ICP6000(X)/ICP9000(X) Hardware Description and Theory of Operation

DC 900-0408E

Simpact, Inc.
9210 Sky Park Court
San Diego, CA 92123
January 1998

Simpact, Inc.
9210 Sky Park Court
San Diego, CA 92123
(619) 565-1865

ICP6000(X)/ICP9000(X) Hardware Description and Theory of Operation

Printed in the United States of America

Contents

List of Figures

List of Tables

Preface

Purpose of Document

This manual describes Simpact's ICP6000(X)/ICP9000(X) front-end communications processor, its architecture, and how it works in the VMEbus-based system. The information in this manual supplements the basic information that appears in the installation guides delivered with the ICP.

Intended Audience

This manual should be read by maintenance technicians, computer system integrators, and software developers who need detailed information about the operating theory and features of the ICP hardware.

Organization of Document

Chapter 1 is a functional overview of the ICP.

Chapter 2 describes how to unpack and configure the ICP.

Chapter 3 describes how to install the ICP.

Chapter 4 defines the mechanical and environmental specifications.

Chapter 5 contains a detailed theory of operation with hardware descriptions.

Chapter 6 describes the hardware self-test diagnostics.

Chapter 7 describes the boot load procedure.

Appendix A describes the daughterboard that provides the 16-port EIA-232 electrical interface.

Appendix B describes the daughterboard that provides the 8-port MIL-STD-188C and EIA-232 electrical interfaces.

Appendix C describes the daughterboard that provides the 8-port V.35 CCITT electrical interface.

Appendix D describes the daughterboard that provides the 8-port EIA-422 electrical interface.

References

While reading this manual, you might also need to refer to the manuals listed below:

- *ICP6000/ICP9000 Software Installation Guide for UNIX Products*, Simpact, Inc., DC 900-0550
- *MC68020 32-Bit Microprocessor User's Manual*, Motorola, Inc.
- *MC68030 32-Bit Microprocessor User's Manual*, Motorola, Inc.
- *MC68901 Multi-function Peripheral Specification*, Motorola, Inc.
- *PTBUG Debug and Utility Program Reference Manual*, Simpact, Inc., DC 900-0424
- *VMEbus Specification Manual*, Revision C.1, VMEbus International Trade Association (VITA)
- *VSI VMEbus Slave Interface ASIC Manual*, Performance Technologies, Inc.
- *Z8530 Serial Communications Controller Technical Manual*, Zilog, Inc.
- *Z85230 Serial Communications Controller Technical Manual*, Zilog, Inc.

Document Conventions

The following conventions apply throughout this document:

- When a specific model number is not mentioned, ICP or ICP6000(X)/ ICP9000(X) refers to all models including the ICP6000X/ICP9000X. The ICP6000 and ICP9000 consist of the same board housed in different form factors.
- A signal name that appears with an overline, for example, $\overline{\text{ECS}}$, indicates that the signal is asserted low.
- Hexadecimal values are shown preceded by the characters “0x” or with the notation (hex).
- Bits are numbered from right to left, beginning with zero. Bit zero is the low-order bit.

Revision History

The revision history of the *ICP6000(X)/ICP9000(X) Hardware Description and Theory of Operation*, Simpact document DC 900-0408E, is recorded below:

Document Revision	Release Date	Description
DC 900-0408E	January 1998	Converted to FrameMaker 5 Updated for the high-speed ICP6000X

Customer Support

If you are having trouble with any Simpact product, call us at 1-800-275-3889 Monday through Friday between 8 a.m. and 5 p.m. Pacific time.

You can also fax your questions to us at (619) 560-2838 or (619)560-2837 any time. Please include a cover sheet addressed to “Customer Service.”

We are always interested in suggestions for improving our products. You can use the report form in the back of this manual to send us your recommendations.

Chapter

1 Overview of the ICP

The ICP (intelligent communications processor) is an ancillary computer dedicated to the processing of communications-related data.

The ICP hardware is a single-board computer that has a central processing unit (CPU), random access memory (RAM), programmable read-only memory (PROM), and input/output (I/O) circuitry. The software consists of an onboard operating system, diagnostic tests, host interface drivers, and application routines.

1.1 Purpose of the ICP

The main purpose of the ICP is to improve the overall computing efficiency of the host computer. To do this, low-level communications tasks that are traditionally performed by the host central processor are migrated to the ICP.

An ICP increases overall system bandwidth by distributing the I/O processing away from the host CPU. In the traditional minicomputer architecture, the host services all I/O requests. This load on the CPU has grown steadily as computer peripherals have become increasingly more powerful. Modern operating systems allow intelligent front-end processors to perform these relatively simple tasks. The result is an overall increase in system throughput.

Table 1–1 summarizes the features of the ICP.

Table 1–1: ICP Hardware Features

Feature	Description
68020 CPU running at 16 MHz or 68030 CPU running at 30 MHz	High-performance processor with linear addressing space
Up to 16 synchronous/asynchronous communications ports	Serial protocols may be implemented
32-channel DMA controller	Full-duplex DMA support provided for all 16 serial communications channels; aggregate DMA bandwidth of 4.0 megabits/second
VMEbus master interface	Local processor has direct access to the VMEbus
VMEbus slave interface ASIC	16-byte dual-ported mailboxes with programmable slave address, programmable interrupt of onboard CPU upon VMEbus access of mailboxes, VMEbus interrupter with programmable level and vector, and reset function initiated by VMEbus software
EEPROM	Storage of startup parameters
EIA-232 console port	Debugging of onboard code using port as console
Two 32-pin JEDEC PROM sockets	Space for power-on diagnostics, debugger, boot loader, and other code

1.2 ICP Applications

The ICP is a general-purpose computer that can do many different tasks. At system startup, the ICP gains its run-time personality from downloaded application software. The application software may be customer-specific or part of a Simpact connectivity product.

The following are some typical applications for the ICP as a front-end processor:

Communications Protocols — ICP-resident software can implement complex communications protocols, freeing the host for end-user tasks. The programmable ICP can easily be reconfigured for many serial protocols, and the software can be modified when protocol requirements change.

Multiple Lines Multiple communication ports enable the ICP to perform network management functions such as message routing, error logging, line-usage monitoring, and various checkout and testing functions.

Data Acquisition ICP-resident software can poll remote stations for status messages or maintain a database of alarm-point states. Additional remote stations can be handled easily and economically by adding ICPs.

Device Control The ICP is effective in formatting or modifying large amounts of output data in ways that are inefficient or inappropriate at the host level.

Chapter

2 Unpacking and Configuration

This chapter describes how to unpack and configure an ICP.

2.1 Antistatic Precautions

The ICP circuit board contains integrated circuits that are sensitive to electrostatic discharge (ESD), that is, static electricity. Improper handling can damage the ICP and result in symptoms ranging from unreliable operation to total failure.

Caution

Never handle the ICP when it is outside its protective bag without wearing a static-guard wrist strap or taking an equivalent grounding precaution.

Standard ESD handling precautions are sufficient to protect the ICP. If you are not familiar with these techniques, take the following precautions:

- Always work at a static-safe workstation, wearing the static-guard wrist strap provided with each installation kit. Directions for using the wrist strap are on the back of the package.
- Leave the board inside its antistatic plastic bag until you are ready to inspect, configure, or install the board.
- When inspecting or configuring the board, keep the solder side in direct contact with the antistatic bag.

- Return the board to the bag immediately after inspection, configuration, or removal from the host backplane.

2.2 Unpacking the ICP

Inspect the shipping carton for any damage that may have occurred during shipment. If such damage is noted, an agent of the shipping carrier should be present at any further unpacking and contents inspection.

Remove the packing list and check it against the items shipped to ensure that you have received the correct board, cables, and so on.

Carefully remove the board from its antistatic bag and observe normal electrostatic discharge precautions (described in Section 2.1) as you inspect the board for shipping damage.

2.3 Initial Inspection

Use the following procedure when inspecting your board:

1. Check overall appearance for breaks or cracks that may have occurred during shipping. If such damage is noted, report it to Simpact, Inc. and *do not proceed* with any further configuration and installation.

2. Check for loose or missing hardware such as unseated ICs.

2.4 Configuration

Configuration of the ICP takes place on two levels:

Jumper Configuration Options — The jumpers are generally set at the factory, either to a factory default or to customer requirements. These options do not usually require field alteration, but should alteration be

necessary, full configuration information is provided in Section 2.5.

Software Programmable Configuration Options

Many of the configuration options normally associated with jumpers or switches, such as bus request level and mode, and interrupt request level, are software/firmware controllable on the ICP. This procedure is described in Section 2.6.

2.5 Jumper Options

Jumper options are typically installed at the factory to a default specification. They may be changed using the information in the following sections.

2.5.1 U39 (ICP6000/ICP9000) Pinout Configurations

The jumpers at locations K1 and K2 allow for variations in the pinout of PROM/ROM socket, U39 (see Table 2-1). When installing 28-pin devices, be sure that pin 14 of the device enters pin 16 of the socket (bottom justified).

Table 2-1: U39 (ICP6000/ICP9000) Configuration

Device Type	Jumpers Installed All Other K1, K2 Removed
32Kx8, 28-pin PROM/ROM	K1-2 to K1-3 K2-2 to K2-3
64Kx8, 28-pin PROM/ROM	K1-2 to K1-3 K2-1 to K2-2
128Kx8, 28-pin PROM/ROM	K2-1 to K2-2
256Kx8, 32-pin PROM/ROM	K1-1 to K1-2 K2-1 to K2-2

2.5.2 U7 (ICP6000X/ICP9000X) Pinout Configurations

The jumpers at locations K1 and K2 allow for variations in the pinout of PROM/ROM socket, U7 (see Table 2-2). When installing 28-pin devices, be sure that pin 14 of the device enters pin 16 of the socket (bottom justified).

Table 2-2: U7 (ICP6000X/ICP9000X) Configuration

Device Type	Jumpers Installed All Other K1, K2, K10, K11 Removed
32Kx8, 28-pin PROM/ROM	K1-2 to K1-3 K2-2 to K2-3 K10-2 to K10-3 K11-2 to K11-3
64Kx8, 28-pin PROM/ROM	K1-2 to K1-3 K2-1 to K2-2 K10-2 to K10-3 K11-2 to K11-3
128Kx8, 32-pin PROM/ROM	K1-2 to K1-3 K2-1 to K2-2 K10-2 to K10-3 K11-2 to K11-3
256Kx8, 32-pin PROM/ROM	K1-1 to K1-2 K2-1 to K2-2 K10-2 to K10-3 K11-2 to K11-3
512Kx8, 32-pin PROM/ROM	K1-1 to K1-2 K2-1 to K2-2 K10-1 to K10-2 K11-2 to K11-3
1024Kx8, 32-pin PROM/ROM	K1-1 to K1-2 K2-1 to K2-2 K10-1 to K10-2 K11-1 to K11-2

2.5.3 U19 (ICP6000/ICP9000) Pinout Configurations

The jumpers at locations K5, K6, and K7 allow for variation in the pinout of the PROM/ROM/EEPROM/SRAM socket, U19 (see Table 2-3). When installing a 28-pin device, be sure that pin 14 of the device enters pin 16 of the socket (bottom justified). EEPROMs used in this application must have write protection built in.

Table 2-3: U19 (ICP6000/ICP9000) Configuration

Device Type	Jumpers Installed All Other K5, K6, K7 Removed
8Kx8, 28-pin EEPROM/SRAM	K6-1 to K6-2 K7-1 to K7-2
32Kx8, 28-pin EEPROM/SRAM	K5-2 to K5-3 K6-1 to K6-2 K7-2 to K7-3
64Kx8, 28-pin PROM/ROM	K5-1 to K5-2 K6-2 to K6-3 K7-2 to K7-3
128Kx8, 32-pin PROM/ROM	K5-1 to K5-2 K6-2 to K6-3
256Kx8, 32-pin PROM/ROM	K5-1 to K5-2 K6-2 to K6-3 K7-1 to K7-2

2.5.4 U8 (ICP6000X/ICP9000X) Pinout Configurations

The jumpers at locations K5, K6, and K7 allow for variation in the pinout of the PROM/ROM/EEPROM/SRAM socket, U8 (see Table 2-4). When installing a 28-pin device, be sure that pin 14 of the device enters pin 16 of the socket (bottom justified). EEPROMs used in this application must have write protection built in.

Table 2-4: U8 (ICP6000X/ICP9000X) Configuration

Device Type	Jumpers Installed All Other K5, K6, K7, K8, K9 Removed
8Kx8, 28-pin EEPROM/SRAM	K6-1 to K6-2 K7-2 to K7-3
32Kx8, 28-pin EEPROM/SRAM	K5-2 to K5-3 K6-1 to K6-2 K7-2 to K7-3
64Kx8, 28-pin PROM/ROM	K5-1 to K5-2 K6-2 to K6-3 K7-2 to K7-3 K8-2 to K8-3 K9-2 to K9-3
128Kx8, 32-pin PROM/ROM	K5-1 to K5-2 K6-2 to K6-3 K7-2 to K7-3 K8-2 to K8-3 K9-2 to K9-3
256Kx8, 32-pin PROM/ROM	K5-1 to K5-2 K6-2 to K6-3 K7-1 to K7-2 K8-2 to K8-3 K9-2 to K9-3
512Kx8, 32-pin PROM/ROM	K5-1 to K5-2 K6-2 to K6-3 K7-1 to K7-2 K8-1 to K8-2 K9-2 to K9-3
1024Kx8, 32-pin PROM/ROM	K5-1 to K5-2 K6-2 to K6-3 K7-1 to K7-2 K8-1 to K8-2 K9-1 to K9-2

2.6 Software Programmable Configuration Options

Many of the configuration options normally associated with jumpers or switches are software/firmware controllable on the ICP.

2.6.1 Configuring the Slave Address Decoder

During hardware installation, parameters for configuring the VMEbus slave address decoder may be programmed into the ICP's EEPROM using PTBUG, the ICP's debugging tool. An additional configurable parameter selects diagnostic and boot load options. Some fixed values must be programmed into the EEPROM as well. Once the EEPROM is programmed, the PROM-resident boot loader reads the values from EEPROM and/or the board select switches and uses them to program the slave address decoder each time the ICP is powered-on or reset.

Table 3–2 shows the EEPROM map and recommended values. (The values shown would be a typical configuration for a Sun-3 or Sun-4.) Entries shown with an asterisk are configurable options. All other entries are not configurable and must be programmed into the EEPROM with the values shown.

Note

If the board select switches are set to select an ICP number, EEPROM values are updated for consistency.

The configurable parameters in Table 3–2 are:

- boot flags
- slave address compare (high and low words of address)
- mode selection
- slave address modifier compare
- slave address modifier compare don't care

The options for the configurable parameters are described in the following sections. After you have selected a value for each of these parameters, follow the installation steps in Chapter 3 to program the EEPROM.

2.6.1.1 Boot Flags

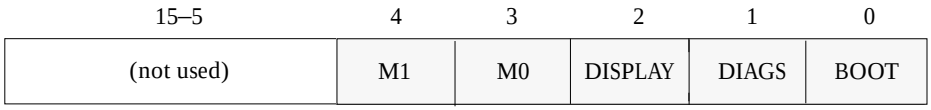
The boot flags word determines the operations performed by the PROM-resident firmware when the ICP is powered-on or reset. The PROM code includes a set of diagnostic self-tests that may be enabled or disabled (see Chapter 6). If enabled, errors encountered during the tests may be displayed on a terminal attached to the console port.

The PROM code also includes a boot loader that allows code and data to be downloaded from the host system (see Chapter 7). On completion of the diagnostics (or immediately after reset if diagnostics are disabled), control is normally transferred to the boot loader. If this function is disabled, control is transferred to PTBUG instead, and the PTBUG "boot" command must be used to execute the boot loader.

The five low-order bits of the boot flag word are used to configure these options, as shown in Figure 2–1. The remaining bits are unused.

2.6.1.2 Slave Address Compare

Addresses 0x2E and 0x30 in the EEPROM specify the base VME address where the mailboxes reside. The default of 0xFE000000 assumes A32 addressing. This address can be changed as required for a given environment and must be changed if more than one ICP exists in the chassis. (Each ICP must have a different base address.) The base address of the mailboxes must be on a 512-byte boundary.



Mnemonic	Name	Description
BOOT	Boot Load Execution	This bit selects the conditions under which the boot loader will execute. 0 = boot on power-up or reset 1 = boot on PTBUG boot command only
DIAGS	Diagnostic Execution	This bit selects whether or not the diagnostic self-tests will execute on power-up and reset. 0 = no diagnostics 1 = diagnostics enabled
DISPLAY	Diagnostic Error Display	This bit enables or disables the display of diagnostic errors (if any) on the console terminal. 0 = no output to console 1 = display errors on console
M1/M0	DRAM Memory Size	These bits enable the message size. 0 0 = 1 megabyte 0 1 = 4 megabytes 1 0 = 8 megabytes

Figure 2–1: EEPROM Boot Flag Word

2.6.1.3 Mode Selection

The value stored in EEPROM address 0x32 selects whether the ICP will be an A32, A24, or A16 slave. Legal values for this word are as follows:

A32 slave	0x04
A24 slave	0x14
A16 slave	0x34

2.6.1.4 Slave Address Modifier Compare

The value stored in EEPROM address 0x34 specifies the address modifier pattern for slave address compares. Table 2–5 lists the legal values for this byte.

Table 2–5: Values for the Slave Address Modifier Compare

Hex Code	Function
3F	A24 Standard Supervisory Block Transfer
3E	A24 Standard Supervisory Program Access
3D	A24 Standard Supervisory Data Access
3B	A24 Standard Non-Privileged Block Transfer
3A	A24 Standard Non-Privileged Program Access
39	A24 Standard Non-Privileged Data Access
30	A24 Don't care about lower four bits
2D	A16 Short Supervisory Access
29	A16 Short Non-Privileged Access
20	A16 Don't care about lower four bits
0F	A32 Standard Supervisory Block Transfer
0E	A32 Standard Supervisory Program Access
0D	A32 Standard Supervisory Data Access
0B	A32 Standard Non-Privileged Block Transfer
0A	A32 Standard Non-Privileged Program Access
09	A32 Standard Non-Privileged Data Access
00	A32 Don't care about lower four bits

2.6.1.5 Slave Address Modifier Compare Don't Care

Any or all of the address modifier bits can be ignored in the comparison. The value stored in location 0x36 in the EEPROM specifies a comparison mask where a one in the bit pattern signifies "don't care."

2.6.2 Configuring the VMEbus Master Interface and Interrupter

The ICP allows the local processor to have direct access to the VMEbus through the VMEbus master interface. In addition, the VMEbus interrupter enables the ICP to generate VMEbus interrupts at a given level and vector. Configuration of the VMEbus master interface and interrupter is accomplished each time the ICP is reset, when the host driver passes the relevant parameters to the ICP's boot loader as the first step of the download procedure (described in Chapter 7).

The following 8-bit external variable arrays must be defined by the user during driver installation. One element in each array must be initialized for each ICP configured into the host system:

icp_vector This variable is the vector number to be supplied to the VMEbus as part of the interrupt process.

icp_level This variable is the interrupt request level at which the interrupt will be generated by the ICP.

icp_request A portion of this variable specifies the VMEbus request mode and the VMEbus bus request level. This value is bit-encoded and must take the form "*xxx*10001," where *xxx* are the user-programmable bits for VMERMS, VMEBR1, and VMEBR0. Refer to Section 5.11 on page 65. The recommended value for this variable is 0xF1.

icp_options This variable specifies the data size and address parameters for VMEbus master mode data transfers. This value is bit-encoded and all bits are user-selectable. Refer to Section 5.11 on page 65 for the selection of these eight bits, which are stored into Control Register 1 by the boot loader. For A32/D32 operation, this variable's recommended value is 0x8D.

