KDJ-11-B (Firmware)

REVISIONS

REV	DESCRIPTION	CHG NO	ORIG	DATE	APPD BY	DATE

TABLE OF CONTENTS   ROM CODE V6.0   Parts 23-077E5 & 23-078E5

TABLE OF CONTENTS ROM CODE V6.0 Parts 23-077E5 & 23-078E5

1.0 OVERVIEW OF DIAGNOSTIC PRODUCT

1.1 PRODUCT DESCRIPTION

This functional specification describes the KDJ11-B Boot and diagnostic Rom code. There are basically three parts to the rom code. One part of the code is a diagnostic test to be run at power up. The diagnostic checks out the CPU module, the memory module(s) and the Unibus adapter module for Unibus systems. The second part is the boot code. This code allows most supported devices to be booted up. The boot code can also start programs stored in the EEPROM or programs stored in M9312 type boot roms located on the UBA module.The third part of the code allows storing and modification of parameters for the CPU, the UBA and the system. This portion of the boot code also provides support for the EEPROM itself.

****
This specification only applies to EPROM's marked 23-077E5 and 23-078E5. The version of this code is V6.0. The version number is printed out when Setup mode is entered.
****

1.2 PRODUCT USERS AND USES

Engineering, Manufacturing, Field Service and Customers will use this program to test, configure, and boot their system. The diagnostics are automatically run at system power up and selected tests are executed after a power failure. The user has the capability of by-passing certain portions of.the diagnostics via the EEPROM. Testing can be terminated early by the user typing Control C (^C) on the console terminal.

This product is needed to give the user confidence that the system is functioning properly, or, if a failure is detected, to report to the user which field replaceable unit (FRU) is malfunctioning. The diagnostic will isolate failures to the CPU module, the memory module(s), the UBA module, or to the Boot device and its interface. The program is also needed by the user to Boot his/her system media.

The program uses Led's and console printouts as means of conveying diagnostic progress and error reports to the user. Multiple methods are used in case one of the methods has a failure. When error reports occur, the program will wait for an operator's response before continuing testing. Some errors such as Cache and ECC errors can be overridden by the user, however, doing so degrades the overall system performance.

2.0 PRODUCT GOALS

2.1 PERFORMANCE GOALS

The diagnostic will perform all of its tests in less than 30 seconds with 4 megabytes of MSV11-J memory and a 15 M Hz J-11 chip.

16K word's of ROM is used to provide: a "User Friendly" interface; a high "STUCK AT" fault coverage; fault isolation to a Field Replaceable Unit; progress reports; error reporting; passing of error information to loadable software; EEPROM Boot ROM support; Kernel system EEPROM configuration support, and; intelligent Boot Programs.

2.2 COMPATIBILITY GOALS

The KDJ11-B Boot rom code is being written specifically for the KDJ11-B CPU module only. Because of the use of stand-alone mode and other features unique to the KDJ11-B CPU, the program cannot be used with any existing CPU that I know of.

2.3 FAILSOFT GOALS

Errors will be clasified as being either fatal or non fatal errors. At the beginning of the diagnostic all errors are considered fatal. If an error occurs in the first two tests the rom code will attempt to type out the error number and then blink the display on and off with the error number. In order to proceed the operator would have to restart the system. During the blinking of the display if the Cpu is halted, the address of the failing test will be in R0. If any input is received at the console the rom program will retype the error number. This allows the error information to be redisplayed if the console had not been powered up at the time of the error.

After the first initial tests are complete if any errors occur the rom program will try to provide the user with more infomation and more options. The operator can loop on a test, rerun a test or bypass the test if the error is non fatal. Even fatal errors can be bypassed, but the method to do this is not obvious to the user who has not read the user's manual.

As a failsafe mechanism, progress reports and error messages are indicated to the User via the console device and LEDs on the KDJ11-B module. These methods use unassociated logic, therefore, the chances that both areas of logic are malfunctioning are very small. After being validated, the J11's Memory Management Registers will be used to pass information between layers of the diagnostic firmware and to the loadable software.

Before the execution of each diagnostic subtest, the LEDs on the KDJ11-B module will be updated with a test number. At various meaningful points in the diagnostic program, messages will be printed out on the console terminal. In the event of an error, the LEDs and the console error message will contain the test number of the diagnostic test being performed. If the error is a controlled error and the console device is functioning properly, the program will attempt to indicate the Field Replaceable Unit by the error message. At a minimum on an uncontrolled error, the LEDs will contain the last diagnostic test being executed at the time of failure. Therefore an intelligent decision can be made to what could be the Field Replaceable Unit.

If an error occurs in Supervisor or User mode, the program will switch to Kernel mode before reporting the error.

Virtual addresses 0-376 will be mapped to ROM area on initial startup. This will provide a mechanism for unexpected traps or errors in the Cache logic. Also the hardware Stack is initialized to the Memory Management Unit's PAR registers.

2.4 RESTRICTIONS

If the code is restarted at 173000 without using power up or Restart, the user should verify that the BCSR (17777520) bit 07 (HB DIS) is reset. If this is not done, then uncontrolled results may occur. This is not a recommended way to restart the code.

2.4.1 VT52 RESTRICTION AS A CONSOLE DEVICE

In some cases the rom code sends an ANSI clear screen command to the console if the terminal type in the EEPROM is set for ANSI video terminal. VT52's are not ANSI video terminals. Almost all ANSI commands begin with the ascii codes ESC and [. If these codes are sent to a VT52 it will go into hold screen mode. The KDJ11-B rom code can not be used when a VT52 is present unless it is in force dialog mode via the baud rate switch or the terminal type in the EEPROM is set for hard copy. This restriction does not apply to VT1xx or VT2xx series terminal that are set to VT52 mode.

2.5 NON-GOALS

This product does not attempt to isolate to the chip level.

This product does not do a 100% test of the Kernel system, however, its goal is to provide a 94% "STUCK AT" fault coverage of the CPU, Memory and the UBA modules and that the next layer of software can be loaded.

This product can not isolate system bus related problems, such as a "Stuck" bit.

At the present time, an ANSI MENU driven Diagnostic/Boot ROM is not a goal. There is a video mode bit which illiminates the use of backslash during deletes. The character to be deleted is erased from the screen by sending a <BACKSPACE> <SPACE> <BACKSPACE>. Other than this exception video mode is identical to hardcopy mode. The advantages of this is that code space is not wasted on duplicate routines and that if a hardcopy terminal is powered up in video mode you will not end up with a mess.

3.0 REQUIREMENTS

In order for the Diagnostic/Boot code to be executed, the J11 must be functioning to the point of executing instructions and the PCR registers must be initialized to select the Diagnostic ROM.

4.0 FUNCTIONAL DESCRIPTIONS

4.1 GENERAL DESCRIPTION OF THE KDJ11-B BOOT AND DIAGNOSTIC CODE.

The KDJ11-B CPU module (M8190) contains two EPROM's and one EEPROM. The two EPROM's together make up the KDJ11-B's boot and diagnostic rom code. At power up or restart the KDJ11-B passes control to the rom code. The rom code tests out the KDJ11-B and all available memory and then boots a previously selected device. In unibus systems the UBA is tested also. The EEPROM is used by the rom code to store information needed by the rom code to set up the KDJ11-B for normal operation. Some typical information stored in the EEPROM would be the selected boot device, test selections and hardware setup features. The EEPROM can also be used to store a foreign language file and additional boot programs that are not in the base rom. The following lists the general uses of the EPROM's and the EEPROM.

EPROM (16,384 by 16 bits in 2 EPROM's)

Internal power up diagnostics for CPU and memory.
Standard bootstrap programs.
General support routines.
EEPROM setup program.
UBA diagnostics and UBA/M9312 boot rom support for unibus sytems.

EEPROM (2,048 or 8,192 by 8 bits in 1 EEPROM)

Hardware parameters.
Boot device selection.
Foreign language text if needed.
Optional customer bootstrap programs.

For the rest of this description the two EPROM's will be referred to as the rom code.

The primary purpose of the EEPROM is to store all variable parameters and selections and to allow them to be changed easily without removing the KDJ11-B module from the box. Changes to the EEPROM are made under control of the rom code in a program called setup mode.

After all power up diagnostics are completed the rom code determines what action is to be taken. The rom code will either boot a previously selected device or enter a dialog mode in which the user inputs the device to be booted through the console device.

The following is a list of the default selections in the EEPROM. These selections are present when the EEPROM is initialized.

Power up and restart modes are both set to automatic boot mode (turnkey).
The automatic boot program determines which MSCP devices (units 0-7) are available and attempts to boot removable media devices first then fixed devices. If no devices are ready the rom code will try DL0 then MS0.
The clock logic is enabled and the clock source is from the power supply.
All testing is enabled.
The Halt on Break from the console SLU is disabled for Qbus systems. For unibus systems, enabling of halt on break is determined by the front panel lock switch.
There are other selections which are not listed here.

To change any of the selections the user would enter dialog mode and then enter setup mode which allows all of the selections to be changed.

4.2 FORCING ENTRY INTO DIALOG MODE

If dialog mode is not selected the user may force dialog mode by either typing CTRL P or CTRL C during testing or by selecting dialog mode through the console baud rate selection switch. The console baud rate selection switch has 16 positions. Positions 0 to 7 select the baud rate and the rom code mode is determined by the contents of the EEPROM. Positions 8 to 15 also select the baud rate but override the selection in the EEPROM and force dialog mode to be entered after completion of all diagnostics. The following table shows how dialog mode is selected with the baud rate selection.

Baud rate	EEPROM selects mode	Dialog mode is unconditionally selected
38400	0	8
19200	1	9
9600	2	10
4800	3	11
2400	4	12
1200	5	13
600	6	14
300	7	15

Switch postions

NOTE: In order for the rotary switch to work properly, switches 5-8 on the CPU module must be off.

4.3 TYPICAL MESSAGES DURING ROM CODE TESTS

Various messages are printed on the console terminal by the rom code during a normal power up sequence. There are two formats that the messages may occur in, they are the user friendly format and the standard format. The user friendly format is designed for the end user customers of Qbus systems. The standard format provides additional information. Examples of both formats follow for a typical system. The User friendly format is not available on unibus systems.

The following is an example of a typical system booting up in automatic boot mode. For this example the user friendly format is selected and the operating system on the disk is RT11.

```
                         Testing in progress - Please wait

                                1 2 3 4 5 6 7 8 9

                                 Starting system


RT-11FB (S) V05.00

.R DATIME
Date?
```

The following example is the same as above but the user friendly format is not selected. The user friendly format is not used on unibus systems.

```
Testing in progress - Please wait
Memory Size is 256 K Bytes
9 Step memory test
  Step 1 2 3 4 5 6 7 8 9


Starting system


RT-11FB (S) V05.00
```

The following is an example of a typical system powering up, running the internal diagnostic and then entering dialog mode.

```
Testing in progress - Please wait
Memory Size is 256 K Bytes
9 Step memory test
  Step 1 2 3 4 5 6 7 8 9


Commands are: [Help, Boot, List, Setup, Map, Test]
Type a command then press the RETURN key:
```

4.4 DIALOG MODE

When dialog mode is entered the user has six commands to choose from. The six commands are listed in the command line. The user may obtain a brief description of each command by typing H followed by the RETURN key or by typing ? only. The following example shows the help information being requested.

```
Commands are: [Help, Boot, List, Setup, Map, Test]
Type a command then press the RETURN key: H

Command     Description

Help        Type this message
Boot        Load and start a program from a device
List        List boot programs

Setup       Enter Setup mode
Map         Map memory and I/O page
Test        Continuous self test - Type CTRL C to exit
```

All of the commands may be invoked by typing only the first letter of the command followed by pressing the RETURN key. For example, the Map command can be invoked by typing either M or Ma or Map followed by pressing RETURN. On input all lower case letters are converted to upper case and leading spaces and tabs are ignored. On input the standard CONTROL R and CONTROL U functions are present. The DELETE function is present. If an invalid input is received an invalid message will be typed out and more input will be requested. On input any line of data must contain 16 letters or less.

Example of an invalid entry.

```
Commands are: [Help, Boot, List, Setup, Map, Test]
Type a command then press the RETURN key: MP


Invalid entry


Commands are: [Help, Boot, List, Setup, Map, Test]
Type a command then press the RETURN key:
```

The following is a brief description of each of the dialog mode commands.

4.4.1 DIALOG MODE HELP COMMAND

Help

This command types out a brief description of all of the commands. This command can be executed by either typing H followed by RETURN or by typing ? only. Dialog mode is restarted at the end of this command.

TABLE OF CONTENTS   ROM CODE V6.0   Parts 23-077E5 & 23-078E5

4.4.2 DIALOG MODE BOOT COMMAND

Boot

This command allows a device to be booted. This is a command that uses arguments. The arguments are the device name and the unit number. If the device name is left off the program will prompt the user for it. If the unit number is left off then program assumes that unit zero was desired. The unit ranges from 0 to 255(10) depending on the device and the boot program. There are three optional switches that can be used with the boot command, they are:

/A Request that the user type in a non standard CSR address.
/O The unit number is octal instead of decimal for unit numbers above 8.
/U In unibus systems if the boot exists in the base rom and on the UBA override the base rom boot and use the boot from the UBA board or M9312 module.

The format when using a switch is to type the device name and unit number followed by / and the switches. When there is more than one switch, use only one slash.

When the rom code has a mnemonic it searches for the first boot program with the same mnemonic. The rom code looks for matches from the following areas in the following order. The /U switch overrides the order for Unibus machines and effectively skips the 1st and 2nd areas.

1st area to search = EEPROM
2nd area to search = CPU ROM code
3rd area to search = UBA module for Unibus only
4th area to search = M9312 module for Unibus only

Examples:

B DL	Boot DL0
B DL1	Boot DL1
B DU8	Boot DU unit 8
B DU10/0	Boot DU unit 8.
B DU/A Address = 17760400	Boot DU0 with non standard CSR address of 17760400
B DU3/U	Boot DU3 boot using UBA or M9312 rom boot instead of CPU ROM code.
B DU11/U0	Boot DU unit number 9 using UBA or M9312 rom boot instead of CPU ROM.
B DU11/U/0	Invalid format will cause invalid entry error message
B DU3:	Boot DU3

If the user types a colon after the unit number it will be ignored. i.e. B DL1:

The single letter B implements a method of supporting non DEC boot devices on the Qbus or Unibus.
For a Qbus machine the letter B causes the rom code to disable the CPU ROM and check location 17773000 for the existence of a rom on the bus. If a rom is present and location 17773000 of the rom is not 0 (HALT) the rom code will pass control to address 173000 with MMU off and R0 set to the unit number.
For a Unibus machine the letter B causes the rom code to transfer control to the address contained in location 17773024 of a rom on the Unibus if there is any and the address is 165000 or greater.
In either case above if all of the conditions are not met the rom code will type out an Invalid device message.

