 8, 8K x 8, or 32K x 8 EEPROM devices |
| Dynamic RAM | 4Mb (32 bits wide) with bytewide parity |
| Serial I/O ports | Two multiprotocol EIA-232-D through connector P2<br>One asynchronous EIA-232-D debug serial port on front panel and connector P2<br>Front panel accessible multiprotocol EIA-232-D serial port (dual) |
| Battery backup real-time clock | 1 second resolution |
| Battery backup SRAM | 2Kb |

**Table 1-1.** MVME143S Specifications (cont'd)

| Characteristics | Specifications |
|---|---|
| Timers | 5 total (4 available) |
| Software tick | 8-bit |
| Watchdog | 8-bit |
| Baud rate generator | 8-bit |
| Delay mode only | 8-bit |
| Parallel interface timer | 24-bit |
| Bus configuration | 32-bit or 24-bit address and 32-bit or 16-bit data |
| Operating temperature | 0° to 55° C at point of entry of forced air (approximately 350 LFM) |
| Storage temperature | -40° to 85° C |
| Relative humidity | 5% to 90% (non-condensing) |
| Physical characteristics | (excluding front panel) |
| Height<br>Depth<br>Thickness | 9.187 inches (233.35 mm)<br>6.299 inches (160.00 mm)<br>0.063 inches (1.6 mm) |

## Cooling Requirements

The Motorola MVME143S VMEmodule is specified, designed, and tested to operate reliably with an incoming air temperature range from 0 degrees C to +55 degrees C (+32 degrees to +131 degrees F) with forced air cooling at a velocity typically achievable by using a 71 CFM axial fan. Temperature qualification is performed in a standard Motorola VMEsystem 1000 chassis. Twenty-five watt load boards are inserted in two card slots, one on each side, adjacent to the board under test, to simulate a high power density system configuration. An assembly of two axial fans, rated at 71 CFM per fan, is placed directly under the VME card cage. The incoming air temperature is measured between the fan assembly and the card cage, where the incoming airstream first encounters the module under test. Test software is executed as the module is subjected to ambient temperature variations. Case temperatures of critical, high power density integrated circuits are monitored to ensure component vendors specifications are not exceeded.

While the exact amount of airflow required for cooling depends on the ambient air temperature and the type, number, and location of boards and other heat sources, adequate cooling can usually be achieved with 6 CFM and 350 LFM flowing over the module. Less airflow is required to cool the module in environments having lower maximum ambients. Under more favorable thermal conditions, it may be possible to operate the module reliably at higher than +55 degrees C with increased airflow. It is important to note that there are several factors, in addition to the rated CFM of the air mover, which determine the actual volume and speed of air flowing over a module.

## FCC Compliance

This VMEmodule (MVME143S) was tested in an FCC-compliant chassis, and meets the requirements for Class A equipment. FCC compliance was achieved under the following conditions:

1. Shielded cables on all external I/O ports.
2. Cable shields connected to earth ground via metal shell connectors bonded to a conductive module front panel.
3. Conductive chassis rails connected to earth ground. This provides the path for connecting shields to earth ground.
4. Front panel screws properly tightened.

For minimum RF emissions, it is essential that the conditions above be implemented; failure to do so could compromise the FCC compliance of the equipment containing the module.

## General Description

The MVME143S is a double-high VMEmodule. It occupies one slot in a VME system and requires power from connectors P1 and P2. The module has large onboard DRAM (4Mb), ROM/PROM/EPROM/EEPROM capability (256Kb), serial ports including debug port, floating point coprocessor, tick timer, watchdog timer, real-time clock with battery backup SRAM, and VMEbus interface with system controller functions.

The MVME143S is a single-board MPU module intended to be used in a single processor system, but not stand-alone. It is an excellent choice for applications requiring real-time operation such as industrial automation and robotics.

The MVME143Bug debug monitor firmware package (installed on the module) offers debug commands, up/downline load, and disk bootstrap load commands, as well as a full set of onboard diagnostics and a one-line assembler/disassembler.

