QED 95 CPU

(11/84)

PDP-11 PROCESSOR UPGRADE

Version 4.3

Quickware Engineering & Design, Inc.
225 Riverview Avenue
Waltham, MA. 02154-3874
(800) 237-1185
In MA: (617) 647-3800

Material in this manual is believed to be correct but is subject to change without notice. We are constantly striving to improve our products and our documentation. We appreciate any suggestions you may have concerning our products and welcome comments for revising our manuals.

PER JAN '2000:

PDP QED 95

* PARIS2	dish : ϕ11ϕ
* WB2 ONTWIKKEL	" : ϕ255
* KW12-CC	" : ϕ618
* KW12-PC	" : ϕ619

1 Introduction

The QED 95 is a DEC Unibus-compatible processor that is fully compatible with all PDP–11 Unibus hardware and software. The QED 95 addresses four megabytes of memory and implements the complete PDP-11 architecture, excluding the Commercial Instruction Set, (CIS, found only as an option on the 11/23, 11/24, and 11/44 systems). The QED 95 consists of two processor boards, a memory board, and optional host adapter boards. The QED 95 system is designed to replace your current PDP-11 processor and memory.

The QED 95 is manufactured under license from Digital Equipment Corporation. A separate software sublicense available through Quickware is required for each QED processor running a Digital operating system.

2 Features

- High-speed CMOS processor
- 32 KB cache memory
- Private memory bus (QMI bus)
- Boot ROMs
- Configuration menu
- On-board serial line units
- On-board clock
- Unibus interface
- Massbus interface (when used with 11/70)
- Four megabyte error-correcting memory
- Unibus map
- Concurrent MMU, ALU and DMA operation
- Optional Floating point implemented in hardware

2.1 QED 95 CPU

2.1.1 CMOS Processor

The CPU has a major cycle time of 110 nanoseconds (ns), allowing the execution of up to nine million instructions (MIPS) per second. In a typical program, you can expect an average of four MIPS. By comparison, the PDP-11/70 runs at 1.36 MIPS average while the 11/84 runs at 1.56 MIPS. Actual system performance will depend

on the number of users and the amount of I/O activity. The QED 95 memory system allows transfers at over five MB/sec.

2.1.2 Cache Memory

The QED 95 has 32 KB of cache memory. This cache memory allows data and instruction fetches to occur in as little as 110 ns. The 32 KB cache is four times larger than other PDP-11 cache memories and provides a hit rate of over 95 percent.

Parity checking on both cache data and tags helps ensure accurate program execution. If a parity error occurs, the QED 95 will either signal the operating system with a trap or simply ignore the data stored in the cache and refetch the data from main memory. Due to this redundancy, most operating systems will disable the cache parity interrupts by setting the appropriate bit in the Cache Control Register (CCR).

When a Direct Memory Access (DMA) device writes to memory, the CPU checks to see if the address being written is in the cache. If it is, the CPU simply invalidates the cache location. The next time the program requests the data at that location, the CPU will recognize that it is not in the cache and request the data from the memory.

Control bits in the Memory Management Unit (MMU), the CCR and a CPU register allow the cache to be disabled completely, bypassed on data reads only, or bypassed selectively by virtual page address. The CPU never caches the Unibus addresses located in the upper 8 KB of address space.

2.1.3 Floating Point

When configured for floating point instructions, the QED 95 uses a high performance AMD 29C327 floating-point chip assisted by microcode. This part includes a full FP register set and supports F and D data formats. FIS-11 floating point opcodes are presently not recognized by the QED 95 (these exist only on the 11/03 and 11/40).

2.1.4 QMI Bus

The Quickware Memory Interconnect (QMI) is a proprietary bus that connects the QED 95-CPU and the MS95 memory. This bus allows transfers between the CPU and memory at over five MB per second. Support for disk and tape controllers is provided for future expansion.

2.1.5 Boot ROMs

The QED 95 CPU has two boot ROMs that are mapped into the Unibus address space. Each ROM occupies 256 words of Unibus space. The ROMs are paged via the Boot Page Control Register. The CPU supports up to 64 KB ROMs through this

mapping process. In addition, one of the ROMs may be an Electrically Erasable ROM, which allows configuration data to be changed and updated while the board is installed.

The standard configuration is a 32 KB EPROM at location 173000 and an 8 KB EEROM at 165000. Since these addresses are the same as those used by standard Digital bootstrap boards, you can independently disable the ROMs by setting appropriate bits in the Boot and Diagnostic Status Register.

2.1.6 Configuration Menu

In order to eliminate confusion over switch settings, Quickware has provided a configuration menu that enables you to configure the system from the console terminal. Configuration options include boot device selection and on-board device setup.

Some configuration takes place automatically. For instance, if the system has a DL11 terminal interface installed for the console terminal, the boot ROMs and menu automatically disable the on-board SLU.

2.2 MS95 Memory Board

2.2.1 On-board Serial Line Units

There are two DL11-compatible Serial Line Units (SLUs) on the MS95 memory board. They have CSRs located at 17756X and 17650X and vectors at 60 & 300. You can disable the SLUs by setting the appropriate bits in the SLU Configuration Register or through the configuration menu. The baud rates are controlled in the same way.

You can connect the on-board SLUs to terminals, printers, or modems in three different ways. The first method is used when the QED 95 is installed in a PDP-11/24. The 11/24 has connections on the backplane to bring the RS232 signals from the MS95 memory board to a two-port distribution panel mounted on the system cabinet. The second method is via a Quickware cable assembly that connects to J3 on the memory board. The assembly consists of two RS232 cables attached to a 34-pin socket header. The RS232 termination occurs via standard DB25 male connectors. The third method is to use a Quickware Console Interface Module (CIM). These CIMs are either dual or quad boards that connect to J3 on the MS95 board via a ribbon cable. The CIM provides several types of connectors for use with Digital cables. The CIM also allows you to use 20 mAmp devices. The CIM is also used to interface the MS95 memory board to various front panels.

2.2.2 On-board Clock

The on-board clock is compatible with the KW11, the standard Digital line clock used for system time. This same type of clock is found on the DL11-W, which is often used as the console terminal interface.

You can select the clock rate from four sources – 1) AC Line 2) 50 Hz crystal 3) 60 Hz crystal and 4) 800 Hz crystal. The AC line is the standard and most accurate over the long term.

2.2.3 Unibus Interface

The MS95, which acts as a host adapter to connect the QED 95 CPU with the Unibus, adheres strictly to the Digital Unibus standard referenced in the *PDP-11 Bus Handbook*. This ensures compatibility with all Digital peripherals.

2.2.4 Error-Correcting Memory

The MS95 includes 4 MB of Error-Correcting Code (ECC) memory. Using the parity CSR located at 172100, you can disable correction or enable parity-error traps.

An AMD 29C60 Error Detect and Correct chip performs the error-correction. This chip generates the check bits on writes and the error check and correction on reads. The AMD 29C60 detects and corrects all single-bit errors and detects all double-bit errors and most triple-bit errors. This sophisticated error correction scheme protects system integrity better than traditional error-checking techniques. It also increases the system up-time.

Memory reads and writes from the CPU are extremely fast, since they occur over the QMI bus. Reads occur in 400 nanoseconds. Writes occur even faster, by latching write data and allowing the CPU to continue execution before the write is complete.

If you are using the QED 95 system to replace an 11/04/34 or other system with 256 KB or less memory, the operating system may need to be re-sysgened in order to take advantage of the extra memory.

2.2.5 Unibus Map

Quickware has implemented the standard Unibus Map used in all Unibus systems with more than 256 KBytes of memory. The Unibus Map converts traditional 18-bit Direct Memory Access addresses to the 22-bit address space of newer PDP-11s. This allows the use of standard Unibus disk and tape controllers with the 4 MB memory of the QED 95/MS95. The Unibus Map is supported by all DEC operating systems and is transparent to applications running under those operating systems.

An additional feature allows you to disable parts of the Unibus Map in special applications. Use this feature when a peripheral device uses a window in memory for communicating with the application.

2.2.6 Battery Backup

Battery backup is supported via the standard DEC battery backup option, except on the PDP-11/70. Using battery backup on the 11/70 requires the Quickware BB70 board with battery backup option and an MS95 revision E or higher. The BB70 has a sealed battery and can maintain data integrity for up to two hours.

3.8 QED 95E Modules to Upgrade the PDP-11/84

Module	Board Size	Description
95-CPU	Hex	CPU
95-AUX	Hex	Boot Logic and CPU Assist
MS95	Hex	Memory
CIM-84	Quad	Front Panel Interface
PSL-84	Quad	Power Supply Load module

4 Jumper Configuration

The on-board devices are configured through the startup menu; however, several jumpers are provided for configuring the boards for installation in various systems.

4.1 95-CPU Board Jumpers

95-CPU Jumper Selection.

See Page 17 for Jumper Layout	Jumper	Position
PDP-11/70	W2,W3,W4 W5,W6,W8,W9,W10	In Out
PDP-11/24	W2,W3,W4 W5,W6,W8,W9,W10	Out In
PDP-11/04, 34, 35, 40, 44, 60, 84, 94	W2,W3,W4 W5,W6,W8,W9,W10	Out Out

4.2 MS95 Memory Board Switch and Jumpers

The MS95 jumpers consist of both soldered wire jumpers, multi-post movable shunts and switch SW1. The factory default is not set for battery backup. The three-post shunts have two positions. If the shunt is installed on the two posts marked by *IN*, the jumper is active. If the shunt is installed on the middle post and the post marked

OUT the jumper is inactive. Unused shunts may be stored in the inactive configuration.

MS95 SW1 Switch Settings

See Page 17 for SW1 location	Position	
All PDP-11s except PDP-11/70	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	OPEN
PDP-11/70	1,2,3,5,6,7,8,9,10 4	OPEN CLOSE
Note: Positions 5 and 6 are reserved for future configuration options.		

MS95 Solder Wire for Battery Backup

See Page 17 for Jumper Layout	Jumper	Position
No-Battery Backup All PDP-11s	W15, W16 W13, W14	In Out
Battery Backup 11/70	W13, W14, W15, W16	Out

MS95 Jumpers to Select PDP-11 Backplane

See Page 17 for Jumper Layout	Jumper	Position
Backplane 11/24	W11, W12 W1,2,3,4,5,6,7,8,9,10 RN1	Out In In
Backplane 11/04, 34	W11, W12 W1,2,3,4,5,6,7,8,9,10 RN1	In Out Out
Backplane 11/35, 40, 44, 60, 70, 84, 94	W11, W12 W1,2,3,4,5,6,7,8,9,10 RN1	Out Out Out

4.3 95-AUX Board Jumpers

95-AUX Jumper Selection.

See Page 17 for Jumper Layout	Jumper	Position
PDP-11/70	W1, 2, 3, 4, 5 W6, 7, 8, 9	Out In
PDP-11/24	W1, 2, 3, 4, 5 W6, 7, 8, 9	In Out
PDP-11/04 , 34, 35, 40, 44, 60, 84, 94	W1, 2, 3, 4, 5 W6, 7, 8, 9	Out Out

4.6 CIM-84 Jumpers for Minimum Load

The Digital H7204 power supply provides 100 Amps at +5 Vdc, 5 Amps. at +15Vdc and 6 Amps. at -15Vdc. To ensure proper regulation, the power supply must have minimum loads. Quickware makes available the minimum load for the +15 Vdc on the CIM-84 module located in slot 1.