Example of typing L to List the supported devices

```
Commands are: [Help, Boot, List, Setup, Map, Test]
Type a command then press the RETURN key: L

Device  Unit
name    numbers Source   Device type

 DU     0-255   CPU ROM  MSCP (RA80/81/60, RD51/52, RX50, RC25, ....)
 DL     0-3     CPU ROM  RL01/RL02
 DX     0-1     CPU ROM  RX01
 DY     0-1     CPU ROM  RX02
 DD     0-1     CPU ROM  TU58
 DK     0-7     CPU ROM  RK05
 MS     0-3     CPU ROM  TK25, TS05
 XH     0-1     CPU ROM  DECNET DEQNA
 NU     0-15    CPU ROM  DECNET DUV11
 NE     0-15    CPU ROM  DECNET DLV11-E
 NF     0-15    CPU ROM  DECNET DLV11-F
```

4.4.4 DIALOG MODE SETUP COMMAND

Setup

This command causes the rom code to enter setup mode which is a separate program with it's own set of commands. Setup mode allows the user to list or change most of the parameters in the EEPROM. The user may at any time exit setup mode and return to dialog mode. Setup mode is describe in detail later.

4.4.5 . DIALOG MODE MAP COMMAND

Map

This command will try to identify all memory in the system and then map all locations in the I/O page. Memory is mapped from location 0 to 17756000 in 1,024 byte increments. Memory is not mapped every location due to the amount of time it takes. The routine will try identify the size of each memory, the CSR address for each memory if applicable, the CSR type (ECC or Parity) and the general bus type. It is important to note that if two memories share some common addresses or have CSR's with the same address the map command will not work properly. After memory is mapped the routine will then typed out all addresses in the I/O page that respond. The I/O page map goes from addresses 17760000 to 17777776. In addition, all addresses that respond that are on the KDJ11-B or on the KTJ11-B are further described with a short description. There is no description for addresses that respond and are on the external bus with the exception of memory CSR's if present. Dialog mode is restarted at completion of the map command.

Example of typing the map command.

```
Commands are: [Help, Boot, List, Setup, Map, Test]
Type a command then press the RETURN key: M

Memory Map
Starting    Ending         Size in    CSR           CSR      Bus
Address     address        K Bytes    address       type     type

00000000 - 00777776        256        17772100      Parity   PMI
01000000 - 02777776        512        17772102      Parity   Qbus

Press the RETURN key when ready to continue
```

I/O page Map

Starting Address	Ending address	
17765000	- 17765776	CPU ROM or EEPROM
17772100	- 17772102	Memory CSR's
17772200	- 17772276	Supervisor I & D PDR/PAR's
17772300	- 17772376	Kernel I & D PDR/PAR's
17772516		MMR3
17773000	- 17773776	CPU ROM
17774400	- 17774416	
17777170	- 17777172	
17777520	- 17777524	BCSR, PCR, BCR/BDR
17777546		Clock CSR
17777560	- 17777566	Console SLU
17777572	- 17777576	MMR0,1,2
17777600	- 17777676	User I & D PDR/PAR's
17777744	- 17777752	MSER, CCR, MREG, Hit/Miss
17777766		CPU Error
17777772		PIRQ
17777776		PSW

4.4.6 DIALOG MODE TEST COMMAND

Test

This command causes the rom code to run most of the power up tests in a continous loop. The rom code starts at test 70 and runs all applicable tests and then restarts the loop after test 30 is complete. If an error occurs the general error routine is entered. The user may exit the test loop by typing CTRL C at the console. At the time the test loop is exited the rom code will print out the total number of loops and the total number of errors if any. The user may also type a test number after the test command and if the test is applicable the rom code will loop on that specific test only until an error occurs or CTRL C is type. If the test number selected is not a loopable test the general test loop will be entered and all loopable tests will be run.

NOTE: CTRL C is not echoed by the rom code on the console terminal.

Example of Booting a device

```
Commands are: [Help, Boot, List, Setup, Map, Test]
Type a command then press the RETURN key: B DL2

Trying DL2

Starting system

RT-11FB (S) V05.01

.SET TT QUIET

.R DATIME
Date? [dd-mmm-yy]?
```

4.4.3 DIALOG MODE LIST COMMAND

List

This command will list out all available boot programs in the rom code and will also list any programs available in the EEPROM. The information listed is the device name, unit number range, source of the boot program and a short device description. The device name is normally a two letter mnemonic. In some cases the name may be a single letter. The device name must always be letters from A to Z. At input the rom code always converts all lower case letters to upper case. The unit number range is the allowable range of unit numbers that is valid for a particular boot program. The range varies from 0 only to 0 to 255 depending on the device. The source lists where the boot program resides. The programs reside in either the rom code or in the EEPROM. The description is intended to be the name of on the outside of the device to be booted. An example would be for a device name of DL the description would be RL02 which is the name on the outside of the physical device itself. Dialog mode is restarted at the completion of the list command.

On unibus systems the list command will list any boot programs found on the UBA board and on a M9312 module if it is present on the unibus. Each boot program will be identified with a label of either UBA ROM or M9312 in the source column of the list description.

The following is an example of the test command to run all tests and typing CTRL C after 4 passes of all tests.

```
Commands are: [Help, Boot, List, Setup, Map, Test]
Type a command then press the RETURN key: T

Continuous self test - Type CTRL C to exit

Total Passes = 4
Total Errors = 0
```

The following is an example of executing a particular test and typing CTRL C after 1876 passes.

```
Commands are: [Help, Boot, List, Setup, Map, Test]
Type a command then press the RETURN key: T 60


Looping on test 60 - Type CTRL C to exit


Total Passes = 1876
Total Errors = 0
```

4.5 SETUP MODE DESCRIPTION

Setup mode

Setup mode is entered by typing S followed by pressing RETURN in dialog mode. Setup mode allows the user to list or change most of the parameters in the EEPROM. Setup mode also allows changes to any bootstrap programs stored in the EEPROM. Setup mode has fifteen commands. After power up or restart and the completion of all tests the rom code loads the first 105 bytes of the EEPROM into memory beginning at location 2000. This area in memory is referred to as the setup table. The setup table contains all of the parameters except the EEPROM resident boot programs. The EEPROM may contain various types of information for the system. The first 105 bytes is information needed by the rom code to configure the KDJ11-B and to determine the boot device, test selections and modes. Other information in the EEPROM could be user bootstrap programs and a foreign language file. Setup mode allows changes to the first 105 bytes and to the user bootstrap programs. The foreign language area if present cannot be changed in setup mode.

When setup mode is first entered it types out a list of all commands and gives a short description of each command. The following is an example of setup mode being entered from dialog mode after the user types S followed by RETURN.

```
Commands are: [Help, Boot, List, Setup, Map, Test]
Type a command then press the RETURN key: S

KDJ11-B Setup mode      KDJ11-B ROM V6.0

Command    Description

   1       Exit

   2       List/change parameters in the Setup table
   3       List/change boot translations in the Setup table
   4       List/change the Automatic boot selections in the Setup table
   5       List/change the terminal Setup message in the Setup table
   6       List/change the switch boot selections in the Setup table
   7       List boot programs

   8       Initialize the Setup table
   9       Save the Setup table into the EEPROM
   10      Load EEPROM data into the Setup table

   11      Delete an EEPROM boot
   12      Load an EEPROM boot into memory
   13      Edit/create an EEPROM boot
   14      Save boot into the EEPROM
   15      Enter ROM ODT

Type a command then press the RETURN key:
```

The following is a detailed description of each command. To execute a command, type the command number followed by pressing the RETURN key. At any time the user may type CTRL C to return to dialog mode or CTRL Z to return to the beginning of setup mode.

> NOTE: Never terminate a change of any parameter with CTRL C or CTRL Z. If this is done the change is ignored and lost. Always use the terminating character RETURN after any change and then use CTRL C or CTRL Z.

4.5.1 SETUP COMMAND 1

This is the exit command. This command returns the user to dialog mode. Dialog mode is also entered if CTRL C is typed.

4.5.2 SETUP COMMAND 2

This command prints out the current status of various parameters and then allows the user to change them if desired. The following is a list of the parameters that can be changed under this command.

Enable Halt on break. When set to 1 this parameter enables the processor to halt if the console SLU detects a break condition. When reset, console breaks are ignored. After power up or restart the halt on break bit will not be enabled until after the first break or valid character has been received. This parameter only applies to Qbus systems. On unibus systems, enabling of halt on break is determined by the front panel lock switch.

Disable user friendly format. This parameter determines the types of messages sent to the console at power up. User friendly mode is selected when this bit is a 0. When this bit is a 1 the mode is actually still user friendly but there are some differences. User friendly mode is normally used only when automatic boot mode is selected. This parameter does not apply to unibus systems, the standard format is always selected for unibus systems.

ANSI Video terminal present. When set to 1 this indicates that the console terminal is an ANSI video terminal. When 0 it indicates that the console terminal is hard copy or non ANSI video. When video terminal is selected the DELETE key will erase the previous character on the screen. The rom code accomplishes this by sending a backspace, space then backspace to the console terminal. When hardcopy terminal is selected and the DELETE key is used the rom code identifies deleted characters by using the slash character. At power up if ANSI video terminal is selected the rom code will send an ANSI screen clear and then position the cursor at line 9 coloumn 1. The video terminal present parameter is used only by the rom code. This information is not used by the operating system.

> NOTE: VT52's are not ANSI compatible. If a VT52 is present you must set this parameter to 0 to prevent the clear screen command from locking up the VT52.

Power up mode and restart mode. These are two separate parameters. When the rom code is started it checks a status bit to determine if the unit is powering up or if the restart switch was activated. The rom code then uses the approppriate mode selected. There are four choices for either the power up mode or the restart mode. The choices are the same for either.

0 - Dialog mode. At completion of the diagnostics, dialog mode is entered.

1 - Automatic mode. At completion of the diagnostics the rom code enters an automatic boot routine that will try to boot a previously selected device or device's. The device's are selected in the EEPROM. The list of devices can be from 1 to 6 devices long. Each device is tried until a successful boot occurs or there are no more devices to try. The default list of devices is A, DL0 and MS0 in this order. "A" is a special single letter mnemonic which tries to boot a MSCP device in the range of unit 0 to 7.

2 - ODT mode. At completion of a very limited set of tests the rom code executes a halt instruction and passes control to J11 micro ODT. If the user types P at this point without changing any registers the rom code will continuing normal testing and then enter dialog mode. This mode normally only be used in debug environments.

3 - 24 mode. At completion of a limited set of tests the rom code loads the PSW with the contents of location 26 and then transfers control to the address located in location 24. This mode is used when battery backup memory or non volatile memory is present and power fail recovery is desired.

Ignore battery. This parameter is used only when the current power up or restart mode is set to 24 (3). Normally this parameter is set to 0 meaning that the memory battery ok signal must be present in order to execute mode 24. If this parameter is set to 1 mode 24 is executed regardless of the status of the battery. If ignore battery is 0 and the current mode is power up mode with the battery status indicating that volages were not maintained then use the restart mode to determine the action. If the restart mode is mode 24 also, then default to dialog mode.

PMG count. This parameter sets the value of the processor mastership count in the BCSR. The range is 0 to 7. When set to zero the counter is disable. When set, the count value enables the KDJ11-B to suppress DMA requests and give the processor bus mastership during the next DMA arbitration cycle after the counter overflows. The following table shows the time needed for the counter to overflow for the different values of the PMG count. This parameter is normally set to 0.

Value	Time for counter to overflow
0	disabled
1	0.4 usec
2	0.8 usec
3	1.6 usec
4	3.2 usec
5	6.4 usec
6	12.8 usec
7	25.6 usec

Disable clock CSR. When set to 1 this parameter disables the clock CSR at address 17777546. When set to 0 the clock CSR is enabled. This parameter is normally set to 0.

Force clock interrupts. When set to 1 the clock will unconditionally request interrupts when the processor priority is 5 or less. When set to 0 the clock can request interrupts only if the clock CSR is enabled, clock CSR bit 6 is 1 and the processor priority is 5 or less. This parameter is normally set to 0.

NOTE: If force clock interrupts is selected the user should always disable the clock CSR.

Clock select. This parameter determines the source of the clock to be used. The choices are listed below.

Value	Source
0	Clock sourced from backplane pin BR1. The power supply normally drives this signal at 50 or 60 Hz.
1	Clock sourced internally at 50 Hz
2	Clock sourced internally at 60 Hz
3	Clock sourced internally at 800 Hz

Enable ECC test. When set to 1 this parameter enables the ECC memory test to be run if the memory is an ECC type memory. This test is automatically disabled for parity type memories. The rom code uses bit 4 of the memory CSR to determine if the memory is ECC or parity. If bit 4 is a read/write bit and can be written as a 1 and a 0 then the rom code assumes the memory is ECC. When this parameter is set to 0 the ECC test is always bypassed. This parameter is normally set to 1 even when only parity memory is present. The ECC test is never run on unibus memory.

Disable long memory test. When set to 1 this parameter bypasses the memory address shorts data test for all memory above 256 K bytes. When set to 0, the address shorts data test is run on all available memory. This parameter is normally set to 0.

NOTE: If the long memory test is disabled and parity memory exists above 256 K bytes then it is very likely after power up that the parity memory above 256 K bytes will contain parity errors.

Disable ROM. This parameter allows the user to selectively disable all or part of the rom code after the selected device has been booted. Normally the rom code responds to two 256 word pages in the I/O page. One page responds to address's from 17773000 to 17773777, the other page responds to address's from 17765000 to 17765777. Both of these pages are automatically enabled at power up or restart. After a device is booted one or both of these pages may be disabled by the rom code. The following table lists the variations of this parameter. This parameter is normally set to 0.

Value	Rom page/s disabled.		
0	None		
1	17765000-17765777		
2			17773000-17773777
3	17765000-17765777	and	17773000-17773777

Enable trap on halt. If this parameter is set to 1 the processor will trap to location 4 if a halt instruction is executed in kernel mode. If this parameter is set to 0 the processor will enter J11 micro ODT if a halt instruction is executed in kernel mode. This parameter is normally set to 0.

Allow alternate boot block. After the boot block of a device is loaded into memory the rom code looks at word location 0 to see if the device looks bootable. If the data is not correct the rom code will type out an error message indicating that the media is not bootable. When this parameter is set to 1 the rom code looks for location 0 be any non zero number. If this parameter is set to 0 the rom code looks for location 0 to be a value of 240 to 277 and for location 2 to be 400 to 777. This parameter is normally 0 but may have to be changed to 1 to allow some user's operating systems to boot properly.

