Table of Contents

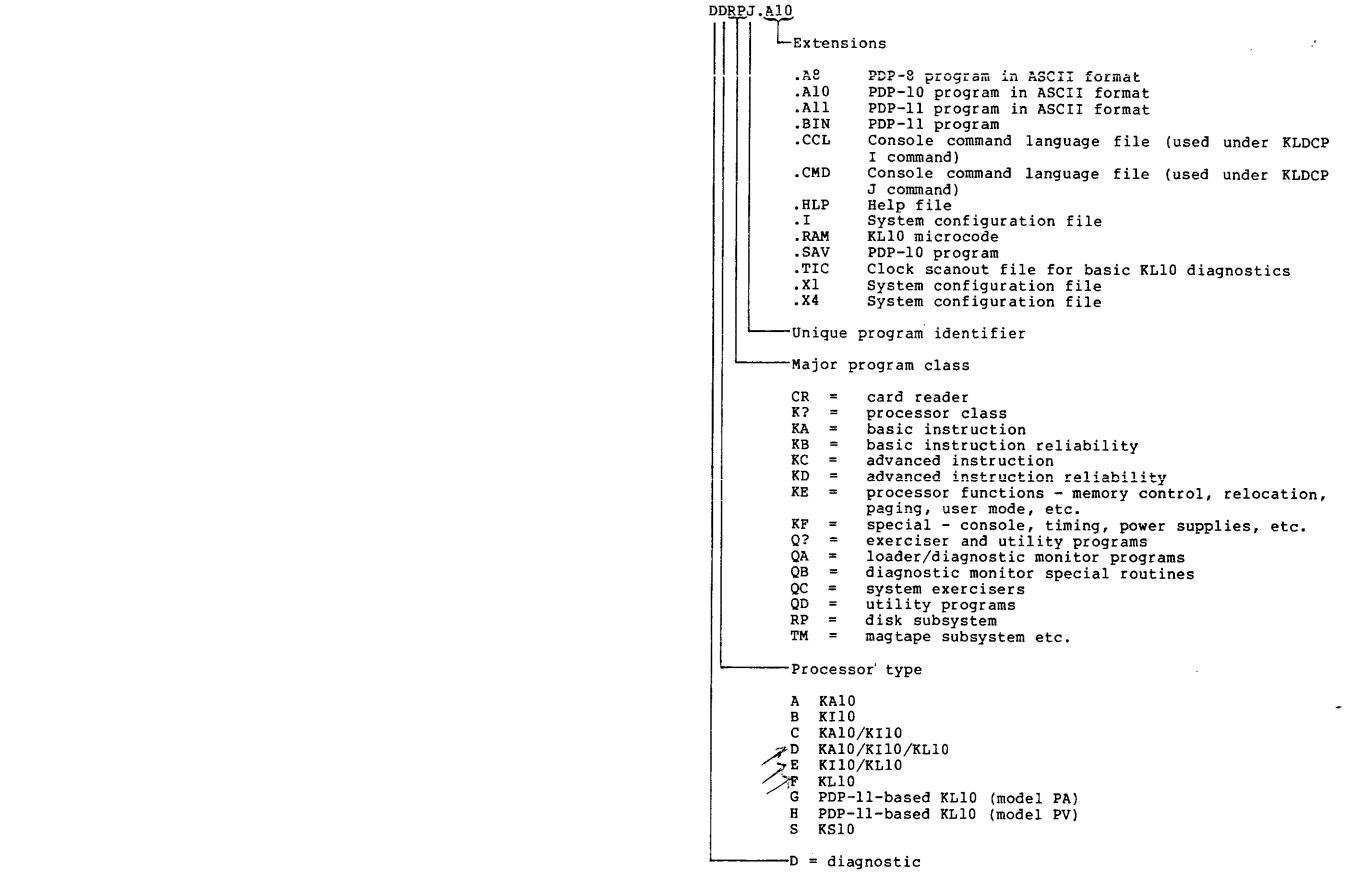
Summary	Microfiche Classification	Version
10/10 STD	NA	
DIAMON	KA,KI,KL10(DDQDC)	0.14
D20MON	KL10(DDQDH)	0.2
KLAD10	--	0.2
MAGMON	KA,KI,KL10(DDQDE)	0.12
MAGTAP	KL10(DDQDF)	0.5
SUBXXX	KA/KI/KL10(DDQDA)	0.14
XXDDT	KA10/KI/KL(DDQDB)	0.12
DDCPA	CP10	0.5
DDCRA	CR10	0.2
DDDFA	DF10/DF10C	0.3
DDDXA	DX10	0.4
DDDXB	DX10	0.5
DDLPA	LP10	0.5
DDMMC	DEC SYS 10	0.1
DDMMD	KA10/KI/KL	0.2
DDMME	KA10,KI10,KL10	0.1
DDMMF	KA10,KI10,KL10	0.1
DDPCA	PC0X	0.2
DDQCB	KA10,KI/KL	0.7
DDRHA	RH10	0.3
DDRPD	KA10,KI10,KL10	0.3
DDRPH	RH10/RP04	0.3
DDRPI	RH10,RS04	0.6
DDRPK	RH10/RP06	0.3
DDTMH	TM10/TU70	0.3
DDTUA	TU70	0.7
DDTUB	TU70	0.7
DDXYA	XY10	0.1
DEMMG	PDP-10	0.1
DFDTE	KL10	0.10
DFKAA	KL10	0.2
DFKAB	KL10	0.1
DFKAC	KL10	0.1
DFKAD	KL10	0.1
DFKBA	KL10	0.1
DFKBB	KL10	0.1
DFKCA	KL10	0.1
DFKDA	KL10	0.2
DFKFB	KL10	0.1

Table of Contents (Cont)

Summary	Microfiche Classification	Version
DFKFD	KL10	0.2
DFRHB	RH20	0.12
DFRPH	RH20/RP04	0.3
DFRPK	RH20/RP06	0.3
DFSXA	KL10	0.4

10-BASED 10 STANDARD INFORMATION
This module summarizes standard information and procedures which are common to many of the programs in the 10-Based 10 Maintenance Library.

PROGRAM IDENTIFICATION CODE
The following figure describes the naming convention used to identify 10-Based 10 Maintenance Library and 11-Based 10 Maintenance Library programs.



STANDARD PROGRAM STARTING ADDRESSES
Table 1 lists the standard starting addresses for both the diagnostic and utility programs.

Table 1 Standard Program Starting Addresses

Address	Tag	Function
Diagnostic Monitors (DIAMON, D20MON, MAGMON)		
20000	N/A	This is the standard starting address for all 10/10 diagnostic monitors.
20001	N/A	This address is used in conjunction with diagnostic monitor control files. Restarting the CPU at 20001 will cause the program currently being run under the monitor to be stopped and the next program in the control file to be started.
20002	N/A	This address is used in conjunction with control switch 15. Restarting the CPU at 20002 will cause the title of the program currently being run under the monitor to be printed. After the title is printed the CPU will halt. The user may restart either the diagnostic monitor or the program at this time.
2003	N/A	This address is used to restart the current program.
Diagnostic Programs		
30000	BEGIN	Stand-alone start
30001	$START	Mode check starting address
30002	DIAGMN	Diagnostic monitor start
30003	SYSEXR	System exerciser start
30004	SFSTRT	Special feature start
30005	PFSTRT	Power fail restart
30006	REENTR	Reenter start
30007	DDTSRT	DDT start
30010	BEGIN1	Start next program pass
30011	SBINIT	PGMINT linkage
30012	RETURN	Return address storage
30013	START1	Optional starting address/instructions
30014	START2	Optional starting address/instructions
30015	START3	Optional starting address/instructions
30016	START4	Optional starting address/instructions
30017	START5	Optional starting address/instructions

STANDARD CONSOLE CONTROL SWITCHES

Table 2 lists and describes the standard functions of the console data switches used to control the operation of most 10-based 10 diagnostics. Exceptions to the standard and right half switches 18 through 35 are described in the individual program summaries.

Setting the Console Data Switches

Exec Mode

The switches used to control single programs run on KA10- and KI10-based systems are derived directly from the console data switches. The switches used to control single programs run on KL10-based systems are derived by exclusive ORing the 11/40 console data switches with the six octal digits supplied to the KLDCP ES command, and appending the result to the seven octal digits supplied to the KLDCP SW command. Refer to Figure 1.

The right-half program control switches are redefined when a diagnostic monitor and control file are used to automatically load and sequence a series of diagnostic programs. Refer to the appropriate diagnostic monitor summary module for further information.

User Mode

A diagnostic monitor must be used to run diagnostic programs in user mode. If a single program is run, the program will request that the left and right half switches be entered as octal digits. If less than six digits are typed for either half, the digits typed will be right-justified and zero-filled.

If a diagnostic monitor and control file is used to automatically load and sequence a series of programs, the monitor will request that the left half switches be entered as octal digits. If less than six digits are typed they will be right-justified and zero-filled. The left half digits typed will be shared by all programs in the control file. The right half switches for each program will be supplied by the control file.

10/10 STD = (1000)

ES 200

Table 2 10/10 Standard Control Switch Summary

Switch	Mnemonic	State	Function
0 (400000)	ABORT	0 1	Normal operation Abort at end of pass
1 (200000)	RSTART	0 1	No function List totals and restart
2 (100000)	TOTALS	0 1	No function List totals and continue
3 (040000)	NOPNT	0 1	Normal typeout Inhibit all printing except forced
4 (020000)	PNTLPT	0 1	Normal output to terminal Print on line printer (user, logical DEV)
5 (010000)	DING	0 1	No function Ring terminal bell on error (forced output)
6 (004000)	LOOPER	0 1	Proceed to next test Enter scope loop on test error
7 (002000)	ERSTOP	0 1	No function Halt on error after reporting error (exec mode), resume normal sequence by pressing CONTINUE. In user mode, this switch causes a CALL AC, EXIT to be executed. Normal test sequence may be resumed by typing .CONT.
8 (001000)	PALERS	0 1	Print only first error in loop Print all errors
9 (000400)	RELIAB	0 1	Quick verify mode Reliability mode
10 (000200)	TXTINH	0 1	Print full error messages. Inhibit comment portion of error messages.
11 (000100)	INHPAG	0 1	KI10 and KL10 - allow full 256K/4096K addressing KI10 and KL10 - inhibit paging; i.e., treat memory as 112K-1 maximum
12 (000040)	MODDVC	0 1	No device code change Modify device codes The program will ask the following questions when invoked: CHANGE DEVICE CODES, Y OR N <CR> - OLD DEVICE CODE - NEW DEVICE CODE - Answer questions appropriately. The change device codes question will be reasked until you say no. Change as many device codes as wanted.
13 (000020)	INHCSH	0 1	KL10 - allow cache use KL10 - inhibit cache
14 (000010)	OPRSEL	0 1	Run default operations Operator test selections
15 (000004)	CHAIN	0	This switch used by DIAMON, etc., to control chain operations
16 (000002)	KAHZ50	0 1	Normal operation KA10 50 Hz switch
17 (000001)			Reserved
18 thru 35			Refer to the individual diagnostic summaries.