The switchpack (SW1) contains ten individual switches, eight of which are optionally used to select the baud rate of the console SLU and other boot options. Switch positions 9 and 10 are not used. Switch positions 1 through 8 are provided for compatibility with the 11/84 switches, but should normally be left open, as the Quickware Boot Menu controls all of these functions as well. See the Digital *KDJ11-B CPU Module User's Guide* for more information on the Baud Rate Selction, if you cannot use the QED menu.

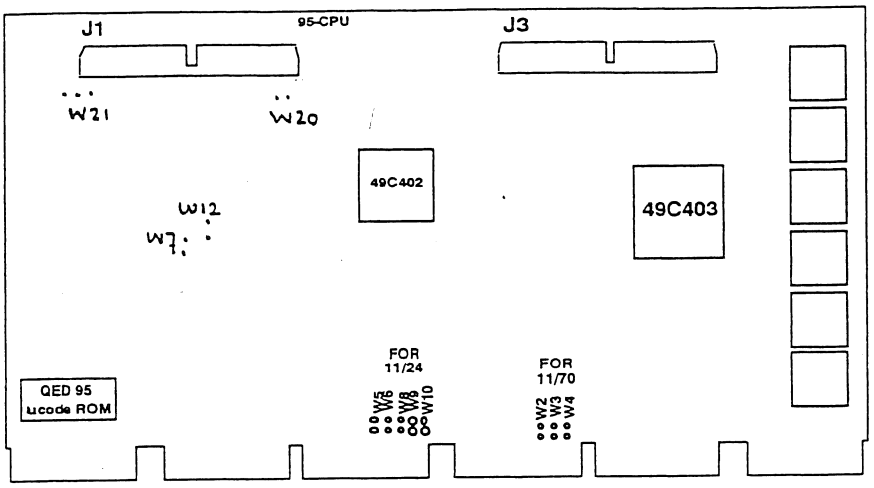
Module	Jumper	Position
CIM-84 Normal operation 44 mA (+15 V load)	W1 W2	In Out
If required 140 mA	W1 W2	Out In
Baud Rate Select (Normal setting) (See Digital KDJ11-B manual)	SW1 thru SW10	Open

4.8 PSL-84 Jumpers for Minimum Load

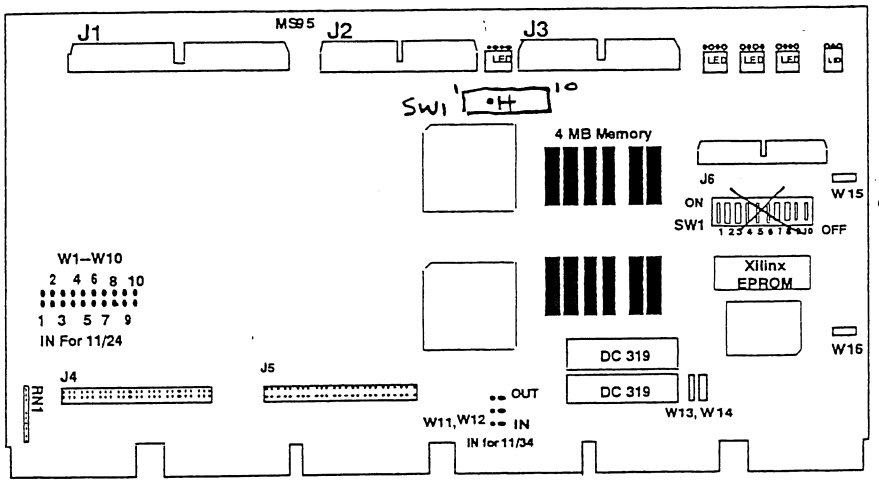
To ensure proper regulation, the Digital H7204 power supply must have minimum loads. Quickware makes available the minimum load for the +5.1 VB on the PSL-84 located in slot 3. The PSL-84 also provides power to the MS95 memory array through the attached cable. This provides battery backup for the memory, if your system has a battery backup unit installed. The regulation of the -15Vdc is maintained by a minimum load from either your installed peripherals, or the Digital M7556 module or the M9049 module inserted in slot nine of the backplane. See the Digital *PDP-11/84 Maintenance Guide* for more information on the M7556 or M9049.

Module	Jumper	Position
PSL-84 Normal operation 1.0 A (+5.1 VB load)	W1 W2	Out In
If required 0.1 A	W1 W2	In Out

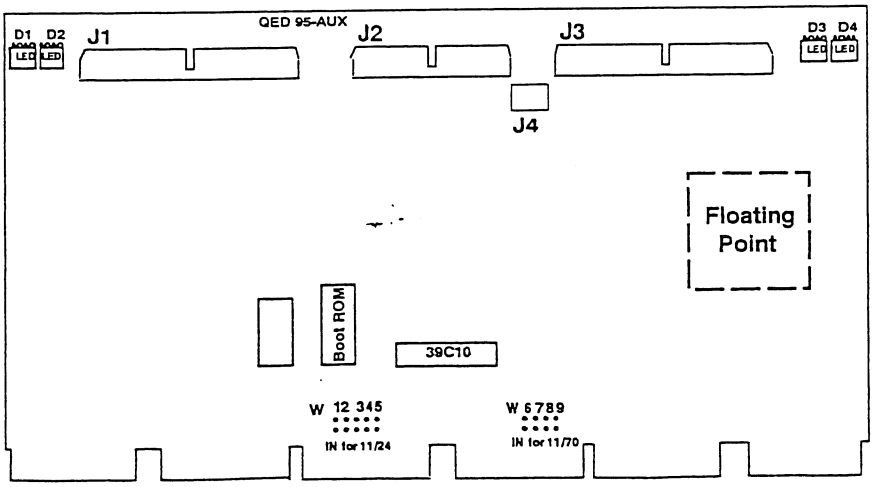
95-CPU

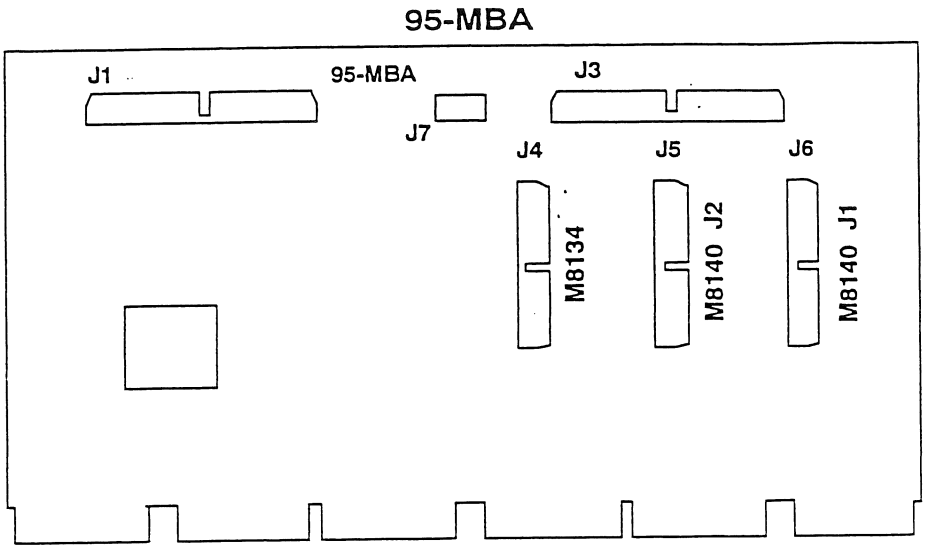


MS95

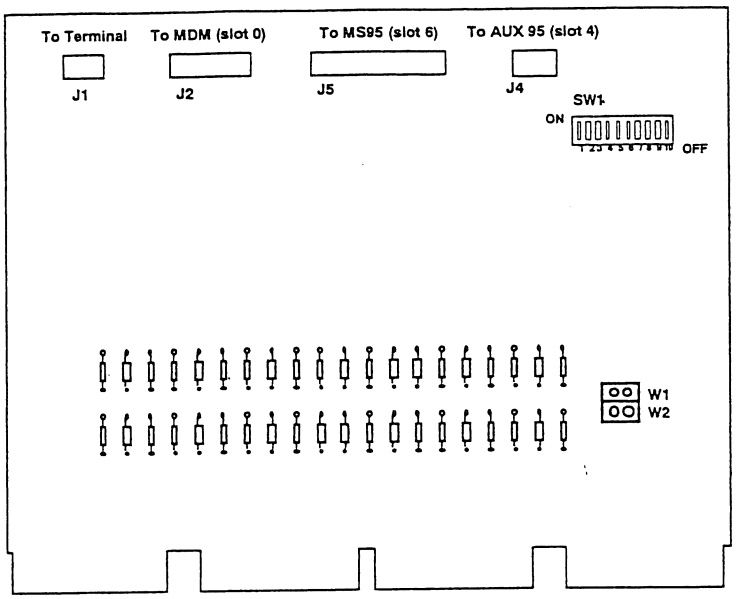


95-AUX

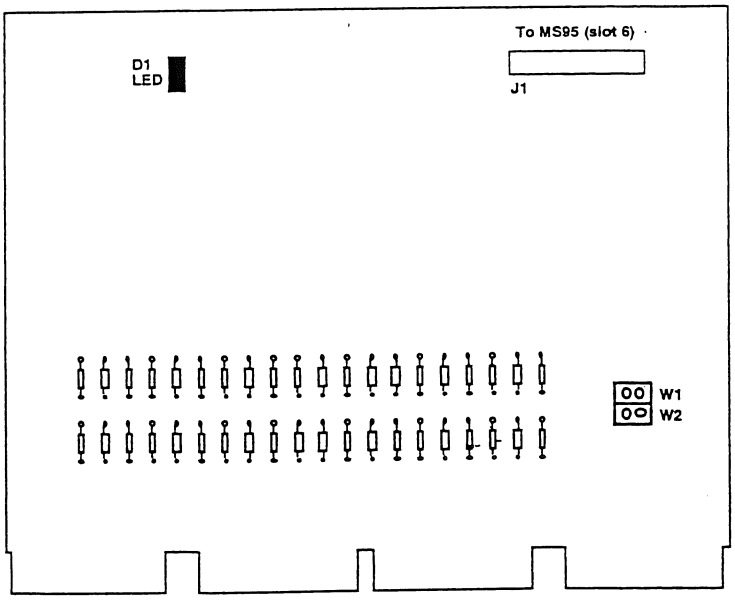



95-MBA
J1
95-MBA
J3
J7
J4
J5
J6
M8134
M8140 J2
M8140 J1

CIM-84


To Terminal
To MDM (slot 0)
To MS95 (slot 6)
To AUX 95 (slot 4)
J1
J2
J5
J4
SW1
ON
OFF
W1
W2