After selecting the appropriate values for these variables, refer to the driver installation instructions for your host system in the appropriate ICP software installation guide. These instructions will tell you how to define and initialize the variables. The host driver uses these values to configure the VMEbus master interface and interrupter each time the host system is powered up.

Chapter

3 VMEbus Installation

This chapter describes how to install an ICP. During installation you will perform the following steps:

- Set the board select switches and jumpers
- Plug in the console cable and ICP
- Check and set the EEPROM
- Attach the cables to the upper and lower ports
- Verify the hardware and software installation

You should perform the software installation described in the *ICP6000/ICP9000 Software Installation Guide for UNIX Products* before proceeding with the following hardware installation steps. This avoids extra power-up and power-down cycles.

Step 1:
Turn off the power to the host system.

Caution

The ICP should never be inserted into or removed from the VMEbus cardcage while power is applied. Insertion or removal with power applied could seriously damage the system or the ICP components.

Step 2:

Refer to the documentation provided with your host system for instructions on installing boards in the VMEbus cardcage. This documentation should explain how to access and set the "daisy-chain" backplane jumpers, IACK and Bus Grant. Other preparations, such as setting jumpers on the system's CPU board, may also be required.

Note

If the host system' s daisy-chain backplane jumpers are not set correctly, the system or the ICP will not operate.

Step 3:

Set the board select switches to the desired position as defined by the required ICP address. Figure 3–1 shows where the switches are on the ICP. Table 3–1 shows the valid settings for the switches and the VMEbus address configured for each.

Note

An easy way to remember the orientation of the board select switches is: For debug, set all switches "ON" (toward the console port); for host mode, set all switches "OFF" (toward the VMEbus).

Step 4:

The jumpers from the factory are set for internal clocking. Refer to the appropriate appendix for your electrical interface for figures showing the jumper locations on the ICP.

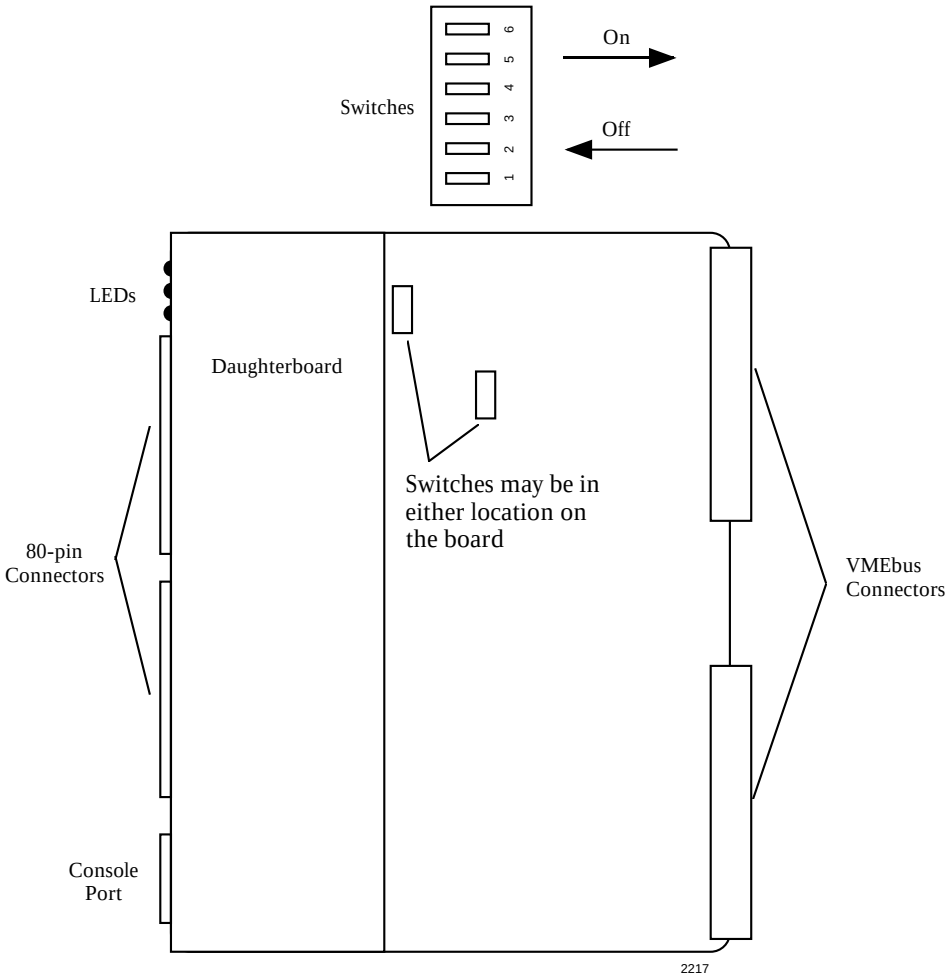


Figure 3–1: ICP6000(X)/ICP9000(X) Switch Settings

Table 3–1: VMEbus Settings

ICP[a] Number	Base Address Select		Board Select Number				VMEbus Slave Address
	SW6	SW5	SW4	SW3	SW2	SW1	
ICP*x*	0	0	*x*	*x*	*x*	*x*	Go to PTBUG
ICP0	1	0	0	0	0	1	F2000000
ICP1	1	0	0	0	1	0	F4000000
ICP2	1	0	0	0	1	1	F6000000
ICP3	1	0	0	1	0	0	F8000000
ICP4	1	0	0	1	0	1	FA000000
ICP5	1	0	0	1	1	0	FC000000
ICP6	1	0	0	1	1	1	FE000000
ICP7	1	0	0	0	0	0	F0000000
ICP8	0	1	0	0	0	0	FE000000
ICP9	0	1	0	0	0	1	FE000200
ICP10	0	1	0	0	1	0	FE000400
ICP11	0	1	0	0	1	1	FE000600
ICP12	0	1	0	1	0	0	FE000800
ICP13	0	1	0	1	0	1	FE000A00
ICP14	0	1	0	1	1	0	FE000C00
ICP15	0	1	0	1	1	1	FE000E00
ICP16	0	1	1	0	0	0	FE001000
ICP17	0	1	1	0	0	1	FE001200
ICP18	0	1	1	0	1	0	FE001400
ICP19	0	1	1	0	1	1	FE001600
ICP20	0	1	1	1	0	0	FE001800
ICP21	0	1	1	1	0	1	FE001A00
ICP22	0	1	1	1	1	0	FE001C00
ICP23	0	1	1	1	1	1	FE001E00
ICP*x*	1	1	*x*	*x*	*x*	*x*	Use address, etc. as defined in EE memory and disregard board select.

[a] Note: 0 is off, and 1 is on.

Step 5:

Insert the ICP so that the component side is facing the same direction as the other boards in the backplane. If no other boards are in the host' s backplane, refer to your system documentation for the correct orientation.

Caution

The ICP will be severely damaged if inserted into the backplane incorrectly.

Step 6:

To access PTBUG, connect a terminal to the ICP's console port connector. (See Figure 3–1 on page 37 for the location of the console port.) The terminal should be set to 9600 baud, 8 bits, no parity, one stop bit. For the ICP6000(X) (6U form factor), a console cable is shipped with the board. For the ICP9000(X) (9U form factor), the console cable connects the console port on the ICP board to the corresponding port on the 9U carrier. You will need to provide a null modem (crossover) cable to connect terminal to the console port.

Step 7:

Turn on the power to your host system. Type "Control-C" at the terminal to cause the **PTBUG >** prompt to appear on the terminal screen.

Program or verify the EEPROM using the PTBUG "EE" command. The following example verifies that address 0x2E is set to 0xFE00 and sets address 0x7C to 0x8000. Press carriage return to accept the current value; press period to return to the prompt. Because the EEPROM is organized as 64 16-bit words, this command operates in 16-bit word operands. For more information, see the *PTBUG Debug and Utility Program Reference Manual*.

Syntax: PTBUG > ee address

Example:
```
PTBUG > ee 2e
0000002E FE00 ?
00000030 0000 ? .
PTBUG > ee 7c
0000007C 00B7 ? 8000
00000080 C6A5 ? .
PTBUG >
```

Use this command to program each of the configurable values shown in Table 3–2 into the EEPROM. When programming the configurable parameters (at addresses 0x1C and 0x2E through 0x36), substitute the configurable parameters you selected in Chapter 2 for those shown in the table.

Note

During power-up/reset diagnostics, one of the tests calculates a checksum of the ICP' s PROM and compares it to a checksum stored at EEPROM address 0x7C, generating an error if the values do not match. As shown in Table 3–2, EEPROM address 0x7C must initially be programmed to the value 0x8000. This instructs the diagnostic (one time only) to calculate the PROM checksum and store it at this location (overwriting the value 0x8000), rather than do the comparison. Therefore, if you review the contents of EEPROM after you have executed the diagnostics one or more times, address 0x7C will no longer equal 0x8000. This is normal.

Table 3–2: EEPROM Default Values

Address (Hex)		Description	Value (Hex)	[a]
0		Validity word	CAFE	
2		Validity word	F00D	
4–18		Reserved		
1A		Validity word	B0BB	
1C	[b]	Boot flags[c]	6 or E	
1E		Startup code EEPROM address high word	0	
20		Startup code EEPROM address low word	C000	
22		Address to move startup code to in RAM high word	400E	
24		Address to move startup code to in RAM low word	0	
26		Number of bytes of startup code to move high word	0	
28		Number of bytes of startup code to move low word	600	
2A		Startup code execution address high word	400E	
2C		Startup code execution address low word	020A	
2E	[b]	Slave address compare high word	0000 or FE00	[d]
30	[b]	Slave address compare low word	F600 or 0200	[d]
32	[b]	Mode selection	34 or 4	
34	[b]	Slave address modifier compare	20	
36	[b]	Slave address modifier don't care	F	
38		Master enable	60	
3A–7E		Reserved		

[a] PTBUG is used to verify these values.

[b] Entries shown with a "b" are configurable options.

[c] See Figure 2–1 on page 30 for further information about the boot flags.

[d] This value is determined by the board select switches shown in Figure 3–1 on page 37.

Step 8:

When EEPROM programming is complete, enter "BO" <CR> to re-enter the download routine.

Step 9:

Attach the cables to the upper and lower ports. Install the loopback cable from Port 0 to Port 1 in preparation for verification testing. Refer to Figure 3–2.

Step 10:

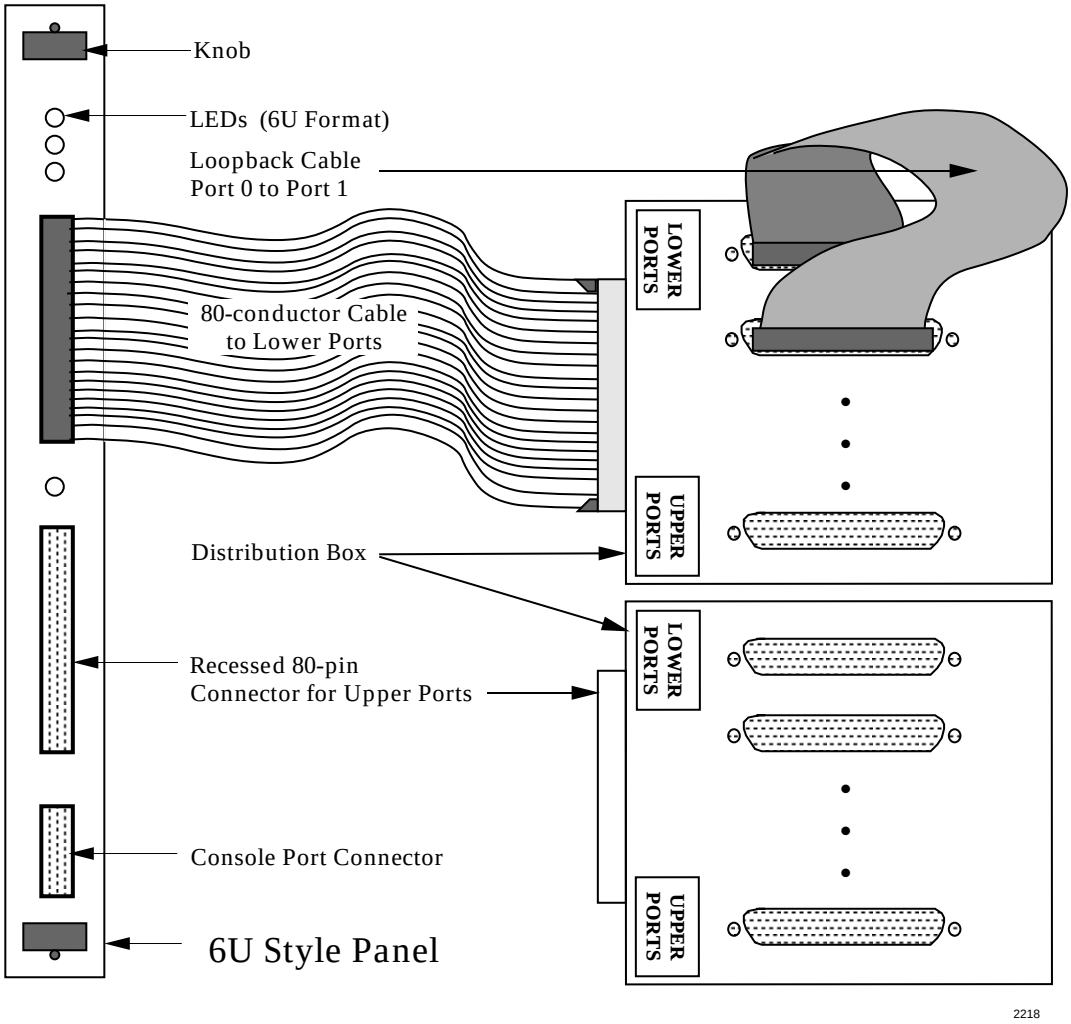
If you previously performed the software installation to configure the new kernel or boot operating system for Simpact' s hardware, power on the system to boot the new kernel or operating system. Otherwise, boot the existing kernel and refer to the *ICP6000/ICP9000 Software Installation Guide for UNIX Products* to perform the software installation.

Step 11:

After you boot the new kernel or operating system, verify the hardware and software installation by running the verification test. Refer to the *ICP6000/ICP9000 Software Installation Guide for UNIX Products*.

Note

Some operating systems support loadable drivers rather than requiring the build of a new image.



Note: Connections for the 9U panel are similar.

Figure 3–2: ICP6000(X) Labeling and Loopback Connections

Chapter

4 Hardware Specifications

The ICP6000(X)/ICP9000(X) consists of a base board that is a VMEbus controller, an Electrical Interface Module (EIM) daughterboard, and one or more passive distribution panels. The base board contains the 680*x*0, memory, VMEbus interfaces, and DMA controller. The EIM contains the communications and line interface devices. The distribution panel provides the required transition from the ICP to the connector interface.

4.1 Mechanical and Environmental Specifications

The following sections define the requirements for the ICP.

4.1.1 Physical Characteristics

	ICP6000(X)		ICP9000(X)	
Width	234 mm	(9.2 in)	368 mm	(14.5 in)
Depth	160 mm	(6.3 in)	403 mm	(15.9 in)
Front Panel	20.3 mm	(0.8 in)	20.3 mm	(0.8 in)

4.1.2 Power Requirements

+5Vdc +/-5%	5.0 amps (typical)
	6.0 amps (maximum)
+12Vdc +/-5%	0.15 amps (maximum)
-12Vdc +/-5%	0.15 amps (maximum)

The host power supply *must* meet these requirements. Be sure to consider the needs of all devices that are serviced by the power supply. Consider both total current for each voltage source and total power.

4.1.3 Environment

Several environmental factors should be considered to ensure reliable operation of the ICP and its host system.

The room air conditioning system should provide cool, filtered, humidified air. Temperature should be held as stable as possible to prevent thermal-related failures. Low humidity contributes to static electricity, which can cause catastrophic failures. The allowable temperature and humidity ranges are as follows:

Temperature	0° to 55° C, operating
	-55° to 85° C, non-operating
Humidity	10% to 90%, non-condensing

Recommended operating conditions are 20 degrees Centigrade (70 degrees Fahrenheit) and 45 percent relative humidity.

4.2 Device Specifications

Table 4–1 lists the specifications for the devices on the base board.

Table 4–1: Specifications for ICP Devices

Device	Specification	
Processor	68020 at 16 MHz or 68030 at 30 MHz	
Main RAM	1 or 4 Megabytes Dynamic	
PROM/EEPROM/SRAM	Two 32-pin JEDEC Byte Ports	
	Socket 1	32Kx8 PROM or
		64Kx8 PROM or
		128Kx8 PROM or
		256Kx8 PROM or
		512Kx8 PROM or
		1024Kx8 PROM
	Socket 2	8Kx8 EEPROM or
		8Kx8 SRAM or
		32Kx8 EEPROM or
		32Kx8 SRAM or
		64Kx8 PROM or
		128Kx8 PROM or
		256Kx8 PROM or
		512Kx8 PROM or
		1024Kx8 PROM
VMEbus Master	A32/A24: D32/D16	
	Block Mode	
	Unaligned Transfers	
VMEbus Slave	A32/A24/A16 D8(0)	

Chapter

5 Hardware Overview

This chapter describes the ICP6000(X)/ICP9000(X) architecture, memory map, buses, and design. Figure 5–1 shows a block diagram of the board, Table 5–1 shows the allocation of memory space, and Table 5–2 shows the interrupt levels, sources, and vectors. All circuitry, pinout, and distribution panel information that applies to specific electrical interfaces, such as EIA-232, EIA-422, and so on is found in Appendix A through Appendix D.

5.1 Internal Architecture

The ICP VMEbus controller allows accesses to the VMEbus that are of indeterminate length and take place without compromising the ability of the I/O DMA controller to transfer data.

Many VMEbus setup parameters usually associated with jumpers or switches are software programmable on the ICP. VMEbus slave address, master request level, and master request mode are programmable by the local processor. An EEPROM is provided for non-volatile storage of these parameters. The board select switches, accessible to the firmware, are provided for those applications that demand a manual input (see Section 5.14).

5.2 Microprocessor

The ICP6000/ICP9000 uses a Motorola 68020 microprocessor running at 16 MHz. The ICP6000X/ICP9000X uses a Motorola 68030 microprocessor running at 30 MHz.