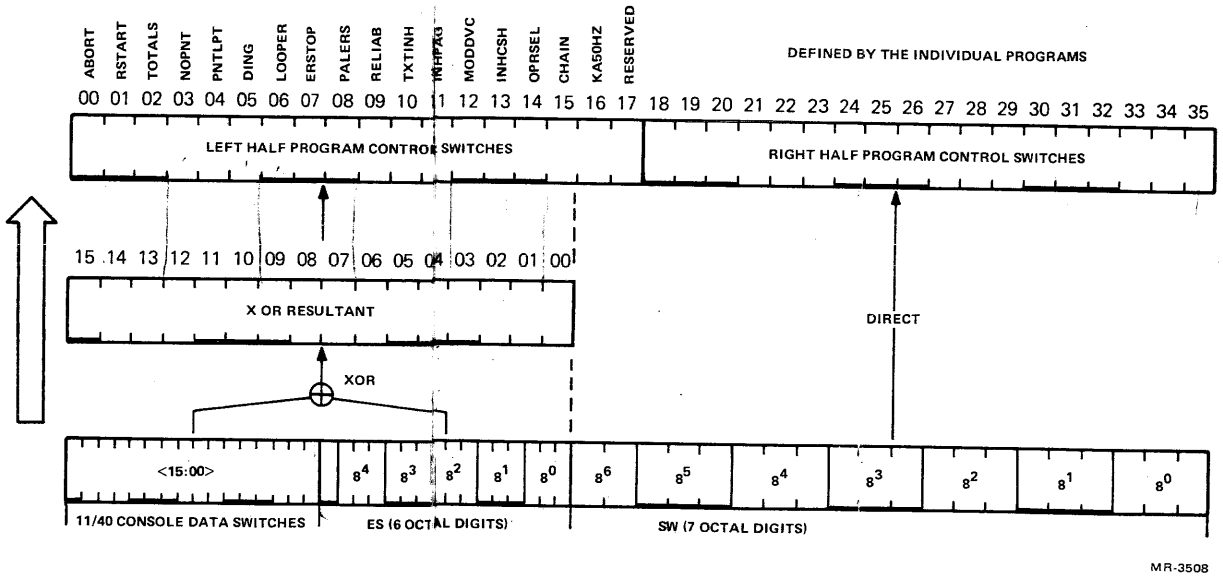


Figure 1 KL10 Console Switch Formation

UTILITY PROGRAMS
Table 3 lists and briefly describes the 10-based 10 utility programs. Under MODE:

E indicates that the program may be run in exec mode.
U indicates that the program may be run in user mode.

Table 3 10/10 Maintenance Library Utility Programs

Utility	Mode	Description
DIAMON.SAV (DDQDC)	E U	A diagnostic monitor used to load and sequence diagnostic programs from any of the following. a. Paper tape b. KLDCP-selected load device c. DECtape (in either PDP-10 or PDP-11 format) d. Disk (having TOPS-10 file structure) Standard Control Files KLPROC.CMD - runs KL10 processor functional diagnostics (exec mode) KLUSR.CMD - runs KL10 processor functional diagnostics (user mode)
D20MON.SAV	E U	A version of the diagnostic monitor designed to handle TOPS-20 file structures. Standard Control Files - same as DIAMON.
KLAD10.SAV	U	A utility program used to construct and update KLAD-10 disk packs. Standard Control Files P109 - for KLAD-10 REV 9 (6, 11) area P1010 - for KLAD-10 REV 10 (6, 11) area
KLDDT.SAV	E U	An abbreviated version of DDT which can be loaded with and used to control diagnostic programs.
MAGMON.SAV (DDQDE)	E U	A diagnostic monitor used to load and sequence diagnostic programs from magtape in exec mode and magtape or disk in user mode. MAGMON will run under either the TOPS-10 or TOPS-20 operating system. Standard Control Files - Same as DIAMON.
MAGTAP.SAV (DDQDF)	U	A utility program used to generate and read diagnostic distribution tapes. MAGTAP also provides a simple method for transfering files between magtape and disk.
SUBRTN.SAV (DDQDA)	E U	A collection of subroutines commonly used by KA10 and KI10 diagnostic and utility programs. SUBRTN loads with and is used by almost all 10/10 diagnostic programs.
SUBKL.SAV	E	A version of Subroutine for use with KL10 exec mode diagnostics.
SUBUSR.SAV	U	A version of Subroutine for use with all user mode diagnostics.

DIAGNOSTIC PROGRAM HIERARCHIES
The following tables describe the 10-Based 10 Maintenance Library diagnostic hierarchies.

Table 4 System Exercisers

Table 5 KA10 Processor Diagnostic Hierarchy

Table 6 KA10 Memory Diagnostic Hierarchy

Table 7 KI10 Processor Diagnostic Hierarchy

Table 8 KI10 Memory Diagnostic Hierarchy

Table 9 KL10 Processor Diagnostic Hierarchy

Table 10 KL10 Memory Diagnostic Hierarchy

Table 11 Disk Subsystem Diagnostic Hierarchies

Table 12 Magtape Subsystem Diagnostic Hierarchies

Table 13 DECtape Diagnostics

Table 14 Hard Copy Equipment Diagnostics

Table 15 DL10 Diagnostics

Table 16 DC10 Diagnostics

Table 17 Terminal Diagnostics

Table 18 Miscellaneous Peripheral Equipment Diagnostics

Under MODE:

E indicates that the program may be run in exec mode.
U indicates that the program may be run in user mode.

Table 4 [redacted]

Diagnostic	Mode	Title
DDQCB.SAV	E	Memory Part Interation Test
DFSXA.A10	E	KL10/Channel/DTE20 Interaction Test

Table 5 KA10 Processor Diagnostic Hierarchy

Diagnostic	Mode	Title (Description)
Basic Instruction Tests		
DAKAA.SAV	E U	Part 1 (MOVE, SKIP, AND, XOR, EQV, BOOLE)
DAKAB.SAV	E U	Part 2 (MOVE, COMPARE, TEST, HWT, ADD, BOOLE)
DAKAC.SAV	E U	Part 3 (LOGICAL TEST, HWT)
DAKAD.SAV	E U	Part 4 (Register Addressing, JFCL, AR FLAGS, AOS, SOS)
DAKAE.SAV	E U	Part 5 (FWT, ADD, SUB PC change, COMPARE)
DAKAF.SAV	E U	Part 6 (BOOLE, HWT, TEST)
DAKAG.SAV	E U	Part 7 (PUSH, POP, XCT, Shift and Rotate)
DAKAH.SAV	E U	Part 8 (PI System, Interrupts, LUUOs, I/O)
DAKAI.SAV	E U	Part 9 (Shift and Rotate)
DAKAJ.SAV	E U	Part 10 (Shift and Rotate)
DAKAK.SAV	E U	Part 11 (MUL)
DAKAL.SAV	E U	Part 12 (MUL, IMUL, DIV, IDIV)
DAKAM.SAV	E U	Part 13 (BYTE, BLT, JFFO)
Advanced Instruction Tests		
DAKCA.SAV	E U	Part 1 (Floating Scale/ADD/SUB, UFA)
DAKCB.SAV	E U	Part 2 (Floating MUL/DIV, DFN)
Instruction Reliability		
DAKBA.SAV	E U	Part 1 (COMPARE, EXCH, BOOLE, ROTATE, TEST)
DAKBB.SAV	E [redacted]	, AOX, SOX, CAI, etc.)
DAKBC.SAV	E U	Part 3 (HWT, ADD, SUB, JFFO, etc.)
DAKBD.SAV	E U	Part 4 (Memory/Both modes, excluding MUL, DIV, BYTE JSR, JSA, JSP, JRA, PUSH, POP, PUSHJ, POPJ)
DAKDA.SAV	E U	Arithmetic Reliability (Floating Point)
DAKDB.SAV	E U	Random Instruction Reliability
Protection Relocation and Timing		
DAKEA.SAV	E	Relocation and Protection
DAKEB.SAV	E	Reenterent Reliability (Protect and Relocate)
DAKFA.SAV	E	Instruction Timing Test

Table 6 KA10 Memory Diagnostic Hierachy

Diagnostic	Mode	Title
DDMMC.SAV	E U	Fast AC Test
DDMMD.SAV	E U	256K Basic and Memory Reliability Test
Supplementary Tests		
DDMMF.SAV	E U	Floating Ones and Zeros Test
DDMME.SAV	E U	BLT/Memory Exerciser

Table 7 KI10 Processor Diagnostic Hierarchy

Diagnostic	Mode	Title (Description)
Basic Instruction Tests		
DBKAA.SAV	E U	Part 1 (MOVE, SKIP, AND, XOR, EQV, BOOLE)
DBKAB.SAV	E U	Part 2 (MOVE, COMPARE, TEST, HWT, BOOLE, ADD)
DBKAC.SAV	E U	Part 3 (LOGICAL TEST, HWT)
DBKAD.SAV	E U	Part 4 (Register Addressing, JFCL, AR FLAGS, AOS, SOS JRST) (AOBJX, JSP, XCT, indirect/index addressing)
DBKAE.SAV	E U	Part 5 (FWT, ADD, SUB, PC change, COMPARE)
DBKAF.SAV	E U	Part 6 (BOOLE, HWT, TEST)
DBKAG.SAV	E U	Part 7 (PUSH, POP, XCT, Shift and Rotate)
DBKAH.SAV	E U	Part 8 (PI System, Interrupts, LUUOs, I/O)
DBKAI.SAV	E U	Part 9 (Shift and Rotate)
DBKAJ.SAV	E U	Part 10 (Shift and Rotate)
DBKAK.SAV	E U	Part 11 (MUL)
DBKAL.SAV	E U	Part 12 (MUL, IMUL, DIV, IDIV)
DBKAM.SAV	E U	Part 13 (BYTE, BLT, JFFO)
Advanced Instruction Tests		
DBKCA.SAV	E U	Part 1 (Floating Scale/ADD/SUB, UFA)
DBKCB.SAV	E U	Part 2 (Floating MUL/DIV,DFN)
DBKCC.SAV	E U	Part 3 (FIX, FIXR, FLTR, DOUBLE MOVES)
DBKCD.SAV	E U	Part 4 (Double Precision DFAD, DFSB, DFMP, DFDV)
DBKCE.SAV	E U	Part 5 (Double Precision ADD, SUB, MUL, DIV)
Instruction Reliability		
DBKBA.SAV	E U	Part 1 (COMPARE, SKIP, EXCH, BOOLE, ROTATE, TEST)
DBKBB.SAV	E U	Part 2 (TEST, SKIP, JUMP, AOX, SOX, CIA etc.)
DBKBC.SAV	E U	Part 3 (HWT, ADD, SUB, JFFO etc.)
DBKBD.SAV	E U	Part 4 (Memory/Both modes, excluding MUL,DIV, BYTE JSR, JSA, JSP, JRA, PUSH, POP, PUSHJ, POPJ)
DBKDA.SAV	E U	Arithmetic Reliability (Fixed, Floating BYTE)
DBKDB.SAV	E U	Random Instruction Reliability
DBKDC.SAV	E	IOT/PI System Reliability
Paging and Mode Tests		
DBKEA.SAV	E	Paging Hardware Test
DBKEB.SAV	E	Monitor UUO and Modes Test
DBKEC.SAV	E	Paged Execute Diagnostic
Special Purpose Tests		
DBKFA.SAV	E	Console Functions Test
DBKFB.SAV	E	Instruction Timing Test
DFKFC.SAV	E	Power Supply Calibration Program
DFKFD.SAV	E	I/O Bus Tester (Requires I/O bus tester)

10/10 STD
= BSTRING RUN

Table 8 KI10 Memory Diagnostic Hierarchy

Diagnostic	Mode	Title
DDMMC.SAV	E U	Fast AC Test
DDMMD.SAV	U	256K Basic and Memory Reliability Test
DEMMG.SAV	E	4096K Basic and Memory Reliability Test
Supplementary Tests		
DDMMF.SAV	E U	Floating Ones and Zeros Test
DDMME.SAV	E U	BLT/Memory Exercises

