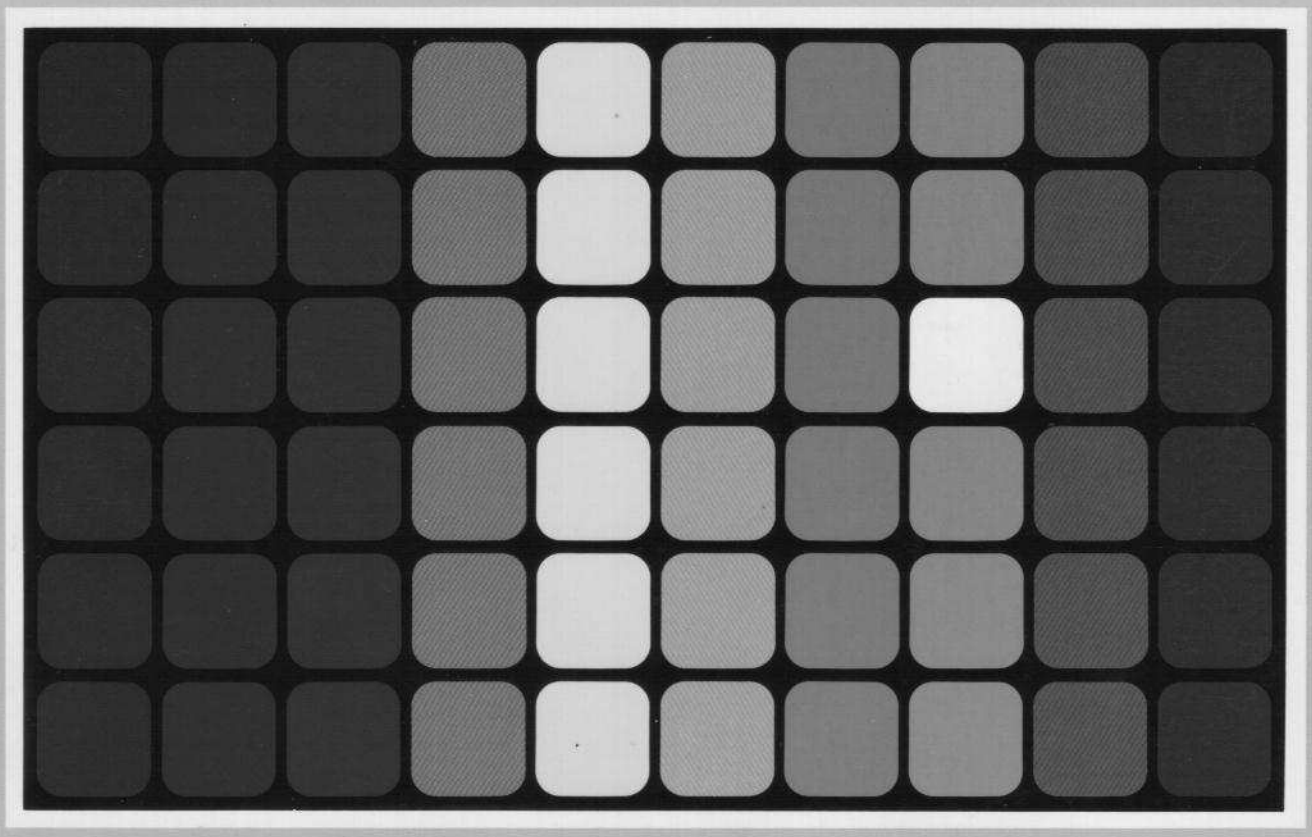
EK-KC750-TM-002

KC750 Microdiagnostics and Technical Manual



digital

EK-KC750-TM-002

KC750 Microdiagnostics and Technical Manual

Prepared by Educational Services
of
Digital Equipment Corporation

1st Edition, October 1980
2nd Edition, August 1981

Copyright © 1980, 1981 by Digital Equipment Corporation
All Rights Reserved

The reproduction of this material, in part or whole, is strictly prohibited. For copy information, contact the Educational Services Department, Digital Equipment Corporation, Maynard, Massachusetts 01754.

The information in this document is subject to change without notice. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

Printed in U.S.A.

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts.

DEC	DECnet	OMNIBUS
DECUS	DECsystem-10	OS/8
DIGITAL	DECSYSTEM-20	PDT
Digital Logo	DECwriter	RSTS
PDP	DIBOL	RSX
UNIBUS	EduSystem	VMS
VAX	IAS	VT
	MASSBUS	

10/84-15

CONTENTS

TABLES

PREFACE

INTRODUCTION

This manual explains the application, installation, and function of the KC750 option. This option allows remote diagnosis of the VAX-11/750 computer system and provides it with a self-microdiagnostic capability.

A remote diagnostic module (RDM) is the primary component of the KC750 option. The RDM is a stored-program microcomputer residing on a circuit board that installs in the VAX-11/750 CPU backplane. This manual shows DIGITAL Field Service engineers how to install the KC750 option and use it to troubleshoot a VAX-11/750 system. The text includes an overview of how the RDM works, designed to aid Field Service engineers in using the KC750 option.

Chapter 1 provides an overview of the KC750 option and how it fits into DIGITAL's general diagnostic service.

Chapter 2 explains how to use the KC750 option. This includes customer responses to system problems, VAX console panel indicators, RDM operating states, and diagnostic commands.

Chapter 3 covers installation of the KC750 option. This includes connecting the RDM and modem, and verifying the installation.

Chapter 4 provides a functional description of the RDM. This helps Field Service engineers working with the RDM.

Chapters 5 through 8 describe the RDM software used to run microdiagnostics on the VAX-11/750.

CONVENTIONS

The following syntax and dialogue conventions apply to this manual.

Command Syntax Conventions

The syntax used in this statement is explained in the text that follows.

```
COM <argument-1> <argument-2> [optional] [LITERAL]
{<argument-3>/<argument-4>}
```

COM	The only characters of a command name you may type to enter the command.
<argument>	Angle brackets indicate information you must enter with the command (such as an address). Each argument is given a generic name spelled in lowercase letters, and surrounded by angle brackets. Do not type the brackets; they only specify arguments. An argument must be typed unless it is surrounded by square brackets, in which case it is optional.

{<a-3>/<a-4>}	Braces indicate that you must select one argument from a list of two or more arguments. The braces enclose the list and a slash separates the choices. Do not type the braces or slashes; they clarify the choices.
[optional]	Square brackets indicate an optional argument in the command line. Do not type the brackets; they only designate the options.
[LITERAL]	Literal arguments appear in all uppercase letters; if entered, the argument must be typed exactly as shown.

Dialogue Conventions

This manual uses sample dialogues to demonstrate commands being described. In these dialogues, the following symbols represent terminal keys that perform special functions.

Symbol	Key(s)	Description
CTRL/C	CTRL and C	These keys cancel current program execution.
CTRL/U	CTRL and U	These keys delete all characters typed on current line. The system prints carriage return/line feed sequence; operator may retype the entire line.
CTRL/O	CTRL and O	These keys suppress output to terminal.
CTRL/S	CTRL and S	These keys suspend output to terminal until operator types CTRL/Q.
CTRL/Q	CTRL and Q	These keys resume output interrupted by CTRL/S.
<DEL>	DELETE	This key deletes last character typed.
<RET>	RETURN	This key serves as normal delimiter for all input to system.
CTRL/P	CTRL and P	These keys change system state from program I/O to VAX console.
CTRL/D	CTRL and D	These keys change system state from VAX console to RDM console.
CTRL/R	CTRL and R	These keys display current command string.
↑		This symbol indicates display of a control character on the terminal screen.

NOTE

Type control characters by pressing the CTRL key in combination with another character. The combined code has special meaning to the system.

Boldface

In sample dialogues, text in **boldface** indicates data the operator types. Text not in **boldface** indicates data the computer types.

CHAPTER 1
INTRODUCTION

1.1 SCOPE

This chapter provides an overview of the KC750 option. It refers to applicable documents, defines the different option versions, and lists the required specifications for modems used with the option.

1.2 OPTION OVERVIEW

A fully configured KC750 option consists of a remote diagnostic module (RDM) that fits into the VAX-11/750 CPU backplane, a modem that allows the RDM to communicate with a remote operator, and cabling that connects the RDM and modem. Figure 1-1 shows how the KC750 option components relate to the VAX-11/750.

The option provides the VAX-11/750 with two diagnostic tools: remote diagnosis and microdiagnosis.

Remote diagnosis lets you connect the VAX-11/750 directly to DIGITAL's Remote Diagnostic Center (DRDC) by turning a keyswitch on the VAX console panel. This DRDC connection allows a remote engineer to log on the system and control the console the same as a local user. DRDC can then monitor system execution of user programs and run diagnostic programs on the CPU and peripherals.

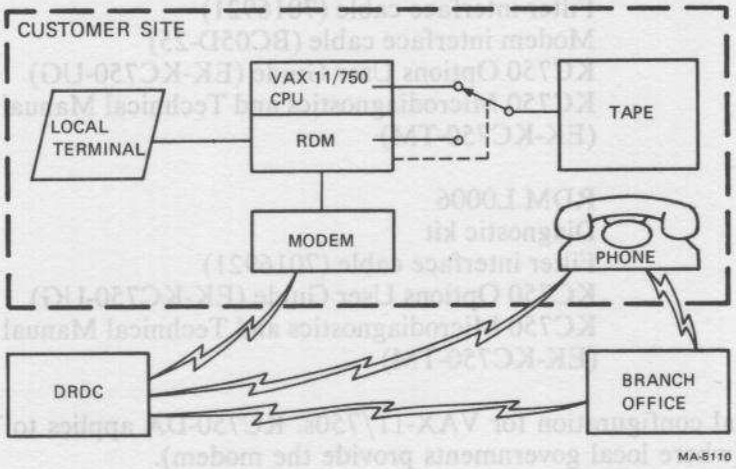


Figure 1-1 VAX-11/750 Remote Diagnosis Block Diagram

A major benefit of the KC750 option is the RDM's ability to test the VAX CPU at the microlevel. This includes the ability to control execution of microdiagnostic programs on the CPU, even if the only part of the CPU that works is the clock.

Microdiagnostic programs are stored on cassette tape at the customer site. To test the CPU, the RDM reads these programs from the tape and stores them in its own memory. Under RDM control, the microdiagnostics then drive the CPU. During diagnosis, the RDM relays responses from the CPU to the console terminal, the DRDC, or both.

1.3 APPLICABLE DOCUMENTS

The following documents apply to the operation and installation of the KC750 option.

- *KC750 Options User Guide* – Document No. EK-KC750-UG
- Electronic Industries Association (EIA) – Document No. RS-232
- System Configuration Worksheets – This package is available in the *Site Management Guide*. It contains bus maps, interface layouts, and appliques for laying out system configurations.
- Modem instruction manuals – These are supplied by the modem vendor and shipped with individual units.

1.4 OPTION VARIATIONS

There are two configurations of the KC750 option.

Option	Parts (one each)
KC750-CA	RDM L0006 Standalone modem Diagnostic kit Filter interface cable (7016921) Modem interface cable (BC05D-25) KC750 Options User Guide (EK-KC750-UG) KC750 Microdiagnostics and Technical Manual (EK-KC750-TM)
KC750-DA	RDM L0006 Diagnostic kit Filter interface cable (7016921) KC750 Options User Guide (EK-KC750-UG) KC750 Microdiagnostics and Technical Manual (EK-KC750-TM)

KC750-CA is the usual configuration for VAX-11/750s. KC750-DA applies to VAX-11/750 installations outside the US (where local governments provide the modem).

1.5 MODEM SPECIFICATIONS

Modems used with KC750 options must conform to certain specifications and function states. They must be built in accordance with CCITT Standard V24 and EIA Standard RS-232C. The modem type is low speed asynchronous with a transfer rate of 300 baud. Appendix E gives detailed modem specifications.

CHAPTER 2
OPERATING PROCEDURES

2.1 SCOPE

This chapter explains how to use the KC750 option to diagnose the VAX-11/750. The chapter contains two main sections corresponding to the two diagnostic approaches used by the RDM. Paragraph 2.3 explains how to use the KC750 option to perform remote diagnosis. Paragraph 2.4 explains how to use the RDM command set to perform microdiagnosis.

Remote diagnosis is usually the first procedure used to test the system, since KC750 option customers make their initial request for service through the DIGITAL Remote Diagnosis Center (DRDC). If the CPU is working, the DRDC logs on the customer system and tests the CPU and peripherals with diagnostics. If the CPU is not working, the DRDC may use the RDM command set to test the CPU. Should remote diagnosis fail, an on-site technician may need to use the RDM (along with other tools) to test the system.

2.2 KEYSWITCH AND FRONT PANEL INDICATORS

Using the RDM requires a knowledge of the VAX-11/750 keyswitch and front panel indicators. The keyswitch determines various operating states; the front panel indicators show what the RDM may do at a given moment. Figure 2-1 shows the front panel console with keyswitch and indicators. Table 2-1 describes the various keyswitch positions. Table 2-2 describes the front panel indicators that apply to the KC750 option.

2.3 REMOTE DIAGNOSIS

To perform remote diagnosis on a VAX-11/750 equipped with the KC750 option, phone the DIGITAL Remote Diagnosis Center (DRDC) at one of the following numbers.

800-525-6570	Outside Colorado (US only)
800-332-7189	Inside Colorado

Outside the US, ask the local DIGITAL branch office for the procedure to contact the DRDC.

When calling, be prepared to give the following information.

- Customer name (company)
- Caller's name and phone number .
- Address and location of system
- System type and serial number
- Description of problem
- Device and device unit number on which diagnostic media will be mounted

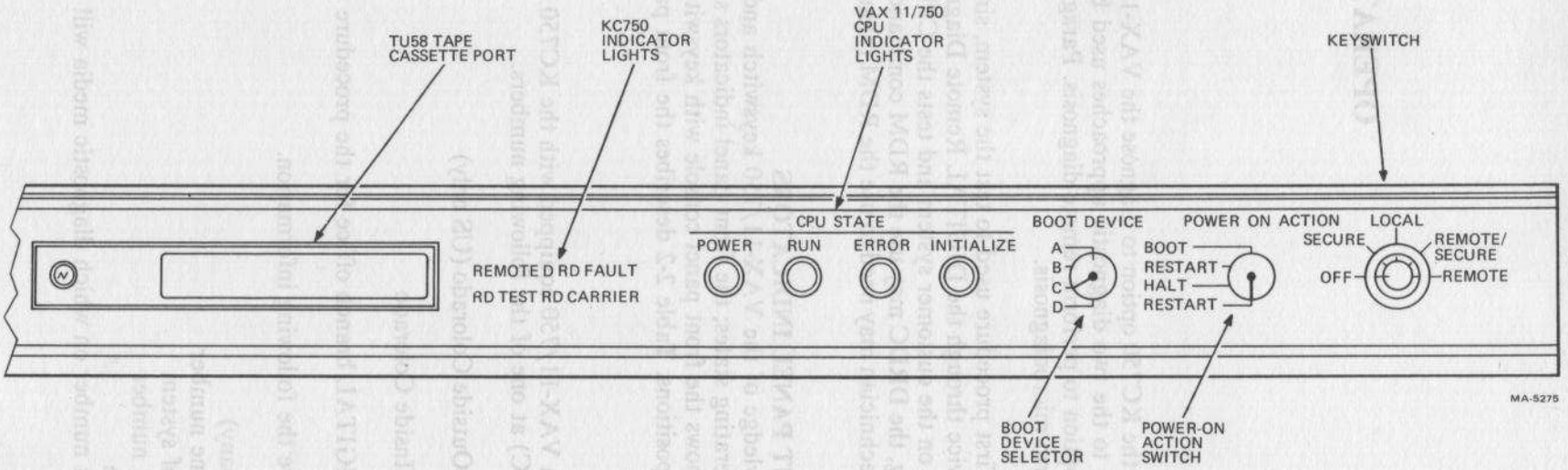


Figure 2-1 VAX-11/750 Front Panel

Table 2-1 VAX-11/750 Keyswitch Positions

Position	Description
LOCAL SECURE	System only responds to local terminal. Program I/O state is enforced. This is SECURE position on keyswitch.
LOCAL	System only responds to local terminal. System responds to CTRL/D and CTRL/P to change states.
REMOTE SECURE	System only responds to remote terminal. Program I/O state is enforced, except that remote terminal can make system enter talk state.
REMOTE	System only responds to remote terminal. System responds to CTRL/D and CTRL/P to change states.

Table 2-2 KC750 Option Front Panel Indicators

Indicator	Description
REMOTE D	This indicates that the keyswitch is in REMOTE SECURE position.
REMOTE	This indicates that the keyswitch is in REMOTE position.
RD FAULT	This indicates RDM logic failure. Fault indicator should come on for about ten seconds during console power-up as part of logic self-test.
RD TEST	This indicates that the DRDC host computer is performing diagnostic tests.
RD CARRIER	This indicates that the carrier signal is detected from DRDC.

If the DRDC decides to run a remote diagnostic session, it requests that you remove all disks (except the system/diagnostic disk) and tapes from the system. Then you must follow this procedure.

1. Mount the diagnostic media and a scratch disk.
2. Turn the POWER ON ACTION switch to HALT.
3. Turn the keyswitch to REMOTE; this transfers system control to the DRDC.

2.4 MICRODIAGNOSIS

There are two ways to use the RDM to diagnose the VAX-11/750 at the microlevel. First, you may use RDM commands to perform direct actions, such as examining CPU memory locations or stepping through a particular microinstruction. Second, you may use RDM commands to load microdiagnostic programs into the RDM, and have these programs drive the VAX CPU control lines.

Many factors determine which method to use. Usually Field Service technicians run a microdiagnostic first. If that fails to run or produces unsatisfactory information, they resort to RDM commands. The following paragraphs explain how to access and use RDM commands, including those that load and run microdiagnostics. Chapters 5 through 8 describe in detail how to use microdiagnostics to test the VAX-11/750.

2.4.1 Entering Diagnostic Commands

The VAX-11/750 must be powered up and in the proper operating state to enter RDM diagnostic commands. Power is applied to the system when you turn the keyswitch from OFF to another position. The VAX-11/750 powers up in either the VAX console state or the program I/O state, depending on the position of the front panel switches. To test the VAX-11/750 from the local terminal, turn the keyswitch to LOCAL. On power-up, the RDM runs an initialization routine that tests RDM logic. Table 2-3 describes RDM power-up self-test activities.

Initialization causes the fault indicator to turn on and then off. A fault exists in the RDM if the fault indicator does not turn on, or turns on but not off. The RDM may continue to function despite a fault; however, you should not use the RDM when a fault is suspected.

The current system operating state determines which VAX and RDM commands the system recognizes. Table 2-4 describes the operating states available. Figure 2-2 shows how to get from one state to another. RDM commands that test the VAX-11/750 are recognized only if entered while the system is in the command mode of the RDM console state.

To enter diagnostic commands at the local terminal, turn the keyswitch to LOCAL. If the CPU is running (under program control), type CTRL/P followed immediately by CTRL/D . This takes the system from the program I/O state to the command mode of the RDM console state. If the CPU is halted (program execution), the system is in the VAX console state; type a CTRL/D to invoke the command mode of the RDM console state. Next, type the desired diagnostic command.

NOTE

If you use an LA120 terminal with the RDM, set the terminal's auto-disconnect feature to OFF. Otherwise, CTRL/D will not invoke the command mode of the RDM console state.

Table 2-3 RDM Power-Up Activities

Activity	Description
ROM Test	This test performs a checksum calculation on the set of four RDM ROM chips. It compares results to a hard-coded character located on one of the ROM chips. Any mismatch between the calculated and hard-coded character sends an error message (ERR – ROM) to the active terminal and the fault indicator remains on.
RAM Test	This test writes, reads back, and verifies a set of bit patterns for each memory location. Any mismatch sends an error message (ERR – RAM) to the active terminal and the fault indicator remains on.

Table 2-4 System Operating States

State	Description
Program I/O	This state is the normal operating state of the VAX-11/750. CPU is under macro program control.
VAX Console	This state places CPU under the control of its own console microcode. It supports CPU console commands. Active terminal displays console prompt (>>>).
RDM Console Control Mode	This state is the same as program I/O state if CPU is running, except that CTRL/D changes system to RDM console state command mode; running program supplies terminal prompt. If CPU is halted, this state is the same as VAX console state where VAX console prompt (>>>) is disabled.
RDM Console Command Mode	Under this state, the system recognizes RDM commands only. (See Paragraph 2.4.2 for a list of available commands.) RDM prompt (RDM>) is displayed. Character output from the CPU to the terminal is disabled.
Talk	This state allows direct communication between an operator at the local terminal and an operator at the DRDC. (See Paragraph 2.4.2.26 for further discussion of the talk state.)
Micro-diagnostic	This state places RDM under control of the microdiagnostic monitor (MICMON). Microdiagnostic prompt (MIC>) is displayed.

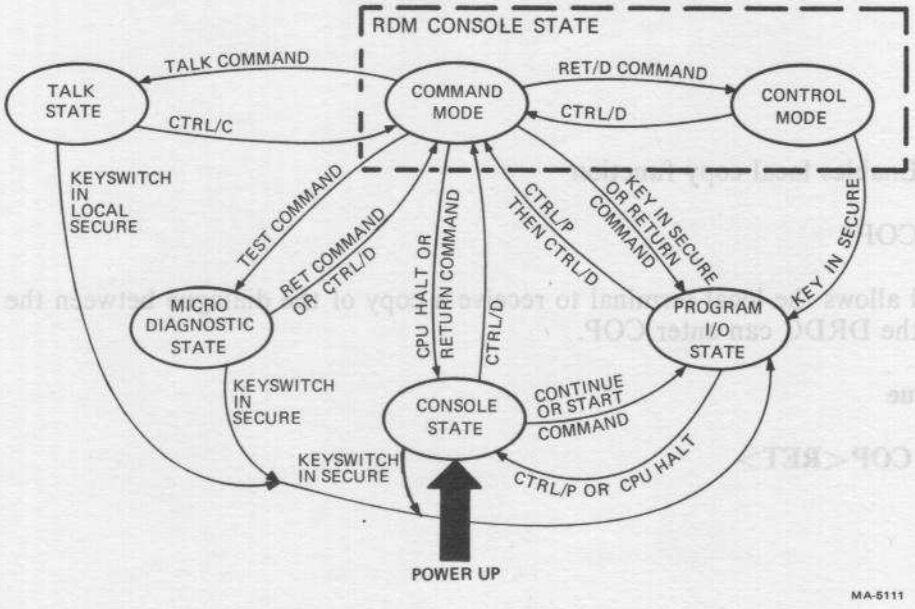


Figure 2-2 RDM Operating State Transitions

Example

The keyswitch is in LOCAL position, and CPU status is halted (program not running, but CPU clock working). The operator wants to execute a trace of CPU control store addresses. The system leaves console state and enters RDM console state; then the operator gives diagnostic commands to stop the CPU clock and do a trace.

>>> **CTRL/D** (>>> is console state prompt.)

RDM> **STO<RET>** (RDM> is RDM console state prompt.)