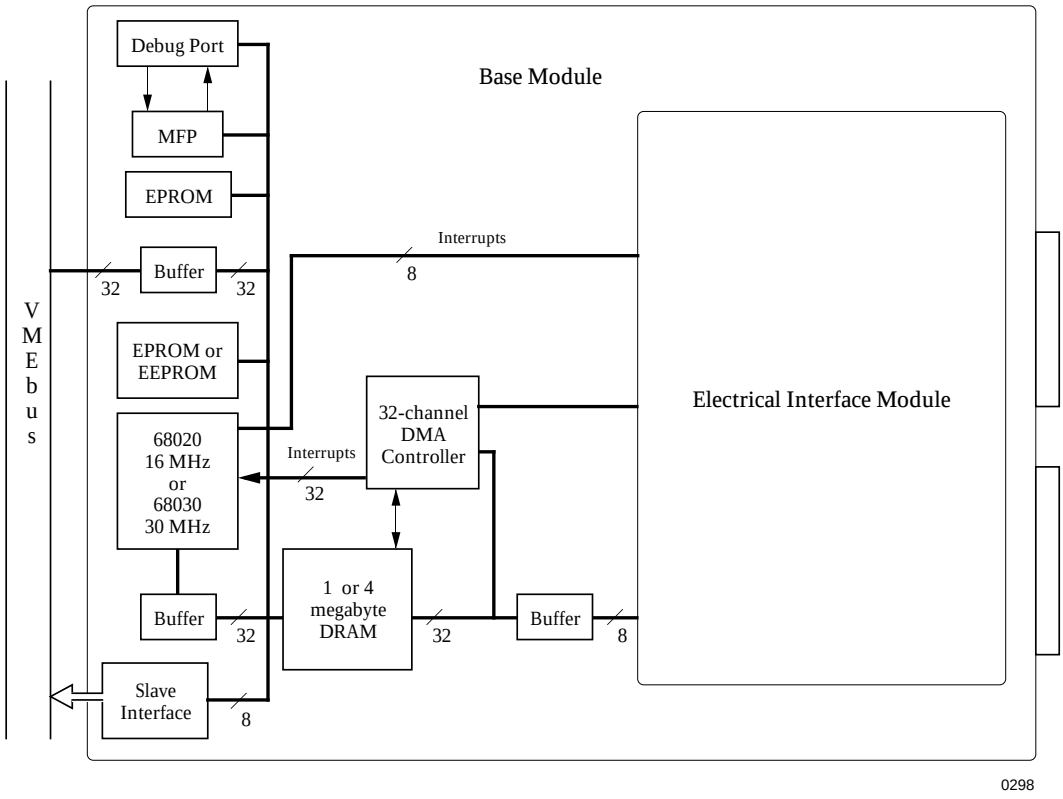


Figure 5–1: ICP6000(X)/ICP9000(X) Block Diagram

Table 5–1: General Memory Map

Memory Range (hex)	Device
00000000–0001FFFF or 00000000–0003FFFF	Read-only Memory Socket, U39 or U7 (jumper-selectable, K3)
00020000–0FFFFFFF or 00040000–0FFFFFFF	Read/Write Memory Socket, U19 or U8 (jumper-selectable, K3)
10000000	I/O DMA Command Register (write-only, byte-wide)
10000001	General Control Register 0 (write-only, byte-wide)
10000002	General Control Register 1 (write-only, byte-wide)
10000003	General Status Register (read-only, byte-wide)
20000000	MFP General Purpose I/O Data Register (GPDR)
20000001	MFP Active Edge Register (AER)
20000002	MFP Data Direction Register (DDR)
20000003	MFP Interrupt Enable Register A (IERA)
20000004	MFP Interrupt Enable Register B (IERB)
20000005	MFP Interrupt Pending Register A (IPRA)
20000006	MFP Interrupt Pending Register B (IPRB)
20000007	MFP Interrupt In-Service Register A (ISRA)
20000008	MFP Interrupt In-Service Register B (ISRB)
20000009	MFP Interrupt Mask Register A (IMRA)
2000000A	MFP Interrupt Mask Register B (IMRB)
2000000B	MFP Vector Register (VR)
2000000C	MFP Timer A Control Register (TACR)
2000000D	MFP Timer B Control Register (TBCR)
2000000E	MFP Timers C & D Control Register (TCDCR)
2000000F	MFP Timer A Data Register (TADR)
20000010	MFP Timer B Data Register (TBDR)
20000011	MFP Timer C Data Register (TCDR)
20000012	MFP Timer D Data Register (TDDR)
20000013	MFP Sync Character Register (SCR)

Table 5–1: General Memory Map (*Cont'd*)

Memory Range (hex)	Device
20000014	MFP USART Control Register (UCR)
20000015	MFP Receiver Status Register (RSR)
20000016	MFP Transmitter Status Register (TSR)
20000017	MFP USART Data Register (UDR)
30000000	VSI Mailbox Register 0
30000001	VSI Mailbox Register 1
30000002	VSI Mailbox Register 2
30000003	VSI Mailbox Register 3
30000004	VSI Mailbox Register 4
30000005	VSI Mailbox Register 5
30000006	VSI Mailbox Register 6
30000007	VSI Mailbox Register 7
30000008	VSI Mailbox Register 8
30000009	VSI Mailbox Register 9
3000000A	VSI Mailbox Register 10
3000000B	VSI Mailbox Register 11
3000000C	VSI Mailbox Register 12
3000000D	VSI Mailbox Register 13
3000000E	VSI Mailbox Register 14
3000000F	VSI Mailbox Register 15
30000010	VSI Mailbox 7–0 Interrupt Mask Register (write-only)
30000010	VSI Mailbox 7–0 Interrupt Pending Register (read-only)
30000011	VSI Mailbox 15–8 Interrupt Mask Register (write-only)
30000011	VSI Mailbox 15–8 Interrupt Pending Register (read-only)
30000012	VSI Mailbox 03–00 Interrupt Pin Select
30000013	VSI Mailbox 07–04 Interrupt Pin Select
30000014	VSI Mailbox 11–08 Interrupt Pin Select
30000015	VSI Mailbox 15–11 Interrupt Pin Select
30000016	VSI Slave Address Modifier Compare Register
30000017	VSI Slave Address Modifier Don't Care Register

Table 5–1: General Memory Map (*Cont'd*)

Memory Range (hex)	Device
30000018	VSI Slave Address A31–A24 Compare Register
30000019	VSI Slave Address A23–A16 Compare Register
3000001A	VSI Slave Address A15–A09 Compare Register
3000001B	VSI VMEbus Interrupter Request Level Register (write-only)
3000001B	VSI VMEbus Interrupter Level Pending Register (read-only)
3000001C	VSI VMEbus Interrupt Vector Number Register
3000001D	VSI Slave Auxiliary Address Compare/Don't Care
3000001E	VSI Global Interrupt Status Register (read-only)
3000001E	VSI Master Enable Register (write-only)
3000001F	VSI Mode Selection Register
40000000–40*x*FFEFF[a]	Dynamic RAM (read/write, longword-wide)
40*x*FFF00–40*x*FFF3F	Ports 0–15 Receive DMA Memory Address Register (MAR) (read/write, longword-wide)
40*x*FFF40–40*x*FFF7F	Ports 0–15 Transmit DMA MAR (read/write, longword-wide)
40*x*FFF80–40*x*FFFBF	Ports 0–15 Receive DMA Terminal Count Register (TCR) (read/write, longword-wide)
40*x*FFFC0–40*x*FFFFF	Ports 0–15 Transmit DMA TCR (read/write, longword-wide)
60000000	EIM SCC Port 0, Data (read/write, byte-wide)
60000001	EIM SCC Port 0, Control
60000002	EIM SCC Port 1, Data
60000003	EIM SCC Port 1, Control
60000004	EIM SCC Port 2, Data
60000005	EIM SCC Port 2, Control
60000006	EIM SCC Port 3, Data
60000007	EIM SCC Port 3, Control
60000008	EIM SCC Port 4, Data
60000009	EIM SCC Port 4, Control
6000000A	EIM SCC Port 5, Data
6000000B	EIM SCC Port 5, Control
6000000C	EIM SCC Port 6, Data

Table 5–1: General Memory Map (*Cont'd*)

Memory Range (hex)	Device
6000000D	EIM SCC Port 6, Control
6000000E	EIM SCC Port 7, Data
6000000F	EIM SCC Port 7, Control
60000010	EIM SCC Port 8, Data
60000011	EIM SCC Port 8, Control
60000012	EIM SCC Port 9, Data
60000013	EIM SCC Port 9, Control
60000014	EIM SCC Port 10, Data
60000015	EIM SCC Port 10, Control
60000016	EIM SCC Port 11, Data
60000017	EIM SCC Port 11, Control
60000018	EIM SCC Port 12, Data
60000019	EIM SCC Port 12, Control
6000001A	EIM SCC Port 13, Data
6000001B	EIM SCC Port 13, Control
6000001C	EIM SCC Port 14, Data
6000001D	EIM SCC Port 14, Control
6000001E	EIM SCC Port 15, Data
6000001F	EIM SCC Port 15, Control
60000020–6000003F	EIM Bit Registers
80000000–FFFFFFFF	VMEbus Access

[a] *x* = 0 for ICP6000/ICP9000 or 3 for ICP6000X/ICP9000X

Table 5–2: General Interrupt Map

Level	Source	Vector (hex)
7	NMI (Toolkit Panel)	7C
6	DMA Channel Terminal Count	EO–EF (receive)
		F0–FF (transmit)
5	SCC (via MFP)	x0–xF (programmable)
4	Software Timer	1C (Auto Vector 4)
3	VSI Interrupt 0 (LIRQ0)	1B (Auto Vector 3)
2	VSI Interrupt 0 (LIRQ1)	1A (Auto Vector 2)
1	VSI Interrupt 0 (LIRQ2)	19 (Auto Vector 1)

5.3 Main Memory

Main memory is one or four megabytes of dynamic RAM.

5.4 PROM

The board is equipped with two 32-pin JEDEC sockets configured as byte ports. One socket is used with read-only devices. The second socket can be configured for read-only or read/write devices. As supplied by Simpact, the read-only socket (U39 or U7) contains a 64Kx8 PROM. The diagnostics, boot loader, and PTBUG debugging tool reside in this PROM. The other socket is empty and available for a user-added device. See Section 2.5 on page 23 for the jumper configurations to enable these sockets for the type of memory to be used.

5.5 Timers

The 68901 Multi-Function Peripheral (MFP) provides four timers. The timers perform the following functions:

- Operating system clock
- VMEbus timeout timing
- Console port data rate generation

The timers are driven by a 3.6864 MHz clock.

5.6 Console Port

The ICP's console port is provided by the single-channel USART on the Multi-Function Peripheral. It is equipped with an EIA-232 interface and supports transmit and receive signals over the common asynchronous speed range.

The console port connector is a 14-pin dual-row header located on the ICP's front panel. This connector is designed to allow construction of a simple adapter to the standard 25-pin D EIA-232 connector using insulation displacement connectors and a 14-conductor ribbon cable. Table 5–3 shows the pin assignments.

5.7 Communications Devices

The Electrical Interface Module (EIM) is a daughterboard that contains up to eight Z85C30 SCCs operating at 7.3728 MHz (ICP6000/ICP9000) or Z85230 SCCs operating at 14.7456 MHz (ICP6000X/ICP9000X) and, optionally, bit registers for modem and other control applications. All drivers and receivers are located on the EIM. One or two 80-conductor ribbon cables connect the EIM to the distribution panel. The distribution panel is completely passive and serves only to mount connectors. Specific EIM descriptions can be found in the appendices of this manual.

Table 5–3: Console Port Pin Assignments

EIA-232 25 Dsub Pin	Front Panel (P4) Connector Pin	Description
1	1	Ground
2	3	Transmit Data (driven by ICP)
3	5	Receive Data (received by ICP)
4	7	Jumpered to P4-9
5	9	Jumpered to P4-7
6	11	Jumpered to P4-14
7	13	Ground
-	8 and 10	Reserved for NMI switch
-	12	Reserved for Reset switch
8–19	-	Not connected
20	14	Jumpered to P4-11
21–25	-	Not connected

5.7.1 Register Access

Direct access to the SCC registers can be made by using the addresses listed in the general memory map (see Table 5–1).

5.7.2 SCC Interrupts

The parallel I/O port of the MFP is used to receive interrupt requests from the SCCs residing on the EIM daughterboard. The MFP's Data Direction Register (DDR) should be programmed so that all bits are inputs (see Table 5–4). The Active Edge Register (AER) should be programmed so that all bits are triggered by the falling edge. When an SCC asserts its interrupt line, the MFP causes an interrupt request to the 680*x*0 on level 5. The interrupt vector is based on the SCC requesting the interrupt and the contents of the MFP Vector Register (VR). The Vector Register should not be programmed for a base of 0xE0 or 0xF0 as these vector ranges are reserved for the I/O DMA controller.

Table 5–4: MFP DDR Programming

Bit	I/O	Function
7	I	SCC Interrupt Request Ports 14, 15
6	I	SCC Interrupt Request Ports 12, 13
5	I	SCC Interrupt Request Ports 10, 11
4	I	SCC Interrupt Request Ports 8, 9
3	I	SCC Interrupt Request Ports 6, 7
2	I	SCC Interrupt Request Ports 4, 5
1	I	SCC Interrupt Request Ports 2, 3
0	I	SCC Interrupt Request Ports 0, 1

5.8 Peripheral Device DMA Controller

The SCC devices are served by a 32-channel DMA controller. This controller is configured so that, for each communications port, one channel is allocated to transmit and a separate channel is allocated to receive.

For each channel, the controller stores a Memory Address Register (MAR) value and a Transfer Count (TC) value in a dedicated 64-longword field at the top of onboard main memory. (Refer to the General Memory Map in Table 5–1.)

During operation, a high-speed state machine receives the DMA request and fetches the MAR and Terminal Count Register (TCR) values from memory. The MAR and TCR are incremented or decremented and replaced in memory. The byte read or write cycle, at the address specified by the MAR, is then conducted and the data transferred to or from the SCC.

The DMA controller can interrupt the local processor on terminal count for both transmit and receive operations. An interrupt is generated when a DMA channel reaches terminal count. Each DMA channel has a separate interrupt vector. DMA terminal count interrupts may be disabled for receive channels only.

The I/O DMA controller is programmed using the I/O DMA Command Register and the MARs and TCRs.

After programming the channel's SCC appropriately, the channel buffer address is loaded into the channel's MAR memory location and the transfer (byte) count into the channel's TCR memory location. The channel is then started using a command to the I/O DMA Command Register.

5.8.1 MAR/TCR Memory Locations

Each I/O DMA channel is assigned a dedicated longword near the top of the main memory as its MAR and a second longword as its TCR (see Table 5–1). Each longword has 20 or 22 valid bits, allowing an I/O DMA address range of 1 or 4 megabytes.

The MAR counts up and the TCR counts down as the DMA operation progresses.

5.8.2 DMA Terminal Count Interrupts

Typical operation of SCCs in DMA mode is to have transmit completions signaled by DMA Terminal Count Interrupts and receive completions signaled by SCC Receive Complete Interrupts. In this case, the TCR value of receive channels is set to a value larger than the largest anticipated receive packet, and the TCR acts as a safety net should a rogue packet exceed anticipated size.

Transmit Channel Terminal Count Interrupts are generated by the I/O DMA Controller at level 6. The vector number is determined by combining the number of the channel interrupting with a fixed base vector. Vector numbers are in the form F*x* (hex), where *x* is the port number. (For example, the transmit vector number for port 5 is 0xF5.)

Receive Channel Terminal Count Interrupts can be globally enabled or disabled using bit 4 in Control Register 0 as shown in Figure 5–2. These interrupts are also at level 6 with vectors of the form E*x* (hex). (For example, the receive vector number for port 11 is 0xEB.)

Address: 10000001

7	6	5	4	3	2	1	0
VMERMS	VMEBR1	VMEBR0	RCINT	EEPDI	EEPSK	EEPCS	FAULT

Mnemonic	Name	Description
RCINT	DMA Receive Interrupt	0 = Interruption on receive terminal count 1 = No interruption on receive terminal count The post-reset state of this bit is 0.

Figure 5–2: Control Register 0 Use for DMA

5.8.3 I/O DMA Command Register

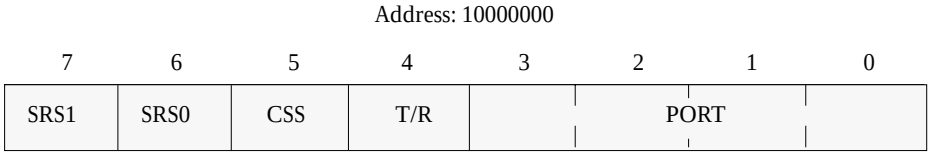
The I/O DMA Command Register, defined in Figure 5–3, is used to start or stop any one of the 32 DMA channels. A bit in the Status Register, shown in Figure 5–4, is used to control write access to this register. Writes to this register must be made only when bit 4 of the Status Register is 1 (DMARDY).

5.9 VMEbus Slave Interface

The VMEbus slave interface is implemented using the “VMEbus Slave Interface” (VSI) ASIC.

The VSI’s 16 byte-wide mailboxes are accessible from both the VMEbus and the local processor bus. The mailbox control logic monitors accesses from the VMEbus and has the capability of interrupting the local processor when VMEbus accesses are made. A flexible system of controls allows this monitor/interrupt function to be adapted to specific firmware requirements.

The VSI also provides the VMEbus interrupter function. This interrupter is fully programmable. Both request level and vector number can be selected by software.



Mnemonic	Name	Description
SRS	Scanner Range Select	These bits select one of four request scanner ranges for the peripheral DMA controller. Maximum peripheral DMA performance is achieved when the scanner range covers only the active channels. 0 0 = Scan ports 0–15 0 1 = Scan ports 0–7 1 0 = Scan ports 0–3 1 1 = Scan ports 0–1
CSS	Channel Start/Stop	This bit commands the channel addressed by T/R and PORT3–0. 0 = Start Channel DMA 1 = Stop Channel DMA
T/R	Transmit or Receive Channel	This bit selects either the transmit or receive channel for the port selected by PORT3–0. 0 = Receive Channel Selected 1 = Transmit Channel Selected
PORT	Port Select Bits	This nibble may contain a value between 0 and 0xF and, along with the T/R bit, selects the DMA channel to be started or stopped.

Figure 5–3: I/O DMA Command Register

Address: 10000003

7	6	5	4	3	2	1	0
SW6	SW5	EEPDO	DMARDY	SW4	SW3	SW2	SW1

Mnemonic	Name	Description
DMARDY	I/O DMA Controller Ready	0 = Do not write to DMA Command Register 1 = Okay to write to DMA Command Register

Figure 5–4: Status Register Use for DMA

The host may issue a hardware reset of the ICP by accessing a specific location decoded by the VSI. The location must be accessed using a specific data pattern to reduce the chances of inadvertent reset (0xAA to offset 0x3D).

5.9.1 VMEbus Slave Address Decoder

The VMEbus slave address decoder provides address decoding to allow the mailboxes to be addressed from the VMEbus. The PROM-based startup code loads the VSI Slave Address Decoder Registers from values it reads from the EEPROM or the board select switches. (Refer to Table 5–1 for VSI register addresses.)

5.9.2 VMEbus Mailboxes and Software Reset

Using its registers, the VSI device decodes a 512-byte region of the VMEbus address space. The 16 byte-wide mailboxes and the software reset decoder are in this region. VMEbus access to the mailboxes can be made to interrupt the local processor bus. The type of access and the interrupt pin to be used can be programmed. The mailboxes are accessible on the VMEbus at odd addresses (B(0)), beginning at offset 1 from the base of the 512-byte region. The Software Reset Register is accessed on the VMEbus at offset 0x3D from the base with 0xAA. The mailboxes are also accessible by the local processor

bus, but the Software Reset Register is not. (See Table 5–1 for mailbox addresses on the local processor bus.) Because setup of the VSI requires multiple operations, it is not enabled onto the VMEbus until all setup operations are complete.

5.9.3 Mailbox Slave Address Decoder

The following example sets up the mailboxes at a VMEbus address of 0xFE000000 in Extended Supervisor or User space. A similar procedure is performed by the PROM-resident boot loader according to the EEPROM configuration parameters (see Section 2.6.1 on page 28). This is shown for your information only; user application code does not normally program the mailbox slave address decoder.

1. Load bits 31–24 (0xFE) of the desired address into the VSI Slave Address Compare Register (A31–A24)

2. Similarly, load bits 23–16 (0x00) into the VSI Slave Address Compare Register (A23–A16) and bits 15–9 (0x00) into the VSI Slave Address Compare Register (A15–A09).

3. The Extended Supervisor/User Address space requires that Address Modifier Bits 2–0 be decoded “don’t care” and that Address Modifier bits 5–3 be compared to 001. To accomplish this, the VSI Slave Address Modifier Compare Register is loaded with 0x08 and the VSI Slave Address Modifier Don’t Care Register is loaded with 0x07.