NOTE: The following parameters apply only to unibus systems.

Enable unibus memory test. If this parameter is a 1 then any available unibus memory will be tested. When reset unibus memory is not tested. This parameter is normally set.

Disable UBA ROM. This parameter is copied to bit 3 in the DCSR of the UBA after a normal boot. When this bit is set, the UBA roms are disabled. This allows other rom boards on the unibus to show up now. When this bit is reset the UBA roms are enabled. This bit is normally cleared. When a user tries to boot either a UBA or M9312 rom boot, this parameter is ignored.

Enable UBA cache. This parameter when set causes the UBA cache to be enabled and to be tested by the rom code. If a failure occurs during testing of the UBA cache then it will be disabled. When reset the UBA cache is always disabled. This parameter is normally set.

Enable 18 bit mode. This bit is copied to bit 5 of the KMCR on the UBA. When set this causes 18 bit addressing only. When reset 22 bit addressing occurs. This parameter is normally reset.

The following is an example of the parameters print out for a Qbus system. The value of the parameters shown is the default values.

```
List/change parameters in the Setup table

     A - Enable Halt on break                 0=No,    1=Yes   = 0
     B - Disable User friendly format         0=No,    1=Yes   = 0
     C - ANSI Video terminal (1)              0=No,    1=Yes   = 1
     D - Power up  0=Dialog, (1)=Automatic, 2=ODT, 3=24        = 1
     E - Restart   0=Dialog, (1)=Automatic, 2=ODT, 3=24        = 1
     F - Ignore battery                       0=No,    1=Yes   = 0
     G - PMG count                                     (0-7)   = 0
     H - Disable clock CSR                    0=No,    1=Yes   = 0
     I - Force clock interrupts               0=No,    1=Yes   = 0
     J - Clock  0=Power supply, 1=50Hz, 2=60Hz, 3=800Hz        = 0
     K - Enable ECC test (1)                  0=No,    1=Yes   = 1
     L - Disable long memory test             0=No,    1=Yes   = 0
     M - Disable ROM 0=No, 1=Dis 165, 2=Dis 173, 3=Both        = 0
     N - Enable trap on Halt                  0=No,    1=Yes   = 0
     O - Allow alternate boot block           0=No,    1=Yes   = 0

Type CTRL Z to exit or press the RETURN key for No change

Enable Halt on break             0=No,    1=Yes   = 0  New =
```

The following is an example of the parameters print out for a unibus system. If 124 KW of unibus memory is present then the last two parameters will not be present (Enable UBA cache and enable 18 bit mode). When this condition occurs UBA cache is always disabled and 18 bit mode is forced unconditionally.

```
List/change parameters in the Setup table

    A - ANSI Video terminal (1)           0=No,    1=Yes   = 1
    B - Power up  0=Dialog, (1)=Automatic, 2=ODT, 3=24   = 1
    C - Restart   0=Dialog, (1)=Automatic, 2=ODT, 3=24   = 1
    D - Ignore battery                    0=No,    1=Yes   = 0
    E - PMG count                                  (0-7)   = 0
    F - Disable clock CSR                 0=No,    1=Yes   = 0
    G - Force clock interrupts            0=No,    1=Yes   = 0
    H - Clock  0=Power supply, 1=50Hz, 2=60Hz, 3=800Hz   = 0
    I - Enable ECC test (1)               0=No,    1=Yes   = 1
    J - Disable long memory test          0=No,    1=Yes   = 0
    K - Disable ROM 0=No, 1=Dis 165, 2=Dis 173, 3=Both   = 0
    L - Enable trap on Halt               0=No,    1=Yes   = 0
    M - Allow alternate boot block        0=No,    1=Yes   = 0
    N - Enable unibus memory test (1)     0=No,    1=Yes   = 1
    O - Disable UBA ROM                   0=No,    1=Yes   = 0
    P - Enable UBA cache (1)              0=No,    1=Yes   = 1
    Q - Enable 18 bit mode                0=No,    1=Yes   = 0

Type CTRL Z to exit or press the RETURN key for No change

ANSI Video terminal (1)           0=No,    1=Yes   = 0  New =
```

When setup mode command 2 is executed the rom code prints out the current status of all parameters and then repeats the first parameter and then waits for user input. The user can type RETURNs to position the program at the desired parameter to be changed. The user can also go directly to the parameter by typing the letter to the left of the parameter in the first list. To change a parameter the user types in the new value and presses the RETURN key. Typing RETURN, Line feed or . will cause the rom code to proceed to the next parameter. Typing ^ or - will cause the rom code to proceed to the previous parameter. Any of these characters can be used to change a value.

4.5.3 SETUP COMMAND 3

This command prints out the current contents of the translation table and allows the translation table to be changed. The translation table is used to allow devices to be booted using non standard CSR addresses. When the rom program enters the boot routine, R0 contains the unit number and R2 contains the device name (mnemonic). The rom code tries to find a match in the translation table for the device name and unit number. If no match is found the boot program will use the default CSR address for the device. If a match is found the translation table will define the CSR address to be used. The translation table was needed to allow multiple MSCP devices with different controllers to be booted. The following is an example of the command be executed with all entries in the table being blank (default).

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 3

List/change boot translations in the Setup table

TT1     blank
TT2     blank
TT3     blank
TT4     blank
TT5     blank
TT6     blank
TT7     blank
TT8     blank
TT9     blank

Type CTRL Z to exit or press the RETURN key for No change

TT1     blank
Device name     ■
```

The rom code is now waiting for the user to enter a new device name. If the user does not desire to change any items in the translation table, he/she would Type CTRL Z to return to the setup mode prompt. The user may skip over any entry and go to the next entry by pressing the RETURN key for each entry to skip. When the user desires to enter a new entry or change an entry he/she types in the new device name, the unit number and the CSR address. The following describes a possible example of the use of the translation table.

In this example the user has a system that has one RD52 and a RX50 using a RQDX1 controller at the standard address of 172150. The user also has a RC25 with a KLESI controller. Since the RQDX1 and the KLESI share the same standard CSR address one of them must be set to respond to a different address. In this example the KLESI interface is set to respond to address 17760500. The RC25 has a unit number plug set for units 4 and 5. The RD52 is unit 0 and the RX50 is unit 1 and 2. Since the RC25's interface is at a non standard CSR address and there are two unit numbers there will be two entries in the translation table for units 4 and 5.

```
TT1      blank
Device name      = DU
Unit number      = 4
CSR address      = 17760500
TT1      DU4 address 17760500

TT2      blank
Device name      = DU
Unit number      = 5
CSR address      = 17760500
TT2      DU5 address 17760500

TT3      blank
Device name      =

TT4      blank
Device name      =

TT5      blank
Device name      =

TT6      blank
Device name      =

TT7      blank
Device name      =

TT8      blank
Device name      =

TT9      blank
Device name      =
```

The translation table also provides a means of handling multiple controllers such as RL02 controllers. For example if the user had two RL02 controllers with six drives of which 2 where on the second controller at address 17760400 the translation table could be set up to handle this. Drives 0-3 would be on the first controller at the standard address and would not require any entries. Drives 0 and 1 on the second controller would be labeled as drives 4 and 5 and entered into the translation table. Since RL02 controllers only recognize unit numbers from 0-3 the unit numbers 4 and 5 would have to be translated to unit numbers 0 and 1. The following shows the entries into the translation table for this example.

```
TT1     blank
Device name     = DL
Unit number     = 4 0
CSR address     = 17760400
TT1     DL4 = DL0 address 17760400

TT2     blank
Device name     = DL
Unit number     = 5 4
CSR address     = 17760400
TT2     DL5 = DL1 address 17760400

TT3     blank
Device name     =
```

4.5.4 SETUP COMMAND 4

This command allows the user to select the devices to be tried in the automatic boot sequence. The user creates a small list that defines the devices and the order in which they are to be tried. One entry is needed to define a device and its unit number. If the same device is used more than once with different unit numbers, then one entry is needed for each unit number. There a three special single letter device names. The first one is the letter A which causes the rom code to find up to eight MSCP devices (units 0-7) at the standard CSR address and determine if they have removable or fixed media. The rom code will then try to boot one at a time each removable media device and then try to boot each fixed media device. MSCP units not associated with the standard CSR address are not affected by this mnemonic and must be selected individually. The second single letter mnemonic is the letter B. This mnemonic causes the rom code to check for an off board rom at address 17773000. If an off board rom exists and the first location is not zero the rom code will disable the internal rom code and jump to location 173000 in the off board rom. The third mnemonic is the letter E. The only purpose of this mnemonic is to indicate to the rom code that there are no other devices to try in the list. This is used when there are five or less devices in the list. It follows the last device in the list to be tried. When this command is executed the rom code will prompt the user for a device name. The user would then type in either the single or double letter mnemonic assocciated with the device to be selected. The rom code will then prompt for the unit number. The rom code will continue prompting for all six entries in the table. The following is an example of a user adding DY0 to the table at the end.

NOTE: The selections for this command use the same
      locations in the EEPROM as command 6 below.

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 4

List/change the Automatic boot selections in the Setup table

Boot 1 = A    MSCP Automatic boot
Boot 2 = DL0
Boot 3 = MS0
Boot 4 = E    Exit Automatic boot
Boot 5 = blank
Boot 6 = blank

Type CTRL Z to exit or press the RETURN key for No change

Boot 1 = A    MSCP Automatic boot
Device name   =

Boot 2 = DL0
Device name   =

Boot 3 = MS0
Device name   =

Boot 4 = E    Exit Automatic boot
Device name   = DY
Unit number   = 0

Boot 5 = blank
Device name   = E

Boot 6 = blank
Device name   =

KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key:
```

4.5.5 SETUP COMMAND 5

This command allows the user to define an optional message to be sent to the terminal at the beginning of the rom code and at any time a CTRL Q is received by the console that is not in response to a previous CTRL O (i.e. not an XOFF/XON). This entry would normally only be used in systems that have terminals that do not power up with the currect language characters selected. There is a one to ten byte message for English and another one to ten byte message for the resident foreign language if there is any. Only one of the messages is sent and it is the one that matches the current selected language. The message must be enter in octal form only. To enter an ESC character you would type in 033, an A would be 101 etc. The first byte of 000 will terminate the message. The default value for both messages is all ten bytes set to 000. The appropriate message is also sent any time the current language is changed.

NOTE: Please note that this parameter will seldom be used.

The following is an example of setup mode command 5.

```
Type a command then press the RETURN key: 5

List/change the terminal Setup message in the Setup table

Non
ENGLISH               ENGLISH
0 = 000               0 = 000
1 = 000               1 = 000
2 = 000               2 = 000
3 = 000               3 = 000
4 = 000               4 = 000
5 = 000               5 = 000
6 = 000               6 = 000
7 = 000               7 = 000
8 = 000               8 = 000
9 = 000               9 = 000

Type CTRL Z to exit or press the RETURN key for No change

Non ENGLISH
0 = 000 New = 031
1 = 000 New = 042
2 = 000 New =
3 = 000 New =
4 = 000 New =
5 = 000 New =
6 = 000 New =
7 = 000 New =
8 = 000 New =
9 = 000 New =

ENGLISH
0 = 000 New =
1 = 000 New =
2 = 000 New =
3 = 000 New =
4 = 000 New =
5 = 000 New =
6 = 000 New =
7 = 000 New =
8 = 000 New =
9 = 000 New =
```

4.5.6 SETUP COMMAND 6

This command allows the user to define the value of three of the switches at the edge of the CPU to boot specific devices. These three switches plus another are paralleled to a connector at one end of the board. The switches that go to this connector are 1-4. The command defines switches 2-4. There are six entries in the table that are referred to as SB 1 to SB 6. The switches match up as follows:

SW2	SW3	SW4	
On	On	On	Special configuration. See EEPROM definition near end of spec.
On	On	Off	SB 1 selection determines boot device
On	Off	Off	SB 2 selection determines boot device
On	Off	Off	SB 3 selection determines boot device
Off	On	On	SB 4 selection determines boot device
Off	On	Off	SB 5 selection determines boot device
Off	Off	On	SB 6 selection determines boot device
Off.	Off	Off	Action determined by EEPROM. This is the normal configuration.

Note: If switch 5 is off and the baud rate selector switch is 8 or greater then force dialog mode is selected and this table is overridden.

When one of the selections (SB 1 to SB 6) is selected the rom code will attempt to boot only the device selected.

The following is an example of setup mode command 6.

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 6

List/change the switch boot selections in the Setup table

Switches 2,3,4  on    on    off    = DX0
Switches 2,3,4  on    off   on     = DL2
Switches 2,3,4  on    off   off    = DL0
Switches 2,3,4  off   on    on     = E0
Switches 2,3,4  off   on    off    = blank
Switches 2,3,4  off   off   on     = blank

Type CTRL Z to exit or press the RETURN key for No change

Switches 2,3,4  on    on    off    = DX0
Device name     =

Switches 2,3,4  on    off   on     = DL2
Device name     =
```

4.5.7 SETUP COMMAND 7

This command performs the exact same function as the List command in Dialog mode that was previously described. This command will list out all available boot programs in the rom code and will also list any programs available in the EEPROM. The information listed is the device name, unit number range, source of the boot program and a short device description. The device name is normally a two letter mnemonic. In some cases the name may be a single letter. The device name must always be letters from A to Z. At input the rom code always converts all lower case letters to upper case. The unit number range is the allowable range of unit numbers that is valid for a particular boot program. The range varies from 0 only to 0 to 255 depending on the device. The source lists where the boot program resides. The programs reside in either the rom code or in the EEPROM. The description is intended to be the name of on the outside of the device to be booted. An example would be for a device name of DL the description would be RL02 which is the name on the outside of the physical device itself. Setup mode is restarted at the completion of the list command (7).

4.5.8 SETUP COMMAND 8

This command initializes the current contents of the setup table in memory to the default values. This command does not affect the contents of the EEPROM itself. The save command that follows must be executed in order to save the setup table into the EEPROM.

4.5.9 SETUP COMMAND 9

This commands copies the current contents of the setup table in memory into the EEPROM. This is the only command that actually writes anything into the first 105(10) bytes of the EEPROM.

4.5.10 SETUP COMMAND 10

This command will restore the setup table in memory with the values actually stored in the EEPROM.

The following five commands allow the user create and edit boot programs that are stored in the EEPROM.

4.5.11 SETUP COMMAND 11

If this command is executed the rom code will ask the user for the device name of the EEPROM boot to be deleted. After the device name is inputted the rom code will look for the first boot program in the EEPROM and delete it if found. If there are any boot programs following the deleted program the rom code will automatically move all of these programs up to use the space made available by the deleted program.

4.5.12 SETUP COMMAND 12

If this command is executed the rom code will ask the user for the device name of an EEPROM boot to be loaded in memory. This command is normally used to allow a user to load an EEPROM boot program so it can be examined and/or edited.

4.5.13 SETUP COMMAND 13

This command is used to either create a new EEPROM boot program or to edit a program previously loaded with command 12 above. This command allows the user to change the device name, the device description, the allowable unit number range, the beginning address and ending address of the program in memory and the start address of the program. When these changes are complete the rom code enters rom ODT which is a rom code version of J11 micro ODT. When this command is first entered it will list the available space in the EEPROM for boots.

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 13

Edit/create an EEPROM boot

Type CTRL Z to exit or press the RETURN key for No change

1410 Bytes free in the EEPROM

Device name              = AA          New = EA

Beginning  address       = 000600         New = 10000

Last byte address        = 000615         New = 10177

Start address            = 000600         New = 10000

Highest Unit number      = 3              New = 255

Device Description       = EA BOOT        New = RM02/03

ROM ODT> 010000/000000 012705
ROM ODT> 010002/000000 101
ROM ODT> 010004/000000 12706
ROM ODT> 010006/000000 1000
```

etc. Type CTRL Z exit back to the setup mode menu.