RDM> **TR<RET>**

2.4.2 Preprogrammed Commands

Table 2-5 lists the RDM preprogrammed commands. The following paragraphs describe the commands in alphabetical order.

2.4.2.1 Clear

Purpose: Clears stop-on-micromatch function

Syntax: CL

This command disables the stop-on-micromatch function (enabled by Set command), but does not change the contents of the match register. This allows the operator to sync test equipment on the match without stopping the micromachine. Refer to Table A-3 Appendix A for backplane test points.

Sample Dialogue

RDM>**CL<RET>**

RDM>

2.4.2.2 Copy

Purpose: Enables local copy function

Syntax: COP

This command allows the local terminal to receive a copy of the dialogue between the RDM and the DRDC. Only the DRDC can enter COP.

Sample Dialogue

RDM> **COP<RET>**

RDM>

Table 2-5 RDM Command Set

Command	Syntax	Function
Clear	CL	Clear stop-on-micromatch
Copy	COP	Copy to local terminal
Copy-Disable	COP/D	Disable local copy function
Deposit	D [/modifier][address]<data>	Deposit to VAX memory location
Examine	E [/modifier][address]	Examine VAX memory location
Examine-Console	E/C [address]	Examine RDM status registers
Initialize	INI	Initialize
Link	LIN	Enter link control file
Load	LO <file name.ext>[address]	Load TU58 file to VAX memory
Local	CTL	Enable local copy control
Local-Disable	CTL/D	Disable local copy
Microaddress	UA <address>	Load CS address bus
Microaddress/C	UA/C <address>	Load CS address bus until next M clock only
Parity	PAR <address>	Run control store parity check
Perform	PER	Perform link control files
Repeat-Last-Command	REP	Repeat console command
Repeat-Next-Command	R <command>	Repeat following command
Return	RET	Return to program I/O state
Return/D	RET/D	Return to RDM control mode
Set	SE [address]	Set stop-on-micromatch
Show	SH	Show CPU state
Show-Version	SH/V	Show current version of RDM firmware
Step	STE	Step through single microinstruction
Step-Tick	STE/T	Step through single clock tick
Stop	STO	Stop clock
Talk	TA	Enter talk state
Test	TE	Load and run microdiagnostics
Test/C	TE/C	Load microdiagnostic and await command
Test-File	TE <file name.ext>	Load and run user RDM program
Trace	TR	Display trace of CS address

2.4.2.3 Copy Disable

Purpose: Disables local copy function

Syntax: COP/D

This command stops the local terminal from printing DRDC dialogue.

Sample Dialogue

```
RDM> COP/D<RET>

RDM>
```

2.4.2.4 Deposit

Purpose: Deposits data to a VAX memory location

Syntax: D [/modifier] <address> <data>

This command writes hexadecimal data to a hexadecimal VAX memory address specified in the command string. Modifiers (/W for word, /L for long word, and /B for byte) define the data type to be deposited. If a modifier is used, it becomes the default data type; if not, then the current default data type is assumed.

If a /N modifier is used, the address field does not appear in the command. The last location where data was deposited is incremented by the default data type and used as the default address.

The address field may be replaced by an asterisk (*) or a plus sign (+). An asterisk causes the RDM to use the address referenced in the last Examine or Deposit command. A plus sign (+) increments the address by the default data type.

The same data may be deposited to blocks of successive addresses by using the Repeat command syntax (for example, R D + <data>).

Sample Dialogue 1

```
RDM>D/L 0 11223344<RET>          Long word 11223344 deposited to address zero

RDM>
```

Sample Dialogue 2

```
RDM>D/W 2 6677<RET>              Word 6677 deposited to address 2
RDM>E/L 0 <RET>                  Long word examined at location 0
    P 000000 66773344            P indicates physical address
RDM>
```

Sample Dialogue 3

```
RDM>D/B 3 88<RET>            Byte 88 deposited to address 3
RDM>E/L 0<RET>               Long word examined at location 0
    P 000000 88773344
RDM>
```

Sample Dialogue 4

RDM>**D 0 10101010<RET>**	Long word 10101010 deposited to address 0
RDM>**E *<RET>**	Long word examined at address 0
P 000000 10101010	
RDM>**D/B * 11<RET>**	Byte 11 deposited to address 0
RDM>**D + 22<RET>**	Byte 22 deposited to address 1
RDM>**D + 33<RET>**	Byte 33 deposited to address 2
RDM>**D + 44<RET>**	Byte 44 deposited to address 3
RDM>**E/L 0<RET>**	Long word examined at address 0
P 000000 44332211	
RDM>	

2.4.2.5 Examine

Purpose: Examines data at a VAX memory address

Syntax: E [/modifier] [address]

This command reads data located at a hexadecimal VAX memory address specified in the command string. Modifiers (/W for word, /L for long word, and /B for byte) define the data type to be examined. If a modifier is used, it becomes the default data type; if not, then the current default data type is assumed.

If the address field in the command is omitted, the last memory location examined is incremented by the specified or default data type and used as the default address. The address field may be replaced by an asterisk (*) in order to use the address referenced in the last Examine or Deposit command.

Blocks of addresses may be examined by using the Repeat command syntax (for example, R E).

Sample Dialogue 1

RDM>**E/L 0<RET>**	Long word examined at memory location 0
P 000000 12345678	P indicates physical address 000000 is address examined 12345678 is data examined
RDM>	

Sample Dialogue 2

```
RDM>E<RET>          Address incremented by default data type (long word) and examined

  P 000004 9ABCDEF0

RDM>
```

Sample Dialogue 3

```
RDM>E/W 0<RET>      Word examined at address 0

  P 000000 5678

RDM>
```

Sample Dialogue 4

```
RDM>E<RET>          Address incremented by default data type (word) and examined

  P 000002 1234
```

Sample Dialogue 5

```
RDM>E/B 1<RET>      Byte examined at address 1

  P 00001 56
RDM>E *<RET>        Byte examined at address 1

  P 000001 56
RDM>
```

2.4.2.6 Examine Console

Purpose: Inspects data at RDM random access memory (RAM) or control register address

Syntax: E/C [address]

This command displays the contents of the RDM status registers and RAM addresses. If an address is not specified, the current default address is incremented and examined. The default address is the one used in the previous Examine Console command. When using this command, data may be inspected only at addresses within the hexadecimal range 8000 to FFFF.

The RDM status register at address F820 is especially important to engineers who are testing the VAX remotely. This register indicates the state of the processor as shown in Figure 2-3.

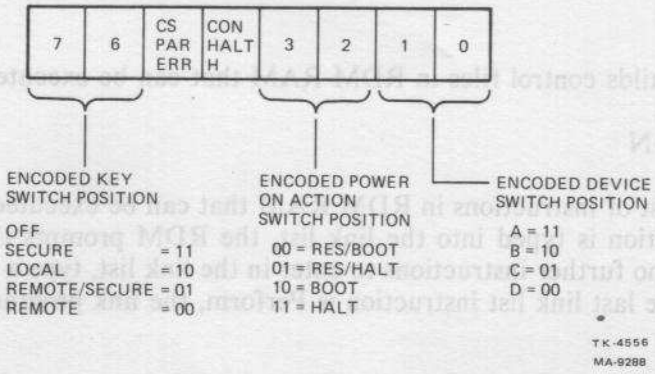


Figure 2-3 RDM Status Register F820 Decoding

Sample Dialogue

RDM>**E/C F820<RET>**	RDM status register F820 examined
R F820 DF RDM>	R indicates RDM RAM or status register F820 is address examined, and DF is contents of examined register in hexadecimal

2.4.2.7 Initialize

Purpose: Initializes CPU

Syntax: INI

This command simulates a power-fail sequence in order to recover the CPU from a hung condition. The command asserts the CPU's ACLO signal, followed 5.8 milliseconds later by DCLO. This sequence initializes the CPU to a newly powered-up state; however, main memory is not swept clean so that a good error correction code (ECC) will be present.

Sample dialogue 2 shows how the initialize command may be used with Repeat Next Command to generate a scope loop of power-fail activity. This might help to deal with CPU power-fail and recovery problems.

Sample Dialogue 1

RDM>**INI<RET>**

RDM>

Sample Dialogue 2

RDM>**R INI<RET>**

RDM>

2.4.2.8 Link

Purpose: Builds control files in RDM RAM that can be executed

Syntax: LIN

This command builds a list of instructions in RDM RAM that can be executed with the Perform command. After each instruction is typed into the link list, the RDM prompts the next instruction with LNK>. When there are no further instructions to enter in the link list, type a CTRL/C in response to the LNK> prompt. If the last link list instruction is Perform, the link program will loop continuously until CTRL/C is typed.

NOTE

A link list is destroyed by powering down or by entering any of the following commands: Return, Return/D, Test, Test Command, or Test File.

Sample Dialogue

Refer to the sample dialogue provided in Paragraph 2.4.2.15 (Perform).

2.4.2.9 Load

Purpose: Loads files from TU58 tape to VAX main memory.

Syntax: LO <filename.extension> [address]

This command causes the RDM to locate a specific file on the TU58, read it from tape, and deposit it at sequential addresses in VAX main memory. The file must be formatted for the RT11 operating system. The specified address is the starting address in VAX memory of the deposited file. If an address is not specified, the default address is zero. The filename must have exactly six characters. The extension must have exactly three characters.

The Load command loads files from the TU58 when the CPU cannot bootstrap them. This happens when the CPU is not functional enough to bootstrap, or the desired file is not properly hooked to an RT-11 boot block. In either case, the RDM needs only the CPU B clock to achieve the load.

This command may be used to load diagnostics such as EVKAA, in order to get useful error information about a failing CPU.

Sample Dialogue 1

RDM>**LO ECKAL.EXE<RET>**	Diagnostic ECKAL loaded into VAX main memory starting at address 0
RDM>	

Sample Dialogue 2

RDM>**LO EVKAA.EXE 10000<RET>**	Diagnostic EVKAA loaded into VAX main memory starting at address 10000
RDM>	

Sample Dialogue 3

RDM>**LO ECSAA.EXE FE00**<**RET**> Diagnostic ECSAA loaded into VAX main memory starting at address FE00

RDM>

2.4.2.10 Local

Purpose: Provides dual local/remote control of RDM

Syntax: CTL

This command allows the local user or DRDC to communicate with and control the RDM. You may enter this command only from the DRDC.

CAUTION

When both local and remote terminals communicate with the RDM, do not enter data from both terminals at the same time. Otherwise command strings could become mixed.

Sample Dialogue

RDM> **CTL**<**RET**>

RDM>

2.4.2.11 Local Disable

Purpose: Disables Local command function

Syntax: CTL/D

This command cancels the RDM's ability to accept commands from or send copy to the local terminal during remote diagnosis. You may enter this command only from the DRDC.

Sample Dialogue

RDM> **CTL/D**<**RET**>

RDM>

2.4.2.12 Microaddress

Purpose: Latches contents of selected microaddress into CPU control store latches

Syntax: UA <address>

This command halts the CPU in order to load a selected address on the control store (CS) address bus. When the CPU restarts, the new address takes control and the CPU checks the parity of the addressed data.

NOTE

This command disables any stop-on-micromatch function previously set.

This command can perform the following functions.

1. It applies the contents of an address with bad parity to the CPU logic checkers. You may then find the bad bits statically with a scope, logic probe, or other test equipment.
2. It performs a scope loop of a given microaddress at RDM execution speed. This is done by specifying Repeat execution (Paragraph 2.4.2.17) of the command and typing CTRL/O to suppress terminal output.
3. It starts the micromachine at a desired address.

NOTE

Use the Continue command to restart the CPU at the new address.

Sample Dialogue

```
RDM>UA 17FD<RET>

  CLK STOPPED CSAD 17FD          NEXT 0020

RDM>
```

NOTE

The microinstruction at location 17FD has incorrect control store parity bits to test the VAX parity checking logic.

2.4.2.13 Microaddress/C

Purpose: Temporarily loads a specific microaddress on the control store (CS) address bus

Syntax: UA/C <address>

This command places a selected address on the CS address lines until the next master (M) clock tick. At that point, the CPU latches data on the CS bus prior to the command. This command, therefore, does not change the program flow in the CPU.

NOTE

This command disables any stop-on-micromatch function previously set.

This command can isolate a failure in the CPU CS latching mechanism to one of the following areas.

- CS bus
- Control store
- CS latch
- CS control signal

Sample Dialogue

```
RDM>UA/C 17FD<RET>
RDM>
```

2.4.2.14 Parity

Purpose: Checks VAX control store (CS) parity

Syntax: PAR <address>

This command runs a parity check of the CPU control store. The parity check starts at the selected address and ends at the first parity error. If no parity error is found, the parity check stops at preset error address 17FD. The command displays the location of any parity error found. The CPU clock must be stopped before using this command.

Sample Dialogue

```
RDM>PAR 0<RET>          CPU stopped and parity check run from address 0
 PARITY ERROR CSAD 0AD1 Parity error found at address 0AD1
RDM>
```

NOTE

The writable control store (WCS) may be included in the parity check by specifying hexadecimal addresses 2000 through 23FF. The WCS must be loaded after power-up to avoid apparent errors resulting from power-up. The WCS is an option on the VAX-11/750.

2.4.2.15 Perform

Purpose: Executes a link program

Syntax: PER

This command executes the program in the link file until the program stops or CTRL/C is typed.

Sample Dialogue

```
RDM>LIN<RET>            User builds a link
                        program that deposits
LNK>D/B 0 11<RET>       data to successive
                        bytes in the
LNK>D + 22<RET>         long word
                        at address 0,
LNK>D + 33<RET>         and then
                        examines the long
LNK>D + 44<RET>         word at address
                        0. The last instruction
```

LNK>**E/L 0<RET>**	is Perform, to make the program loop
LNK>**PER<RET>**	until a CTRL/C is typed.
LNK>**CTRL/C**	CTRL/C terminates building of link list.
RDM>**PER<RET>**	Link list performs continuously until
D/B 0 11	CTRL/C.
D + 22	
D + 33	
D + 44	
E/L 0	
P 000000 44332211	P indicates physical address
PER	
D/B 0 11	
D + 22	
D + 33	
D + 44	
E/L 0	
CTRL/C	
P 000000 44332211	
↑C	
RDM>	

2.4.2.16 Repeat Last Command

Purpose: Repeats execution of the last RDM command

Syntax: REP

This command continuously executes the preceding command until CTRL/C is typed. It helps generate scope signals off the continuous execution of selected commands. When using Repeat, type CTRL/O to stop terminal output. Repeat also helps examine or deposit to blocks of memory area. (Refer to Paragraphs 2.4.2.4 and 2.4.2.5.)

Sample Dialogue

RDM>**E 0<RET>**	Location 0 examined
P 000000 12345678	P indicates physical address
RDM>**REP<RET>**	Location 0 examined continuously until CTRL/C typed
P 000000 12345678	
P 000000 12345678	
CTRL/C ↑C	
RDM>	

2.4.2.17 Repeat Next Command

Purpose: Repeats execution of selected RDM command

Syntax: R <command>

This command continuously executes a given command until CTRL/C is typed. It helps generate scope signals off the continuous execution of selected commands. When using Repeat, type CTRL/O to stop terminal output. Repeat also helps examine or deposit to blocks of memory area. (Refer to Paragraphs 2.4.2.4 and 2.4.2.5.)

Sample Dialogue

RDM>**R E 0<RET>**	Location 0 continuously examined until CTRL/C typed
P 000000 12345678	P indicates physical address
P 000000 12345678	
P 000000 12345678	
CTRL/C ↑C	
RDM>	

2.4.2.18 Return

Purpose: Returns system to program I/O state

Syntax: RET

This command switches the system from RDM console state to program I/O state

NOTE
This command disables any stop-on-micromatch function previously set.

Sample Dialogue

RDM>**RET<RET>**
$

2.4.2.19 Return/D

Purpose: Returns system to the control mode of RDM console state

Syntax: RET/D

This command switches the system from the command mode to the control mode of RDM console state. In control mode, the system is at the prompt level of the program currently running in the CPU. However, unlike program I/O state, a CTRL/D returns the system to the RDM console command mode whether the CPU is running or not. Any previously set stop-on-micromatch function remains enabled after the RET/D command.

Sample Dialogue

```
RDM>RET/D<RET>
$
```

2.4.2.20 Set

Purpose: Enables stop-on-micromatch function

Syntax: SE [address]

This command stops the CPU clock when the contents of the control store (CS) address bus equal the contents of the RDM match register. The address to match can be specified by entering an address in the command syntax. If an address is omitted in the command syntax, the CPU stops at the address which matches the one already contained in the match register. When the CPU stops, the RDM displays the addresses of both the last instruction executed and the next instruction to execute. *This function remains set if you switch to the Control Mode of the RDM Console State.*

The command stops the CPU at a particular address from which trace of the VAX control store can be done. This shows which path the CPU took to reach that point in the control store.

Sample Dialogue

```
RDM>SE 3FE<RET>

  CPU STOPPED CSAD 03FE                NEXT 03FF

RDM>
```

2.4.2.21 Show

Purpose: Displays current CPU state

Syntax: SH

This command displays the current operating state of the CPU: running, halted, or stopped. Running means the CPU is executing code. Halted means the CPU is not executing code, but its clock is running. Stopped means the CPU clock is stopped. The command also displays the addresses of both the last instruction executed and the next instruction to execute.

Sample Dialogue

```
RDM>SH<RET>

    CPU STOPPED CSAD 3402          NEXT 3403

RDM>
```

2.4.2.22 Show Version

Purpose: Displays version and date of RDM firmware.

Syntax: SH/V

This command displays the current revision level and date of the firmware running in the RDM. If your procedure depends on the version of the RDM you have, ask the RDM for the version number. The version number you see may be different from the one shown in the Sample Dialogue below.

Sample Dialogue

```
RDM>SH/V<RET>

18-DEC-80 LV=25

RDM>
```

2.4.2.23 Step

Purpose: Steps CPU through single microinstruction

Syntax: STE

This command causes the CPU to execute a single microinstruction and stop. It then displays the address (CSAD) of the current microinstruction latched in the CPU and the address of the next microinstruction to latch.

Sample Dialogue

```
RDM>STE<RET>

  CSAD 1701                    NEXT 1702

RDM>
```

2.4.2.24 Step Tick

Purpose: Advances CPU in base (B) clock increments

Syntax: STE/T

Use the Step Tick command to do the following:

- Step through a microinstruction in B clock increments
- Display the control store address (CSAD) whose contents currently are latched by the CPU
- Display the CSAD whose contents will be latched next by the CPU.

It takes one M clock cycle to execute (or "latch") each microinstruction. There are normally two B clock cycles for each M clock cycle. During the first half of the M clock cycle, the CSAD lines are in the HI-Z state. (That is, the third state of a tristate output.) Therefore, the next CSAD value is not yet visible to the RDM. However, after stepping the B clock to the second half of the microinstruction, the next CSAD value is clearly visible. By stepping the B clock once again, the CPU should latch the microinstruction at the next CSAD. The Step Tick command reveals this latching sequence. It also identifies those microinstructions that for some reason may take longer than two B clock cycles to execute.

Step Tick can be used by itself, or with the Repeat-Command syntax, after stopping the CPU clock.

Sample Dialogue 1

```
RDM>STE/T<RET>

  CSAD 17DB                NEXT IFE9

RDM>
```

Sample Dialogue 2

```
RDM>R STE/T<RET>

  CSAD 17DB                NEXT IFE9
  CSAD 1FE9                NEXT 17D7
  CSAD 1FE9                NEXT 17DF
  CSAD 17DF                NEXT 17DF
  CSAD 17DF                NEXT 17DF

CTRL/C ↑C

RDM>
```

2.4.2.25 Stop

Purpose: Stops CPU clock

Syntax: STO

This command stops the CPU clock, then displays the addresses of both the last instruction executed and the next instruction to execute.

Sample Dialogue

```
RDM>STO<RET>

CPU STOPPED CSAD 0964          NEXT 0966

RDM>
```

2.4.2.26 Talk

Purpose: Changes system from RDM console state to talk state

Syntax: TA

This command allows communication between local and remote terminals. You enter it from the terminal currently controlling the VAX-11/750 system. The Talk command displays characters typed at either terminal on both terminals.

If you enter the talk state with the keyswitch turned to LOCAL, the system asserts DTR to the modem and awaits a telephone link to DRDC. Completion of the DRDC link establishes a talk state.

If you enter the talk state with the keyswitch turned to REMOTE or REMOTE SECURE, the command immediately places the system in the talk state.

Type a CTRL/C to return from talk state to RDM console state. To return from talk state to program I/O state, turn the keyswitch to LOCAL SECURE.

Sample Dialogue

```
RDM> TA<RET>

ENTER TALK MODE

RDM>
```

2.4.2.27 Test

Purpose: Loads and runs microdiagnostics

Syntax: TE

This command loads the microdiagnostic monitor into RDM RAM, and the monitor runs a series of microdiagnostics on the VAX-11/750 system. The monitor (MICMON) and microdiagnostics reside on TU58 tape cassettes. The correct cassette must be inserted before using Test.

MICMON cancels the tests if it detects an error or CTRL/C is typed while the tests are running. The monitor then prints an error message and gives the MIC> prompt on a new line. The MIC> prompt indicates communication with MICMON in the microdiagnostic state.

If CTRL/C is typed while the RDM is loading MICMON into RAM, the RDM stops loading MICMON and returns to RDM console command mode.

If tests have not run to completion, perform one of the following procedures to exit the microdiagnostic state.

- Type RE to exit MICMON.
- Turn the keyswitch to LOCAL SECURE or REMOTE SECURE in order to return to the program I/O state.
- Type CTRL/D to return to RDM console command mode.

Refer to Chapter 5 for information and sample dialogues on how to apply the Test command.

NOTE
The logical name for MICMON is ECKAA.EXE.

2.4.2.28 Test/C

Purpose: Loads microdiagnostic monitor

Syntax: TE/C

This command loads the microdiagnostic monitor (MICMON) into RDM RAM, and gives the monitor control of the RDM. With Test/C (unlike Test), MICMON waits for terminal commands before running any microdiagnostics. With MICMON in RAM, the system enters the microdiagnostic state and the active terminal displays the MIC> prompt.

MICMON and the microdiagnostics reside on TU58 tape cassettes. The correct cassette must be inserted before using Test Command.

If CTRL/C is typed while the RDM is loading MICMON into RAM, the RDM stops loading MICMON and returns to RDM console command mode.

To exit the microdiagnostic state, perform one of the following procedures.

- Type RE to exit MICMON.
- Turn the keyswitch to LOCAL SECURE or REMOTE SECURE in order to return to the program I/O state.
- Type CTRL/D to return to RDM console command mode.

Refer to Chapter 6 for information and sample dialogues on how to apply Test/C Command.

2.4.2.29 Test File

Purpose: Loads user program into RDM RAM and runs it

Syntax: TE <filename.extension>

This command loads programs other than the microdiagnostic monitor into RDM RAM and executes them. The filename in the command argument must consist of exactly six characters. The extension must consist of exactly three characters.

