Overpack Sheet:

Technical Manual for the Model 7418A VDA/255A Host Adapter (014–002384–00)

Systech Corporation manufactured the Model 7418-A VDA/255A host adapter for Data General. Accordingly, we have enclosed the manufacturer's document, *HPS VMEbus Advanced Host Adapter (HPS–6245–25) Technical Manual (80–000937–7 Revision A)*. It contains technical information that you may want to refer to from time to time.

If you want to install the VDA/255A host adapter

Use the following manuals to install the VDA/255A host adapter in an AViiON® system or when you want to connect devices to the VDA/255A host adapter:

- *Setting Up and Installing VMEbus Options in AViiON® Systems* (014–001867)
- *Technical Notice: Setting Jumpers on the Model 7418A VDA/255A Host Adapter* (014–002385)

If you want technical help

Please disregard any problem-reporting procedures described in the manufacturer's technical manual. For help with hardware or software problems, contact the Data General Help Line at 1–800–DG–HELPS (U.S. customers only), or contact your Data General representative at the nearest Data General office.

If you want warranty service

The warranties specified in your Data General sales and field engineering contracts supersede any and all warranties described in the manufacturer's documentation. Please note that except for the AViiON 5500 series systems, the AViiON 5000, 6000, 7000, 8000, and 9000 series systems are *not* user serviceable.

WARNING: Data General Corporation supports the maintenance and expansion of the AViiON 5000, 6000, 7000, 8000, and 9000 series systems by DG qualified service providers only. Others who remove, install, or service internal components or options in these systems may cause personal injury, equipment damage, and also void the products' warranties.

For more information regarding warranties, refer to your Data General sales and field engineering contracts. For assistance, contact Data General as described in your system's installing or operating manual.

Ordering No. 014–002384

Printed in the United States of America
Revision 00, September 1994

HPS

VMEbus Advanced Host Adapter

(HPS-6245-25)

Technical Manual

(80-000937-7 Revision A)

HPS
VMEbus Advanced Host Adapter
(HPS-6245-25)
Technical Manual

United States Patent Number 4,845,609

Change Record Page

Manual Part No. 80-000937-7

Date	Rev	By	Description	Pages Affected
1/15/92	A	ELB	Initial production release.	All

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

This manual describes in detail the operation of SYSTECH Corporation's VMEbus High-Performance Serial (HPS) host adapter, specifically the model HPS-6245-25.

The HPS-6245-25 was designed to comply with the requirements of *The VMEbus Specification, Revision C.1* (also known as IEC 821 BUS and IEEE P1014/D1.2).

1.1 GENERAL PRODUCT DESCRIPTION

The HPS-6245-25 is part of a distributed communications subsystem developed by SYSTECH for handling terminals, printers, and similar devices. Used in conjunction with SYSTECH's "cluster" controllers, this HPS host adapter handles all serial line discipline issues, releasing the host from the high-interrupt service burdens associated with traditional multi-channel serial interfaces. This innovative scheme conserves host system panel space by not requiring any of this limited space to be taken up with connectors for user terminals or printers. The single coaxial cable connects to one BNC connector on the host panel.

The HPS host adapter can support up to 64 cluster controllers within the bounds of various cabling limitations and options. Use of the SYSTECH Pluriaxial Unplug™ Repeater (SPUR) products greatly simplifies the generation of modified star or tree cabling topologies and, by adding more cable segments, overcomes the loading limitations of 16 nodes per coaxial cable segment.

Depending on the model, cluster controllers can handle up to 8, 16, 24, or 32 serial devices. The cluster controllers can be located up to 1,000 feet away from the host with coaxial cable or up to 3 miles away with fiber optics and SPURs. All signals to and from as many as 256 terminals are carried on a single coaxial cable which "daisy chains" from the HPS host adapter to the multiple cluster controllers. Adding fiber optic SPURs allows network expansion to modified star or tree topologies and to greater distances. (For additional information about SPURs, contact SYSTECH).

Figure 1-1 illustrates a possible serial I/O subsystem configuration using the HPS-6245-25 host adapter. (For additional information about the cluster controllers, refer to an HPS cluster controller technical manual.)

1.2 HARDWARE

A block diagram of the HPS-6245-25 appears as figure 1-2. The "brain" of the HPS host adapter is a 68020 microprocessor with a highly adaptable direct memory access (DMA) scheme. The combination of the high-performance circuitry and the 68020 instruction cycle time provides the power necessary to service a large number of asynchronous channels. The HPS VMEbus host adapter features:

- 1 Mbyte of parity-checked dynamic RAM (DRAM), including 16 Kbytes dual-port RAM
- 64 Kbytes of EPROM
- Memory mapping in VMEbus 24- or 32-bit address space
- Data transfers on 8-, 16-, and 32-bit data paths
- Individual flag interrupts for the host adapter and host processor
- 2.5 Mbit/second token-passing bit serial bus access and data link protocols
- Support of both master and slave bus modes

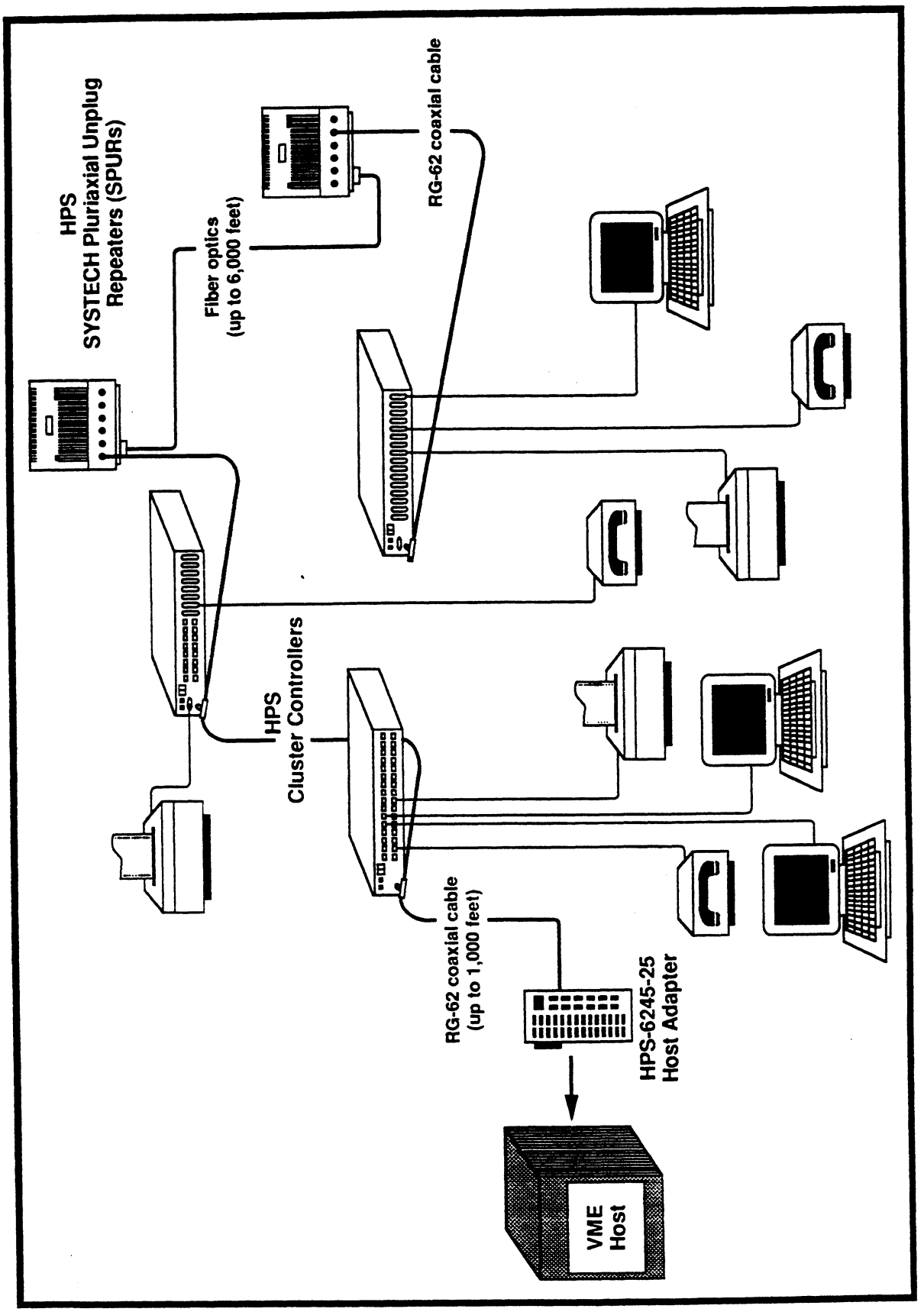


Figure 1-1. HPS Communications Subsystem Overview

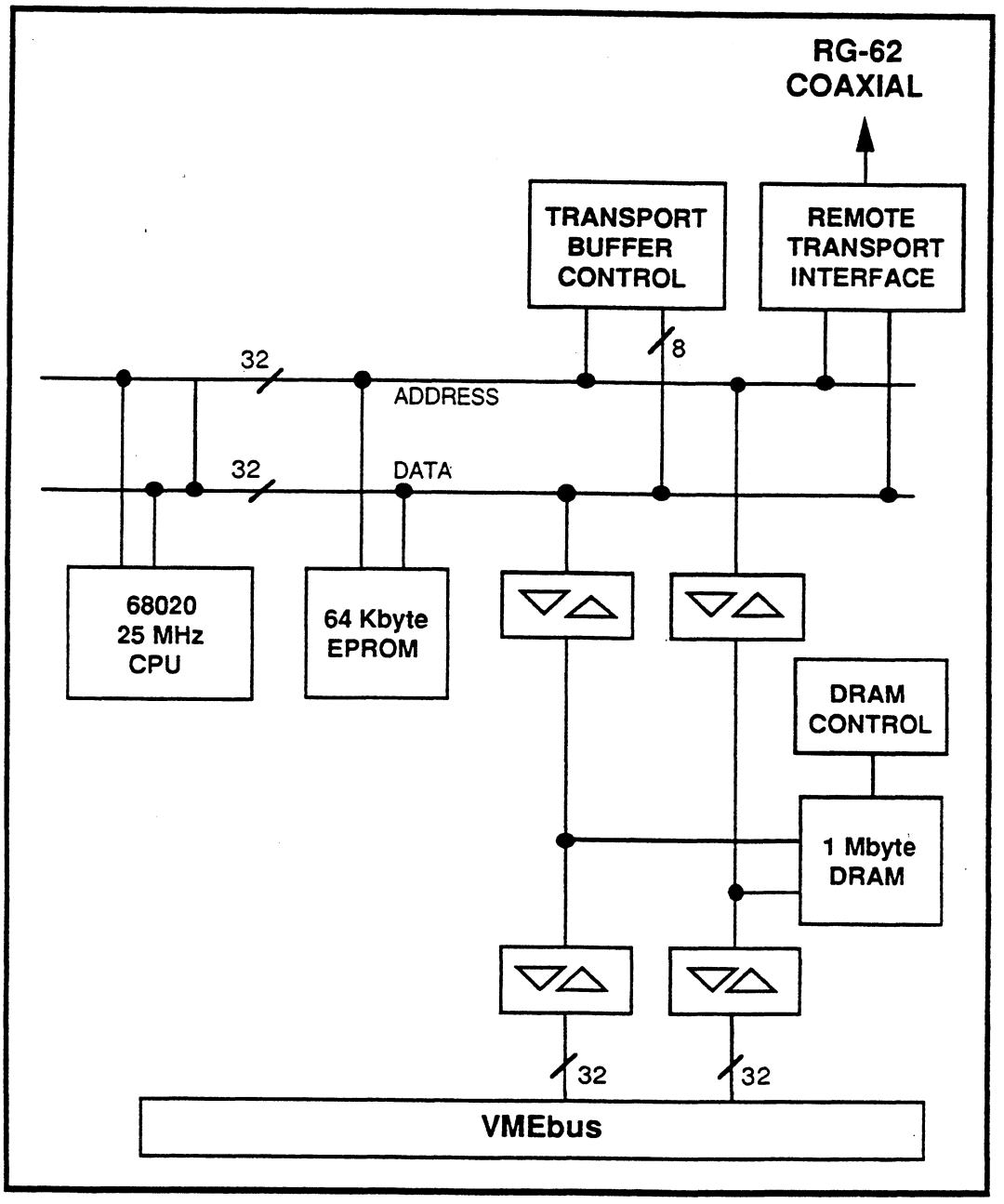


Figure 1-2. HPS-6245-25 Host Adapter Block Diagram

1.3 SOFTWARE

1.3.1 TERMINAL CONTROL SOFTWARE

SYSTECH's Terminal Control Software for the HPS series products allows the HPS host adapter and cluster controller subsystem to appear to the host as a single-board, 256-line asynchronous multiplexer. The software provides all capabilities expected of a high-performance asynchronous multiplexer, plus a set of functions that can be selected to "mix and match" dynamically, on a per-channel basis, to achieve a high degree of preprocessing. This can significantly lighten the terminal-handling burden on the host operating system. For additional information about the Terminal Control software, refer to the *HPS Terminal Control Software User Manual.*

1.3.2 DIAGNOSTICS

SYSTECH-supplied software and firmware for the HPS series products includes a comprehensive set of power-up and in-service diagnostics. During initialization, the HPS host adapter checks its own hardware complement, then conducts a "handshaking" session with the host computer to verify the host-to-HPS data paths. The cluster controllers conduct their own power-up diagnostics, then "log onto" the high-speed serial bus only upon successful completion of their diagnostics. The host adapter includes a tri-color LED which displays diagnostic status.

In-service diagnostics may be invoked by the host system at any time without disturbing on-going terminal I/O. Loopbacks may be run from the host to the HPS host adapter or to one or more cluster controllers. In-service diagnostics may be invoked remotely on host systems equipped for remote diagnosis.

Both the cluster controllers and the host adapters collect usage and error-log statistics to aid in fault isolation and system tuning. Serial bus errors, out-of-memory errors, and invalid command errors are logged for examination by service personnel. The number of characters and packets sent to and received from each cluster controller can be logged on command. These statistics are useful for load balancing and for noting changes in capacity utilization over time.

1.3.3 HPSX

To support the Terminal Control Software, SYSTECH has developed an on-board operating system for the HPS host adapter. This operating system, HPSX, provides a powerful environment for executing board-resident applications. (HPSX is based on Ready Systems Inc.'s Versatile Real-Time Executive for the MC68020 Microprocessor, VRTX32/68020[1].)

The HPSX operating system is a high-performance kernel which provides interrupt-driven, priority-based scheduling; intertask communication and synchronization; dynamic memory allocation; and real-time clock control.

A program loader provided as part of the HPSX software permits you to quickly download SYSTECH-supplied software during system initialization.

[1] VRTX32, VRTX32/68020, RTscope, and RTscope 68000/10/20 are trademarks of Ready Systems.

1.3.4 RTscope 68000/10/20

Optional debugging capability is provided through RTscope, a companion to the HPSX kernel. RTscope, which operates independently of HPSX, supports the display of kernel data structures, multiple breakpoints, memory and register display, execution control (including single-step operation), and program downloading (via a serial port) in S-record or Intel hexadecimal format. (For additional information about the HPSX/RTscope package, refer to SYSTECH's *HPS Application Installation Manual.* For specific information about RTscope, refer to the *RTscope 68000/10/20 Real-Time Debugger and VRTX32 System Monitor for Motorola 68000 Family User's Guide* (available from SYSTECH).

RTscope is provided as an installation option on the SYSTECH HPS-1000 expansion interface port board. With the RTscope option installed, the board becomes the HPS-1000-RTS board (the last part of the name denotes the installed option). The HPS-1000-RTS connects to the HPS-6245-25 to provide the debugging capability.

1.4 FCC INFORMATION

The Federal Communications Commission (FCC) has adopted rules (47 CFR 15) which impose limits on electromagnetic interference and radio frequency interference emanating from digital electronic equipment. Because individual printed circuit boards are considered to be a subassembly of a completed mainframe system, final compliance with these regulations is the responsibility of the mainframe system manufacturer or system integrator.