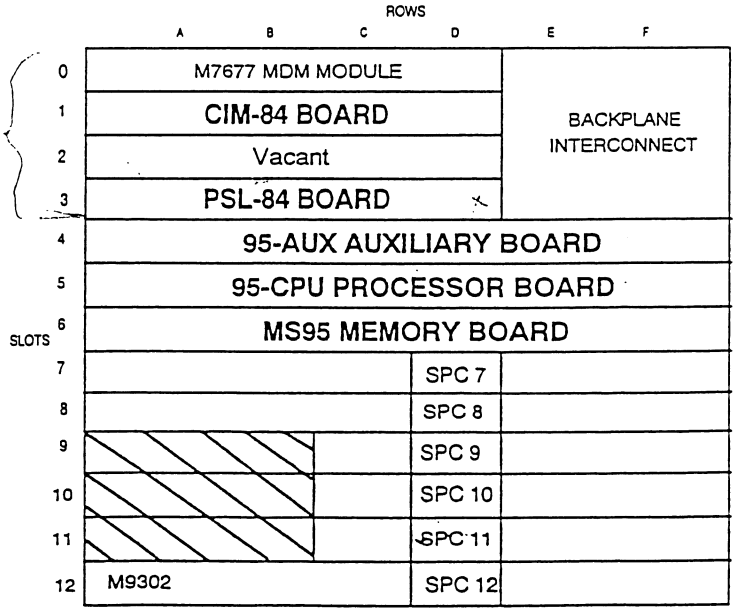
PSL-84


To MS95 (slot 6)
D1
LED
J1
W1
W2

5.8 QED 95E Installation in the PDP-11/84

1. Exercise the PDP-11/84 system to ensure that it is operating properly.
2. Turn off all power to the system.
3. Ensure that the cabinet is properly stabilized before sliding the processor box out.
4. Open the processor box. Remove the quad modules in slots 1, 2 and 3. Set these modules aside, as they are not used with the QED 95E CPU upgrade. **DO NOT remove the M7677 MDM module in slot 0.**
5. Remove the hex modules in slots four, five and six. Set the M8191 (removed from slot 4) aside as it will not be used with the QED 95E CPU upgrade.
6. Install the PSL-84 board in slot 3. Slot 2 remains vacant.
7. Install the CIM-84 board in slot 1. Consult section 5.12 for CIM-84 & PSL-84 cable installation.
8. Install the 95-AUX Processor board in slot four.
9. Install the 95-CPU Auxiliary board in slot five.
10. Install the MS95 Memory board in slot six.
11. Install the cables that connect the 95-CPU, 95-AUX and MS95 boards.
12. You will now have to rearrange the remaining modules in order to reinsert the boards that were removed from slots five and six. It may also be necessary to install another nine slot backplane to accommodate more modules.
13. Close the processor box and push it back into the cabinet. Make sure that no cables are pinched or strained as the system box is slid back into place. Turn on all the power.
14. After turning on the power, the CPU will run self-test diagnostics. This will take approximately 15 seconds. As the tests run, the LEDs on the CPU will cycle. Also, a series of periods will be sent to the terminal. When the tests complete, a prompt ">" is sent to the terminal asking for a carriage return. This carriage return is used to synchronize the baud rates. The first time the system is installed, several carriage returns at one second intervals may be necessary. The configuration menu will then be displayed.

QED 95E Processor Upgrade
Layout in PDP-11/84 Backplane



6 Startup & Configuration Menu

6.1 Front Panel Operation

6.1.7 PDP 11/84 or 94 Front Panel

The 11/84, 11/94 front panel will work when the CIM-84/94 has been installed in slot one of the backplane and connected to the MS95: To reboot, press the Halt/Run/Restart switch to the Restart position. To halt the processor put the Halt/Run/Restart switch in the Halt position. This will put the QED 95 into ODT. To exit out of ODT and continue processing, restore the Halt/Run/Restart switch to the Run position and press the "P" key on the console terminal. See chapter eight for further information on using ODT to examine and deposit memory.

6.2 Bootstrapping the System

The QED 95 has two diagnostic/boot ROMs on-board. The diagnostics run when the power is turned on or the system is rebooted. These diagnostics take 15 seconds to complete. As the tests execute, the LEDs cycle. If an error is encountered, the system will halt, and the code displayed by the LEDs will indicate the type of error which occurred. Appendix C lists the LED codes. At the end of the tests, if the system is coming up for the first time (a cold boot), the memory is cleared and periods "." are printed at the console terminal. After the built-in diagnostics are complete and the memory is initialized, the ROMs print a ">" prompt and wait for

several carriage returns from the console terminal to determine its baud rate. If you are using an existing DL11 interface for the console terminal, there is no need to determine the baud rate, and so the program will move on to the next step.

Next, the menu program is loaded into memory, starting at location 100000. The menu will print either the full menu or the device prompt. If you receive the device prompt and wish to run the full menu, type "**QM**" [RETURN] to see the full menu. The first time you run the QED 95 in a system, you should use the full menu to verify the correct settings.

Full Menu Screen Display:

```
   QED90/95 Configuration Program
   Copyright Quickware Inc., 1988-1991
   Version 2.02

   Cache memory fully ENABLED
   4 MB on-board memory
   Boot to: MENU
   [Additional parameters here...
   ...]
   1 = Restore Default Settings     D = Display Settings
   2 = Customize Settings           S = Save Configuration
   V = Display Version              P = Proceed to Boot
Prompt
   H = Halt to ODT                  T = Boot using M9301/12
   X = Exit
   Selection: _
```

Boot Device Screen Display:

```
   QED90/95 Configuration Program
   Copyright Quickware Inc., 1988-1991
   Version 2.02

   Boot Device [DL0] ? _
```

Section 6.3 below explains the various options of the full configuration menu. To boot a disk or tape when you are in the full menu, select the "P" option. The prompt "Boot Device ?" will then be displayed. If you type a question mark "?" the system will display a list of supported boot devices. Those device whose CSRs are found at standard locations are displayed in parentheses. The device "QM" is the Quickware Menu. Typing **QM** [RETURN] will return to the menu. Any other device code may be typed. All boot ROMs default to unit zero. This may be overridden by designating the desired unit.

Examples of responses to the Boot Device prompt:

QM [RETURN]

Displays the main menu for configuration.

DL [RETURN]

Boots RL01/02 unit zero.

DL2 [RETURN]

Boots RL01/02 unit two.

When you enter a valid boot command, the menu asks if you would like to save the command as the default. Typing "**Y**" will save the command. Typing "**N**" or [RETURN] will not save it.

The program will then attempt to boot the requested device. If the device is not ready, the message "Retrying. Type ^C to abort retries." will be displayed. If you wish to abort the command, type the "**C**" key while holding the CTRL key down. This will return you to the previous prompt; otherwise the program will continue to retry the device until it succeeds in booting.

When control is transferred to the operating system, the message "Booting..." will be displayed.

6.3 Using the Configuration Menu

The configuration menu is used in place of switches to set up the QED 95. The menu controls on-board Serial Line Units, on-board clock, memory size, and action to take at boot time.

This section will describe each of the menu options and sub-menus in turn.

6.3.1 Restore Default Settings

This option of the main menu allows you to easily set up the QED 95 for typical configurations. Seven standard configurations are presented, allowing you to choose which type of system the QED 95 is installed in, and which processor the QED 95

should emulate. There are four distinct systems in which the QED 95 may be installed, as follows: 1) PDP 11/24 systems, 2) PDP 11/44 systems, 3) PDP 11/70 systems and 4) all other systems. In 11/44 and 11/70 systems, you also have a choice of the processor emulation. You may choose to have the QED 95 emulate the original processor, or, for certain applications, you may want the QED 95 to emulate a J-11 processor instead. In 11/24 and all other systems (excepting 44s and 70s) the QED 95 will emulate a J-11 processor. The table on the next page describes the default settings.

We recommend that if you have an 11/44, you choose the normal 11/44 emulation, rather than the J-11 emulation. Also, if you have an 11/70, we recommend that you choose the 11/70 (A/full) emulation, rather than the C/J11 option or the B/Limited MMU option.

Option	11/24,44	11/70, other
SLU 1	Enabled	note 1
Halt on Break	Enabled	Disabled
SLU 2	Enabled 9600 Baud	Disabled
Line Clock	Enabled, Line Clock	note 2
Boot Action	Device Prompt	Device Prompt
Auto-Baud T.O.	No Time Out	No Time Out
Memory	All available QED memory	All available QED memory
Processor	note 3	note3

Notes:

1. If the system has a DL11 console board, such as the DL11-W (M7856), the on-board SLU 1 is disabled; otherwise the SLU 1 is enabled and the baud rate is set automatically at boot time.

2. If the system has a KW11-compatible clock, such as on the DL11-W (M7856), the on-board clock is disabled; otherwise the on-board clock is enabled and set to use the clock pulse provided on the backplane.

3. When you choose an 11/24 or other PDP-11, the QED 95 will emulate a J-11 processor. When you choose an 11/44, you can choose to emulate either an 11/44 processor or the J-11. When you choose an 11/70, you can choose to emulate either a J-11, or an 11/70 with a full or a limited MMU.

6.3.2 Display Version

This option displays the current version of the Boot Roms and the Menu program. Section 6.4 lists the changes in each version.

6.3.3 Display Settings

This option displays the settings you have currently chosen. Note that if you have changed settings, they will not be saved and made permanent until you select the Save Configuration option. A message will be displayed to indicate that you have modified the configuration, and that you should save it to allow the changes to take effect.

6.3.4 Save Configuration

After you have made your selections, store the configuration in the EEROM by choosing this option. Please note that the settings will not take effect if you exit the menu without first saving the changes. The message "`Configuration has been modified, remember to Save it.`" will display in the main menu whenever there are unsaved changes.

6.3.5 Proceed to Boot Prompt

The boot prompt allows you to select a device for booting. The prompt "`Boot device {[DD0]} ?`" asks you for the device to boot. The current default, if any, will be displayed in the "[ ]" brackets. Pressing RETURN selects the default. You can override the default by typing the device code and an optional unit number followed by RETURN. (Example **DL2**). If a device is entered, the question "Save for later auto-boot?" will ask if you want to store the device code as the new default.

To see a list of supported and installed devices type a "**?**." All available boot programs are displayed, including those in any M9312 that is installed. Devices that are installed at standard addresses are displayed in parentheses "( )". The table on the next page shows the boot programs provided.

In addition to booting the devices at the standard addresses, a device installed at an alternate address may be booted using the **/A:n** option. If, for instance, a second RM02 controller was installed at address 176300, drive zero on this second controller could be booted using the syntax "**DB0/A:176300**", while drive three would be booted using "**DB3/A:176300**".

Some devices from third parties come with boot roms on their controller boards. If you want to boot using these roms, you would use the **FB** device code to indicate a "foreign boot rom" and specify the boot starting address using the **/A:n** option. For example, to boot a controller rom that is at address 171000, you would use the command "**FB/A:171000**".

If you need to boot a device which is not supported by the Quickware Boot Roms (see the next page for the table of supported devices), and you do not have a separate boot rom for it, you can enter the boot program into the EEPROM, where it will be

stored, and can be run at boot time by using the UUn pseudo-device code. Four user boot programs, UU0 through UU3, may be stored in the EEPROM. See section 6.3.8.5 for more information on writing and storing user boot programs.