Sample Dialogue

```
RDM>TE MICMON.TST<RET>
```

2.4.2.30 Trace

Purpose: Displays the 65 most current control store (CS) addresses

Syntax: TR

This command displays in reverse order the CS addresses stored in the RDM diagnostic control store (DCS). Except in the microdiagnostic state, the RDM stores the addresses of the 64 most recently executed VAX CS instructions in its DCS. The address of the next instruction to execute is stored in a DCS trap register and is also displayed.

Type CTRL/C or CTRL/D to stop the display of the 65 stored addresses at any point. Either command returns the RDM> prompt at once.

The CPU clock must be stopped before using the Trace command. Trace can be used to see what path the CPU took just before its clock stopped.

Sample Dialogue

```
RDM>TR<RET>

  CSAD 1391                    NEXT 1392
  CSAD 1392
  CSAD 1393
  CSAD 13FC

CTRL/C

RDM>
```

2.4.3 Error Messages

The terminal prints appropriate error messages for a variety of conditions. Error messages print on the line following the command string that caused the error condition.

With some exceptions, error messages consist of six-character codes, where the message ERR- prints, followed by the code. Table 2-6 lists these error codes.

Example

```
RDM>TES<RET>          Command to load and run microdiagnostics mistyped
ERR - SYNTAX ERROR

RDM>
```

Table 2-6 Error Message Codes

Error Code	Definition
TAP:14	Tape – read length error, not all records fit
TAP:13	Tape – flag received, not command or data
TAP:12	Tape – directory error
RDM:11	Invalid operation code in macro
RDM:10	Operation already in progress
TRM:0E	Terminal – remote line CRC error
TRM:0D	Terminal – length of input longer than buffer
TRM:0B	Terminal – command input buffer overloaded
TAP:09	Tape – file not found
TAP:08	Tape – invalid packet received
TAP:07	Tape – no end packet, invalid operation code received
TAP:06	Tape – tape count byte received exceeds maximum
TAP:05	Tape – tape check sum error received
	NOTE **UARTs are RDM resident.**
TAP:04	Tape UART – overflow received
TAP:03	Tape UART – data set ready dropped
TAP:02	Tape UART – error received from UART
TAP:01	Tape UART – device timed out
CPU:04	CPU UART – overflow received
CPU:03	CPU UART – data set ready dropped
CPU:02	CPU UART – error received from UART
CPU:01	CPU UART – device timed out
TRM:04	Terminal UART – overflow received
TRM:03	Terminal UART – data set ready dropped
TRM:02	Terminal UART – error received from UART
TRM:01	Terminal UART – device timed out
REM:04	Remote UART – overflow received
REM:03	Remote UART – data set ready dropped
REM:02	Remote UART – error received from UART
REM:01	Remote UART – device timed out
TAP:FF	Tape – diagnostic failure
TAP:EE	Tape – partial operation (end of medium)
TAP:F8	Tape – bad unit number
TAP:F7	Tape – no cartridge
TAP:F5	Tape – write protocol
TAP:EF	Tape – data check error
TAP:EO	Tape – see error (block not found)
TAP:DF	Tape – motor stopped
TAP:DO	Tape – bad operation code
TAP:C9	Tape – bad record number
SYNTAX ERROR	Error in entering console commands
INVALID COMMAND	RDM does not recognize command

Table 2-6 Error Message Codes (Cont)

Error Code	Definition
CMI:nn	Error in VAX main memory – (two digits, nn, are error code) results from EXAMINE if area addressed has error
CMI:00	Nonexistent memory
CMI:01	Corrected read data
CMI:02	Read data substitute
ROM	ROM failed RDM power-up self-test
RAM	RAM failed RDM power-up self-test

CHAPTER 3
INSTALLATION

3.1 SCOPE

This chapter describes how to unpack, install, and inspect the KC750 remote diagnostic module (RDM) and modem.

3.2 CUSTOMER REQUIREMENTS

Paragraphs 3.2.1 through 3.2.4 describe the steps you and the customer must take to prepare for a KC750 option installation.

3.2.1 System Configuration Worksheet

Make sure the prospective customer site is evaluated and System Configuration Worksheets supplied to DIGITAL's Remote Diagnostic Center (DRDC) before scheduling any installations of a KC750 option. The DRDC requires complete and accurate worksheets to build a configuration file.

3.2.2 Telephone Company Notification (US Only)

The customer must furnish the following information to the telephone company before the installation of a KC750-CA option.

1. Modem manufacturer's name
2. Modem model number
3. Modem FCC registration number (supplied by DIGITAL district office)
4. Modem ringer equivalence number (supplied by DIGITAL district office)
5. Modem speed
6. Type of modem voice jack direct-connect receptacle
7. Telephone number of line/RJ11C, if installed

Table 3-1 lists the data for modems supplied with the KC750-CA option.

For the KC750-DA option, the customer must order both the phone line and modem. The modem must be a Bell Model 103J or an equivalent.

Table 3-1 KC750-CA Option Modem Data

Modem Manufacturer	Model Number	FCC Registration Number	Ringer Equivalence Number
GDC	103A3	AG697J-62418-DM-E	0.6B
Vadic	VA355P	AJ496M-70263-DM-N	1.0B
DIGITAL	DF02	A09994Q-67693-DM-R	Q.3A

NOTE
All three modems use an RJ11C voice jack and operate at 300 bits/s.

3.2.2.1 Discontinuing Telephone Service (US Only) – Normally the modem should not disturb the telephone network. However, if the telephone network malfunctions due to a defective modem, the local telephone company will discontinue service. If practical, the telephone company notifies customers before discontinuing service. However, this is not always possible.

If service is discontinued, the local telephone company should perform the following steps.

1. Notify customers promptly that service has been discontinued.

2. Give customers the opportunity to correct the situation that caused the temporary break in service.

3. Inform customers of their right to bring a complaint to the FCC. Complaint procedures can be obtained from the local telephone company or the modem manufacturer.

3.2.2.2 Notifying Customers of Telephone Service Changes (US Only) – Any local telephone company changes of equipment or procedures must be consistent with FCC regulations. Customers should receive adequate written notice if the changes might create any incompatibility between customer terminal equipment and new equipment or procedures. This allows them the opportunity to maintain uninterrupted service.

3.2.3 Customer Acknowledgement

The customer must sign an Installation Acknowledgement form to ensure DIGITAL's full and free access to equipment. (Appendix C includes a sample form.)

3.2.4 Customer Responsibilities

The customer is responsible for preparing the system for remote diagnosis after installation of the KC750 option. Preparation includes mounting the system diagnostic pack and scratch media.

3.3 MODEM REPAIR AND REPLACEMENT (US Only)

Return modems shipped with the KC750-CA option to Stockroom 126 in Woburn for repair and replacement. If one vendor's modem is replaced with a different vendor's modem (or even a different model type), the customer must notify the telephone company of the new FCC registration number, ringer equivalence, and other items listed in Paragraph 3.2.2.

The on-site Field Service engineer is responsible for notifying the customer of changes and providing technical information which the customer must report to the local telephone company.

3.4 UNPACKING AND INSPECTION

Open the shipping container carefully. Check the contents against the inventory list in Paragraph 1.4 and inspect all parts for damage. Immediately report any damage to the responsible carrier and branch office supervisor. Do not start installation if any item is missing or damaged; wait until the item is replaced or repaired.

3.5 INSTALLATION PROCEDURES

Perform the following procedure to install the KC750 option in the VAX-11/750 processor. Refer to Figure 3-1 for the cabling diagram.

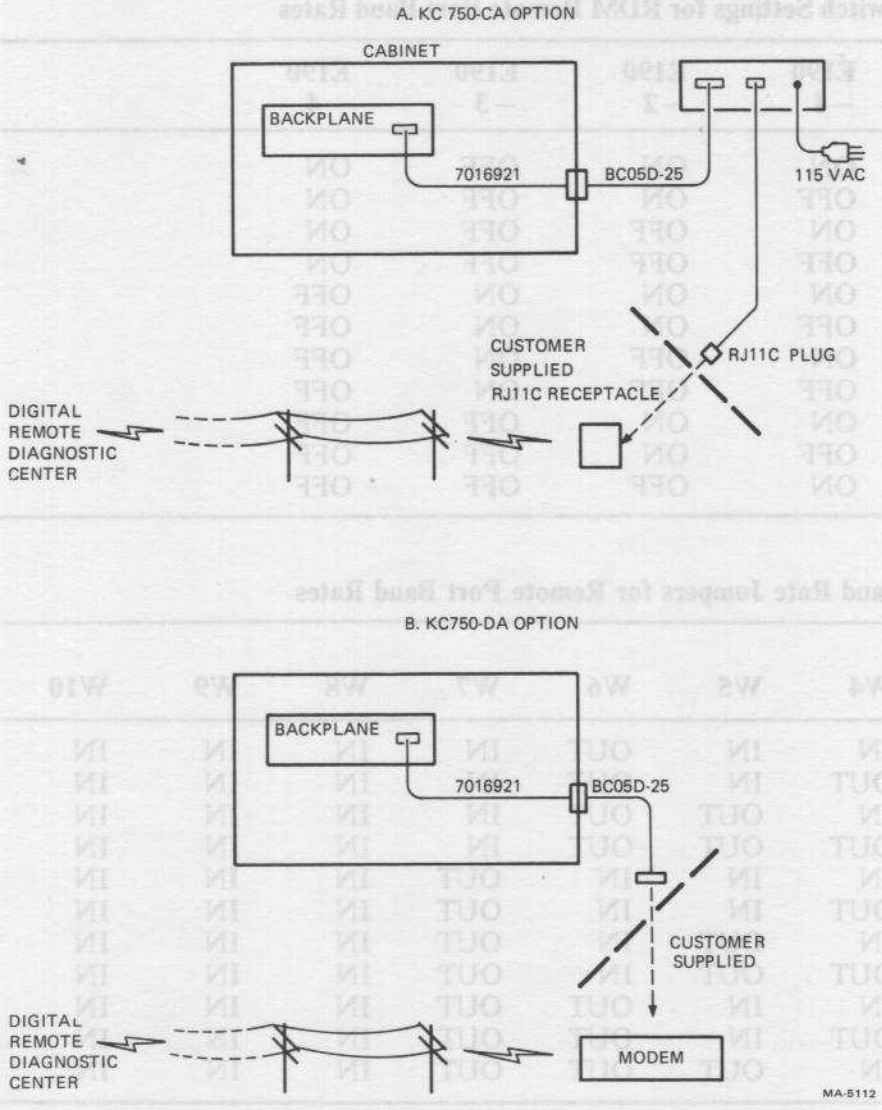


Figure 3-1 KC750 Cabling

1. Power system off by moving the keyswitch on the operator control panel to the OFF position.
2. Set the POWER ON ACTION switch to the HALT position.
3. Set DEVICE switch to position that corresponds to the TU58 tape drive.
4. Program the RDM remote port line baud rate. If the RDM has the 4-position switch pack E190, set the switches according to Table 3-2. If the RDM has jumpers, insert or remove wires according to Table 3-3.

Table 3-2 Switch Settings for RDM Remote Port Baud Rates

Baud Rate	E190 −1	E190 −2	E190 −3	E190 −4
300	ON	ON	OFF	ON
400	OFF	ON	OFF	ON
600	ON	OFF	OFF	ON
1200	OFF	OFF	OFF	ON
2400	ON	ON	ON	OFF
3600	OFF	ON	ON	OFF
4000	ON	OFF	ON	OFF
4800	OFF	OFF	ON	OFF
7200	ON	ON	OFF	OFF
9600	OFF	ON	OFF	OFF
19200	ON	OFF	OFF	OFF

Table 3-3 Baud Rate Jumpers for Remote Port Baud Rates

Baud Rate	W4	W5	W6	W7	W8	W9	W10	W11
300	IN	IN	OUT	IN	IN	IN	IN	IN
400	OUT	IN	OUT	IN	IN	IN	IN	IN
600	IN	OUT	OUT	IN	IN	IN	IN	IN
1200	OUT	OUT	OUT	IN	IN	IN	IN	IN
2400	IN	IN	IN	OUT	IN	IN	IN	IN
3600	OUT	IN	IN	OUT	IN	IN	IN	IN
4000	IN	OUT	IN	OUT	IN	IN	IN	IN
4800	OUT	OUT	IN	OUT	IN	IN	IN	IN
7200	IN	IN	OUT	OUT	IN	IN	IN	IN
9600	OUT	IN	OUT	OUT	IN	IN	IN	IN
19200	IN	OUT	OUT	OUT	IN	IN	IN	IN

Jumpers W8, W9, W10, and W11 may not be present on a RDM. If they are present, W8, W9, and W10 must always be connected. W11 is simply a spare jumper and may be removed. Figure 3-2 shows the locations of the baud rate jumpers on the RDM. Figure 3-2 also shows a closeup of the baud rate jumpers. If the RDM is to be installed in Europe, remove W3 as shown in Figure 3-3.

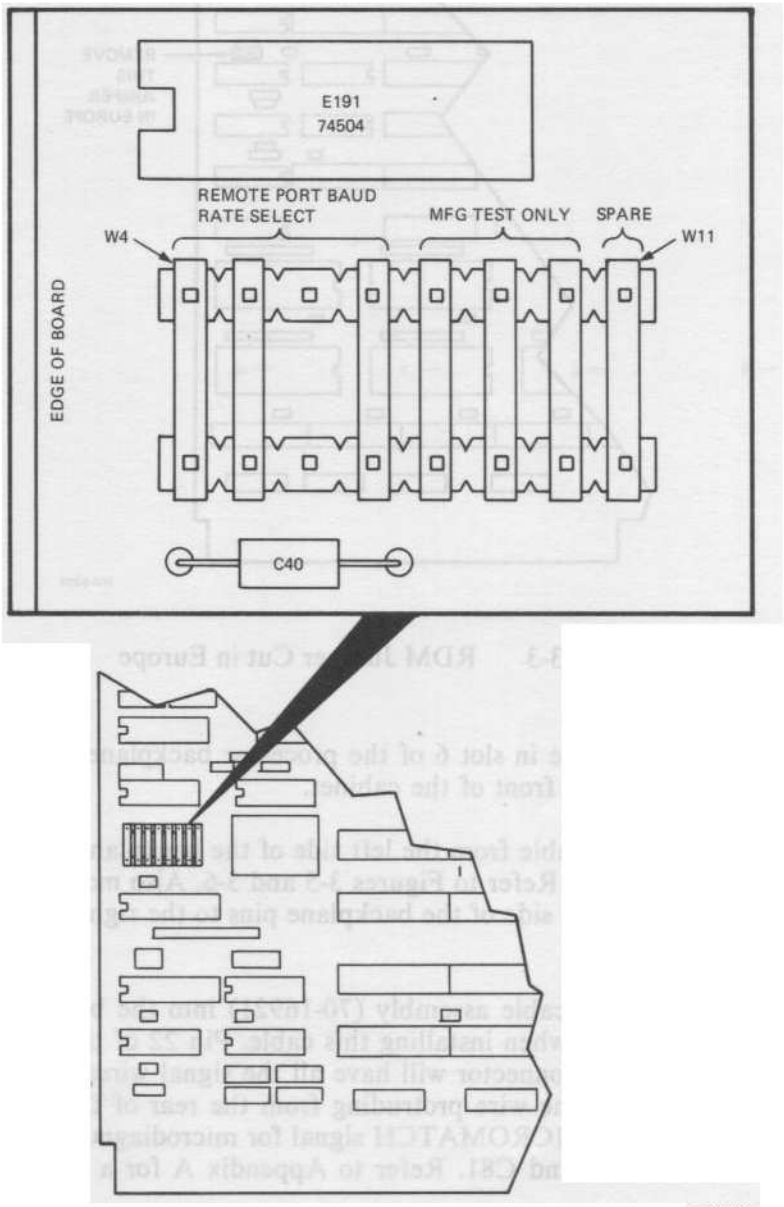


Figure 3-2 Location of Baud Rate Jumpers on the RDM

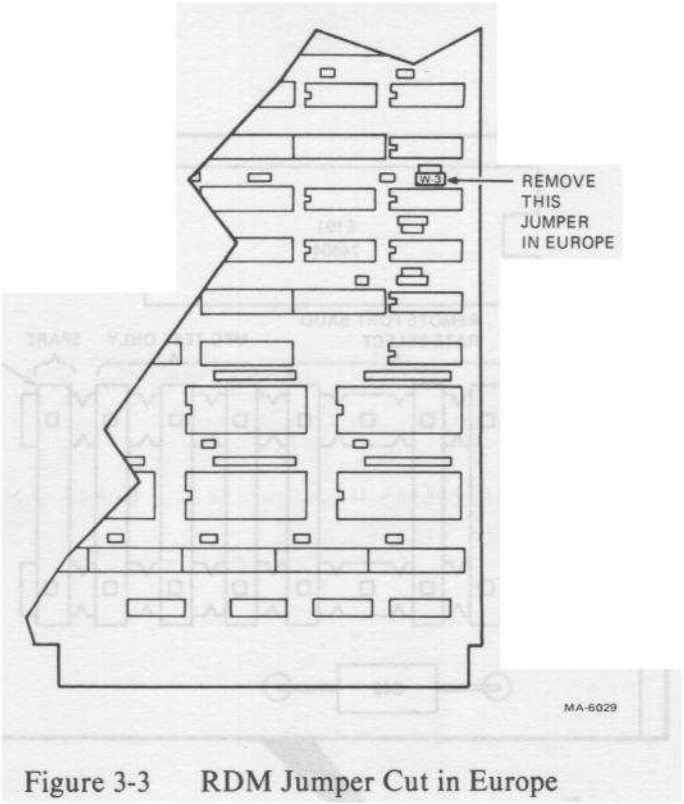


Figure 3-3 RDM Jumper Cut in Europe

5. Install the L0006 Module in slot 6 of the processor backplane. Refer to Figure 3-4 for the module placement in the front of the cabinet.

6. Move the TU58 signal cable from the left side of the backplane pins to the right side of the backplane pins on slot 6. Refer to Figures 3-5 and 3-6. Also move the local console terminal signal cable from the left side of the backplane pins to the right side. Again refer to Figures 3-5 and 3-6.

7. Plug the RDM filtered cable assembly (70-16921) into the bottom of section C of slot 6. Observe label markings when installing this cable. Pin 22 of the connector must connect to pin C94 on slot 6. The connector will have all the signal wires connected to the even numbered backplane pins. The wire protruding from the rear of the connector is used for connecting a scope to the MICROMATCH signal for microdiagnostic use. MICROMATCH is connected to Pins TP1 and C81. Refer to Appendix A for a description of the backplane signals.

8. Inspect the supplied modem and make sure all programmable switches and jumpers are correctly installed. Also set all the modem controls as defined in Appendix B.

9. Attach the other end of the filtered cable assembly to I/O connector panel as shown in Figure 3-7.

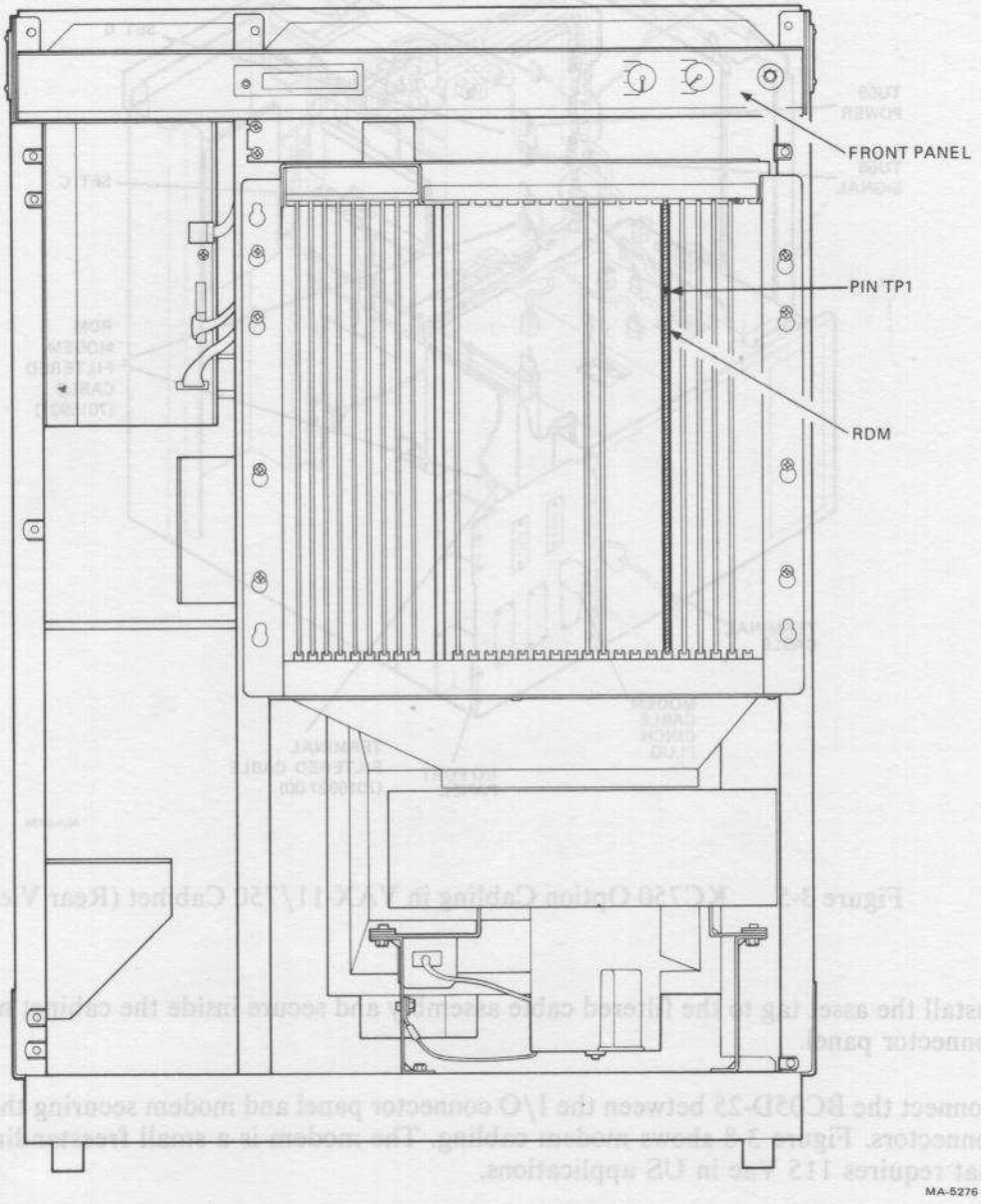


Figure 3-4 RDM in VAX-11/750 Cabinet (Front View)