Note that the MVME143S contains no parallel ports. To use a parallel device, such as a printer, with the MVME143S, it is necessary to add a module such as the MVME050 System Controller Module to the system.

## Related Documentation

The following publications are applicable to the MVME143S and may provide additional helpful information. If not shipped with this product, they may be purchased from Motorola, Inc, Technical Literature Center, 1919 West Fairmont Drive, Suite 8, Tempe, AZ 85282; Telephone 1-800-458-6443; FAX (602) 438-0240. Non-Motorola documents may be obtained from the sources listed.

| Document Title | Motorola Publication Number |
|---|---|
| MVME143 Debug Monitor User's Manual | MVME143BUG |
| VERSAdos to VME Hardware and Software Configuration User's Manual | |
| MC68030 32-Bit Enhanced Microprocessor User's Manual | MC68030UM |
| MC68881/MC68882 Floating Point Coprocessor User's Manual | MC68881UM |
| MC68901 Multifunction Peripheral Data Book | MC68901/D |
| MC68230 Parallel Interface/Timer Data Sheet | MC68230/D |
| MVME143S MPU VMEmodule Support Information (Refer to the *Support Information* section section in this chapter) | SIMVME143S |

**NOTE:** Although not shown in the above list, each Motorola Computer Group manual publication number is suffixed with characters which represent the revision level of the document, such as /D2 (the second revision of a manual); each supplement bears the same number as the manual but has a suffix such as /A1 (the first supplement to the manual).

The following publications are available from the sources indicated.

ANSI/IEEE Std 1014-1987 Versatile Backplane Bus: The Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, New York, NY 10017, USA. (VMEbus specification)

MK48T02 2K x 8 Zeropower/Timekeeper RAM Data Sheet, Thompson Components, Mostek, 1310 Electronics Drive, Carrollton, TX 75606

Z8530 Serial Communications Controller Data Sheet, Zilog, Inc., Corporate Communications, Building A, 1315 Dell Ave, Campbell, CA 95008

## Support Information

The SIMVME143S manual contains the connector interconnect signal information, parts list, and the schematics for the MVME143S.

This manual may be obtained free of charge from Motorola, Inc, Computer Group Technical Literature Center, 1919 West Fairmont Drive, Suite 8, Tempe, AZ 85282; Telephone 1-800-458-6443; FAX (602) 438-0240.

## Manual Terminology

Throughout this manual, a convention has been maintained whereby data and address parameters are preceded by a character which specifies the numeric format as follows:

| | | |
|---|---|---|
| $ | dollar | specifies a hexadecimal number |
| % | percent | specifies a binary number |
| & | ampersand | specifies a decimal number |

Unless otherwise specified, all address references are in hexadecimal throughout this manual.

An asterisk (*) following the signal name for signals which are level significant denotes that the signal is true or valid when the signal is low.

An asterisk (*) following the signal name for signals which are edge significant denotes that the actions initiated by that signal occur on high to low transition.

In this manual, assertion and negation are used to specify forcing a signal to a particular state. In particular, assertion and assert refer to a signal that is active or true; negation and negate indicate a signal that is inactive or false. These terms are used independently of the voltage level (high or low) that they represent.

# CHAPTER 2
# HARDWARE PREPARATION AND INSTALLATION



## Introduction

This chapter provides unpacking instructions, hardware preparation, and installation instructions for the MVME143S.

## Unpacking Instructions

NOTE

If the carton is damaged upon receipt, request carrier's agent be present during unpacking/ inspection of equipment.

Unpack equipment from shipping carton. Refer to packing list and verify that all items are present. Save packing material for storing and reshipping of equipment.