Code	Device	Address of CSR	Notes
QM		N/A	Use to exit back to the Quickware menu.
DB	RM02/3/5	176700	RP04/5/6
DK	RK05/03	177400	
DL	RL01/02	174400	
DM	RK06/07	177440	
DP	RP02/03	176700	
DS	RS03/04	172040	
DU	MSCP	172150	MSCP class disk drives such as RA80, RA60...
DX	RX01	177170	
DY	RX02	177170	
DD	TU58	176500	If SLU 2 is enabled, this may appear to be installed.
MM	TU16 TE16 TM02/03	172440	
MS	TS04/11	172520	This may appear to be installed due to a conflict with MT tape drives.
MT	TU10 TE10/TS03	172522	
MU	TMSCP	174500	TMSCP class tape drives such as TU81, TK50...
RF	RF11	177460	
FB		use /A: n option	This is provided to boot Foreign Boot roms, such as might be supplied by third party peripheral vendors. These may be booted by specifying their address. e.g. FB/A:171000 or FB/A:175000
UU		N/A	This is the code for a User-defined boot program. If you select this without first storing a boot program, unpredictable results will occur.

Devices supported by the QED Boot Roms

6.3.6 Boot using M9301/9312

This option of the main menu will allow you to boot a device using a boot program from a 9301 or 9312 boot board, if one is installed. If the boot program for a device is already present in the QED boot roms, (see table above in section 6.3.5) you should consider using that program, rather than the 9312 version, since the QED versions have better error recovery. If you do need to use a boot program installed on a 9301 or 9312, this option will allow you to boot the program on the boot board according to the switches on the boot board. If you wish to boot some other device (other than the one selected by the boot board switches) you can run any program on the boot board by specifying its address with the "FB" pseudo-device. e.g. "**FB/A:173204**" This command would boot a "foreign-boot" device (i.e. a 9312 board) using 173204 as the starting address.

6.3.7 Halt to ODT

After you select this option, the processor will halt, and ODT will print out a prompt "@". See chapter eight for more information on using ODT to examine and to deposit to the memory and to the registers. Type a "**P**" to proceed from ODT back into the menu.

6.3.8 Customize Settings

This option allows you to change the configuration from that selected by the standard defaults above.

6.3.8.1 Configure DL11 Ports

The MS95 has two DL11-compatible serial line units. They are referred to as SLU 1 and SLU 2. The CSR address and vector of SLU 1 is set at the standard console address of 177560/60. SLU 2 is located at the first DL11 address 176500 with a vector of 300. These addresses cannot be changed.

You can set the baud rates for SLU 1 and SLU 2 through this part of the menu. Each SLU can be individually disabled. It is generally unnecessary to change the baud rate for SLU 1, since it is set at boot time by the boot ROMs.

You can halt the QED 95 CPU from the console by pressing the break key. In order for this feature to be used, the console terminal must be connected to SLU 1. This is the standard configuration in the 11/24 and 11/44. To use this feature in other systems such as the 11/34, the existing console DL11 board must be removed and the console terminal must be connected through J3 on the MS95. The 11/24 or 11/44 can also control the Halt-On-Break option by setting the front panel key switch to "Local Disable." There is no such override in other systems. We recommended that you disable this feature for these other systems. This action prevents accidental halting of

the system while it is on-line. Please note that if you enable Halt on Console Break, the system will halt anytime a break is received from the console terminal, regardless of the setting of the 11/24 or 11/44 front panel. If you disable Halt on Console Break in an 11/24 or 11/44, then the front panel switch on these machines will control this feature. Also, note that when you turn off the console terminal, it may send a break, in which case the system will halt if this option has been enabled.

6.3.8.2 Configure KW11 Clock

The MS95 has a KW11-compatible line clock. This option allows you to disable the on-board KW11 or to set its clock source. This clock should be enabled in 11/24 and 11/44 systems. In most other systems this should be enabled if the system clock has been removed during installation of the QED 95. If the clock is enabled and the system clock is still installed, the system time may advance at double its normal rate.

If the clock is enabled, four sources for the clock rate are available. The default and recommended rate is the AC Line. This is used in most systems and will assure that the clock rate is consistent with the original system. The other three sources are from a crystal oscillator. The available rates are 50 Hz, 60 Hz, and 800 Hz.

6.3.8.3 Configure Cache, Processor Registers, and Memory

This sub-menu provides six options, most of which do not usually need to be changed. The first option controls the memory size, but the QED 95 automatically sets the memory size at boot time, so this should not be necessary. The second option is used to enable the Extended Unibus, if you have a 2 MB MS95 installed in an 11/24 or 11/44, and wish to use some of the original EUB memory. (This option is no longer recommended or supported.)

The third option may be used to disable the cache for diagnostic purposes. The cache has three settings - Fully Enabled, Cache Instructions only (No data), or Disabled. For normal operation, the cache should be fully enabled.

The fourth option is used to configure the MS95 for special hardware requirements. If you need to disable extra pages of the Unibus Map, you would use this option. Also, this option can be used to force the MS95 to run only in 18-bit mode, although this would limit the memory size to 256 KB. If you need assistance in using either of these functions, please call our technical support department.

The last two options on this sub-menu are provided to adjust the processor-specific features of the QED 95. Option five, "Configure Switch Register" will allow you to make the switch register nonexistent, read-only, write-only, read and write to the 11/70 front panel, or read and write to an internal register. If you choose to emulate an 11/70, this will already have chosen the 11/70 front panel to be used as the switch register.

The last option, "Configure Quickware special registers", allows you to change many of the special internal registers which are used by the microcode for emulation

variations. You should not need to change any of these registers, except perhaps the System ID register. This register contains the value which can be read at location 177764 in an 11/70 system. On the original PDP-11/70, there was a set of switches which allowed you to change this value, as a type of serial number for each machine. Sometimes, in networked applications, this number is used to uniquely identify each machine. If you need this feature, you can change the System ID register to any value you require. Note that this value is distinct from the value returned by the MFPT (Move from Processor Type) instruction. The microcode for MFPT uses the value in IReg720 to decide what value to return. When the QED 95 is emulating a J-11, this register contains a 5, when emulating an 11/44, it contains a 1, and when emulating an 11/70, it contains a 0, which means that MFPT is not a legal instruction.

6.3.8.4 Select Boot Action

You choose the default boot-time behavior of the QED 95 using this menu selection.

Set Boot to Device Prompt

Selecting this boot action is the same as typing "P" at the main configuration menu prompt. When the QED 95 boots, it will prompt: "`Boot device [DDn] ?`" You can type "?" to see which devices are available. For each supported device, the program attempts to reference the standard CSR for the device. For each device that it finds, the mnemonic is typed in parentheses.

The prompt may also have a default saved with it. This is done after selecting a device at boot time by responding "Y" to the question "Save response for later auto-boot? ". If answered "Y," the response to the "Boot Device?" prompt is stored in the EEROM. It is then used as the default for the boot prompt or used in the auto-boot routine described below.

Set Boot to this Menu Program

This option would boot the main configuration menu each time you turn the system on. Generally, you won't want to see the menu each time you boot the machine, so you would choose boot to device prompt or auto boot, rather than this option.

Set Boot to Auto-Boot

You can set the configuration program so that it boots a selected device automatically. To do this, boot the device from the boot device prompt and save the response. Then select "Boot to Auto-Boot" as the boot action. When you reboot the system, the selected device will boot automatically. There is a time-out of about 5 seconds during which you can cancel the auto-boot process with a Ctrl-C. If the desired device is unavailable or does not become ready, the auto-boot procedure will be cancelled.

Set Boot to M9312/M9301

If the system has an M9301 or M9312 bootstrap terminator board installed, the QED 95 can use those ROMs to execute the boot. Since these bootstraps are located at the same address as the QED 95 boot ROMs, the QED 95 boot ROMs are disabled, and then control is passed to the M9301/12. Location 17773024 on the bootstrap board contains the starting address for these ROMs.

Normally it is unnecessary to use the M9301/12 boot ROMs since most devices are supported by the configuration menu. Use this option only if your boot device is not supported by the QED boot menu.

Set Auto-Baud Time Out

On a cold boot, the boot ROMs check to see if there is a DL11 installed in the system. If so, control is transferred according to the option selected in 6.3.8.4. If there is not a DL11, SLU 1 is enabled and the ROMs begin to wait for a RETURN from the console terminal. The ROMs use the RETURN to synchronize the baud rate. After receiving the RETURN, the configuration menu is loaded and control is transferred according to the option selected in 6.3.8.4. This option allows you to set the amount of time that the ROMs wait for the RETURN. If the time-out elapses, the previous setting for SLU 1 is loaded from the EEROM and the configuration menu appears.

We recommend that you select a time out of 15 seconds.

Set Device Timeout

This option selects how long the Auto-Boot process will wait for a device to become ready before giving up on it. After the specified period of time, an auto-boot is canceled, and the menu prompts for another device to boot.

6.3.8.5 Edit User Defined Boot Programs

If you require a special boot program that is unavailable on an M9301/9312 bootstrap terminator, you may load a program into the EEROM and select it from the boot prompt. You can define up to four user defined boot programs, each 510 bytes long. If you require a longer boot program, you can chain the four programs together. The user boot programs are named UU0 to UU3. They appear as one device mnemonic in the device list. There is no support for multiple devices in one boot program.

To load a new program into a user boot, first retrieve the desired boot program. The program will copy the old EEROM contents into memory from 1000 to 1774 and then halt with the ODT prompt "@." Using ODT, type the user boot into locations 1000 (8) through 1774 (8). Type "P" to exit ODT. Next select "Store user boot." You will be prompted for the device to save. If the EEROM save is successful, the program responds with "User Boot Saved." The user boot program can now be run by exiting back to the main menu and selecting "UUn" at the boot prompt.

Guidelines for writing user boot programs:

1. Write code in position-independent form.
2. Note that, although boot programs are individually edited at locations 1000-1774, all four are loaded into memory from 10000-13776 before they are executed.
3. When selected, the user boot program begins running at 1n000 (8), where n is the UUn unit number. That is, UU0 will be in memory from 10000-10776, UU1 from 11000-11776, etc.
4. Each program will be stored on a separate page in the EEROM at 17765000 (8).
5. The program may not exceed 776 (8) bytes unless it is broken into segments. Please note that location xx776 is reserved for a checksum, and should not be used by the program code. Multi-segment boot programs should branch around this location.
6. To load a segmented boot program, you must store each segment separately in the EEROM.
7. Use the command "RTS PC" to return to the configuration menu.
8. On return to the configuration menu, R5 has the following meaning:

   1 = Successful boot of first block.

   270 = Drive not ready.

   271 = Non-bootable media present.

   272 = No disk present.

   273 = No tape present.

   274 = Non-existent controller.

   275 = Non-existent drive.

   276 = Invalid unit #.

   277 = Invalid device

   300 = Controller error.

   301 = Drive error.

   If 1 is returned, the configuration menu will continue with the boot by jumping to location 0 (8). The other errors result in an error message and the boot is retried.
9. If control is returned to the configuration program to continue the boot, R0-R3 will be preserved.

10. Magtape boots generally require R4 point 20 (8) bytes past the ASCII device code. This is to maintain compatibility with M9301/12 boot ROM formats. ***R4 is not preserved on a return from the user boot programs***. Because of this, it is advisable to jump directly to 0 (8) from the user boot program for magtape boots.

11. A checksum is stored with each user boot, in word xxx776. If the EEProm area containing a user boot becomes corrupted, the self-diagnostics will write a 0 (halt) instruction as the first word of each corrupted user boot. In this way, the machine will halt, rather than executing a damaged boot program.

6.4 Revisions

V2.01

This was the first version of the boot roms for use with the QED 95.

1. Fixed bug which was clearing PSW in memory clearing routine.
2. Added boots for RK05 (DK), RP03 (DP), RS03 (DS), and RX01 (DX).
3. Increased maximum legal unit number for several devices.