The beginning address is the first location of the program in memory.

The last byte address is the address of the last byte of code used in memory. If in doubt, use the last address of data + 2 for this value. Do not use a much larger number because this wastes EEPROM space.

The start address is the address that the rom code will pass control to.

The highest unit number defines the allowable range of valid unit numbers for this device. If the value is set to 3 the allowable range is 0 to 3. The highest range is 0 to 255. If a unit number is typed in at boot time and it is not in range then a invalid unit number error will occur.

The device description is an optional but recommended description of the device name. The maximum length of this name is 11 characters or spaces. The name should normally be the name that is physically marked on the outside of the device (i.e. RL02).

4.5.14 SETUP COMMAND 14

This command allows the user to save the existing boot program in memory into the EEPROM. This is the only command that actually writes a boot into the EEPROM. The other commands only change a copy of the boot program that resides in memory. When saving a boot program into memory the device name of the program must not match the name of an existing program in the EEPROM. If the program name already exists the user must delete that program first or change the name of the program to be saved. If two or more programs were written into the EEPROM with the same name only the first one would be bootable.

4.5.15 SETUP COMMAND 15

This command puts the user into ROM ODT. The rom code will open up the address defined by the beginning address of the program. ROM ODT is not the same as J11 micro ODT.

In ROM ODT the only allowable addresses that can be examined are the addresses of memory from 0-28 KW (0-00157776). Any other addresses and any attempt to accessed the I/O page or any registers is not allowed.

Command	Symbol	Use
Slash	/	Prints contents of specified location or if no address is defined then print contents of the last location that was opened. If location opened is an odd number then print out only the contents of the byte. If location is even then mode is even, if location is odd then mode is byte. Leading zeroes are assumed. Only the last six octal digits are used.

Examples:

```
ROM ODT > 200/100000               ;Open location 200

ROM ODT > 1001/240                 ;Open byte location 1001

ROM ODT > 77777750020/100000       ;Open location 00150020

ROM ODT > 77770000/                ;Illegal location > 157776
ROM ODT >
```

RETURN	<CR>	Closes an open location
LINE FEED	<LF>	Closes an open location and then opens the next location. If word increment by 2, if byte by 1.
Period	.	Alternate character for line feed. This command is useful when the terminal is a VT2xx series terminal. It is also convenient to use with the keypad.
Up arror	^	Closes an open location and then opens the previous location. If in word mode then decrement by 2, if byte decrement by 1.
Minus	-	Alternate character for Up arror. This command is useful when the terminal is a VT2xx series terminal. It is also convenient to use with the keypad.
Delete	DELETE	Deletes the previous character typed.
CTRL Z	^Z	Exit ROM ODT and return to setup mode.

4.7 POWER UP MODE OR RESTART

When the boot code is started the code determines if it is in a power up sequence or a restart sequence. For either sequence there are four possible modes of operation. Restart is determined by checking the reboot pulse bit in the BCSR for Qbus systems or in the KMCR for Unibus systems. The selection of the mode is stored in the EEPROM or selected by the switches on the Cpu.

MODE 0 - (DIALOGUE MODE)

Execute diagnostics determined by the contents of the EEPROM. Then dispatch to Dialogue Mode. Operator may now boot a device, enter setup mode or select continuous testing.

MODE 1 - (AUTO BOOT MODE)

Execute diagnostics determined by the contents of the EEPROM. Then dispatch the program to Auto Boot Mode (Turnkey) and execute the selected boot program.

MODE 2 - (ODT)

Execute a HALT instruction in the rom code with the PCR = 000000. J11 micro ODT will now be entered and the contents of the PC will be printed out on the terminal. The instruction following the HALT instruction will return program control to Dialogue Mode if ODT proceed is executed and the PC & PCR has not been changed.

The KDJ11-B is different than the KDJ11-A module, such that, the KDJ11-B executes Macro level code before entering ODT, thus, the PC and other internal J11 registers will be modified.

MODE 3 - (24/26)

Load the PSW with the contents of location 26 and then transfer control to the address specified in location 24. This mode is used by systems with non volatile memory or for some communications devices that have remote load detect and restart capabilities. For battery backed up Unibus systems, if mode 3 is the selected power up mode and the Battery is dead, the selected restart mode will be executed.

4.8 MESSAGE PRINTOUT CONSTRAINTS

As a result of power up self test in the newer DEC terminals, which may take three or more seconds to execute when power is applied, certain constraints are imposed on the software trying to output messages to the terminal. The main constraint on the software is that it must wait for the terminal to indicate that it is ready to accept data. The terminal indicates this by transmitting a XON character at the end of its self test. Any data sent to the terminal before the XON character is transmitted will be ignored by the terminal. Another constraint imposed on the software is that the terminal may already be powered up, therefore a XON character will not be transmitted, or the terminal may be of the older type which does not support the XON protocall on power up.

This rom code handles terminals powering up as follows. At the beginning of the rom code various tests are run. Within a half second a console test is run using the DLART maintenance mode feature. When this test is complete the first message is sent to the console. The rom code assumes that the console device is ready to receive messages. If during the testing the rom code receives an XON indicating that a terminal just became available the must important test in progress messages will be retype. If during an error an XON is received the rom code will retype the error message. Once dialog mode is enter the XON will be ignored. The user could get the current message retyped by typing RETURN.

4.9 BOOT CODE

This portion of the boot and diagnostic rom code provides the primarv bootstrap for the devices as shown below for the Qbus and the Unibus. These will be supported in the KDJ11-B Boot ROM.

Unibus Bootstraps

Mnemonic	Device
DU	MSCP
DK	RK05
DL	RL02
DX	RX01
DY	RX02
DD	TU58

Qbus Bootstraps

Mnemonic	Device
DU	MSCP
DL	RL01/02
DX	RX01
DY	RX02
DK	RK05
DD	TU58
XH	DEQNA
NU	DUV11
NE	DLV11-E
NF	DLV11-F
MS	TS05, TK25

The primary boot program normally reads 256 words from the device into memory starting at location 0. If the secondary bootstrap is loaded without any errors the rom code will transfer control to location 0 with MMU off, R0 equal to the unit number of the device booted and R1 equal to the base address of the device CSR. For some devices R1 will be the base address plus an offset.

The rom code boot programs will attempt to detect errors during the boot process and take the appropriate action. The following is a list of possible errors the rom code will try to detect. Not all errors are applicable for all boot programs.

Drive not ready
Non bootable media in the drive
No disk or drive unloaded
No tape
Non existent controller
Non existent drive
Invalid unit number
Invalid device
Controller error
Drive error

After successful completion of the loading of a secondary bootstrap the display is blanked out. Before transferring control to the secondary boot the rom code will print out "Starting system".

4.9.1 ERROR MESSAGES FOR DISK AND TAPE BOOT PROGRAMS

The following are the error messages associated with the boot programs for disks, tapes and DECNET devices. These errors are applicable to all CPU ROM resident boots and any EEPROM boots which have been written to pass these errors back to the CPU ROM. The only errors that can apply to UBA or M9312 type rom boots are 14, 16 and 17.

Message 10
Drive not ready

Message 11
Non bootable media in the drive

Message 12
No disk or drive unloaded

Message 13
No tape

Message 14
Non existent controller, address 17772152

Message 15
Non existent drive

Message 16
Invalid unit number 5

Message 17
Invalid device

Error 20
Controller error

Error 21
Drive error

There is also a general purpose error message used only on Qbus machines with user friendly mode selected and automatic boot mode selected. The message is a general statement telling the operator to check for something obvious like no disk or drive not ready. The message is:

```
Waiting for media to be loaded, or drive to go ready

    Press the RETURN key when ready to continue
```

The following are examples of errors from boot programs:

Example 1. Bootstrap program error.

```
Commands are: [Help, Boot, List, Setup, Map, Test]
Type a command then press the RETURN key: B DL1

Trying DL1
Message 12
No disk present, or drive is not loaded

Command    Description

  1        Reboot
  2        Go to Dialog mode

Type a command then press the RETURN key: 2
```

Example 2. Bootstrap program error.

```
Commands are: [Help, Boot, List, Setup, Map, Test]
Type a command then press the RETURN key: B DL3

Trying DL3
Message 15
Non existent drive

Commands are: [Help, Boot, List, Setup, Map, Test]
Type a command then press the RETURN key:
```

Example 3. Bootstrap program error.

```
Trying DU1
Message 14
Non existent controller, address 17772152


Commands are: [Help, Boot, List, Setup, Map, Test]
Type a command then press the RETURN key:
```

5.0 FUNCTIONAL TEST DESCRIPTIONS

5.1 KDJ11-B STANDALONE MODE TESTS

When the CPU is powered up or restarted the DCOK signal causes the display to be set to 77. If the CPU hangs with the display set to 77 then either the halt switch is on or the CPU does not have enough logic functioning to execute an instruction out of the rom.

The first test number is 76. The tests then count down from here.

TEST 76

Set the LED display to 76 to indicate the first instruction has been executed without hanging up the processor. Enter standalone mode, set PSW to priority of 7 and turn off MMU. Clear PCR and set up SP. Jump to the high page of the rom if not already there (173xxx).

Read the switch pack at the edge of the module (BCR). If switch 1 if off and switches 2-3 are on then halt immediately. If operator proceeds then continue as normal.

Execute a few simple CPU tests. General register writes and reads, branch instructions, and a simple JSR instruction.

Test all 48 PAR's using a rotating ones pattern and a address uniqueness pattern. Check all PDR read/write bits (14:8 and 3:1) using a rotating ones pattern and an address uniqueness pattern. At completion of test leave all PDR's set to 077406. Set KIPDR0 to 077402.

Verify that the second page of the rom code can be selected in the low byte of the PCR and that the page shows up at address 165xxx. Jump to the beginning of the second page of the rom in address range 165xxx.

Execute a stack pointer register test. Test all three stack pointers.

Test MMR2 and verify that it logs the address of the instruction fetch.

Test MMR3 bits 0-5 with and decrementing pattern from 77 to 0.

Jump to the high page running from 173xxx.

Enable the trap area into the low page at address 165xxx.

Enable 22 bit mapping and turn on the MMU.

Verify that the trap area is available with MMU on.

TEST 75

Test the ability to write data with MMU using all three modes and both spaces. Test that the W bit will set properly in one PDR. Initialize all unused PAR's and PDR's.

For the duration of the testing MMU will always be on but only Kernel I space will be used. All free PAR's and PDR's will be used as flag storage, keyboard input storage, the stack etc.

Test the IOT, TRAP, BPT and EMT instructions. Test that the central subroutine call program works and that the change page routine works.

From now on the rom will execute virtually all tests running out of the high page addresses (17773xxx) and use the low page addresses (17765xxx) to handle all traps etc to virtual addresses 0-276. Jumping to another page, calling a routine in another page, etc is accomplished by executing EMT, TRAP and BPT instructions which transfer control to the low page. Routines here determine what action is to take place now.

Transmit a nul character to the console. If the console address times out ignore the timeout for now.

Change page to the next page in the rom code.

TEST 71
EEPROM TEST

With the program running out of the high byte PCR area (17773000 - 17773776), set bit RS3 65 in the Boot and Diagnostic Control Status Register (BCSR) to select the 8-bit ROM (EEPROM). The 8-bit ROM will be enabled into address area (17765000 - 17765776). To check that the 8-bit ROM was selected, read location 165314 in this area and check that the low byte of the location is readback as 252. Also check location 165022 and verify that it is 0. If either of these locations is not correct then automatically initialize the first 105(10) bytes of the EEPROM to the factory defaults. This allows a new module in stage 1 manufacturing to automatically initialize itself. When this is complete read and accumulate an 8 bit checksum for the first 105(10) bytes of the EEPROM. Verify that the result is zero. If an error occur check to see if the foreign language file is available and go print out the error in the selected language.

Normal error printouts are now fully enabled.

Now check the EEPROM to determine the power up and restart modes. If both modes are either dialog or automatic boot then proceed to the next test. If force dialog is selected in the switches then proceed to the next test.

If either mode is ODT or 24/26 then check to see if this is a restart or a power up. If the system is a Unibus system then standalone mode must be exited long enough to read the KMCR on the UBA to determine if this is a restart or power up.

TEST 74 For Unibus systems exit standalone mode to read restart bit.

If the selected mode is 24/26 then check the status of the ignore batter status bit in the EEPROM. If ignore battery is selected then execute mode 24/26 unconditionally. If ignore battery is not selected and if the the current condition is power up then use the restart mode to determine the current mode. If the restart mode is 24/26 also then default to dialog mode. If the current condition is restart then default to dialog mode.

TEST 72 Mode 24/26

Save the contents of location 24 in memory. Execute a rotating one's test on location 24 in memory. If the test passes restore location 24 with it's original data and transfer control to the address specified by location 24 and set the PSW with the contents of location 26. The display is blanked before transferring control.

TEST 73 Enter J11 ODT

This is not a test. It is the display that indicates that the selected mode is ODT. If the operator proceeds from ODT without changing any registers the rom code will continue to run selected tests and enter dialog mode when complete.

TESTING CONTINUED

Go map the EEPROM to determine if a foreign language file or a UFD file is present.

TEST 70 CPU ROM checksum and page addressing test

Using the high byte PCR area (17773000-17773776) as the program area, checksum the ROM by using the low byte of the PCR to select the 64 ROM pages. The low byte of the PCR will enable the ROM pages into address area 17765000-17765776. The PCR will be validated each time that it is loaded by checking the next to last location in each page and verifying that each byte contains the selected page number. The only exception to this is pages 70-76(8) which do not contain page numbers due to storage of ascii text.