## Hardware Preparation

To select the desired configuration and ensure proper operation of the MVME143S, certain modifications may be made before installation. These changes are made through jumper or wire wrap arrangements on the headers. The location of the headers, switches, LEDs, and connectors on the MVME143S is illustrated in Figure 2-1. The module has been factory tested and is shipped with factory-installed jumper configurations that are described in the following paragraphs with each header description. The module is operational with factory-installed jumper configurations. The module is configured to provide the system functions required for a VMEbus system. It is necessary to make changes in the jumper and wire wrap arrangements for the following conditions:

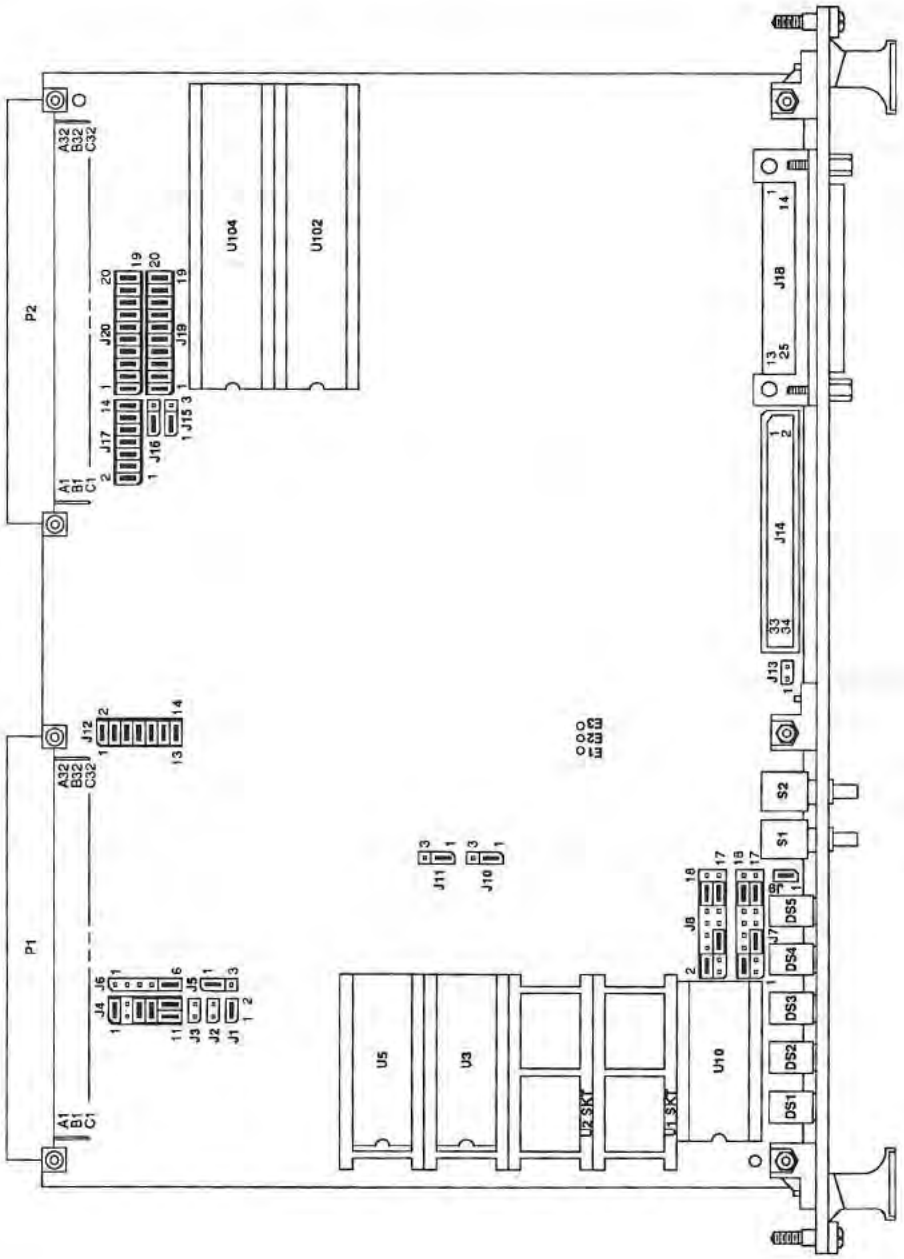



**Figure 2-1.** MVME143S Header Locations

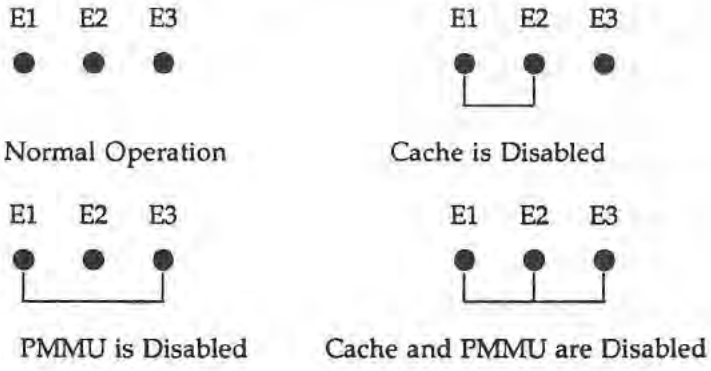
- Onchip cache and PMMU control select (E1, E2, E3)
- System controller enable/disable select (J1)
- VMEbus arbiter mode select (J2)
- VMEbus requester mode select (J3)
- VMEbus requester level select (J4, J6)
- Global timeout select (J5)
- ROM/EEPROM size select (J7, J8)
- RESET switch enable/disable select (J9)
- Serial clock select (J10)
- Local timeout select (11)
- VMEbus interrupt handler select (J12)
- EPROM address map select (J13)
- Front panel accessible dual serial port (J14)
- SCC RTXCA source select (J15)
- SCC RTXCB source select (J16)
- Serial port 1 configuration select (J17)
- Front panel debug port (J18)
- Serial port 2 and port 3 configuration select (J19, J20)

## Onchip Cache and PMMU Control Select Terminals (E1, E2, E3)

The MC68030 MPU includes a 256-byte onchip instruction cache and a 256-byte onchip data cache that are accessed by logical (virtual) addresses. These caches improve performance by reducing external bus activity and increased instruction throughput. This cache can be disabled by connecting terminal E1 to terminal E2.