5.9.4 Configuring the Mailbox Interrupts

In this example, the mailbox interrupts are set so that LIRQ2 (autovector 1) is activated when mailbox 0 is written by the VMEbus, LIRQ0 (autovector 3) is activated when mailbox 7 is written, and LIRQ1 (autovector 2) is activated when mailbox 8 is written.

1. Load a value of 0x02 into Mailbox Interrupt Pin Select (3–0) (offset 0x12). This routes mailbox 0 to LIRQ2 (autovector 1).

2. Similarly, load a value of 0x00 into Mailbox Interrupt Pin Select (7–4) (offset 0x13) and a value of 0x01 into Mailbox Interrupt Pin Select (11–8) (offset 14).

3. Enable mailbox 0, 7, and 8 interrupts by loading a value of 0x81 into Mailbox Interrupt Mask (7–0) (offset 0x10) and a value of 0x01 into Mailbox Interrupt Mask (15–8) (offset 0x11).

5.9.5 Software Reset

The software-controlled reset location is connected directly to the local reset pin and cannot be disabled except by disabling the VSI VMEbus address decode. A reset is generated when a value of 0xAA is written to the Software Reset Register by the VMEbus.

5.10 VMEbus Interrupter

The VMEbus Interrupter is integrated into the VSI. Two VSI registers are used to control the interrupter. The VMEbus Interrupt Request Level Register specifies the desired interrupt level and also contains "interrupter done" status bits. The VMEbus Interrupt Request Vector Number Register specifies the desired interrupt vector and initiates the interrupt. The VMEbus interrupter is normally programmed by system-level downloaded code provided by Simpact. The programming sequence is summarized below. Details are available in the *VMEbus Slave Interface (VSI)* manual.

1. Poll the VMEbus Interrupt Request Level Register until status indicates "not busy."

2. Load the desired request level into the VMEbus Interrupt Request Level Register.

3. Load the desired vector number into the VMEbus Interrupt Vector Number Register. This load initiates the interrupt process.

4. The interrupter goes "busy" (as indicated by the Interrupt Request Level Register) and remains "busy" until the VMEbus interrupt acknowledge cycle is completed.

Optionally, one of the LIRQ pins can be configured as an “interrupter done” interrupt.

5.11 VMEbus Master Interface

The ICP allows the local processor to have direct access to the VMEbus through the VMEbus Master Interface.

VMEbus access appears as a portion of the local processor memory map and can be freely read from or written to using full A32:D32 capability. During VMEbus accesses, AM0–5 and A31 are supplied from a local register. All byte position translations and unaligned transfers are automatically performed.

The VMEbus request level and request mode (Release-When-Done and Release-On-Request) are software-programmable.

Because VMEbus accesses may be of indeterminate length (due to bus contention from unknown sources), the ICP provides a level of logical isolation between the local memory array and the VMEbus master interface. This isolation ensures that the real-time requirements of the SCCs’ DMA controller are not jeopardized by long processor accesses to the VMEbus.

A VMEbus Master requests the VMEbus with one of five priority levels and using one of two modes. Bits for controlling the requester priority level and mode are found in Control Register 0, as shown in Figure 5–5.

A VMEbus Master also issues Address Modifier codes with each address. The ICP Master Interface obtains these codes from bits in Control Register 1, as shown in Figure 5–6.

A VMEbus Master must adapt to limitations and options in the system of which it is a part. Control bits to limit transfers to 16 bits and to perform internal data alignment are provided in Control Register 1. This allows operation with systems that do not support D32 and/or non-aligned transfers.

Address: 10000001

7	6	5	4	3	2	1	0
VMERMS	VMEBR1	VMEBR0	RCINT	EEPDI	EEPSK	EEPCS	FAULT

Mnemonic	Name	Description
VMERMS	VMEbus Request Mode Select	0 = Release On Request 1 = Release When Done
VMEBR1–0	VMEbus Bus Request Level	These bits select the corresponding bus request level (0–3) after the next VMEbus cycle arbitration completes.

Figure 5–5: Control Register 0 Use for VMEbus

Bit 31 of the local address bus is used to decode VMEbus accesses and is not available for direct use on the VMEbus. In order to provide access to the full 32-bit address range of the VMEbus, the VMEbus Master Interface uses bit 31 supplied from Control Register 1.

The VMEbus Master may access non-existent or non-responsive VMEbus addresses, particularly during software development. To prevent a hang condition, the ICP is equipped with a VMEbus cycle timer that is programmed by the PROM-resident code.

Address: 10000002

7	6	5	4	3	2	1	0
VMEDA	VMEDS	AM5	AM4	A31	AM2	AM1	A0

Mnemonic	Name	Description
VMEDA	VMEbus Data Alignment	This bit controls the data alignment mode of the ICP. When the bit is 1, the internal VMEbus master logic aligns non-aligned data. When the bit is 0, non-aligned transfers are performed on the VMEbus. 0 = Non-aligned transfers performed 1 = Alignment performed internally
VMEDS	VMEbus Data Size	This bit controls the maximum data size used by the ICP for VMEbus transfers 0 = D:32 master transfers allowed 1 = D:16 master transfers only
AM5–0	VMEbus Address Modifier Codes	These bits provide VMEbus address modifier bits 5–0, respectively, during VMEbus read and write cycles. AM3 is not programmable as it is derived directly from AM5 and AM4.
A31	VMEbus Address Bit 31	This bit provides A31 during VMEbus master read and write cycles.

Figure 5–6: Control Register 1

5.12 Operating Controls and Indicators

This section describes the LEDs on the board. Figure 5–7 shows their locations on the ICP.

5.12.1 Fault Indicator

The ICP FAULT indicator is a red LED, controlled by a bit in Control Register 0 as shown in Figure 5–8.

5.12.2 I/O DMA Indicator

The green “I/O DMA” indicator is illuminated whenever the local DMA controller has control of the internal bus. The LED illumination intensity is a relative indication of the DMA activity.

5.12.3 VME Access Indicator

The green “VMEbus” indicator is illuminated whenever the ICP is accessing the VMEbus.

5.13 Serial EEPROM Operation

The serial EEPROM stores setup parameters and is reserved for use by the PROM-based code. Accessing the serial EEPROM is a software-intensive operation because the control lines (Chip Select, Clock, Data In, and Data Out) must be manipulated on a bit-by-bit basis to clock data into or out of the EEPROM. Consult the EEPROM manufacturers’ documentation regarding the required timing and control.

The output lines (Chip Select, Clock, and Data In) are controlled from bits in Control Register 0 (see Figure 5–9). The input line (Data Out) is accessed from the Status Register (see Figure 5–10).

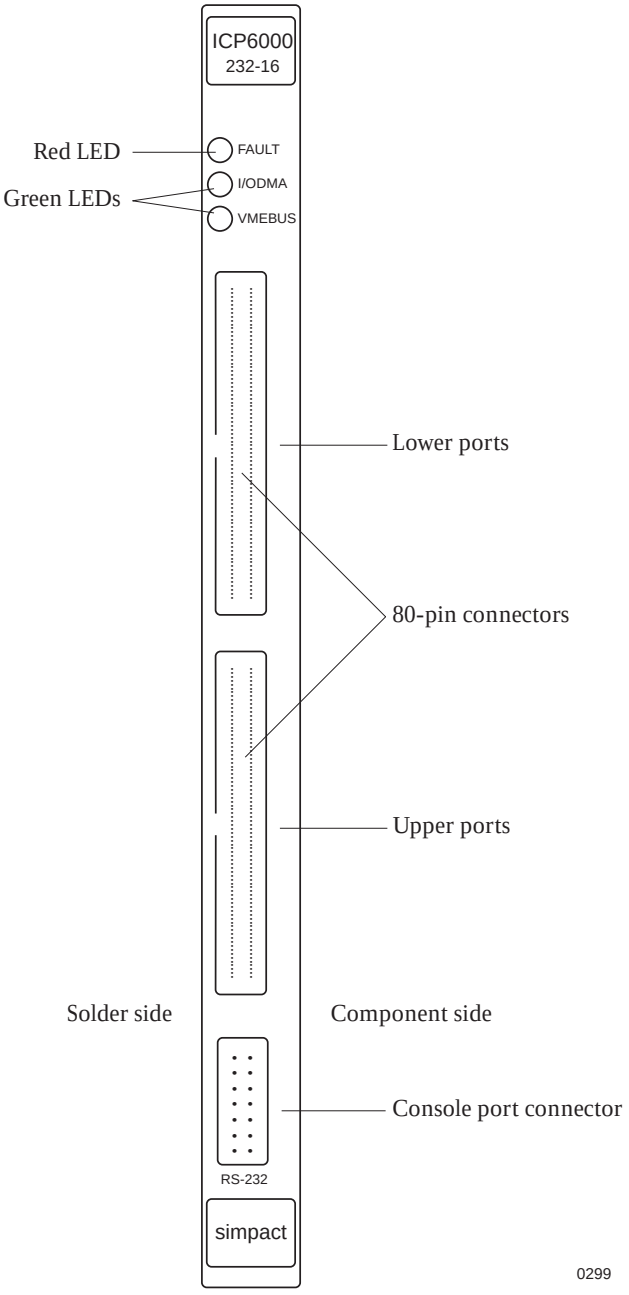


Figure 5–7: ICP Front Panel Showing Location of LEDs

Address: 10000001

7	6	5	4	3	2	1	0
VMERMS	VMEBR1	VMEBR0	RCINT	EEPDI	EEPSK	EEPCS	FAULT

Mnemonic	Name	Description
FAULT	FAULT LED and SYSFAIL* Control	0 = FAULT LED and SYSFAIL* asserted 1 = FAULT LED and SYSFAIL* negated The post-reset state of this bit is 0 (LED on and SYSFAIL* asserted).

Figure 5–8: Control Register 0 Use for FAULT LED

Address: 10000001

7	6	5	4	3	2	1	0
VMERMS	VMEBR1	VMEBR0	RCINT	EEPDI	EEPSK	EEPCS	FAULT

Mnemonic	Name	Description
EEPDI	EEPROM Data In Line	This bit selects the “Data In” line to the serial EEPROM.
EEPSK	EEPROM Serial Clock Line	This bit selects the clock to the serial EEPROM.
EEPCS	EEPROM Chip Select	0 = EEPROM not selected 1 = EEPROM selected The post-reset state of this bit is 0.

Figure 5–9: Control Register 0 Use for EEPROM Operation

Address: 10000003

7	6	5	4	3	2	1	0
SW6	SW5	EEPDO	DMARDY	SW4	SW3	SW2	SW1

Mnemonic	Name	Description
EEPDO	EEPROM Data Out Line	This bit selects the “Data Out” line from the serial EEPROM.

Figure 5–10: Status Register Use for EEPROM Operation

5.14 Board Select Switches

The board select switches are located on the base board, near the end of the electrical interface module (the daughterboard). There are six switches, SW1 through SW6. Refer to Section 2.5 on page 23 for information on setting the board select switches. The switch positions can be read from the Status Register as shown in Figure 5–11.

Address: 10000003

7	6	5	4	3	2	1	0
SW6	SW5	EEPDO	DMARDY	SW4	SW3	SW2	SW1

Mnemonic	Name	Description
SW6–SW1	Board Select Switches	0 = Switch on toward front panel 1 = Switch off toward VME backplane
	SW6–SW5	0 0 = Go to PTBUG 0 1 = Use extended space slave addressing 1 0 = Use short space slave addressing 1 1 = Ignore SW3:1 and use address values currently in EE
	SW4–SW1	Binary encoded board number

Figure 5–11: Status Register Use for Board Select Switches

Chapter 6 Diagnostics

The ICP PROM contains a comprehensive hardware self-test that is normally executed when the ICP is powered-on or reset.

During execution of the self-test, the FAULT LED (shown in Figure 5–8 on page 70) is illuminated. On successful completion of the test and subsequent loading of the VSI address compare registers, etc., the LED is extinguished. An error detected by the self-test terminates the test and the FAULT LED remains illuminated.

6.1 Tests Performed

The hardware self-test is composed of the subtests described in the following sections.

6.1.1 ROM Checksum

The ROM checksum subtest sums the bytes of the onboard PROM. This value is then compared to the precalculated checksum located in “ROM checksum word” at address 0x7C of the EEPROM (see Table 3–2 on page 41).

6.1.2 Dynamic RAM (DRAM)

The DRAM subtest performs a series of pattern comparisons on the ICP’s one or four megabytes of DRAM. The patterns are:

- Alternating bits
- Checkerboard with lookahead

- Unique address check
- All ones with zeros lookahead
- All zeros with lookahead using read-modify-write (RMW) cycles

6.1.3 68901 Multi-function Peripheral (MFP)

Subtests performed by the MFP are:

- Write/read/compare of the MFP local registers
- Countdown by timers A and B
- Interrupt generation by timers A and B
- Interrupt generation by the software timer

6.1.4 VMEbus Slave Interface (VSI)

Subtests performed on the VSI are:

- Mailbox registers write/read/compare
- Mailbox unique address check
- Mailbox registers RMW cycle check

6.1.5 EEPROM

The EEPROM is tested by performing ten read cycles of locations zero and two of the EEPROM. In order for the test to pass, the values read must be 0xCAFE and 0xF00D, respectively.

6.1.6 DMA Controller

The DMA controller is verified by the following subtests:

- DMA controller command check
- DMA controller interrupt check
- Transfer count registers check

Chapter

7 Boot Load Procedure

The ICP PROM contains a boot loading program that downloads code and data from the host system to the ICP's RAM. This chapter describes the interface between the boot loader and the host system.

7.1 Download Message Format

The boot loader, which resides in PROM on the ICP, communicates with the host by defining the ICP's 16 byte-wide mailboxes as a protocol exchange region (PXR). The PXR occupies the first twelve mailboxes, and is defined in Table 7–1.

Table 7–1: Protocol Exchange Region

Mailbox Number(s)	Field Name	Description
0	`h_cmd`	Host command register
1	`i_cmd`	ICP command register
2–3	`h_bytes`	Byte count
4–7	`h_iaddr`	Load or execution address
8–11	`h_haddr`	Host buffer address

The ICP writes function codes to the `i_cmd` field, and the host writes function codes to the `h_cmd` field. The host must clear the `i_cmd` field after reading a value from it, indicating to the ICP that it may write another value to the field. Likewise, the ICP clears the `h_cmd` field after it reads a value written by the host. The host must not write another value to the `h_cmd` field until it has been cleared by the ICP.

The following function codes are written to the h_cmd field by the host processor:

Function	Value (hex)
Download ready	10
Write block	04
Init procedure	08

The following function codes are written to the i_cmd field by the ICP:

Function	Value (hex)
Download request	80
Acknowledge (ACK)	04
Negative acknowledge (NAK)	10

The remaining fields of the PXR are always written by the host and read by the ICP. For a particular function, these fields should be filled in before writing the function code to the h_cmd field. The h_bytes field is a 16-bit value with the high-order byte in mailbox 2 and the low-order byte in mailbox 3. The h_iaddr and h_haddr fields are each 32 bits, with the high-order byte in the lowest-numbered mailbox (4 or 8) and the low-order byte in the highest-numbered mailbox (7 or 11).

7.2 Download Procedure

On power-on or reset, the self-test diagnostics execute and, on completion, pass control to the boot loader. The boot loader (which is not interrupt driven) initializes and clears the mailboxes, then begins polling the h_cmd field of the PXR, waiting for a download ready function code to be stored in the field by the host. When download ready is received, the boot loader acknowledges the host by storing an ACK function code in the i_cmd field.

Before beginning the download, the host should reset the ICP to ensure that the system is in a known state and that the boot loader is running. To reset the ICP, write the value 0xAA to the software reset register at offset 0x3D in the 512-byte region of VMEbus

address space decoded by the VSI (see Section 5.9 on page 60). After resetting the ICP, the host driver must wait for the mailboxes to appear on the VMEbus. This must be done in such a way that the host system does not crash with a bus error. For example, SunOS provides a subroutine called `peekc` which takes an address as input and returns an error code if the address is invalid. Simpact's SunOS host driver calls `peekc` periodically with the address of the first mailbox until no error is returned.

Next, the host must issue a `download ready` command to provide the ICP with a set of parameters to initialize the VMEbus master interface and interrupter. These parameters must be stored in the PXR before issuing the command. The fields of the PXR have a special definition used for this command only, as shown in Table 7–2. Refer to Section 2.6.2 on page 32 for a description of appropriate values for these fields. After setting the parameters, the host writes the `download ready` command code to the `h_cmd` field, and waits for an acknowledgment from the ICP (an `ACK` in the `i_cmd` field).

Table 7–2: Protocol Exchange Region for Download Ready Command

Mailbox Number	Field Name	Description
0	`h_cmd`	Host command register
1	`i_cmd`	ICP command register
2	`h_vec`	Vector
3	`h_lev`	Level
4	`h_mode`	Mode
5	`h_amod`	Address modifier

When the host has read the `ACK` function code from the `i_cmd` field, the boot loader writes a `download request` function code to the `i_cmd` field.

On receipt of the `download request`, the host stores the address of a block of code/data in the `h_haddr` field of the PXR, stores the onboard transfer address for the block in the `h_iaddr` field, and stores the byte count in the `h_count` field. The host then stores a write block function code in the `h_cmd` field.

On receipt of a `write block` command, the boot loader validates the load address specified by the host in the PXR and, if it is a valid RAM address, transfers the block of data from the specified host address to the load address and stores an `ACK` function code in the `i_cmd` field to acknowledge completion of the operation. The boot loader stores a `NAK` function code in the `i_cmd` field if the load address is invalid. When the host has read the `ACK` or `NAK`, the boot loader stores another `download request` function code in the `i_cmd` field.

On receipt of the `download request`, the host prepares the next block of code/data and issues another `write block` command. This process continues until the final block has been transferred to the ICP.

When it receives the next `download request` from the boot loader, the host uses the `init procedure` function code to begin execution of the downloaded code. For this function, the host must write the onboard execution address to the `h_iaddr` field of the PXR and store an `init procedure` function code in the `h_cmd` field. (The `h_haddr` and `h_count` fields are not used.)

On receipt of an `init procedure` command, the boot loader validates the execution address that the host has stored in the PXR and, if it is a valid RAM address, stores an `ACK` function code in the `i_cmd` field. (If the address is invalid, a `NAK` is returned and processing of the command is terminated.) After writing the `ACK`, the boot loader waits for the host to read the `i_cmd` field and then makes a subroutine call to the specified address. In general, this subroutine call initializes the run-time system and never returns to the boot loader. However, the `init procedure` command can be used to execute a subroutine that does, in fact, return on completion. In this case, the boot loader will then send another `download request` to the host, and download operations may continue. Note that the host will receive an `ACK` completing the operation whether or not the subroutine call returns to the boot loader.

Figure 7–1 illustrates a typical download sequence.