When properly installed in a configuration the FCC considers to be typical, this SYSTECH product was

> . . . tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his [sic] own expense (FCC 47 CFR Ch. 1 (10-1-89 Edition)).

To ensure compliance with these rules, install this product in a properly shielded enclosure and connect it to its intended peripherals with properly shielded (and terminated) cables.

1.5 SPECIFICATIONS

Refer to table 1-1 for the HPS-6245-25 specifications.

Table 1-1. HPS-6245-25 Host Adapter Specifications

Parameter	Specification
PCB Dimensions	6.299 in. by 9.187 in. (160 mm by 235.35 mm)
Weight	13.3 oz (375 grams), not including cables
Host system slot requirements	One standard double-high 6U VMEbus slot – 0.76 in. (19.30 mm)
VMEbus interface	The HPS-6245-25 is compatible with the VMEbus revision C.1 specification, also known as IEC 821 BUS and IEEE P1014/01.2 • Master/slave data transfer option: A32:D16, A24:D16, A32:D32, A24:D32 • Requestor option: Any one of R(0), R(1), R(2), or R(3) (STAT), RWD • Interrupt option: Any one of I(1), I(2), I(3), I(4), I(5), I(6), or I(7) (STAT), D08(0) (STAT)
Environment	Operating: 0°C to 50°C Storage: -10°C to +70°C Humidity : 10% to 90% noncondensing
Data transfer method	Direct memory access or dual-port RAM
Microprocessor	High-performance 25 MHz 68020
EPROM memory	One socket for installation of up to 64 Kbytes
Dynamic RAM	1 Mbyte of parity protected memory, including 16 Kbytes of dual-port RAM

Table 1-1. HPS-6245-25 Host Adapter Specifications (Continued)

Parameter	Specification
Transport protocol	Token-passing bit serial bus system
Transport interface	2.5 Mbit, high-impedance, hubless transceiver capable of driving fifteen cluster controllers
Transport medium	RG-62, 93 ohm coaxial cable terminated at each end with 93 ohm passive terminators
Maximum transport length	1000 foot maximum network length without the use of repeaters
Indicators	Tri-color status LED
Power requirements	HPS-6245-25 board: • +5V ±5% at 6A maximum • -12V ±5% at 125 mA maximum HPS-1000-RTS board: • +5V ±5% at 500 mA maximum • ±12V ±5% at 50 mA maximum

Chapter 2
Configuration and Installation

This section describes how to configure the HPS host adapter's jumper-selectable options, and how to install the board. The jumpers are discussed separately throughout this chapter, however, a numerical order jumper listing is provided as a quick reference.

2.1 SELF-TEST JUMPER SETTINGS

The jumper **E22** installation configuration is read by the software and used by the HPSX operating system PROM-resident self-test firmware. Refer to Figure 2-1 for the location of jumper **E22** and to chapter 4, "Self-Test Operation," for more detail about the tests identified below.

Jumper E22:

pins 1-2 - reserved for factory use. Do not install these pins.
out* = correct setting
in = incorrect setting

pins 3-4 - out* = perform the "host primitive read/write" portion of the self-test.
in = do not perform this portion of the self-test.

* Factory default setting.

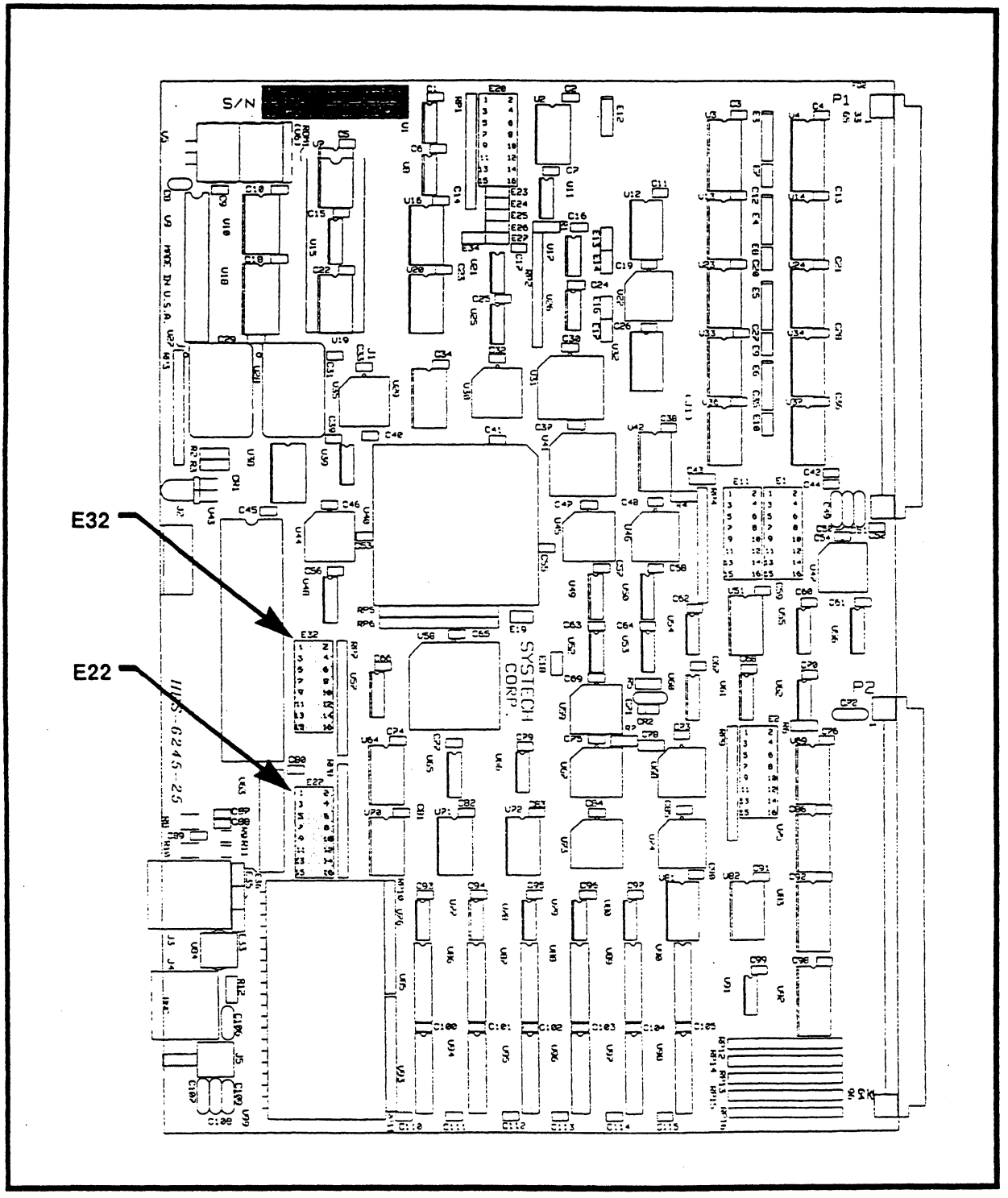


Figure 2-1. Jumpers E22 and E32

pins 7-8	-	out*	= perform all stack DRAM tests. Clear RAM at end of self-test regardless of whether pins 9-10 are installed.
		in	= do not perform this portion of the self-test.
pins 9-10	-	(If pins 7-8 are out, RAM is cleared regardless of whether pins 9-10 are installed.)	
		out*	= Write all RAM locations with 00 at end of self-test. This effectively clears RAM.
		in	= Do not clear RAM at end of self-test.
pins 13-14	-	out*	= All even-aligned long words (32 bit) will be transferred on the VMEbus as two 16-bit words.
		in	= All even-aligned long words will be transferred as 32-bit long words.

* Factory default setting.

2.1.1 JUMPER E22 SPARE POSITIONS

The following jumper **E22** configurations are currently unused by SYSTECH software and should not be installed.

Jumper E22:

pins 5-6	-	out	= reserved for future use
pins 11-12	-	out	= reserved for future use
pins 15-16	-	out	= reserved for future use

2.1.2 JUMPER E22 REGISTER

The jumper **E22** register is a byte register which provides the status read from jumper **E22**. The register reads a zero for an installed configuration. The bit/pin installation cross reference appears as table 2-1.

Table 2-1. Jumper E22 Register

CPU Data Bit Set	Jumper E22 Configuration
bit 24	pins 1-2
bit 25	pins 3-4
bit 26	pins 5-6
bit 27	pins 7-8
bit 28	pins 9-10
bit 29	pins 11-12
bit 30	pins 13-14
bit 31	pins 15-16

2.2 SETTING THE TRANSPORT NODE ADDRESS

Jumper block **E32** (figure 2-1) sets the HPS host adapter transport node address to which it will respond on the network. This block is read into the COM 9026 controller chip whenever the host adapter is either powered on or the token-passing network reconfigures. The host adapter's node address can be set to any address from 01H[1] to 0FFH (1 to 255). Note that zero (00H) is not a valid node address. The HPS self-test will fail if 00H is selected.

[1] Throughout this manual, "H" is used as the abbreviation for "hexadecimal."

Each HPS host adapter and cluster controller on the same network must be set to a unique, nonconflicting node address. SYSTECH recommends that the HPS host adapter address be configured as 0FFH (255) and that the cluster controllers start at address 1 and increase sequentially to 0FH (15).

Table 2-2 is an example of how to set the jumper for a node address of 071H. Note that an installed jumper indicates a "1" and a removed jumper represents a "0."

Table 2-2. Sample Jumper Setting for 071H Node Address

Bit	E32 Pins	Position	Binary	Hexadecimal
Bit 7	1-2	out	0	
Bit 6	3-4	in	1	7
Bit 5	5-6	in	1	
Bit 4	7-8	in	1	
Bit 3	9-10	out	0	
Bit 2	11-12	out	0	1
Bit 1	13-14	out	0	
Bit 0	15-16	in	1	

2.3 VMEbus CONFIGURATION SETTINGS

The HPS host adapter is easily adapted to a variety of different host system configurations. The following sections describe how to configure the board to cover such issues as address modifiers, use of dual-port memory, interrupt level, device address, and a number of other issues.

2.3.1 VMEbus ARBITRATION JUMPERS— BUS GRANT AND BUS REQUEST

Jumpers on the HPS-6245-25 allow you to configure VMEbus interface bus request signals *BR0-BR3*, and bus grant signals *BG0IN-BG3IN* and *BG0OUT-BG3OUT*.

These interface signals are used for requesting and obtaining the bus. The VMEbus has four request and grant levels, as reflected by the numbers in the signal names. Zero is the lowest priority and three is the highest. Board position within the VMEbus chassis also plays a role in determining priority. Within each priority level, the board located closest to the system controller card has priority over the other boards at the same request level. To configure the arbitration level, determine the level required for the system. Select the bus request level by inserting the appropriate jumper, as indicated in table 2-3 (the location of these jumpers is shown in figure 2-2). Install only one of the jumpers listed in the table. For systems with a single-level arbiter, select bus request level 3 on the HPS host adapter.

Table 2-3. Bus Request Level Selection

Bus Request Level	Jumper
3*	E10
2	E9
1	E8
0	E7
*Factory default setting	

The bus grant (*BGxIN* and *BGxOUT*) level must correspond to the bus request level. There are four jumper blocks that select the bus grant level, one for each level. These jumper blocks are listed in table 2-4 and identified on figure 2-2. Install a jumper between pin 1 and pin 2, and between pin 3 and pin 4 of the jumper block that corresponds to the bus grant level you wish to use. Install a jumper between pin 2 and pin 3 on the rest of the jumper blocks. On figure 2-2, the dark squares on jumper blocks **E3** through **E6** identify the pin 1 positions.

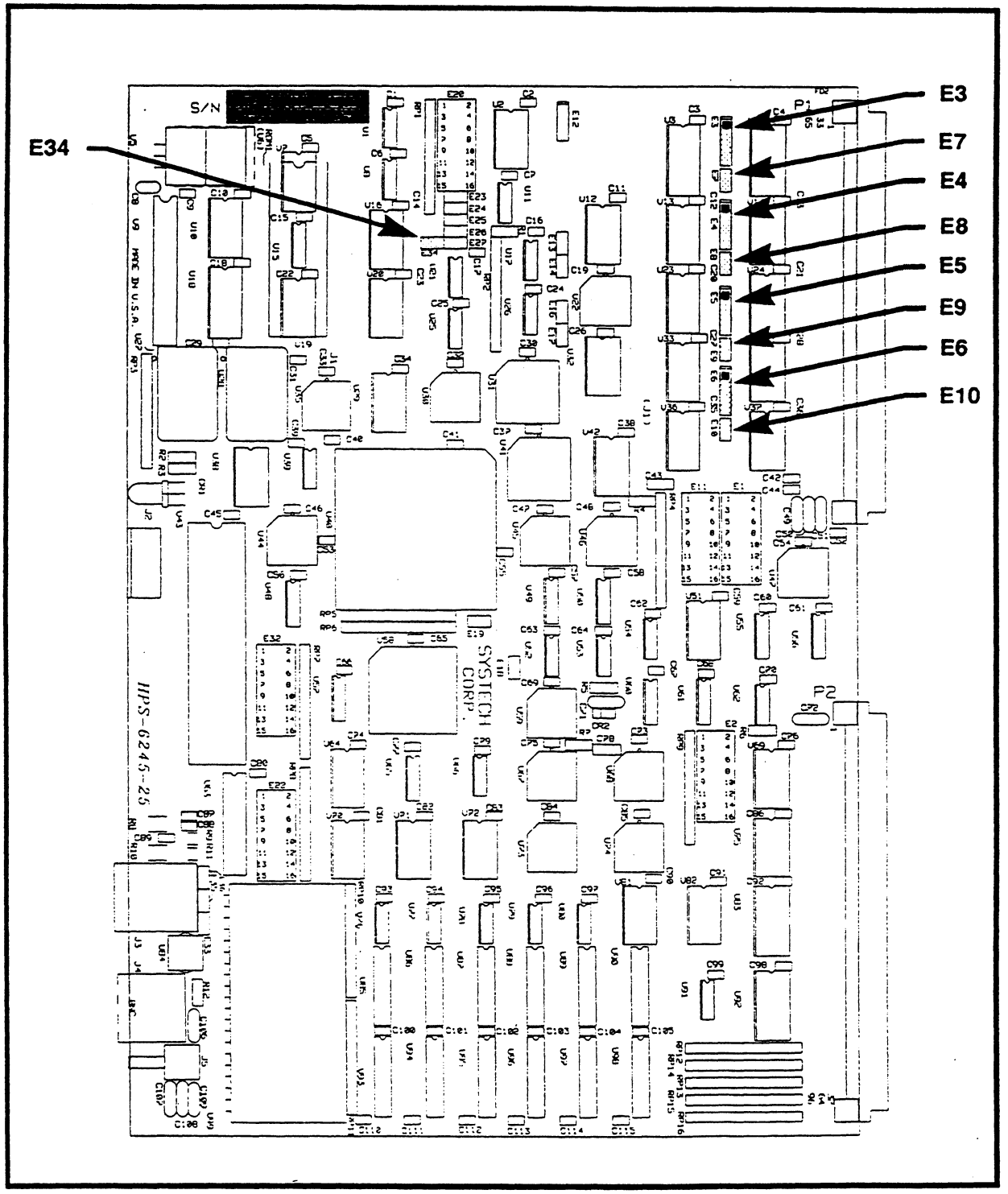
NOTE

If your system generates false *BGxIN* signals, install jumper **E34** to ensure that false signals are ignored. If jumper **E34** is installed, the *BGxIN* signal is sampled twice before it is acted upon.

Table 2-4. Bus Grant Level Selection

Bus Grant Level	Jumper Block Number
3*	E6
2	E5
1	E4
0	E3

* The HPS-6245-25 is factory configured for bus grant level 3 (jumper E6, pins 1-2 and 3-4 in, and jumpers E3, E4, and E5, pins 2-3 in).



On jumpers containing more than two pins, pin 1 is indicated by a square.