V2.02

1. Fixed bug in LTC enable/disable routine.
2. Changed so that maximum memory on MS95 is used by default.
3. Changed some menu options for QED 95 CPU.
4. Added boot for MU (TMSCP tape drives)
5. Changed UU boots, so that they are copied down into memory at 10000 - 13776 before being executed.

V2.03

1. Fixed device timeouts for auto-boot.
2. Fixed MU boot delays.
3. Enabled odd address trapping by default.

8 Diagnostics

This chapter describes the software tools that are available to help you diagnose the QED 95 system. First, we describe ODT, which allows you to examine and modify memory and registers. Then, the built-in diagnostics are discussed. Next, some simple diagnostics are presented which can be entered using ODT, allowing you to verify the data paths of the machine. Larger diagnostic programs are discussed next, including those written by Quickware and by Digital. Finally, the last section details some troubleshooting techniques which may be helpful.

8.1 ODT

The QED 95 processor has an ODT (Octal Debugging Tool) built into it. This ODT allows you to examine memory, deposit values into memory, and execute programs.

Commands include:

n/ **n** may be an octal address, a register specified by R0 thru R17, or RS for the PSW. This will open and display the contents of the specified address, and allow you to enter a replacement value. Carriage Return closes the location, modifying it if you typed a new value, while Line Feed closes (and saves, if necessary) the location and opens the next sequential location. If **n** is not specified, the last address is used.

Anything with an **R** or a **$** will be treated as a register address. (only the low four bits are significant).

Anything with an **S** will be treated as a PSW reference.

P Proceed with program execution at the current PC address. This will restore MMR0 from the R10 scratch register. No bus reset will be issued.

C Continue with program execution at the current PC address. This will not restore MMR0 from R10, rather, the current contents of MMR0 (777572) will be used for the initial MMR0 contents. No bus reset will be issued.

nG Reset the processor and the Unibus, and begin execution at octal location **n**. If **n** is not specified, a default value of 0 is used. Note that MMR0 is cleared, and that a bus Init is part of this sequence.

D Dump the PSW and the eight general registers and eight scratch registers.

Q Force DC LO out onto the bus. This will initialize the bus, reset the system, and restart the boot code.

^S Control-S is ignored.

^Q Control-Q is ignored.

^U Control-U or backspace or delete or any other unknown character will echo a '?' and cancel the input line.

If power fails while the QED 95 is in ODT, an 'F' will print out on the terminal.

If an illegal address is referenced, or any other memory error happens while referencing a memory location, a question mark will print out, and the prompt will be re-displayed.

Addresses and data are always entered and displayed in octal (base 8).

Addresses are always treated as 16 bit values. If you type in more digits, only the lowest 16 bits are used. Thus, 7707123456 is treated as 123456. As a special feature, all addresses are translated using the current state of the MMU. Thus, if the MMU is enabled (bit 0 of MMR0 is on), the PARs selected by the mode in the high bits of the PSW will be used to relocate the address. This may be helpful in debugging, but one must take care that PAR7 of the current mode is always mapped to the I/O page, so that ODT references to the terminal will work.

Note that MMR0 is cleared upon entry to ODT, but a copy of the prior contents is saved away in R10, a scratch register. R10 and MMR0 may be independently modified. R10 will be copied into MMR0 when a **Proceed** command is issued.

Eight scratch registers may be accessed by using the **D** command, or by looking at registers R10 through R17. These are associated as follows:

R10 The value of MMR0 at the time ODT was entered. (777572)

R11	The current MMR1 contents. (read-only) (777574)
R12	The current MMR2 contents. (read-only) (777576)
R13	MMR3 (772516)
R14	Memory system error register (777744)
R15	Cache control register (777746)
R16	System Error register (777766)
R17	Switch register contents (777570)

8.2 Built-in Diagnostics

When the QED 95 system is booted, it runs some diagnostics from the built-in boot roms. These tests verify that the processor is operating and can continue the boot process. Next, the QED 95 boot roms test the memory. If the contents of memory are not valid (as would be the case on a cold boot), then the memory is cleared, starting with the highest memory address, and working down to 0. As the memory is cleared, periods print out on the console terminal. After the memory is cleared, the menu program is loaded into memory and run. During a warm boot, when memory does not need to be cleared, the processor still performs some limited tests, and then proceeds with the boot without clearing memory.

If the built-in diagnostics detect an error, the LEDs on the QED 95 CPU will display a code which indicates the test that failed. Refer to Appendix C for a table of the error codes, and the associated tests and likely causes of failure.

8.3 Simple Diagnostics

This section describes some simple tests which you can try by typing them into the QED 95 using ODT. These tests also demonstrate how ODT may be used.

(computer responses are underlined, your keystrokes are not.

<CR> *means type a Carriage Return,*

<LF> *means type a Line Feed,*

nnnnnn *means any octal number)*

8.3.1 Memory Tests

The most basic test of the system is to access the memory. We will try to access location 0, which should contain memory on every system, and then try to access a

non-existent memory location. Lastly, we will examine the CPU error register, to see if the CPU can access built-in registers.

With the system in ODT (@ prompt) (enter ODT by pressing the HALT switch, typing a break, or by selecting H in the main menu), try the following:

```
@0/nnnnnn 0<CR>
@17772130/?
@17777766/000000 0<CR>
@
```

If the above test functions correctly, the system is working well enough to try the following tests. If not, check the cables for proper connections.

8.3.2 Memory Bus Tests

One very simple test is to try to store several words in the memory and retrieve them, checking that nothing was garbled in between. Again, in ODT, try the following:

```
@0/nnnnnn 0<LF>
000002/nnnnnn 1<LF>
000004/nnnnnn 2<LF>
000006/nnnnnn 4<LF>
000010/nnnnnn 10<LF>
000012/nnnnnn 20<LF>
000014/nnnnnn 40<LF>
000016/nnnnnn 100<LF>
000020/nnnnnn 200<LF>
000022/nnnnnn 400<LF>
000024/nnnnnn 1000<LF>
000026/nnnnnn 2000<LF>
000030/nnnnnn 4000<LF>
000032/nnnnnn 10000<LF>
000034/nnnnnn 20000<LF>
000036/nnnnnn 40000<LF>
```

```
000040/nnnnnn 100000<CR>
@
```

The first seventeen memory locations are now filled with distinct patterns, each testing a separate data bit. If you examine them, you should see the following:

```
@0/000000 <LF>
000002/000001 <LF>
000004/000002 <LF>
000006/000004 <LF>
000010/000010 <LF>
000012/000020 <LF>
000014/000040 <LF>
000016/000100 <LF>
000020/000200 <LF>
000022/000400 <LF>
000024/001000 <LF>
000026/002000 <LF>
000030/004000 <LF>
000032/010000 <LF>
000034/020000 <LF>
000036/040000 <LF>
000040/100000 <CR>
@
```

If you do not get these results, the memory data bus has a problem at some point. If the processor prints out a question mark at any point, it is indicating that it can not access the memory. Check that the memory is correctly installed. If the numbers printed out are different than the ones you entered, then there is likely a memory data ram error.

Another simple test is to exercise each of the address lines. In this test, we will write to addresses in memory so that every memory address line is tested. Enter the following, while in ODT:

```
@0/nnnnnn 0<CR>
@2/nnnnnn 1<CR>
@4/nnnnnn 2<CR>
@10/nnnnnn 3<CR>
@20/nnnnnn 4<CR>
@40/nnnnnn 5<CR>
@100/nnnnnn 6<CR>
@200/nnnnnn 7<CR>
@400/nnnnnn 10<CR>
@1000/nnnnnn 11<CR>
@2000/nnnnnn 12<CR>
@4000/nnnnnn 13<CR>
@10000/nnnnnn 14<CR>
@20000/nnnnnn 15<CR>
@40000/nnnnnn 16<CR>
@100000/nnnnnn 17<CR>
@
```

Now we will verify that the memory has stored our data at the correct addresses.

```
@0/0 <CR>
@2/1 <CR>
@4/2 <CR>
@10/3 <CR>
@20/4 <CR>
@40/5 <CR>
@100/6 <CR>
@200/7 <CR>
@400/10 <CR>
@1000/11 <CR>
@2000/12 <CR>
@4000/13 <CR>
@10000/14 <CR>
@20000/15 <CR>
@40000/16 <CR>
```

```
@100000/17 <CR>
@
```

If there are no errors, the memory system is working, and none of the data or address lines are corrupted. Also, the CPU and the SLU are basically functional.

8.3.3 CPU Tests

To test the CPU, we will first set the registers, examine them, and then run a very short program which will modify the registers.

First, try to set the registers:

```
@R0/nnnnnn 0<LF>
R1/nnnnnn 1<LF>
R2/nnnnnn 2<LF>
R3/nnnnnn 3<LF>
R4/nnnnnn 4<LF>
R5/nnnnnn 5<LF>
R6/nnnnnn 1000<LF>
R7/nnnnnn 1000<CR>
@
```

Now examine them, and verify that they contain the data we wrote:

```
@R0/000000 <LF>
R1/000001 <LF>
R2/000002 <LF>
R3/000003 <LF>
R4/000004 <LF>
R5/000005 <LF>
R6/001000 <LF>
R7/001000 <CR>
@
```

Now, write a simple program to manipulate the registers.

```
@1000/nnnnnn 5005<LF>                (comments are off to the side)
001002/nnnnnn 5201<LF>               (do not type these)
001004/nnnnnn 10203<LF>              ; clear R5
001006/nnnnnn 60302<LF>              ; increment R1
001010/nnnnnn 77402<LF>              ; move R2 to R3
                                     ; add R3 to R2, store in R2
001012/nnnnnn 20004<LF>              ; decrement R4 and repeat
001014/nnnnnn 1401<LF>               ; the add until R4=0
001016/nnnnnn 5201<LF>               ; compare R4 and R0
001020/nnnnnn 0<CR>                  ; skip next inst if R0=R4
@                                    ; increment R1
                                     ; halt.
```

Execute the program by typing:

```
@1000G
001022
@
```

The program should halt within a second, and print the above response. If it does not, halt the system using the HALT switch. Either the system is not working, or the program was entered incorrectly. If the program does halt, examine the contents of the registers, to see if the program worked correctly.

```
@R0/000000 <LF>
R1/000002 <LF>
R2/000012 <LF>
R3/000002 <LF>
R4/000000 <LF>
R5/000000 <LF>
R6/001000 <LF>
R7/001022 <CR>
@
```

If there are no mistakes, the CPU, memory, and SLU are functioning.

8.3.4 Interrupt Tests

Testing the interrupt system will complete the hand-entered tests of the QED 95 CPU. Various I/O devices interrupt the CPU when they need attention. The device requests the CPU's attention, and when the CPU grants the request for attention, the device responds with a vector at which to find an address to begin execution of a service routine. We will enable the SLU transmitter to interrupt the CPU when it is ready to transmit another character. When the CPU responds, it will send another character to be transmitted. If this works, the CPU is able to respond to interrupts.