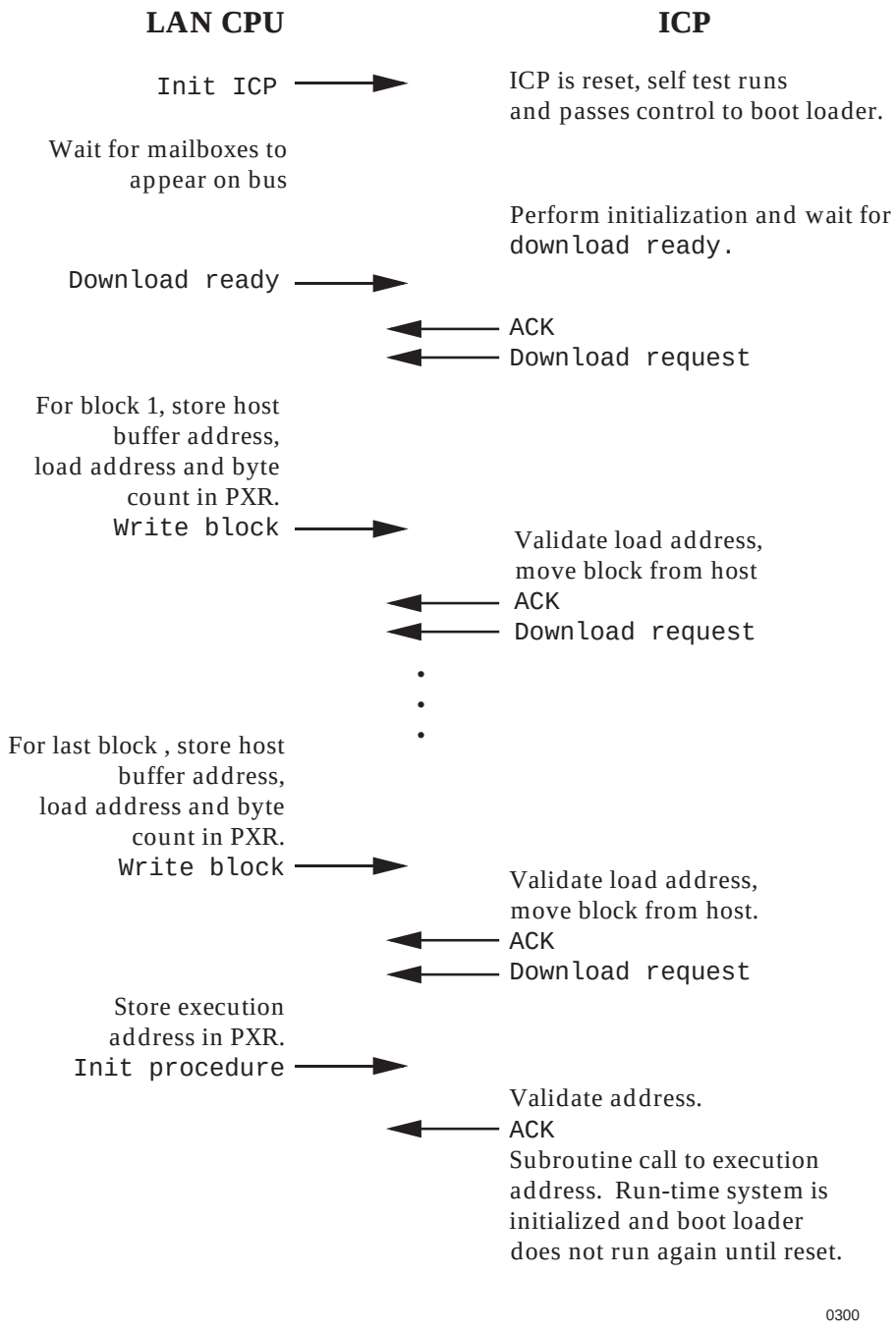


Figure 7–1: Typical Download Sequence

Appendix

A 16-port EIA-232 Electrical Interface Module

This Electrical Interface Module is referred to as EIMD. It is a 16-port daughterboard for the ICP base board and supports 16-port EIA-232 operation.

A.1 Modem Clocks

EIMD supports the full functionality of the serial communications controller (SCC) with respect to external and internal clock sources.

A.1.1 Receive Clock Inputs

The Receive Clock (RxCA and RxCB) pins are always inputs to the SCC. RxC inputs 0–15 are connected to the RxCA (even-numbered ports) and RxCB (odd-numbered ports) inputs of the SCCs.

A.1.2 Transmit Clock Inputs/Outputs

The Transmit Clock (TxCA and TxCB) pins can be programmed as either inputs to or outputs from the SCC. To support this bidirectional interface, EIMD is equipped with both a driver and a receiver for this line.

The TxC0–15 drivers and receivers are selected using the jumper settings shown in Table A–1. Refer also to Figure A–1.

Caution

You must also set the jumper on the distribution panel and use the protocol software to configure the desired clocking.

Table A–1: Clock Jumper Settings for 16-port EIA-232

	Transmit Clock Source[a]	
Port	**Internal (DTE)**	**External (DCE)**
0	K9 1-2	K1 1-2
1	K10 1-2	K2 1-2
2	K11 1-2	K3 1-2
3	K12 1-2	K4 1-2
4	K13 1-2	K5 1-2
5	K14 1-2	K6 1-2
6	K15 1-2	K7 1-2
7	K16 1-2	K8 1-2
8	K25 1-2	K17 1-2
9	K26 1-2	K18 1-2
10	K27 1-2	K19 1-2
11	K28 1-2	K20 1-2
12	K29 1-2	K21 1-2
13	K30 1-2	K22 1-2
14	K31 1-2	K23 1-2
15	K32 1-2	K24 1-2

[a] Only one jumper per port

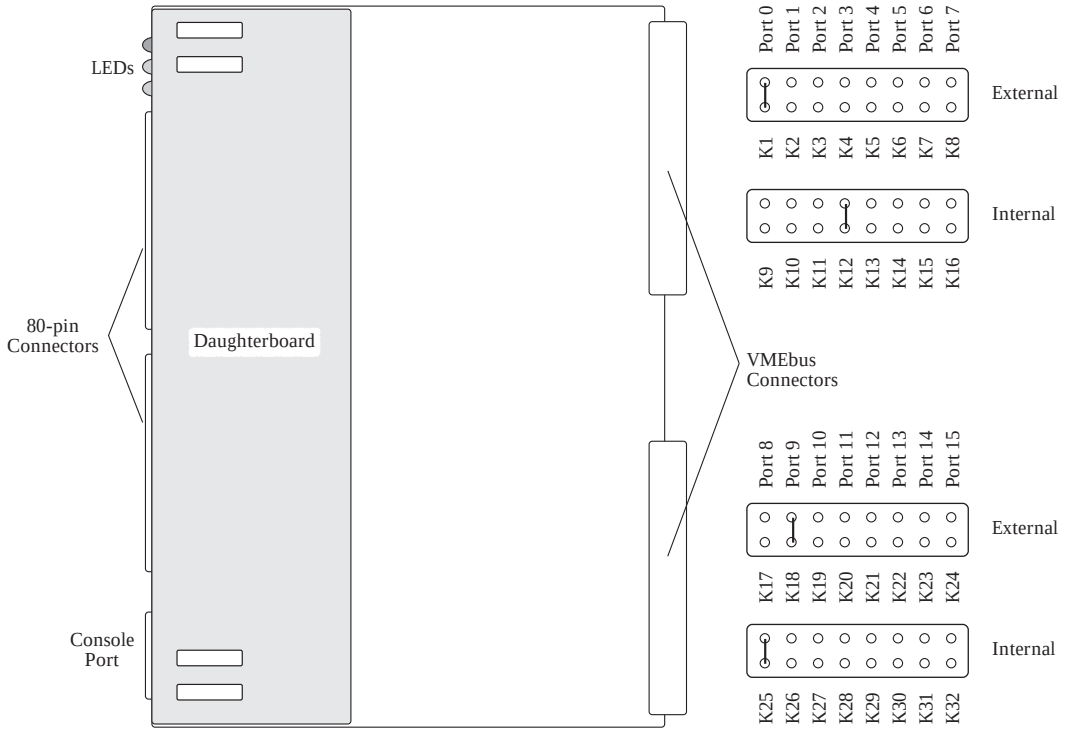


Figure A–1: 16-port EIA-232 Clock Jumper Settings

A.2 Z85C30 or Z85230 Serial Communications Controller

EIMD has eight Z85C30 (ICP6000/ICP9000) or Z85230 (ICP6000X/ICP9000X) serial communications controllers (SCCs). Each SCC implements two communications ports.

A.2.1 SCC Register Access

Access to the SCC data and command registers is made using the addresses shown in Table 5–1 on page 51. Hardware protection is provided so that the SCC "write recovery" limits are automatically met.

A.2.2 SCC Timebase

The SCCs are driven from a 7.3728 MHz (ICP6000/ICP9000) or 14.7456 MHz (ICP6000X/ICP9000X) peripheral clock signal, PCLK.

A.2.3 SCC DMA

Each communications port is assigned two DMA channels, one for transmitting and the other for receiving. Section 5.8 on page 58 describes the setup and operation of the DMA controller.

A.3 Modem Control

EIMD supports five modem control signals: three inputs (Data Set Ready, Clear To Send, and Data Carrier Detect) and two outputs (Data Terminal Ready and Request To Send). These modem controls are connected to either the SCC or to a dedicated hardware register.

A.3.1 Data Set Ready

The Data Set Ready (DSR) inputs 0–15 are connected to a discrete hardware register and are read using two address decodes as shown in Figure A–2.

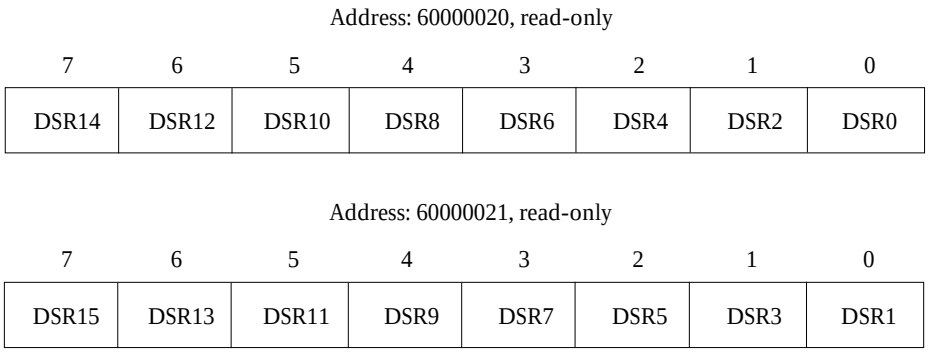


Figure A–2: Data Set Ready Address Decodes

A.3.2 Clear To Send

The Clear To Send (CTS) inputs 0–15 are connected to the CTSA (even-numbered ports) and CTSB (odd-numbered ports) inputs of the SCCs and are read using the appropriate SCC accesses.

A.3.3 Data Carrier Detect

The Data Carrier Detect (DCD) inputs 0–15 are connected to the DCDA (even-numbered ports) and DCDB (odd-numbered ports) inputs of the SCCs and are read using the appropriate SCC accesses.

A.3.4 Data Terminal Ready

The Data Terminal Ready (DTR) outputs 0–15 are connected to a discrete hardware register and are written using two address decodes as shown in Figure A–3.

A.3.5 Request To Send

The Request To Send (RTS) outputs 0–15 are connected to the RTSA (even-numbered ports) and RTSB (odd-numbered ports) outputs of the SCCs and are written using the appropriate SCC accesses.

Address: 60000021, write-only

7	6	5	4	3	2	1	0
DTR15	DTR14	DTR13	DTR12	DTR11	DTR10	DTR9	DTR8

Address: 60000022, write-only

7	6	5	4	3	2	1	0
DTR7	DTR6	DTR5	DTR4	DTR3	DTR2	DTR1	DTR0

Figure A–3: Data Terminal Ready Address Decodes

A.4 Connector Pin Assignments

The ICP is connected to the distribution panel by two cables. Each cable has a high-density 80-pin connector on the ICP end and two 40-pin connectors on the distribution panel end. Figure A–4 shows the orientation of the pins on the ICP connector. The connector nearer the console port (debug port) is designated B, and the connector farther away is designated A. See Figure 5–7 on page 69 for the location of the connectors.

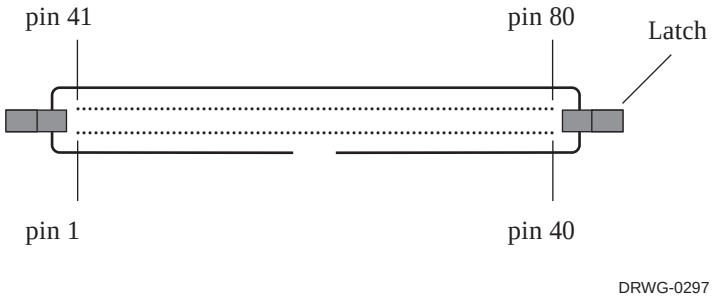
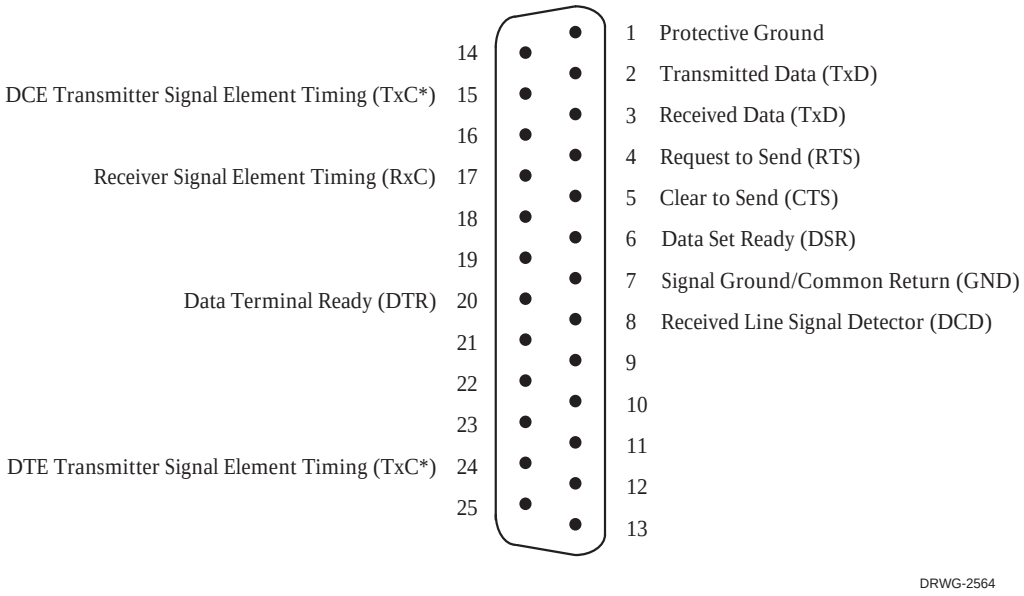


Figure A–4: Front View of 80-pin Connector on the ICP

Table A-2 shows the signal mapping for the cables that connect the ICP to the distribution panel. Figure A–5 shows the EIA-232 connector with the supported signals, their proper signal names, and the three-letter mnemonics used in Table A-2.



* Jumpers on the distribution panel and the ICP determine which of these pins is connected to the SCC.

Figure A–5: EIA-232 Interface

Table A-2: Pin Assignments for 16-port EIA-232 Cable Connectors

80-pin Connector	40-pin Connector	Port	Signal	80-pin Connector	40-pin Connector	Port	Signal
A-1	J8-1	00	GND	A-41	J9-1	04	GND
A-2	J8-2	00	TXD	A-42	J9-2	04	TXD
A-3	J8-3	00	RTS	A-43	J9-3	04	RTS
A-4	J8-4	00	RXD	A-44	J9-4	04	RXD
A-5	J8-5	00	CTS	A-45	J9-5	04	CTS
A-6	J8-6	00	DCD	A-46	J9-6	04	DCD
A-7	J8-7	00	DTR	A-47	J9-7	04	DTR
A-8	J8-8	00	RXC	A-48	J9-8	04	RXC
A-9	J8-9	00	DSR	A-49	J9-9	04	DSR
A-10	J8-10	00	TXC	A-50	J9-10	04	TXC
A-11	J8-11	01	GND	A-51	J9-11	05	GND
A-12	J8-12	01	TXD	A-52	J9-12	05	TXD
A-13	J8-13	01	RTS	A-53	J9-13	05	RTS
A-14	J8-14	01	RXD	A-54	J9-14	05	RXD
A-15	J8-15	01	CTS	A-55	J9-15	05	CTS
A-16	J8-16	01	DCD	A-56	J9-16	05	DCD
A-17	J8-17	01	DTR	A-57	J9-17	05	DTR
A-18	J8-18	01	RXC	A-58	J9-18	05	RXC
A-19	J8-19	01	DSR	A-59	J9-19	05	DSR
A-20	J8-20	01	TXC	A-60	J9-20	05	TXC
A-21	J8-21	02	GND	A-61	J9-21	06	GND
A-22	J8-22	02	TXD	A-62	J9-22	06	TXD
A-23	J8-23	02	RTS	A-63	J9-23	06	RTS
A-24	J8-24	02	RXD	A-64	J9-24	06	RXD
A-25	J8-25	02	CTS	A-65	J9-25	06	CTS
A-26	J8-26	02	DCD	A-66	J9-26	06	DCD
A-27	J8-27	02	DTR	A-67	J9-27	06	DTR
A-28	J8-28	02	RXC	A-68	J9-28	06	RXC
A-29	J8-29	02	DSR	A-69	J9-29	06	DSR
A-30	J8-30	02	TXC	A-70	J9-30	06	TXC
A-31	J8-31	03	GND	A-71	J9-31	07	GND
A-32	J8-32	03	TXD	A-72	J9-32	07	TXD
A-33	J8-33	03	RTS	A-73	J9-33	07	RTS
A-34	J8-34	03	RXD	A-74	J9-34	07	RXD
A-35	J8-35	03	CTS	A-75	J9-35	07	CTS
A-36	J8-36	03	DCD	A-76	J9-36	07	DCD
A-37	J8-37	03	DTR	A-77	J9-37	07	DTR
A-38	J8-38	03	RXC	A-78	J9-38	07	RXC
A-39	J8-39	03	DSR	A-79	J9-39	07	DSR
A-40	J8-40	03	TXC	A-80	J9-40	07	TXC

Table A-2: Pin Assignments for 16-port EIA-232 Cable Connectors (*Cont'd*)

80-pin Connector	40-pin Connector	Port	Signal	80-pin Connector	40-pin Connector	Port	Signal
B-1	J8-1	08	GND	B-41	J9-1	12	GND
B-2	J8-2	08	TXD	B-42	J9-2	12	TXD
B-3	J8-3	08	RTS	B-43	J9-3	12	RTS
B-4	J8-4	08	RXD	B-44	J9-4	12	RXD
B-5	J8-5	08	CTS	B-45	J9-5	12	CTS
B-6	J8-6	08	DCD	B-46	J9-6	12	DCD
B-7	J8-7	08	DTR	B-47	J9-7	12	DTR
B-8	J8-8	08	RXC	B-48	J9-8	12	RXC
B-9	J8-9	08	DSR	B-49	J9-9	12	DSR
B-10	J8-10	08	TXC	B-50	J9-10	12	TXC
B-11	J8-11	09	GND	B-51	J9-11	13	GND
B-12	J8-12	09	TXD	B-52	J9-12	13	TXD
B-13	J8-13	09	RTS	B-53	J9-13	13	RTS
B-14	J8-14	09	RXD	B-54	J9-14	13	RXD
B-15	J8-15	09	CTS	B-55	J9-15	13	CTS
B-16	J8-16	09	DCD	B-56	J9-16	13	DCD
B-17	J8-17	09	DTR	B-57	J9-17	13	DTR
B-18	J8-18	09	RXC	B-58	J9-18	13	RXC
B-19	J8-19	09	DSR	B-59	J9-19	13	DSR
B-20	J8-20	09	TXC	B-60	J9-20	13	TXC
B-21	J8-21	10	GND	B-61	J9-21	14	GND
B-22	J8-22	10	TXD	B-62	J9-22	14	TXD
B-23	J8-23	10	RTS	B-63	J9-23	14	RTS
B-24	J8-24	10	RXD	B-64	J9-24	14	RXD
B-25	J8-25	10	CTS	B-65	J9-25	14	CTS
B-26	J8-26	10	DCD	B-66	J9-26	14	DCD
B-27	J8-27	10	DTR	B-67	J9-27	14	DTR
B-28	J8-28	10	RXC	B-68	J9-28	14	RXC
B-29	J8-29	10	DSR	B-69	J9-29	14	DSR
B-30	J8-30	10	TXC	B-70	J9-30	14	TXC
B-31	J8-31	11	GND	B-71	J9-31	15	GND
B-32	J8-32	11	TXD	B-72	J9-32	15	TXD
B-33	J8-33	11	RTS	B-73	J9-33	15	RTS
B-34	J8-34	11	RXD	B-74	J9-34	15	RXD
B-35	J8-35	11	CTS	B-75	J9-35	15	CTS
B-36	J8-36	11	DCD	B-76	J9-36	15	DCD
B-37	J8-37	11	DTR	B-77	J9-37	15	DTR
B-38	J8-38	11	RXC	B-78	J9-38	15	RXC
B-39	J8-39	11	DSR	B-79	J9-39	15	DSR
B-40	J8-40	11	TXC	B-80	J9-40	15	TXC

Appendix B 8-port MIL-STD-188C and 8-port EIA-232 Electrical Interface Module

The Electrical Interface Modules referred to as EIME or EIMF are 8-port daughter-boards for the ICP base board and support MIL-STD-188C or EIA-232 operation, respectively. Because their jumper locations and pinouts are identical, and their functionality is very similar, they are both discussed in this Appendix.