Figure 2-2. VMEbus Arbitration Jumpers

2.3.2 SLAVE ADDRESS JUMPERS

2.3.2.1 Slave Address Modifier

The HPS' dual-port memory appears as a slave to the VMEbus. The jumpers listed in table 2-5 (and identified on figure 2-3), determine dual-port memory placement in the address spaces noted. Table 2-6 defines the jumper installation for each possible memory configuration. To select extended address space, refer to section 2.3.2.2 for information about how to enable decoding of address bits A31 through A24.

Table 2-5. Dual-Port Memory Jumpers

Location	Jumper
Data space	E14*
Program space	E13*
Supervisor space	E17*
Standard space	E16*

* Factory installed.

Table 2-6. Dual-Port Memory Jumper Configuration

	Jumper Installation			
Function	E16	E17	E13	E14
Extended, supervisor, program	out	in	in	out
Extended, supervisor, data	out	in	out	in
Extended, supervisor, data or program	out	in	in	in
Extended, user, program	out	out	in	out
Extended, user, data	out	out	out	in
Extended, user, data or program	out	out	in	in
Standard, supervisor, program	in	in	in	out
Standard, supervisor, data	in	in	out	in
Standard, supervisor, data or program	in	in	in	in
Standard, user, program	in	out	in	out
Standard, user, data	in	out	out	in
Standard, user, data or program	in	out	in	in

2.3.2.2 Slave Address Base

Table 2-7 lists the address jumpers used to select the base address of the HPS-6245-25 as viewed by tne VMEbus system (these jumpers are identified on figure 2-3). "In" indicates a "0," and "out" indicates a "1." Note that jumper **E2** connections are only valid in extended (32-bit) addressing mode, and that for extended addressing mode to be enabled, jumper **E16** must be "out" (i.e., not installed).

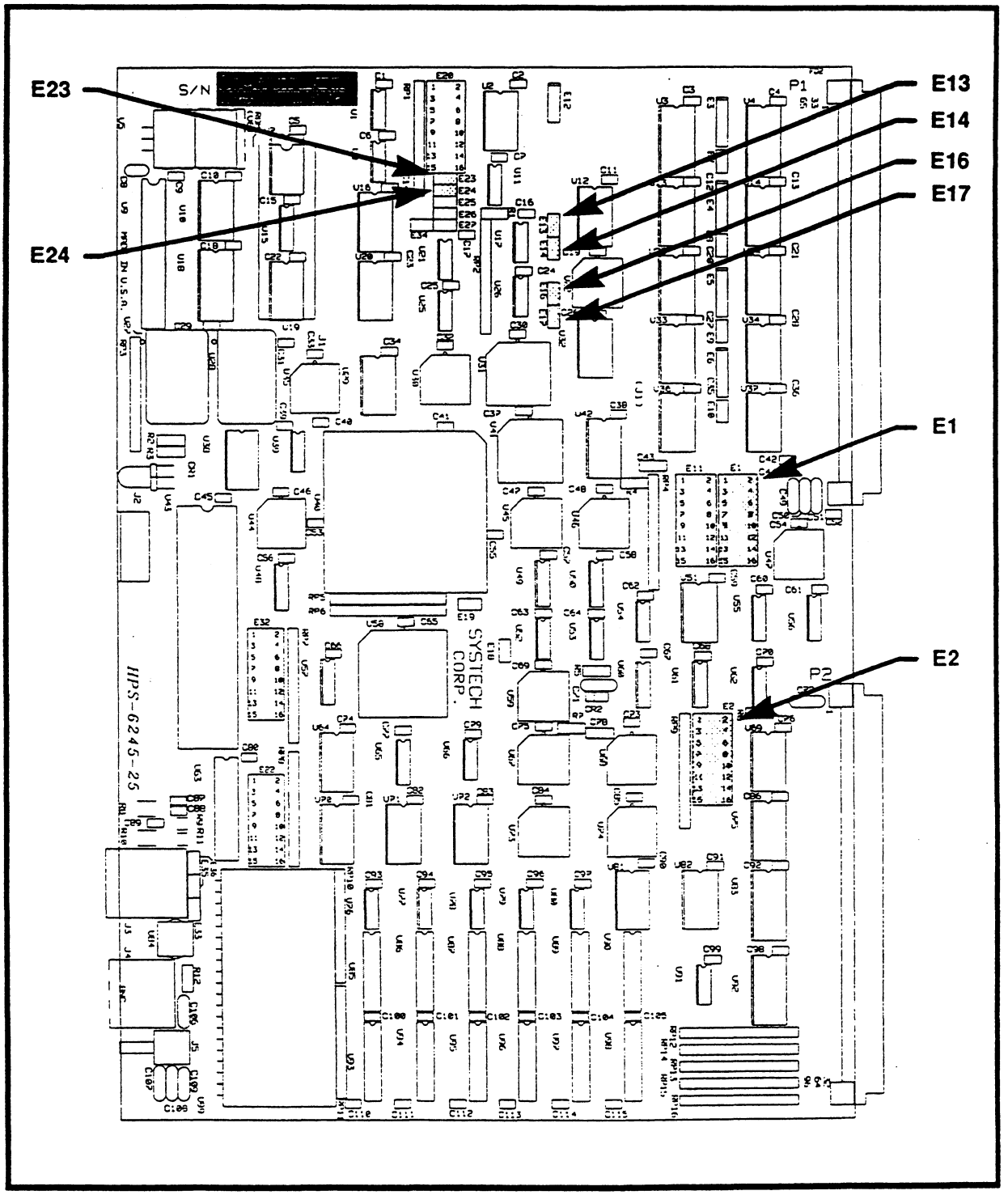


Figure 2-3. Slave Address Jumpers

Table 2-7. Base Address Memory Configuration

Address Bit	Jumper Pins
A31	E2, pins 1–2
A30	E2, pins 3–4
A29	E2, pins 5–6
A28	E2, pins 7–8
A27	E2, pins 9–10
A26	E2, pins 11–12
A25	E2, pins 13–14
A24	E2, pins 15–16
E2 connections are valid in extended (32-bit) addressing mode only.	
A23	E1, pins 1–2*
A22	E1, pins 3–4*
A21	E1, pins 5–6*
A20	E1, pins 7–8*
A19	E1, pins 9–10
A18	E1, pins 11–12
A17	E1, pins 13–14
A16	E1, pins 15–16
A15	E23*
A14	E24*
* Factory installed; configures the HPS-6245-25 for a base address of 0F0000H.	

2.3.3 VMEbus INTERRUPT CONFIGURATION

The HPS host adapter can be jumper configured to present an interrupt request on any of the bus request levels, *IRQ7** through *IRQ1** as shown in table 2-8. These jumpers are identified on figure 2-4.

Select one of the interrupt levels, then make sure that the jumpers are all configured for the same level.

Table 2-8. Interrupt Level Jumpers

Interrupt Level	Request Jumper E11 Pins	Acknowledgement Jumper		
		E25	E26	E27
IRQ7*	13-14	out	out	out
IRQ6*	11-12	out	out	in
IRQ5*	9-10	out	in	out
IRQ4*	7-8	out	in	in
IRQ3*	5-6	in	out	out
IRQ2*	3-4	in	out	in
IRQ1*	1-2	in	in	out

Note that only one pair of pins on request jumper E11 may be installed at one time.

Pins 15 and 16 are unused.

IRQ2* is factory configured as shown in this table.

When the interrupt is acknowledged, the HPS host adapter presents a STATUS/ID byte. This byte is typically used as an interrupt vector. You can select the value of this byte by installing jumper **E20** as indicated in table 2-9. Any combination of these pin installation configurations may be used. An installed jumper indicates a "1," and a removed jumper represents a "0." The location of these jumpers is shown on figure 2-4.

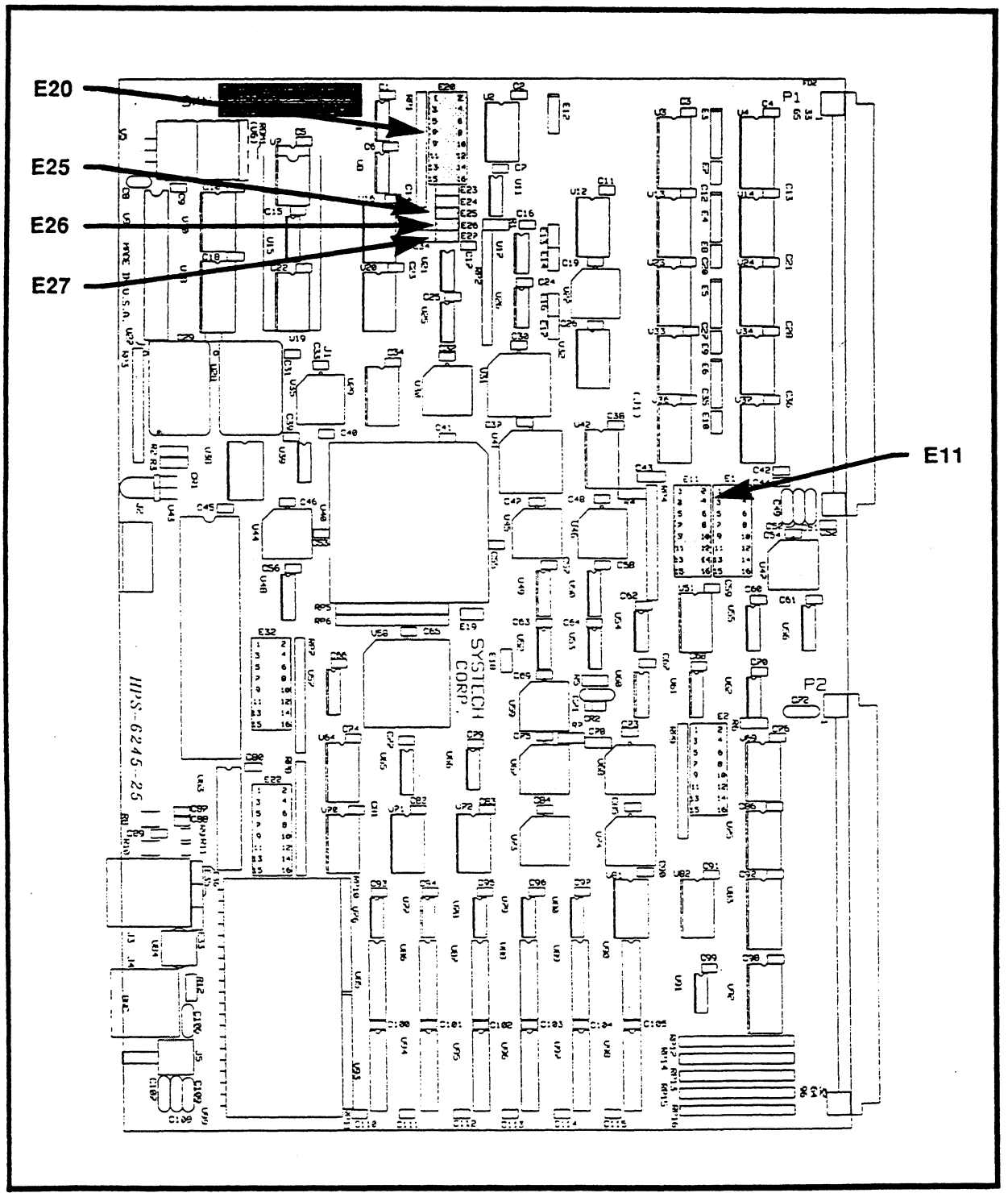


Figure 2-4. VMEbus Interrupt Jumpers

Table 2-9. STATUS/ID Byte Jumpers

STATUS/ID Byte Bit	Jumper E20
D7	pins 15-16
D6*	pins 13-14
D5	pins 11-12
D4	pins 9-10
D3	pins 7-8
D2	pins 5-6
D1	pins 3-4
D0	pins 1-2
* Factory installed.	

2.4 EPROM SIZE CONFIGURATION

The HPS-6245-25 is equipped with one EPROM socket location. You can configure this location to accept industry standard 256K or 512K EPROMs. The EPROMs are configured for byte mode operation. The host adapter comes equipped with 150 ns EPROM devices. EPROMs of this speed or faster are required for proper operation. Table 2-10 explains how to configure jumper **E12** to support either a 256K or 512K EPROM. The jumper location is shown on figure 2-5; the dark square shown on jumper **E12** identifies the pin 1 position.

Table 2-10. EPROM Jumper

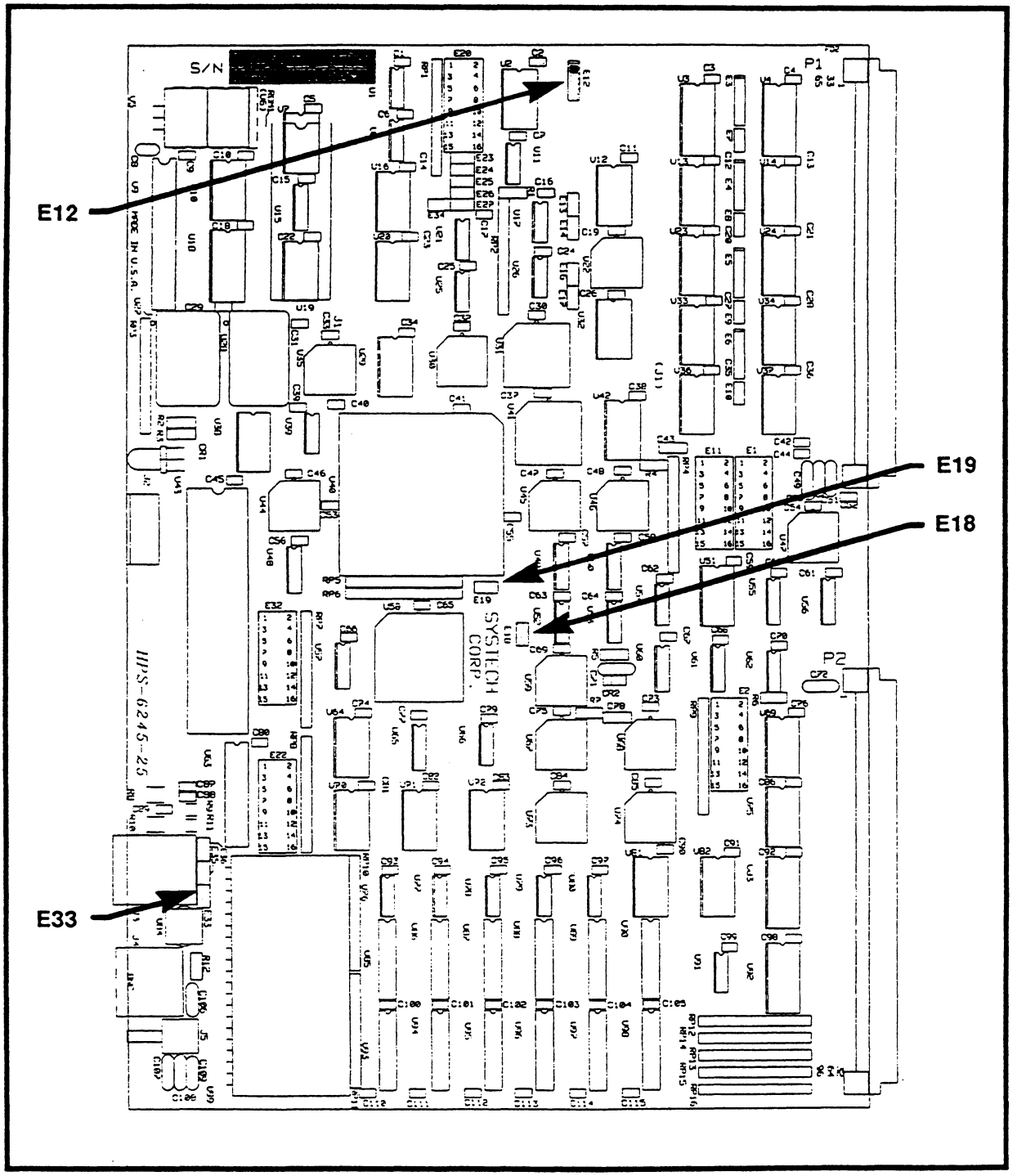
Device (U50)	Jumper E12	Capacity
27256	2 to 3	32K bytes
27512	1 to 2	64K bytes
The HPS-6245-25 is factory equipped with a 27256 EPROM containing HPSX. The RTscope option is provided on a SYSTECH HPS-1000 expansion interface board.		