Enter the following:

```
@1000/nnnnnn 12737<LF>
001002/nnnnnn 1014<LF>
001004/nnnnnn 177746<LF>
001006/nnnnnn 12737<LF>
001010/nnnnnn 100<LF>
001012/nnnnnn 177564<LF>
001014/nnnnnn 777<CR>
@2000/nnnnnn 12737<LF>
002002/nnnnnn 120<LF>
002004/nnnnnn 177566<LF>
002006/nnnnnn 2<CR>
@60/nnnnnn 62<LF>
000062/nnnnnn 0<LF>
000064/nnnnnn 2000<LF>
000066/nnnnnn 0<CR>
@R6/nnnnnn 1000<CR>
@
```

Now execute the program by typing:

```
@1000G
PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
@
```

The program will continue to print the letter **P** until you press the HALT switch. If no output appears, the processor is not responding to interrupts.

8.4 Distributed Diagnostics

Quickware has available a diagnostic package which includes several diagnostic programs written by Quickware to test the QED 95 system, and also includes the Digital XXDP package, distributed under a license agreement.

8.4.1 Quickware Diagnostics

Quickware provides diagnostic programs to test the QED 95 and associated hardware. Programs written specifically for the QED 95 are named **Qtttrp.BIC** where **ttt** is the test name, **r** is the revision level, and **p** is the patch level. Also, Quickware distributes some DEC diagnostic programs which have been patched to work with the QED 95. These have a **Q** for the patch level.

The following table lists the currently available diagnostics. Consult the file **QFILES.TXT** on your distribution media for the most current information.

Name	Description
QCFGD0	Processor identification
QCP1D0	Processor data bus and instruction tests
QCP2D0	Processor data sensitivity test
QCP3D0	Extensive Processor instruction test
QFP1D0	Floating Point test
QFP2D0	Extensive Floating Point test
QCA1D0	Cache memory size test
QCA2D0	Cache memory full test
QSL1D0	Serial Lines 1 and 2 tests
QLTCD0	Line Time Clock tests
QMS910	Memory test

A command file is supplied to automatically run all of the tests. To start it, type the following:

```
.C Q95TST
```

If you want to only run one pass of each test (to quickly verify that everything works) type

```
.C Q95TST/QV
```

The individual tests may be run by typing **R** *filename* for example:

```
.R QCP1D0
```

All Quickware diagnostics follow a similar format: They will print the program title, and then ask for a new switch register setting. Typing a carriage return will enter the default setting, which is 2000, bell on error. Typing a question mark will list all the values defined for the switch register. These are octal values which may be added together, for example, to select both Halt On Error and Bell On Error, you would enter 102000 followed by a carriage return. The diagnostics will run for ten passes and then return to the XXDP monitor. If you wish to change the switch register while the diagnostics are running, type a control-G to force the program to prompt for a new switch setting. To stop the diagnostics, type a control-C.

The Quickware-patched DEC diagnostics and QMS910 are designed to be run until the processor is rebooted.

8.4.2 DEC Diagnostics

DEC diagnostics for peripherals should all function normally. These are named **Ztttrp.BIC** where **ttt** is the test name, **r** is the revision level, and **p** is the patch level. If problems do occur with some diagnostics, it may be the result of the QED 95 CPU's high speed finishing some timing loops faster than expected. We recommend that you disable the cache memory, and try the test a second time. See section 6.3.8.3 for help in disabling the cache. Disabling the cache drastically reduces the CPU speed.

You can build a DECX diagnostic exerciser to test your entire system. The DECX modules are all named **Xtttrp.OBJ**, and the program DXCL is used to build an exerciser. If you are not familiar with this process, please see the help files QXHELP.TXT and QXHLP1.TXT on the diagnostic medium. Note that the DECX module FPB* which tests the floating point unit will not work properly, due to the

asynchronous floating point operation of the QED 95. Modules FPA, CPA, CPB, KWA, BMH, and others work fine.

DEC diagnostics for specific processors or memories will generally not work. Most of these examine processor-specific bits, and will find subtle differences between the DEC processors and the QED 95. The supplied Quickware processor tests should be used to test the processor instead.

8.5 Troubleshooting and Fault Isolation

The following steps will assist you in troubleshooting a QED 95 system. There are two major categories of problems, those which prevent the machine from booting, and those which happen intermittently. Call our technical support department if you are experiencing difficulties.

8.5.1 Hard Failures

Hard failures which occur after the machine has been successfully installed and operating are most likely caused by a single failed component, or by a bad cable. Hard failures during the initial installation are most likely due to power supply failures, improper installation, static damage, or cables which were not installed properly.

Check list:

1. Turn on the machine, and watch the LEDs

   If they blink or change, go to step 5

2. If all LEDs are off, check that all of the power supplies are providing the correct voltage to the system. Oftentimes a power supply will fail when it is shut down after being in use for a long time. Also check that all the circuit breakers for the system are on.

3. If all LEDs are on, check that the Halt Switch is not on. If it is on, move it to the continue position, and try turning the power off and and then back on.

4. If all LEDs are on, check whether each expansion box is powered up. If any expansion box is shut down, DC LO will be asserted, preventing the QED 95 from powering up.

5. Remove the Unibus cable going from the processor box to the expansion boxes. The system should be able to power up part way, even if the expansion boxes are disconnected.

6. Although a far-end bus terminator is essential for good bus characteristics (the MS95 provides bus termination on its end) try removing the terminator to see if this affects the problem. If it does, this normally indicates an open NPG jumper, or the terminator's power supply may be bad.

7. Check all the cables and check the jumpers on each board.

8. If you have a 95-MBA board (in 11/70 systems only), remove it from the backplane. The MBA is only needed for Massbus transfers, and to operate the 11/70 front panel, so you should be able to get to the menu without it.

9. If the LEDs change, look up the value in the table in Appendix C, which will indicate the most likely defective board. If the value does not appear in the table, it normally indicates a CPU error.

10. If the machine halts before displaying a prompt or displaying the menu, check the LEDs. If the system displays the configuration menu, but cannot boot, the problem is most likely in the MS95, or else the grants may not be passing down the bus correctly.

11. If a DMA device blinks repeatedly, but is not able to transfer the boot block, it is quite likely that the DMA device is installed in a slot which is wired to pass NPG grants. Check that the jumper from CA1 to CB1 on that slot of the backplane is removed.

12. If the red or yellow LEDs on the MS95 come on, there might be a memory error. Note that it is normal for these LEDs to blink on during the first few seconds of the self-diagnostics. However, once the memory has begun the clearing process, the LEDs should stay off.

13. If the machine displays the menu, but the normal operating system will not run, try to boot the XXDP diagnostics, and run the Quickware diagnostics.

8.5.2 Intermittent Failures

In order to debug intermittent failures, it is important to develop a log of errors. Each time the system has an intermittent failure, you should make an entry in this log describing the problem as completely as possible. Hopefully, a pattern will develop, indicating that a particular piece of hardware or a particular program exacerbates the problem. If a hardware problem is indicated, run the appropriate diagnostic on that device. Also, build a DECX exerciser which can exercise the suspect device in conjunction with memory accesses and processor operations. Try turning the cache on and off (use the menu option described in 6.3.8.3, or the CON, COFF options in DECX) while running a test, to see if the cache affects the problem. Also, try running the Quickware diagnostics.

If the red or yellow LEDs on the MS95 come on during a failure, it is very likely that there is a DRAM memory chip failure. If the cache memory affects the problem, the cache (which is located on the CPU board) may be implicated. Errors while using the Massbus are most likely to be caused by the MBA board, but may also be caused by missing jumpers or defective circuits on the CPU or AUX boards.

9 Processor Differences

The QED 95 is very similar to the 11/84 and 11/70 processors; however, some differences are inevitable. This section reviews these differences. It may be helpful to software specialists who must determine the exact behavior of the machine.

The QED 95 implements every instruction implemented by the J-11. CIS is not implemented. Software that expects either CIS, or the absence of certain other instructions may fail. However, the majority of software will reconfigure itself at run or compilation time if CIS is not available. The Halt instruction may be configured to Halt, Trap-4, or Trap-10, in Kernel and also in Supervisor or User modes. A configuration register controls this option. A Jump to a register may be configured in the same way as Halt in User mode.

FIS is not implemented; these and all other illegal instructions will Trap through 10. FP-11 instructions are implemented and run asynchronously to the processor (other instructions may execute while the floating point finishes).

The value returned from a move from processor type (MFPT) instruction is controlled by a configuration register. If the register contains 0, a Trap-10 occurs; otherwise, the value will be moved into R0. Thus, MFPT can return any value, except 0.

Reset lasts a minimum of 15 microseconds. It may last longer if no power failure condition is present, and if the MS95 stretches INIT out for the system bus.

You can access the PSW at 777776 or through MTPS, MFPS. The MMU is a complete 22-bit MMU with Kernel, Supervisor and User modes and Instruction and Data spaces.

MMR 0, 1, 2, and 3 are implemented. MMR0 bit 8 (maintenance mode) and bit 12 (MMU trap) are implemented. MMR2 does not track interrupt vectors, as in the 11/45 and 11/70. The PARs are complete, and the PDRs include bit 15, (bypass cache) and a configuration register selects whether PDR bits 0 and 7 are active or read as zero. In this way, 11/70-style PDRs can be emulated.

Programs can run out of the PARs.

CLR, SXT, and MOV perform write-only cycles for the destination operands.

EIS instructions perform read-only cycles for the source operands.

No instructions can be interrupted mid-execution.

Two general register sets are implemented. These are not available at 777700-777717. The console can reference the registers through ODT. Locations 777700-777717 are used for configuration registers to alter the processor behavior slightly. These locations may be set invisible by setting bit 0 of the byte at 777715.

Odd address and time-out errors are detected.

The Stack Limit register is implemented and programmable.

Appendix A System Registers

A.1 Notes on System Registers

All unused bits should be assumed to be zero.

The last line indicates which machines implement a register, if it is not universal.

Some registers are cleared on power-up, start (Go), or by the Reset instruction. The Bus Reset line does not usually effect CPU registers such as these.

Bit values: (octal)

0 = 1;	1 = 2;	2 = 4;
3 = 10;	4 = 20;	5 = 40;
6 = 100;	7 = 200;	8 = 400;
9 = 1000;	10 = 2000;	11 = 4000;
12 = 10000;	13 = 20000;	14 = 40000;
15 = 100000		

A.2 System Register Descriptions

165000-165776 Boot ROM - EEProm, or CpuRom

1702xx-1703xx Unibus Map Registers

Only the first 31 pairs are used, although all are implemented. Low order bits are stored in the first word of the pair, high order bits in the second word. The 00 location contains low bits; 02 contains high bits. The low-order word contains bits 15-1; bit 0 is always 0. The high order word contains bits 21-16 in bits 5-0. Other bits are always 0.
Only exists on 22 bit Unibus machines. (95, J-11, 44, 70)

172100 Memory Parity Control/Status Register

Resides on memory board. Additional memory boards have CSRs at 172102, 172104, etc. Bit definition varies based on type of memory.
15: Parity error; double bit error on ECC memory
4: Single bit error on ECC memory
1: Disable correction on ECC memory
0: Enable Parity error

1722xx Supervisor PARs, PDRs

Inst PDRs 00-16
Data PDRs 20-36
Inst PARs 40-56
Data PARs 60-76
Undefined at power-up. Unaffected by Reset or Start.
PARs are 12 bits on 18-bit machines, 16 bits on 22-bit machines.
PAR bit 0 corresponds to Address bit 6.
PDR bits--
15: bypass cache
14-8: Page Length field
7: Attention bit (11/70 only)
6: Page-written
3: Expansion direction 0 = up, 1= down

2-1 Access control field

00 = Non-Res;	01 = Read Only;
10 = Not used, Abt;	11 = Read/write.)