The MC68030 MPU includes a Paged Memory Management Unit (PMMU) that translates addresses in parallel with instruction execution and internal cache accesses. This PMMU can be disabled by connecting terminal E1 to terminal E3.

These terminals are located on the module to the right of the RESET switch near the front panel bracket. The module is shipped for normal operation (no connections).




## System Controller Enable/Disable Select Header (J1)

Header J1 allows you to select the MVME143S as system controller. The module is not system controller with the jumpers removed. The module is shipped with jumpers installed (system controller).


## VMEbus Arbiter Mode Select Header (J2)

Header J2 provides a choice of Prioritized (PRI) mode or Round-Robin-Select (RRS) mode of arbitration. The PRI arbiter prioritizes the four bus request lines, from BR0* (the lowest) to BR3* (the highest), and responds with BG0IN* through BG3IN*, as appropriate. The Round-Robin-Select arbiter, upon release of the bus, scans the bus until a request is found and sends a bus grant over the appropriate line. The as-shipped factory configuration is with the jumper removed (PRI mode).

## VMEbus Requester Mode Select Header (J3)

Header J3 provides a choice of fairness mode (RONR) (Request On No Request) on or fairness mode off. As long as a system has no more than four masters or interrupt handlers (one per bus request line), the Round-Robin-Select (RRS) arbitration algorithm assures fairness. That is, no bus master is prevented from accessing the bus indefinitely by higher level requests. In systems with more than four masters or interrupt handlers, fairness can be provided (jumper installed on header J3). The as-shipped factory configuration is with the jumper removed (fairness off).