2.5 DYNAMIC RAM CONFIGURATION

The HPS-6245-25 contains eight 1Mbit dynamic RAM (DRAM) devices plus four 256K-bit DRAMs for parity. This results in a maximum memory capacity of 1M byte, and includes full parity circuitry. There are no jumpers required for configuring DRAM.

2.6 REAL-TIME CLOCK

The real-time clock, which is controlled by an ASIC, produces the watchdog timeout by providing a simple periodic interrupt source for the microprocessor. The default power-up value is 52 ms. One of eight timeout periods—from 819 µs to 104 ms—may be selected through software. The real-time clock resides at address 600000H.



On jumpers containing more than two pins, pin 1 is indicated by a square.

Figure 2-5. EPROM, Line Termination, Halt, and Watchdog Timer Jumpers

2.7 LINE TERMINATION

You must terminate the coaxial line leading from the HPS host adapter to each of the HPS cluster controllers at both the beginning and end of the line. The end is terminated with a BNC terminator at the last cluster controller on the line (refer to the cluster controller installation guide for specific cabling and termination instructions). To terminate the beginning of the line at the host adapter, jumper **E33** must be installed. **E33**, which connects a terminating resistor, is factory installed. Refer to figure 2-5 for the location of this jumper.

2.8 HALT

Jumper **E19** (figure 2-5) determines whether a microprocessor halt will activate the bus *SYSFAIL** line. When **E19** is installed, a host adapter halt will activate *SYSFAIL**. When **E19** is removed, *SYSFAIL** will not be affected by a host adapter halt. Jumper **E19** is factory installed.

2.9 WATCHDOG TIMER

Jumper **E18** (figure 2-5) allows you to disable the watchdog timer while debugging the HPS-6245-25 board. Remove jumper **E18** to disable the watchdog timer, however, be sure to reinstall the jumper before operating the board to ensure that it passes self-test. Jumper **E18** is factory installed.

NOTES

Jumper E18, the watchdog timer jumper, must be installed for the board to pass self-test.

The watchdog timer is disabled if RTscope is installed.

2.10 NUMERICAL ORDER JUMPER LISTING

For reference, table 2-11 lists each jumper on the HPS host adapter board by number, along with a brief explanation of its function and a notation of the section number in which it was discussed.

Table 2-11. Numerical Order Jumper Listing

Jumper	Factory Installed Position	Function	Section
E1	pins 1-2, in	in: Specifies bit A23 of the HPS base address as viewed by the VMEbus system.	2.3.2.2
	pins 3-4, in	in: Specifies bit A22 of the HPS base address.	
	pins 5-6, in	in: Specifies bit A21 of the HPS base address.	
	pins 7-8, in	in: Specifies bit A20 of the HPS base address.	
	pins 9-10, out	in: Specifies bit A19 of the HPS base address.	
	pins 11-12, out	in: Specifies bit A18 of the HPS base address.	
	pins 13-14, out	in: Specifies bit A17 of the HPS base address.	
	pins 15-16, out	in: Specifies bit A16 of the HPS base address.	
E2	pins 1-2, out	in: Specifies bit A31 of the HPS base address as viewed by the VMEbus system.	2.3.2.2
	pins 3-4, out	in: Specifies bit A30 of the HPS base address.	
	pins 5-6, out	in: Specifies bit A29 of the HPS base address.	
	pins 7-8, out	in: Specifies bit A28 of the HPS base address.	
	pins 9-10, out	in: Specifies bit A27 of the HPS base address.	
	pins 11-12, out	in: Specifies bit A26 of the HPS base address.	
	pins 13-14, out	in: Specifies bit A25 of the HPS base address.	
	pins 15-16, out	in: Specifies bit A24 of the HPS base address.	
E3	pins 2-3, in	Bus grant level 0 primary jumper; configure for this level by connecting pins 1-2 and 3-4 of E3, then pins 2-3 of E4, E5, and E6.	2.3.1

Table 2-11. Numerical Order Jumper Listing (Continued)

Jumper	Factory Installed Position	Function	Section
E4	pins 2-3, in	Bus grant level 1 primary jumper; configure for this level by connecting pins 1-2 and 3-4 of E4, then pins 2-3 of E3, E5, and E6.	2.3.1
E5	pins 2-3, in	Bus grant level 2 primary jumper; configure for this level by connecting pins 1-2 and 3-4 of E5, then pins 2-3 of E3, E4, and E6.	2.3.1
E6	pins 1-2 , 3-4, in	Bus grant level 3 primary jumper; configure for this level by connecting pins 1-2 and 3-4 of E6, then pins 2-3 of E3, E4, and E5.	2.3.1
E7	out	in: Selects bus request level 0.	2.3.1
E8	out	in: Selects bus request level 1.	2.3.1
E9	out	in: Selects bus request level 2.	2.3.1
E10	in	in: Selects bus request level 3.	2.3.1
E11	pins 1-2, out	in: With E27 out and E25 and E26 in, configures for interrupt request level IRQ1*.	2.3.3
	pins 3-4, in	in: With E25 and E27 in and E26 out, configures for interrupt request level IRQ2*.	
	pins 5-6, out	in: With E25 in and E26 and E27 out, configures for interrupt request level IRQ3*.	
	pins 7-8, out	in: With E26 and E27 in and E25 out, configures for interrupt request level IRQ4*.	
	pins 9-10, out	in: With E25 and E27 out and E26 in, configures for interrupt request level IRQ5*.	
	pins 11-12, out	in: With E25 and E26 out and E27 in, configures for interrupt request level IRQ6*.	
	pins 13-14, out	in: With E25, E26, and E27 out, configures for interrupt request level IRQ7*.	
	pins 15-16	unused	

Table 2-11. Numerical Order Jumper Listing (Continued)

Jumper	Factory Installed Position	Function	Section
E12	pins 1-2, out pins 2-3, in	in: 27512 EPROM. in: 27256 EPROM.	2.4
E13	in	in: Places DPRAM in program space.	2.3.2.1
E14	in	in: Places DPRAM in data space.	2.3.2.1
E16	in	in: Places DPRAM in standard space. Note that E16 must be "out" for extended addressing mode to be enabled.	2.3.2.2
E17	in	in: Places DPRAM in supervisor space. out: Places DPRAM in nonprivileged space.	2.3.2.1
E18	in	in: Enables the watchdog timer.	2.9
E19	in	in: Allows a host adapter halt to activate *SYSFAIL**. out: Prevents *SYSFAIL** from being affected by a host adapter halt.	2.8
E20	pins 1-2, out pins 3-4, out pins 5-6, out pins 7-8, out pins 9-10, out pins 11-12, out pins 13-14, in pins 15-16, out	in: Selects STATUS/ID byte bit D0. in: Selects STATUS/ID byte bit D1. in: Selects STATUS/ID byte bit D2. in: Selects STATUS/ID byte bit D3. in: Selects STATUS/ID byte bit D4. in: Selects STATUS/ID byte bit D5. in: Selects STATUS/ID byte bit D6. in: Selects STATUS/ID byte bit D7.	2.3.3

Table 2-11. Numerical Order Jumper Listing (Continued)

Jumper	Factory Installed Position	Function	Section
E22	pins 1-2, out	reserved for factory use.	2.1
	pins 3-4, out	in: "Host primitive read/write" portion of self-test not performed.	
	pins 5-6, out	reserved for future use.	
	pins 7-8, out	in: Stack and RAM portions of self-test not performed.	
	pins 9-10, out	in: RAM is not cleared at end of self-test. Note that if pins 7-8 are not installed, RAM is cleared regardless of whether pins 9-10 are installed.	
	pins 11-12, out	reserved for future use.	
	pins 13-14, out	in: All even-aligned long words are transferred as 32-bit long words.	
	pins 15-16, out	reserved for future use.	
E23	in	in: Specifies bit A15 of the HPS base address.	2.3.2.2
E24	in	in: Specifies bit A14 of the HPS base address.	2.3.2.2
E25	in	Used in combination with request jumper E11 and acknowledgement jumpers E26 and E27, determines the interrupt acknowledgment level.	2.3.3
E26	out	Used in combination with request jumper E11 and acknowledgement jumpers E25 and E27, determines the interrupt acknowledgment level.	2.3.3
E27	in	Used in combination with request jumper E11 and acknowledgement jumpers E25 and E26, determines the interrupt acknowledgment level.	2.3.3

Table 2-11. Numerical Order Jumper Listing (Continued)

Jumper	Factory Installed Position	Function	Section
E32	pins 1-2, in pins 3-4, in pins 5-6, in pins 7-8, in pins 9-10, in pins 11-12, in pins 13-14, in pins 15-16, in	Sets the host adapter transport node address.	2.2
E33	in	in: Terminates the coaxial line at the host adapter.	2.7
E34	out	in: Samples *BGxIN* signal twice to prevent false bus grant.	2.3.1
E35	out	in: Reserved for future use; must be "out" for proper operation.	
E36	out	in: Reserved for future use; must be "out" for proper operation.	

2.11 INSTALLATION

This section explains how to install the HPS host adapter board after configuring the jumpers.

2.11.1 MOUNTING IN CAGE

Before installing the HPS host adapter board, **turn off power to the system**. In selecting a slot for the host adapter, priority must be considered. Refer to section 2.3.1 for an explanation of board positioning/priority.

NOTE

To prevent damage to the HPS host adapter board, take proper protection measures to ensure against static discharge (e.g., wearing a properly grounded antistatic wrist strap when handling the board).

To install the board into the card cage:

- Slide the board into the selected VMEbus slot until it seats firmly.

NOTE

When installing an HPS-6245-25 board equipped with a front panel, tighten the captive retainer screw at each end of the front panel. Failure to install these screws may cause radiated emissions to exceed regulatory agency limits.

- On the VMEbus back panel, remove the jumpers that jumper the *BGXIN**, *BGXOUT**, and interrupt acknowledge lines around the selected slot. Simultaneously, check to make sure that these jumpers are installed both on empty slots and on any slot containing a board that doesn't use these signals and doesn't pass them on to the next slot.

2.11.2 CABLE INSTALLATION

After installing the board, attach the appropriate coaxial cable to the connector on the board.

2.11.3 BOARD REMOVAL AND REINSTALLATION

To remove the host adapter board for any reason (i.e., for reconfiguration, shipping, or storage), **first turn off power to the system**. Then disconnect all cables and slide the board out.

NOTE

To operate the system with the HPS host adapter absent, install the *BGXIN**, *BGXOUT**, and interrupt acknowledge jumpers on the system backplane.

2.11.4 HPS-1000-RTS BOARD

The HPS-1000 is an expansion interface board which provides a means of connecting the optional RTscope debugger to the HPS-6245-25 (with the RTscope option installed the interface board is called the HPS-1000-RTS). While the HPS-6245-25 host adapter normally requires one double-high VMEbus slot, with the HPS-1000-RTS board installed, an additional slot is required because of the interface board's added height. For this reason, the HPS-1000-RTS board is designed to be used primarily during initial system integration.

2.11.4.1 Board Installation

The HPS-1000-RTS board is equipped with two rows of pins on the back side; one row of 25 pins, and one row of 20. To connect the HPS-1000-RTS board to the host adapter, line up the pins with the comparable sockets on the HPS-6245-25, then carefully push the interface board straight down, sliding the pins into the matching sockets.

CAUTION

To prevent damage to the boards, ensure that each row of pins lines up with the proper socket before connecting the HPS-1000-RTS board to the host adapter.

2.11.4.2 Cable Installation

A 26-pin RS-232 cable matching the specifications of SYSTECH part number 65-000042-7 (figure 2-6) connects the HPS-1000-RTS board to a terminal. To do this, connect the cable's 26-pin plug to the 26-pin socket on the HPS-1000-RTS board, then connect the other end (terminated with a 25-pin DB-25 connector) to the terminal. Table 2-12 lists the signal pinout information of the HPS-1000-RTS 26-pin header.

Table 2-12. HPS-1000-RTS Pinout List

Pin	Connections
3	TXD
5	RXD
13	Digital ground

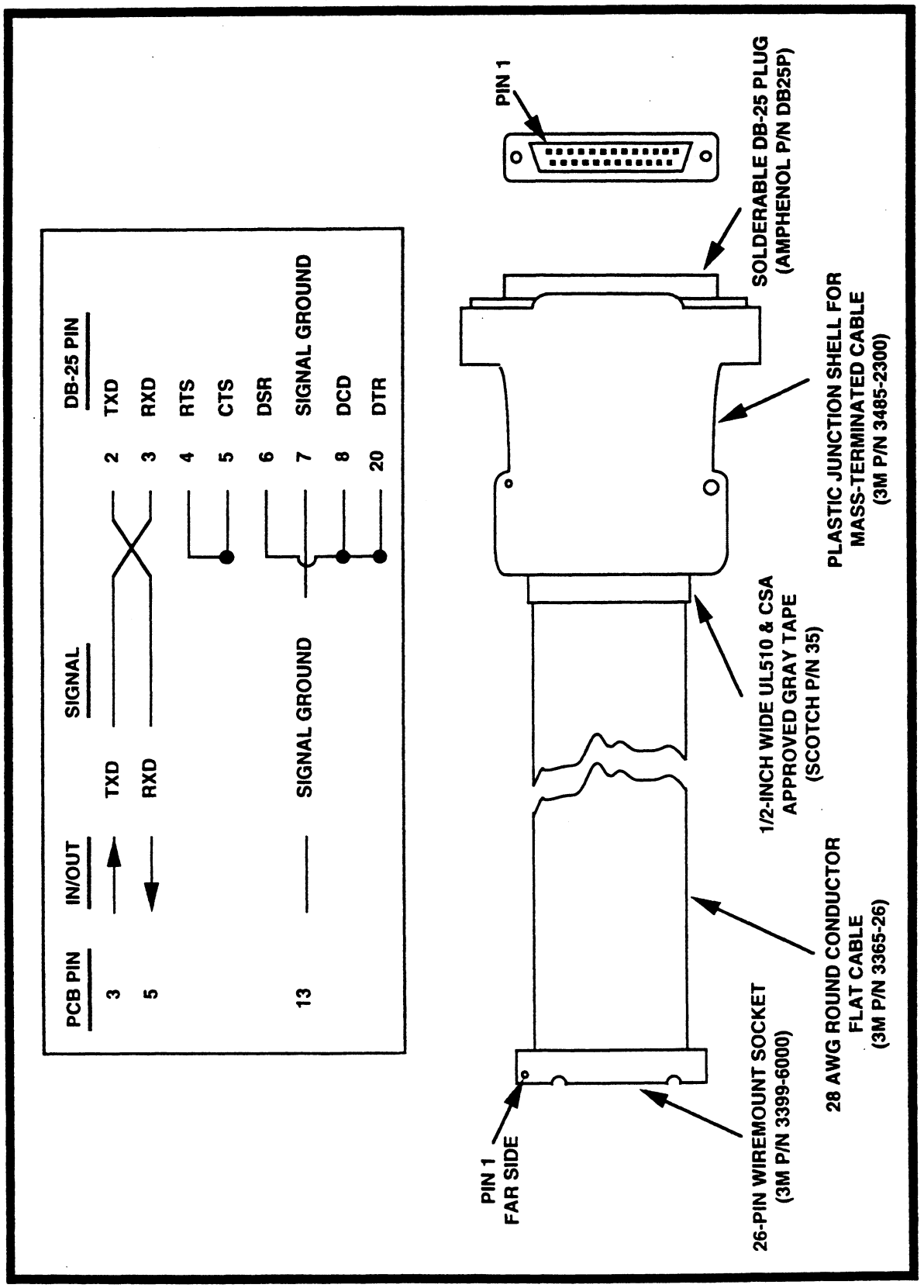


Figure 2-6. RS-232 Cable

Chapter 3
Theory of Operation

This chapter describes the internal electronic operation of the HPS-6245-25 VMEbus host adapter. Refer to the block diagram shown as figure 3-1 for a pictorial representation of the board's component/functional interrelationship.