B.1 Modem Clocks

EIME and EIMF support the full functionality of the serial communications controller (SCC) with respect to external and internal clock sources.

B.1.1 Receive Clock Inputs

The Receive Clock (RxCA and RxCB) pins are always inputs to the SCC. RxC inputs 0–7 are connected to the RxCA (even-numbered ports) and RxCB (odd-numbered ports) inputs of the SCCs.

B.1.2 Transmit Clock Inputs and Outputs

The Transmit Clock (TxCA and TxCB) pins can be programmed as either inputs to or outputs from the SCC. To support this bidirectional interface, EIME and EIMF are equipped with both a driver and a receiver for this line.

The TxC0–7 drivers and receivers are selected using the jumper settings shown in Table B–1. Refer also to Figure B–1.

Caution

You must also set the jumper on the distribution panel and use the protocol software to configure the desired clocking.

Table B–1: Clock Jumper Settings for 8-port MIL-STD-188C or EIA-232

	Transmit Clock Source[a]	
Port	**Internal (DTE)**	**External (DCE)**
0	K8 1-2	K16 1-2
1	K7 1-2	K15 1-2
2	K6 1-2	K14 1-2
3	K5 1-2	K13 1-2
4	K4 1-2	K12 1-2
5	K3 1-2	K11 1-2
6	K2 1-2	K10 1-2
7	K1 1-2	K9 1-2

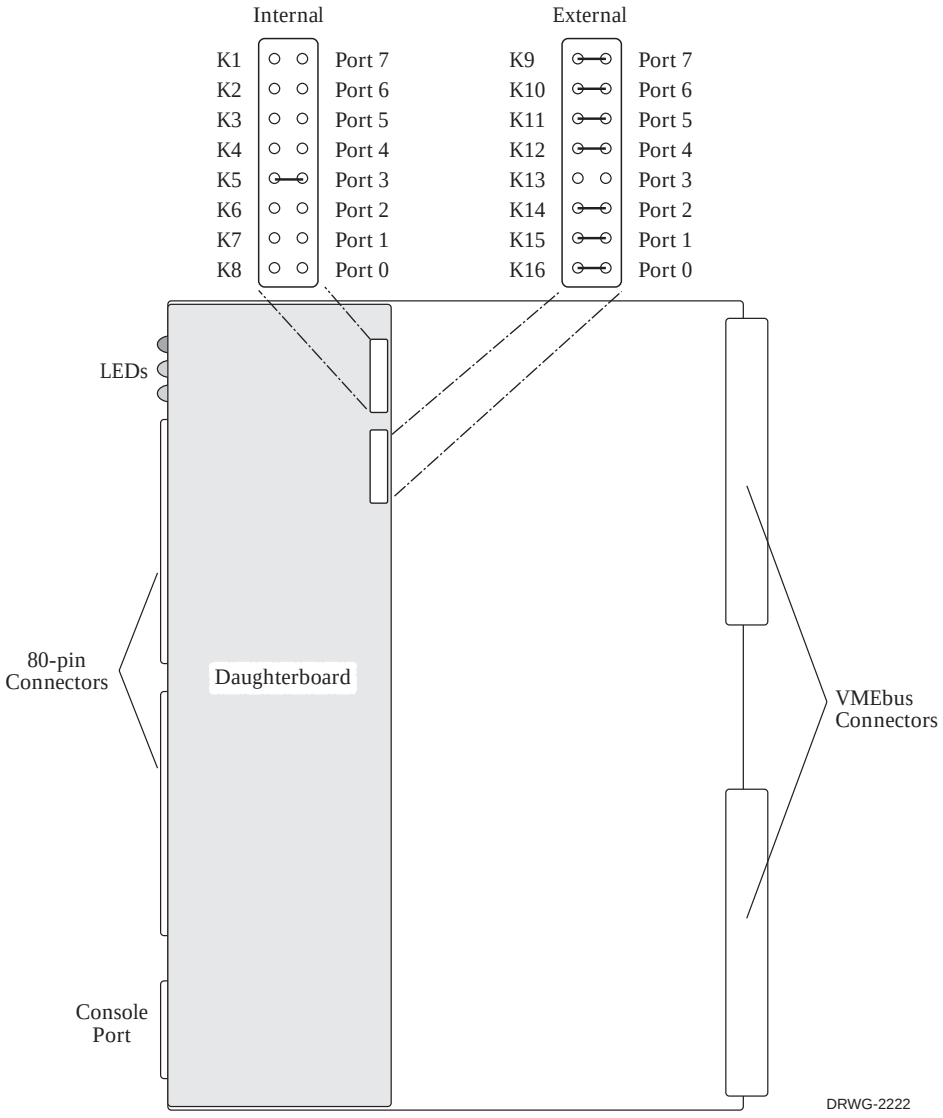
[a] Only one jumper per port

B.2 Z85C30 or Z85230 Serial Communications Controller

EIME and EIMF have four Z85C30 (ICP6000/ICP9000) or Z85230 (ICP6000X/ ICP9000X) serial communications controllers (SCCs). Each SCC implements two communications ports.

B.2.1 SCC Register Access

Access to the SCC data and command registers is made using the addresses shown in Table 5–1 on page 51. Hardware protection is provided so that the SCC "write recovery" limits are automatically met.



Notes:
1. All settings from the factory are external.
2. This example shows one internal and seven external clock jumper settings.
3. You must also set the distribution panel jumpers to match these settings.

Figure B–1: 8-port MIL-STD-188C or EIA-232 Clock Jumper Settings

B.2.2 SCC Timebase

The SCCs are driven from a 7.3728 MHz (ICP6000/ICP9000) or 14.7456 MHz (ICP6000X/ICP9000X) peripheral clock signal, PCLK.

B.2.3 SCC DMA

Each communications port is assigned two DMA channels, one for transmitting and the other for receiving. Section 5.8 on page 58 describes the setup and operation of the DMA controller.

B.3 Transmit/Receive Polarity

As required by MIL-STD-188C, EIME is equipped with jumper options to select the polarity of the transmit and receive data signals. These jumpers are set as shown in Table B–2.

EIMF is configured for EIA-232 operation. The jumper positions are permanently soldered into inverting mode.

B.4 Driver Supply Voltages

As required by MIL-STD-188C, EIME is equipped with Zener diodes to reduce the voltage supplied to the line driver devices from the +12V and -12V supplied by the VMEbus to MIL-STD-188C levels.

EIMF has no Zener diodes and utilizes the full +12V and -12V levels.

B.5 Interface Driver Waveshaping

As required by MIL-STD-188C, EIME is equipped for the installation of waveshaping capacitors used in conjunction with the 1488 line drivers. The capacitor locations for each output signal are shown in Table B–3. No capacitors are installed in the standard factory version of EIME or EIMF.

Table B–2: Jumper Settings for 8-port MIL-STD-188C

Function[a]	Inverted	Not Inverted
RXD00	K32 1-2	K32 2-3
TXD00	K24 2-3	K24 1-2
RXD01	K31 1-2	K31 2-3
TXD01	K23 2-3	K23 1-2
RXD02	K30 1-2	K30 2-3
TXD02	K22 2-3	K22 1-2
RXD03	K29 1-2	K29 2-3
TXD03	K21 2-3	K21 1-2
RXD04	K28 1-2	K28 2-3
TXD04	K20 2-3	K20 1-2
RXD05	K27 1-2	K27 2-3
TXD05	K19 2-3	K19 1-2
RXD06	K26 1-2	K26 2-3
TXD06	K18 2-3	K18 1-2
RXD07	K25 1-2	K25 2-3
TXD07	K17 2-3	K17 1-2

[a] One jumper per signal

B.6 Interface Receiver Response Control

As required by MIL-STD-188C, EIME is equipped for the installation of response control capacitors used in conjunction with 1489 line receivers. The capacitor locations for each input signal are shown in Table B–4. No capacitors are installed in the factory standard version of EIME or EIMF.

B.7 Modem Control

EIME and EIMF support five modem control signals: three inputs (Data Set Ready, Clear To Send, and Data Carrier Detect) and two outputs (Data Terminal Ready and

Table B–3: Capacitor Locations for Waveshaping

Signal Name	Capacitor Location	Signal Name	Capacitor Location
DTR00	C29	DTR04	C39
TXD00	C9	TXD04	C19
RTS00	C11	RTS04	C21
TXC00	C4	TXC04	C12
DTR01	C31	DTR05	C41
TXD01	C3	TXD05	C10
RTS01	C43	RTS05	C33
TXC01	C45	TXC05	C35
DTR02	C22	DTR06	C32
TXD02	C13	TXD06	C23
RTS02	C15	RTS06	C25
TXC02	C6	TXC06	C16
DTR03	C20	DTR07	C30
TXD03	C5	TXD07	C14
RTS03	C36	RTS07	C26
TXC03	C34	TXC07	C24

Request To Send). These modem controls are connected to either the SCC or to a dedicated hardware register.

B.7.1 Data Set Ready

The Data Set Ready (DSR) inputs 0–7 are connected to a discrete hardware register and are read using two address decodes as shown in Figure B–2.

B.7.2 Clear To Send

The Clear To Send (CTS) inputs 0–7 are connected to the CTSA (even-numbered ports) and CTSB (odd-numbered ports) inputs of the SCCs and are read using the appropriate SCC accesses.

Table B–4: Capacitor Locations for Response Control

Signal Name	Capacitor Location	Signal Name	Capacitor Location
DSR00	C104	DSR04	C100
RXD00	C79	RXD04	C49
CTS00	C81	CTS04	C51
RXC00	C76	RXC04	C42
DCD00	C75	DCD04	C40
TXC00	C53	TXC04	C69
DSR01	C105	DSR05	C102
RXD01	C83	RXD05	C59
CTS01	C85	CTS05	C61
RXC01	C82	RXC05	C52
DCD01	C80	DCD05	C50
TXC01	C55	TXC05	C64
DSR02	C103	DSR06	C97
RXD02	C89	RXD06	C67
CTS02	C91	CTS06	C69
RXC02	C86	RXC06	C62
DCD02	C84	DCD06	C60
TXC02	C46	TXC06	C16
DSR03	C101	DSR07	C95
RXD03	C94	RXD07	C73
CTS03	C96	CTS07	C74
RXC03	C92	RXC07	C70
DCD03	C90	DCD07	C68
TXC03	C44	TXC07	C54

B.7.3 Data Carrier Detect

The Data Carrier Detect (DCD) inputs 0–7 are connected to the DCDA (even-numbered ports) and DCDB (odd-numbered ports) inputs of the SCCs and are read using the appropriate SCC accesses.

Address: 60000020, read-only

7	6	5	4	3	2	1	0
Unused	Unused	Unused	Unused	DSR6	DSR4	DSR2	DSR0

Address: 60000021, read-only

7	6	5	4	3	2	1	0
Unused	Unused	Unused	Unused	DSR7	DSR5	DSR3	DSR1

Figure B–2: Data Set Ready Address Decodes

B.7.4 Data Terminal Ready

The Data Terminal Ready (DTR) outputs 0–7 are connected to a discrete hardware register and are written using a single address decode as shown in Figure B–3.

Address: 60000020, write-only

7	6	5	4	3	2	1	0
DTR7	DTR6	DTR5	DTR4	DTR3	DTR2	DTR1	DTR0

Figure B–3: Data Terminal Ready Address Decode

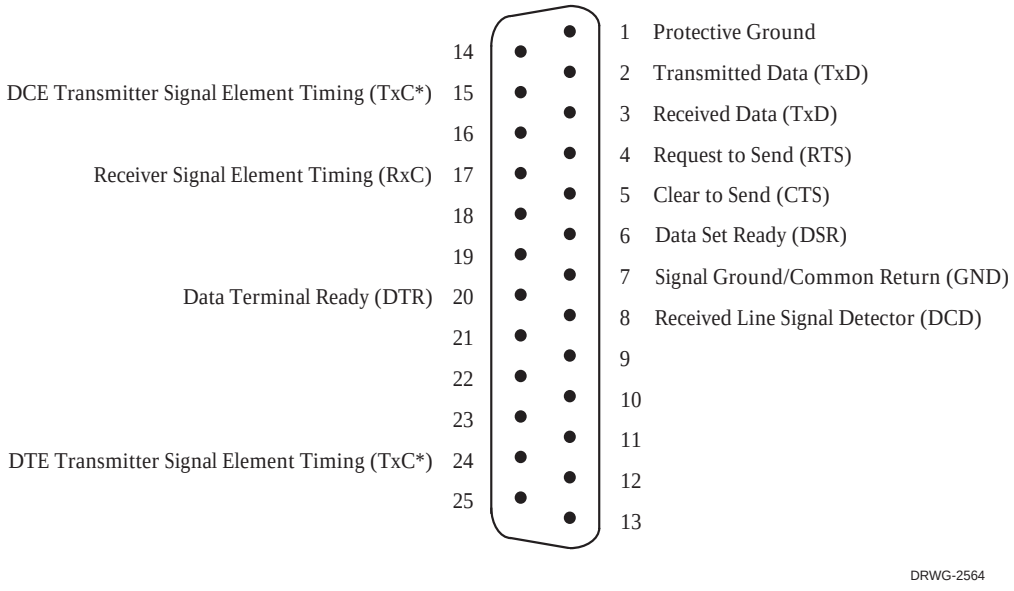
B.7.5 Request To Send

The Request To Send (RTS) outputs 0–7 are connected to the RTSA (even-numbered ports) and RTSB (odd-numbered ports) outputs of the SCCs and are written using the appropriate SCC accesses.

B.8 Connector Pin Assignments

The ICP is connected to the distribution panel by a cable that has a high-density 80-pin connector on the ICP end and two 40-pin connectors on the distribution panel end. Figure A–4 on page 88 shows the orientation of the pins on the ICP connector. See Figure 5–7 on page 69 for the location of the connectors. The ICP in this case has only the top 80-pin connector.

Table B–5 shows the signal mapping for the cable that connects the ICP to the distribution panel. Figure B–4 shows the EIA-232 or MIL-STD-188C connector with the supported signals, their proper signal names, and the three-letter mnemonics used in Table B–5.



* Jumpers on the distribution panel and the ICP determine which of these pins is connected to the SCC.

Figure B–4: EIA-232 or MIL-STD-188C Interface

Table B–5: Pin Assignments for 8-port MIL-STD-188C & EIA-232 Cable Connectors

80-pin Connector	40-pin Connector	Port	Signal	80-pin Connector	40-pin Connector	Port	Signal
A-1	J8-1	00	GND	A-41	J9-1	04	GND
A-2	J8-2	00	TXD	A-42	J9-2	04	TXD
A-3	J8-3	00	RTS	A-43	J9-3	04	RTS
A-4	J8-4	00	RXD	A-44	J9-4	04	RXD
A-5	J8-5	00	CTS	A-45	J9-5	04	CTS
A-6	J8-6	00	DCD	A-46	J9-6	04	DCD
A-7	J8-7	00	DTR	A-47	J9-7	04	DTR
A-8	J8-8	00	RXC	A-48	J9-8	04	RXC
A-9	J8-9	00	DSR	A-49	J9-9	04	DSR
A-10	J8-10	00	TXC	A-50	J9-10	04	TXC
A-11	J8-11	01	GND	A-51	J9-11	05	GND
A-12	J8-12	01	TXD	A-52	J9-12	05	TXD
A-13	J8-13	01	RTS	A-53	J9-13	05	RTS
A-14	J8-14	01	RXD	A-54	J9-14	05	RXD
A-15	J8-15	01	CTS	A-55	J9-15	05	CTS
A-16	J8-16	01	DCD	A-56	J9-16	05	DCD
A-17	J8-17	01	DTR	A-57	J9-17	05	DTR
A-18	J8-18	01	RXC	A-58	J9-18	05	RXC
A-19	J8-19	01	DSR	A-59	J9-19	05	DSR
A-20	J8-20	01	TXC	A-60	J9-20	05	TXC
A-21	J8-21	02	GND	A-61	J9-21	06	GND
A-22	J8-22	02	TXD	A-62	J9-22	06	TXD
A-23	J8-23	02	RTS	A-63	J9-23	06	RTS
A-24	J8-24	02	RXD	A-64	J9-24	06	RXD
A-25	J8-25	02	CTS	A-65	J9-25	06	CTS
A-26	J8-26	02	DCD	A-66	J9-26	06	DCD
A-27	J8-27	02	DTR	A-67	J9-27	06	DTR
A-28	J8-28	02	RXC	A-68	J9-28	06	RXC
A-29	J8-29	02	DSR	A-69	J9-29	06	DSR
A-30	J8-30	02	TXC	A-70	J9-30	06	TXC
A-31	J8-31	03	GND	A-71	J9-31	07	GND
A-32	J8-32	03	TXD	A-72	J9-32	07	TXD
A-33	J8-33	03	RTS	A-73	J9-33	07	RTS
A-34	J8-34	03	RXD	A-74	J9-34	07	RXD
A-35	J8-35	03	CTS	A-75	J9-35	07	CTS
A-36	J8-36	03	DCD	A-76	J9-36	07	DCD
A-37	J8-37	03	DTR	A-77	J9-37	07	DTR
A-38	J8-38	03	RXC	A-78	J9-38	07	RXC
A-39	J8-39	03	DSR	A-79	J9-39	07	DSR
A-40	J8-40	03	TXC	A-80	J9-40	07	TXC

Appendix

C 8-port V.35 CCITT Electrical Interface Module

This Electrical Interface Module is referred to as EIGH. It is an 8-port daughterboard for the ICP base board and supports V.35 operation.

C.1 Modem Clocks

EIGH supports the full functionality of the serial communications controller (SCC) with respect to external and internal clock sources.

C.1.1 Receive Clock Inputs

The Receive Clock (RxCA and RxCB) pins are always inputs to the SCC. RxC inputs 0–7 are connected to the RxCA (even-numbered ports) and RxCB (odd-numbered ports) inputs of the SCCs.

C.1.2 Transmit Clock Inputs and Outputs

The Transmit Clock (TxCA and TxCB) pins can be programmed as either inputs to or outputs from the SCC. To support this bidirectional interface, EIGH is equipped with both a driver and a receiver for this line.

The TxC0–7 drivers and receivers are selected using the jumper settings shown in Table C–1. Refer also to Figure C–1.

Caution

You must also use the protocol software to configure the desired clocking.