Table 9 KL10 Processor Diagnostic Hierarchy

Diagnostic	Mode	Title
Basic, Advanced and Reliability Tests		
Note		Micocode must be loaded and started and memory must be configured.
DFKAA.A10	E U	Basic Instruction Test (Part 1)
Note		The SUBRTN package must be loaded for the remaining diagnostics.
DFKAB.A10	E U	Basic Instruction Test (Part 2)
DFKAC.A10	E U	Basic Instruction Test (Part 3)
DFKAD.A10	E U	Basic Instruction Test (Part 4)
DFKCA.A10	E U	Advanced Instruction Test
DFKBA.A10	E U	Instruction Reliability Test
Paging, MUUO, Modes and DTE20 Tests		
DFKEA.A10	E	Paging Hardware Test
DFKEB.A10	E	MUUO and User Mode Test
DFDTE.A10	E	DTE20 Interface Test
Processor, Priority Interrupt and Memory Reliability Test		
DFKDA.A10	E U	CPU PI and Memory Reliability Test
Supplementary Tests		
DFKBB.A10	E U	Instruction Reliability Test (Version 2)
DFKFB.A10	E	Instruction Timing Test
DFKFD.A10	E	I/O Bus Tester (Requires I/O bus tester)

Table 10 KL10 Memory Diagnostic Hierarchy

Diagnostic	Mode	Title
DDMMC.A10	E U	Fast AC Test
DDMMD.A10	U	256K Basic and Memory Reliability Test
DEMMG.A10	E	4096K Basic and Memory Reliability Test
DFMMH.A10	E	KL10 with MOS 4096K Basic and Memory Reliability Test
Supplementary Tests		
DDMMF.A10	E U	Floating Ones and Zeros Test 256K
DDMME.A10	E U	BLT/Memory Exerciser 256K

Table 11 Disk Subsystem Diagnostic Hierarchies

Diagnostic	Mode	Title
RH10-RP04/5/6 Disk Subsystem		
DDRHA.A10	E	RH10 Deviceless Test
DDRPH.A10	E U	RH10-RP04 Basic Device Test
DDRPI.A10	E U	RH10-RP05 Basic Device Test
DDRPJ.A10	E U	PH10-RP06 Basic Device Test
DDRPI.A10	E U	Disk Reliability Test
DDDFA.A10	E	DF10 Data Channel Test (Supplementary)
RH20-RP04/5/6 Disk Subsystem		
DFRHB.A10	E	RH20 Fault Isolator
DFRPH.A10	E U	RH20-RP04 Basic Device Test
DFRPJ.A10	E U	RH20-RP05 Basic Device Test
DFRPK.A10	E U	RH20-RP06 Basic Device Test
DDRPI.A10	E U	Disk Reliability Test
RH10-RS04 Disk Subsystem		
DDRHA.A10	E	RH10 Deviceless Test
DDRSC.A10	E U	Basic Device Test
DDRSB.A10	E U	Disk Transfer and Reliability Test
DDDFA.A10	E	DF10 Data Channel Test (Supplementary)
RP10/10C-RP02/03 Disk Subsystem		
DDRPA.SAV	E	Interface Test (Part 1)
DDRPB.SAV	E	Interface Test (Part 2)
DDRPC.SAV	E	Disk Reliability Test
DDRPD.SAV	U	User Mode Disk Performance Test
DDRPE.SAV	U	User Mode Formatting and Mapping
DDDFA.SAV	E	DF10 Data Channel Test (Supplementary)

Exercises
DDQCB Sys Exercisers/PORT Interaction Test

Table 12 Magtape Subsystem Diagnostic Hierarchies

Diagnostic	Mode	Title
RH10-TM01/03-TE16/TU16/45 Tape Subsystem		
DDRHA.A10	E	RH10 Deviceless Test
DDTUG.A10	E 2	Basic Tape Subsystem Test
DDTUK.A10	E 2	Tape Subsystem Reliability Test
DDTMH.A10	1	Magtape Reliability Test
DDDFA.SAV	E	DF10 Data Channel Test (Supplementary)
		1 TOPS-10 only 2 TOPS-20 only
RH20-TM02/03-TE16/TU16/45 Tape Subsystem		
DFRHB.A10	E	RH20 Fault Isolator
DFTUE.A10	E 2	Basic Tape Subsystem Test
DFTUK.A10	E 2	Magtape Reliability Test
DDTMH.SAV	1	Magtape Reliability Test
		1 TOPS-10 only 2 TOPS-20 only
TX01-DX10-TU70/71/72 Tape Subsystem		
DDDXA.A10	E U	DX10 Data Channel (Part 1)
DDDXB.A10	E U	DX10 Data Channel (Part 2)
DDTUA.A10	E U	Basic Tape Subsystem Test
DDTUB.A10	E U	Magtape Reliability Test
DDTMH.SAV	1	Magtape Reliability Test
		1 TOPS-10 only
TM10A-TU10/20/30/40/41		
DDTMA.SAV	E	Magtape Control Test (Part 1)
DDTMB.SAV	E	Magtape Control Test (Part 2)
DDTMC.SAV	E	Magtape Control Test (Part 3)
DDTMH.SAV	E 1	Magtape Reliability Test
		1 TOPS-10 only
TM10B-TU10/20/30/40/41		
DDTMD.SAV	E	Magtape Control Test (Part 1)
DDTME.SAV	E	Magtape Control Test (Part 2)
DDTME.SAV	E	Magtape Control Test (Part 3)
DDTMH.SAV	E 1	Magtape Reliability Test
DDDFA.SAV	E	DF10 Data Channel Test (Supplementary)
		1 TOPS-10 only

Table 13 DECtape Diagnostics

Diagnostic	Mode	Title
DDTDA.A10	E	TD10 Test (Part 1)
DDTDB.A10	E	TD10 Test (Part 2)
DDTDC.A10	E	TD10 TEST (Part 3)
DDTDD.A10	E U	DECtape Reliability Diagnostic
DDTDE.A10	E	DECtape Certification (Formatter)

Table 14 Hard Copy Equipment Diagnostics

Diagnostic	Mode	Title
DDCPA.A10	E U	Card Punch Diagnostic
DDCRA.A10	E U	Card Reader Diagnostic
DDLPA.A10	E U	Line Printer Diagnostic
DDPCA.A10	E U	Paper Tape Reader/Punch Diagnostic
DDXYA.A10	E U	Plotter Diagnostic

Table 15 DL10 (PDP-10/PDP-11 Interface Channel) Diagnostics

Diagnostic	Mode	Title
DDLA.A10	E U	DL10 Test (Part 1)
DDDLB.A10	E U	DL10 Test (Part 2)
DXDLC.BIN		DL10 Test (11 side for DDDLB)
UTILITIES		
DDDLG.A10	E U	Bootstrap and Dump PDP-11s via DL10
DDDLH.A10	E	Exec Mode Downline Loader for DL10

Table 16 DC10 (Communications) Diagnostics

Diagnostic	Mode	Title
DDDCA.A10	E U	DC10 Diagnostic

Table 17 Terminal Diagnostics

Diagnostic	Mode	Title
DDLAA.A10	E U	LA36 Terminals
DDLTA.A10	E U	Teletype Test
DDVTA.A10	E U	Video Terminals

Table 18 Miscellaneous Peripheral Equipment Diagnostics

Diagnostic	Mode	Title
DDDKA.A10	E	DK10 Real-Time Clock Diagnostic
DDDTA.A10	E	DT03-CC Switch Controller (Part 1)
DDDTB.A10	E	DT03-CC Switch Controller (Part 2)
DDDMA.A10	E	DMA10 Memory Link (10 side) Diagnostic
DXDMB.A10	E	DMA10 Memory Link (11 side) Diagnostic
DDIPA.A10	E	IP10-DB10 Interprocessor Buffer Diagnostic

6 For multiple tapes (only), when the reader halts after one part of the tape has read in, do the following.

a. Remove the tape from the reader.

b. Place the next tape in the series in the reader. Leave approximately two inches of blank tape before the first punched holes.

c. Press the CONTINUE key. The tape will continue to read in.

Repeat step a through c until all tapes in the series have been read in. Then review step 5.

NOTE
DO NOT press the RESET key during multiple tape readins as this will cause a system reset and clear the readin operation.

LOADING FROM DECTAPE - Refer to Procedure 2
A bootstrap readin of the 10-Based 10 Maintenance Library can be performed from either the DTA or DTB DECtape subsystem. A DECtape containing DTBOOT in the boot block (block 0) must be mounted on transport 0.

The readin hardware rewinds the DECtape to block 0. A DATAI DTn, 0: is then executed which causes the first word of block 0 (a BLKI pointer) to be deposited into location 0. The DATAI is then converted to a BLKI and the DTBOOT program is read into core and started.

Procedure 2 Loading the 10/10 Library from DECtape (KA10-KI10)

Step	Procedure
1	Mount a DECtape containing DTBOOTS on transport 0. The DECtape may also contain the program(s) to be loaded.
2	Mount the DECtape containing the program(s) to be loaded on a transport which is in the same DECtape subsystem. This step may be omitted if the DECtape containing the program(s) to be loaded also has a copy of DTBOOTS in the boot block (block 0).
3	Write-protect and enable all transports to be used.
4	Set the DEVICE SELECTION switches. 320 for the DTA subsystem 330 for the DTB subsystem
5	Set the console data switches to 0.
6	Press the RESET key. This will initialize the system.
7	Press the READ IN key. The DTBOOTS program will automatically read in and start.
8	To DTBOOTS type: n:<cr> Where n equals the DECtape transport to be used by DTBOOTS as the load device.
9	Next to DTBOOTS type: file.ext<cr> Where file.ext equals the name of the program to be loaded by DTBOOTS. Usually the first program loaded is the diagnostic monitor DIAMON.

LOADING FROM MAGTAPE - Refer to Procedure 3
A boot load readin of the 10-Based 10 Maintenance Library can be performed from a TM10, DX10 or RH10 tape control unit. The following is a description of the operation of each of these control units.

Via TM10 or TM10B
A readin can be performed from either the MTA or MTB magtape subsystem. The control unit may be either a TM10A or TM10B, and the TM10B can have a DF10C in either KA or KI mode. The diagnostic magtape, however, must be mounted on drive 0 of the selected TM10 for readin, and TM10s must be ECOed to readin at 800 bits/in.

The hardware dumps the readin record into memory starting at location 1000. The program is then started.

The readin program first determines which TM10 was used, then skips the tape forward over two file marks and reads the file MAGMON.SAV. This program is loaded into core and then started.

Via DX10 (TU7x)
A readin can be performed from any of three DX10s on a system. The readin activates a bootstrap in the DX10 which selects the first on-line tape drive, rewinds it and dumps the readin record into memory starting at location 0. The PDP-10 is then started at location 100. The tape drive must be 9-track. The diagnostic magtape can be written in either 800 bits/in or 1600 bits/in.

The readin record first relocates itself to the memory locations where the TM10 would dump it. The DX10 which caused the readin is found by looking for a DX10 that is running. The bootstrap code in the DX10 memory is then modified to read the DX10 microcode (the remaining records in the readin file on the magtape) into PDP-10 memory. The microcode is then written into the DX10 memory and started.

The MAGMON.SAV file is then read and started using the normal DX10 microcode.

Via RH10/TM02
A readin can be peformed by any TM02 controller on any of six RH10s. The TM02 number must be selected by the switch on the maintenance panel of the RH10. The tape unit must be slave 0 and the diagnostic magtape must be written at 800 bits/in.

NOTE
A DF10C must be in KA mode or the readin will not work.