The HPS-6245-25 operates on the VMEbus and conforms to Revision C.1 of the VMEbus standard. It supports full 32-bit address and 32-bit data.

3.1 GENERAL COMPONENT OVERVIEW

3.1.1 CPU

The HPS-6245-25 uses a 68020 CPU running at 25 MHz. The CPU, which provides a 64-entry instruction cache, is downward compatible with the 68010 used on previous HPS host adapter boards and, therefore, will run the existing code base.

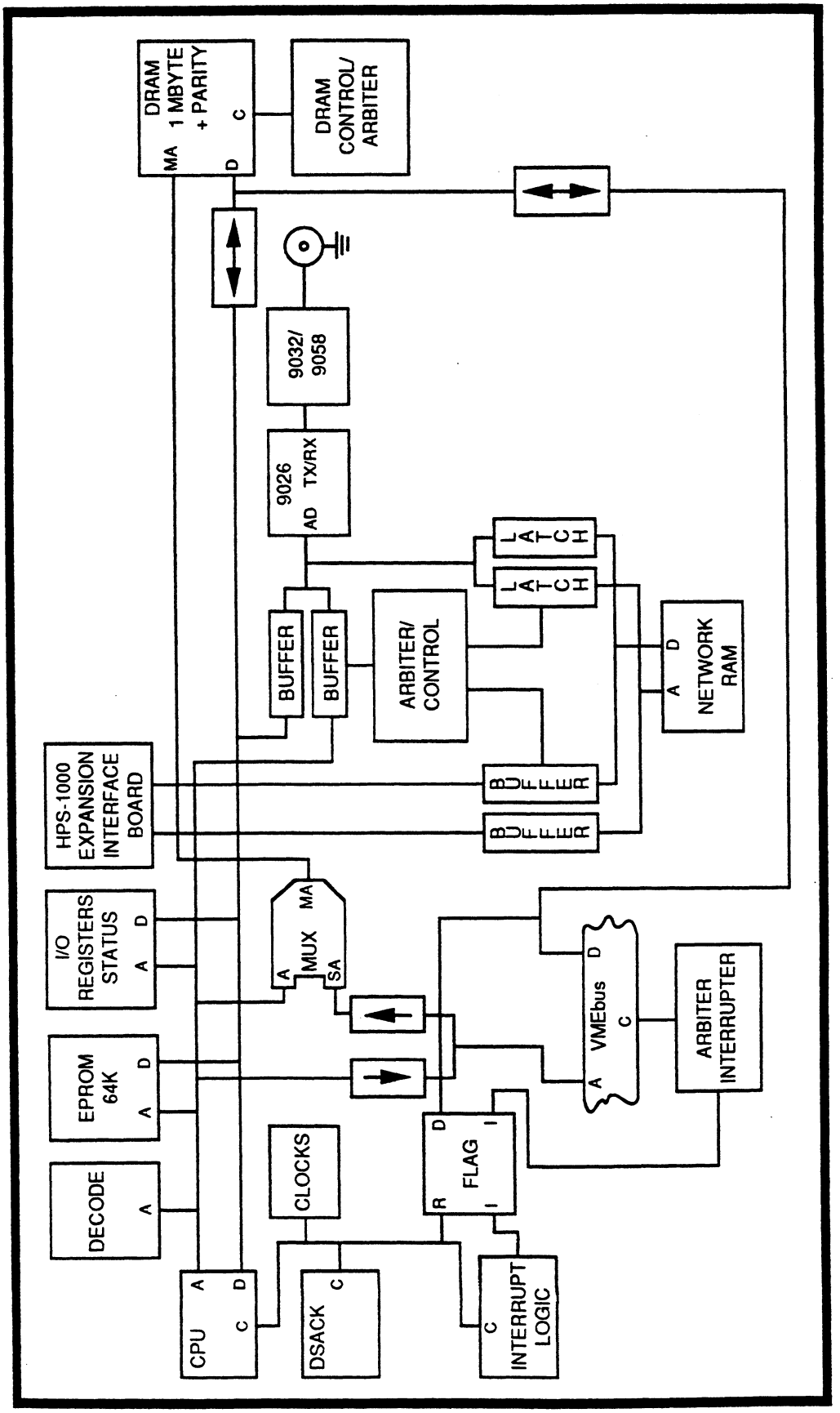


Figure 3-1. HPS-6245-25 Functional Block Diagram

3.1.2 PROM

The HPS-6245-25 is equipped with one JEDEC 28-pin socket capable of supporting either a 32K byte or 64K byte PROM. PROM space is 8-bits wide. The 68020's dynamic bus sizing feature provides the ability to execute from a 8-bit PROM. The CPU always accesses PROM with three wait states on data lines.

3.1.3 MAIN MEMORY

Main memory consists of eight 256K x 4, 60 ns DRAMs (plus four 256K x 1, 60 ns DRAMs for parity) organized to provide 1M byte of parity-protected memory. The DRAMs are arranged so that the CPU can access the main memory 32 bits at a time. The CPU is also capable of performing byte accesses on any boundary, word accesses on any boundary, and long word accesses on any boundary via the dynamic bus sizing. Misaligned accesses are completed with more that one bus cycle if necessary, i.e., a long word transfer to address 1 would result in two bus cycles: 3 bytes to address 1, and 1 byte to address 4.

If memory becomes corrupted, a parity error would be detected. The CPU is informed of this through a nonmaskable interrupt.

Main memory is tri-ported by the following three requestors in the priority shown:

1. Refresh
2. VMEbus dual-port
3. CPU

The CPU has control of main memory when no other request is pending. The CPU accesses main memory with one wait state.

3.2 DETAILED OPERATION

3.2.1 MAIN MEMORY ARBITER

Main memory is arranged to provide up to 1 Mbyte of memory. The DRAMs are 60 ns access time parts.

A custom PAL[1] arbitrates access to main memory. Three requestors vie for memory: refresh, the VMEbus, and the CPU, in that order. The priority is based strictly upon which requestors are present.

The main memory arbiter is completely synchronous to the inverted CPU clock. All signals are synchronized to this clock before being used inside the arbiter.

The refresh method is *CAS* before *RAS*, which was selected because it eliminates the need for an external refresh address counter.

Refresh is requested once every 15 µs by a counter that counts down from the CPU clock. The counter generates the *REFREQ/* signal. The arbiter receives the *REFREQ/* signal and, with the *REFAEN/* signal, grants the next memory cycle to the refresh generator (*REFREQ/* is deasserted when *REFAEN/* is asserted).

The refresh bus cycle is five clocks long. The clock cycle begins when *REFAEN/* is asserted after receiving a *REFREQ/* from the real-time clock ASIC. The CAS enable signal, *CASEN/*, is asserted after *REFAEN/*, allowing the four *CAS/* signals to be driven to every DRAM on the next positive clock edge. *RASOFF/* is asserted at the first clock of the cycle, followed by *MUX/* on the second clock. *RAS/* is then asserted and driven to each DRAM, and *RASOFF/* is deasserted on the third clock. *REFAEN/* is deasserted on the fourth clock and *MUX/* is held low to guarantee *RAS/* minimum pulse width. Following a refresh cycle, one clock of "dead time" is used to guarantee *RAS/* precharge.

[1] PAL is a registered trademark of Monolithic Memories, Inc.

3.2.2 CPU DRAM CYCLES

A CPU cycle begins with *AS/* going low and *DRAMCS/* asserted. On the next clock state after *AS/* is asserted, *RAS/* is asserted, followed by *CASEN/*. *MUX/* is asserted on the first clock of the CPU cycle. Next, the appropriate *CAS/* signals are asserted, depending on the *SIZE 0*, *SIZE 1*, *address 0*, and *address 1* signals. *RASOFF/* is asserted on the second clock of the cycle, and *RAS/* and *CASEN/* are deasserted during the next clock state. *MUX/* is deasserted on the third clock of the cycle, and then the *CAS/* signals are deasserted. *RASOFF/* is deasserted on the fourth clock of the cycle, which terminates the CPU DRAM cycle.

DRAMCS/ and *AS/* must be asserted for any DRAM timing to begin. *AS/* is required because in the case of a cache hit, the addresses will be present; however, *AS/* is never asserted. Both DSACKs—*DSACK0/* and *DSACK1/*—are asserted to the CPU, in time for one wait state, at the second clock when *DRAMACK/* is asserted. The DSACK PAL determines—from what is decoded and what "acknowledges" it receives—which DSACKs to assert.

3.2.3 VMEbus DUAL-PORT CYCLES

VMEbus dual-port cycles begin when a VMEbus master asserts onto the VMEbus an address and address modifier code that matches the value selected by the jumper fields. The DRAM arbiter receives a request upon receiving a falling edge on either *DS0* or *DS1*. *VMEREQ/* is asserted and held until the arbiter asserts *VMEAEN/*. The assertion of *VMEAEN/* indicates to the memory logic that this is a VMEbus dual-port cycle. Data buffers are then enabled from the VMEbus to memory. Since *VMEAEN/* is synchronized to the inverted CPU clock, at the next clock *RAS/* is asserted to the DRAM array. Then *MUX/* is asserted, followed by one of the four *CAS/* lines. The selection of the *CAS/* line is determined by the *DS0**, *DS1**, *LWORD**, and *A01* bus signals. One and one-half clock ticks—60 ns—after the *CAS/* line(s) are asserted, *DTACK** is asserted to the bus. Waiting this amount of time ensures that data is valid on the bus before *DTACK** is asserted. Dual-port memory cycle timing terminates with *RASOFF/* deasserted.

3.2.4 VMEbus MASTER INTERFACE

VMEbus cycles begin by writing a "1" to address 6000017 to request the bus. This informs the VMEbus front-end arbiter that the 68020 is interested in performing a VMEbus cycle. After the bus has been obtained, bit 0 in the status register will go low, indicating that VMEbus cycles may be initiated. This step must be performed to prevent the 68020 from "hanging up." Since the VMEbus is a 32-bit address bus, the 68020 can only view only a portion of the bus space—4M byte of the 4G byte space—at a time. The remaining ten address lines, along with the address modifier codes, are driven from a register. These must be set prior to any transfers.

The 68020 uses the same data path for DRAM as for the VMEbus. This means to begin a bus cycle it must first obtain control of the DRAM bus, accomplished by asserting *CPUVME/*, followed by *CPUAEN/*. *CPUVME/* informs the VMEbus master PAL that it can begin a VMEbus cycle as a master. The address bus is driven first, followed by *AS**, *DS0**, and *DS1**. The cycle ends when *DTACK** is received, which is passed on to the 68020 in the form of *DSACK0/* and *DSACK1/*.

The 68020 is able to perform 32-bit transfers to the VMEbus. You can, however, use long word instructions to devices on the bus which only accept short word transfers by setting the words/long words control bit at address 6000012 to high. This "breaks up" the cycles into short words.

3.2.5 MEMORY MAP FOR THE 68020 CPU

The 68020 memory map is defined in table 3-1.

Table 3-1. HPS-6245-25 Memory Map

Memory Location	Use	
00000000 - 00FFFFFF	Prom	(8-bit read only)
01000000 - 01FFFFFF	Memory	(32-bit read/write)
02000000 - 027FFFFF	COM 9026 registers	(8-bit read/write)
02800000 - 02FFFFFF	Network RAM	(8-bit read/write)
03000000 - 03FFFFFF	Flag space	(32-bit read/write)
04000000 - 04FFFFFF	VMEbus memory	(32-bit read/write)
05000000 - 05FFFFFF	Expansion interface bus (section 3.2.6)	(8-bit read/write)
06000000 - 06FFFFFF	Local control/status	(8-bit read/write)
Read 6000000H	Status register	(8-bit read only)
Write 6000000H	RTCCS/	(8-bit write only)
Read 6000010H	User configuration register	(8-bit read only)
6000010H - 6000017H	Control register	(8-bit write only)
All bits power up low:		
6000010H	unused	
6000011H	Enable parity checking	(active high)
6000012H	VMEbus long/short words	(active high)
6000013H	Green LED	(active high)
6000014H	Red LED	(active high)
6000015H	DTM data out	(active high)
6000016H	Unused	(active high)
6000017H	VME bus request	(active high)
Read 6000020H	Reset SYSFAIL (dummy read)	(8-bit read/write)
Write 6000020H	DTM Clock (dummy write)	(8-bit write only)
Read 6000030H	Unused	
Write 6000030H	Reset VME bus error (dummy write)	(8-bit write only)
07000000 - 07FFFFFF	Local I/O	
7000000H:	VMEbus master address register	(16-bit write only)

3.2.6 HPS-1000 EXPANSION INTERFACE BUS

The SYSTECH HPS-1000 expansion interface board is an 8-bit read/write slave-only device available for installation in any HPS 6200-series host adapter board. The HPS-1000 can respond to 17 bits of address, and provides a *DTACK** signal to the host board to indicate a completed cycle.

The HPS-1000 provides a means of adding options to the HPS host adapter. For the HPS-6245-25, an HPS-1000 board equipped with an RTscope option (HPS-1000-RTS) is available for customers who require the debugging capabilities provided by RTscope.

The HPS-1000-RTS board was designed for initial system integration. The height of this board violates the board spacing specification established by the VMEbus specification, Version C.1. Sufficient clearance (a minimum of two VMEbus slots) must be available while the HPS-1000-RTS board is installed.

When an HPS-1000 board is installed, three sense pins are grounded; these bits may be read to see if the board is installed, and what type of board it is (e.g., HPS-1000-RTS).

3.2.7 REAL-TIME CLOCK ASIC

The real-time clock is controlled by a programmable real-time clock ASIC. A 20 MHz clock provided to the ASIC generates the real-time clock. Unless changed in software, the real-time clock ASIC generates a default level 6 interrupt to the CPU every 52 milliseconds. If for any reason this interrupt is not serviced during the following 52 ms period, a level 7 nonmaskable watchdog interrupt is generated to the CPU.

3.2.8 NETWORK INTERFACE

The token-passing network interface consists of three major blocks: the shared RAM buffer, the network controller/transceiver, and the network driver. These three blocks provide a complete network controller so that the entire network operation is entirely transparent to the 68020 microprocessor. The network controller handles the token-passing protocol and network reconfiguration as nodes are added or deleted. The 68020 "sees" the network as a series of memory addresses that constitute up to four pages of memory in the shared RAM buffer.

3.2.8.1 Shared RAM Buffer

A dedicated 2K-byte bank of static RAM is shared by the 68020 and the COM 9026 network controller. This RAM resides at 68020 addresses 2800000H to 28007FFH. The 68020 accesses this RAM with 1, 2, or 3 wait states depending upon the COM 9026's usage.

3.2.8.2 Network Controller/Transceiver

The network controller/transceiver consists of the COM 9026/9032 chip set. These chips implement the token-pass bit serial bus access and data link protocols. The internal registers of the COM 9026 reside at addresses 2000000H and 2000001H. COM 9026 logic arbitrates accesses into these registers.

3.2.8.3 Network Driver

This analog module interfaces the digital signals of the COM 9032 transceiver to the network's coaxial cable medium. The network driver presents a transformer-isolated interface to the cable. The driver is a high-impedance tap for bus-style connections to the serial coaxial cable. A termination resistor terminates the coaxial connection at the beginning of the line (at the host computer). Jumper E33 is factory installed to enable this option (refer to chapter 2).

3.2.9 INTERRUPTS

The main CPU is the ultimate recipient of on-board interrupts. The 68020 supports seven interrupt levels, with seven being the highest and nonmaskable (NMI). Priority is determined external to the CPU and the highest requestor's interrupt is presented to the CPU. The interrupts are defined in table 3-2.

Table 3-2. Interrupt Summary

Interrupt Level	Function
Level 7	Watchdog/parity error
Level 6	Real-time clock
Level 5	COM 9026
Level 4	HPS-1000 expansion interface board
Level 3	unused
Level 2	unused
Level 1	Flag Byte

3.3 DIAGNOSTIC TEST MODULE OPERATION

At the front edge of the HPS host adapter board, approximately centered, is a six-pin connector to which the SYSTECH Diagnostic Test Module (DTM) can be connected.