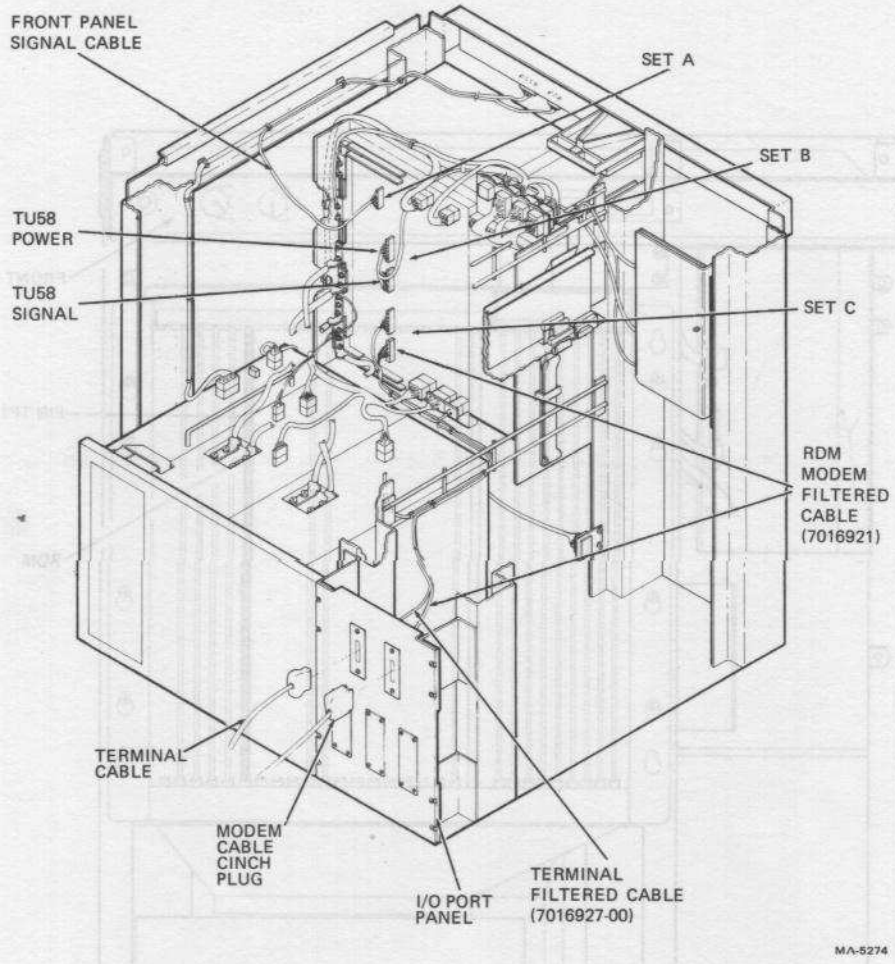


Figure 3-5 KC750 Option Cabling in VAX-11/750 Cabinet (Rear View)

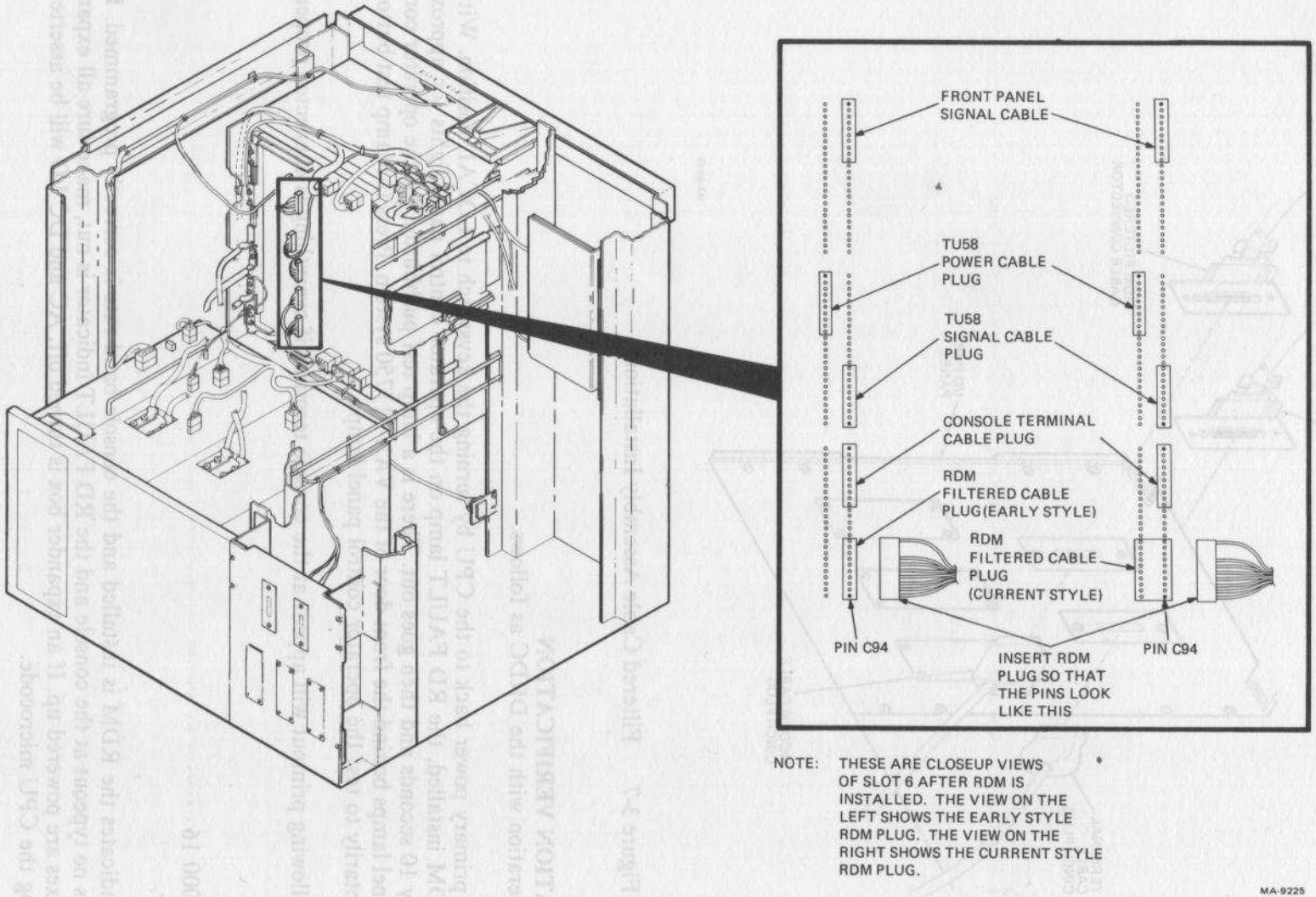
10. Install the asset tag to the filtered cable assembly and secure inside the cabinet near the I/O connector panel.

11. Connect the BC05D-25 between the I/O connector panel and modem securing the cinch plug connectors. Figure 3-8 shows modem cabling. The modem is a small freestanding assembly that requires 115 Vac in US applications.

CAUTION

Modem ac power should not come from the VAX-11/750 power controller since this violates UL regulations.

12. Connect the modem to the telephone jack that has been already installed. This is an RJ11C receptacle in the US. Refer to Figure 3-1 for complete circuit block diagram.



NOTE: THESE ARE CLOSEUP VIEWS OF SLOT 6 AFTER RDM IS INSTALLED. THE VIEW ON THE LEFT SHOWS THE EARLY STYLE RDM PLUG. THE VIEW ON THE RIGHT SHOWS THE CURRENT STYLE RDM PLUG.

MA-9225

Figure 3-6 Closeup of Backplane Slot Six

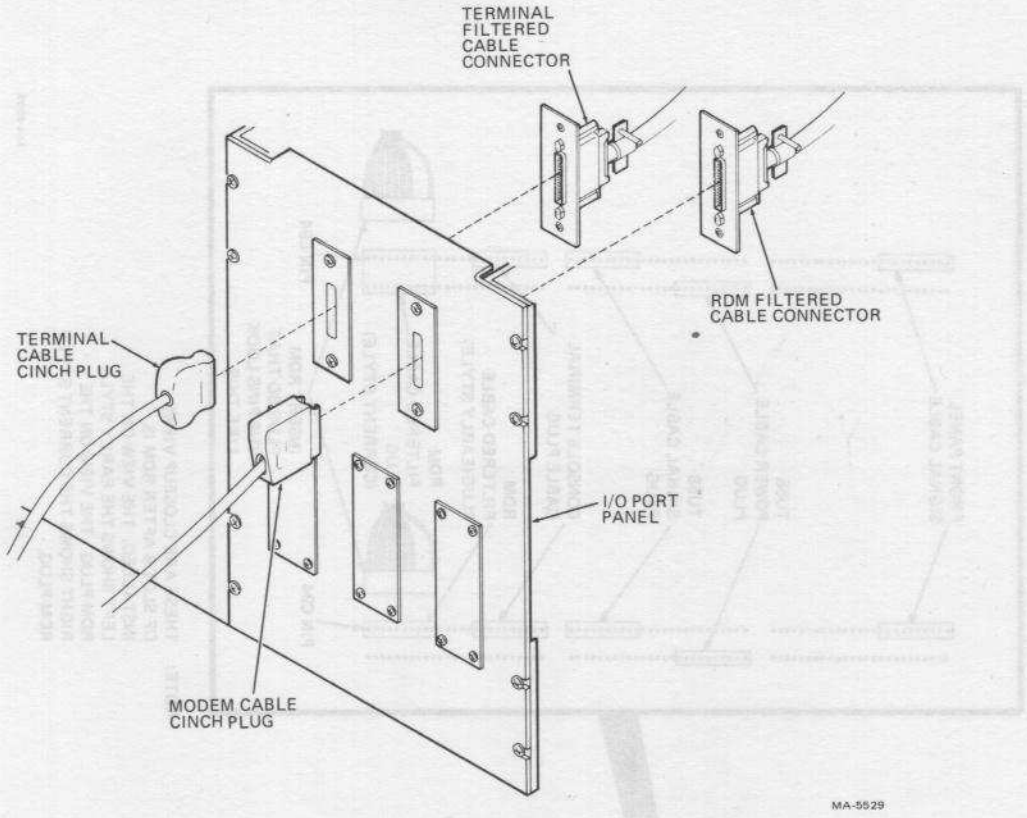


Figure 3-7 Filtered Cable Assembly Installation

3.6 INSTALLATION VERIFICATION

Verify system operation with the DRDC as follows.

1. Apply primary power back to the CPU by turning the keyswitch to LOCAL position. With the RDM installed, the RD FAULT lamp on the operator control panel lights for approximately 10 seconds and then goes out. There is a lamp test pushbutton for the operator control panel lamps behind the front door of the VAX-11/750 system. Press the lamp test button momentarily to test the operator control panel lamps.

2. The following printout will appear at the console terminal if everything is connected properly.

   ```
   %%
   00000000 16
   >>>
   ```

3. This indicates the RDM is installed and the console baud rate is correctly programmed. If there is no typeout at the console and the RD FAULT indicator is out, make sure all expander boxes are powered up. If an expander box is turned off, AC and DC LO will be asserted hanging the CPU microcode.

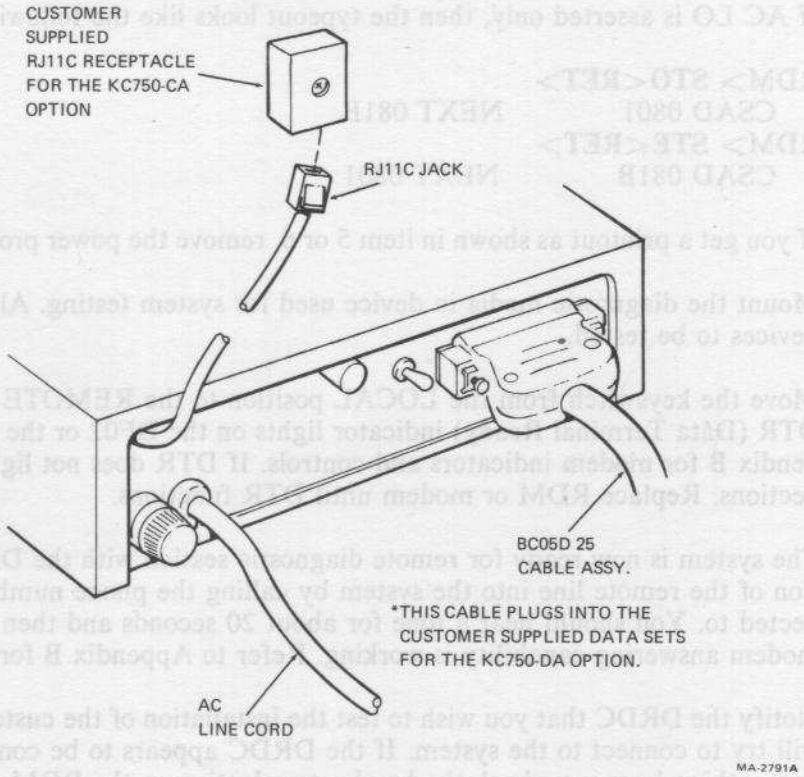


Figure 3-8 KC750-CA Modem Cabling

4. Test the RDM console by typing CTRL/D.

```
>>> CTRL/D ↑D

RDM> SH/V<RET>

10-Dec-80 LV=25
```

NOTE
This date may be different on your RDM.

5. Check for power problems with the following dialogue.

```
RDM> STO<RET>
   CSAD 0000          NEXT 0000
RDM> STE<RET>
   CSAD 0000          NEXT 0000
```

The recurring zeros in the typeout indicate a DC LO assertion on the UNIBUS somewhere.

6. If AC LO is asserted only, then the typeout looks like the following.

```
RDM> STO<RET>
   CSAD 0801          NEXT 081B
RDM> STE<RET>
   CSAD 081B          NEXT 0801
```

7. If you get a printout as shown in item 5 or 6, remove the power problem before continuing.

8. Mount the diagnostic media in device used for system testing. Also mount scratch media in devices to be tested.

9. Move the keyswitch from the LOCAL position to the REMOTE position. Observe that the DTR (Data Terminal Ready) indicator lights on the DF02 or the VADIC modems. See Appendix B for modem indicators and controls. If DTR does not light, check cabling and connections. Replace RDM or modem until DTR functions.

10. The system is now ready for remote diagnostic session with the DRDC. Test the basic function of the remote line into the system by calling the phone number that the modem is connected to. You should hear a tone for about 20 seconds and then silence. This indicates the modem answering capability is working. Refer to Appendix B for modem loopback testing.

11. Notify the DRDC that you wish to test the installation of the customer's system. The DRDC will try to connect to the system. If the DRDC appears to be connecting but then drops off and nothing happens, check the baud rate selection on the RDM switch pack E190 or backplane baud rate jumpers.

12. The indication of successful connection for each modem is explained in Appendix B. The first event you should see is the CARRIER indicator on the operator control panel light. Shortly after this, the RD TEST lamp should light and a message will be printed at the console terminal similar to the one below.

    ```
    Digital Diagnostic Center
    Connection established to XXXXXXXXXX
    ```

    where XXXXXXXXXX is the name of the system as defined in the system configuration file at the DRDC. The DRDC engineer can talk with the operator of the local terminal by typing the following.

    CTRL/D ↑D

    RDM> **TA<RET>**

    At this point the operator and the DRDC engineer can talk over the terminal. He may ask the operator to load diagnostic tapes to assist in the repair of the system and ask for help in diagnosing problems.

CHAPTER 4
FUNCTIONAL DESCRIPTION

4.1 SCOPE

This chapter provides information about the internal operations of the VAX-11/750 Remote Diagnostic Module (RDM). It describes the general RDM functions, its major components, and how they work together. While reading this chapter, refer to Figure 4-1 (the RDM system block diagram).

4.2 GENERAL FUNCTIONS

The following list describes the general functions of the diagnostic module.

1. Allows users to perform all regular processor console functions from the console terminal or DRDC
2. Allows remote and local control of the TU58 tape unit
3. Supports DRDC's running of microdiagnostic scripts (Scripts are programs that call and run sequences of routines to test the CPU and system peripherals.)
4. Loads and reads VAX-11/750 main memory
5. Loads and reads processor W-BUS in support of microdiagnostics
6. Drives VAX-11/750 control lines in place of regular control store during microdiagnosis
7. Traces and stores addresses of executed VAX-11/750 control store instructions

NOTE

You cannot perform functions 6 and 7 at the same time.

8. Compares the addresses of instructions executed by the main processor with a preselected address, and stops the processor when a match occurs (at the micro break point)
9. Runs a parity check on the processor control store
10. Conducts logic self-test of the RDM on power-up

4.3 MAJOR COMPONENTS

The major RDM components are a preprogrammed memory, a microprocessor, a diagnostic control store, and miscellaneous registers and control logic.

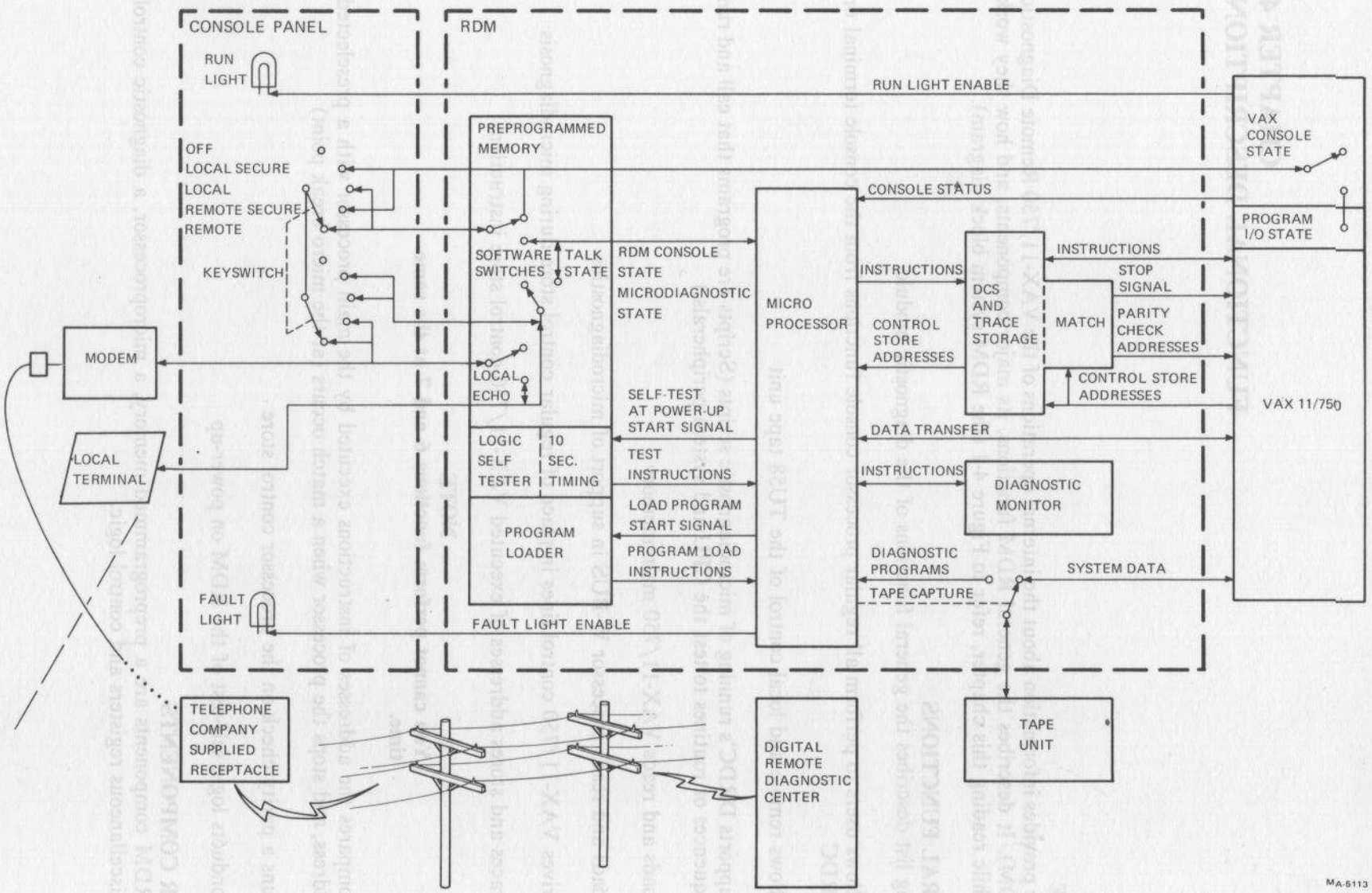


Figure 4-1 RDM System Block Diagram

4.3.1 Preprogrammed Memory

Preprogrammed memory consists of three parts: software switches, logic self-tester, and program loader. Some RDM preprogrammed memory functions connect to preprogrammed memory functions in the VAX-11/750 console. Therefore, the following discussion includes appropriate VAX console functions to complete the descriptions.

4.3.1.1 Software Switches – Software switches are instructions that change the RDM state and perform certain control functions. Each RDM state corresponds to a particular position of the software switches. Refer to Figures 4-2, 4-3, 4-4 and 4-5 for examples.

In program I/O state, the console terminal communicates with the VAX-11/750 at the program level. In VAX console state, the terminal communicates with VAX-11/750 firmware. The VAX console and RDM monitor these lines for control characters that set software switches to change states. In program I/O state or VAX console state, the RDM's microprocessor does not play an active role in relaying information to the VAX-11/750. Switching between these two states is the exclusive job of the VAX console.

The RDM microprocessor accepts commands from the console terminal only after the system switches to RDM console state. The RDM does this switching. The RDM processor only communicates directly with the terminal when the system is in RDM console state. The RDM console state (and other states entered via the RDM console state, such as the microdiagnostic state) may only be entered from the terminal if the keyswitch is not in the SECURE position. RDM preprogrammed memory contains another switch for the talk state. This state provides communication between the DRDC and local terminal.