J3

1 □ □ 2

Fairness (RONR) Off

J3

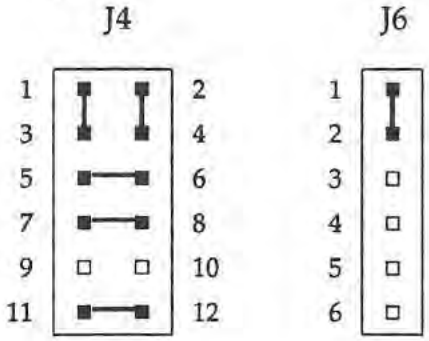
1 ■—■ 2

Fairness (RONR) On

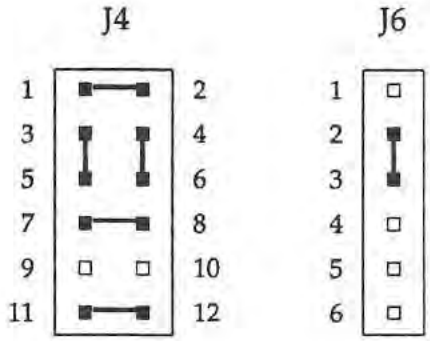
2

## VMEbus Requester Level Select Headers (J4, J6)

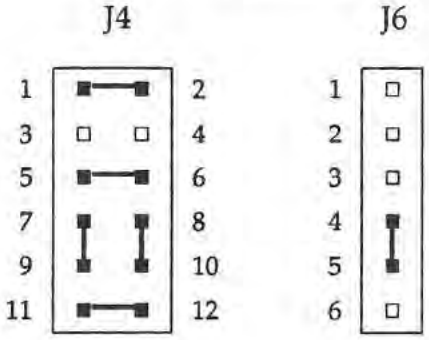
Headers J4 and J6 are used to select the VMEbus request level. Any one of four request levels can be selected. The module is shipped with the factory configuration of level 3.



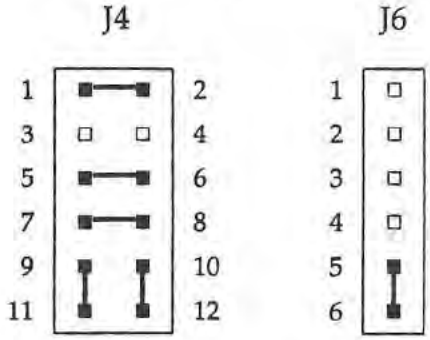

Bus Grant/Request Level 0




Bus Grant/Request Level 1



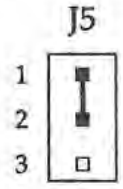

Bus Grant/Request Level 2



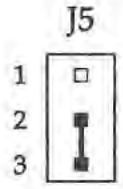

Bus Grant/Request Level 3

## Global Timeout Select Header (J5)

Header J5 provides a choice of the global timeout period to be 64 µs, 128 µs, or infinity. When the PI/T Global Timeout Enable (GTOEN) bit is high, the global bus and the arbitration timers are enabled. The as-shipped factory configuration is with the timeout period of 64 µs. A timeout period of infinity may be set by removing the jumper.




Timeout period is 64 µs for system timer and arbitration timer




Timeout period is 128 µs for system timer and arbitration timer

## ROM/EEPROM Size Select Headers (J7, J8)

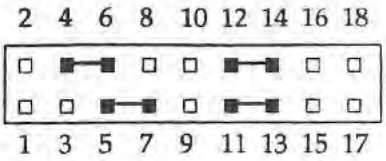
The MVME143S supports various sizes of ROM/PROM/EPROM/EEPROM devices. The module must be configured for the device type and size used as shown below. Each pair of sockets may be individually configured. Four 32-pin sockets are provided on the module. They are organized as two banks with two sockets per bank. Sockets U3 (even) and U5 (odd) form bank 1. Sockets U1 (even) and U2 (odd) form bank 2. Each bank appears as a 16-bit word port to the MPU. Each bank can be configured for 8K x 8, 16K x 8, 32K x 8, 64K x 8, 128K x 8, 256K x 8, 512K x 8, or 1M x 8 ROM/PROM/EPROM or for 2K x 8, 8K x 8, or 32K x 8 EEPROM devices.

There are several different algorithms for erasing/writing to EEPROMs, depending on the manufacturer. The MVME143S supports only those devices which have a "static RAM" compatible erase/write mechanism.