The DTM provides two 16-position hexadecimal rotary switches and hexadecimal LED readouts to display diagnostic status.

The DTM communications protocol is a very simple synchronous serial bitstream via the status register at location 6000000H, bit 2, and the addressable latch at location 6000015H. Refer to the memory map shown as table 3-1 for specific DTM address locations.

3.3.1 SENDING STATUS TO THE DTM

Serial data transfer sets the DTM's status LEDs. This is accomplished by setting the addressable latch bit high by writing a "01H" to DTM Data Out (reset it by writing "00H" to this location). Writing any value to DTM Clock then toggles DTM Clock. Send the MSB of the 8-bit value first. After eight passes through this loop, the DTM LEDs will display the desired byte.

3.3.2 READING THE DTM SWITCHES

To read the DTM switches, "zero out" the DTM's shift register by resetting the addressable latch bit to "00H" and writing any value to the DTM Clock eight times. Then write a "01H" to DTM Data Out followed by any value to DTM Clock. The status register bit 8 then holds the inverted value of the first, or LSB, of the switches. Reset the addressable latch and complete seven more writes of any value to DTM Clock while testing status register bit 8 between "writes." This operation leaves the status LEDs equal to "00H." Restore them by performing the procedure described above in section 3.3.1.

Chapter 4
Self-Test Operation

The HPS-6245-25 performs a series of self-tests on three separate occasions:

- Each time it is powered up
- Whenever the reset button is pressed on a connected HPS Diagnostic Test Module (DTM), model 3320
- Whenever the host system writes a "02H" to the flag byte, which has the effect of generating a board reset

These tests thoroughly check the HPS hardware, in addition to performing a ROM checksum to ensure firmware integrity. The self-tests run without user intervention, however, with the aid of a DTM, the self-test procedure can be monitored by the codes displayed on the DTM's two hexadecimal display LEDs.

4.1 DIAGNOSTIC TEST MODULE

The DTM is a small test "fixture" that SYSTECH developed as a diagnostic aid. With the DTM, you can select special function tests and view the checkpoints and/or error codes as the HPS encounters them during self-test or under operating system control.

The DTM, which attaches to the HPS-6245-25 host adapter at connector J2 (see figure 4-1), contains two 16-position rotary switches, two hexadecimal-display LEDs, and a pushbutton switch. The rotary switches select 1 of 256 codes to be read by the host adapter at power-up (refer to appendix A for the code definitions). The LEDs display the checkpoints or error codes during self-test, and error codes during operating system control. Pressing the pushbutton switch resets the HPS-6245-25 (this is the equivalent of a power-on reset).

4.2 SELF-TEST ERRORS AND CHECKPOINTS

When an error is encountered during system power-up or reset, an error code or checkpoint value identifying the error is written to the *host_code* field location of DPRAM and, if the DTM is installed, is also displayed on the DTM's LEDs.

4.2.1 CHECKPOINTS

Checkpoints are numerical values, each of which corresponds to a particular test or portion of a test. As each test is successfully completed, the appropriate checkpoint is written to DPRAM and displayed on the DTM if attached. In this way, you can monitor the tests as they are performed. If an error is encountered that does not allow the HPS to display the appropriate error code, the sequence of writing and displaying checkpoints stops, and the last checkpoint identified (corresponding to the last test successfully completed) remains written/ displayed.

Since checkpoints indicate "passed" tests rather than "failed" tests, they do not directly indicate the test on which the error occurred. However, you can deduce the failed test by checking to see which test would have been performed next in the self-test sequence. Appendix B provides a sequential listing of checkpoints and corresponding self-tests.

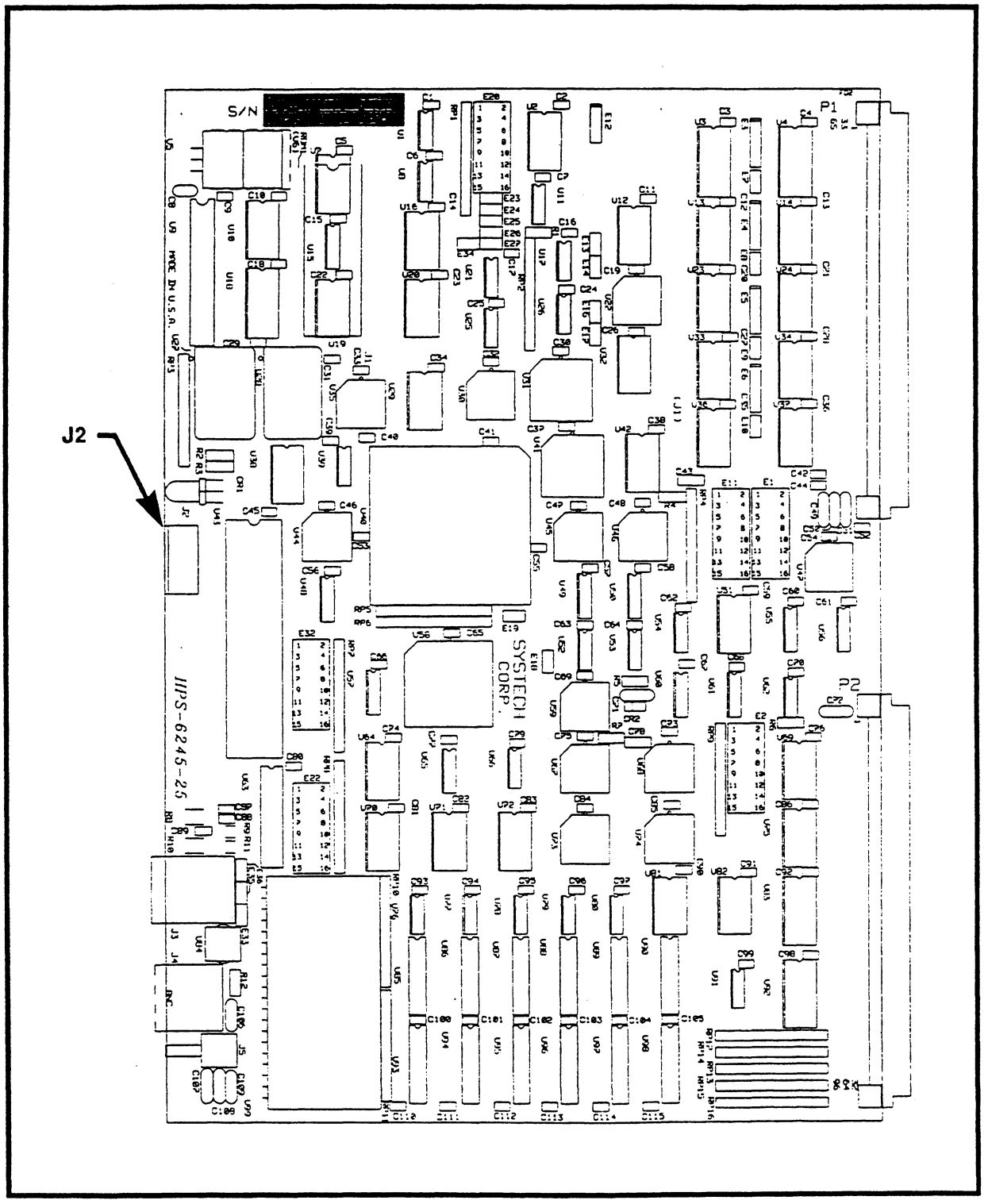


Figure 4-1. J2 Connector for Diagnostic Test Module

4.2.2 SELF-TEST ERROR CODES

In addition to checkpoints, error codes can be written to DPRAM and displayed on the DTM. These codes indicate specifically which test or portion of a test failed. The self-test procedure is described below, along with each error's corresponding error code and checkpoint. (Refer to appendix C for a sequential listing of the error codes).

1. **Start of self-test**
   The code 00H is written/displayed, indicating that the self-test is ready to begin.

2. **Set host adapter busy**
   The *hps_flag* byte location (DPRAM offset 01H) of DPRAM is set to 10H to indicate to the host that the HPS-6245-25 is busy and should not be interrupted.

3. **Clear remaining DPRAM I/O block locations**
   DPRAM offsets 02H through 1FH are written with 00.

4. **Enable VMEbus access**
   The hardware on the HPS is enabled to allow slave-mode accesses to the HPS board from the VMEbus.

5. **Test for the DTM**
   If the DTM is connected, the rotary switches are read and the appropriate action is taken. Refer to appendix A for DTM commands.

6. Stack data test

The stack tests (6, 7, 8, 9) check a 256-byte page of on-board RAM starting at absolute address 101FF00H. This page is used as a processor stack for subsequent portions of the self-test that require a stack, such as the interrupt tests.

If pins 7–8 of jumper E22 are installed, this test is not performed and its associated checkpoint is not identified.

If pins 7–8 of jumper E22 are not installed, the first word location on the stack (101FF00H) is written with 01, then checked. The bit is then rotated left and checked; this procedure is repeated 15 times to check all the bits.

If an error is detected, **error code 82H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

If no errors are detected, **checkpoint 01** is indicated.

7. Stack address test

If pins 7–8 of jumper E22 are installed, this test is not performed and its associated checkpoint is not identified.

If pins 7–8 of jumper E22 are not installed, the lower half address of each word location of the stack is pushed onto the stack and then popped off and checked.

If an error is detected, **error code 83H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

If no errors are detected, **checkpoint 02** is indicated.

8. **Stack checkerboard test**

   If pins 7–8 of jumper E22 are installed, this test is not performed and its associated checkpoint is not identified.

   If pins 7–8 of jumper E22 are not installed, all stack locations are first pushed with the value 5555H, and then popped off and checked. Next, all stack locations are pushed with the value AAAAH, and then popped off and checked.

   If an error is detected, **error code 84H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

   If no errors are detected, **checkpoint 03** is indicated.

9. **Zero stack test**

   If pins 7–8 of jumper E22 are installed, this test is not performed and its associated checkpoint is not identified.

   If pins 7–8 of jumper E22 are not installed, all stack locations are pushed with the value 0000, and then popped off and checked.

   If an error is detected, **error code 85H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

   If no errors were detected, **checkpoint 04** is indicated.

10. **RAM size**

    The size of RAM is determined. This value is saved for future use by the self-test and the operating system.

    If a nonvalid RAM size is encountered, **error code 9AH** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

    If no errors are detected, **checkpoint 05** is indicated.

11. Clear parity bits

If pins 7–8 of jumper E22 are installed, this test is performed. RAM parity is turned off, then all RAM locations from 1000020H to 10FFFFFH are read, then the value is written back. Following the test, RAM parity is turned on again.

No error code is associated with this test; **checkpoint 06** is indicated.

12. Stack addressing conflicts test

If pins 7–8 of jumper E22 are installed, this test is not performed and its associated checkpoint is not identified.

If pins 7–8 of jumper E22 are not installed, various data patterns are written to RAM locations below and above the stack and then the stack is checked for proper data.

If an error is detected, **error code 86H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

If no errors are detected, **checkpoint 07** is indicated.

13. HPS-1000-RTS detection test

The LED on the HPS-1000-RTS board (if installed) will be green. The following tests must all successfully complete before the HPS-1000-RTS board is recognized as being installed and functional.

- Verify that ROM address 5000000H can be read.
- Verify that UART base address 5010000H can be read.
- Verify that the LED registers 5010010H and 5010011H can be read.

If all of the above tests complete successfully, the HPS-1000-RTS LED will flash green in unison with the HPS-6245-25's LED. If one or all of the tests fail, the LED will turn solid red.

14. Watchdog timeout test

If this portion of the code is not reached within two clock ticks (approximately 100 ms) from the time the self-test starts, **error code B8H** is indicated, the HPS' LED blinks red, and the self-test stops.

If the above error is not encountered, the self-test will wait for a watchdog timeout to occur by setting a delay loop for 200 ms and not servicing the real-time clock. If no watchdog timeout occurs within that time, **error code 87H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

If no errors are detected, **checkpoint 08** is indicated.

15. Real-time clock test

This is the first time in the self-test where the real-time clock interrupts (interrupt level 6 of the microprocessor) are enabled.

A delay loop of 200 ms is entered waiting for a real-time clock interrupt. This occurs twice to ensure that the real-time clock interrupt occurs prior to 200 ms elapsing.

If no real-time clock interrupt occurs, **error code 88H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

If no real-time clock interrupt occurs during the second delay loop, **error code 88H** is indicated, the HPS' LED blinks red, and the self-test stops

If no errors are detected, **checkpoint 09** is indicated.

16. ROM checksum test

With the exception of the locations immediately following, all ROM locations from 0H to the end of ROM are added, then compared with a precalculated checksum value:

- 34H – 35H: precalculated ROM checksum value
- 40H – 43H: address modifier lines default value
- 44H – 47H: reserved
- 48H – 4BH: upper address default value

If the values do not match, **error code 81H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

17. HPS-1000-RTS checksum test

If the HPS-1000-RTS board is installed and functional, all ROM locations—except the first 8 bytes—are added, then compared with a precalculated checksum value stored at location 5000006H - 5000007H.

If the checksums do not match, the HPS-1000-RTS LED turns red and self-test continues as if the HPS-1000-RTS board is not installed. No error code or checkpoint is associated with this test.

If no errors are detected, **checkpoint value 0A** is indicated.

18. Real-time clock pulse test

The self-test checks the real-time clock period.

The real-time clock period is checked to ensure that it occurs within 250 ms. If a clock tick is not detected within 250 ms, **error code 89H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

If no errors are detected, **checkpoint 0B** is indicated.

19. On-board RAM address test

If pins 7–8 of jumper E22 are installed, this test is not performed and its associated checkpoint is not identified.

If pins 7–8 of jumper E22 are not installed, RAM locations 1000020H to 10FFFFFH are written with a pattern, and then are read and compared. The pattern is one that will detect any address problems that might occur.

If an error is detected, **error code 8BH** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

If no errors are detected, **checkpoint 0C** is indicated.

20. On-board RAM checkerboard test

If pins 7–8 of jumper E22 are installed, this test is not performed and its associated checkpoint is not identified.

If pins 7–8 of jumper E22 are not installed, RAM locations 1000020H to 10FFFFFH are first written with 5555H, and then are read and verified. Secondly, the same locations are written with AAAAH, and then are read and verified.

If an error is detected, **error code 8CH** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

If no errors are detected, **checkpoint 0D** is indicated.

21. Zero on-board RAM test

If pins 7–8 of jumper E22 are installed, this test is not performed and its associated checkpoint is not identified.

If pins 7–8 of jumper E22 are not installed, all RAM locations (except the stack area) are written with 0000H, and then are read and verified.

If an error is detected, **error code 8DH** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

If no errors are detected, **checkpoint 0E** is indicated.

22. Network RAM data test

Network RAM, at absolute address location 2800002H, is written with 01H. The bit is rotated left and checked. This is done eight times, until all bit positions in network RAM location 2800002H have been tested.

If an error is detected, **error code 92H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

If no errors are detected, **checkpoint 16** is indicated.

23. Network RAM address test

All network RAM locations (2800002H through 28007FFH) are written with a pattern, then are read and compared. The pattern is one that will detect any address problems that might occur.

If an error is detected, **error code 93H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

If no errors are detected, **checkpoint 18** is indicated.

24. Network RAM checkerboard test

First, all network RAM locations (2800002H through 28007FFH) are written with 55H, and then are read and verified. Secondly, the same locations are written with AAH, and then are read and verified.

If an error is detected, **error code 94H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

If no errors are detected, **checkpoint 19** is indicated.

25. Zero network RAM test

All network RAM locations (2800002H through 28007FFH) are written with 00H, and then are read and verified.

If an error is detected, **error code 95H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

If no errors are detected, **checkpoint 1A** is indicated.