The RH10 requests a readin operation of the TM02 which rewinds the tape drive and then reads the bootstrap into memory and starts it. The bootstrap program then reads in and starts the magtape monitor MAGMON SAV.

Procedure 3 Loading the 10/10 Library from Magtape (KA10-KI10)

Step	Procedure
1	Mount 10/10 Maintenance Library magtape on transport 0 of the subsystem to be used. In the case of DX10s the first on-line transport should be used.
2	Make the transport READY and ON-LINE.
3	Set the DEVICE SELECTION switches TM10(B) DX10(TU7x) RH10-TM02 RH10-TM02 340-MTA 220 DX10#0 270 RH10#0 364 RH10#3 350-MTB 224 DX10#1 274 RH10#1 370 RH10#4 034 DX10#2 360 RH10#2 374 RH10#5
4	RH10/TM02 (only). Set the maintenance panel to select the TM02 to be used. The tape must be mounted on slave 0.
5	Clear the console data switches.
6	Press the RESET key. This will initialize the system.
7	Press the READ IN key. MAGMON, the diagnostic monitor for magtape, will automatically read in and start. Refer to the MAGMON command summary.

LOADING FROM DISK PACK - Refer to Procedure 4 or 4A.
A boot load readin of the 10-Based 10 Maintenance Library can be performed from an RP10 or RH10 control unit. The disk pack used must conform to the standard TOPS-10 file structure. BOOTS, the disk pack bootstrap loader program, must be in the boot block (block 0). If BOOTS is not in the boot block, procedure 4A must be used.

Via RP10 or RP10C
A readin can be performed using either the DPA, DPB, DPC or DPD disk subsystem. The control unit may be either an RP10 or an RP10C and may use either a DF10 or DF10C data channel. The disk pack must be mounted on drive 0 of the selected subsystem. The READ IN key causes the BOOTS (block 0 of the disk pack) to be loaded into core through the data channel and started. The BOOT program is then used to load and start a diagnostic monitor (usually DIAMON). The monitor is then used to load and sequence other 10-based 10 diagnostic or utility programs.

Via RH10
A boot load operation can be performed using any one of up to six RH10 control units. In addition, any one of up to eight (0-7) disk units may be used. The disk drive unit must be selected on the maintenance panel of the RH10. Functionally the readin operation for RH10s works the same as the readin operation for RP10s.

Procedure 4 Loading the 10/10 Library from Disk (KA10-KI10)

Step	Procedure
1	Mount the disk pack containing the 10/10 Maintenance Library on drive 0. For RH10-controlled subsystems, any drive (0-7) may be used. The pack must conform to the TOPS-10 file structure and must have a copy of BOOTS in the boot block (block 0).
2	Make the disk unit READY and ON-LINE
3	Set the DEVICE SELECTION switches RP10 or RP10C / RH10-RP04/5/6 / RH10-RP04/5/6 250 DPA / 270 RH10#0 / 364 RH10#3 254 DPB / 274 RH10#1 / 370 RH10#4 260 DPC / 360 RH10#2 / 374 RH10#5 264 DPD
4	RH10 (only). Set the maintenance panel switches to select the drive to be used.
5	Clear the console data switches.
6	Press the RESET key. This will initialize the system.
7	Press the READ IN key. BOOTS will automatically read in and start. Refer to the BOOTS command summary for a complete description of the BOOTS commands.
8	To BOOTS, type the name of the disk structure, program name, and project programmer number to be used (e.g., MAINT: DIAMON.SAV[6,10]<cr>)

Procedure 4A Loading the 10/10 Library from Disk (KA10-KI10)

Step	Procedure
1	Mount the KLAD-10 maintenance pack on a disk drive. The KLAD-10 pack is being used only as an example; actually, any maintenance pack which does not have BOOTS in the boot block can be used.
2	Make the drive READY and ON-LINE.
3	Load a copy of BOOTS from any of the following. Paper tape (Refer to Procedure 1.) DECtape (Refer to Procedure 2.) Any other disk pack (Refer to Procedure 4)
4	To BOOTS, type: KLAD:DIAMON.SAV[6,10]<cr> BOOTS will automatically load and start the program specified, usually DIAMON, the diagnostic monitor.

Procedure 5 Loading the 10/10 Library via KLDCP (KL10 Only)

Step	Procedure
1	Load and start KLDCP. Refer to the 11/10 STD module.
2	Select the KLDCP load device and mount the medium containing the 10-based 10 programs to be run.
3	To KLDCP type BT<CR> This initializes the KL10 to run 10-based 10 programs.
3A	To KLDCP type B<CR> This is an alternative to the BT command. It initializes the KL10 and runs both the 11/10 and 10/10 processor functional diagnostics.
4	To KLDCP type one of the following. I file.CCL<CR> Run the specified KLDCP indirect file. J file.CMD<CR> Run the specified KLDCP double indirect file. P file.ext<CR> Run the specified diagnostic program. STD<CR> P file.ext<CR> Load and start the specified 10-based 10 STL<CR> diagnostic monitor. The monitor is then used to run additional 10-based 10 programs.

The following further explains Procedure 5.

STEP 1 - Refer to the 11-Based 10 Maintenance Library for KLDCP loading instructions.

STEP 2 - In most cases this will be either a floppy disk, DECtape, or KLAD-10 pack. Mount the medium, make the unit ready and on line, and type one of the following KLDCP commands.

DTn<CR> Selects DECtape transport n.
RXn<CR> Selects floppy unit n.
RPn<CR> Selects disk pack drive n. This drive should be dual-ported.

STEP 3 - The BT command causes KLDCP to execute the BT.CMD command files for KL10 model PA mainframes and the BBT.CMD command file for KL10 model PV mainframes. The command files direct KLDCP to load the appropriate microcode, configure memory, start the

microcode, clear the first 256K of KL10 memory, load the subroutine package, load KLDDT and finally start the subroutine package. The BT command is fast; it requires less than ten seconds for execution and should be used to initialize the KL10 when the mainframe is known to be fully operational.

STEP 3A - The B command causes KLDCP to execute the B.CMD command file for KL10 model PA mainframes and the BB.CMD command file for KL10 model PV mainframes. The command file first directs KLDCP to run the 11-based 10 diagnostics. Then it directs KLDCP to initialize the KL10 to run 10-based 10 programs and finally the command file directs KLDCP to run the 10-based 10 processor functional diagnostics.

The B command requires approximately twenty minutes for executions and should be used when the operational status of the KL10 mainframe is unknown or questionable. The twenty-minute wait can be used for power supply or other maintenance checks.

STEP 4 - Once the KL10 has been initialized, one of two methods may be used to load and run 10-based 10 diagnostics. The preferred method is to continue to use KLDCP. The method requires that the EBox, MBox, main memory, and DTE20 data paths be fully operational. This can be verified, if necessary, by using the 11-Based 10 Maintenance Library.

This method is preferred because it does not rely on any untested KL10 data paths or I/O subsystems for program loading. Each program loaded and run by KLDCP is read from the KLDCP load device and buffered in the console front-end subsystems main memory. A checksum operation is then performed and the .A10 file is converted from the ASCII A10 format to KL10 machine language. The program is then transferred through the DTE20 to KL10 memory for execution.

The second method of loading and running 10-based 10 diagnostic programs is to use one of the 10-based 10 diagnostic monitors. This method has two minor advantages and one major disadvantage over using KLDCP. The advantages are (1) speed (the program is loaded faster) and (2) convenience, (the user can take advantage of existing control files). The disadvantage is that diagnostic monitors require the use of a (possibly untested and unreliable) I/O subsystem for program loading purposes. For this reason it is recommended that KLDCP be used whenever possible to load and run 10-based 10 diagnostic programs.

USER MODE

Most systems allocate a 5000-block area for use by field service during timesharing. TOPS-10 operating systems usually designates the [60,60] area for this purpose; TOPS-20 operating systems usually designate the F-S area. The area is used to store the most commonly used user-mode maintenance programs. The less used programs are usually stored on a separate maintenance pack (e.g., KLAD) or on a diagnostic magtape. The procedures which follow:

User Mode Diagnostics (TOPS-10) Procedure 6
User Mode Diagnostics (TOPS-20) Procedure 7

list the steps involved in running maintenance programs in user mode. Both procedures assume that the user is logged in, that the program to be run is not in the [60,60] or F-S area, that special user-mode privileges are required to run the program, and that the device to be tested must be assigned to the user's job. The user's responses are underscored and a brief description of each step follows the individual procedures.

NOTE

To output error messages to a user-specified device instead of the CTY, assign to the desired output device the logical name DEV. Any device that can accept ASCII output may be used (LPT, DSK, DTA etc.). The output file is given the name of the diagnostic and the extension LPT (e.g., DDTMH.LPT).

The output file (if used) may then be listed by using the normal monitor commands (PRINT, LIST, TYPE, PIP, etc.).

If the program is aborted before completion (by ^C, etc.) the output file may be closed by using the monitor REENTER command.

Procedure 6 Running User Mode Diagnostics Under TOPS-10

Step	Command or System Response
1	↑C
2	.MOUNT KLAD<CR> REQUEST QUEUED WAITING...2 C'S TO EXIT KLAD MOUNTED
3	.MOUNT MTB1:FIELD/REEL:TEST/WE<CR> REQUEST QUEUED WAITING...1 C'S TO EXIT FIELD MOUNTED, MTB261 USED
4	.R WHEEL<CR> SETTING WHEEL CAPABILITY
5	.RUN DIAMON<CR> *DIAMON [DDQDC] - DIAGNOSTIC USER & EXEC MONITOR VER 0.12 HELP ? - <CR>
6	DIAMON CMD - DDTMH<CR> DDTMH TM10A AND TM10B MAG-TAPE RELIABILITY (USER & EXEC MODE) VERSION 0.3, SV-0.13
7	TTY SWITCH CONTROL ? - 0,S,Y OR N <CR> - 0<CR> SWITCHES = 000000 000000 MEMORY MAP = FROM TO SIZE/K 00000000 00077777 32 WANT MONITOR TO ATTEMPT ERROR RECOVERY? Y OR N <CR> - N<CR> WANT ERRORS REPORTED TO SYSTERR? Y OR N <CR> - N<CR> CHANNEL #0 - MTB1
8	TYPE "L" FOR TEST LIST WHAT TEST - RTEST<CR>
9	PARITY ERROR CHANNEL #0 (WRITE) - RECORD #14 PHYSICAL DEVICE = MTB1 FILE STATUS = 100117 WHICH EQUALS: DENSITY = SYSTEM STANDARD PARITY ERROR, NO RETRY ******
10	↑C

STEP 1 - The user enters TOPS-10 command mode.

STEP 2 - The user requests that KLAD, a maintenance pack, be mounted and entered into his search list.

STEP 3 - The user requests the magtape transport to be tested, and that MTB1 be assigned, given the logical name FIELD with a READ ID TEST and be write-enabled.

STEP 4 - The user requests special privileges by running the TOPS-10 system program WHEEL.

STEP 5 - The user runs DIAMON, the TOPS-10 (file structure) diagnostic monitor.

STEP 6 - The user directs DIAMON to run DDTMH, a user-mode magtape diagnostic.

STEP 7 - The user directs DDTMH to use a control switch setting of 0, disable monitor error recovery and disable error reporting to SYSERR.

STEP 8 - The user directs DDTMH to execute RTEST.

STEP 9 - DDTMH detects and reports a magtape parity error.

STEP 10 - The user exits DDTMH and returns to TOPS-10 command mode.