11/70 defines 2-0 as ACF, with

000 = N/R;	001 = R/O, trap on reads;
010 = R/O;	011 = unused, abort;
100 = R/W, but trap;	101 = R/W, trap writes;
110 = R/W;	111 = abort all

Typical PAR values are:
0,200,400,600,1000,1200,1400,177600 (identity map)
Typical PDR values are
77406; 77506 (page ok; but written)
Attention and written bits are cleared when PAR or PDR is written.

1723xx — Kernel PARs, PDRs

Inst PDRs 00-16
Data PDRs 20-36
Inst PARs 40-56
Data PARs 60-76
See Supervisor PARs/PDRs for bit definitions.

172516 — MMR3 - Memory Management Control Register 3

5: Enable Unibus Map (on Unibus machines)
4: 22 bit enable
3: enable CSM instruction
2: enable kernel data space
1: enable super data space
0: enable user data space
Cleared by power-up, start, or reset
95, J-11, 11/44: bits 5-0 R/W
11/70: bit 3 n/a

173000-173776 — Boot ROM, EPROM, or 9312 ROMs(4)

17650x — Second (Auxiliary) DL-11/SLU

Standard DL-11-W registers on 85,90,95,24,44

177520 — BtDiag - Boot and Diagnostic Status Register

11/53: Boot and configuration info (Native Register)

11/84: Controls clock select, ROM select, etc.
15: Reboot enable
14: write for byte select for EERom writing
14: read as Power Ok (normally 1)
13: Force Line clock interrupt enable
12: Disable LTC
11-10: LTC clock select:
00=LTC line; 01=50Hz; 10=60Hz; 11=800Hz.
9: Enable Halt-on-Break
8: Stand-alone mode
7: Disable 173000 ROM
6: Disable 165000 ROM (make invisible)
5: Disable 165000 ROM outputs
4: ROM Write enable (165000)
3: No QMI Memory board present
2-0: Implemented, but no meaning

177522 ROM PCR - Page Control Register for Boot ROMs

15:9 control page of ROM at 173000
6:1 control page of ROM at 165000
90: bits 8,7,0 are always 0. Word only.
95: bits 8,7,0 may be set, but will not affect ROMs. Word only
11/84: bits 15,8,7,0 are not implemented.

177524 CfgDsp-Configuration/Display Register

Has lights for 84, 90, 95
On 84, six bits display, and reads as 8 bit configuration switch
On 90, lower 8 bits display, but all can be r/w. Word only.
On 95, all 16 bits display and are r/w. Word only.

177526 SluCfg, SluCtl - SLU Configuration Register

90, 95: Word only
12: EUB enable
11: 2 MB installed
10: Enable Memory
9: Enable Aux
8: Enable Cons
7: Aux Programmable Baud Rate Disable
6: Cons Prog Baud Rate Disable (default =1)

5-3: Aux Baud Rate
2-0: Cons Baud Rate

177546 LTCCsr - Line Time Clock(KW-11) Control and Status Register

7: LTC monitor (read only)
6: LTC Interrupt Enable
Other bits 0

17756x Console DL-11 CSRs

177570 SWR - Switch/Display Register

Has toggle switches/lights on older machines.
90: not implemented
95: may be configured by reg 710 as R/O, W/O, R/W or NXM.
11/44: modified by console program. R/O from program
11/60: cleared on power-up. R/W
11/70: Lights/Switches

177572 MMR0 - Memory Management Control Register 0

15: Non-Resident Abt
14: Page Length Abt
13: Access Mode Abt
(15-13, when set, freeze MMR0[6-1], MMR1, MMR2)
12: MMU Trap
9: Enable MMU Trap
8: Maintenance mode (relocate only dest refs)
7: Instruction completed
6-5: Mode of abort ref.
4: D/I of abort ref. (D = 1)
3-1: Page # of abort ref
0: Enable MMU
95:
J-11: 12-7 unused. Cleared on power-up, start.
11/44: 12-9,7 unused
11/60: 12-9,7,4 unused
11/70: 11-10 unused

177574 MMR1 - Memory Management Control Register 1

15-11: Amount reg changed, second ref.

10-8: Second reg. changed
7-3: Amount reg changed, first ref.
2-0: First reg. changed
Read-Only
95, J-11, 11/44, 11/70

177576 MMR2 - Memory Management Control Register 2

Loaded with 16-bit virtual address (PC) at the start of each instruction.
Read-Only
95, J-11, 11/44, 11/60, 11/70

1776xx User PARs, PDRs

Inst PDRs 00-16
Data PDRs 20-36
Inst PARs 40-56
Data PARs 60-76
See Supervisor PARs/PDRs for bit defs.

17770x 95 Scratch Regs.

These four words may be enabled on the 95.

177710 95-CPU Configuration Reg.

On 90 rev C, always writeable, readable when bit 8 is on.
Word writes only.
8: enable reads of this register
5: Disable NXM on writes
4: Force cache miss on data
3: Map data reads through Unibus map.
15-14: CPU rev 00= 90 CPU Rev C
95: Byte writeable
15: r/o VectSeen. Set and used during interrupt vector processing
on 95. Not defined at other times.
14-11: 1000 = 95-CPU Rev A-C
7: R/W but no effect
6: Set non-cachable I/O page to 8k/256k
2: Implement Red Stack traps like 11/70
1: Check odd addresses
0: Enable 11/70 style MMU (expanded MMR0, PDR defns)

177712 — 95 MBA Configuration Reg.

0-2: MBA arbitration priority selection
3: MBA parity polarity
Only low byte may be written

177714 — 95 Aux Config Reg. 1

0: Enable Stack Limit Register (774)
1: Enable uPrgBrk Register (770)
2: Enable SysID, SysSz Regs. (760,762, 764)
3: Allow writes to SysID, SysSz, Cache Data
4: Disable Cache Data Reg (754)
5: Disable Maintenance Reg. reads (750)
6: Disable Maintenance Reg. writes (750)
7: Enable Error Address Regs. (740, 742)
8: Enable Diag Regs. (730, 732)
9: Disable Aux Cfg Reg. 2 reads(716)
Disable Scratch Regs. (70x, 72x, 736, 756)
10: Disable SWR reads (570)
11: Disable SWR writes (570)
12: Disable Boot Regs. (520, 522, 524)
13-15: Switch Register select
Note that 73x and 520 respond from memory board

177716 — 95 Aux Config Reg. 2

0: Enables reads of 712, 714
7: 4 MB of memory (written to tell MBA logic about NXM limit)
15: ODT

17772x — 95 Scratch Regs

These 4 regs may be enabled on the 95.
177720 contains the processor type.
177722 contains an option mask.

177730 — Diagnostic Control Register

11/84 uses for diagnostics.
90, 95: always reads as 0.

177732 Diagnostic Data Register

11/84 uses for diagnostics.
90, 95: always reads as 0.

177734 KMCR - Memory Configuration Register

11/84 Implements bits 15-6 for diagnostic purposes.
bit 5 forces 18-bit mode.
bits 4-0 indicate how many extra 8 KB pages should be allocated to the unibus. 0 = default, one 8 KB page
Cleared by DCLO.

177736 95 Scratch Reg.

90: Always reads as 0.
This word may be enabled on the 95.

177740 ErrAdrL - Error Address Low Register

11/70: Contains 16 low bits of address causing parity error
11/60: WCS status register
95: Latches low bits of physical address when MSER bit 15 is set.

177742 ErrAdrH - Error Address High Register

11/70: Bits 5-0 contain high six bits of address causing parity Error bits 15-14 contain C1,C0 of bus tran type. (00 if read.)
95: Latches high bits of physical address when MSER bit 15 is set.

177744 MSER - Memory System Error Register

15: Cache memory parity error (set by Aborts to 114)
7: High byte parity error
6: Low byte parity error
5: Tag parity error
1: 90 (Rev A & B) force cache miss on data read (0 on 90c)
0: 90 (Rev A & B) map data reads through Unibus Map (0 on 90c).
J-11 - Cleared on Power-up, Start.
90: Word only
95: Word only
11/44 - bits 15, 7-5;
7-5 all set together unless CCR[7] Par Err Abt is set

Cleared by writes, or Start.
11/60 - bits 7-5 only
11/70 - all bits (15-0) implemented. Each bit is cleared by writing a one to it.
11/84 also has bits 14,13.

177746 CCR - Cache Control Register

10: Write wrong parity tag
9: Unconditional cache bypass
8: Flush cache (w/o) (always reads as 0)
7: Parity error abort
6: Write wrong parity data
3: Force miss 1
2: Force miss 0
1: Diagnostic mode: forces NXM word writes to allocate cache.
Present on 11/84 and 90 rev A and B; but moved to 710.5 for 90c,95
0: disable cache parity traps
84,85,90,95 - bits 5-4 are 0
90c,95 - Bit 1 is always 0
11/44- Bits13-12, 10-6, 3,2,0
11/60- Bits 7,6,3,2,0 only.
11/70- Bits 5-0 are implemented.
Bits 7,0 vs cache parity errors: 00-miss, update, trap to 114;
01-miss, update, no trap; 1x-miss, abort to 114.

177750 Maint - Maintenance Register

0: Power OK
7-4: Module type
(0010 = 11/84; 0100 = 11/53; 0011 = KXJ11;)
9: Unibus system (valid on J-11 machines)
8: FPJ11 installed. (valid on J-11 machines)
95: All bits r/w, byte accessible
90: Word only
J-11
11/44 has this for cache maintenance.
11/70 has this for cache and memory maintenance.

177752 HMR - Cache Hit/Miss Register

5-0: Cache hit bits. 0 is most recent ref. 1 = hit,.0 = miss.

Read-only
On 11/44, has cache match maintenance features in upper bits.
95, J-11, 11/44, 11/60, 11/70

177754 CDR - Cache Data Register

11/44: contains data loaded into cache from memory.
95: May be configured as r/w
90: Reads contents of 177524 (display reg.).

177756 95 scratch reg

This word may be enabled on the 95.