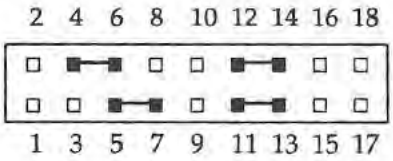
The module is shipped with factory configuration for 64K x 8 ROM/PROM/ EPROM.

2

J7 - Bank 2

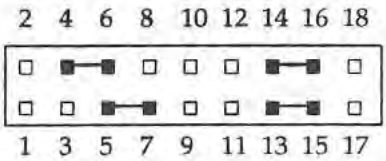



J8 - Bank 1

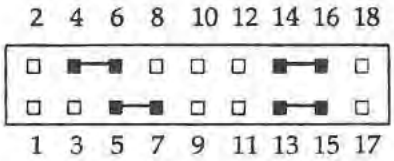



8K x 8K or 16K x 8 ROM/PROM/EPROM

J7 - Bank 2

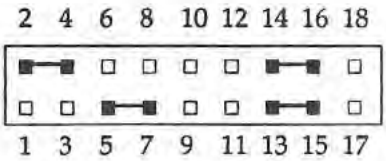



J8 - Bank 1

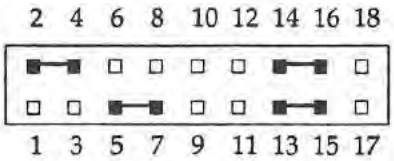



32K x 8 ROM/PROM/EPROM

J7 - Bank 2

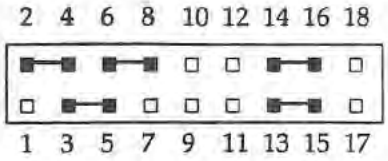



J8 - Bank 1

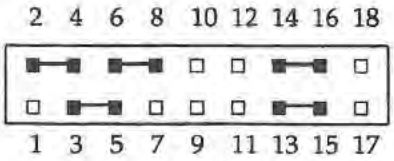



64K x 8 ROM/PROM/EPROM

J7 - Bank 2

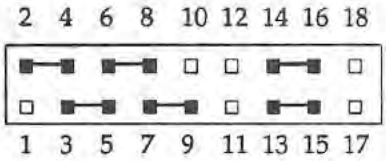



J8 - Bank 1

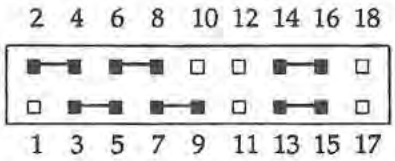



128K x 8 ROM/PROM/EPROM

J7 - Bank 2

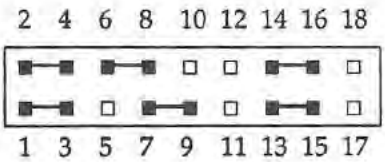



J8 - Bank 1

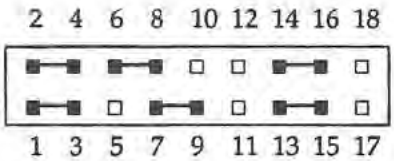



256K x 8 ROM/PROM/EPROM

J7 - Bank 2



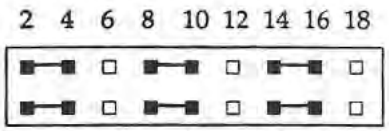

J8 - Bank 1



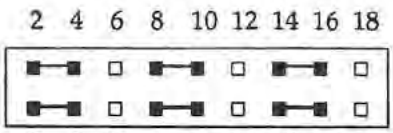

512K x 8 ROM/PROM/EPROM

2

J7 - Bank 2

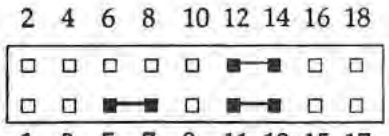



J8 - Bank 1

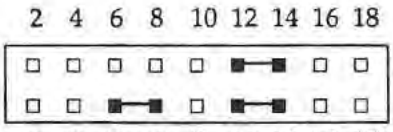



1M x 8 ROM/PROM/EPROM

J7 - Bank 2

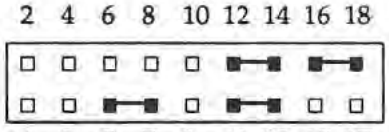



J8 - Bank 1

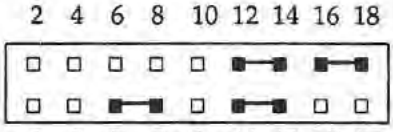



2K x 8 or 8K x 8 EEPROM

J7 - Bank 2




J8 - Bank 1

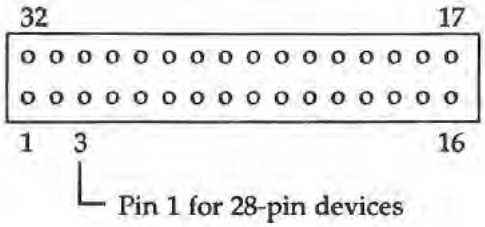



32K x 8 EEPROM

The following figures show the definitions of the ROM/PROM/EPROM/EEPROM socket pins, depending upon the configuration used. The address lines shown are device address lines not local bus address lines. Devices with 28 pins are installed with pin 1 of the device aligned with pin 3 of the socket as shown in the figures below. The configurations shown in the figures are as follows:

| Configuration Number | Device Type |
|---|---|
| 1 | 8K x 8, 16K x 8 EPROM |
| 2 | 32K x 8 EPROM |
| 3 | 64K x 8 EPROM |
| 4 | 2K x 8 (28-pin), 8K x 8 EEPROM |
| 5 | 128K x 8 EPROM |
| 6 | 256K x 8 EPROM |
| 7 | 512K x 8 EPROM |
| 8 | 1M x 8 EPROM |
| 9 | 32K x 8 EEPROM |

The sockets are installed on the module with pins oriented as shown below:




| Configuration | | | | | | | | Configuration | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | | | | | 4 | 3 | 2 | 1 |
| | | | | 1 | | | 32 | +5V | +5V | +5V | +5V |
| A17 | A17 | A17 | A17 | 2 | | | 31 | | | | |
| +5V | +5V | A16 | | 3 | 1 | 28 | 30 | +5V | +5V | +5V | +5V |
| A13 | A13 | A13 | A13 | 4 | 2 | 27 | 29 | WE* | A15 | A15 | WE* |
| A8 | A8 | A8 | A8 | 5 | 3 | 26 | 28 | A14 | A14 | A14 | A14 |
| A7 | A7 | A7 | A7 | 6 | 4 | 25 | 27 | A9 | A9 | A9 | A9 |
| A6 | A6 | A6 | A6 | 7 | 5 | 24 | 26 | A10 | A10 | A10 | A10 |
| A5 | A5 | A5 | A5 | 8 | 6 | 23 | 25 | A12 | A12 | A12 | A12 |
| A4 | A4 | A4 | A4 | 9 | 7 | 22 | 24 | OE* | OE* | OE* | OE* |
| A3 | A3 | A3 | A3 | 10 | 8 | 21 | 23 | A11 | A11 | A11 | A11 |
| A2 | A2 | A2 | A2 | 11 | 9 | 20 | 22 | CE* | CE* | CE* | CE* |
| A1 | A1 | A1 | A1 | 12 | 10 | 19 | 21 | D7 | D7 | D7 | D7 |
| D0 | D0 | D0 | D0 | 13 | 11 | 18 | 20 | D6 | D6 | D6 | D6 |
| D1 | D1 | D1 | D1 | 14 | 12 | 17 | 19 | D5 | D5 | D5 | D5 |
| D2 | D2 | D2 | D2 | 15 | 13 | 16 | 18 | D4 | D4 | D4 | D4 |
| GND | GND | GND | GND | 16 | 14 | 15 | 17 | D3 | D3 | D3 | D3 |