Using the low byte PCR area (17765000-17765776) as the program area, read each of the 64 pages into the high byte PCR area at 17773000-17773000 and verify that the page number is in each byte of the next to last word in each page.

TEST 67 Misc. CPU tests and EIS test.

Test JSR, RTI and RTS. Test the ASHC instruction.

TEST 66 Console SLU test 1

Test that all four console registers respond..

TEST 65 Console SLU test 2

Set console DLART to maintenance mode. Delay to allow any possible incoming characters to settle. Empty the receive buffer. Transmit two characters (0,377) and verify that both are received properly.

Note: Even when the DLART is in internal loopback mode the transmit output is still connected to the EIA output driver. In order to prevent garbage characters from being printed on the screen the SLU test transmits nul's and delete characters which are non printing.

TEST 64 Console SLU test 3

The test will check that the transmitter and receiver can cause an interrupt and that it interrupts at the correct priority level. Using the transmit logic, cause receiver error bits to be set and check that the receiver errors can be cleared.

Transmit the first message to the screen. If terminal type in the EEPROM is ANSI video and force dialog is not selected then send an ANSI screen clear and position the cursor at line 8 column 1.

Type out "Testing in progress - Please wait"

TEST 63 Test MMU abort logic

Test that the protection bits in the PDR's will cause aborts when the conditions are violated. Verify that abort occurs through virtual address 250. Verify that MMR1 properly records changes in the general purpose registers affected by the abort.

5.2 STANDALONE MODE CACHE TESTS

TEST 62 Cache tests

The cache will not be tested to see if it is flushed after power up. At power up the cache tag fields are loaded with the upper address bits of the cache control register. Since this address is in the I/O page any access to it will cause a miss regardless of the Valid bit. The rom code will check the ability to flush the cache later.

In the following tests any trap to location 114 is a test failure due to parity error on the CPU tag store.

Execute the CCR Read/Write bit test.

Execute the MSER timout test.

Cache data path word test. To check the data path to and from the Cache memory, write and read one location of the Cache every 256 words using a floating ones and then a floating zeroes pattern.

Execute the Hit/Miss register timout test.

Check the ability of the Cache to perform byte operations using a 125 and a 252 data pattern on one location.

Check the 4k word Cache using an address short/data test.

Cache DMA tag store comparator test using force miss in CCR
Set force miss and verify that the cache does not respond.

CACHE CPU AND DMA TAG STORE TIMEOUT TESTS
Test from 40000 to 57776 and verify timout for each location and also verify that DTS CMP and DTS PAR are reset in the Memory System Error Register (MSER). Test one location every 1 KW from 60000 to 17760000 to verify that a timeout occurs and that DTS CMP and DTS PAR are reset in the MSER. This test verifies that CPU and DMA tag fields only respond to the correct addresses for misses.

CACHE FLUSH TEST
Read the 4k word Cache again checking the Cache to contain the address short/data pattern previously written. Flush the Cache and check that the Cache was flushed by reading the locations that were just read. If the Cache was flushed, each of these locations should cause a timeout to occur. Check DTS CMP and DTS PAR to be reset for each location. Set status display to .

CACHE HIT/MISS REGISTER TEST
Test that hits and misses are logged correctly into the Hit/Miss.

CPU CACH TAG STORE FLOATING 1'S AND 0'S TEST

CACHE CPU AND DMA TAG STORE SHORTS DATA TEST

CACHE UNCONDITONAL BYPASS ON WRITES AND READS

CACHE DATA STORE PARITY ERROR TEST

CACHE CPU/DMA TAG STORE PARITY ERROR TEST

TEST 61 Line clock logic test

Test the Line Clock according to the EEPROM configuration data. If the EEPROM indicates that the Clock Status Register (LKS) is enabled, check the LKS register to respond and that bits 6 and 7 and be set and cleared.

Test the ability of the clock to interrupt to location 100 and that it can interrupt at the correct BR Level. On an interrupt, once the clock is in sync, check that bit 7 (LCM) is cleared on an interrupt.

TEST 60 Floating point instruction test

Briefly check some of the floating point instructions.

TEST 57 CPU Commercial instruction set (CIS) tests

If the CIS option is available execute some tests on the instruction set.

5.3 NON STANDALONE MODE TESTS AND MEMORY TESTS

TEST 56 Exit standalone mode

Exit standalone mode and using the guaranteed timout address of 17760000 in 22 bit mode verify that the timout logic works with out hanging up the CPU.

TEST 55 UBA register response test

If Unibus test that the DCSR, KMCR and DDR respond properly on the UBA. With the UBA in diagnostic mode verify that the unibus address, data and control lines that available through the DDR all off.

TEST 54 Memory size routine

For unibus systems first see if any unibus memory is present. If any is present set up the KMCR to correct value to map it properly.
For all systems size memory in 4 KW increments. Memory testing is based on 4 KW blocks.

For QBUS systems size memory from 0 to 2 M words in 1 K word increments. This is done on a one word basis every 1k words. The sizing routine will only detect consecutive memory starting from location 0 and proceeding upward. When sizing the routine reads the contents of the location and then rewrites the data into the location. This is a non destructive memory sizing routine. Sizing proceeds upward until a timout is detected or address 17760000 is reached.

When running in 22 bit mode and the first 128 KW boundary is reached the routine will verify that location 0 and the first location in the second 128 KW can be addressed uniquely. If they cannot be addressed uniquely then the memory is determined to be 18 bit only and memory size is set at 124 KW.

If any non consecutive memory is found, the rom code will type out a message indicating that gaps in memory are present. There will only be a message, no error will occur.

When memory sizing is complete type out all memory size messages that are applicable unless user friendly mode is selected in which the messages are suppressed.

TEST 53 Verify that memory exists at location 0

This test only checks to see if memory at location 0 will respond without timing out.

TEST 52 Test memory from 0 to 4 KW

This test fully checks out the first 4 KW of memory before the main memory tests are loaded and run.

Execute a rotating one's test and a rotating zero's test for Physical address 0.

Test ability to write separate bytes into memory for one word location.

Execute an address short/data test for address bits 0 to 10 for physical addresses 0 to 17776.

TEST 51 Cache test with memory on

Verify that cache is allocated during a read to memory and that cache is bypassed when the cache bypass bit is set.

TEST 50 Memory data/byte test for all of memory

NOTE: Unibus memory is not tested unless enabled in the EEPROM. If unibus memory testing is enabled then the testing must wait until the UBA tests are complete and the UBA is removed from diagnostic mode.

The following tests will walk up through memory in 4k word blocks, testing the first two locations in each 4k words.

```
For physical address 0 to top of memory do by 4k blocks.
  Check location using a rotating 1's and then a
  rotating 0's pattern.
  Test the location for byte operations using 252
  and 125 data pattens.
```

TEST 47 Parity/ECC logic tests for all of memory

The following tests will walk up through memory in 4k word blocks, testing the first two locations in each 4k words. At each 4 k word boundary the test will determine which CSR controls the current address range. If there is a CSR, it's ability to log the address during an error will be checked. The test will also test the ability to abort due to an error.

If bit 4 of the CSR is a read/write bit and ECC testing is enabled in the EEPROM then the test will verify the ability of the ECC logic to correct single bit errors in all 16 bits for a floating one's and zeroes test. This test when enabled is only run on 32 K word boundaries.

TEST 46 Address shorts test for every location in memory

The following test will first be relocated to the first 4 KW of memory to allow the test to execute out of the cache to improve the speed. The MMU will be set up such that any memory address to be tested will have the cache bypass bit set in it's PDR to prevent the cache from responding to the memory location under test. During the tests PDR 1 will be set up to prevent any writes to the first 4K words of memory to protect the test in the cache memory.

```
For physical address 0 to top of memory do by 4k blocks.
  For first address in block to last address in block by 1 word
    Write location with 125252 pattern.

For physical address 0 to top of memory do by 4k blocks
  For first address in block to last address in block by 1 word.
    Read 125252 - Write location with 052525 pattern.

For physical address 0 to top of memory do by 4k blocks
  For first address in block to last address in block by 1 word.
    Read location for pattern 052525
```

At the end of the test verify that none of the memory CSR's has bit 15 set to a one indicating that an error occurred.

5.4 KTJ11-B UNIBUS BUS ADAPTOR MODULE (UBA) TESTS WHEN UBA IS PRESENT

TEST 45 KTJ11-B UBA BOOT ROM LOGIC TEST

Verify that all 256 words of the UBA rom respond to addresses from 17773000 to 17773776.

Disable the UBA rom and check for the presence of a rom board on the unibus. If a rom board exists and responds to all locations then set a flag indicating that it exists. This flag is used latter to allow the rom code to look for boot roms on a M9312 module on the unibus when it exists.

TEST 44 UBA map registers data path test

Test all 32 Unibus Map Register pairs with a rotating one's and zero's pattern and an address uniqueness pattern.

TEST 43 UBA unmapped diagnostic data in/out cycles test

With mapping disabled execute a floating 1's and 0's test through a floating address pattern using diagnostic Data IN and Out cycles for 124 K words of memory if present.

The following tests (42-34) are not run if 22 bit.mode is disabled in the KMCR by the EEPROM or the presence of 124 K words of unibus memory.

TEST 42 UBA mapped diagnostic data in/out cycles test

Test that each mapping register can be indirectly selected and will relocate a physical address from the CPU. The test also checks that if unibus memory is present that the applicable mapping register is disabled.

TEST 41 UBA floating address/data tests using mapped cycles

With mapping enabled execute a floating 1's and 0's test through a floating address pattern using diagnostic Data IN and Out cycles for up to 2044 K words of memory if present. This test floats a one and zero across both inputs to the UBA address summer.

TEST 40 UBA address overflow test

Test that a carry can be rippled across the adder. PA is set to 2 and the map register is set to 17777776 and the resulting address should be 0.

The following tests (37-34) are run only if the UBA cache enable bit is set in the KMCR.

TEST 37 UBA cache data test

Test the 32 cache locations with a floating 1's and 0's pattern. Check the Cache Valid bits and the hit logic.

TEST 36 UBA cache LRU test

Test all 24 valid combinations of the LRU logic in the cache.

TEST 35 UBA cache floating address test in tag store

Execute a floating 1's and 0's test through the cache tag store.

TEST 34 UBA cache parity error detection test

Test that if a ECC error occurs during a read from memory, that the applicable set is invalidated.

The following test is always run on unibus systems.

UNIBUS MEMORY TESTS

If unibus memory exists and unibus memory testing is enabled then testing is done at this point.

Unibus memory tests are the same as the PMI memory tests above.

TEST 33 is the same as test 50

TEST 32 is the same as test 47 with the exception that ECC tests are never run on unibus memory.

TEST 31 is the same as test 46

TEST 30 This is the exit test routine.

Clear the lower 28 K words of memory and dispatch to either dialog mode or automatic boot mode.

5.5 DIAGNOSTIC ERROR MESSAGES

The following is a list of the possible error messages that can get typed out. When an error is printed out there will also be an error number and other information. The error number is always the number of the current that the rom code is running at the time of the error. The only exception is when an unexpected trap occurs, in this case the error is the test number plus 100. An unexpected trap in test number 62 would print out as 162.

M8190 CPU Error
M8190 CPU Cache Error
M8190 CIS Error
M8190 FP Error
M8190 CPU ROM Error
M8190 EEPROM checksum Error
M8190 clock Error
Unibus signal Error
No memory in location 0
Memory Error
Memory CSR Error
M8191 UBA Error
M8191 UBA Cache Error

Unexpected trap to location xxx

When an error occurs the rom code will print out the following information.

1) The error number.
2) A short verbal description of the error.
3) A message referring the user to the troubleshooting documentation.
4) The address of the error. This is broken down into the actual PC, the page number in the rom and the address to refer to in the program listing. In the case of unexpected traps the error address is address following the error of the unexpected trap.
5) The contents of R0-R6 of register set 0 and also the contents of kernel PAR 3. The tests do not use register set 1. Register set 1 is used mainly by rom code support routines.
6) For some memory tests the failing address, good data and bad data will be printed out also.
7) A command line with up to four options for the user to do. The options are:
   1) Rerun the test. If the test passes the rom code will continue testing. If all other tests pass the rom code will print out the total number of errors and enter dialog mode regardless of the mode selection in the EEPROM.
   2) Loop on test. This option causes the rom code to continously loop on the test that failed. Theses loops are generally very large and are not intended to be used as scope loops. The test runs until an error occurs or end of test has been reached, in either case the test is started again and continues to loop until the user types CTRL C at the console.
      At this point the rom code will print out the total number of errors and the total number of successful passes if any. Both the error counter and the pass counter have a maximun value of 65535. If either counter reaches this value the count will not overflow back to zero, it will stay at this value.
   3) Map memory and I/O page. This command is normally used if a memory error occurs. If a memory is not configured properly the map command may point to where the memory actually is. In a multi memory system in which one of the memory fails this command can be used as a method of physically identifying the failing memory if it has a CSR. This command is only available when the bus is turned on. During tests 76 down to 56 this command is not available.
   4) Advance to the next test. This command allows the user to bypass the failing test and continue. This command will only show up for errors that are generally considered non fatal. If the error is fatal and field service would like to bypass the error it is possible by typing CTRL O then typing 4 and. RETURN and this command will be executed anyway.

NOTE: It is important to note that bypassing any error, fatal or non fatal is normally risky !!!!!!!

8) At this point the rom code is waiting for a response from the user.