Table C–1: Clock Jumper Settings for 8-port V.35

	Transmit Clock Source[a]	
Port	**Internal (DTE)**	**External (DCE)**
0	K1 1-2	K1 3-4
1	K2 1-2	K2 3-4
2	K3 1-2	K3 3-4
3	K4 1-2	K4 3-4
4	K5 1-2	K5 3-4
5	K6 1-2	K6 3-4
6	K7 1-2	K7 3-4
7	K8 1-2	K8 3-4

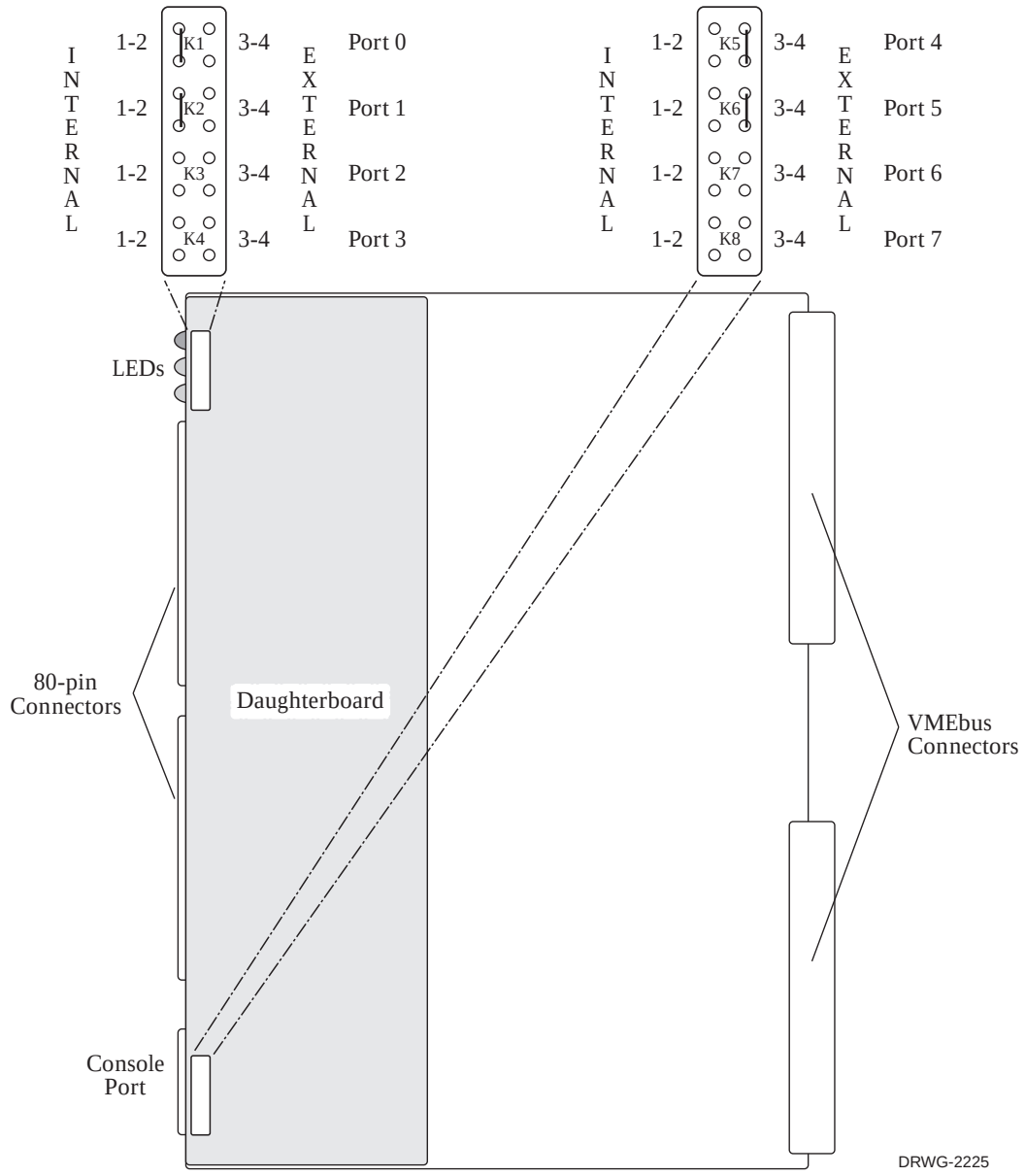
[a] Only one jumper per port

C.2 Z85C30 or Z85230 Serial Communications Controller

EIGH has four Z85C30 (ICP6000/ICP9000) or Z85230 (ICP6000X/ICP9000X) serial communications controllers (SCCs). Each SCC implements two communications ports.

C.2.1 SCC Register Access

Access to the SCC data and command registers is made using the addresses shown in Table 5–1 on page 51. Hardware protection is provided so that the SCC "write recovery" limits are automatically met.



Notes:
1. All settings from the factory are external.
2. This example shows two internal and two external clock jumper settings.

Figure C–1: 8-port V.35 Clock Jumper Settings

C.2.2 SCC Timebase

The SCCs are driven from a 7.3728 MHz (ICP6000/ICP9000) or 14.7456 MHz (ICP6000X/ICP9000X) peripheral clock signal, PCLK.

C.2.3 SCC DMA

Each communications port is assigned two DMA channels, one for transmitting and the other for receiving. Section 5.8 on page 58 describes the setup and operation of the DMA controller.

C.3 Driver Supply Voltages

EIGH is equipped with TO-92 three-terminal regulators to reduce the available -12V to -5V as required by the V.35 differential drivers and receivers.

C.4 Single-ended Interface

EIGH is equipped with 14C88 single-ended line drivers that feature internal slew rate control as required by the V.35 specification. EIGH is also equipped with 14C89 single-ended line receivers that meet the V.35 specifications without the use of external capacitors.

C.5 Differential Interface

EIGH is equipped with XR-T3588 V.35 drivers and XR-T3589 V.35 receivers that are designed to meet the requirements of the V.35 specification. The differential lines are terminated externally to the drivers and receivers.

C.6 Modem Control

EIGH supports seven single-ended modem control signals: four inputs (Data Set Ready, Calling Indicator, Clear To Send, and Data Carrier Detect) and three outputs (Data Ter-

minal Ready, Local Test, and Request To Send). These modem controls are connected to either the SCC or to dedicated hardware registers.

C.6.1 Data Set Ready and Calling Indicator

The Data Set Ready (DSR) inputs 0–7 and Calling Indicator (CI) inputs 0–7 are connected to a discrete hardware register and are read using two address decodes as shown in Figure C–2.

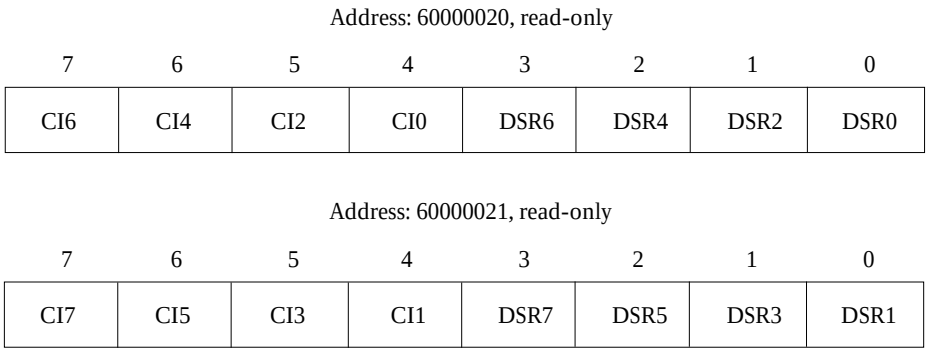


Figure C–2: Data Set Ready and Calling Indicator Address Decodes

C.6.2 Clear To Send

The Clear To Send (CTS) inputs 0–7 are connected to the CTSA (even-numbered ports) and CTSB (odd-numbered ports) inputs of the SCCs and are read using the appropriate SCC accesses.

C.6.3 Received Line Signal Detector

The Received Line Signal Detector (RLSD) inputs 0–7 are connected to the RLSDA (even-numbered ports) and RLSDB (odd-numbered ports) inputs of the SCCs and are read using the appropriate SCC accesses.

C.6.4 Data Terminal Ready and Local Test

The Data Terminal Ready (DTR) outputs 0–7 and Local Test (LT) outputs 0–7 are connected to a discrete hardware register and are written using two address decodes as shown in Figure C–3.

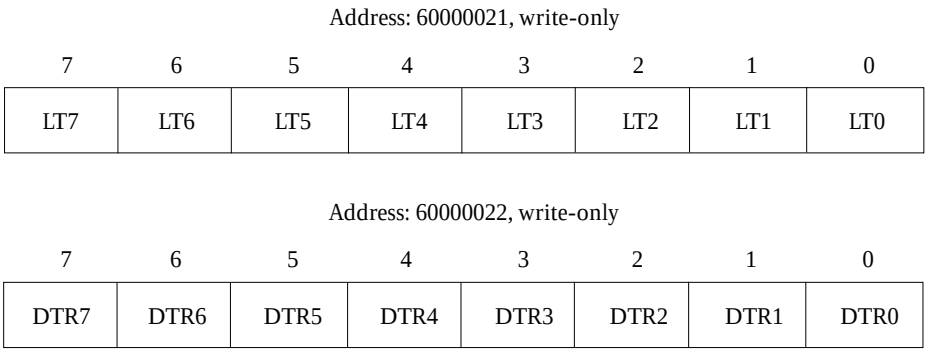


Figure C–3: Data Terminal Ready and Local Test Address Decodes

C.6.5 Request To Send

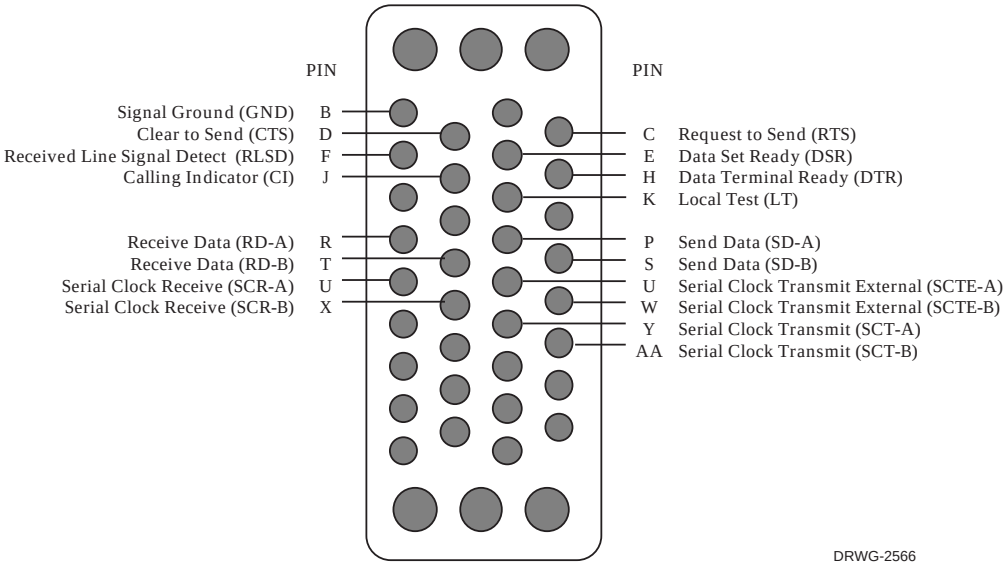
The Request To Send (RTS) outputs 0–7 are connected to the RTSA (even-numbered ports) and RTSB (odd-numbered ports) outputs of the SCCs and are written using the appropriate SCC accesses.

C.7 Connector Pin Assignments

The ICP is connected to the distribution panel by two cables. Each cable has a high-density 80-pin connector on the ICP end and two 40-pin connectors on the distribution panel end. Figure A–4 on page 88 shows the orientation of the pins on the ICP connector. The connector nearer the console port (debug port) is designated B, and the connector farther away is designated A. See Figure 5–7 on page 69 for the location of the connectors.

Adapter cables are used to map the 25-pin D-connector pinout on a distribution panel to the V.35 connector shown in Figure C–4.

Table C–2 shows the signal mapping for the cables that connects the ICP to the distribution panel. Figure C–4 shows the V.35 connector with the supported signals, their proper signal names, and the mnemonics used in Table C–2.



Signal suffixes -A and -B refer to differential signal pairs.

Figure C–4: V.35 Interface

Table C–2: Pin Assignments for 8-port V.35 Cable Connectors

80-pin Connector	40-pin Connector	Port	Signal	80-pin Connector	40-pin Connector	Port	Signal
A-1	J8-1	00	GND	A-41	J9-1	02	GND
A-2	J8-2	00	GND	A-42	J9-2	02	GND
A-3	J8-3	00	GND	A-43	J9-3	02	GND
A-4	J8-4	00	CTS	A-44	J9-4	02	CTS
A-5	J8-5	00	RLSD	A-45	J9-5	02	RLSD
A-6	J8-6	00	DSR	A-46	J9-6	02	DSR
A-7	J8-7	00	CI	A-47	J9-7	02	CI
A-8	J8-8	00	RTS	A-48	J9-8	02	RTS
A-9	J8-9	00	LT	A-49	J9-9	02	LT
A-10	J8-10	00	DTR	A-50	J9-10	02	DTR
A-11	J8-11	00	SD-A	A-51	J9-11	02	SD-A
A-12	J8-12	00	SD-B	A-52	J9-12	02	SD-B
A-13	J8-13	00	RD-A	A-53	J9-13	02	RD-A
A-14	J8-14	00	RD-B	A-54	J9-14	02	RD-B
A-15	J8-15	00	SCR-A	A-55	J9-15	02	SCR-A
A-16	J8-16	00	SCR-B	A-56	J9-16	02	SCR-B
A-17	J8-17	00	SCTE-A	A-57	J9-17	02	SCTE-A
A-18	J8-18	00	SCTE-B	A-58	J9-18	02	SCTE-B
A-19	J8-19	00	SCT-A	A-59	J9-19	02	SCT-A
A-20	J8-20	00	SCT-B	A-60	J9-20	02	SCT-B
A-21	J8-21	01	GND	A-61	J9-21	03	GND
A-22	J8-22	01	GND	A-62	J9-22	03	GND
A-23	J8-23	01	GND	A-63	J9-23	03	GND
A-24	J8-24	01	CTS	A-64	J9-24	03	CTS
A-25	J8-25	01	RLSD	A-65	J9-25	03	RLSD
A-26	J8-26	01	DSR	A-66	J9-26	03	DSR
A-27	J8-27	01	CI	A-67	J9-27	03	CI
A-28	J8-28	01	RTS	A-68	J9-28	03	RTS
A-29	J8-29	01	LT	A-69	J9-29	03	LT
A-30	J8-30	01	DTR	A-70	J9-30	03	DTR
A-31	J8-31	01	SD-A	A-71	J9-31	03	SD-A
A-32	J8-32	01	SD-B	A-72	J9-32	03	SD-B
A-33	J8-33	01	RD-A	A-73	J9-33	03	RD-A
A-34	J8-34	01	RD-B	A-74	J9-34	03	RD-B
A-35	J8-35	01	SCR-A	A-75	J9-35	03	SCR-A
A-36	J8-36	01	SCR-B	A-76	J9-36	03	SCR-B
A-37	J8-37	01	SCTE-A	A-77	J9-37	03	SCTE-A
A-38	J8-38	01	SCTE-B	A-78	J9-38	03	SCTE-B
A-39	J8-39	01	SCT-A	A-79	J9-39	03	SCT-A
A-40	J8-40	01	SCT-B	A-80	J9-40	03	SCT-B

Table C–2: Pin Assignments for 8-port V.35 Cable Connectors (*Cont'd*)

80-pin Connector	40-pin Connector	Port	Signal	80-pin Connector	40-pin Connector	Port	Signal
B-1	J8-1	04	GND	B-41	J9-1	06	GND
B-2	J8-2	04	GND	B-42	J9-2	06	GND
B-3	J8-3	04	GND	B-43	J9-3	06	GND
B-4	J8-4	04	CTS	B-44	J9-4	06	CTS
B-5	J8-5	04	RLSD	B-45	J9-5	06	RLSD
B-6	J8-6	04	DSR	B-46	J9-6	06	DSR
B-7	J8-7	04	CI	B-47	J9-7	06	CI
B-8	J8-8	04	RTS	B-48	J9-8	06	RTS
B-9	J8-9	04	LT	B-49	J9-9	06	LT
B-10	J8-10	04	DTR	B-50	J9-10	06	DTR
B-11	J8-11	04	SD-A	B-51	J9-11	06	SD-A
B-12	J8-12	04	SD-B	B-52	J9-12	06	SD-B
B-13	J8-13	04	RD-A	B-53	J9-13	06	RD-A
B-14	J8-14	04	RD-B	B-54	J9-14	06	RD-B
B-15	J8-15	04	SCR-A	B-55	J9-15	06	SCR-A
B-16	J8-16	04	SCR-B	B-56	J9-16	06	SCR-B
B-17	J8-17	04	SCTE-A	B-57	J9-17	06	SCTE-A
B-18	J8-18	04	SCTE-B	B-58	J9-18	06	SCTE-B
B-19	J8-19	04	SCT-A	B-59	J9-19	06	SCT-A
B-20	J8-20	04	SCT-B	B-60	J9-20	06	SCT-B
B-21	J8-21	05	GND	B-61	J9-21	07	GND
B-22	J8-22	05	GND	B-62	J9-22	07	GND
B-23	J8-23	05	GND	B-63	J9-23	07	GND
B-24	J8-24	05	CTS	B-64	J9-24	07	CTS
B-25	J8-25	05	RLSD	B-65	J9-25	07	RLSD
B-26	J8-26	05	DSR	B-66	J9-26	07	DSR
B-27	J8-27	05	CI	B-67	J9-27	07	CI
B-28	J8-28	05	RTS	B-68	J9-28	07	RTS
B-29	J8-29	05	LT	B-69	J9-29	07	LT
B-30	J8-30	05	DTR	B-70	J9-30	07	DT
B-31	J8-31	05	SD-A	B-71	J9-31	07	SD-A
B-32	J8-32	05	SD-B	B-72	J9-32	07	SD-B
B-33	J8-33	05	RD-A	B-73	J9-33	07	RD-A
B-34	J8-34	05	RD-B	B-74	J9-34	07	RD-B
B-35	J8-35	05	SCR-A	B-75	J9-35	07	SCR-A
B-36	J8-36	05	SCR-B	B-76	J9-36	07	SCR-B
B-37	J8-37	05	SCTE-A	B-77	J9-37	07	SCTE-A
B-38	J8-38	05	SCTE-B	B-78	J9-38	07	SCTE-B
B-39	J8-39	05	SCT-A	B-79	J9-39	07	SCT-A
B-40	J8-40	05	SCT-B	B-80	J9-40	07	SCT-B

Appendix

D 8-port EIA-422 Electrical Interface Module

This Electrical Interface Module is referred to as EIMJ. It is an 8-port daughterboard for the ICP base board and supports both EIA-449 and EIA-530 operation.

EIMJ is equipped with 26LS31 drivers and 26LS32 receivers which are designed to meet the requirements of EIA-422. Provision is made on the EIMJ PCB for termination on all receive differential lines. The termination components are not installed on the standard factory version of the assembly.

D.1 Modem Clocks

EIMJ supports the full functionality of the serial communications controller (SCC) with respect to external and internal clock sources.

D.1.1 Receive Clock Inputs

The Receive Clock (RxCA and RxCB) pins are always inputs to the SCC. RxC inputs 0–7 are connected to the RxCA (even-numbered ports) and RxCB (odd-numbered ports) inputs of the SCCs.