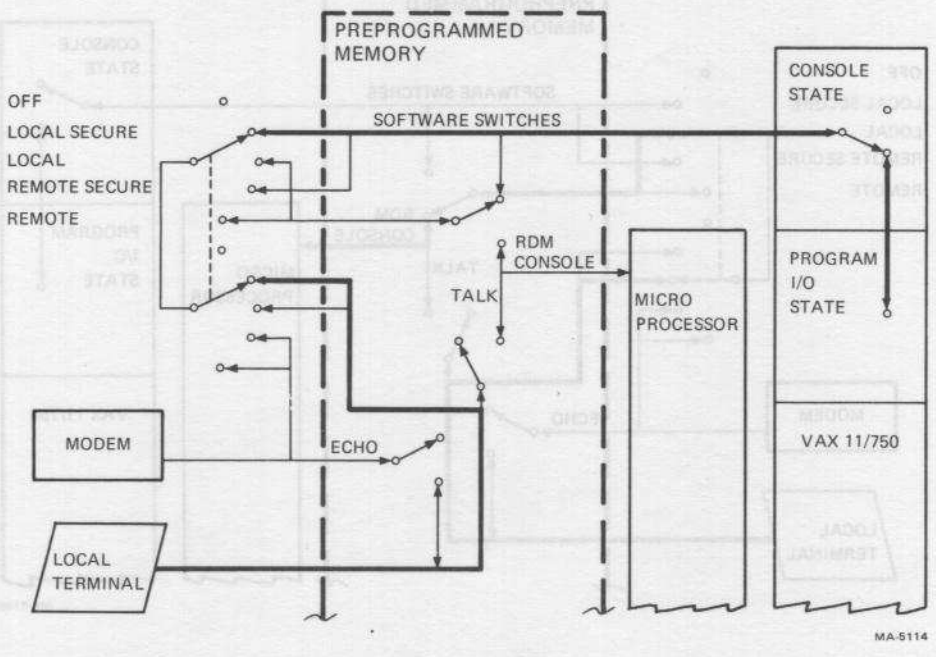


Figure 4-2 Local Secure Program I/O State

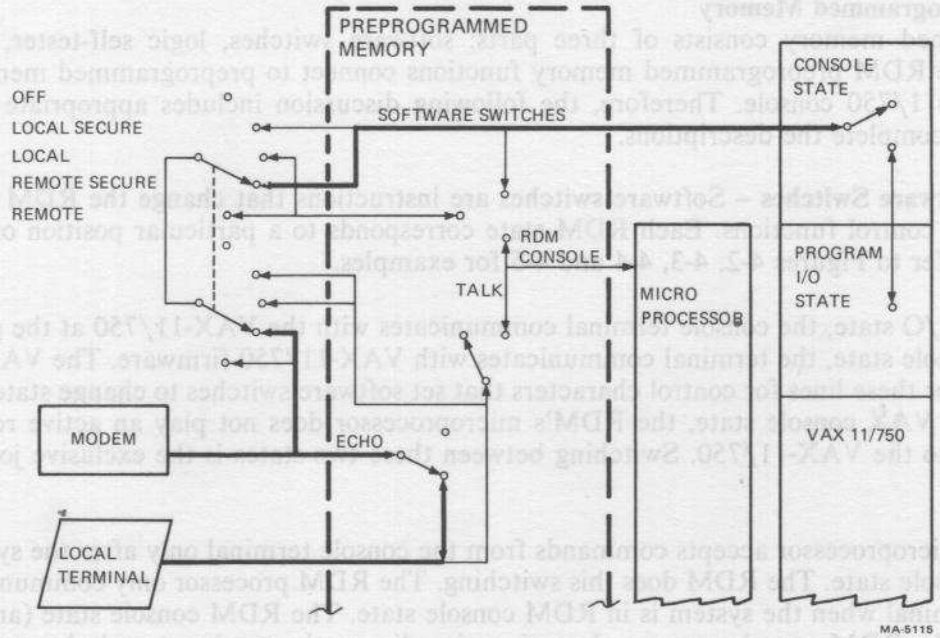


Figure 4-3 Remote Secure VAX Console State with Local Echo

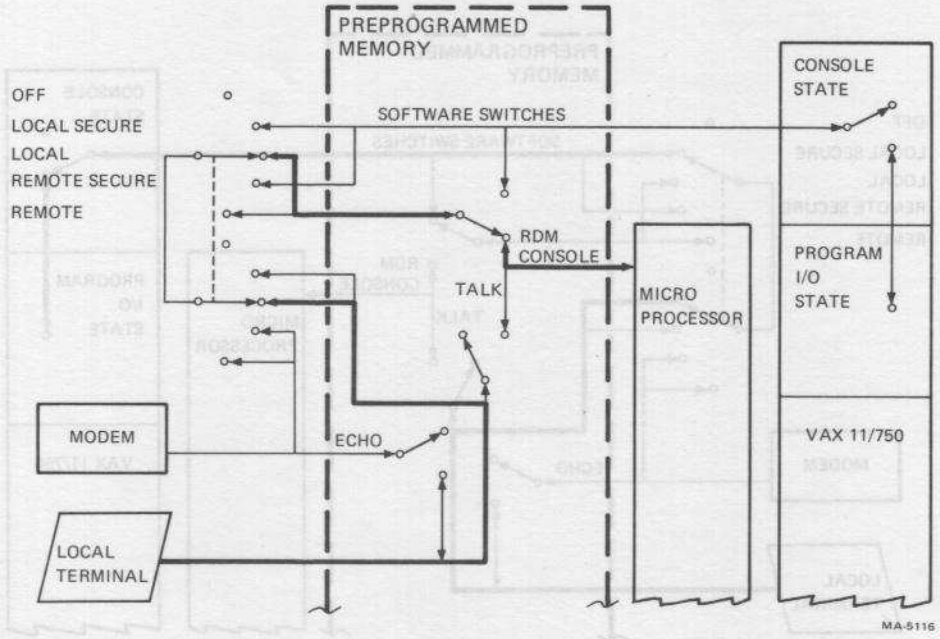


Figure 4-4 Local RDM Console State

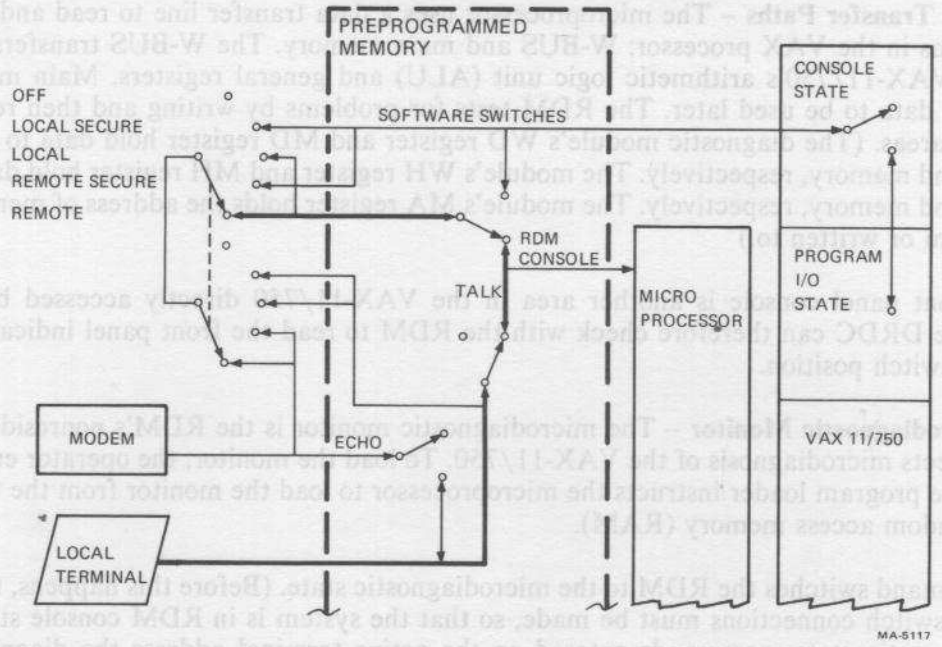


Figure 4-5 Remote Talk State

A similar switch exists for local copy. Local copy allows dialogue between the DRDC and RDM or VAX-11/750 to echo on the local terminal. The copy works in one direction only; that is, local terminal dialogue cannot echo on the DRDC terminal. (Local terminal input decodes as an interrupt at the DRDC. This interrupt alerts DRDC that the local operator requires attention.)

Chapter 2 describes all commands that switch system states and perform other control functions. Preprogrammed memory supports the control commands. The VAX console, RDM console, or both may recognize a given command. The current system state determines whether VAX or RDM supports a command. For example, VAX supports CTRL/O in either the program I/O or VAX console states; RDM supports CTRL/O in the RDM console or microdiagnostic states.

4.3.1.2 Logic Self-Tester – The logic self-tester is a group of preprogrammed instructions that verify RDM logic. On RDM powerup, the microprocessor signals the self-tester for instructions. These instructions tell the microprocessor to check various hardware parts and turn on the fault indicator. If the microprocessor does not find a hardware failure within 10 seconds, the microprocessor turns the fault indicator off. The logic self-tester runs only at powerup.

4.3.1.3 Program Loader – The program loader is a group of preprogrammed instructions called by the microprocessor to load the microdiagnostic monitor from the tape unit into the RDM's memory. (The role of the microdiagnostic monitor is discussed in Paragraph 4.3.2.2.) There is no hard connection between the tape unit and RDM. The RDM has no knowledge of VAX's use of the tape unit. The RDM has a logical connection to the tape unit that the microprocessor uses to gain control of the tape unit from the main processor.

4.3.2 Microprocessor

The microprocessor (an 8085) executes instructions in the preprogrammed memory and microdiagnostic monitor. Since the RDM has its own processor, preprogrammed memory, and stored program capacity, the microprocessor does not depend on the VAX processor to run tests. Only the VAX clock must be operating.

4.3.2.1 Data Transfer Paths – The microprocessor uses a data transfer line to read and write to two important areas in the VAX processor: W-BUS and main memory. The W-BUS transfers information between the VAX-11/750's arithmetic logic unit (ALU) and general registers. Main memory stores programs and data to be used later. The RDM tests for problems by writing and then reading known data to these areas. (The diagnostic module's WD register and MD register hold data to be written to the W-BUS and memory, respectively. The module's WH register and MH register hold data read from the W-BUS and memory, respectively. The module's MA register holds the address of memory that has been read from or written to.)

The VAX front panel console is another area in the VAX-11/750 directly accessed by the microprocessor. The DRDC can therefore check with the RDM to read the front panel indicators or determine the keyswitch position.

4.3.2.2 Microdiagnostic Monitor – The microdiagnostic monitor is the RDM's nonresident operating system. It directs microdiagnosis of the VAX-11/750. To load the monitor, the operator enters the Test command. The program loader instructs the microprocessor to load the monitor from the tape unit into an area of random access memory (RAM).

The Test command switches the RDM to the microdiagnostic state. (Before this happens, the keyswitch and software switch connections must be made, so that the system is in RDM console state.) Once in the microdiagnostic state, commands entered on the active terminal address the diagnostic monitor which controls the microprocessor.

4.3.3 Diagnostic Control Store

The following paragraphs discuss the function of the diagnostic control store and its associated hardware.

4.3.3.1 Diagnostic Program Storage – The diagnostic control store (DCS) is the area of memory where the microprocessor loads specific diagnostic routines. Once loaded, these diagnostic programs drive the VAX-11/750, in effect replacing the VAX control store. The microdiagnostic monitor loads the DCS and initiates driving of the VAX processor.

4.3.3.2 Trace Storage – Trace is another function that occupies the DCS. The trace function continually records the 64 most recently executed control store addresses, as well as the address of the current microinstruction for the VAX to execute. Trace storage shares part of the memory used for storing diagnostic programs, so the two functions cannot be performed at the same time.

If the VAX processor stops, tracing can determine why. The microprocessor can relay control store addresses stored in memory to the terminal for analysis.

In trace mode, the RDM continually and automatically clocks CS addresses from the CS address lines into a trap register, and from there into 64 DCS RAM locations. The RDM is in trace mode whenever it is not in the microdiagnostic state.

When the RDM loads a new CS address into the trap register, the previous CS address in the trap register moves into a DCS RAM location. The value of a DCS pointer determines the DCS RAM location to which the CS address moves. Every time an address moves from the trap register to RAM, the pointer increments by one. The RDM loads the trap register when the CPU latches a word.

The DCS pointer has a capacity of 64. After 64 words have been loaded into RAM, the pointer increments to the location loaded first. The DCS RAM, therefore, is a circular buffer in which new CS addresses from the trap register overwrite the oldest information.

Chapter 2 describes the Trace command. That command displays a readout of the DCS. Reading out of the DCS is the reverse process of writing into the DCS. The microprocessor reads CS addresses out of the DCS and forwards them to the terminal for display.

The first address read is the address in the trap register, which is always the latest CS address. The next address is read from the current RAM location specified by the pointer. After each DCS location is read, the pointer decrements by one, until all locations have been read.

4.3.3.3 Match Register – In parallel with the DCS, a match register also monitors VAX CS address lines. The match register has four functions.

1. Stop-on-micromatch
2. Generating a known test pulse
3. Gaining control of the VAX control lines for the DCS
4. VAX control store parity checking

Stop-on-micromatch is a ROM-encoded function that loads the match register with a specific control store address. It then compares the address with that of each control store instruction executed. When the address of the current control store instruction matches the selected address, the match logic stops the processor. Primarily, stop-on-micromatch permits you to trace the path the processor took to a particular microinstruction.

When the address in the match register matches the address on the CS address lines, a pulse generates on pin C0681 of the CPU backplane and pin TP1 on the front of the RDM. By deliberately making the CPU cycle on a matching address, the match signal can be used as a known pulse to trigger test equipment.

In addition, the match register helps the DCS gain control of the VAX processor. Under microdiagnostic monitor control, the microprocessor loads the match register with a DCS address. The match register places this DCS address on the control store address lines to force control of the VAX to the DCS. Once the DCS has control, the particular microdiagnostic in the DCS drives the VAX processor control lines. The microdiagnostic is then responsible for making the CPU continue to select DCS as a source of instructions.

The match register also helps to check parity in the VAX control store. The microprocessor can make the match register increment through successive control store addresses. At each location, the VAX-11/750's own parity check logic is enabled to perform a parity check. The match register addresses each control store location until the register increments to an address in the CPU control store that holds a deliberately set parity error. (If the parity check uncovers an accidental parity error, the match register does not increment past that address.) Detecting the known parity error is a test of the VAX-11/750 parity check logic.

NOTE

When parity checking the writable control store (WCS), first load the WCS with data. WCS addresses are 2000 to 23FF.

CHAPTER 5
RUNNING MICRODIAGNOSTICS

5.1 SCOPE

This chapter and the next three describe how to use Remote Diagnostic Module (RDM) software to test the VAX-11/750 and the RDM. This software resides, in external storage, on TU58 tape. It is brought into RDM memory by the Test and Test/C commands introduced in Chapter 2.

Chapter 5 will cover specifically the microdiagnostic monitor (ECKAA.EXE) and test programs ECKAB.EXE, ECKAC.EXE, and ECKAF.EXE. These programs (or "overlays") test the Data Path Module, Memory Interconnect Module, and the RDM.

The microdiagnostic programs have the capability to test almost 90 percent of the CPU hardware. The microdiagnostic monitor also provides error reports of gate arrays to be replaced for specific failures. The maintenance strategy is to replace the string of gate arrays suggested by the error report, and if that procedure fails, replace the module.

It should be noted that the microdiagnostics are repair level tools. The operator must have a thorough understanding of the processor and the microdiagnostic programming language to fully appreciate the capability of these tools. The microdiagnostics can be used to repair the system to the component level, if time permits the engineer to set up the necessary test equipment to do so.

When the microdiagnostic program starts, a ROM-based TU58 driver routine loads the microdiagnostic monitor from the TU58 tape cartridge into 8085 RAM memory. The monitor remains in the 8085 memory for the duration of the test session.

Once the monitor is loaded, the program initialization routine begins. When several passes are run, this routine executes at the beginning of each. The program initialization routine tracks program flow, initializes parameters, and invokes the TU58 driver routine to load the first test overlay for execution. Normally, the last instruction in each test overlay invokes the TU58 driver routine to load the next test overlay. The test overlays are read into memory one at a time.

Test overlays generally consist of pseudo-instructions, test data, and microinstructions to execute from the diagnostic control store (DCS). The monitor reads the pseudo-instructions and invokes appropriate routines. These routines implement the required functions in 8085 code. The pseudo-instructions set up and control the execution of microinstructions in DCS.

The monitor may be used to set and clear flags, and issue other commands to control the program flow with precision.

5.2 RUNNING MICRODIAGNOSTIC TESTS

DIGITAL distributes the VAX-11/750 microdiagnostics on a series of TU58 tape cartridges. Each tape cartridge contains the microdiagnostic monitor and a number of test overlays. The microdiagnostic monitor is identical on all tape cartridges, but the test overlays are unique. Each tape cartridge tests a portion of the CPU or memory. Normally, you should run the tapes in the following order: DPM (ECKAB), MIC (ECKAC).

To run microdiagnostics, perform the following procedure.

1. Insert the desired TU58 cartridge into the slot on the CPU front panel.

2. Turn the POWER ON ACTION switch to HALT.

3. Turn the keyswitch to LOCAL to power up the CPU. If the console terminal executes microverify successfully, it should type two percent signs, and then the console prompt >>>.

4. Type CTRL/D to invoke the RDM console command mode. The console terminal should respond with the RDM prompt RDM>.

5. Type TE to run the microdiagnostic tests. If the test sequence on the tape cartridge runs to completion without error, the program types a series of messages on the console terminal confirming completion. The following example shows a typical printout with no errors.

Sample Dialogue

```
RDM>TE<RET>                                  ! Test command

ECKAA-V0.22 MIC (L0003)-V00.04               ! Program
                                               identification
01,02,03,04,05,06,07,08,09,0A,0B,0C,0D,      ! Test numbers
                                               typed
0E,0F,10,11,12,13,14,15,16,17,18,19,1A,      ! at execution
                                               time
1B,1C,1D,1E,1F,20,21,22,23,                  ! Number of tests
END OF PASS 01                               ! varies depending
                                             ! on revision of
                                               microdiagnostics

MIC>                                         ! Microdiagnostic
                                               monitor prompt
```

6. Replace the tape cartridge with the next microdiagnostic tape and type DI.

7. Type RE<CR> after the MICMON prompt to return to the RDM console command mode.

8. Type RET<CR> after the RDM command mode prompt to return to the VAX console state.

5.3 MICRODIAGNOSTIC MESSAGES

A microdiagnostic prints messages on the console terminal to indicate the following items and functions.

1. Monitor name, version, and module under test
2. Test number to execute next (current test)
3. Error messages

If the program detects an error, it halts test execution, types an error message, and passes control to the microdiagnostic monitor. Refer to the following example.

Sample Dialogue

```
RDM>TE<RET>

ECKAA-V0.22 MIC (L0003)-V00.04          ! Program identification
01,                                     ! Test number
?ERROR: 0030 TEST: 01 SUBTEST: 01

DATA:   AAAAAAAA
        AAAA8AAA                        ! MODULE NAME is listed
        00000001                        ! only if it is not the
                                        ! module being tested

FAILING GATE ARRAYS: ADK,

MIC>                                    ! Microdiagnostic monitor
                                        ! prompt
```

The only test number printed is 01, indicating that the program started test 01 but did not reach test 02. The next line in the message confirms the location of the failure in test 01.

```
?ERROR: 0030 TEST: 01 SUBTEST: 01
```

The 0030 entry indicates the test program counter (PC) value of the failing pseudo-instruction. The data patterns indicate the expected and received data and the loop count. If the program identifies failing gate arrays, they are listed at this point. The message then lists the names of other modules that may be causing the failure.

In most cases, the machine should be repaired by turning off power, replacing the module or gate arrays called out, and rerunning the microdiagnostic program. If the message calls out several modules, replace them one at a time in the order listed and run the program after each replacement until the fault is repaired.

Replacing the parts called out may not repair the fault. Also, spare modules may not be available. In such cases, you may want to examine the program listing for the failing test. You may then use microdiagnostic monitor commands to set up scope loops and step through the failing test. The following chapters cover these topics.

CHAPTER 6
MICRODIAGNOSTIC MONITOR COMMANDS

6.1 SCOPE

Several commands control the microdiagnostic monitor. The monitor may be entered either directly or indirectly. To enter directly from the RDM console command mode, type TE/C (Test command) instead of TE (Test). To enter directly during test execution, type CTRL/C. The microdiagnostic program enters the monitor indirectly (automatically) if it finds either a TU58 driver error or a hardware error when the halt on error flag is set. When the monitor is waiting for a command, it prompts with MIC>.

The list below contains legal command op codes.

Diagnose
Loop
Set
Clear
Show
Return
Continue

The list below contains legal command keyword arguments.

TEST
VBUS
LOOP
STEP
PASS
NER (no error report)
CYCLE (one microcycle)
BELL
TICK (one-half microcycle)
FLAG
HALT
SA (signature analyzer)
SOMM (stop-on-micromatch)
INSTRUCTION
CONTINUE
CF (control file)
IB (inhibit burst)
QA (quality assurance)
TR (trace flag)
QV (quick verify)
DM (diagnostic module)

VA (virtual address)
MA (memory address)
PC (program counter)
PB (PC backup)
RT (RTemps)
MT (MTemps)
PS (process status longword)
MD (memory data)
WD (write data)
SR (status and control registers)
SF (status flags)
ST (step counter)

In the following command descriptions, an exclamation point means exclusive OR. In sample dialogues and examples, an exclamation point indicates that what follows on that line is a comment.

6.2 DIAGNOSE

Syntax: DI [TE:<test-number> [[<test-number>] ! [CO]] ! QV
PA:<pass-count>!DM]

This command takes three types of arguments. It initializes the program control flags and starts program execution.

Flag	Initial Setting
LOOP	clear
NER	clear
BELL	clear
HALT	set
IB	clear
SA	clear
QA	clear
TR	no change

If Diagnose is typed without any arguments (switches), the monitor executes all tests on the installed tape once. Control then returns to the RDM console command mode, if the program detects no errors.

Sample Dialogue

```
MIC>DI<RET>                                    ! Diagnose command

ECKAA-V0.21 MIC (L0003)-V00.04

01,02,03,04,05,06,07,08,09,0A,0B,0C,0D,0E,0F,10,11,12,13,
14,15,16,17,18,19,1A,1B,1C,1D,1E,1F,20,21,22,23,
END OF PASS 01
MIC>
```

6.2.1 Diagnose Test

The TEST argument may be used with the Diagnose command in three ways.

1. Diagnose TEST followed by a single test number executes the specified test indefinitely. Type CTRL/C to escape from the loop and return to the monitor.

2. Diagnose TEST followed by a test number and Continue executes the specified test and all following tests on the installed tape cartridge.

3. Diagnose TEST followed by two test numbers executes all tests between and including the two numbers specified. Control then returns to the monitor. If the first test number typed is out of range, the program types an error message and returns control to the monitor.

Sample Dialogue 1

```
MIC>DI TE:2<RET>                    ! Test 2 loops
                                    ! indefinitely.
ECKAA-V0.21 MIC (L0003)-V00.04      ! Monitor identifies itself.
                                    ! Test runs, looping
CTRL/C                              ! continuously.
02↑C                                ! Test number is 2.
                                    ! Use CTRL/C to escape.
MIC>                                ! This is the microdiagnostic
                                    ! monitor prompt.
```

Sample Dialogue 2

```
MIC>DI TE:2 CO<RET>                 ! Start with test 2 and
                                    ! run remaining tests
                                    ! on tape cartridge.
ECKAA-V0.21 MIC (L0003)-V00.04      ! Monitor identifies itself.
02,03,04,05,06,07,08,09,            ! Each test number is
0A,0B,0C,0D,0E,0F,10,11,12,13,      ! typed as that test starts.
14,15,16,17,18,19,1A,1B,1C,
1D,1E,1F,20,21,22,23,
END OF PASS 01
MIC>                                ! This is the microdiagnostic
                                    ! monitor prompt.
```

Sample Dialogue 3

```
MIC>DI TE:3 6<RET>                  ! Execute tests 3,4,5,6.
ECKAA-V0.21 MIC (L0003)-V00.04      ! Program identifies itself.
03,04,05,06,                        ! Each test number is typed
                                    ! as that test starts.
MIC>                                ! This is the microdiagnostic
                                    ! monitor prompt.
```

6.2.2 Diagnose Pass

The PASS argument may be used with the Diagnose command to set the pass count to a specified number. The default pass count is 1. A pass count of 0 causes the program to run indefinitely. At the end of the specified number of passes, control returns to MICMON.