The following is a typical example of an error that may occur.

```
Testing in progress - Please wait
Memory Size is 256 K Bytes
9 Step memory test
  Step 1 2 3 4 5 6 7 8

Error 46
Memory CSR Error

See troubleshooting documentation

Error PC = 173436       PCR page = 15   Program listing address = 015436

R0 = 060000      R1 = 052525      R2 = 172100      R3 = 172344
R4 = 100000      R5 = 040000      R6 = 172300      Par3 = 010000

Command    Description

   1       Rerun test
   2       Loop on test
   3       Map memory and I/O page
   4       Advance to the next test

Type a command then press the RETURN key:
```

The following is an example of an error in the memory test in which the failing address, failing data and good data is printed out in addition to the standard error information.

```
Testing in progress - Please wait
Memory Size is 256 K Bytes
9 Step memory test
  Step 1 2 3

Error 46
Memory Error

Error PC = 173256       PCR page = 15   Program listing address = 015256

R0 = 060000      R1 = 125252      R2 = 000002      R3 = 052525
R4 = 000100      R5 = 040000      R6 = 172300      Par3 = 001000

Expected data    = 125252
Bad data         = 000002
Address          = 00100000

Command    Description

   1        Rerun test
   2        Loop on test
   3        Map memory and I/O page

Type a command then press the RETURN key:
```

The following is an example of the message printed out when an unexpected trap occurs. The error number of unexpected traps is always the current test number plus 100. In this example the test number was 62 so the error number is 162. The display itself will only display 62 since it is only a two digit display. Unexpected traps are always considered fatal errors.

```
Testing in progress - Please wait

Error 162
Unexpected trap to location 250 MMU

See troubleshooting documentation


Updated PC = 173436       PCR page = 15   Program listing address = 015436

R0 = 101365     R1 = 076410     R2 = 177746     R3 = 177744
R4 = 101367     R5 = 000250     R6 = 172276     Par3 = 052400

Command    Description

   1       Rerun test
   2       Loop on test

Type a command then press the RETURN key:
```

5.6 BRIEF DESCRIPTIONS OF TEST AND STATUS NUMBERS FOR DISPLAY

77 CPU hung or Halt switch on at power up or restart

76 First Cpu pretests, MMU register tests

75 Turn MMU on and run MMU tests & CPU tests

74 For Unibus systems turn on PMI to look at the UBA reboot bit

73 Power up to ODT

72 Power up to 24/26

71 EEPROM checksum tests

70 Cpu ROM checksum & PCR test

67 Misc Cpu tests and EIS test

66 SLU test1 - Check for each register's response

65 SLU test2 - Xmit & receive data in maintenance mode

64 SLU test3 - Check interrupts & errors in maintenance mode

63 Test MMU aborts

62 Samode Cpu cache tests

61 Clock test

60 FPP test

57 CIS test

56 Samode exit - Check location 17760000 for timout

55 UBA register response test, check unibus through DDR for hung lines

54 Memory size routine

53 Check memory present at location 0

52 0-4 kw memory test

51 Cache test using memory

50 Memory test 1 - Data tests/byte tests

47 Memory Parity/ECC tests

46 Memory address/shorts test

45 UBA rom response test

44 UBA map registers data path test

43 UBA unmapped diagnostic cycles test

42 UBA mapped diagnostic cycles test

41 UBA floating address/data tests using mapped diagnostic cycles

40 UBA address overflow test

37 UBA cache data test

36 UBA cache LRU test

35 UBA cache floating address test in tag store

34 UBA cache parity error detection test

33 Unibus memory data path test

32 Unibus memory parity logic test

31 Unibus memory address/shorts test

30 Test exit routine

Test numbers and status numbers continued.

27 spare - not used
26 spare - not used
25 spare - not used
24 DECNET boot (DLV11-E/F or DUV11) waiting for a reply from host
23 XON not received after XOFF - To correct type CTRL Q at console
22 Xmit ready bit never sets
21 Drive error
20 Controller error
17 Boot device selection was invalid (i.e. AA)
16 Invalid unit number selected
15 Non existent drive
14 Non existent controller
13 No tape
12 No disk
11 Invalid boot block
10 Drive not ready
07 No bootable device found in automatic boot mode (sniffer)
06 Console disabled by switch 1 = On and no force dialog or
APT break received and rom code has entered ODT for APT
05 spare
04 Dialog mode
03 UBA rom boot in progress
02 EEPROM boot in progress
01 Cpu rom boot in progress
00 Start boot with display blanked

6.0 FORMAT FOR THE EEPROM

The following sections show the format of the EEPROM. For the standard 8K by 8 bit EEPROM there are thirty two pages of data of 256(10) bytes each. The EEPROM is divided into four areas for different purposes. The first 105 bytes of the first page stores the base hardware parameters for the KDJ11-B CPU module and the KTJ11-B UBA module. This area also contains information on the boot device. The remainding area of the EEPROM is divided into three areas. The area immediately following the base area of the first 105 bytes can be loaded with optional bootstrap programs. When properly formatted these programs can be loaded and started by the base rom code. If foreign language is to be supported the foreign text is located at the end of the EEPROM. Immediately before the foreign language text is the User Friendly Diagnostic system description area. If a 2K by 8 bit EEPROM is used the format is the same.

```
                          -----------------------------------
PAGE 0                   | Base area                          |
1ST 105 BYTES            | CPU and UBA Hardware parameters    |
                         | Boot device information            |
                         | Translation table                  |
                         | Selection information              |
                         |------------------------------------|
                         | Optional bootstrap # 1             |
                         |------------------------------------|
                         | Expansion for bootstraps 2 & up    |
                         |                                    |
                         |                                    |
                         | Expansion for UFD area             |
                         |------------------------------------|
                         | UFD system description area        |
                         |------------------------------------|
                         | Optional Foreign language text or  |
End of EEPROM            | UFD area if no foreign language    |
                          -----------------------------------
```

Section 6.1 below gives a graphic representation of the first 105 bytes of the EEPROM.

Section 6.2 gives a detailed description of the EEPROM diagnostic/boot format for the first 105 bytes in page 0.

Section 6.3 gives a graphic representation of the rest of the EEPROM.

Section 6.4 gives a detailed description of the format to be used by users to write a new bootstrap or custom program in the User application area. If this format is followed, the new program or bootstrap can be loaded and started by the rom program.

Section 6.5 gives a detailed description of the UFD header block. The UFD area itself is described in the UFD spec.

Section 6.6 gives a detailed description of the Foreign language header block and text area.

6.1 GRAPHIC REPRESENTATION OF THE FIRST 105(10) BYTES OF EEPROM

The first 105 bytes (addresses 17765000 to 17765320) in the EEPROM contain information which is used by the boot rom code. This information can be updatted by using the SETUP mode program in the rom code. See the following section for a detailed explanation of each location and its meaning.

This section is a short cryptic look at the first 105 bytes.

OCTAL ADDRESS	BASE ADDRESS = 17765XXX, PCR LB = 0, BCSR 06 = 0, BCSR 05 = 1
000	0,OVRBF,FRC LCIE, DIS LKS, CLK SEL<1:0>, E HOB,0
002	PUP MODE <1:0>, RESTART MODE <1:0> HALT/TRAP, PMG CNT <2:0>
004	0, UBA CACHE ENA, ENA 18 BIT MODE, 0, 0, 0, 0, 0
006	ENA FOREIGN TEXT, VT PRES., 0,0,0,0, ALT BOOT, 0
010	DIS 73, DIS 65, 0, 0, DIS UBA ROM , ENA ECC TEST ENA UNIBUS MEM TST, DIS LONG MEMORY TST
012-020	spare
022	Byte 000 for test purposes only
024-046	Reserved for up to a 10 byte message to be transmitted to the terminal to enable it to a foreign language character set if needed.
050-072	Reserved for up to a 10 byte message to be transmitted to the terminal to enable it to standard English character set if needed.
074-246	Boot Translation table entries 1 to 9. 54 bytes
250-312	Switch boot table entries 1 to 6. 18 bytes
314	Byte 252 for test purposes only
316	For use by rom code only. This is an alternate checksum to increase the EEPROM's life.
320	8 bit checkword for this block. Checkword is the two's complement of the sum of the previous 104(10) bytes.
322-330	These four bytes reserved for customer's use only. Not to be used by any digital software.

6.1.1 FACTORY SETTINGS FOR EEPROM (INITIALIZED VALUE)

The following list the contents of the first 105(10) bytes of the EEPROM after it is initialized. All of the data is assessed by setting the low byte of the PCR to 0 and selecting the EEPROM in the BCSR. The 105 bytes are addressed from location 17765000 to 17765320.

Address	Data	Descriptions of settings
17765000	000	Standard clock selectins, HOB disabled
17765002	120	Power up/restart is Auto boot, PMG 0
17765004	100	UBA cache enabled, 22 bit mode selected
17765006	100	ANSI video terminal present
17765010	006	ECC and Unibus memory tests enabled
17765012	000	Reserved
17765014	000	Reserved
17765016	000	Reserved
17765020	000	Reserved
17765022	000	Must be 0 for test purposes
17765024	000	Foreign language terminal setup message blank
17765026	000	blank
17765030	000	"
17765032	000	"
17765034	000	"
17765036	000	"
17765040	000	"
17765042	000	"
17765044	000	"
17765046	000	"
17765050	000	English language terminal setup message
17765052	000	blank
17765054	000	"
17765056	000	"
17765060	000	"
17765062	000	"
17765064	000	"
17765066	000	"
17765070	000	"
17765072	000	"
17765074	000	Boot translation table blank
17765076	000	"
17765100	000	"
17765102	000	"
17765104	000	"

```
17765106   000        Boot translation table blank
17765110   000                   "
17765112   000                   "
17765114   000                   "
17765116   000                   "
17765120   000                   "
17765122   000                   "
17765124   000                   "
17765126   000                   "
17765130   000                   "
17765132   000                   "
17765134   000                   "
17765136   000                   "
17765140   000                   "
17765142   000                   "
17765144   000                   "
17765146   000                   "
17765150   000                   "
17765152   000                   "
17765154   000                   "
17765156   000                   "
17765160   000                   "
17765162   000                   "
17765164   000                   "
17765166   000                   "
17765170   000                   "
17765172   000                   "
17765174   000                   "
17765176   000                   "
17765200   000                   "
17765202   000                   "
17765204   000                   "
17765206   000                   "
17765210   000                   "
17765212   000                   "
17765214   000                   "
17765216   000                   "
17765220   000                   "
17765222   000                   "
17765224   000                   "
17765226   000                   "
17765230   000                   "
17765232   000                   "
```

```
17765234  000         Boot translation table blank
17765236  000               "
17765240  000               "
17765242  000               "
17765244  000               "
17765246  000               "
17765250  101         First auto boot device is A (MSCP
17765252  000           sniffer boot for units 0-7)
17765254  000               "
17765256  104         Second auto boot device is DL0.
17765260  114               "
17765262  000               "
17765264  115         Third auto boot device is MS0.
17765266  123               "
17765270  000               "
17765272  105         Terminate auto boot table here with E.
17765274  000               "
17765276  000               "
17765300  000         Fifth auto boot device is blank.
17765302  000               "
17765304  000               "
17765306  000         Sixth auto boot device is blank.
17765310  000               "
17765312  000               "
17765314  252         This byte must be 252 for test purposes.
17765316  000         Alternate checksum  * See note
17765320  312         Primary checksum    * See note
```

Note: The value of the data in the alternate and primary checksum will not always agree with what is shown above. As long as the sum of the first 105 bytes is 0 then they are ok.

6.2 DETAILED FORMAT FOR FIRST 105 BYTES OF PAGE 0

```
ADDRESS 17765000, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
                               ---------------------------------
                               | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
                               ---------------------------------
                                 |   |   |   |   |   |   |   |
0 - Spare      ------------------'   |   |   |   |   |   |   |
OV BB          ----------------------'   |   |   |   |   |   |
FRC LCIE       --------------------------'   |   |   |   |   |
DIS LKS        ------------------------------'   |   |   |   |
CLK SEL1       ----------------------------------'   |   |   |
CLK SEL0       --------------------------------------'   |   |
ENB HOB        ------------------------------------------'   |
0 - Unused     ----------------------------------------------'
```

Bit   Mnemonic   Meaning

07   0   Spare.

06   OVRBF   Override battery back up failure. Execute power up mode unconditionally, regardless of battery status.

05   FRC LCIE   The rom code will copy this bit to bit 13 in the BCSR. When this bit is set, assertion of the BEVENT line unconditionally requests interrupts.

04   LKS DIS   The rom code will copy this bit to bit 12 in the BCSR. When set, this bit disables response to the Line clock status register at 177546.

03   CLK SEL1
02   CLK SEL0   The rom code will copy bits 03-02 to bits 11-10 of the BCSR. These read-write bits select the source of the line clock:

CLK SEL1	CLK SEL0	Source
0	0	External BEVENT
0	1	On board 50 Hz
1	0	On board 60 Hz
1	1	On board 800 Hz

01 ENB HOB The rom code will copy this bit to bit 09 in the BCSR. When this bit is set, Halt on break from the console SLU is enabled.

00 0 Unused, must always 0.

```
ADDRESS 17765002, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
                                ---------------------------------
                                | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
                                ---------------------------------
                                  |   |   |   |   |   |   |   |
PUP MODE BIT1         ------------    |   |   |   |   |   |   |
PUP MODE BIT0         ----------------    |   |   |   |   |   |
RESTART MODE BIT1     --------------------    |   |   |   |   |
RESTART MODE BIT0     ------------------------    |   |   |   |
HALT/TRAP Option      ----------------------------    |   |   |
PMG CNT 02            --------------------------------    |   |
PMG CNT 01            ------------------------------------    |
PMG CNT 00            ----------------------------------------
```

Bits Meaning

07-06 POWER UP MODE
These two bits select the power up option.

BIT 07	BIT 06	CODE (See codes below.)
0	0	00
0	1	01
1	0	02
1	1	03

Code	Meaning
00	Run all selected tests depending on the contents of the EEPROM and then enter dialogue mode.
01	Run all selected tests depending on the contents of the EEPROM and then boot selected device in the EEPROM. This mode is auto boot mode (turnkey).

02 Run selected stand-alone tests and then halt, causing processor to enter ODT. Note that if the Trap on halt option is selected, this option must not be selected. The rom code setup mode will not allow the Halt on trap option to be selected if power up to ODT is selected at the same time.

03 Run selected stand-alone tests and then simulate a trap to 24/26. However, if BCSR bit 14 is set, restart system according bits 05-04 below.

05-04 RESTART MODE
These two bits select the restart mode. See power up modes for description.

03 The rom code will copy this bit to bit 03 of the maintenance register. When this bit is set, executing a Halt instruction in Kernal mode will cause a trap to location 4.

02 PMG CNT2 The rom code will copy bits 02-00
01 PMG CNT1 to bits 02-00 respectively in the
00 PMG CNT0 BCSR. These bits determine how long the CPU will wait before suppressing DMA requests.

```
ADDRESS 17765004, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
                              ---------------------------------
                              | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
                              ---------------------------------
                                |   |   |   |   |   |   |   |
0 - Spare           ------------    |   |   |   |   |   |   |
KMCR 06 CACHE ENA   ----------------    |   |   |   |   |   |
KMCR 05 ENA 18 BIT  --------------------    |   |   |   |   |
KMCR 04             ------------------------    |   |   |   |
KMCR 03             ----------------------------    |   |   |
KMCR 02             --------------------------------    |   |
KMCR 01             ------------------------------------    |
KMCR 00             ----------------------------------------
```

Address 17765004 contains the information needed to set up the memory configuration register (KMCR) on the KTJ11-B module.

```
ADDRESS 17765006, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
                              ---------------------------------
                              | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
                              ---------------------------------
                                |   |   |   |   |   |   |   |
ENA FOREIGN LANG.   ------------    |   |   |   |   |   |   |
VIDEO TEMINAL PRES. ----------------    |   |   |   |   |   |
0 - Spare           --------------------    |   |   |   |   |
0 - Spare           ------------------------    |   |   |   |
0 - Spare           ----------------------------    |   |   |
0 - Spare           --------------------------------    |   |
ALTERNATE BOOT BLK  ------------------------------------    |
0 - Spare           ----------------------------------------
```

ADDRESS 17765006, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)

Bits	Meaning
07	If bit 6 is set, Native language text from the EEPROM is used for all printouts from the rom code. If reset all printouts are in English text.