26. COM 9026 test

This consists of a series of subtests:

a. The first test makes sure that a COM 9026 interrupt is not pending. If an interrupt is pending, **error code B1H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

b. COM 9026 interrupts are then enabled on the microprocessor (interrupt 2) and a short delay is entered. If after the short delay the COM 9026 has not interrupted the microprocessor, **error code 98H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

c. The COM 9026 status register is then compared for the value E5H. If it is incorrect, **error code 96H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

d. Location 2800000H of the network RAM is checked for the value D1H. If this location is incorrect, **error code 96H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

e. Location 2800001H of the network RAM is checked for non-zero value. If it is incorrect, **error code 97H** is indicated (in the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the self-test stops.

If no errors are detected, **checkpoint 1B** is indicated (refer to Standard Microsystems Corporation's data sheet for the COM 9026 for additional information).

27. HPS-1000-RTS UART test

These tests are performed only if the HPS-1000-RTS board is installed and has passed all previous tests. If either the HPS-1000-RTS board is not installed or if it failed one of the previous tests, this test is not performed.

This test consists of two parts; the first tests the UART in polled mode while the second loops data in interrupt mode.

Serial I/O data loop test, part 1 - polled loopback mode

The UART is set up for 8 bits, 2 stop bits, 19.2K baud, no parity, internal or external loopback mode, and no serial transmit or receive interrupts.

If a DTM is attached and the rotary switches are set to 31, the UART is set up for external loopback mode. In this mode a jumper must be connected between pins 3 and 5 (TXD and RXD, respectively) of connector J2 on the HPS-1000-RTS board.

If either no DTM is attached or if the attached DTM's rotary switches are set to a number other than 31, internal loopback mode is selected. No jumper is needed in this mode.

The loopback mode portion of this test performs the following function checks:

- Status register is checked to ensure TxEMT and TxRDY are set.

- Data ranging from 00H to FFH is written to the transmit holding register, one byte at a time. After each byte is sent, the self-test waits briefly for the TxRDY bit of the receive port status register to be set.

  If after 1 second the bit is not set, the HPS-1000-RTS LED turns red and self-test continues as if the HPS-1000-RTS board was not installed. The rest of this test is then skipped.

- After each byte is received, it is compared with the sent byte, and then the status register is checked for parity error, overrun error, framing error, and break. It is checked again to ensure that only one byte was received.

  If an error is detected, the HPS-1000-RTS LED turns red and self-test continues as if the HPS-1000-RTS board was not installed. The rest of this test is then skipped.

  If no errors are detected, **checkpoint 1C** is indicated.

Serial I/O data loop test, part 2 - interrupt loopback mode

The UART is set up as for the polled loopback mode test, except that the transmit (Tx) and receive (Rx) interrupts are enabled. The interrupt loopback mode portion of the test performs the following functional checks:

- A 256-byte Tx buffer is created, written with data ranging from 00H to FFH.

- A 256-byte Rx buffer is cleared to 00H.

- Interrupts are turned on.

- When a Tx interrupt is generated, a data byte is read from the Tx buffer and placed in the UART transmitter. The count of bytes is checked, and if all characters have been transmitted, the transmitter is turned off.

- When an Rx interrupt is generated, the UART's status is checked to determine why the interrupt occurred.

  If it occurred because of a parity error, overrun error, framing error, or a break, the HPS-1000-RTS LED turns red and self-test continues as if the HPS-1000-RTS board was not installed. The rest of this test is then skipped.

- The character in the receiver is read and placed in the Rx buffer. If this was the last character to receive, the receiver is turned off and the Tx and Rx interrupts are disabled.

- If 1 second passes before this test has finished, then the HPS-1000-RTS LED turns red and self-test continues as if the HPS-1000-RTS board was not installed. The rest of this test is then skipped.

- The data in the Rx buffer is compared with the data in the Tx buffer. If the data does not compare, the HPS-1000-RTS LED turns red and self-test continues as if the HPS-1000-RTS board was not installed.

  If no errors are detected, **checkpoint 20** is indicated.

28. Clear SYSFAIL*

The *SYSFAIL** line is turned off to indicate that the HPS-6245 passed this test.

29. Clear RAM

If pins 9–10 of jumper E22 are installed, this test is not performed.

If pins 9–10 of jumper E22 are not installed, all RAM locations from 1000020H to 10FFFFFH are written with 00H. No error codes or checkpoints are associated with this test.

30. Host primitive function tests

These two tests require the cooperation of the host, and are therefore not part of the "self-test" in the strictest sense.

NOTE

A host-resident "handshake" program must be resident for these tests to be completed. If a host program is not available, bypass these tests by installing jumper E22, pins 3–4.

These tests execute only if pins 3–4 of jumper E22 are not installed. If these jumper pins are not installed, these tests execute after the HPS-6245 indicates "ready" to the host. The host should wait for the "ready" indication, and then begin its portion of the tests. If pins 3–4 of jumper E22 are not installed, all of the tests will be executed, in the order presented here. (Refer to appendix D for the "C" code listing of the host primitive tests.)

The primitive function tests verify correct operation of the host adapter/host interface, in preparation for normal IOCB processing.

a. **Checkpoint 40** is indicated while the HPS waits for the host "ready" signal.

b. Host primitive data loopback test

If pins 3–4 of jumper E22 are installed, this test is not performed and its associated checkpoint is not identified.

When the self-test code reaches this point, the value FFFFH is written to DPRAM offset 08H. The self-test will wait indefinitely for the host to start the loopback procedure. It is up to the host to time out and take action if, after some specified timeout period, the self-test has not reached this point of the test sequence.

When the host reads the value FFFFH, it should write FFFFH to DPRAM offset 0AH. This signals the self-test to proceed.

The self-test then writes 0001H to DPRAM offset 08H. The host should echo the value to DPRAM offset 0AH. The self-test then "walks" the bit across DPRAM offset 08H. Each time a new value is written, the host should echo it to DPRAM offset 0AH. The last value the host should echo is 8000H.

If the host does not respond by echoing the value back to offset 0AH within 1 second after the self-test writes a new value to DPRAM offset 08H, **error code C0H** is indicated, the HPS' LED blinks red, and the self-test stops.

If no errors were detected, **checkpoint 1E** is indicated.

c. Host primitive interrupt test

If pins 3–4 of jumper E22 are installed, this test is not performed and its associated checkpoint is not identified.

This test is similar to the previous host primitive test in that a data pattern starting with 0001H is sent back and forth between the host and the HPS-6245-25, is verified, and then is rotated left 1 bit position until the last value of 8000H is sent.

The self-test code starts this test by enabling host interrupts to the microprocessor (interrupt level 1). Then the self-test clears the *hps_flag* byte DPRAM (offset 01H).

The cleared *hps_flag* byte signals the host to write the first (or next) data pattern to DPRAM offset 0AH, then to interrupt the HPS-6245-25 by writing 11H to the *hps_flag* byte DPRAM (offset 01H). This signals the HPS-6245-25 that the next (or the first) data pattern has just been sent by the host.

If the host does not interrupt the HPS-6245-25 within 1 second after the host adapter clears its busy bit, **error code C1H** is indicated, the HPS' LED blinks red, and the self-test stops.

After servicing the first host interrupt, the self-test then waits 10 ms and checks to see if the host has interrupted the host adapter again. It should not interrupt again because the HPS busy bit is still set.

If the host has interrupted the host adapter again, **error code C2H** is indicated, the HPS' LED blinks red, and the self-test stops.

The self-test reads DPRAM offset 0AH and verifies that the data sent by the host is correct.

If the data is incorrect, **error code C4H** is indicated, the HPS' LED blinks red, and the self-test stops.

The self-test then waits up to 1 second for the host to become "not busy," that is, for DPRAM offset 00H to become 00H. The host should clear this byte when it is ready to receive another interrupt from the HPS-6245-25.

If 1 second elapses and the host is still busy, **error code C5H** is indicated, the HPS' LED blinks red, and the self-test stops.

The self-test writes the previously read value from DPRAM offset 0AH to DPRAM offset 08H, then interrupts the host by writing 11H to DPRAM offset 00H.

The previously read value is shifted left 1 bit and saved to compare with the next value that the host sends. If the shifted value equals 0, this test has completed, otherwise it is repeated with a new value being sent from the host.

If no errors were detected, **checkpoint 1F** is indicated.

d. Burn-in loopback/termination

If the burn-in mode was previously selected (DTM rotary switches set to FFH), the self-test will start over from the beginning.

31. HPS-1000-RTS UART initialization

If the HPS-1000-RTS board is installed and functional, the UART is set up for 8 bits, 1 stop bit, no parity, and 9600 baud.

32. VMEbus transfer width

If pins 13–14 of jumper E22 are installed, set up the HPS-6245 for 32-bit transfers; otherwise set it up for 16-bit transfers.

No error code or checkpoint is associated with this function. The self-tests are complete at this point, and control is passed to the operating system.

Appendix A
DTM Rotary Switch Code Definitions

A.1 HOW TO USE THE DTM

To execute a function on the Diagnostic Test Module (DTM), enter, on the DTM's two rotary switches, the code associated with that function (the codes are defined below). Then press the DTM's RESET button.

At this point, if the function is a test or scope loop function, the desired action will continue until a new code is entered and the RESET button is pressed again.

If the desired function is a mode type function, the self-test will execute as described in chapter 4, with the exception of the change associated with the DTM code.

NOTE

Invalid codes (that is, any that are not listed and defined below) are handled in the same manner as a setting of 00.

A.2 CODE DEFINITIONS

The DTM codes and their corresponding definitions are listed below.

- **00 = Default value**
  Enter this code if no test or mode is desired.

- **03 = Reserved for factory use**

- **05 = Display real-time clock calibration count**
  This test, which is used only for factory internal calibration, will display the 16-bit value obtained by a software loop running during one real-time clock tick. If no real-time clock tick occurs within 1 second, an error code will be displayed on the DTM's LEDs.

- **10 = Disable watchdog timeout**
  This switch setting still allows watchdog timeouts to occur, but they are not considered to be an error when they do occur.

  This mode is useful when an in-circuit emulator (ICE) is used and emulation is frequently being halted.

- **20 = HPS-1000-RTS UART output loop**

  This scope loop verifies (with the aid of an oscilloscope) that the TxDATA line and some of the internal registers on the UART function properly.

  The UART is set up for 19.2K baud, no parity, 8 data bits, and 1 stop bit.

  Data pattern 55H, AAH is repeatedly written, 1 byte at a time, to the UART transmit holding register. The status register is then polled to see if the UART can accept another character. If so, another character is written; if not, the status register is polled for up to 200 ms. If the UART TxDATA register is not empty after 200 ms, **error code A0H** is indicated (the the DPRAM *host_code* field and on the DTM if attached), the HPS' LED blinks red, and the HPS-1000-RTS LED turns solid red.

  To stop this test, enter 00 or a new code on the DTM's rotary switches, then press the DTM's reset button.

- **30 = DTM test**

  This test verifies the DTM and interface circuitry.

  The code entered on the DTM's rotary switches will be displayed on the DTM's LEDs.

  To stop this test, enter 00 or a new DTM code, then press the DTM's RESET button.

- **31 = Serial data external loop mode**

  If the HPS-1000-RTS board is installed, this mode sets up the board to send the serial data, during the UART portion of the self-test, to the RS-232 connectors' TxDATA serial lines.

  For the UART test to successfully complete, you must install a jumper between pin 3 and pin 5 (TxD and RxD, respectively) on connector J2 of the HPS-1000-RTS board.

  If an error is detected during the UART portion of the self-test with this mode enabled, the HPS-1000-RTS LED turns red and the self-test continues as if the HPS-1000-RTS board is not installed.

  Since this is a DTM mode command, after the UART test has been completed, the remainder of the self-test completes as described in chapter 4 of this manual.

- **41 = Option jumper E22 test**

  This test verifies that the option jumper, E22, functions properly.

  Indication of which jumper pins are installed is displayed on the DTM's LEDs in the following manner:

  - 00: all jumper pins installed.
  - 01: jumper pins 1-2 only not installed.
  - 02: jumper pins 3-4 only not installed.
  - 04: jumper pins 5-6 only not installed.
  - 08: jumper pins 7-8 only not installed.
  - 10: jumper pins 9-10 only not installed.
  - 20: jumper pins 11-12 only not installed.
  - 40: jumper pins 13-14 only not installed.
  - 80: jumper pins 15-16 only not installed.
  - FF: no jumper pins installed.

  To stop this test, enter 00 or a new DTM code, then press the DTM's RESET button.

- **42 = Host adapter network ID address jumper E32 test**

  This test verifies that the host adapter network ID address jumpers function properly.

  Indication of which jumper pins are installed is displayed on the DTM's LEDs in the following manner:

  - 00: no jumper pins installed.
  - 01: jumper pins 1-2 only installed.
  - 02: jumper pins 3-4 only installed.
  - 04: jumper pins 5-6 only installed.
  - 08: jumper pins 7-8 only installed.
  - 10: jumper pins 9-10 only installed.
  - 20: jumper pins 11-12 only installed.
  - 40: jumper pins 13-14 only installed.
  - 80: jumper pins 15-16 only installed.
  - FF: all jumper pins installed.

  To stop this test, enter 00 or a new DTM code, then press the DTM's RESET button.

- **60 = Jump to RTscope's command mode if nonfatal error occurs**

  If the HPS-1000-RTS board is installed on the HPS-6245 and a nonfatal error is encountered by the operating system, program control will be passed to the RTscope module. Refer to the *HPS 6200-Series Application Installation Manual* for a complete discussion of nonfatal errors.

  RTscope will enter its command mode and then send the following message to the attached CRT:

  ```
  Anomalous Bpt trap, PC=address
  RC >
  ```

At this point, DPRAM parameters and program variables can be examined to determine the cause of the nonfatal error.

To resume normal operating system execution, type `task` and press the ENTER or CARRIAGE RETURN (CR) key on the attached CRT.

NOTE

Tests 70 through 76 are scope loop tests that are not intended to be executed as a normal power-up procedure. They are contained in the firmware for debugging this HPS product.

- **70 = Dual-port arbitration test**

  This test works in conjunction with a host-resident complement test. The portion that resides on the HPS-6245 is a simple loop that:

  a. reads DPRAM word location 08H.
  b. writes DPRAM word location 0AH with the previously read word.
  c. clears the real-time clock interrupt flip-flop.

  Real-time clock interrupts are enabled during this test. Rather than the interrupt being serviced, each time the loop is executed, the real-time clock interrupt circuitry is reset, preventing the real-time interrupt from ever occurring. This is done to prevent a watchdog timeout from occurring.

  The host portion of this test can be tailored to your needs. For example, a simple test would involve sending data from 0H to FFFFH out of DPRAM word location 08H, then waiting a small amount of time, and then reading DPRAM word location 0AH. The data should match.

With the sample test (above) running on both the host and the HPS-6245-25, DPRAM arbitration is being exercised.

To stop this test, enter 00 or a new DTM code, then press the DTM's RESET button.

- **71 = Read DRAM location**

  This scope loop test only reads dynamic RAM (DRAM) byte location 1001500H. All interrupts are disabled, causing the watchdog timeout to occur every two clock ticks.

  When a watchdog timeout occurs, the interrupt is cleared and RAM location 1001500H is again continuously read.

- **72 = Write DRAM location**

  This scope loop test first reads DRAM byte location 1001500H and saves the read value. The read value is then continuously written to DRAM byte location 1001500H. All interrupts are disabled, causing the watchdog timeout to occur every two clock ticks.

  When a watchdog timeout occurs, the interrupt is cleared and RAM location 1001500H is again continuously written with the value that was first read.

  To stop this test, enter 00 or a new DTM code, then press the DTM's RESET button.

- **73 = Read and write DRAM location**

  This scope loop test only reads DRAM byte location 1001500H, then writes the read value back. All interrupts are disabled, causing the watchdog timeout to occur every two clock ticks.

When a watchdog timeout occurs, the interrupt is cleared and the read/write loop is again entered.

To stop this test, enter 00 or a new DTM code, then press the DTM's RESET button.

- **74 = Read all DRAM**

  This scope loop test reads all DRAM byte locations from 1000000H to the top of RAM (10FFFFFH), one byte at a time. All interrupts are disabled, causing the watchdog timeout to occur every two clock ticks.

  When a watchdog timeout occurs, the interrupt is cleared and the scope loop continues where it left off.