Procedure 7 Running User Mode Diagnostics Under TOPS-20

Step	Command or System Response
1	↑C
2	@ ENABLE<CR>
3	$ SMOUNT KLAD:,<CR> $$ STRUCTURE ID (IS) PS:<CR> STRUCTURE KLAD:MOUNTED
4	$ TMOUNT FIELD:<CR> [OPERATOR NOTIFIED] [MTA3:ASSIGNED]
5	$ RUN D20MON<CR> *D20MON [DDQDH] - DECSYSTEM20 DIAGNOSTIC MONITOR - VER 0.1*
6	D20MON CMD - DFTUK<CR> 'VERSION 0.1, SV=7.0 CPU#=2102
7	TTY SWITCH CONTROL ? - 0,S, Y OR N <CR> - 0<CR> SWITCHES = 000000 000000 MEMORY MAP = FROM TO SIZE/K 00000000 00102777 33 TYPE MTAPE TO BE TESTED IN THE FORM "MTAN:<CR>" : MTA3: TM03 (Y=TM03, N=TM02) ? Y OR N <CR> - N<CR> SLV TYPE (TU16, TE16 OR TU45): TU45<CR> CONFIGURATION: RH20# 3 DEVICE CODE= 554 TM02# 1 TU45# 3 SN= 0148.
8	WHAT TEST (H<CR> FOR HELP): R1<CR>
9	1600 BPI RELIABILITY TEST TM02/TU45 1/3 WRT TAPE ERROR; REC# 38. ORIGINAL ERROR... REPEATING... RECOVERED (#REPTS=4) (WILL RETRY ONCE MORE) WITH 0. RETRIES RECOVERED (NO CURRENT REPEATS) WITH 1. RETRY
10	↑C
11	$ DISABLE<CR> @ SDISMOUNT (FILE STRUCTURE) KLAD:<CR> STRUCTURE KLAD:DISMOUNTED @

STEP 1 - The user enters TOPS-20 command mode.

STEP 2 - The user requests that special privileges be enabled. The prompt changes to an ($). Special privileges are required to mount a second public structure such as KLAD-20.

STEP 3 - The user requests the KLAD-20 pack be mounted.

STEP 4 - The user requests the magtape transport to be tested be assigned his job. The operator makes the actual physical assignment. Therefore, the user must tell the operator what transport he wants assigned for test purposes.

STEP 5 - The user run D20MON, the TOPS-20 (file structure) diagnostic monitor.

STEP 6 - The user directs D20MON to run the DFTUK magtape diagnostic.

STEP 7 - The user directs DFTUK to use a control switch setting of 0 and indicates that the magtape subsystem consists of a TM02/TU45 combination.

STEP 8 - The user directs DFTUK to execute R1 TEST.

STEP 9 - DFTUK detects a write tape error.

STEP 10 - The user exits DFTUK and returns to TOPS-20 command mode at enable level.

STEP 11 - The user returns to the normal command level and dismounts the KLAD-20 pack.

STANDARD ERROR FORMATS
Most 10-based 10 diagnostic programs use one of the following standard error message formats. Unique error message formats are described in the individual diagnostic summary.

Fatal Halts
The following fatal halt addresses are used to report total inoperation of the PDP-10 processor. If any of these halts occurs, run a more basic diagnostic which does not use the SUBRTN package.

Address	Tag	Reason
1010	NOEXEC	Program not coded for EXEC mode operation
1011	PLERR	Fatal push list pointer error
1012	PLERR1	Initial push list pointer incorrect
1013	MUOERR	MUUO with LUUO handler wiped out
1014	DTEBER	DTE20 interrupt without doorbell
1015	DTECER	DTE20 clock interrupt without flag set
1016	CPIERR	CPU initialization error
1017	EOPERR	End of program error
1020	LUOERR	Interrupt with LUUO handler wiped out

The following message is typed and a halt (exit) is executed.

FATAL PROGRAM ERROR AT ######

points to the PC of the error handler. Program execution should not be continued until the problem has been corrected.

General Error Message Format
The general error message format specifies up to six items concerning a test failure.

They are as follows.

1. Name of the major test sequence being executed.
2. Name of the function being tested.
3. The correct test data results.
4. The actual test data results.
5. Diagnostic comment concerning the test failure.
6. Additional error print routine.

A typical printout using the general error message format follows:

```
TEST PASS COUNT = #
PC=XXXXXX
ERROR IN TITLE-FUNCTION
CORRECT:   XXXXXX XXXXXX
ACTUAL:    YYYYYY YYYYYY
DISCREP:   ZZZZZZ ZZZZZZ
DIAGNOSTIC COMMENT
ADDITIONAL COMMENT
```

Test pass count is only printed on errors which occur when the pass counter is non-0.

PC is the absolute address of the error call instruction.

DISCREP is the octal discrepancy between the CORRECT and ACTUAL test data. (DISCREP is the XOR of C and A).

TITLE, FUNCTION, and DIAGNOSTIC COMMENT portions of the error typeout may be inhibited by setting the TXTINH switch (10). This allows for shorter printouts on repetitive failures.

GENERAL INFORMATION

Code	DDQDC.SAV
Title	DIAMON - DECsystem-10 Diagnostic monitor
Abstract	DIAMON is the basic 10/10 diagnostic monitor. It runs in either exec or user mode. In exec mode DIAMON can load and sequence program from any of the following. Paper tape DECtape (either PDP-10 or PDP-11 format) Disk pack (using a TOPS-10 file structure). In user mode, DIAMON will run under TOPS-10 (only). The load medium is restricted to disk. DIAMON is command-controlled and can be directed to load and run a single program or execute a control file which will direct DIAMON to run a sequence of programs. Control files enable DIAMON to be used for the following purposes. Rapid checkout of the hardware Acceptance testing Reliability testing Unattended overnight testing DIAMON also supports KI10 margining and special user mode operations.
Hardware Required	KA10, KI10 or KL10 mainframe/32K of core (minimum)/load device: paper tape, DECtape, disk, or console load device (KL10 only).
Preliminary and Associated Programs	DIAMON assumes that the basic instructions and the selected load device are operational.
Restrictions	The dianostic monitor may be used to call only those programs which follow the prescribed diagnostic formats.
Notes	1. If the monitor fails to operate, use the diagnostic programs individually to isolate the problem. 2. The DECsystem subroutine program and DDT are automatically loaded on system startup or device specification if they are not already resident in the PDP-10 memory.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	The state of the control switches does not affect the operation of DIAMON unless a control file is being used. A control file lists, as part of each command line, the program to be run and the right half switches to use with that program. This allows the actual (console) right hand switches to be used to control the operation of DIAMON. The switches which affect the operation of DIAMON when a control file is in use are listed in Table 1.

OPERATIONAL CONTROL
After the diagnostic monitor is started it will type the following message:

DIAMON - DECSYSTEM DIAGNOSTIC MONITOR

DEV: T, K, D, V, P -

In user mode, the disk is automatically selected and this question is not asked.

Table 2 describes the device selection commands.

After selection of the load device DIAMON will automatically load SUBRTN and KLDDT and print:

DIAMON CMD -

Table 3 describes general DIAMON commands.
Table 4 describes program starting commands.
Table 5 lists DIAMON manual starting addresses.

Table 1 DIAMON Control Switch Summary

Switch	State	Descriptions
9	0	Reduces the iteration count in a control file by a factor of 100 to 1, thus reducing the run time for each program in the file. This is useful for a quick check of the hardware and margining operations.
	1	Each program listed in a control file is run the specified number of iterations.
15	0	Normal operation
	1	Inhibit printing the test title of each program executed by DIAMON.
18	0	Normal operation
	1	Expand the basic command set to include margining and special user mode operations. Refer to the X command, Table 3.

Table 2 DIAMON Load Medium Selection Commands

Command	Description
D	D<CR> Indicates to DIAMON that a PDP-10 formatted DECtape is to be used as the load medium. DIAMON will request the DECtape unit number. Type: 0 - 7 to indicate which DECtape unit contains the program(s) to be run. S to direct DIAMON to search all mounted and selected DECtapes to find the program(s).
K	K<CR> Use the load device selected by KLDCP. This response is only valid for KL10-based systems.
P	P<CR> Selects a disk unit as the load medium. DIAMON will request the disk name and the [P,PN] project, programmer number to use as follows: DISK: [P,PN] Typing a <CR> will cause DIAMON to use the default. The default to DISK will cause a pack search from KLAD and DSKA to DSKO. The default [P,PN] is [6, 10].

Table 2 DIAMON Load Medium Selection Commands (Cont)

Command	Description
T	T<CR> Selects the paper tape reader as the load medium.
V	V<CR> Indicates to DIAMON that a PDP-11 formatted DECtape is to be used as the load medium. DIAMON will request the DECtape unit number. Type: 0 - 7 to indicate which DECtape unit contains the program(s) to be run. S to direct DIAMON to search all mounted and selected DECtapes to find the program(s).

Table 3 Diamon General Command Summary

Command	Description
<CR>	Standard command terminator.
$	Altmode - a special command terminator which causes a single program to be loaded but not started.
↑Z	A control Z is used to terminate the T command.
D	D<CR> Directs DIAMON to read a control file from the load medium. DIAMON will respond by printing FILE.EXT-. Respond by typing the name of the control file.
F	F<CR> Directs DIAMON to print a directory of the load medium.
G	G<CR> Directs DIAMON to start or restart execution of the program currently loaded in core.
I	I<CR> Directs DIAMON to begin execution of the control file currently in core.
L	L<CR> Directs DIAMON to print a file stored on the load medium. DIAMON will request the name of the file to be printed by printing FILE.EXT-.
R	R<CR> Directs DIAMON to reinitialize itself. DIAMON will begin by requesting the load medium to be used.
S	S<CR> Directs DIAMON to load a single program. DIAMON will request the name of the program by printing FILE.EXT-. This question may be answered with only the file name as the extension will default to .A10 or .SAV unless the console load device is selected, and then the default is .A10.
T	T<CR> Directs DIAMON to open a buffer and begin building an internal control file. A control Z (↑Z) terminates the T command. Refer to section on building control files which follows Table 5.
X	X<CR> Directs DIAMON to run through the expanded command set dialogue.

EXPANDED COMMAND DIALOGUE
The following additional command sequences are added when either the X command is used or the XPAND switch (18) is set.

TYPE Y FOR CLOCK MARGINS -

    <CR> = no margins
    Y<CR> = clock margins

Clock margins speeds up the basic clock cycle by 10 percent.

TYPE Y OR A FOR SPECIAL USER MODE -

    <CR> = no
    A<CR> = special user mode after first pass
    Y<CR> = special user mode on all passes

    Special user mode is a pseudo-user mode where the diagnostic program being run is run in user mode (with paging, etc.) and the I/O is trapped back to the diagnostic monitor for processing. This provides a method of checking user mode operation with functional and reliability diagnostics without actually having to use a monitor and timesharing.

The following additional question is asked if special user mode is selected.

TYPE Y OR A FOR CONCEALED MODE -

    <CR> = public mode
    Y<CR> = concealed mode
    A<CR> = alternate public and concealed modes

TYPE L, S, I, R FOR VOLTAGE MARGINS -

    <CR> = No margins

    L<CR> = Limits. Margins are done at the ±.25 V settings only.

    S<CR> = Sweep margins. The 5 V power supplies are varied by 1-increment steps (21 MV) up to and down to the ±.25 V limits.

    I<CR> = Increment. The user may specifiy the margin step per program pass. If I is typed then the following question will be asked.

            SPECIFY MARGIN INCREMENT (1 TO 17) -

            One increment step equals 21 millivolts (e.g., 4 would specify an 84-millivolt increment).