Sample Dialogue

```
MIC>DI PA:2<RET>

ECKAA-V0.21 MIC (L0003)-V00.04

01,02,03,04,05,06,07,08,09,0A,0B,0C,0D,0E,0F,10,11,12,13,
14,15,16,17,18,19,1A,1B,1C,1D,1E,1F,20,21,22,23,
END OF PASS 01

01,02,03,04,05,06,07,08,09,0A,0B,0C,0D,0E,0F,10,11,12,13,
14,15,16,17,18,19,1A,1B,1C,1D,1E,1F,20,21,22,23,
END OF PASS 02
MIC>
```

6.2.3 Diagnose Quick Verify

The QV (quick verify) argument may be used with the Diagnose command to run a predetermined subset of tests on the microdiagnostic tape. It was designed for use by DIGITAL manufacturing groups.

Sample Dialogue

```
MIC>DI QV<RET>

ECKAA-V0.21 MIC (L0003)-V00.04
26,27,28,....
END OF PASS 01
MIC>
```

6.2.4 Diagnose Diagnostic Module

The DM (diagnostic module) argument may be used with the Diagnose command to run those tests that check out the diagnostic module only. This command should be used only if the VAX-11/750 is working properly.

Sample Dialogue

```
MIC> DI DM<RET>
ECKAA-V0.21 RDM (L0006)-V00.04
01,02,....
END OF PASS 01

MIC>
```

6.3 EXAMINE

Syntax: EX <NAME_LIST>[:<REG_NUMBER>]

This command examines various registers and flags within the 750 CPU.

NAME_LIST	REG_NUMBER		
VA		!	virtual address register
MA		!	memory address register
PC		!	program counter
PB		!	PC backup register
RT	0→2F	!	RTemp registers
MT	0→0F	!	MTemp registers
PS		!	processor status longword
MD		!	memory data register
WD		!	write data register
SR	0→E	!	status & control registers
SF		!	status flags
ST		!	step counter

6.4 SHOW FLAGS

Syntax: SH FL

This command displays the current states of program control flags listed in Table 6-1.

Table 6-1 Program Control Flags

Flag	Description
HALT	Halt – This flag calls the monitor when error is detected and error message has been typed.
LOOP	Loop on error – This flag is useful only in the event of a program-detected error. Flag may be set with the Loop command, or manually by typing SE FL LO. When flag is set and program detects an error, test loops on minimum amount of code necessary to recreate the error. The loop executed may include pseudo-instructions (8085 operations) and DCS microinstructions, running from ERRLOOP instruction to IFERROR instruction in failing test. Or loop may include DCS microinstructions only. If IB flag is set or a microinstruction trap occurs, program does not loop on microinstructions in DCS. Refer to Paragraph 7.17 (BRSTCLK) and Paragraph 6.15 (Loop command). Error messages are not inhibited while loop is executing unless NER flag is set. Type CTRL/C to escape from loop and return to monitor.
NER	No error report – If flag is set, program does not report errors.
BELL	Bell on error – If flag is set, program rings terminal bell on first occurrence of an error and on every fifth occurrence.
SA	Signature analysis – If flag is set, program loops on test in progress, between BEGINSA and ENDSA pseudo-instructions. Loop occurs whether or not the test detects errors. Set SA flag when using a signature analyzer to help diagnose faults. This flag provides two sync points on the backplane: a start/stop window (slot 6, pin C75) and a clock pulse (slot 6, pin C73). With test looping and signature analyzer connected to sync points, signature analyzer analyzes any test point that probe samples. Signature analyzer displays a value (signature) if signal pattern is steady. Compare this value with corresponding value from a known good module to locate failures.
QA	Quality assurance – If flag is set, program responds to each test as if it had detected a failure.
IB	Inhibit burst flag – If this flag and the loop flag are set, the program does not try to loop on DCS microcode only. Rather, it loops between ERRLOOP and IFERROR pseudo-instructions.
TR	Trace flag – If flag is set, monitor types test names as well as test numbers.

Sample Dialogue

```
MIC>SH FL<RET>                          ! Show flags

FLAGS SET: LO, NE, BE,
FLAGS CLEAR: HA, SA, IB, QA, TR

MIC>
```

6.5 SET FLAG

Syntax: SE FL <flag-name-list>

This command sets (enables) any of the program control flags.

Sample Dialogue

```
MIC>SE FL HA                    ! Set HALT flag
MIC>SE FL LO                    ! Set LOOP flag
MIC>SE FL NE                    ! Set NO ERROR
                                ! REPORT flag
MIC>SE FL BE                    ! Set BELL flag
MIC>SE FL SA                    ! Set SIGNATURE
                                ! ANALYSIS flag
MIC>SE FL QA                    ! Set QUALITY
MIC>SE FL QA:<loop-count>       ! Issues error messages on
                                ! loop counts greater than or
                                ! equal to <loop-count>
                                ! ASSURANCE flag
MIC>SE FL IB                    ! Set INHIBIT BURST flag
MIC>SE FL LO NE BE              ! Set LOOP, NO ERROR, and BELL
                                ! flags
```

6.6 CLEAR FLAG

Syntax: CL FL <flag-name-list>

This command clears (disables) any of the program control flags.

Sample Dialogue

```
MIC>CL FL HA<RET>               ! Clear HALT flag
MIC>
```

6.7 SET STOP-ON-MICROMATCH

Syntax: SE SO:<cs-address>

This command stops execution of code in DCS at the specified address. DCS addresses range from 0 to 3F. Add 1800 to the desired DCS address. Type the Continue command after the MIC> prompt to proceed from your current halt, after typing SE SO <CS-address> to get to that address.

Sample Dialogue

```
MIC>SE SO:1803<RET>             ! Stop at DCS address 3
MIC>CO<RET>                     ! Continue
MICRO BREAK MATCH, CS ADDR=1803, DCS ADDR=03.
MIC>
```

The function of the Set Stop-On-Micromatch command may be deceiving at times. Suppose you want to stop microcode execution at a location that follows a microinstruction specifying the NEXT field. Assume that the following code is now in the DCS.

```
DCS_DATA

;% 00      NOP
;% 01      MEMSCAR_R[TEMP0]                          ; Physical/Virtual
                                                     ; address to
                                                     ; MEMSCAR
;% 02      MEMSCR_R[TEMP1]                           ; Set MME off
;% 03      MEMSCAR_R[TEMP2]                          ; Reference cache
                                                     ; only to MEMSCAR
;% 04      MEMSCR_R[TEMP3]                           ; Turn CMI off
;% 05      PSL(PREV_CURM ISCURM_R[TEMP4])            ; Set mode to
                                                     ; executive
;% 06      STROBE.V.BUS,WB_RDM,PTE CHECK READ?,NEXT/1800
                                                     ; Perform PTE check
;% 07      NOP,STOP.CLOCK.LATCH.CS.ADDR              ; Stop CPU clock
```

If this code was in the DCS, typing SE SO:1807 would not access DCS address 7. Instead, to get to DCS address 7, stop-on-micromatch must be set to stop at the address specified by the NEXT field in DCS address 6 (MIC>SE S0:1800). This is because Set Stop-on-Micromatch compares the control store address specified in the command with the address asserted on the NEXT control store lines. The stop always occurs at the next DCS address after the address whose NEXT field contains the value specified in the command.

```
MIC>SE SO:1800<RET>                    ; Stop at DCS address 7
MIC:CO<RET>                            ; Continue
MICRO BREAK MATCH, CS ADDR = 1800, DCS ADDR = 07
MIC>
```

The NEXT address field never affects the DCS microcode. When DCS is active, the NEXT address field merely selects the DCS range of addresses (1800 to 183F). DCS address control resides in a counter. The counter can only increment or return to 0. A microdiagnostic test specifies the NEXT field to test the microaddress generation function, not to create a microcode jump.

6.8 CLEAR STOP-ON-MICROMATCH

Syntax: CL SO[:<cs-address>]

This command clears the stop-on-micromatch function. Add 1800 to the desired DCS address to create a scope sync pulse at that address. If <cs-address> is specified, a scope sync pulse is generated on slot 6, pin C81 when the current address matches <cs-address>. The pulse occurs with the M clock that marks the beginning of the specified microcycle.

Sample Dialogue

```
MIC>CL SO:1803<RET>          ! Clear the stop-on-micromatch
                             ! function
                             ! Load 1803 into the RDM match
                             ! register for a scope sync
MIC>CO<RET>                  ! Continue
```

6.9 SET CONTROL FILE

Syntax: SE CF:<dcs-address> <bit-number>

This command sets the specified <bit-number> in the control file of the specified <dcs-address>. (Refer to Paragraph 8.6.) Table 6-2 lists the control file bit functions and bit numbers.

Sample Dialogue

```
MIC>SE CF:3 4<RET>          ! Set control file bit
                            ! 4 (HRWBUS) in
                            ! DCS address 3
MIC>
```

6.10 CLEAR CONTROL FILE

Syntax: CL CF: <dcs-address> <bit-number>

This command clears the specified bits in the control file of the <dcs-address>.

Sample Dialogue

```
MIC>CL CF:3 4<RET>          ! Clear control file bit
                            ! 4 (HRWBUS) in
                            ! DCS address 3
```

Table 6-2 Control File Bit Functions

Bit	Mnemonic	Function
0	VSTB	This bit strobes visibility bus at rising edge of next M clock.
1	DCSACL	This bit clears dcs address register.
2	STPCLK	This bit stops VAX-11/750 CPU clock.
3	ENTRACE	This bit latches trace register at rising edge of the next M clock. When this bit is not set, trace register monitors control store address lines continuously.
4	HRWBUS	This bit reads W-BUS by strobing WH register at rising edge of the next M clock.
5	WBUSDR	This bit enables WD register onto W-BUS (wri
6	STBCMI	This bit reads CMI data by strobing MH regist edge of next M clock.
7	SATRIG	This bit inhibits clocking signature analyzer on connects to a post on module for use by signatu C73 on slot 6.)

6.11 SET STEP INSTRUCTION

Syntax: SE ST IN[:<test-pc>]

This command steps through the pseudo-instructions (8085 code) in the current test. If <test-pc> is specified, the step function does not start until the instruction at the <test-pc> address is ready to execute. If <test-pc> is not specified, the step function begins at the next pseudo-instruction of the current test, following a Loop or Continue command.

In either case, when the program stops at the next pseudo-instruction, press the space bar to continue stepping through the pseudo-instructions. Type a carriage return to escape from the step mode back to the monitor.

Sample Dialogue 1

```
MIC>SE ST IN<RET>                    ! Set step instruction
MIC>LO<RET>                          ! Start looping
  TPC = 001F                         ! Test PC 1F
                                     ! Space bar
  TPC = 0024
  <RET>                              ! Carriage return
MIC>
.
.
.
```

Sample Dialogue 2

```
MIC>SE ST IN:2C<RET>                 ! Set step instruction
                                     ! starting at 2C
MIC>CO<RET>                          ! Continue
  TPC = 002C                         ! Test PC is 2C
                                     ! Space bar
  TPC = 0038
                                     ! Space bar
  TPC = 003E
  <RET>                              ! Carriage return
MIC>
```

6.12 SET STEP CYCLE

Syntax: SE ST CY

This command steps through DCS microinstructions one CPU machine cycle at a time (M clock).

After entering in this command, the Continue, Loop, or Diagnose commands must be typed to begin stepping. After the program stops at the first DCS address in the current test, press the space bar to step through the test. Type a carriage return to escape and return control to the monitor.

CHAPTER 7
PSEUDO-INSTRUCTIONS

7.1 SCOPE

In VAX-11/750 microdiagnostic programs, the monitor interprets and implements pseudo-instructions. Pseudo-instructions consist of many 8085 instructions. This chapter describes the pseudo-instructions used to diagnose the VAX-11/750.

In the pseudo-instructions described below, an unspecified or byte data type equals eight bits. A word data type is 16 bits, and a long data type equals 32 bits. A microword equals 11 bytes.

7.2 INITIALIZE

Syntax: INITIALIZE

This pseudo-instruction initializes the VAX-11/750 CPU as follows.

1. Cycles DC LO
2. Asserts PROCINIT (processor initialize)
3. Disables CMI
4. Disables memory management
5. Loads PSL with zeros

7.3 LOAD DCS

Syntax: LOADDCS <src-addr>, [src-index], <dcs-addr>, <wrd-cnt>

This pseudo-instruction moves <wrd-cnt> microwords from the 8085 memory buffer to DCS, starting at <dcs-addr>. <src-addr>, indexed by [src-index], specifies the starting 8085 memory address.

7.4 LOAD FIELD

Syntax: LDFIELD <src-addr>,[src-index],[data-type], <dst-addr>, {<start-bit>, /<address-of-start-bit-table>} {<end-bit>, /<address-of-end-bit-table>} [longlit], [start-bit-ind], [end-bit-ind], [no-parity], [NS]

This pseudo-instruction acts only on RDM 8085 memory. It is normally used before the LOADDCS pseudo-instruction to set up a microword before moving it into DCS. LDFIELD moves a field of length (<end-bit>-<start bit>) indexed by [SRC index] (times [data-type], to a field in <dst-addr> starting at <start-bit>.

If <address-of-start-bit-table> or <address-of-end-bit-table> is specified, the field is specified by the contents of the associated table. If [start-bit-ind] or [end-bit-ind] is specified, the respective table address is indexed by bytes. If [longlit] is specified, data is inverted before being placed in the LONLIT field. If you specify [no-parity], the monitor does not calculate parity on that microword. Use the optional [NS] argument to load data unscrambled into the 8085 memory. [NS] moves data other than microwords, which must be scrambled when loaded into the DCS.

Example

```
        .
        .
        .
LDFIELD         2$,,B,1$,0,7,
LOADDCS         1$,,0,1
1$:
;% NOP,NEXT/0                    ; Microinstruction.
2$:     .BYTE       ↑X3F
```

In the previous example, the LDFIELD instruction moves 3F at 2$, to the microinstruction stored in 8085 memory at 1$. The 3F is inserted from bit 0 to bit 7. The microinstruction at 1$ becomes NOP,NEXT/3F. LOADDCS then moves this microinstruction to DCS address 0. The next subtest may use the same microinstruction, but place a different value in any field with the LDFIELD pseudo-instruction.

7.5 LOAD REGISTER

Syntax: LOADREG <reg-addr>,<src-addr>,[src-index],[data-type]

This pseudo-instruction loads a RDM register from the 8085 memory. LOADREG moves the contents of <src-addr>, indexed by [src-index], to the RDM register specified by <reg-addr>.

Example

```
LOADREG         WDREG,1$,I,L

1$:             .LONG       ↑XFFFFFFFF
```

In the previous example, the LOADREG instruction loads 32 bits of the WD register with ones. Assume that the index I equals 1.

7.6 LOAD INDEX

Syntax: LDINDEX <index-nam>,{<index-value>/
<index-value-add>,[index]}

This pseudo-instruction initializes the specified index, <index-nam> (I,J, or K), to the specified value, <index-value>. The instruction can also initialize the <index-nam> to <index-value-add> times [index].

7.7 SAVE INDEX

Syntax: SAVEINDEX <index-nam>,<dst-addr>

This pseudo-instruction moves the current value of the specified index, <index-nam>, into <dst-addr> in 8085 memory.

7.8 INCREMENT INDEX

Syntax: INCINDEX <index-nam>

This pseudo-instruction increments the current value of <index-nam> (I,J, or K).

7.9 STUCK AT ZERO OR ONE PATTERN

Syntax: 1PAT <index-nam>,<dst-addr>,[longlit],[dst-addr-x]

This pseudo-instruction selects one of the following patterns each time it is executed.

P1 10101010101010101010101010101010 (B)
P2 01010101010101010101010101010101 (B)
P3 00110011001100110011001100110011 (B)
P4 00001111000011110000111100001111 (B)
P5 00000000111111110000000011111111 (B)
P6 00000000000000001111111111111111 (B)

The pseudo-instruction chooses a pattern according to the current value of <index-nam>. If [longlit] is not specified, it places the pattern in the test data region of the 8085 memory at <dst-addr>. An <index-nam> with a value of 1 selects pattern P1, and so on. The current value of <index-nam> must be less than or equal to 6.

If [longlit] is specified, the pseudo-instruction places the pattern in bits <62:31> of the microword starting at <dst-addr>. If [dst-addr-x] is specified, the pattern is also placed at this address.

7.10 BEGIN SIGNATURE ANALYZER

Syntax: BEGINSA

If the SA flag is set, this pseudo-instruction causes the program to loop between the BEGINSA and ENDSA pseudo-instructions.

7.11 END SIGNATURE ANALYZER

Syntax: ENDSA

This pseudo-instruction checks the SA flag. It marks the end of the signature analyzer loop.

If the SA flag is clear, the program executes the next pseudo-instruction. (Refer to the example in Paragraph 8.3.)

If CTRL/C is typed in an SA loop, control returns to the monitor.

7.12 LOOP

Syntax: LOOP <loop-name>,<start-value>,<end-value>

This pseudo-instruction initializes loop parameters for a specified index <loop name> (I,J, or K). If the <start-value> is less than the <end-value>, the loop is an incrementing loop. If the <end-value> is less than the <start-value>, the loop is a decrementing loop. Loops of this type are used to repeat a test or subtest, using a different data pattern indexed by <loop-name> each time.

7.13 END LOOP

Syntax: ENDLOOP <loop-name>

This pseudo-instruction terminates a programmed loop. The program jumps to the pseudo-instruction following the corresponding LOOP pseudo-instruction, unless the required number of loops have been completed. A loop of this type functions whether or not an error has been detected.

7.14 ERROR LOOP

Syntax: ERRLOOP

This pseudo-instruction saves the test PC. The instruction works with the IFERROR pseudo-instruction. When the test detects an error and the LOOP flag is set, the program loops on the error.

7.15 IF ERROR

Syntax: IFERROR [data-cnt],[gate-array-list],[module-name]

This pseudo-instruction checks the results of one of the three compare pseudo-instructions: CMPREG, CMPREGMSK, and CMPVBUS.

Based on flag settings, the instruction executes a loop or continues with the program. The optional parameters specify the nature of the error messages. (Refer to the example in Paragraph 5.3.)

[Data-cnt] specifies the number of RTEMP registers to type in the error message. For example, a [data-cnt] of 4 specifies RTEMPs 0, 1, 2, 3. These registers contain test data as described in the error description in the test listings.

7.16 FETCH

Syntax: FETCH <dcs-addr>,[MIC-ADR-INH]

This pseudo-instruction initiates a DCS microinstruction sequence. It loads the microword stored at <dcs-addr> into the control store latches. If [MIC-ADR-INH] is specified, the instruction leaves the Micro Address Inhibit signal asserted, after execution of the instruction, to inhibit the processor from driving the control store lines.

7.17 BURST CLOCK

Syntax: BRSTCLK <stop-addr>, [INHIBIT]

This pseudo-instruction bursts the VAX-11/750 CPU clock under normal circumstances. It checks to see that the microcode stopped at <stop-addr>. Control then passes to the next pseudo-instruction.

If the following conditions exist, the instruction performs the functions set control file bit 1 (DCSACL) and clear control file bit 2 (STPCLK) at <stop-addr>.

1. Error occurs
2. LOOP flag set
3. <INHIBIT> argument not specified
4. SA flag clear

In this way, BRSTCLK creates a DCS loop. Otherwise, control passes to the next pseudo-instruction.

BRSTCLK also controls the step cycle, step tick, and stop-on-micromatch functions. (Refer to the Loop command description in Paragraph 6.15 and the LOOP flag description in Paragraph 6.3 for details of the looping function.)

7.18 EXPECT MICROTRAP

Syntax: EXPUTRAP

This pseudo-instruction enables the BRSTCLK function to recognize a microtrap. If a microtrap occurs as expected, the burst clock routine does not report an error and passes control to the next pseudo-instruction.

If a microtrap occurs, the control store address lines are forced outside the range of DCS address space, 1800 to 183F. The RDM monitors the control store address lines, stopping the CPU clock when they are forced outside the range of DCS.

If an EXPUTRAP pseudo-instruction precedes a microtrap, the program handles the trap. Otherwise, the program prints an unexpected clock stop error message on the console terminal.

If a microtrap occurs, whether or not it is expected, you cannot create a DCS loop.

7.19 COMPARE REGISTER

Syntax: CMPREG <reg-addr>,<dst-addr>,[dst-index], [data-type], [mode-type]

This pseudo-instruction compares the contents of a RDM register <reg-addr> with <dst-addr>, indexed by [dst-index] in DCS.

If the [mode-type] is blank, the CMPREG pseudo-instruction expects the data to be equal. If the [mode-type] is NE, it expects the data to be unequal.

On a comparison failure, the IFERROR pseudo-instruction following CMPREG prints an error message.

7.20 COMPARE REGISTER MASKED

Syntax: CMPREGMSK
<reg-addr>,<dst-addr>,[dst-index],<msk-addr>, [msk-index],[data-type],[mode-type]

This pseudo-instruction works exactly like CMPREG, except that the comparison is masked by <mask-addr> and indexed by [mask-index]. Only bit positions where the mask equals 1 are compared, as shown in the following example.

Register Contents	1 1 1 1 1 1 1 0
Destination Contents	0 1 1 0 1 1 1 0
Mask	0 1 1 1 0 0 0 0

Only these three bit positions are compared.

7.21 MASK

Syntax: MASK
<src-addr>,[src-index],<msk-addr>, [msk-index], [data-type]

This pseudo-instruction masks bits in an 8085 memory location. Bits that are ones in the mask are preserved in the location at <src-addr>.

7.22 COMPARE VISIBILITY BUS

Syntax: CMPVBUS <data-table-ptr>,[index]

This pseudo-instruction compares the values of bits on the visibility bus (VBUS) with the expected values specified in the data table. If the comparison fails, the IFERROR pseudo-instruction prints an error message.

Example:

```
        BRSTCLK   7                          !  End DCS program
        CMPVBUS   2$,I                       !  Compare VBUS results
        IFERROR   ,<<ACV><ADK><UTR>>,        !  Possible failing
                                                arrays

2$:     .BYTE     ↑X4                        !  VBus data table
        VBUSDATA  <↑X08>,0                   !  Iteration 1
        VBUSDATA  <↑X09>,0
        VBUSDATA  <↑X0A>,1
        VBUSDATA  <↑X0B>,1
```

In the above example the VBUS data is compared with data in the table at 2$. .BYTE ↑X4 at the top of the table indicates that the comparison involves four bits. VBUSDATA <↑X08>, 0 indicates that bit 8 will be the first bit position compared, and the expected value is 0.

7.23 NEW TEST

Syntax: NEWTEST <title>, [section], [alt-test-number]

NEWTEST is always the first pseudo-instruction in a test. It defines the beginning of the test and performs a variety of housekeeping functions.

7.24 SUBTEST

Syntax: SUBTEST [subtest number]

This pseudo-instruction increments the current subtest number or takes the value given in the optional argument.

7.25 SKIP

Syntax: SKIP [address],[condition] [index_0] [index_1]

This pseudo-instruction causes the program to skip to [address]. If you do not specify [address], the program skips to the ENDTEST pseudo-instruction. If you specify [condition], it may be any of three types. If the [condition] is ONERROR, the program skips only when the error flag is set. If the [condition] is NOERROR, the program skips only when the error flag is not set. If the [condition] is ONEQUAL, [index_0] and [index_1] are also specified. The SKIP pseudo-instruction compares them and skips if they are equal.