  To stop this test, enter 00 or a new DTM code, then press the DTM's RESET button.

- **75 = Write all DRAM**

  This scope loop test writes all DRAM byte locations from 1000000H to the top of RAM (10FFFFFH) with the byte value that resides in DRAM location 100000H. All interrupts are disabled, causing the watchdog timeout to occur every two clock ticks.

  When a watchdog timeout occurs, the interrupt is cleared and the scope loop continues where it left off.

  To stop this test, enter 00 or a new DTM code, then press the DTM's RESET button.

- **76 = Read and write all DRAM locations**

  This scope loop reads, then writes back a byte of DRAM. All DRAM locations, 1000000H to the top of RAM (10FFFFFH) are read/written. All interrupts are disabled, causing the watchdog timeout to occur every two clock ticks.

  When a watchdog timeout occurs, the interrupt is cleared and the scope loop continues where it left off.

  To stop this test, enter 00 or a new DTM code, then press the DTM's RESET button.

- **FD = Display last error encountered**

  While not actually a mode or a test, this function will only be performed if the "burn-in mode" (described below) was previously selected (DTM's rotary switches set to FFH). This function displays the last error that was encountered during continuous execution of the self-test.

  After selecting this function, **do not press** the DTM RESET button to invoke it. The error code display may take several seconds to appear because the current self-test loop **must** complete before the error code can be displayed.

  If no errors were previously detected, the DTM's LEDs will display 00. If an error was detected, the DTM's LEDs will alternate between 00 and the error code.

  To continue the burn-in mode, rotate the DTM's rotary switches back to FFH.

- **FE = Display total count of errors**

  While not actually a mode or a test, this function will only be performed if the burn-in mode was previously selected (DTM's rotary switches set to FFH). This function displays the total number of errors that have been encountered during continuous execution of the self-test.

When this function is selected, **do not press** the DTM RESET button to invoke it. The error count display may take several seconds to appear because the current self-test loop **must** complete before the count can be displayed.

If 255 or more errors have been detected, the DTM's LEDs will display FFH.

To continue the burn-in mode, rotate the DTM's rotary switches back to FFH.

- **FF = Burn-in mode**

  This mode differs from the other modes in that after rotating the DTM's rotary switches to the FFH position, you must press the DTM's RESET button to start executing the continuous self-test. Once you've initiated the self-test, you must refrain from pressing the RESET button again.

  This mode executes the self-test repeatedly, keeping track of the last error and the total error count.

  If an error is detected, the LED located on the HPS host adapter will blink red (instead of yellow). This indicates that an error has occurred, and by using the DTM's "error type" and "error count" functions (described above), you can determine the error type and frequency of errors.

  To stop this test, enter 00 or a new DTM code, then press the DTM's RESET button.

Appendix B
Self-Test Checkpoints

The checkpoints recognized by the HPS self-test software (and described in chapter 4) range in value from 01H to 7FH. The checkpoints are listed below, along with a brief description of what each checkpoint indicates.

- **00** = Reset condition. The HPS-6245-25 is starting the self-test operation.
- **01** = Stack data test was completed successfully.
- **02** = Stack address test was completed successfully.
- **03** = Stack checkerboard test was completed successfully.
- **04** = Zero stack test was completed successfully.
- **05** = Size of HPS dynamic RAM (DRAM) was determined successfully.
- **06** = RAM parity test was completed successfully.
- **07** = Stack addressing conflicts test was completed successfully.
- **08** = Watchdog timeout test was completed successfully.
- **09** = Real-time clock test was completed successfully.
- **0A** = ROM checksum test was completed successfully.

- **0B** = Real-time clock calibration test was completed successfully.
- **0C** = DRAM addressing test was completed successfully.
- **0D** = DRAM checkerboard test was completed successfully.
- **0E** = DRAM zero test was completed successfully.
- **16** = Network RAM data test was completed successfully.
- **18** = Network RAM address test was completed successfully.
- **19** = Network RAM checkerboard test was completed successfully.
- **1A** = Zero network RAM test was completed successfully.
- **1B** = COM 9026 test was completed successfully.
- **1C** = HPS-1000-RTS polled data loopback test was completed successfully.
- **20** = HPS-1000-RTS interrupt data loopback test was completed successfully.
- **40** = HPS-6245-25 is waiting for the host to start the primitive functions test. If pins 3–4 of jumper E22 are installed, this checkpoint is not displayed. The next displayed checkpoint is FF.
- **1E** = Host primitive function data loop test was completed successfully.
- **1F** = Host primitive function flag byte handshaking test was completed successfully.
- **21** to **FD** (with the exception of **40**) are not used at this time.
- **FF** = Entire self-test was completed successfully.

Appendix C
Self-Test Error Codes

The error codes recognized by the HPS self-test software (and described in chapter 4) range in value from 81H to FDH. The error codes are listed sequentially below, along with a brief description of the test that corresponds to each error code.

- **81** = ROM checksum error.
- **82** = Stack data test error.
- **83** = Stack address test error.
- **84** = Stack checkerboard test error.
- **85** = Zero stack test error.
- **86** = Stack addressing conflict error.
- **87** = Watchdog timeout test error.
- **88** = Real-time clock interrupt error (no real-time clock present).
- **89** = Real-time clock interrupt error (bad clock pulse width).
- **8B** = Dynamic RAM address test error.
- **8C** = Dynamic RAM checkerboard test error.

- **8D** = Zero dynamic RAM test error.
- **92** = Network RAM data test error.
- **93** = Network RAM location addressability test error.
- **94** = Network RAM checkerboard test error.
- **95** = Zero network RAM test error.
- **96** = COM 9026 test: status register incorrect value.
- **97** = COM 9026 test: network ID address jumper E32 is not installed (i.e., address of 0).
- **98** = Network interrupt error. An interrupt did not occur when expected.
- **9A** = RAM size error.
- **B1** = A COM 9026 interrupt occurred, but the interrupt was not enabled.
- **B2** = The POR bit of the COM 9026 is not set during an interrupt.
- **B6** = Dynamic RAM parity error.
- **B7** = Watchdog timeout error.
- **B8** = Watchdog timeout occurred too early in the self-test sequence.
- **C0** = Host to HPS I/F test: data wrap test error.
- **C1** = Host to HPS I/F test: flag byte interrupt test error.
- **C2** = Host to HPS I/F test: host interrupt bit will not reset.

- **C4** = Host to HPS I/F test: data returned < > data sent.
- **C5** = Host to HPS I/F test: host not ready for more data error.
- **D0** = RAM parity error.
- **F0** = HPS local bus exception error.
- **F1** = HPS local address exception error.
- **F2** = Illegal instruction exception error.
- **F3** = Interrupt exception error.
- **F4** = Trap exception error.
- **F5** = Other exception error.

Appendix D
Host Primitive "C" Listing

The "C" program listing for the host-primitive test follows:

```
struct HPS6816
{
   bit8  host_flag;      /* HPS writes to interrupt host */
   bit8  hps_flag;       /* Host writes to interrupt HPS */
   bit8  host_code;      /* HPS writes to indicate results */
   bit8  hps_mode;       /* Host writes to set modes */

   union
   {
      struct   /* Set up this way during self-test  */
      {
         bit32 test_ptr;    /* For DMA test */
         bit16 hps_out;     /* For data and … */
         bit16 hps_in;      /* … interrupts tests */
      }  st;

      struct   /* Set up this way during normal operation */
      {
         bit32 iocb_ptr;    /* Address of next IOCB */
         bit32 message;     /* Response message */
      }  n;
   }  un;


   struct   /* VRTX/TRACER I/O area  */
   {
         bit8  input_count;    /* Host writes input char count */
         bit8  output_count;   /* HPS writes output char count */
         bit8  inp_buf[8];     /* Up to 8 input characters */
         bit8  out_buf[8];     /* Up to 8 output characters */
   }  vtio;
};
```

```
/* Shorthand union access: */
#define st_ptr       un.st.test_ptr
#define st_out       un.st.hps_out
#define st_in        un.st.hps_in

/* Flag byte bits: */
#define INTR         0x01     /* Hardware interrupt */
#define HDWR_RESET   0x02     /* Hardware reset */
#define BUSY         0x10     /* Software busy */

#define HPSADDR      0xFF0000 /* Memory-mapped address of DPRAM */

#define HPS          ((struct HPS6816 *)HPSADDR)

#define FALSE        0

#define TRUE         (-FALSE)
#define NULL         0
#define NORMAL       0
#define REVERSE      1
#define LONG_TIME    500000
#define SHORT_TIME   1000
```

```
/*    HPS-6816  -  Primitive Functions Tests

*  This routine executes the last phases of the HPS self-test, in
*  concert with the HPS host adapter.  The routine includes two tests:
*
*     1) Bus Interface Data Path
*        This test walks a bit across the data bus between the
*        host processor and the HPS processor.
*     2) Flag Byte Handshaking
*        This test exercises the interrupt handshaking between
*        the host and the HPS processors
*/


hps_st()
{
   char     t;
   bit8     bexpecting, bgot;
   short    expecting, got, was;
   long timeout;

   /*
   * Wait for the HPS to indicate that self-test has completed
   * up to the Primitive Functions test section
   */
   expecting = 0xFFFF;

   timeout = LONG_TIME;

   while ((got = HPS->st_out )  ! = expecting)
   {
      timeout--;
      if (timeout == 0)
      {
         printf("hps:   Primitive Tests timeout\n");
         printf("       Waiting for\"READY\" status\n");
         printf("       Expected 0x%x, got 0x%x\n",
                  expecting & 0xFFFF, got & 0xFFFF),
         goto error;
      }
   }
   HPS->st_in = was = got;
```

```
/*
* The HPS is ready for the first phase.  Echo the data that appears
* in HPS_OUT back to HPS_IN.
*/
for (expecting = 1; expecting  ! = 0; expecting << = 1)
{
   /* Wait for register to change to a new value */
   timeout = SHORT_TIME;
   while ((got = HPS->st_out) == was)
   {
      timeout--;
      if (timeout == 0)
      {
            printf("hps:   Primitive Tests timeout; Data Path\n");
            printf("       Waiting for 0x%x to change to 0x%x\n",
                     was & 0xFFFF, expecting & 0xFFFF);
            printf("    Got 0x%x", got & 0xFFFF);
            goto error;
      }
   }

   /* Did it change to the right thing? */
   if (got  ! = expecting)
   {
      printf(hps:    Primitive Tests failure; Data Path\n");
      printf("       Expected 0x%x, got 0x%x\n",
               expecting & 0xFFFF, got & 0xFFFF),
      goto error;
   }
   HPS->st_in = was = got;
}

/*
* Now see if interrupts are working
*/

For (expecting = 1; expecting  ! = 0; expecting << = 1)
{
   /* Wait for HPS to be OK to interrupt */
   timeout = SHORT_TIME;
   while ((got = HPS->hps_flag)  ! = 0)
   {
      timeout--;
      if (timeout == 0)
      {
         printf("hps:   Primitive Tests timeout, Flags\n");
         printf("       Waiting for HPS to be ready\n");
         printf("       HPS flag holds 0x%x", got & 0xFF);
         goto error;
      }
   }
```

```
   /*  Give value to host */
   HPS->st_in  =  expecting;
   HPS->hps_flag = INTR | BUSY;     */ Interrupt the HPS */

   /* Wait for interrupt from the host adapter */
   timeout = LONG_TIME;
   while (HPS->host_flag == 0)
   {
      timeout--;
      if (timeout == 0)
      {
         printf("hps:   Primitive Tests timeout, Flags\n");
         printf("       Waiting for interrupt from HPS\n");
         printf("       (Expecting 0x%x)\n", expecting & 0xFFFF);
         goto error;
      }
   }

   /* Did the HPS give us the right stuff? */
   if ((got  =  HPS->st_out)  ! = expecting)
   {
      printf("hps:   Primitive Tests timeout; Flags\n");
      printf("       Expected 0x%x, got 0x%x\n",
               expecting & 0xFFFF, got & 0xFFFF),
      goto error;
   }

   /* Indicate that we're ready for another interrupt */
   HPS->host_flag  =  0x00;
}

/*
* Wait for host adapter to indicate that it has passed the self-test
*/

timeout = LONG_TIME;
bexpecting = 0xFF;                        /* Final self-test checkpoint */
while ((bgot = HPS->host_code)  ! = bexpecting)
{
   timeout--;
   if (timeout == 0)
   {
      printf("hps:   Primitive Tests timeout\n");
      printf("       Waiting for \"Self-test Passed\" status \n");
      printf("       Expected 0x%x, got 0x%x & 0x%x\n",
               bexpecting & 0xFF, bgot & 0xFF,
               &HPS->host_code);
      goto error;
   }
}
```

```
   /* Done; now wait for HPS to become not busy */

   timeout = LONG_TIME;
   while ((t = HPS->hps_flag)  ! = 0)
   {
      timeout--;
      if (timeout == 0)
      {
         printf("hps:   Primitive Tests timeout\n");
         printf("       HPS never became NOT busy\n");
         goto error;
      }
   }

error:
   printf("    Last status:  0x%x\n", HPS->host_code & 0xFFFF);
   return(1);
}
```

Appendix E
OEM Warranty and Repair Procedure

This chapter explains how to contact SYSTECH's Customer Service Product Support group regarding product operating problems, and describes how to return products for repair (both warranty and nonwarranty products).

This information applies to SYSTECH's authorized distributors (OEMs). An "end user's" point of contact for support or repair is the authorized distributor from which the product was purchased.

E.1 CALLING CUSTOMER SERVICE

Before calling the customer service number regarding a product support issue, have the following information available:

- A detailed description of the problem.
- System type.
- Operating software and version level.
- SYSTECH product, model, part number, and revision levels of both the hardware and software product.

The SYSTECH customer service number is: **(619) 453-8970**

E.2 RETURNING PRODUCT FOR REPAIR

Should the product need to be returned for repair, you will need a Return of Material Authorization (RMA) number from SYSTECH. The RMA number is valid for 30 days. RMA items not received by SYSTECH within 30 days of the RMA's issuance will be cancelled; a new RMA number will be required for receipt to be honored. All returns to the factory (warranty, nonwarranty, loaners, evaluation, beta, and exchanges) require an RMA number. The RMA number must appear on the outside packaging of all returned products.

A valid purchase order is required for all repaired products, both warranty and nonwarranty. Products under warranty will be returned prepaid; products not covered under warranty will be returned prepaid and billed.

NOTES

It is SYSTECH policy to invoice at the rate of $125.00 per item for each item which completes the repair cycle with no problem found. Warranty purchase orders should include the amount of $125.00 per item. This amount is returned if a problem is found.

Any parcel received without an RMA number noted on the outside of the package will be refused by the SYSTECH receiving department, and will be returned freight collect to the sender.

For international shipments, we do not recommend the use of Air Parcel Post.

To return product for repair:

- Obtain an RMA number by calling the SYSTECH *order entry department* at (619) 453-8970.
- To enable SYSTECH to verify warranty repairs, provide to the order entry representative the date you received the product.

- Provide a valid purchase order number for the product being returned.

- Include in the warranty purchase order $125.00 per item for any item that is determined to be "no problem found."

- Provide the part number and serial number of the SYSTECH product. For most board assemblies, the serial number is located on the component side while the part number is printed on a small white label on the solder side. For cluster controllers, labels are affixed to the unit's back panel. For SPURs, the serial number appears on a large silver label and the part number on a small white label; both are affixed to the bottom of the unit. For board assemblies, cluster controllers, and SPURs, the part number takes the form 65-xxxxxx-x-xx.

- Verify with the order entry representative your shipping and billing address.

- Describe the problem and include the name and phone number of a technical contact who can provide further information about the failure.

- Package the product for shipment, indicating on the outside of the parcel the RMA number. Then ship the product freight and insurance prepaid to:

  SYSTECH Corporation
  6465 Nancy Ridge Drive
  San Diego, CA 92121

- The product will be returned, repaired or exchanged within 20 working days. A valid hardcopy purchase order must be received by SYSTECH before the product will be returned to the customer. Products under warranty are returned prepaid. Products not covered under warranty are returned prepaid and billed.