06 If bit 7 is set, an ANSI video terminal is present. This affects the DELETE character on input. When video is selected a DELETE erases the previous character on the screen. If this bit is reset slashes are used to show the deleted character. This bit also determines if an ANSI screen clear is executed at the beginning of the power up tests. If this bit is set and if force dialog is not selected the rom code will transmit an ANSI screen clear and cursor position before the first message during the power up rom tests.

05-02 Spare

01 If bit 1 is set, use alternate boot block to determine if a device's boot block is ok. If 0, a good boot must have 240-277 in PA 0 and 1 in byte PA 3. If 1, a good boot block need only have PA 0 not equal to 0.

00 Spare

ADDRESS 17765010, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)

```
                                ---------------------------------
                                | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
                                ---------------------------------
                                  |   |   |   |   |   |   |   |
DIS 73              ---------------   |   |   |   |   |   |   |
DIS 65              -------------------   |   |   |   |   |   |
0 - Spare           -----------------------   |   |   |   |   |
0 - Spare           ---------------------------   |   |   |   |
DIS UBA ROM         -------------------------------   |   |   |
ENA ECC TEST        -----------------------------------   |   |
ENA UBUS MEM TEST   ---------------------------------------   |
DIS LONG MEM TEST   -------------------------------------------
```

07 This bit is copied to BCSR bit 7 when the primary boot is successfully loaded and is to be started.

06 This bit is copied to BCSR bit 6 when the primary boot is successfully loaded and is to be started.

05-04 Spare

03 If bit 03 is set then disable the UBA rom's before starting a bootstrap program. If the boot to be started is on the UBA then ignore this bit.

02 If bit 02 is set then run ECC memory test on any memory that has a CSR with bit 4 being a read/write bit. This test is not run on Unibus memories. All parity memories have a CSR, but bit 4 is not R/W. The ECC test assumes that the MSV11-J hamming code is used.

01 If bit 01 is set run unibus memory tests. This bit is set only when unibus memory is present and is to be tested. When this bit is cleared the unibus memory is not tested.

00 If bit 00 is set, run main memory tests on all memory from 0-124 KW and check 2 locations every 4 KW for the rest of memory. If reset check all locations. This bit is normally cleared.

ADDRESS 17765012, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)

All 8 bits in this location are spare (0).

ADDRESS 17765014, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)

All 8 bits in this location are spare (0).

ADDRESS 17765016, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)

All 8 bits in this location are spare (0).

ADDRESS 17765020, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)

All 8 bits in this location are spare (0).

ADDRESS 17765022, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
This byte must contain a value of 000. If this bit is not 000 at power up the first 105(10) bytes of the EEPROM will automatically be initialized. See description for 17765314 for more information.

ADDRESSS 17765024 to 17765046
PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)

The following ten bytes of EEPROM are a optional message to be sent to the console terminal at power up or restart if Foreign language printouts are selected. The purpose of the message is to select the correct character set in the terminal for the messages if it needs to be set up. If the terminal does not need to be set up this message should have all zeroes in it. If the first byte is a zero the message will never be sent. The message is also sent when CTRL F is typed if foreign language is available.

ADDRESSS 17765050 to 17765072
PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)

The following ten bytes of EEPROM are a optional message to be sent to the console terminal at power up or restart when English language printouts are selected. The purpose of the message is to select the correct character set in the terminal for the messages if it needs to be set up. If the terminal does not need to be set up this message should have all zeroes in it. If the first byte is a zero the message will never be sent. The message is also sent when CTRL E is typed.

ADDRESSS 17765074 to 17765246
PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)

The following 54 bytes of EEPROM (17765074 to 17765246) are used as an optional translation table for boot devices. The maximum number of entries in the table is nine translations. Each translation entry requires 6 bytes of EEPROM. Two bytes are used to define the two letter mnemonic of the bootstrap program that is to be used in ASCII format. The next two bytes are unit numbers. Normally they will be the same value. The last two bytes are used to define the six octal digit address of the CSR of the device to be booted if it is not the default address used by the program. There are two purposes for this table. One is to allow booting of all MSCP devices on a system even when more than one controller interface is present. Each unit number which is not associated with the standard device address is logged into the translation table. The second purpose of the table is the allow muliple controllers of other devices to be booted even if the unit numbers overlap. An example would be as follows. If you had 8 RL02 drives, you would need 2 RLV12 controllers of which one cannot be at the standard address. You would have two duplicate sets of unit numbers 0-3. You would enter four entries into the translation table for the non standard address. The first would be DL4 to tranlate to DL0 with non standard CSR address 1XXXXX. This is where the second unit number is used. The others would be similar. MSCP does not require a second unit number because you are not suppose to duplicate any unit numbers.

ADDRESS's 17765074 to 17765106. Detail of first entry in translation table as follows:

ADDRESS 17765074, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
ASCII value of the first (leftmost) letter of a two letter mnemonic defining the boot program to be used.

ADDRESS 17765076, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
ASCII value of the second (rightmost) letter of a two letter mnemonic defining the boot program to be used.

ADDRESS 17765100, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
Octal value of the unit number of the device to be matched to the following unit number. From 0 to 377.

ADDRESS 17765102, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
Octal value of the unit number of the device to be booted. From 0 to 377.

ADDRESS 17765104, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
Low byte (bits 07-00) of the octal address of the base CSR for the device to be booted.

ADDRESS 17765106, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
High byte (bits 15-08) of the octal address of the base CSR for the device to be booted.

ADDRESS's 17765110 to 17765122	Second entry in table.
ADDRESS's 17765124 to 17765136	Third entry in table.
ADDRESS's 17765140 to 17765152	Fourth entry in table.
ADDRESS's 17765154 to 17765166	Fifth entry in table.
ADDRESS's 17765170 to 17765202	Sixth entry in table.
ADDRESS's 17765204 to 17765216	Seventh entry in table.
ADDRESS's 17765220 to 17765232	Eighth entry in table.
ADDRESS's 17765234 to 17765246	Ninth entry in table.

ADDRESSSS 17765250 to 17765312
PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)

The following 18 bytes of EEPROM (17765250 to 17765312) are used to allow three of the switches on the switch pack on the KDJ11-B module to be designated to boot one of six specific devices selected in this table. Each entry may select one device and unit number from 0 to 255(10). Each entry requires three bytes of EEPROM. The first two bytes are the ASCII value of the two letter mnemonic and the third byte is the unit number. The selections can be changed in setup mode to any valid boot device in the base rom's, the EEPROM or the UBA roms for unibus systems. If the EEPROM is initialized in setup mode then the table is initialized to a default set of devices. There is a set of default devices for both Qbus and Unibus systems.

ADDRESS's 17765250 to 17765254. Detail of first entry in the automatic boot sequence table. The entries in this table determine the order in which the automatic boot sequence is executed, the devices and unit numbers to try to boot. The table has a maximum number of six entries. If less than six entries is desired, a single letter mnemonic of E will terminate the sequence.

ADDRESS 17765250, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
ASCII value of the first (leftmost) letter of a two or one letter mnemonic defining the boot program to be used.

ADDRESS 17765252, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
ASCII value of the second (rightmost) letter of a two letter mnemonic. If mnemonic is a single letter this byte is 0.

ADDRESS 17765254, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
Octal value of the unit number of the device to be booted. From 0 to 377.

ADDRESS's 17765256 to 17765262 Second entry in table.

ADDRESS's 17765264 to 17765270 Third entry in table.

ADDRESS's 17765272 to 17765276 Fourth entry in table.

ADDRESS's 17765300 to 17765304 Fifth entry in table.

ADDRESS's 17765306 to 17765312 Sixth entry in table.

Default values of automatic boot table when EEPROM is initialize.

Qbus or Unibus

A	MSCP automatic boot (sniffer)
DL0	RL01/RL02
MS0	For Qbus TSV05 or TK25. For Unibus MS.
E	End of automatic boot sequence
blank	
blank	

ADDRESS 17765314, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
The contents of this location is 252 for test purposes.

ADDRESS 17765316, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
Address 17765316 contains an alternate checksum which is used only by the base rom to increase the life of the EEPROM.

ADDRESS 17765320, PCR LB = 0, BCSR BIT 05 = 1 (RS3 65)
Address 17765320 contains the eight bit checkword for locations 0-316. This byte is the two's complement of the sum of the previous 104(10) bytes.

6.3 GRAPHIC FORMAT OF BOOTSTRAP, UFD AND FOREIGN LANGUAGE AREAS

See the following sections for detailed description. Each of the three areas describe in the following pages has a five byte header block. Each header has a byte count to describe the size of the block. Bits 15:13 of the byte count specifically describe the type of header that is present. Each header has its own distinct code in bits 15:13 of the byte count. The format was set up this way to make it easy for the headers to be identified to type and to length of the data block with the header.

Note: Last location is page 37(8) for a 8 K byte EEPROM.
It is page 7 for a 2 K byte EEPROM.

```
PAGE  ADDRESS   First location in EEPROM
                 ------------------------------------
 00      000    | Base area - 105(10) bytes          |
                |   CPU and UBA Hardware parameters  |
                |   and other information            |
 00      320    |   Checksum for this block          |
                |------------------------------------
 00      322    | Customer use only. Not used by     |
 00      330    | digital.                           |
                |------------------------------------|
 00      332    | Header of bootstrap # 1 (5 bytes)  |
                |------------------------------------|
                |                                    |
                | Bootstrap # 1                      |
                | 2 K bytes maximum data             |
                |       (variable length)            |
                |                                    |
                |------------------------------------|
                | Expansion for bootstraps 2 & up    |
                |                                    |
                |              \/                    |
                |                                    |
                |                                    |
                |                                    |
                |                                    |
                |              /\                    |
                |                                    |
                | Expansion for UFD area             |
                |------------------------------------|
                |                                    |
                | UFD system description area        |
                | Describe system peripherals        |
                |       (variable length)            |
                |                                    |
                |------------------------------------|
                | UFD area header (5 bytes)          |
                |                                    |
                |------------------------------------|
                |                                    |
                | Optional language text             |
                |       (variable length)            |
                |                                    |
                |------------------------------------|
 37      766    |       (5 bytes)                    |
 37      776    | Optional Foreign language Header   |
Last location    ------------------------------------
```

6.4 USER APPLICATION AREA IN THE EEPROM

The EEPROM has an area set aside for use by customers or field service representatives. This area can be used for any purpose. Field service will normally use it to load in new bootstraps for the system not covered by the base rom code or by the UBA boot roms on UNIBUS systems. The customer could use this area for custom bootstraps or any other purpose. In the manufacturing process this area could be used to store special programs for manufacturing. The size of the area varies according to available space.

The first 105(10) locations of the EEPROM are always set aside for DIGITAL's use only. This corresponds to page 0 in the PCR. In a 2 K by 8 EEPROM there are 1,944(10) bytes locations available for use for bootstraps, UFD descriptions and foreign language text. A 8 K by 8 EEPROM has 8,086(10) byte locations available.

FORMAT FOR USER APPLICATION AREA IN THE EEPROM

The following shows the format that the data must follow to allow the 16 bit rom program to load and start a program stored in the EEPROM. The EEPROM is always addressed through address range 165XXX. Each bootstrap program always has a 5 byte header first. The data follows in a data block. There can only be one data block for each program. The maximum size of the data block is 2K bytes. The following page describes each byte.

GRAPHIC DESCRIPTION OF BOOT HEADER AND DATA BLOCK FORMAT

	000	First letter of mnemonic (BEGINNING OF HEADER)
	002	Second letter of mnemonic
	004	Byte count bits 7:0 of the data block
	006	Boot I.D. 15:11 = 10000 , Byte count 10:8
Header	010	Checksum for previous 4 bytes (END OF HEADER)
Data block		Highest allowable unit number for this device
		Short Device description. (0-11 bytes)
		This byte must be 0. Description terminator
		Sync byte = 1. Load address starts after sync
		Load address bits 7:0 (BEGIN DATA BLOCK)
		Load address bits 15:8
		Data begins here. (variable length < 2K bytes) DATA
		Start address bits 7:0
		Start address bits 15:8
End of data block		Checksum for data block

Note: The bootstrap areas always expands toward the end of the EEPROM.

DETAIL DESCRIPTION OF BOOT HEADER FORMAT

EEPROM
ADDRESS

000 7 bit ASCII value of first (left) letter of mnemonic in bits 6:0. Bit 7 is always 0.

002 7 bit ASCII value of second (right) letter of menomic in bits 6:0. If bit 7 is a 1 then a device descriptor is located at the beginning of the data block after the load address.

004 Byte count bits 7:0. The byte count is the exact number of bytes of data in the data block. It includes everything in the data block including the data block checksum.

006 Bits 7:3 are always 10000. Bits 7:5 are unique at 100 and describe this header as a boot header. Bits 2:0 are byte count bits 10-08. Maximum byte count is 2,048(10).

010 Eight bit checkword for the previous 4 bytes. This is the two's complement of the sum of the 4 bytes.

DETAILED DESCRIPTION OF BOOT DATA BLOCK FORMAT

EEPROM
ADDRESS

The first byte is the highest unit number to be allowed to boot. The range of this entry is 0 - 255(10). Example: If this byte was set to 3 the only unit numbers allowed would be 0,1,2 or 3. Any other unit number would generate an invalid unit number message. If in doubt set to 377(8) to allow all unit numbers from 0 to 255.

The following 0 to 12 bytes are an optional description of the device. The description will be printed out when the list command is executed. The description can be up to 11 bytes long and must be followed with a 0 byte. Example RL01/02 uses 7 bytes. (.ASCIZ "RL01/02").
Spaces may be used.

The loader problem will now look for a sync byte with a value of 1. Any byte other than 1 is ignored until the sync byte is found. The sync byte allows the programmer to generate a source file and then make it even for the next byte.

The next byte is load address bits 07-00. The next byte is Load address bits 15-08. These two bytes describe the 16 bit address where the first byte of data is to be loaded. The balance of the data is loaded sequentially going up towards the end of the EEPROM.