7.26 END TEST

Syntax: ENDTEST

This pseudo-instruction terminates each test. It tests various flags and acts accordingly.

If you type in DI TE:<test-number>, the monitor loops on the test. Otherwise, the instruction continues the program by loading the next test overlay from the TU58.

7.27 ERRLOG

Syntax: ERRLOG

This pseudo-instruction logs in 8085 memory up to 64 single-bit main memory errors.

7.28 DUMPLOG

This pseudo-instruction types out the contents of the error log.

CHAPTER 8
MICRODIAGNOSTIC PROGRAM LISTINGS

8.1 SCOPE

Microdiagnostic program error messages usually provide enough information to isolate and repair faults in the processor. Error messages list the failing test number and call out suspected components and modules. However, if the specified parts are changed and the same message prints after rerunning the program, the program listing provides a further troubleshooting resource. Using the microfiche card for the appropriate set of tests, locate the frame for the failing test through the index in the lower right microfiche frame.

8.2 TEST OVERLAY LISTING FORMAT

Each test overlay consists of a test and associated data. A test may consist of several subtests. Although test overlays vary slightly, they use the same general format.

```
Test title
Documentation
   Functional description
   Test algorithm
   Test patterns
   Logic description
   Assumptions
   Error description
Subtest 1
   Pseudo-instructions                ; comments
Subtest 2...
   Pseudo-instructions                ; comments
Test data
   8085 memory buffer data            ; comments
   DCS data                           ; comments
```

The monitor loads this directly into DCS.

```
   RTEMP data                         ; Data loaded beginning at RTEMP0
   Cache data                         ; Data loaded beginning at cache
                                      ; address 0
```

NOTE

Before analyzing the test code and data, read the documentation carefully. It gives an overall description of the test and many useful details.

8.3 TEST EXECUTION FLOW

The execution flow for each subtest in a test consists of three steps.

1. Pseudo-instructions set up test data and parameters. The 8085 processor performs these functions.

2. Microcode in DCS executes.

3. Pseudo-instructions check the results of microcode execution. If the check reveals no error, control passes to the next subtest or test. Flag settings, error conditions, and test structure cause the program to loop to the beginning of DCS, or to a loop point in the pseudo-instructions, as appropriate.

Consider the test overlay listing in the following example.

1. Functional Description

   This test consists of two subtests.

   Cache error register
   Cache control register

   These are read/write registers in the CAK chip.

2. Test Algorithm

   a. Modify DCS address 1 to select the appropriate temporary register for the subtest.

   b. Set up a loop to select test patterns.

   c. Load WD register (RDM) with test pattern.

   d. Execute DCS program.

      1. Load memory status and control address register (MEMSCAR) with address in temporary register.

      2. Write test pattern in WDREG (RDM) to memory status and control register (MEMSCR).

      3. Read test pattern back to WHREG (RDM).

   e. Compare results (expected and received).

   f. If no error, go to step 3 for remaining test patterns.

3. Test Patterns

Iteration	Test Pattern
1	A
2	5
3	3

4. Logic Description

This test checks only the S/C registers that reside in the CAK gate array. The registers are accessed via WBUS<27:24>. If on error, replacing the CAK does not work, try running the MEMSCAR test, which checks for a multiple addressing problem. If that fails, then one of the other gate arrays with S/C registers is causing trouble.

5. Assumptions

This test assumes that the W-BUS is operational.

6. Error Description

EXPECTED MEMSCR DATA
RECEIVED MEMSCR DATA
LOOP COUNT

```
0004  472  ;SUBTEST 1 Cache Error Register
0004  476  SUBTEST                          ;  Cache error reg
0006  477  INITIALIZE                       ;  Initialize CPU
0008  478  LOADDCS    4$,,01,1             ;  Modify DCS address 01
000F  479  BEGINSA                          ;  Signature analyzer loop
                                            ;  point
0011  480  LOOP       I,1,3                ;  Create test loop
0018  481  LOADREG    WDREG,1$,I,L         ;  Test pattern to WD reg
0020  482  ERRLOOP                         ;  Error loop
0022  483  FETCH      0                    ;  Start DCS program
0026  484  BRSTCLK    4                    ;  End DCS program
002A  485  CMPREGMSK  WHREG,1$,I,3$,,L,    ;  Compare results
0036  486  IFERROR    ,<<CAK>>,            ;  Possible failing array
003C  487  ENDLOOP    I                    ;  End test loop
003E  488  ENDSA                           ;  End signature analyzer
                                            ;  loop
0041  491  ;SUBTEST 2 Cache Control Register

0041  495  SUBTEST                          ;  Cache control reg
0043  496  INITIALIZE                       ;  Initialize CPU
0045  497  LOADDCS    2$,,01,1             ;  Change DCS address 01
004C  498  BEGINSA                          ;  Signature analyzer loop
                                            ;  point
004E  499  LOOP       I,1,3                ;  Create test loop
0055  500  LOADREG    WDREG,1$,I,L         ;  Test pattern to WD reg
005C  501  ERRLOOP                         ;  Error loop
005F  502  FETCH      0                    ;  Start DCS program
0063  503  BRSTCLK    4                    ;  End DCS program
0067  504  CMPREGMSK  WHREG,1$,I,3$,,L,    ;  Compare results
0073  505  IFERROR    ,<<CAK>>,            ;  Possible failing array
0079  506  ENDLOOP    I                    ;  End test loop
007C  507  ENDSA                           ;  End signature analyzer
                                            ;  loop
007E  508  SKIP                            ;  SKIP to line 568
```

```
0084   513                    BEGIN TEST DATA
                    ;-------------------------
0084   515   1$:       .LONG       ↑X0A000000      ;  Test pattern
0088   516             .LONG       ↑X05000000      ;  Test pattern
0086   517             .LONG       ↑X03000000      ;  Test pattern
008E   518   2$:
008E   519   ;%    01    MEMSCAR R[Temp1]          ;  Rtemp for subtest 2
009B   523   3$:        .LONG       ↑X0F000000      ;  Mask for CMPREGMSK
                                                   ;  pseudo op
009F   524   4$:
009F   525   ;%    01    MEMSCAR R[Temp0]          ;  Rtemp for subtest 1

00AA   532                    BEGIN_CPU_DATA

0002   534                    DCS_DATA

0001   536   ;%    00    NOP
000C   540   ;%    01    MEMSCAR R[TEMP0]          ;  Write add of S/C to
                                                   ;  MEMSCAR
0017   544   ;%    02    WB_RDM,WCTRL/             ;  Write test pattern to
                         MEMSCR_WB                 ;
MEMSCR
0022   548   ;%    03    RDM_WB,WCTRL/MEMSCR       ;  Read test pattern
                                                   ;  back to RDM
002D   552   ;%    04    NOP,STOP.CLOCK            ;  Stop CPU clock

0038   557                    RTEMP_DATA

0001   559          .LONG       ↑X04000000      ;  Cache error register
                                                   ;  address
0005   560             .LONG       ↑X06000000      ;  Cache control
                                                   ;  Register address

0009   562                    END CPU DATA
00A2   566                    END TEST DATA

                    ;-------------------------
00A2   568                    ENDTEST
```

Figure 8-1 illustrates the execution flow for the first subtest.

Note that FETCH gets control of the control store address lines. Then the microcode executes during BRSTCLK. The BRSTCLK pseudo- instruction checks that the CPU clock stops at the DCS address specified (4). BRSTCLK then passes control to the next pseudo- instruction.

8.4 MICROINSTRUCTION REFERENCES

For detailed information on the microassembler, refer to the *MICRO2 User's Guide Reference Manual* (AA-H531A-TE).

For accurate interpretation of micro-orders, refer to the machine definitions (micro-order definitions) and the macro definitions in the VAX-11/750 microcode listings. For special macros used only in microdiagnostics, see the microdiagnostic listings.

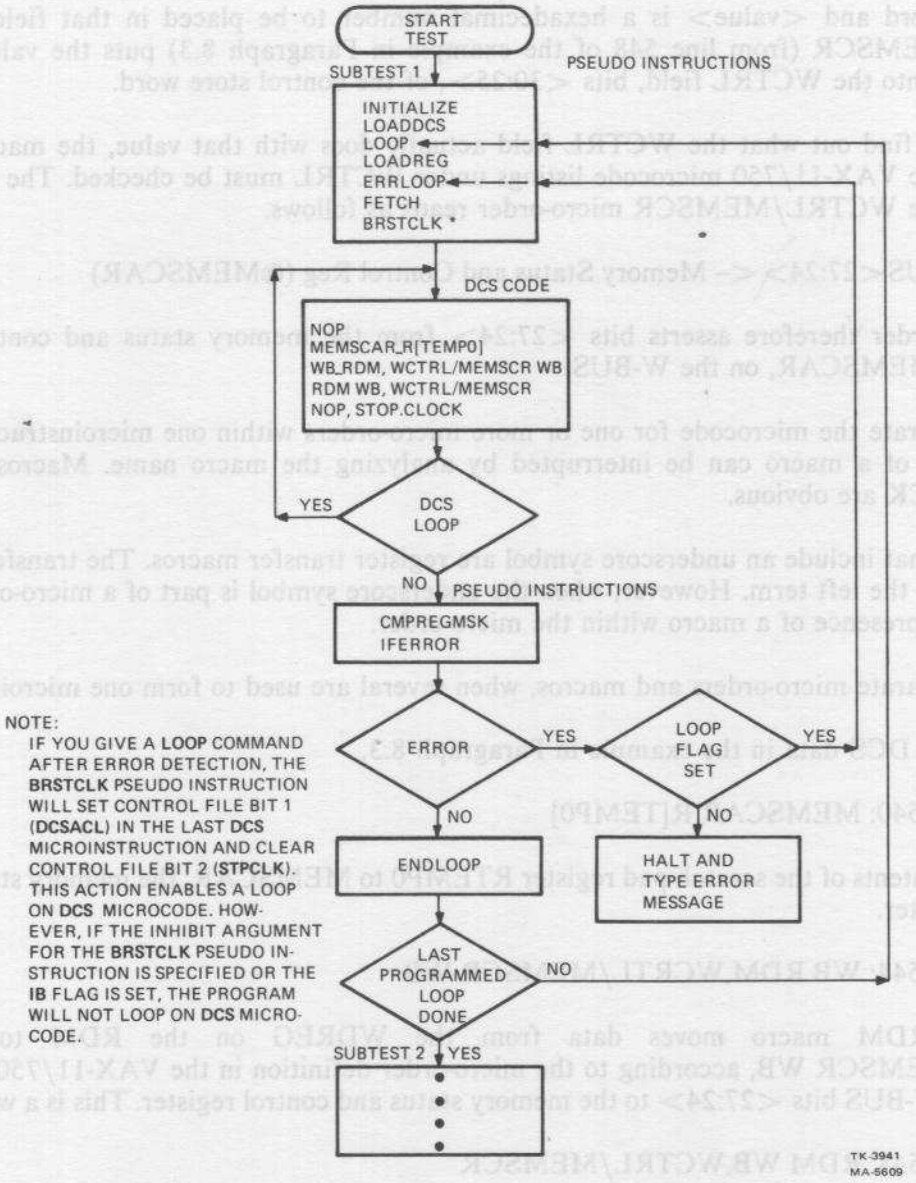


Figure 8-1 Sample Subtest Execution Flow

8.5 MICROINSTRUCTION INTERPRETATION TIPS

The DCS data in the listings consists of two types of statements: micro-orders and microinstruction macros. Micro-orders take the form <field>/<value>, where <field> is a field of bits in the control store word and <value> is a hexadecimal number to be placed in that field. For example, WCTRL/MEMSCR (from line 548 of the example in Paragraph 8.3) puts the value identified by MEMSCR into the WCTRL field, bits <30:25>, of the control store word.

However, to find out what the WCTRL field actually does with that value, the machine definitions section of the VAX-11/750 microcode listings under WCTRL must be checked. The comment in the listing for the WCTRL/MEMSCR micro-order reads as follows.

;WBUS<27:24><– Memory Status and Control Reg (@MEMSCAR)

The micro-order therefore asserts bits <27:24> from the memory status and control register, addressed by MEMSCAR, on the W-BUS.

Macros generate the microcode for one or more micro-orders within one microinstruction. Generally, the function of a macro can be interrupted by analyzing the macro name. Macros like NOP and STOP.CLOCK are obvious.

All macros that include an underscore symbol are register transfer macros. The transfer goes from the right term to the left term. However, when the underscore symbol is part of a micro-order, it does not indicate the presence of a macro within the micro-order.

Commas separate micro-orders and macros, when several are used to form one microinstruction.

Consider the DCS data in the example in Paragraph 8.3.

Line 540: MEMSCAR R[TEMP0]

Move the contents of the scratch pad register RTEMP0 to MEMSCAR, the memory status and control address register.

Line 544: WB RDM,WCRTL/MEMSCR WB

The WB RDM macro moves data from the WDREG on the RDM to the W-BUS. WCTRL/MEMSCR WB, according to the micro-order definition in the VAX-11/750 microcode listing, moves W-BUS bits <27:24> to the memory status and control register. This is a write function.

Line 548: RDM WB,WCTRL/MEMSCR

WCTRL/MEMSCR moves the data from MEMSCR to the W-BUS bits <27:24>. The macro RDM WB moves that data from the W-BUS back to the WHREG on the RDM. This is a read function.

8.6 DIAGNOSTIC CONTROL STORE AND CONTROL FILE

The diagnostic control store (DCS) contains 64 locations that are each 88 bits wide. The DCS address space runs from control store address 1800 to 183F. The first 80 bits in each word, bits <79:00>, correspond to the 80 bits of the VAX-11/750 microword. The upper eight bits in each word, bits <95:88>, make up the control file. They enable and disable control functions on the RDM. The intervening bit numbers, bits <87:80>, are reserved for the macro assembler.

The microdiagnostic tests control the eight bits of the control file with eight microinstruction macros (Table 8-1). These macro functions correspond to the set control file functions listed in Paragraph 6.8.

Table 8-1 RDM Control File Macros

Macro Name	Bit	RDMCF	Function
STROBE.V.BUS	88	0	This macro strobes VBus at rising edge of next M clock.
DCS.ADDR_0	89	1	This macro clears DCS address register. It returns to DCS address 0.
STOP.CLOCK	90	2	This macro stops CPU clock.
LATCH.CS.ADDR	91	3	This macro latches trace register at rising edge of next M clock.
RDM WB	92	4	This macro reads data on W-BUS into WHREG on RDM at rising edge of next M clock.
WBRDM	93	5	This macro asserts (writes) contents of WDREG (on RDM) on W-BUS for entire microcycle.
RDM CMI	94	6	This macro reads data on CMI bus into the MHREG on RDM at the rising edge of next M clock.
INH.CLOCK.SA	95	7	This macro inhibits clocking signature analyzer on M clock.

8.7 LOOPING AND STEPPING THROUGH TESTS

After analyzing the test description, code, and comments for the failing test, you may want to analyze certain signals in the processor. Locate points of interest in the circuit under test in the VAX-11/750 print set.

To loop on the failing test at machine speed, follow the command sequence shown below.

```
RDM>TE/C<RET>                    ! Load the microdiagnostic
                                 ! monitor
MIC>DI TE:4<RET>                 ! Execute the failing
                                 ! test

ECKAA-VO.22 MIC (L0003)-V00.04
04,
?ERROR:  0036 TEST: 04 SUBTEST: 01
```

```
DATA:    0A000000                ! Expected cache error register
                                 ! contents
         08000000                ! Received cache error register
                                 ! contents
         00000001                ! Loop count
                                 ! Refer to Error Description
                                 ! example in Paragraph 8-3
Failing Gate Arrays: CAK,
MIC>LO<RET>                      ! Loop on test 4
```

To step through the entire test, one pseudo-instruction at a time, follow the steps shown in the example below. Note the flow of the subtest sequence in the TPC addresses.

```
MIC>DI TE:4<RET>                 ! Execute the failing test
ECKAA-VO.22 MIC (L0003)-V00.04
04,
?ERROR: 0036 TEST: 04 SUBTEST: 01

DATA:    0A000000                ! Expected data
         08000000                ! Received data
         00000001                ! Loop count
Failing Gate Arrays: CAK,
MIC>SE ST IN                     ! Set step instruction

MIC>CO<RET>
         TPC= 003C               ! Test continues at this point
         TPC= 0018               ! Restart subtest 1, using
         TPC= 0020               ! the second test pattern, I = 2
         TPC= 0022
         TPC= 0026
         TPC= 002A
         TPC= 0036
         TPC= 003C
         TPC= 0018               ! Restart subtest 1, using
         TPC= 0020               ! the third test pattern, I = 3
         TPC= 0022
         TPC= 0026
         TPC= 002A
         TPC= 0036
         TPC= 003C
         TPC= 003E
         TPC= 0041               ! Start subtest 2, using
         TPC= 0043               ! the first test pattern, I = 1
         TPC= 0045
         TPC= 004C
         TPC= 004E
         TPC= 0055
         TPC= 005C
         TPC= 005F
         TPC= 0063
         TPC= 0067
         TPC= 0073
         TPC= 0079
```

```
          TPC= 0055          ! Restart subtest 2, using
          TPC= 005C          ! the second test pattern, I = 2
          TPC= 005F
          TPC= 0063
          TPC= 0067
          TPC= 0073
          TPC= 0079
          TPC= 0055          ! Restart subtest 2, using
          TPC= 005C          ! the third test pattern, I = 3
          TPC= 005F
          TPC= 0063
          TPC= 0067
          TPC= 0073
          TPC= 0079
          TPC= 007C
          TPC= 007E
          TPC= 00AA
          TPC= 0004          ! Loop to restart subtest 1,
          TPC= 0006          ! using the first test pattern,
          TPC= 0008          ! I = 1
CTRL/C    TPC= 000F↑C        ! Type CTRL/C (or carriage
                             ! return) to return to the
                             ! microdiagnostic monitor
MIC>
```

To step through the DCS code with a specific pattern in a specific subtest, step through the pseudo-instructions to the proper point. Then type SE ST CY (set step cycle) to stop at each DCS instruction. The example below shows how to step through the DCS code, using test pattern 2 in subtest 1.

```
MIC>DI TE: 4<RET>                ! Execute the failing test
ECKAA-VO.22 MIC (L0003)-V00.04
04,
?ERROR: 0034 TEST: 04 SUBTEST: 01

DATA:     0A000000               ! Expected data
          08000000               ! Received data
          00000001               ! Loop count
Failing Gate Arrays: CAK,
MIC>SE ST IN<RET>                ! Set step instruction

MIC>CO<RET>

          TPC= 003C
          TPC= 0018              ! Restart subtest 1, using
          TPC= 0020              ! the second test pattern
          <RET>                  ! Carriage return

MIC>SE ST CY<RET>

MIC>CO<RET>                      ! Step through DCS
                                 ! with the second test pattern
```

```
        DCS ADDR= 00
        DCS ADDR= 01
        DCS ADDR= 02
        DCS ADDR= 03
        DCS ADDR= 04
MIC>
```

Note that control returns to the microdiagnostic monitor automatically, after execution of the last micro instruction at DCS ADDR= 04.

Example 4 shows how you can force a loop in the DCS code with the Set Control File command. Set control file bit 1 and clear bit 2 in the last DCS word in the sequence.

Example 4

```
MIC>SE CF:4 1<RET>          ! Set CF bit 1 in DCS
                            ! address 4
MIC>CL CF:4 2<RET>          ! Clear CF bit 2 in DCS
                            ! address 4
MIC>CO<RET>                 ! Continue
```

APPENDIX A
CABLE CONNECTIONS AND TEST POINTS

A.1 GENERAL

Table A-1 describes the interconnection of the RDM to the modem. Table A-2 lists the backplane pin numbers for the local terminal, the TU58, and the front panel cables. Table A-3 lists backplane pin numbers for specific RDM test points.

Table A-1 RDM/Modem Signal Flow and Interconne

RDM Pin	7016921 Cable Connector Pin	EIA (RS-232C) Signal Name
C0671	1	Protective Ground
C0672	7	Signal Ground
C0678	2	Transmitted Data
C0680	3	Received Data
C0690	4	Request to Send
C0688	5	Clear to Send
C0686	6	Data Set Ready
C0682	20	Data Terminal Ready
C0692	22	Ring Indicator
C0684	8	Carrier Detect
	Not used	Force Busy

ction

EIA (Abbreviation)	EIA Circuit
GND	AA
GND	AB
TXD	BA
RXD	BB
RTS	CA
CTS	CB
DSR	CC
DTR	CD
RI	CE
CD	CF
FB	CN

Table A-2 Backplane Pins for Local Terminal, TU58, and Front Panel Cables

Pin	Signal Name
Local Terminal	
C0624	Ground
C0634	RDM terminal serial out
C0632	RDM terminal serial in
C0645	Console baud rate A L
C0646	Console baud rate B L
C0649	Console baud rate C L
C0650	Console baud rate D L
TU58	
B0686	TU58 serial out (signals between TU58 and RDM)
B0688	TU58 serial in (signals between TU58 and RDM)
B0685	EIA TU serial out L (signals between RDM and CPU)
B0687	EIA TU serial in L (signals between RDM and CPU)
Front Panel	
Front panel cable is plugged into SET A (top half), with cable pin 1 on top.	

Table A-3 RDM-Specific Backplane Pins

Pin	Signal Name
C0673	RDM SA CLK L (D)
C0675	SA ST/SP L (D)
C0674	RDM RESET L (R)
C0681	RDM MATCH PULSE (D)
TP1	RDM MATCH PULSE (D) (Located in *front* of RDM near handle. See Figure 3-5)

(D) = Drive, (R) = Receive.

APPENDIX B
MODEMS

B.1 SCOPE

The KC750-CA option comes with one of three modem types: General Datacom (GDC), Racal/Vadic, or DIGITAL.

This appendix covers the following topics for each of these three modems.

- Inspection
- Indicators during a normal session
- Modem testing

B.2 GENERAL DATACOM MODEM (GDC)

B.2.1 Inspection

Remove the screw at the top rear center of the GDC modem. Tilt the cover forward and remove it.

Make sure that all the switches and jumpers are configured as in Figure B-1.

Reassemble in reverse order.