    R<CR> = Rack. Sweep margins are run on the processor logic rack specified. The following question is asked.
            SPECIFY RACK (0 to 37) -

            Type rack number to be margined.

Margins provide the control necessary to operate the KI10 programmable margin system. Answer the above question(s), then:

SET 'MARGIN SELECT' OFF
SET 'MARGIN ENABLE' SWITCH
TYPE ANY CHAR WHEN READY!

Table 4 DIAMON Program Starting Commands

Command	Description
DDT	DDT<CR> Start DDT
PFSTRT	PFSTRT<CR> Power fail restart
REE	REE<CR> Reenter (user mode)
SFSTRT	SFSTRT<CR> Special features start.
START	START<CR> Start diagnostic
START#	START3<CR> Special start. Numbers range from 1 through 5.
STD	STD<CR> Start diagnostic
STL	STL<CR> Start DIAMON
STM	STM<CR> Reinitialize start

Table 5 Standard Manual Starting and Restarting Addresses

Address	Description
20000	DIAMON starting address
20001	If it is desired to abort a test currently in progress or to restart at the next sequential program, the operator may do so by starting at location 20001.
20002	If the diagnostic monitor is running in the mode where titles are not printed [SW 15(1)] and a user program fails such that it is not known which program failed, starting at location 20002 will cause the title to be printed. The computer will then halt at location 20000. The operator may at this time manually restart the user program or restart the diagnostic monitor.
20003	Program starting and restarting address.

DIAMON CONTROL FILES
A control file for DIAMON is an ASCII file consisting of a list of programs to be run. The following apply to constructing a DIAMON control file.

1. A control file can be constructed with any editor program or via the DIAMON T command.

2. A control file can have up to 50 command lines.

3. Each command line consists of five items each separated by a space or tab. The items are as follows.

   a. Program name. If the program name includes an extension, the extension must be included and separated by a period.

      NOTE
      If the special user mode routines are selected, a line that starts with a minus (-) signifies that the program will run in special user mode.

   b. Pass count. The pass count is the number of passes that the program is to run. The pass count may be in the range 0 to 777777. If 0, the program will run on each pass through the control file.

   c. Switches. This is an octal half word (6 digits) to be used by the program as the right half of the console data switches.

   d. Iterations. This is the number, in octal, of iterations the program is to be executed. The iteration count may be in the range 0 to 377777. If 0, one iteration is assumed.

   e. <CR>. A carriage return terminates the command line and opens the next line for input. If the T command was used to build the control file, a ↑Z (control Z) will close the file and return to DIAMON command mode.

   Example:

```
DEKAA.A10    10    0         1000<CR>
DEKAB.A10     1    123456    200<CR>
DEKAC.A10     0    00001     1<CR>
↑Z
```

4. If the control file is being generated via the T command the following headers will be printed. These act as a guide only and are not actually a part of the control file.
   NAME PASSES RH SWS ITERATIONS

5. Typing errors may be corrected by typing a RUBOUT. The RUBOUT will print three Xs and delete the entire line.

The control file is executed via the I command. The diagnostic monitor will read in and execute the first program on the command list. The program will be iterated the requested number of times and control will then revert to the monitor. The monitor will then proceed to the next program on the list until all programs requested have been executed. When the final program on the command list has been executed, the pass count will be printed and then the monitor will restart with the first program again.

EXAMPLE:

```
DIAMON PASS 000001
DIAMON PASS 000002
etc.
```

A control file will remain in core so that if the monitor is restarted the command list does not have to be read in again unless a new control file or single program is selected.

To use the same control file type I.

DIAMON ERROR SUMMARY

CMD'S REQUIRED
The program was commanded to execute the control file, but the list is empty. Input some programs to execute.

Disk Pack Errors
Any disk pack errors will print out the reason, ERROR AT and the octal address of the error. Consult the listing for error explanation.

Margin Errors
If margins are selected and the MARGIN ENABLE switch is not set on startup, the margin setup message will be repeated.

'MARGIN ENABLE' NOT SET?
If during a margin run the MARGIN SELECT switch is reset, the above message will be printed. All subsequent programs will be run in normal mode.

MUUO ERROR
If the diagnostic program being run (in special user mode) causes an MUUO, (not trapped I/O) the above error message will be printed and the program will halt. The operator may examine the user MUUO locations (17424 and 17425) to determine the cause of the error.

PROGRAM NOT FOUND - PROG.EXT
The program requested is not on the load device.

USER TRAP ERROR
If the diagnostic program being run (in special user mode) causes a trap (PAGE FAIL, PUSHDOWN OVERFLOW or TRAP 3), the above message will be printed and the program will halt.

GENERAL INFORMATION

Code	DDQDH.EXE
Title	DECSYSTEM-20 Diagnostic Monitor
Abstract	D20MON is a variation of DIAMON which has been modified to handle TOPS-20 file structures. It will run in either exec or user mode. In exec mode D20MON can load and sequence programs from disk or the KLDCP load device. In user mode the load device is restricted to disk only.
Hardware Required	KL10 mainframe/32K of core (minimum)/load device: 1. KLDCP - KL10 only, use KLDCP selected device. 2. Disk pack, RP04/5/6 RM03 on RH10 or RH20.
Preliminary and Associated Programs	D20MON assumes that the basic instructions and the selected load device are operational.
Restrictions	The diagnostic monitor may be used to call only those programs which follow the prescribed diagnostic formats.
Notes	1. If the monitor fails to operate, use the diagnostic programs individually to isolate the problem. 2. The DECsystem SUBRTN package and DDT are automatically loaded on system startup or device specification if they are not already resident in the PDP-10 memory.
Loading and Starting Procedure	Via KLDCP type: P D20MON<CR> STL<CR> Via TOPS-20 type: RUN D20MON<CR>
Control Switches	The state of the control switches does not affect the operation of D20MON unless a control file is being used. A control file lists, as part of each command line, the program to be run and the right half switches to use with that program. This allows the actual (console) right hand switches to be used to control the operation of D20MON. The switches which affect the operation of D20MON when a control file is in use are listed in Table 1.

OPERATIONAL CONTROL

After the diagnostic monitor is started it will type the following message:

```
*D20MON - DECSYSTEM-20 DIAGNOSTIC MONITOR*
DEV:
```

In user mode, the disk is automatically selected and this question is not asked.

Table 2 describes the device selection commands.

When the disk pack is selected as the load device the monitor operates from the DISK:<DIRECTORY> that is specified. The default disk is PS: and the default directory is <DIAGNOSTICS>. To use the defaults type a <CR>.

After selection of the load device, D20MON will automatically load SUBRTN and KLDDT and print:

```
D20MON CMD -
```

Table 3 describes general D20MON commands.
Table 4 describes program starting commands.
Table 5 lists D20MON manual starting addresses.

Table 1 D20MON Control Switch Summary

Switch	State	Description
9	0	Reduces the iteration count in a control file by a factor of 100 to 1, thus reducing the run time for each program in the file. This is useful for a quick check of the hardware.
	1	Each program listed in a control file is run the specified number of iterations.
15	0	Normal operation
	1	Inhibit printing the test title of each program executed by D20MON.
18	0	Normal operation
	1	Expand the basic command set to include margining and special user mode operations. Refer to the X command, Table 3.

Table 2 D20MON Device Selection Commands

Command	Description
K<CR>	Use the load device selected by KLDCP.
?<CR>	Will cause a list of all available disk structures to be printed.
dev:<CR>	Use the disk specified by dev as the load medium (e.g., KLAD20:<CR>).
dev:?<CR>	Will cause the master directory for the disk specified by dev: to be printed.
<CR>	Will default to the public structure (same as typing PS:<CR>).

Table 3 D20MON General Command Summary

Command	Description
<CR>	Standard command terminator.
$	Altmode - a special command terminator which causes a single program to be loaded but not started.
↑Z	A control Z is used to terminate the T command.
D	D<CR> Directs D20MON to read a control file from the load medium. D20MON will respond by printing FILE.EXT-. Respond by typing the name of the control file.
F	F<CR> Directs D20MON to print a directory of the load medium.
G	G<CR> Directs D20MON to start or restart execution of the program currently loaded in core.
I	I<CR> Directs D20MON to begin execution of the control file currently in core.
L	L<CR> Directs D20MON to print a file stored on the load medium. D20MON will request the name of the file to be printed by printing FILE.EXT-.
R	R<CR> Directs D20MON to reinitialize itself. D20MON will begin by requesting the load medium to be used.

Table 3 D20MON General Command Summary (Cont)

Command	Description
S	S<CR> Directs D20MON to load a single program. D20MON will request the name of the program by printing FILE.EXT-. This question may be answered with only the file name as the extension will default to .A10 or .SAV unless the console load device is selected and then the default is .A10.
T	T<CR> Directs D20MON to open a buffer and begin building an internal control file. A control Z (↑Z) terminates the T command. Refer to section on building control files which follows Table 5.
X	X<CR> Directs D20MON to run through the expanded command set dialogue.

EXPANDED COMMAND DIALOGUE

The following additional command sequences are added when either the X command is used or the XPAND switch (18) is set.

TYPE Y OR A FOR SPECIAL USER MODE -

<CR> = no
A<CR> = special user mode after first pass
Y<CR> = special user mode on all passes

Special user mode is a pseudo-user mode where the diagnostic program being run is run in user mode (with paging, etc.) and the I/O is trapped back to the diagnostic monitor for processing. This provides a method of checking user mode operation with the functional and reliability diagnostics without actually having to use a monitor and timesharing.

The following additional question is asked if special user mode is selected.

TYPE Y OR A FOR CONCEALED MODE -

<CR> = public mode
Y<CR> = concealed mode
A<CR> = alternate public and concealed modes

Table 4 D20MON Program Starting Commands

Command	Description
DDT	DDT<CR> Start DDT
PFSTRT	PFSTRT<CR> Power fail restart
REE	REE<CR> Reenter
SFSTRT	SFSTRT<CR> Special features start
START	START<CR> Start diagnostic
START#	START3<CR> Special start. Numbers range from 1 through 5.
STD	STD<CR> Start diagnostic
STL	STL<CR> Start D20MON
STM	STM<CR> Reinitialize start

Table 5 Standard Manual Starting and Restarting Addresses

Address	Description
20000	D20MON starting address
20001	If it is desired to abort a test currently in progress or to restart at the next sequential program, the operator may do so by starting at location 20001.
20002	If the diagnostic monitor is running in the mode where titles are not printed [SW 15(1)] and a user program fails such that it is not known which program failed, starting at location 20002 will cause the title to be printed. The computer will then halt at location 20000. The operator may at this time manually restart the user program or restart the diagnostic monitor.
20003	Program starting and restarting address.

D20MON CONTROL FILES

A control for D20MON is an ASCII file consisting of a list of programs to be run. The following apply to constructing a D20MON control file.

1. A control file can be constructed with any editor program or via the D20MON T command.

2. A control file can have up to 50 command lines.

3. Each command line consists of five items each separated by a space or tab. The items are as follows.

   a. Program name. If the program name includes an extension, the extension must be included and separated by a period.

   NOTE

   If the special user mode routines are selected, a line that starts with a minus (-) signifies that the program will run in special user mode.

   b. Pass count. The pass count is the number of passes that the program is to run. The pass count may be in the range 0 to 777777. If 0, the program will run on each pass through the control file.

   c. Switches. This is an octal half word (6 digits) to be used by the program as the right half of the console data switches.

   d. Iterations. This is the number, in octal, of iterations the program is to be executed. The iteration count may be in the range 0 to 377777. If 0, one iteration is assumed.

   e. <CR>. A carriage return terminates the command line and opens the next line for input. If the T command was used to build the control file a ↑Z (control Z) will close the file and return to DIAMON command mode.