177760 SysSzL - System Size Register (Low Bits)

11/70 - contains size of memory. Read-only
Low byte is all ones.
High byte indicates number of 8KW blocks -1
(machine with 2048 kb of memory would have 077777
machine with 3840 kb = 167777
4088 kb = 177577)
95 - may be configured as r/w

177762 SysSzH - System Size Register (High bits)

11/70: Read-only. Always 0
95: may be configured as r/w

177764 SysID - System ID Register

11/70 only. Read-only
95: may be configured as r/w

177766 ErrReg - System Error Register

The bits in here record why a Trap-4 occured.
7: 11/60- Microbreak
7: Illegal halt
6: Odd address
5: Non-existent memory
4: Unibus timeout
3: Yellow stack violation
2: Red Zone stack (any stack vio on 44)
1: 11/60- WCS error
0: 11/60- Illegal internal address

0: 11/44- Power Failing.
95,11/70, J-11: only bits 7-2 are implemented. Cleared by writes, power-up, start.
J-11: Cleared by J-11 on power-up, start.
11/44: 7-4,2,0 implemented, cleared by writes.
11/60: Only bits 7,6,4,2,1,0 implemented. Word only

177770 uPrgBrk - microProgram Break Register

11/60: 12 bits, write only.
11/70 Lower 8 bits only.
95 - May be configured as r/w

177772 PIRQ Programmed Interrupt Request Reg.

15-9: Software programmed interrupt request, levels 7-1
7-5, 3-1: Encoded value of highest PIRQ pending. (Read only)
Cleared by power-up, Start(Go), Reset.
95, J-11, 11/44, 11/70

177774 StkLim - Stack Limit Register

15-8 contain lower limit for stack. Cleared on power-up, start (Go), and by Reset.
A value of 0 places the stack limit at 400.
95, 11/70, 11/60

177776 PSW - Program Status Word

15-14: Current mode
(00 = kernel, 01 = super, 10 = illegal, 11 = user)
13-12: Previous mode
11: Register set
8: Suspended instruction (CIS)
7-5: Priority
4: T-Bit (Trace Traps)
3-0: Condition Codes NZVC
95, J-11: 10-8: N/A
11/44: 11-9: N/A
11/60: 11-8: N/A, no Super mode
11/70: 10-8: N/A

Appendix B Cable Pinouts

B.1 QMI

1	Reserved	2	QMAdrLtch L
3	GND	4	QMNextOK L
5	GND	6	QMDataSync L
7	GND	8	QMSlvAck L
9	GND	10	QMast Bus L
11	GND	12	QMInit L
13	GND	14	QMVect L
15	QMMapTrn L	16	Reserved
17	GND	18	QMRdWrt L
19	QMBytPar L	20	GND
21	QMD Bus 0	22	QMD Bus 1
23	QMD Bus 2	24	QMD Bus 3
25	QMD Bus 4	26	QMD Bus 5
27	QMD Bus 6	28	QMD Bus 7
29	QMD Bus 8	30	QMD Bus 9
31	QMD Bus 10	32	QMD Bus 11
33	QMD Bus 12	34	QMD Bus 13
35	QMD Bus 14	36	QMD Bus 15
37	GND	38	QMA Bus 0
39	QMA Bus 1	40	QMA Bus 2
41	QMA Bus 3	42	QMA Bus 4
43	QMA Bus 5	44	QMA Bus 6

45	QMA Bus 7	46	QMA Bus 8
47	QMA Bus 9	48	QMA Bus 10
49	QMA Bus 11	50	QMA Bus 12
51	QMA Bus 13	52	QMA Bus 14
53	QMA Bus 15	54	QMA Bus 16
55	QMA Bus 17	56	QMA Bus 18
57	QMA Bus 19	58	QMA Bus 20
59	QMA Bus 21	60	QMNoMemIn

B.2 Host Adapter

1	N/C	2	N/C
3	N/C	4	N/C
5	GND	6	HHalt L
7	HACLO L	8	HDCLO L
9	GND	10	BGN L
11	Tie Hi	12	PwrDwn L
13	InitRes L	14	N/C
15	HBR4 L	16	HBR5 L
17	HBR6 L	18	HBR7 L
19	SAM	20	HAdrVal
21	HGrtDone L	22	HAbort L
23	GND	24	DmaGrt L
25	RUN	26	HDMAReq L
27	GND	28	N/C
29	SlvCtl L	30	GND
31	IOMapEn	32	HBootEn
33	Tie Hi	34	GND

B.3 CIM Connector

1	N/C	2	N/C
3	ConSin Ext	4	ConSout Ext
5	ConSin	6	ConSout
7	+5	8	GND
9	Boot H	10	Halt H
11	HOpt 1	12	BrkEnb H
13	RUN L	14	BusMstr
15	QMInit L	16	HDCLO L
17	N/C	18	N/C
19	HACLO L	20	N/C
21	DTR	22	RUN
23	BootEn L	24	HOpt 2
25	Halt L	26	Boot L
27	GND	28	+5
29	AuxSout	30	AuxSin
31	AuxSout Ext	32	AuxSin Ext
33	N/C	34	N/C

Appendix C LED Definitions

C.1 MS95 LEDs

The MS95 memory board has three LEDs which are defined as follows:

Green	Power is present in the memory array.
Yellow	A single bit error has occurred.
Red	A double bit error has occurred.

C.2 QED 95 AUX LEDs

The QED 95 AUX has sixteen red LEDs on it. (four groups of four.) These LEDs are addressed by the Boot Display Register at 177524. The Boot Roms write to these LEDs as the internal diagnostics are run. If the internal diagnostics detect an error, the value displayed by the LEDs will indicate what type of error was found. Use the following table to find the error, given an LED value. The value displayed by the LEDs is an octal number, with a lit LED indicating a one, and an off LED indicating a zero.

If the QED 95 is installed in an 11/70 system, the front panel Data LEDs will display the inverted version of the value written to the AUX LEDs. By reading the octal value from the front panel Data LEDs (treating a lit LED as a zero, and an off LED as a one) you can easily see the value displayed by the diagnostic LEDs. Note that these front panel LEDs are also written to when the Display Register (at 177570) is written. Thus, once a diagnostic program or operating system writes to this address, the Boot Diagnostic code will be overwritten on the front panel.

If all LEDs are off, the QED 95 is not receiving proper power. Check the system power supplies, review the installation, and make sure that the proper jumpers are installed, and that the boards are in the correct slots.

If all LEDs are on, the QED 95 system has not powered up properly. DCLO may be on, check all cables, and review the installation.

If some LEDs are on, use the following chart to determine the source of the error.

Value	Meaning	Source of trouble
1	Rom Page Ctl Register not initialized	AUX, CPU
2	Boot and Diagnostic Register wrong	AUX, CPU
3	Memory System Error Register wrong	CPU
4	Cache Control Register wrong	CPU
5	Maintenance Register wrong	AUX, CPU
6	CPU failed register tests	CPU
7	Checksum on Boot Rom page 0 failed	AUX, Boot ROM
10	Rom Page Ctl Register failed	AUX
11	EEROM missing subroutines	AUX, EEROM
12	Checksum on Boot Rom failed	Boot ROM, AUX
13	couldn't jump off of boot page 0	AUX, Boot Rom, Page Ctl Register
14	Burning EEROM	AUX, EEROM
15	Stack Pointer is bad	CPU
16	MMU PARs are bad	CPU
17	MMU PARs are bad	CPU
20	MMU PARs are bad	CPU
21	Subroutine Call failed	CPU
22	CPU type not determined	CPU
23	CPU not a QED 90 or 95	CPU
24	QED 90 CPU test failed	CPU
25	QED 90 CPU test failed (rev A or B)	CPU
26	Maintenance Register failed	AUX, CPU
27	Memory System Error Register failed	CPU
30	Cache Control Register failed	CPU
31	Boot Diagnostic Register failed	AUX, CPU
32	ROM Page Control Register failed	AUX
33	Jump to boot page 14 failed	AUX, Boot ROMs, Page Ctrl Reg.
34	QED 90 CPU test failed (Rev C or later)	Maintenance Register CPU
35	Memory System Error Register failed	CPU
36	Cache Control Register failed	CPU
37	Boot Diagnostic Register failed	AUX, CPU

40	ROM Page Control Register failed	AUX
41	Jump to boot page 14 failed	AUX, Boot ROMs, Page Ctrl Reg.
42	QED 95 CPU test failed, Maintenance or Config Register	AUX, CPU
43	Memory System Error Register failed	CPU
44	Cache Control Register failed	CPU
45	Boot Diagnostic Register failed	AUX, CPU
46	ROM Page Control Register failed	AUX
47	Jump to next boot page failed	AUX, Boot ROMs, Page Ctrl Reg.
50	AUX RAM-based register failed	AUX
51	Stack Limit Register failed	AUX, CPU
52	Jump to boot page 14 failed	AUX, Boot ROMs, Page Ctrl Reg.
53	(Boot Page 14) EERom failed -- version	AUX, EERom, Page Ctrl Reg.
54	EERom failed -- checksum	AUX, EERom, Page Ctrl Reg.
55	Jump to boot page 16 failed	AUX, Boot ROMs, Page Ctrl Reg.
56	Burning EERom	AUX, EERom
57	Burning EERom	AUX, EERom
60	Burning EERom	AUX, EERom
61	Burning EERom -- code	AUX, EERom
62	Burning EERom -- code	AUX, EERom
63	Burning EERom -- code	AUX, EERom
64	Burning EERom -- version	AUX, EERom
65	Burning EERom -- user pages	AUX, EERom
66	Burning EERom -- checksums	AUX, EERom
67	(Boot Page 16) Config/Display LEDs failed	AUX, CPU
70	Hit/Miss Register failed	CPU
71	Subroutine Call through EERom failed	AUX, Boot ROMs, Page Ctl Reg
72	No Memory installed	Memory, Cables, CPU
73	No Memory installed	Memory, Cables, CPU
74	SLU/Cfg Register failed (first memory board access)	Memory
75	Write to Memory CSR failed	Memory
76	Cache Control Register failed	CPU, Memory
77	Quick memory scan failed	Memory
100	Read of Memory CSR failed	Memory
101	Jump to Cold Boot failed	AUX, CPU, Boot ROMs, Page Ctrl Reg
102	Warm Boot failed	Memory
103	Parameter recall/setting failed	Memory, AUX, CPU, Boot ROMs
104	Restarting via (24)	Memory, Power
105	Booting through menu/prompt	Memory, AUX

114		AUX, Boot ROMs, Page Ctrl Reg
106	Jump to boot failed	Memory, CPU
107	(Cold Boot) Stand-alone mode failed	Cache, CPU
110	Cache failed to record a Hit	Cache, CPU
111	Cache failed to hold correct data -- 0	Cache, CPU
112	Cache failed to hold correct data -- 177777	Cache, CPU
113	Cache data store failed	Cache, CPU
114	Cache data retrieval failed	Cache, CPU
115	Cache patterns failed	Cache, CPU
116	Cache address lines failed	Cache, CPU
117	Cache size failed	Cache, AUX, CPU
120	Write to cache or jump boot page failed	CPU
121	MMU enable failed	CPU
122	Writes to cache through MMU failed	Cache, MMU, CPU
123	Cache Tags failed	Cache, MMU, CPU
124	Cache failed	Cache, CPU
125	Subroutine in Cache failed	Cache, CPU
126	Subroutine in Cache failed	Cache, CPU, AUX
127	CCR failed or FP or jump boot page failed	AUX, CPU
130	Floating Point Status failed -- 0	AUX, CPU
131	Floating Point Status failed -- 147757	AUX, CPU
132	Floating Point Registers failed -- 0	AUX, CPU
133	Floating Point Registers failed -- 177777	AUX, CPU
134	Floating Point math error	AUX, CPU
135	Floating Point error	AUX, CPU
136	Jump to next boot page failed	Memory, CPU
137	Memory failed	Memory
140	Low Memory could not be cleared	Memory
141	Memory clear routine could not be transferred	Memory
142	Clearing memory	Memory
143	Setting Parameters	Memory
144	Setting Baud Rate of terminal	Memory
145	Booting	EEROM, AUX, CPU
146	Boot ROM boot selection invalid	Memory
147	Failure while checking SLU	Memory
150	Failure while enabling on-board SLU	
151	Waiting for Carriage Return to Auto-Baud	Memory
152	Failure while checking LTC	Memo
153	Line Time Clock failed	

154 Waiting for Carriage Return to Auto-Baud
155 Checking speed of received character
156 Waiting for Carriage Return to Auto-Baud