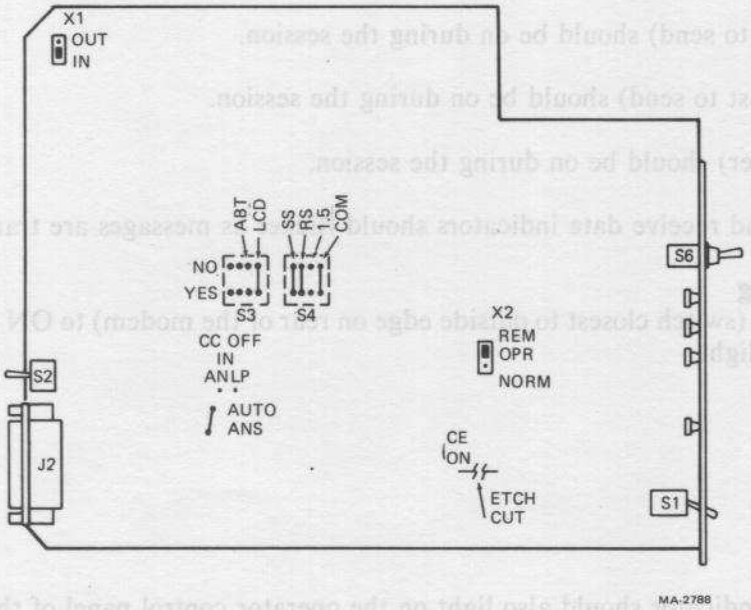


Figure B-1 GDC Jumper Configuration

B.2.2 Indicators During a Normal Session

1.	CARRIER DETECT	LED should be on during remote session.
2.	DATA MODE	LED should be on during remote session.
3.	DATA IN	These LEDs flicker as data is
4.	DATA OUT	transmitted and received.

B.2.3 Modem Testing

Move the NORMAL/ANALOOP switch to the ANALOOP position. The CARRIER DETECT/ANALOOP indicator on the modem should light. The RD CARRIER lamp on the operator control panel should also light.

Make sure the switch is returned to the NORMAL position.

Dial into the modem and listen for carrier for about 20 seconds.

B.3 VADIC/RACAL MODEM

B.3.1 Inspection

Remove the top cover of the VADIC/RACAL modem by prying it gently at the corners.

Make sure all switches and jumpers are connected as shown in Figure B-2.

Reassemble in reverse order.

B.3.2 Indicators During a Normal Session

1. DTR (data terminal ready) should light when the keyswitch on the VAX-11/750 is moved to REMOTE positions.

2. DSR (data set ready) should be on during the session.

3. CTS (clear to send) should be on during the session.

4. RTS (request to send) should be on during the session.

5. CXR (carrier) should be on during the session.

6. Transmit and receive data indicators should flicker as messages are transmitted.

B.3.3 Modem Testing

Move the ALB switch (switch closest to outside edge on rear of the modem) to ON position. The following indicators should light.

1. RTS
2. CTS
3. DSR
4. CXR
5. DTR

The RD CARRIER indicator should also light on the operator control panel of the VAX-11/750. Return switch to normal (off) position.

Again call the number of the modem and listen for 20 seconds of carrier as quick test.

RACAL/VADIC MODEM

OPTION JUMPER	JUMPER IN	CONFIGURATION OUT
A		*
B	*	
C	*	
D		*
E		*
F	*	
G	*	
H	*	

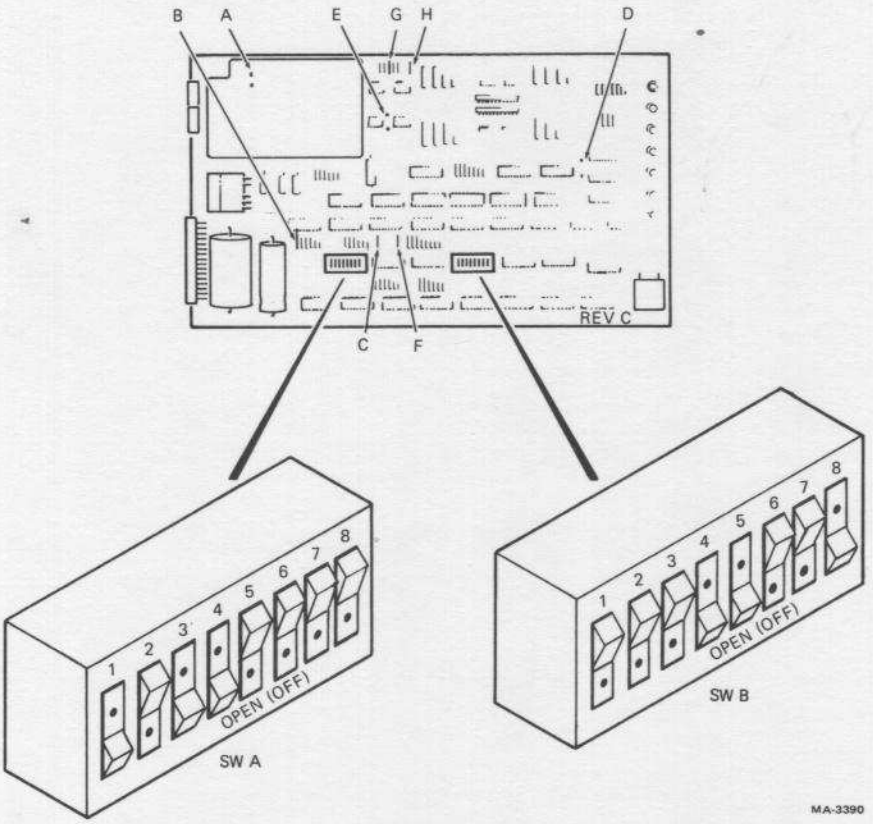


Figure B-2 Racal/Vadic Jumper Configuration and Switch Settings

B.4 DIGITAL EQUIPMENT DF02 MODEM

B.4.1 Inspection

The DF02 modem is shipped ready to use. No inspection is needed.

B.4.2 Indicators During a Normal Session

1. DTR is on during the session
2. CAR is on during the session
3. OH is on during the session

B.4.3 Modem Testing

Dial the modem from another phone and listen for 20 seconds of carrier.

APPENDIX C
INSTALLATION ACKNOWLEDGEMENT

C.1 FORM

Figure C-1 shows an Installation Acknowledgement form. This form must be signed by the customer to ensure DIGITAL's full and free access to equipment. The part number for this form is EN-01632-07.

digital

DIGITAL EQUIPMENT CORPORATION

VAX11/750 REMOTE DIAGNOSIS INSTALLATION ACKNOWLEDGEMENT

I, the undersigned, acknowledge receipt of Remote Diagnosis Kit (KC750).

Asset No. ..... for use on VAX 11/750 ,

Serial No. ..... .

I recognize that this Remote Diagnosis Kit always remains the property of Digital Equipment Corporation, and agree not to remove or tamper with any part of it, nor to disclose or make any part of it available to a third party.

I further acknowledge that I will allow Digital full and free access for the purpose of removal of the Diagnosis Kit and restoration of the system to its original condition if for any reason the Field Service Agreement should be terminated.

Computer System to be located at: _____

Company _____

Address _____

City _____ *State* _____ *Zip Code* _____

By Authorized Representative _____ *Date* _____

Title _____

EN-01632-07 MA-5118

White—Customer Canary—Branch Pink—D.D.C.

Figure C-1 Installation Acknowledgement Form

APPENDIX D
RDM SPECIFICATIONS

Size:	31.12 cm (12.25 in) × 40.01 cm (15.75 in) extended hex
Pin Configuration:	3 × 94 – Standard L series
Power:	
Maximum	+5 V @ 12.9 A, +12 V @ 120 mA, −15 V @ 85 mA
Typical	+5 V @ 9.6 A, +12 V @ 60 mA, −15 V @ 30 mA
Technology:	TTL (Schottky, Low Power Schottky)
	MOS

Environmental standards (temperature, pressure, etc.) are the same as for the VAX-11/750.

APPENDIX E
MODEM OPERATING CHARACTERISTICS

E.1 INTRODUCTION

This appendix defines the required functional characteristics of modems used with DIGITAL remote diagnosis devices. Those devices operate with the following characteristics.

- 300 baud serial data communication
- Asynchronous (start/stop) character framing (one start bit, eight data bits, no parity bit, one stop bit)
- Full duplex
- Automatic answer
- Automatic disconnect

E.2 DEFINITIONS

Table E-1 defines the signals recognized by the RDM and modem. The table lists the signal name, followed by the EIA RS-232C circuit name, the CCITT V.24 circuit number, the position on a 25-pin D-type connector, and the signal definition.

E.3 CALL REQUIREMENTS

The following paragraphs detail the procedure for successful completion of call answering and termination.

E.3.1 Automatic Call Answering

Use the following procedure to enable automatic call answering (Figures E-1 and E-2).

Set the front panel keyswitch to either REMOTE/SECURE or REMOTE. This clears all console device signals (DTR, RTS) and inhibits data line TxD (held at binary 1). It also clears all modem signals (RI, DSR, CTS, CD) and inhibits line RxD (held at binary 1). The console asserts the data terminal ready (DTR) signal and enables call answering.

When the console asserts data terminal ready, the modem attempts to answer the phone when it detects at least one ring indicator (RI) signal. When it answers the phone, the modem asserts data set ready (DSR). Once the console detects DSR, DTR may not deassert for five seconds, unless the console has also detected CD. The console must detect DSR assertion within 20 seconds, or the call terminates. Once it detects DSR, the console asserts the request to send (RTS) signal.

The modem responds to RTS assertion by placing its carrier signal or answer tone on the line. When the modem receives a carrier signal from the calling station, the modem responds by asserting the carrier detect (CD) signal and then asserting the clear to send (CTS) signal (700 $\pm$300 ms later). Note, however, that whenever CD is off, RxD is held at binary 1. The console must detect CD and CTS within 20 seconds of RTS assertion, or the call terminates.

Table E-1 Modem Signal Definitions

Name	RS-232C	V.24	25-pin	Definition
GND	AA	101	1	Protective ground – This signal provides a path between the remote diagnosis device and the modem for discharge of spurious potentials, such as static electricity.
GND	AB	102	7	Signal ground – This signal provides a reference level for the data and control signals that follow in this table.
TxD	BA	103	2	Transmit data (console to modem) – This signal contains the serial bit stream to be sent from the console to the calling station.
RxD	BB	104	3	Receive data (modem to console) – This signal contains the serial bit stream received by the modem from the calling station.
RTS	CA	105	4	Request to sent (console to modem) – This signal is asserted by the console, causing the carrier signal to be placed on the line. Called the station's answer mode carrier.
CTS	CB	106	5	Clear to send (modem to console) – This signal is asserted by the modem to indicate that it has successfully placed its carrier signal on the line.
DSR	CC	107	6	Data set ready (modem to console) – This signal indicates to the console that the telephone has been answered. (The telephone is "off hook".)
DTR	CD	108/2	20	Data terminal ready (console to modem) – This signal is asserted by the console, enabling telephone answering when a ring occurs.
RI	CE	125	22	Ring indicator (modem to console) – This signal is monitored by the console to determine when a ring occurs (when a DDC host is attempting to call the system).
CD	CF	109	8	Carrier detect (modem to console) – This signal is asserted by the modem to indicate that the calling station's carrier signal has been detected.

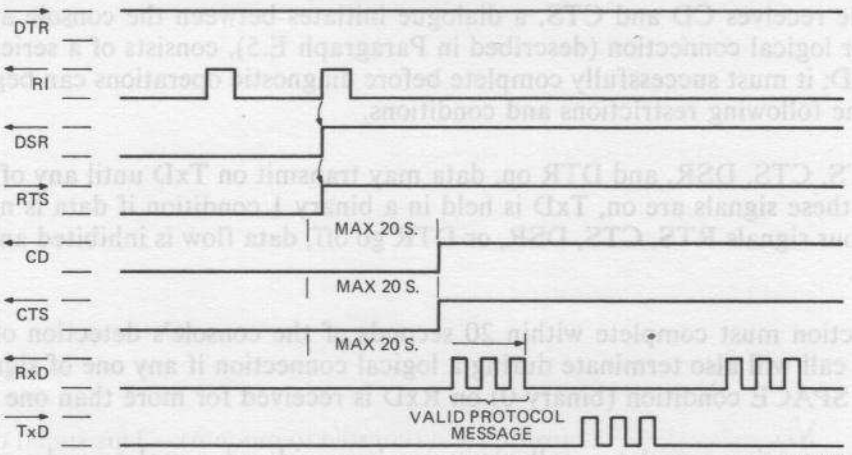


Figure E-1 Automatic Call Answering Sequence

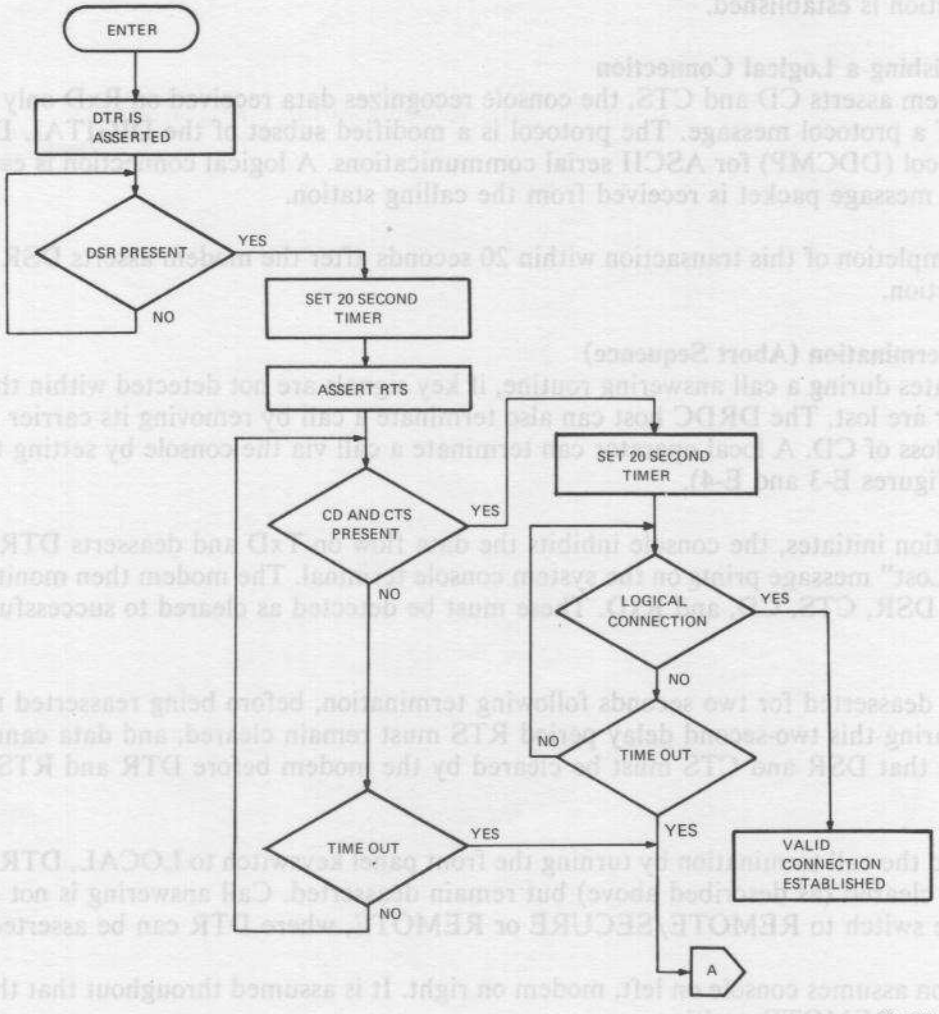


Figure E-2 Automatic Call Answering Flowchart

Once the console receives CD and CTS, a dialogue initiates between the console and calling station. This dialogue, or logical connection (described in Paragraph E.5), consists of a series of data transfers via RxD and TxD; it must successfully complete before diagnostic operations can begin. TxD and RxD are subject to the following restrictions and conditions.

With signals RTS, CTS, DSR, and DTR on, data may transmit on TxD until any of these four signals goes off. While these signals are on, TxD is held in a binary 1 condition if data is not transmitting. If any one of the four signals RTS, CTS, DSR, or DTR go off, data flow is inhibited and TxD returned to a binary 1 state.

A logical connection must complete within 20 seconds of the console's detection of DSR or the call terminates. The call will also terminate during a logical connection if any one of signals DSR, CTS or CD is lost, or a SPACE condition (binary 0) on RxD is received for more than one second.

Once a logical connection completes, call answering is considered complete and a valid connection in effect. At this point, the diagnostic process may proceed.

Note that a call will terminate during call answering, but before logical connection, if any one of signals DSR, CD, or CTS is lost for more than 500 ms. Signal glitches less than 500 ms are ignored until a logical connection is established.

E.3.2 Establishing a Logical Connection

Once the modem asserts CD and CTS, the console recognizes data received on RxD only if the data is in the form of a protocol message. The protocol is a modified subset of the DIGITAL Data Communication Protocol (DDCMP) for ASCII serial communications. A logical connection is established if a valid protocol message packet is received from the calling station.

Successful completion of this transaction within 20 seconds after the modem asserts DSR establishes a logical connection.

E.3.3 Call Termination (Abort Sequence)

A call terminates during a call answering routine, if key signals are not detected within the established time frame, or are lost. The DRDC host can also terminate a call by removing its carrier signal, which results in the loss of CD. A local operator can terminate a call via the console by setting the keyswitch to LOCAL (Figures E-3 and E-4).

Once termination initiates, the console inhibits the data flow on TxD and deasserts DTR and RTS. A "Connection Lost" message prints on the system console terminal. The modem then monitors the states of signals RI, DSR, CTS, CD, and RxD. These must be detected as cleared to successfully terminate the call.

DTR remains deasserted for two seconds following termination, before being reasserted to enable call answering. During this two-second delay period RTS must remain cleared, and data cannot be placed on TxD. Note that DSR and CTS must be cleared by the modem before DTR and RTS can be reasserted.

If you initiated the call termination by turning the front panel keyswitch to LOCAL, DTR and RTS are automatically cleared (as described above) but remain deasserted. Call answering is not enabled until you return the switch to REMOTE/SECURE or REMOTE, where DTR can be asserted.

Signal direction assumes console on left, modem on right. It is assumed throughout that the front panel keyswitch is in a REMOTE position.

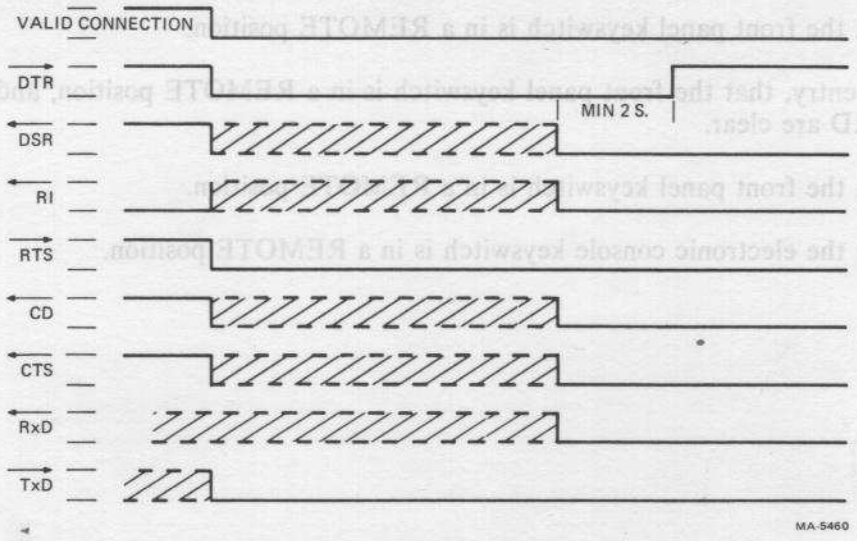


Figure E-3 Call Abort or Termination Sequence

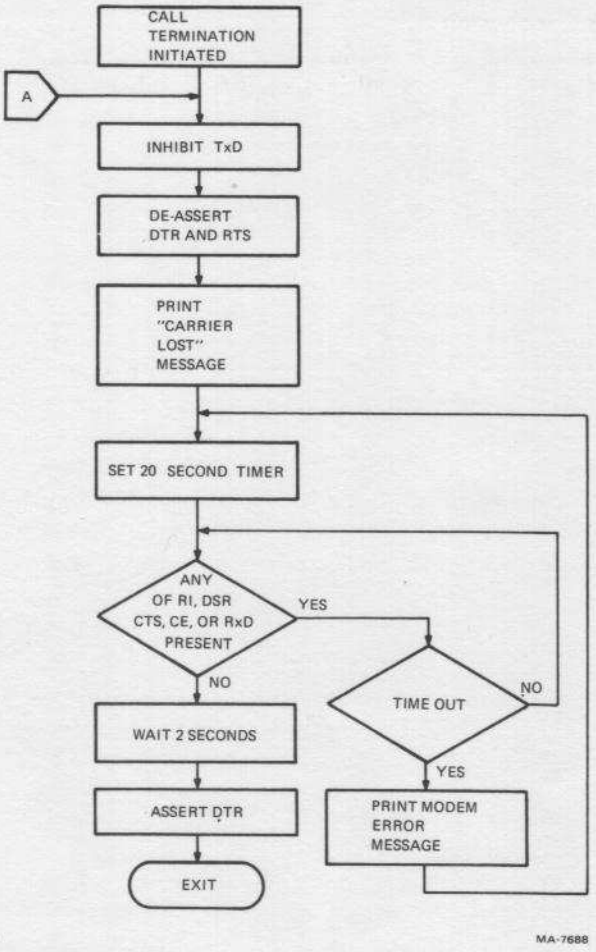


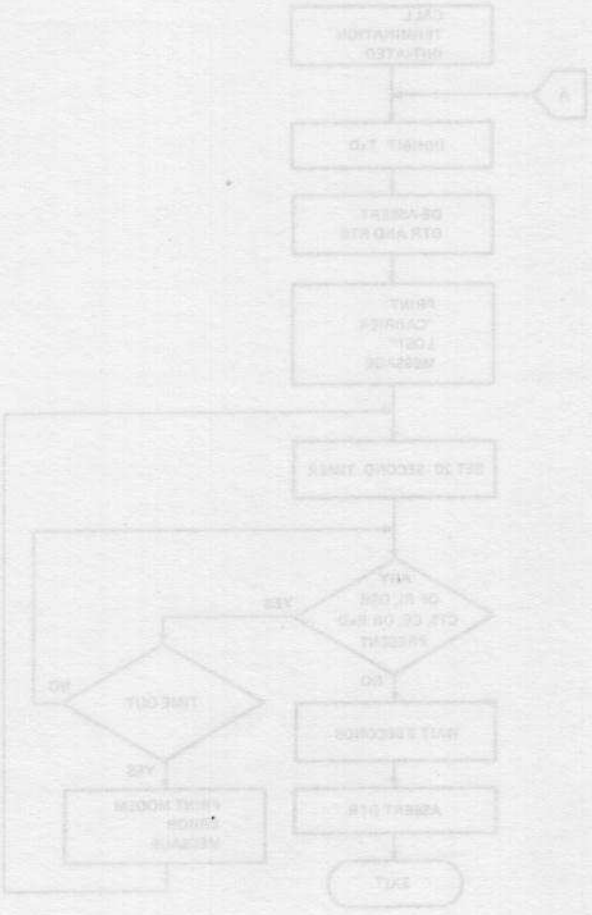
Figure E-4 Call Abort or Termination Flowchart

It is assumed that the front panel keyswitch is in a REMOTE position.

It is assumed, on entry, that the front panel keyswitch is in a REMOTE position, and that RI, DSR, CD, CTS, and RxD are clear.

It is assumed that the front panel keyswitch is in a REMOTE position.

It is assumed that the electronic console keyswitch is in a REMOTE position.



Digital Equipment Corporation • Bedford, MA 01730