   Example:

```
DFKAA.A10    10    0         1000<CR>
DFKAB.A10     1    123456    200<CR>
DFKAC.A10     0    00001     1<CR>
↑Z
```

4. If the control file is being generated via the T command, the following headers will be printed. These act as a guide only and are not actually a part of the control file.

   NAME PASSES RH SWS ITERATIONS

5. Typing errors may be corrected by typing a RUBOUT. The RUBOUT will print three Xs and delete the entire line.

The control file is executed via the I command. The diagnostic monitor will read in and execute the first program on the command list. The program will be iterated the requested number of times and control will then revert to the monitor. The monitor will then proceed to the next program on the list until all programs requested have been executed. When the final program on the command list has been executed, the pass count will be printed and then the monitor will restart with the first program again.

Example:

```
D20MON PASS 000001
D20MON PASS 000002
etc.
```

A control file will remain in core so that if the monitor is restarted the command list does not have to be read in again unless a new command list or single program is selected.

To use the same control file, type I.

D20MON ERROR SUMMARY

CMD'S REQUIRED
The program was commanded to execute the control file, but the list is empty. Input some programs to execute.

Disk Pack Errors
Any disk pack errors will print out the reason, ERROR AT, and the octal address of the error. Consult the listing for error explanation. Disk ECC errors are automatically corrected.

MUUO ERROR
If the diagnostic program being run (in special user mode) causes an MUUO, (not trapped I/O) the above error message will be printed and the program will halt. The operator may examine the user MUUO locations (17424 and 17425) to determine the cause of the error.

PROGRAM NOT FOUND - PROG.EXT
The program requested is not on the load device.

USER TRAP ERROR
If the diagnostic program being run (in special user mode) causes a trap (PAGE FAIL, PUSHDOWN OVERFLOW or TRAP 3), the above message will be printed and the program will halt.

GENERAL INFORMATION

Code	KLAD10.SAV
Title	KL10 KLAD-10 Pack Maker
Abstract	KLAD10 is a TOPS-10 user mode diagnostic utility program used to construct the 6,11 diagnostic area on KLAD-10 disk packs. KLAD10 reads files from the 6,10 diagnostic area of a KLAD-10 disk pack, converts them to 16-bit PDP-11 format, and writes them into the 6,11 diagnostic area.
Restrictions	The 6,10 diagnostic area on the KLAD-10 pack must be intact. KLAD10.SAV requires WHEEL privileges.
Notes	PROCEDURES 1 and 2, restoring the 6,10 and 6,11 area respectively, follow Table 1.
Loading and Starting Procedure	Runs under TOPS-10 only: RUN KLAD10<CR>
Control Switches	None

OPERATIONAL CONTROL
KLAD10 is command-controlled from the user's terminal.

COMMAND SUMMARY
KLAD10 commands are in the form of an independent switch, or a filename followed by switches, or by a control filename followed by an AT sign (@). Refer to Table 1.

Table 1 KLAD-10 Command and Switch Summary

Switch	Description
@	filename@<CR> The control file specified by filename is executed.
CREATE	/CREATE:n<CR> Create a new or supersede an existing front-end directory with n directory entries allocated.
DELETE	file.ext/DELETE<CR> The front-end directory is searched for the file specified by the filename. If found, the file will be deleted from the front-end directory in the [6,11] directory. If file is not found, the user will be notified.
DEV	/DEV:dev<CR> Indicates on what device the front-end directory structure will be located. Device name may be logical, physical or a file structure name.
LIST	/LIST<CR> or /LIST:SUM<CR> Print the front-end directory listing If the LIST switch is appended with :SUM, only a summary of the directory will be printed.
REPLACE	file.ext[P,PN]/REPLACE<CR> Replace file in the 11 directory with a new version.
ALLOC	file.ext/ALLOC:n<CR> Allocate a directory entry with n data blocks for a dummy file specified by filename. No data is transferred. n must be greater than 0.

Table 1 KLAD-10 Command and Switch Summary (Cont)

Switch	Description
ASCII	file.ext[P,PN]/TO11/ASCII<CR> Force data conversion from 7-bit bytes to 8-bit bytes if used with /TO11. Force data conversion from 8-bit bytes to 7-bit bytes if used with /TO10. This switch is assumed (i.e. default) unless the file extension is one of those listed under /BINARY.
BINARY	/BINARY<CR> Cancel data conversion for input files whose file extension is not one of the following. .ABS .BIN .LBO .LDR .MFD .SAV .SYS .BIC .CIL .LDA .LOD .OBJ .SYM .UFD This switch is assumed for files with the above file extensions.
BOOT	file.ext[P,PN]/BOOT<CR> Write the -11 bootstrap program specified by the filename to the hardware boot area of the disk. File must be KLADBT.BIN. :ROM406 If the BOOT switch is appended with :ROM406 the bootstrap program will also be written to disk cylinder 406 block 0 to allow for the old KL10 BM873 ROM.
TO10	file.ext[P,PN] = file.ext/TO10<CR> (output) (input) The front-end directory is searched for the input file specified. If found, the input file is copied to the -10 file system by the output filename. Data conversion is performed unless the file extension is listed under /BINARY switch. Output must not be to the [6,11] area.
TO11	file.ext[P,PN]/TO11 The front-end directory must exist prior to the use of this switch. The -10 directory is searched for the file specified by the file name. If found, the file is copied from the -10 file system to the front-end file system. Data conversion is performed unless the file extension is one of those listed under /BINARY switch. /ASCII and /BINARY are the only switches which may be used with /TO11.

PROCEDURE 1

DIAGNOSTIC DISTRIBUTION MAGTAPE TO KLAD [6,10] RESTORE

.LOG 6,10,<CR> PASSWORD: KL10<CR>	Log in to 6,10 area.
.MOUNT KLAD<CR>	Mount the KLAD pack. This might require OMOUNT mounting by operator.
.R SETSRC<CR> *KLAD<CR> *↑C	Set up search list. KLAD disk only.
.ZERO KLAD:<CR>	Clean out old files.
.AS MTA#:<CR>	Assign a magtape drive. Mount the diagnostic magtape
.REW MTA#:<CR>	Rewind magtape.
.SET DENSITY MTA#:800 BPI<CR> .SET BLOCKSIZE MTA#:512 WORDS<CR> .SKIP MTA#:1 FILE<CR> .COPY MAGTAP.SAV=MTA#:<CR>	Retrieve MAGTAP program.
.RUN MAGTAP<CR>	Run MAGTAP program.
DIAGNOSTIC DISTRIBUTION MAGTAPE CREATOR MAGTAPE DEVICE - MTA#:<CR> WHAT DENSITY - 800<CR>	
COMMAND - GET<CR>	Retrieve all programs from magtape.
COMMAND - UNLOAD<CR>	Unload magtape.
.K/F<CR>	Log out, all done.

PROCEDURE 2

KLAD-10 DISK PACK CREATION PROCEDURES

The following procedure assumes that you have already restored the [6,10] diagnostic area on the KLAD pack from the diagnostic distribution magtape. If not, do Procedure 1 first. If your pack is not KLAD, substitute its name where KLAD is used.

.LOG 6,11<CR> PASSWORD: KL10<CR>	Log in to 6,11 area.
.MOUNT KLAD<CR>	Mount the KLAD pack. This might require OMOUNT mounting by operator.
.R SETSRC<CR>	Set up search list.
*C KLAD<CR> *C/LIB:[6,10]<CR> ↑C	KLAD disk only. Library area from [6,10].
.ASSIGN KLAD: DEV<CR>	Assign logical name DEV to KLAD.
.RUN WHEEL<CR>	Requires privileges to make KLAD.
.RUN KLAD10<CR>	Run the KLAD creation program
COMMAND: P1010@<CR> OR P1011@ .EXIT<CR>	Execute appropriate control file. Done.
.RUN WHEEL<CR>	Release privileges.
.DISMOUNT/REMOVE KLAD<CR>	Remove KLAD pack from system.
.K/F<CR>	Log out.

GENERAL INFORMATION

Code

DDQDE.SAV

Title

MAGMON-DECSYSTEM Diagnostic Magtape Monitor

Abstract

MAGMON is a variation of DIAMON which has been modified to handle magtape as a load medium in both exec and user mode. In exec mode the load medium is restricted to magtape (only). In user mode MAGMON will run with either the TOPS-10 or TOPS-20 operating system and either magtape or disk may be used as the load medium.

MAGMON is command-controlled and can be directed to load and run a single program or execute a control file which will direct MAGMON to run a sequence of programs. Control files enable MAGMON to be used for:

- Rapid checkout of the hardware
- Acceptance testing
- Reliability testing
- Unattended overnight testing.

MAGMON also supports KI10 margining and special user mode operations.

Hardware Required

KA10, KI10 or KL10 mainframe/32K of core (minimum)/load device:

1. Exec mode devices are as follows.
   a. DX10/TX01 with TU7x tape drives. Only 9-track tape drives may be used. If the DX10 microcode is not loaded, the tape will be rewound and the microcode will be read from the READ IN bootstrap file on the magtape using the bootstrap ROM in the DX10.
   b. RH10 with TU16 tape drives.
   c. RH20 with TU16 or TU45 tape drives.
   d. TM10A or TM10B with any combination of TU10, TU20, TU30 or TU40 tape drives. Tape drives may be 7- or 9-track. If a TM10B, the DF10 can be in either KA or KI mode.
2. User mode devices are as follows.
   a. Any magtape device supported by the monitor.

NOTE
When running under a TOPS-20 monitor, the magtape device must be assigned and the density set by a monitor command.
e.g. @ASSIGN MTA0
@SET TAPE DENSITY 800 BPI

   b. The user's disk area.

Preliminary and Associated Programs

MAGMON assumes that the basic instructions and the selected load device are operational.

Restrictions

The diagnostic monitor may be used to call only those programs which follow the prescribed diagnostic formats.

Notes

If the monitor fails to operate, use the diagnostic programs individually to isolate the problem.

The DECsystem SUBRTN package and DDT are automatically loaded on system startup or device specification if they are not already resident in the PDP-10 memory.

Loading and Starting Procedure

Standard (Refer to the 10/10 STD module.)
To load MAGMON from the diagnostic magtape, use either the MAGTAP program or the following monitor commands.

TOPS-10

```
.ASSIGN MTA0:
.REWIND MTA0:
.SET DENSITY MTA0: 800(1600) BPI
.SET BLOCKSIZE MTA0: 512
.SKIP MTA0: 2 FILES
.COPY MAGMON.SAV=MTA0:
```

TOPS-20

```
@ASSIGN MTA0:
@REWIND MTA0:
@SET TAPE DENSITY 800(1600) BPI
@SKIP MTA0: 2 FILES
@COPY (FROM) MTA0: (TO) MAGMON.SAV
```

Control Switches

The state of the control switches does not affect the operation of MAGMON unless a control file is being used. A control file lists, as part of each command line, the program to be run and the right half switches to use with that program. This allows the actual (console) right hand switches to be used to control the operation of MAGMON. The switches which affect the operation of MAGMON when a control file is in use are listed in Table 1.

OPERATIONAL CONTROL
After MAGMON is started it will print the following message:

MAGMON-DECSYSTEM DIAGNOSTIC MAGTAPE MONITOR
DEV:

Table 2 describes the exec mode device selection commands. In user mode respond by typing the physical name of the magtape unit to be used (e.g., MTA0:<CR>) or, to use the disk, type a carriage return <CR>.