D.1.2 Transmit Clock Inputs/Outputs

The Transmit Clock (TxCA and TxCB) pins can be programmed as either inputs to or outputs from the SCC. To support this bidirectional interface, EIMJ is equipped with both a driver and a receiver for this line.

The TxC0–7 drivers and receivers are selected using the jumper settings shown in Table D–1. Refer also to Figure D–1.

Caution

You must also use the protocol software to configure the desired clocking.

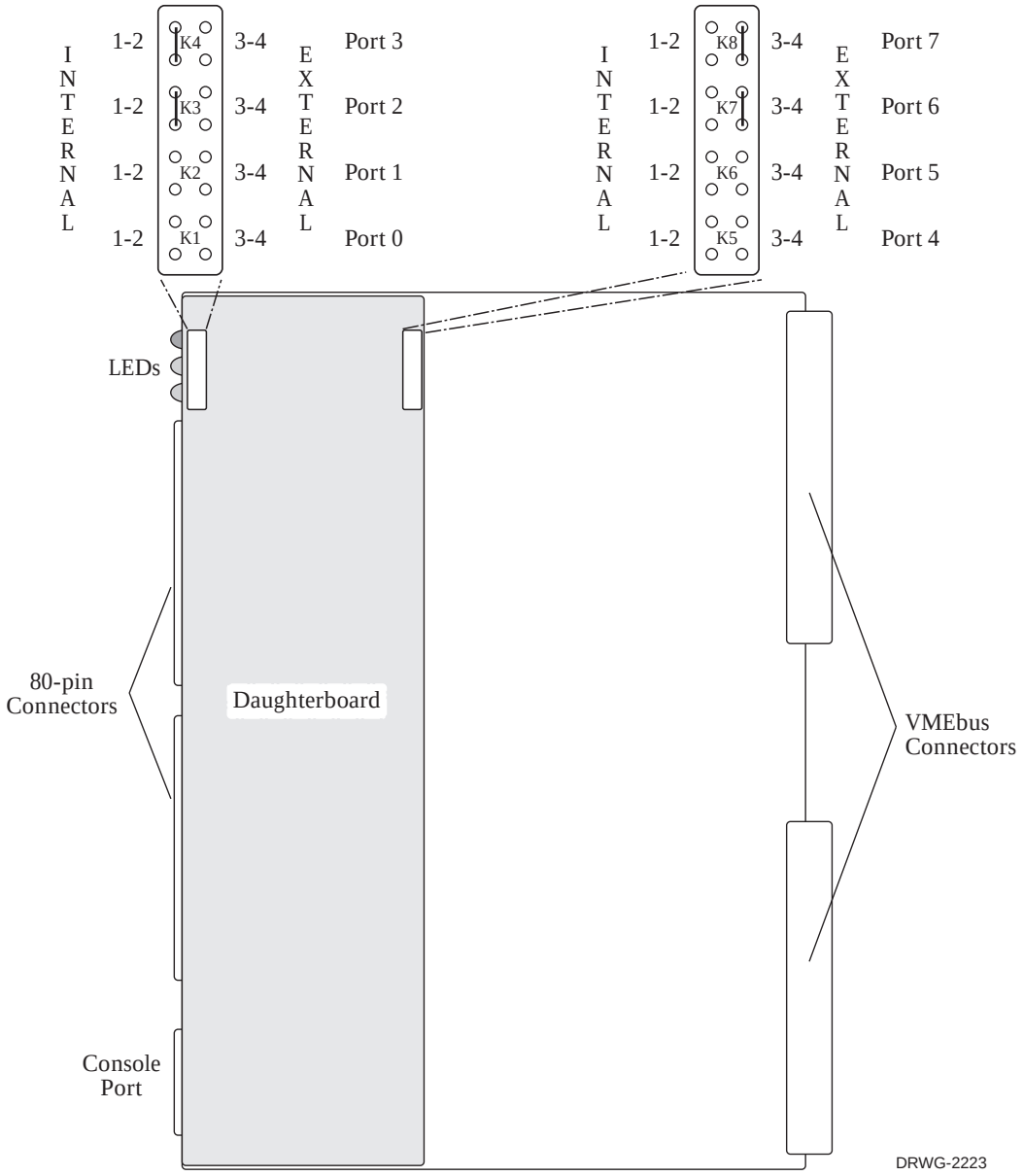
Table D–1: Clock Jumper Settings for 8-port EIA-422

	Transmit Clock Source[a]	
Port	**Internal (DTE)**	**External (DCE)**
0	K1 1-2	K1 3-4
1	K2 1-2	K2 3-4
2	K3 1-2	K3 3-4
3	K4 1-2	K4 3-4
4	K5 1-2	K5 3-4
5	K6 1-2	K6 3-4
6	K7 1-2	K7 3-4
7	K8 1-2	K8 3-4

[a] Only one jumper per port

D.2 Z85C30 or Z85230 Serial Communications Controller

EIMJ has four Z85C30 (ICP6000/ICP9000) or Z85230 (ICP6000X/ICP9000X) serial communications controllers (SCCs). Each SCC implements two communications ports.



Notes:
1. All settings from the factory are external.
2. This example shows two internal and two external clock jumper settings.

Figure D–1: 8-port EIA-422 Clock Jumper Settings

D.2.1 SCC Register Access

Access to the SCC data and command registers is made using the addresses shown in Table 5–1 on page 51. Hardware protection is provided so that the SCC "write recovery" limits are automatically met.

D.2.2 SCC Timebase

The SCCs are driven from a 7.3728 MHz (ICP6000/ICP9000) or 14.7456 MHz (ICP6000X/ICP9000X) peripheral clock signal, PCLK.

D.2.3 SCC DMA

Each communications port is assigned two DMA channels, one for transmitting and one for receiving. Section 5.8 on page 58 describes the setup and operation of the DMA controller.

D.3 Modem Control

EIMJ supports four differential modem control signals: two inputs (Clear To Send and Receiver Ready) and two outputs (Terminal Ready and Request To Send). These modem controls are connected to either the SCC or to dedicated hardware registers.

D.3.1 Clear To Send

The Clear To Send (CS) inputs 0–7 are connected to the CTSA (even-numbered ports) and CTSB (odd-numbered ports) inputs of the SCCs and are read using the appropriate SCC accesses.

D.3.2 Receiver Ready

The Receiver Ready (RR) inputs 0–7 are connected to the DCDA (even-numbered ports) and DCDB (odd-numbered ports) inputs of the SCCs and are read using the appropriate SCC accesses.

D.3.3 Terminal Ready

The Terminal Ready (TR) outputs 0–7 are connected to a discrete hardware register and are written using a single address decode as shown in Figure D–2.

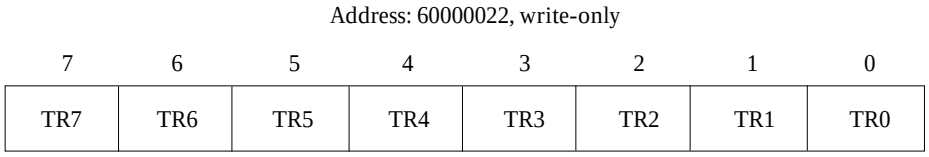


Figure D–2: Terminal Ready Address Decode

D.3.4 Request To Send

The Request To Send (RS) outputs 0–7 are connected to the RTSA (even-numbered ports) and RTSB (odd-numbered ports) outputs of the SCCs and are written using the appropriate SCC accesses.

D.4 Connector Pin Assignments

The ICP is connected to the distribution panel by two cables. Each cable has a high-density 80-pin connector on the ICP end and two 40-pin connectors on the distribution panel end. Figure A–4 on page 88 shows the orientation of the pins on the ICP connector. The connector nearer the console port (debug port) is designated B, and the connector farther away is designated A. See Figure 5–7 on page 69 for the location of the connectors.

Table D–2 shows the signal mapping for the cables that connect the ICP to either the EIA-449 or the EIA-530 distribution panel. Figure D–3 shows the EIA-449 connector with the supported signals, their proper signal names, and the mnemonics used in Table D–2. Figure D–4 shows the same for the EIA-530 connector.

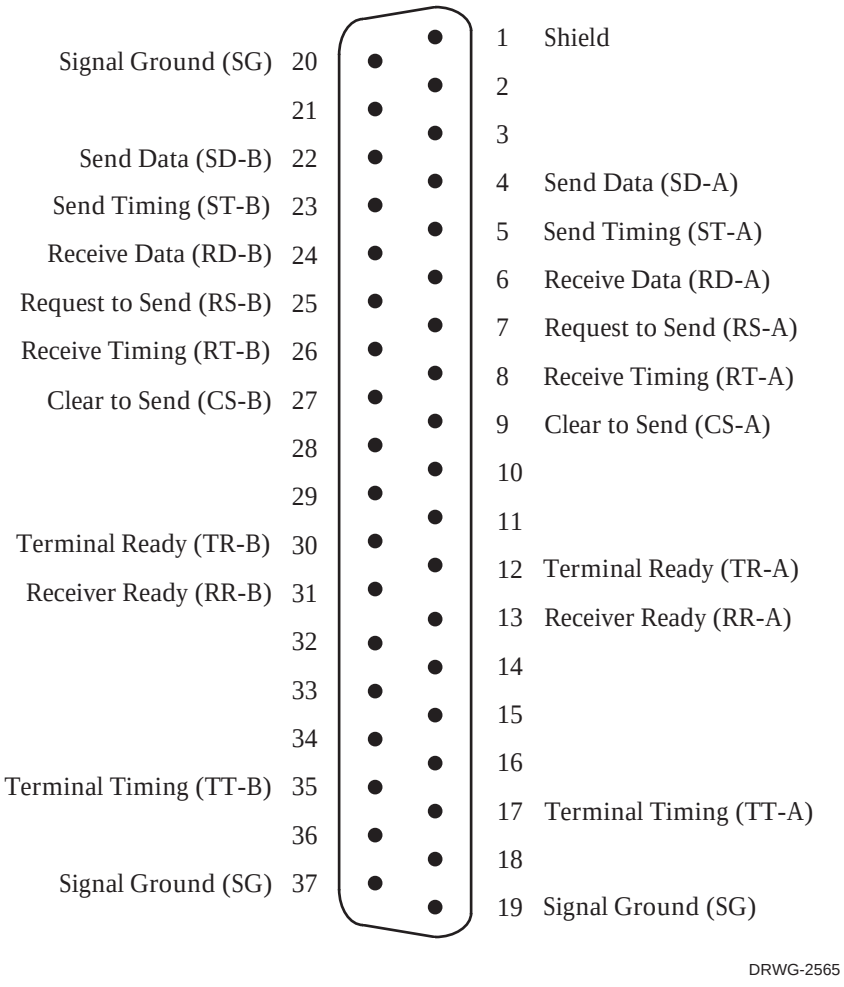


Figure D–3: EIA-449 Interface

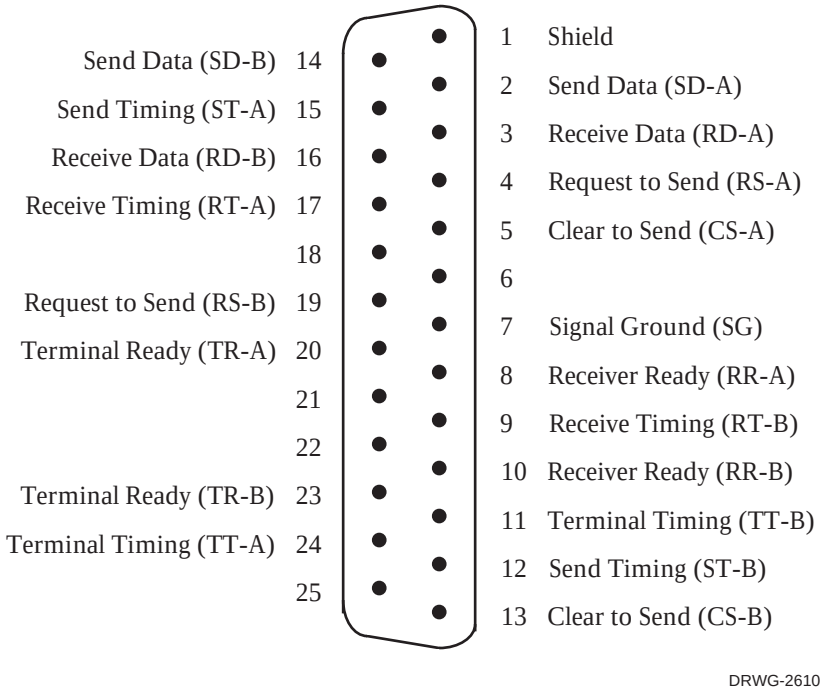


Figure D–4: EIA-530 Interface

Table D–2: Pin Assignments for 8-port EIA-422 Cable Connectors

80-pin Connector	40-pin Connector	Port	Signal	80-pin Connector	40-pin Connector	Port	Signal
A-1	J8-1	00	SG	A-41	J9-1	02	SG
A-2	J8-2	00	SG	A-42	J9-2	02	SG
A-3	J8-3	00	RD-B	A-43	J9-3	02	RD-B
A-4	J8-4	00	RD-A	A-44	J9-4	02	RD-A
A-5	J8-5	00	RT-B	A-45	J9-5	02	RT-B
A-6	J8-6	00	RT-A	A-46	J9-6	02	RT-A
A-7	J8-7	00	CS-B	A-47	J9-7	02	CS-B
A-8	J8-8	00	CS-A	A-48	J9-8	02	CS-A
A-9	J8-9	00	RR-B	A-49	J9-9	02	RR-B
A-10	J8-10	00	RR-A	A-50	J9-10	02	RR-A
A-11	J8-11	00	ST-B	A-51	J9-11	02	ST-B
A-12	J8-12	00	ST-A	A-52	J9-12	02	ST-A
A-13	J8-13	00	SD-B	A-53	J9-13	02	SD-B
A-14	J8-14	00	SD-A	A-54	J9-14	02	SD-A
A-15	J8-15	00	RS-B	A-55	J9-15	02	RS-B
A-16	J8-16	00	RS-A	A-56	J9-16	02	RS-A
A-17	J8-17	00	TR-B	A-57	J9-17	02	TR-B
A-18	J8-18	00	TR-A	A-58	J9-18	02	TR-A
A-19	J8-19	00	TT-B	A-59	J9-19	02	TT-B
A-20	J8-20	00	TT-A	A-60	J9-20	02	TT-A
A-21	J8-21	01	SG	A-61	J9-21	03	SG
A-22	J8-22	01	SG	A-62	J9-22	03	SG
A-23	J8-23	01	RD-B	A-63	J9-23	03	RD-B
A-24	J8-24	01	RD-A	A-64	J9-24	03	RD-A
A-25	J8-25	01	RT-B	A-65	J9-25	03	RT-B
A-26	J8-26	01	RT-A	A-66	J9-26	03	RT-A
A-27	J8-27	01	CS-B	A-67	J9-27	03	CS-B
A-28	J8-28	01	CS-A	A-68	J9-28	03	CS-A
A-29	J8-29	01	RR-B	A-69	J9-29	03	RR-B
A-30	J8-30	01	RR-A	A-70	J9-30	03	RR-A
A-31	J8-31	01	ST-B	A-71	J9-31	03	ST-B
A-32	J8-32	01	ST-A	A-72	J9-32	03	ST-A
A-33	J8-33	01	SD-B	A-73	J9-33	03	SD-B
A-34	J8-34	01	SD-A	A-74	J9-34	03	SD-A
A-35	J8-35	01	RS-B	A-75	J9-35	03	RS-B
A-36	J8-36	01	RS-A	A-76	J9-36	03	RS-A
A-37	J8-37	01	TR-B	A-77	J9-37	03	TR-B
A-38	J8-38	01	TR-A	A-78	J9-38	03	TR-A
A-39	J8-39	01	TT-B	A-79	J9-39	03	TT-B
A-40	J8-40	01	TT-A	A-80	J9-40	03	TT-A

Table D–2: Pin Assignments for 8-port EIA-422 Cable Connectors (*Cont'd*)

80-pin Connector	40-pin Connector	Port	Signal	80-pin Connector	40-pin Connector	Port	Signal
B-1	J8-1	04	SG	B-41	J9-1	06	SG
B-2	J8-2	04	SG	B-42	J9-2	06	SG
B-3	J8-3	04	RD-B	B-43	J9-3	06	RD-B
B-4	J8-4	04	RD-A	B-44	J9-4	06	RD-A
B-5	J8-5	04	RT-B	B-45	J9-5	06	RT-B
B-6	J8-6	04	RT-A	B-46	J9-6	06	RT-A
B-7	J8-7	04	CS-B	B-47	J9-7	06	CS-B
B-8	J8-8	04	CS-A	B-48	J9-8	06	CS-A
B-9	J8-9	04	RR-B	B-49	J9-9	06	RR-B
B-10	J8-10	04	RR-A	B-50	J9-10	06	RR-A
B-11	J8-11	04	ST-B	B-51	J9-11	06	ST-B
B-12	J8-12	04	ST-A	B-52	J9-12	06	ST-A
B-13	J8-13	04	SD-B	B-53	J9-13	06	SD-B
B-14	J8-14	04	SD-A	B-54	J9-14	06	SD-A
B-15	J8-15	04	RS-B	B-55	J9-15	06	RS-B
B-16	J8-16	04	RS-A	B-56	J9-16	06	RS-A
B-17	J8-17	04	TR-B	B-57	J9-17	06	TR-B
B-18	J8-18	04	TR-A	B-58	J9-18	06	TR-A
B-19	J8-19	04	TT-B	B-59	J9-19	06	TT-B
B-20	J8-20	04	TT-A	B-60	J9-20	06	TT-A
B-21	J8-21	05	SG	B-61	J9-21	07	SG
B-22	J8-22	05	SG	B-62	J9-22	07	SG
B-23	J8-23	05	RD-B	B-63	J9-23	07	RD-B
B-24	J8-24	05	RD-A	B-64	J9-24	07	RD-A
B-25	J8-25	05	RT-B	B-65	J9-25	07	RT-B
B-26	J8-26	05	RT-A	B-66	J9-26	07	RT-A
B-27	J8-27	05	CS-B	B-67	J9-27	07	CS-B
B-28	J8-28	05	CS-A	B-68	J9-28	07	CS-A
B-29	J8-29	05	RR-B	B-69	J9-29	07	RR-B
B-30	J8-30	05	RR-A	B-70	J9-30	07	RR-A
B-31	J8-31	05	ST-B	B-71	J9-31	07	ST-B
B-32	J8-32	05	ST-A	B-72	J9-32	07	ST-A
B-33	J8-33	05	SD-B	B-73	J9-33	07	SD-B
B-34	J8-34	05	SD-A	B-74	J9-34	07	SD-A
B-35	J8-35	05	RS-B	B-75	J9-35	07	RS-B
B-36	J8-36	05	RS-A	B-76	J9-36	07	RS-A
B-37	J8-37	05	TR-B	B-77	J9-37	07	TR-B
B-38	J8-38	05	TR-A	B-78	J9-38	07	TR-A
B-39	J8-39	05	TT-B	B-79	J9-39	07	TT-B
B-40	J8-40	05	TT-A	B-80	J9-40	07	TT-A

Index

W

Customer Report Form

We are constantly improving our products. If you have suggestions or problems you would like to report regarding the hardware, software or documentation, please complete this form and mail it to Simpact at 9210 Sky Park Court, San Diego, CA 92123, or fax it to (619) 560-2838.

If you are reporting errors in the documentation, please enter the section and page number.

Your Name: _____

Company: _____

Address: _____

_____

_____

Phone Number: _____

Product: _____

Problem or Suggestion: _____

_____

_____

_____

_____

_____

Simpact, Inc.
Customer Service
9210 Sky Park Court
San Diego, CA 92123