DATA

Data starts now. (variable length according to byte count)

Note: This is the only data actually loaded into memory.

End data

This byte is the start address bits 07:00. The next byte is start address bits 15:08. The start address is always even.

The last byte is the 8 bit checkword for all of the bytes in the data block.

The mnemonic is a two letter ASCII code which the 16 bit program uses to determine if the correct program to load has been found. Lower case ASCII codes are not allowed. Bit 07 must be 0. If the mnemonic was DL then byte 0 would contain 105(8) which is the ASCII code for D and byte 2 would contain 114(8) which is the ASCII code for L.

The load address is the location in memory where loading of the data starts.

The checkword is generated by summing all of the previous locations and then negating the value.

When two or more programs are stored in the EEPROM the second program must follow the first program without a gap between the end of the first program and the header of the second program. This is important because, this allows the 16 bit program to search for the customer's program until it finds the matching mnemonic.

Bootstrap programs loaded in the EEPROM should be written to use the contents of R1 as the base CSR address of the device when the program is entered. If R1 is 0 the the boot program should use a default address. If a unit number is needed it will be located in R0. By following this procedure the boot program will be able to handle booting devices with non standard CSR addresses. The CSR address is sourced from the translation table in page 0 of the EEPROM.

If the program is successfully loaded it will be started at its start address with the MMU off. The stack pointer will be set at address 172300 which points to the first location above the 16 supervisor PAR locations.

6.5 GRAPHIC DESCRIPTION OF THE UFD HEADER AND DATA BLOCKS

```
                   -----------------------------------------------
                  |                                               |
                  |                                               |
                  |                                               |
                  |      UFD DESCRIPTIONS - SEE UFD SPEC          |
                  |            (variable length)                  |
                  |                                               |
                  |                                               |
Header            |-----------------------------------------------|
block             | Checksum for all of previous UFD information  |
------------------|-----------------------------------------------|
Header -010       | UFD descriptions. See UFD specification       |
block             |-----------------------------------------------|
       -006       | UFD descriptions. See UFD specificatio.       |
                  |-----------------------------------------------|
       -004       | Byte count bits 7:0 (Number of UFD bytes + 1  |
                  |-----------------------------------------------|
       -002       | Byte count 15:13 = 001   Byte count bits 12:8 |
End of            |-----------------------------------------------|
data   -000       | Checksum for the previous 4 bytes of header   |
                   -----------------------------------------------
                        If foreign language present, it follows,
                        if not, this is the end of the EEPROM.
```

The UFD area is described in detail in the UFD specification.

The UFD area is always immediately before the foreign language text area in the EEPROM. If the foreign language text is not present then the UFD area is the last portion of the EEPROM. The UFD area always expands toward the beginning of the EEPROM.

DETAILED DESCRIPTIONS OF UFD HEADER BLOCK

The UFD header block is made up of five bytes of data. The header block descibes the length of the UFD area.

EEPROM
BYTE — UFD DATA IS ALWAYS PREVIOUS TO THE HEADER

-010 This byte is for any purpose needed by the UFD area.

-006 This byte is for any purpose needed by the UFD area.

-004 Byte count bits 7:0. The byte count is the total of all UFD description bytes plus the checksum. The header is not included in this byte count.

-002 Bits 7:5 of this byte are always 001. These bits describe this header as a UFD header. Bits 4:0 of this byte are bits 12:8 of the byte count.

-000 Checksum for the previous four bytes of the header.

6.6 GRAPHIC DESCRIPTION OF THE FOREIGN TEXT HEADER AND DATA BLOCK

A detailed description of the foreign language text and dictionary follows this grahphic representation.

```
                 ------------------------------------------------
                 |                                              |
                 |               Foreign text                   |
                 |               (variable length)              |
                 |                                              |
                 |----------------------------------------------|
                 |                                              |
                 |        Foreign Dictionary (Optional)         |
                 |               (variable length)              |
                 |                                              |
Header           |----------------------------------------------|
block            | Checksum for all of previous foreign text    |
-----------------|----------------------------------------------|
Header -010      | Dictionary byte count bits 7:0               |
block            |----------------------------------------------|
       -006      | Bits 15:13 = 000      Dictionary B.C. 12:8   |
                 |----------------------------------------------|
       -004      | Byte count bits 7:0 (Number of text bytes + 1 |
                 |----------------------------------------------|
       -002      | Bits 15:13 = 010/011   Byte count bits 12:8  |
End of           |----------------------------------------------|
data   -000      | Checksum for the previous 4 bytes of header  |
                 ------------------------------------------------
                        This is the end of the EEPROM
```

DETAILED DESCRIPTION OF THE FOREIGN LANGUAGE HEADER

EEPROM
 BYTE   THE DICTIONARY AND TEXT IS ALWAYS PREVIOUS TO THE HEADER

-010   Byte count bits 7:0 of the dictionary byte count.
              The byte count is the total of all dictionary bytes.
              The dictionary is optional.

-006   Bits 7:5 are always 000. Bits 4:0 are byte count bits
              12:8 of the dictionary byte count.

-004 Byte count bits 7:0. The byte count is the total of all foreign text and dictionary bytes and their checksum byte.

-002 Bits 7:5 of this byte are always 010 or 011. These bits describe this header as a foreign language header. If bits 7:5 = 011 then the foreign language uses 7 bit input (bit 7 is always stripped to zero). If bits 7:5 = 010 then input is true 8 bit and no bits are stripped. Bits 4:0 of this byte are bits 12:8 of the byte count.

-000 Checksum for the previous four bytes of the header.

DETAILED DESCRIPTION OF THE FOREIGN LANGUAGE DATA BLOCK

The foreign language data block consists of two general areas. The first area is the text for all the messages that have to be translated from English. The second area is an optional dictionary to help reduce the total amount of space needed for foreign text. The last byte of the area is a checksum for all of the previous foreign language text and dictionary words.

DETAILED DESCRIPTION OF THE FOREIGN LANGUAGE TEXT AREA

All messages in the base rom code are divided into separate messages with a message number for each. All of the messages are stored in order. The message numbers range from 1 to 377(8). For foreign language use each message must be translated. The length of any message must not exceed 256(10) bytes. There will not be any English messages greater than 128(10) bytes. Messages 1-17 have special meaning. Message 1 is the name of the foreign language being translated (i.e. Spanish). Messages 2-17 are translations of the English commands used in dialog mode. The current commands are ?,Help,Boot,List,Setup,Map,Test and A. These are messages 2-11. Messages 12-17 are not used and are reserved for possible future expansion. You will note that the commands are all unique to the first letter. Only the first letter of any command has to be typed to execute that command. If the two or more of the commands translate into words with the same first or more letters then enough of the commands letters will have to typed in to make it unique. If this is not done then the boot program will match the input with the first command that matches it. If desired the commands could be translated to 1,2,3 etc.

7.0 ERROR INFORMATION PASSED TO LOADABLE PROGRAM

A limited amount of error information is passed to the operating system or the higher level diagnostics when they are loaded. The primary purpose of this information is to tell the operating system if any non fatal test failures were bypassed. The error information is passed to the operating system through User Data PDR 6 at address 17777634 as shown below. The operating system or the diagnostic should check location 17765010(8) in the EEPROM to see if any particular tests were bypassed.

Another location keeps track of the total number of errors that occurred during testing. This location is User Data PAR 4 at address 17777670. This location is incremented everytime an error occurs.

In order to know if the previous locations are valid a third location contains the 16 bit checksum of the previous two locations. This location is Kernel I PAR 4 at address 17772354.

```
                             14  12  10   8   6   4   2   0
                              |   |   |   |   |   |   |   |
                           15 |13 |11 | 9 | 7 | 5 | 3 | 1 |
                            | | | | | | | | | | | | | | | |
                           _________________________________
                           | | | | | | | | | | | | | | | | |
                           ---------------------------------
                            | | | | | | | | | | | | | | | |
    15      Cache error     - | | | | | | | | | | | | | | |
    14      Memory error    --- | | | | | | | | | | | | | |
    13      UBA cache err   ----- | | | | | | | | | | | | |
    12      SLU error       ------- | | | | | | | | | | | |
    11      ECC/Parity err  --------- | | | | | | | | | | |
    10      Unibus memory e ----------- | | | | | | | | | |
    09      FP error        ------------- | | | | | | | | |
    08      ROM/clock error --------------- | | | | | | | |
                                            | | | | | | | |
    07      n/a             ----------------- | | | | | | |
    06      n/a             ------------------- | | | | | |
    05      n/a             --------------------- | | | | |
    04      n/a             ----------------------- | | | |
    03      EEPROM error    ------------------------- | | |
    02      CIS error       --------------------------- | |
    01      Any error       ----------------------------- |
    00      n/a             -------------------------------
```

Bit	Mnemonic	Meaning
15	Cache error	If this bit is set it indicates that an error occurred in the cache tests.
14	Memory error	If this bit is set it indicates that an error occurred during the main memory tests.
13	UBA cache err	If this bit is set it indicates that an error occurred during the UBA cache diagnostics.

12	SLU error	If this bit is set it indicates that an error occurred in the SLU tests.
11	ECC/Parity err	If this bit is set it indicates that an error occurred in the ECC or the parity logic tests.
10	Unibus memory e	If this bit is set it indicates that an error occurred during the Unibus memory tests.
09	FP ERROR	If this bit is set it indicates that an error occurred in the Floating Point tests.
08	ROM/Clock error	If this bit is set it indicates that an error occurred in either the ROM logic or clock tests.
07	not used	This bit is not read/write
06	not used	This bit is not read/write
05	not used	This bit is not read/write
04	not used	This bit is not read/write
03	EEPROM error	If this bit is set it indicates that a checksum error occurred during the EEPROM test of the first 105(10) bytes and the error was not corrected.
02	CIS error	If this bit is set it indicates that an error occurred in the CIS tests.
01	Any error	If this bit is set it indicates that an error occurred in any of the tests. This bit can be set without setting any of the above bits.
00	not used	This bit is not read/write

7.1 INTERPRETATION OF EDGE MOUNTED SWITCHES ON KDJ11-B

An eight bit switch pack is mounted at the handle end of the KDJ11-B. Switches 6-8 select the baud rate for the DLART chip. Switches 1-8 correspond to bits 07-00 in the read only configuration register at address 17777524.

> Note: In the following discussion it is assumed that when a switch is off the line connected to it is pulled up high by the CPU module. If an external device is grounding a switch line then the switch is considered on. When using an external device to control the selections, any switch on the CPU that goes to an external switch should be off to pass control to the external device.

Normally switches 1-5 are off and EEPROM contents determine what action is to be taken at power up or restart.

Switch 5 when off, unconditionally forces the rom code to enter Dialog mode at the completion of the selected tests. Auto boot cannot occur if switch 5 is off.

If switch 5 is on and switch 1 is on then the console is disabled and all output to the console is suppressed. Dialog mode cannot be entered. If any input occurs at the console the program will transmit an error message to the console saying the console is disabled.

If switch 5 is on and if switches 2-4 are not equal to octal 0 or 7 the Auto boot mode is selected with the device and unit number to be boot determined by a table in the EEPROM and the value in switches 2-4 (1-6). There are default values in the table for both Qbus and Unibus systems. If these default devices do not meet the needs of the customer they can be changed in setup mode to any value. The selections can be standard rom boots, UBA rom boots or any EEPROM boots.

1 = OFF 0 = ON X = Don't care

1	2	3	4	5	6	7	8	SWITCH NUMBER	
7	6	5	4	3	2	1	0	DATA BIT NUMBER	
								OCTAL	ACTION TAKEN AT POWER UP
X	X	X	X	1	X	X	X		Console enabled. Unconditionally transfer control to dialog after running tests. (FORCE DIALOG)

For the following the console is enabled and 6-8 select the baud rate. Auto boot mode is automatically selected for selections SB 1 TO SB 6.

1	1	1	1	0	X	X	X	36X	Dispatch according to EEPROM
1	1	1	0	0	X	X	X	34X	SB 6 in EEPROM determines boot
1	1	0	1	0	X	X	X	32X	SB 5 in EEPROM determines boot
1	1	0	0	0	X	X	X	30X	SB 4 in EEPROM determines boot
1	0	1	1	0	X	X	X	26X	SB 3 in EEPROM determines boot
1	0	1	0	0	X	X	X	24X	SB 2 in EEPROM determines boot
1	0	0	1	0	X	X	X	22X	SB 1 in EEPROM determines boot
1	0	0	0	0	X	X	X	20X	Power up to ODT immediately.

For the following the console is disabled and 6-8 are not used. Auto boot mode is automatically selected for all selections.

1 = OFF 0 = ON X = Don't care

1	2	3	4	5	6	7	8	SWITCH NUMBER	
7	6	5	4	3	2	1	0	DATA BIT NUMBER	
								OCTAL	ACTION TAKEN AT POWER UP
0	1	1	1	0	X	X	X	16X	Dispatch according to EEPROM
0	1	1	0	0	X	X	X	14X	SB 6 in EEPROM determines boot
0	1	0	1	0	X	X	X	12X	SB 5 in EEPROM determines boot
0	1	0	0	0	X	X	X	10X	SB 4 in EEPROM determines boot
0	0	1	1	0	X	X	X	06X	SB 3 in EEPROM determines boot
0	0	1	0	0	X	X	X	04X	SB 2 in EEPROM determines boot
0	0	0	1	0	X	X	X	02X	SB 1 in EEPROM determines boot
0	0	0	0	0	X	X	X	00X	Run stand alone mode tests in a loop at power up.

The following lists the default values for SB 1 to SB 6 if the EEPROM is initialized in setup mode. SB 4 to SB6 are left blank and are not used. The user may change the values to any they desire by using setup mode.

		System type		
	Qbus		Unibus	
SB 1	A	MSCP Sniffer	A	MSCP Sniffer
SB 2	DL0	RL01/02	DL0	RL01/02
SB 3	MS0	TSV05/TK25	MS0	TS11/TU80
SB 4	E	End Auto boot	E	End Auto boot
SB 5	blank		blank	
SB 6	blank		blank	

Baud rate table.

1	2	3	4	5	6	7	8	<- SWITCH NUMBER
								Baud rate selected
X	X	X	X	X	0	0	0	38400 baud
X	X	X	X	X	0	0	1	19200 baud
X	X	X	X	X	0	1	0	9600 baud
X	X	X	X	X	0	1	1	4800 baud
X	X	X	X	X	1	0	0	2400 baud
X	X	X	X	X	1	0	1	1200 baud
X	X	X	X	X	1	1	0	600 baud
X	X	X	X	X	1	1	1	300 baud

1 = OFF   0 = ON   X = Don't care