After selection of the load device DIAMON will automatically load SUBRTN and DDT and print:

MAGMON CMD -

Table 3 describes general MAGMON commands.
Table 4 describes program starting commands.
Table 5 lists MAGMON manual starting addresses.

Table 1 MAGMON Control Switch Summary

Switch	State	Descriptions
9	0	Reduces the iteration count in a control file by a factor of 100 to 1, thus reducing the run time for each program in the file. This is useful for a quick check of the hardware and margining operations.
	1	Each program listed in a control file is run the specified number of iterations.
15	0	Normal operation
	1	Inhibit printing the test title of each program executed by MAGMON.
18	0	Normal operation
	1	Expand the basic command set to include margining and special user mode operations. Refer to the X command, Table 3.

Table 2 MAGMON Load Medium Selection Commands (Exec Mode)

Command	Description
D	DX10/TX01 tape controller (TU70) device code 220
D2	Second DX01 device code 224
D3	Third DX01 device code 034
Dxxx	A 3-digit octal number may be typed following the D command to specify a nonstandard device code (e.g., D134<CR>). Next the UNIT # will be requested. Respond with a single digit (0-7) which indicates which tape drive is to be used. **NOTE** **If a DX10 is selected, only 9-track drives may be used.**
M M1 M2 M3 M4 M5 M6 M7	RH20 Massbus controller (TU16, TU45) device code 540 Second RH20 device code 544 Third RH20 device code 550 Fourth RH20 device code 554 Fifth RH20 device code 560 Sixth RH20 device code 564 Seventh RH20 device code 570 Eighth RH20 device code 574 UNIT # will be requested. Respond with a 2-digit number, where the first digit specifies the number of the TM02 and the second digit specifies the slave number of the tansport.
R R2 R3 R4 R5 R6	RH10 controller (TU16) device code 270 Second RH10 device code 274 Third RH10 device code 360 Fourth RH10 device code 364 Fifth RH10 device code 370 Sixth RH10 device code 374
Rxxx	A 3-digit octal number may be typed following the R command to specify a nonstandard device code (e.g., R260<CR>). Next the UNIT # will be requested. Respond with a 2-digit number where the first digit specifies the number of the TM02 and the second digit specifies the slave number of the transport.
T	TM10 controller (TU10, TU20, TU30, TU40) device codes 340 and 344
T2	Second TM10 device codes 350 and 354
Txxx	A 3-digit octal number may be typed following the T command to specify a nonstandard device code (e.g., T360<CR>). Next the UNIT # will be requested. Respond with a single digit (0-7) which indicates which tape drive is to be used.

Table 3 MAGMON General Command Summary

Command	Description
<CR>	Standard command terminator.
$	Altmode - a special command terminator which causes a single program to be loaded but not started.
↑Z	A control Z is used to terminate the T command.
D	D<CR> Directs MAGMON to read a control file from the load medium. MAGMON will respond by printing FILE.EXT-. Respond by typing the name of the control file.
F	F<CR> Directs MAGMON to print a directory of the load medium.
G	G<CR> Directs MAGMON to start or restart execution of the program currently loaded in core.
I	I<CR> Directs MAGMON to begin execution of the control file currently in core.
L	L<CR> Directs MAGMON to print a file stored on the load medium. MAGMON will request the name of the file to be printed by printing FILE.EXT-.
R	R<CR> Directs MAGMON to reinitialize itself. MAGMON will begin by requesting the load medium to be used.
S	S<CR> Directs MAGMON to load a single program. MAGMON will request the name of the program by printing FILE.EXT-. This question may be answered with only the file name as the extension will default to .A10 or .SAV.
T	T<CR> Directs MAGMON to open a buffer and begin building an internal control file. A control Z (↑Z) terminates the T command. Refer to section on building control files which follows Table 5.
X	X<CR> Directs MAGMON to run through the expanded command set dialogue.

EXPANDED COMMAND DIALOGUE
The following additional command sequences are added when either the X command is used or the XPAND switch (18) is set.

```
TYPE Y FOR CLOCK MARGINS -

    <CR>  = no margins
    Y<CR> = clock margins

Clock margins speeds up the basic clock cycle by 10 percent.

TYPE Y OR A FOR SPECIAL USER MODE -

    <CR>  = no
    A<CR> = special user mode after first pass
    Y<CR> = special user mode on all passes

    Special user mode is a pseudo-user mode where the diagnostic
    program being run is run in user mode (with paging, etc.) and
    the I/O is trapped back to the diagnostic monitor for
    processing.  This provides a method of checking user mode
    operation with functional and reliability diagnostics without
    actually having to use a monitor and timesharing.
```

The following additional question is asked if special user mode is selected.

TYPE Y OR A FOR CONCEALED MODE -

<CR> = public mode
Y<CR> = concealed mode
A<CR> = alternate public and concealed modes

TYPE L, S, I, R FOR VOLTAGE MARGINS -

<CR> = No margins

L<CR> = Limits. Margins are done at the ±.25 V settings only.

S<CR> = Sweep margins. The 5 V power supplies are varied by 1-increment steps (21 MV) up to and down to the ±.25 V limits.

I<CR> = Increment. The user may specifiy the margin step per program pass. If I is typed then the following question will be asked.

SPECIFY MARGIN INCREMENT (1 TO 17) -

One increment step equals 21 millivolts (e.g., 4 would specify an 84-millivolt increment).

R<CR> = Rack. Sweep margins are run on the processor logic rack specified. The following question is asked.

SPECIFY RACK (0 to 37) -

Type rack number to be margined.

Margins provide the control necessary to operate the KI10 programmable margin system. Answer the above question(s) then:

SET 'MARGIN SELECT' OFF
SET 'MARGIN ENABLE' SWITCH
TYPE ANY CHAR WHEN READY!

Table 4 MAGMON Program Starting Commands

Command	Description
DDT	DDT<CR> Start DDT
PFSTRT	PFSTRT<CR> Power fail restart
REE	REE<CR> Reenter
SFSTRT	SFSTRT<CR> Special features start.
START	START<CR> Start diagnostic
START#	START3<CR> Special start. Numbers range from 1 through 5.
STD	STD<CR> Start diagnostic
STL	STL<CR> Start MAGMON
STM	STM<CR> Reinitialize start

Table 5 Standard Manual Starting and Restarting Addresses

Address	Description
20000	MAGMON starting address
20001	If it is desired to abort a test currently in progress or to restart at the next sequential program, the operator may do so by starting at location 20001.
20002	If the diagnostic monitor is running in the mode where titles are not printed [SW 15(1)] and a user program fails such that it is not known which program failed, starting at location 20002 will cause the title to be printed. The computer will then halt at location 20000. The operator may at this time manually restart the user program or restart the diagnostic monitor.
20003	Program starting and restarting address.

MAGMON CONTROL FILES

A control file for MAGMON is an ASCII file consisting of a list of programs to be run. The following apply to constructing a MAGMON control file.

1. A control file can be constructed with any editor program or via the MAGMON T command.

2. A control file can have up to 50 command lines.

3. Each command line consists of five items each separated by a space or tab. The items are as follows.

   a. Program name. If the program name includes an extension, the extension must be included and separated by a period.

   NOTE

   If the special user mode routines are selected, a line that starts with a minus (-) signifies that the program will run in special user mode.

   b. Pass count. The pass count is the number of passes that the program is to run. The pass count may be in the range 0 to 777777. If 0, the program will run on each pass.

   c. Switches. This is an octal half word (6 digits) to be used by the program as the right half of the console data switches.

   d. Iterations. This is the number, in octal, of iterations the program is to be executed. The iteration count may be in the range 0 to 377777. If 0, one iteration is assumed.

   e. <CR>. A carriage return terminates the command line and opens the next line for input. If the T command was used to build the control file a ↑Z (control Z) will close the file and return to MAGMON command mode.

   Example:

```
DEKAA.A10    10    0         1000<CR>
DEKAB.A10     1    123456    200<CR>
DEKAC.A10     0    00001     1<CR>
↑Z
```

4. If the control file is being generated via the T command the following headers will be printed. These act as a guide only and are not actually a part of the control file.

   NAME PASSES RH SWS ITERATIONS

5. Typing errors may be corrected by typing a RUBOUT. The RUBOUT will print three Xs and delete the entire line.

The control file is executed via the I command. The diagnostic monitor will read in and execute the first program on the command list. The program will be iterated the requested number of times and control will then revert to the monitor. The monitor will then proceed to the next program on the list until all programs requested have been executed. When the final program on the command list has been executed, the pass count will be printed and then the monitor will restart with the first program again.

Example:

```
MAGMON PASS 000001
MAGMON PASS 000002
etc.
```

A control file will remain in core so that if the monitor is restarted the command list does not have to be read in again unless a new command list or single program is selected.

To use the same control file, type I.

MAGMON ERROR SUMMARY

CMD'S REQUIRED
The program was commanded to execute the control file, but the list is empty. Input some programs to execute.

Load Device Errors
Any load device errors will print out the reason, ERROR AT, and the octal address of the error. Consult the listing for error explanation.

Margin Errors
If margins are selected and the MARGIN ENABLE switch is not set on startup the margin setup message will be repeated.

'MARGIN ENABLE' NOT SET?
If during a margin run the MARGIN SELECT switch is reset, the above message will be printed. All subsequent programs will be run in normal mode.

MUUO ERROR
If the diagnostic program being run (in special user mode) causes an MUUO, (not trapped I/O) the above error message will be printed and the program will halt. The operator may examine the user MUUO locations (17424 and 17425) to determine the cause of the error.

PROGRAM NOT FOUND - PROG.EXT
The program requested is not on the load device.

USER TRAP ERROR
If the diagnostic program being run (in special user mode) causes a trap (PAGE FAIL, PUSHDOWN OVERFLOW or TRAP 3) the above message will be printed and the program will halt.

GENERAL INFORMATION

Code	DDQDF.SAV
Title	MAGTAP Diagnostic Magtape Creator
Abstract	The diagnostic distribution magtape creator program is a user mode program that is used to create and read diagnostic distribution magtapes. This program should provide all that is necessary to easily create a magtape that can be read by the diagnostic magtape monitor (MAGMON). This program also provides a simple method of transferring files from a diagnostic magtape to disk.
Hardware Required	KA10, KI10 or KL10 mainframe/32K of core (minimum)/a magtape subsystem/a disk subsystem.
Preliminary and Associated Programs	MAGTAP assumes that the above hardware is fully operational.
Restrictions	If running under TOPS-20 the TOPS-10 UUO simulation facility must be available.
Notes	The standard wild characters construction may be used with the CHECK, GET and VERIFY commands.
Loading and Starting Procedure	Before running MAGTAP a magtape unit must be assigned. For example: @ASSIGN MTA0 @SET TAPE DENSITY 800(1600) BPI If the MAGTAP is already in your disk directory, it can be started simply by typing: RUN MAGTAP.SAV If the MAGTAP program is not on disk, it must be retrieved from the diagnostic magtape. Mount the magtape on a tape drive, then follow the appropriate example below (in the examples, the tape drive is assumed to be MTA0). If TOPS-10 monitor: ASSIGN MTA0: REWIND MTA0: SET DENSITY MTA0: 800(1600) BPI SET BLOCKSIZE MTA0: 512 WORDS SKIP MTA0: 1 FILE COPY MAGTAP.SAV=MTA0: If TOPS-20 monitor: @ASSIGN MTA0: @REWIND MTA0: @SET TAPE DENSITY 800(1600) BPI @SKIP MTA0: 1 FILE @COPY (FROM) MTA0: (TO) MAGTAP.SAV
Control Switch	None

OPERATIONAL CONTROL
Upon starting, the MAGTAP program will print its name and version number, then will ask for the name of the magtape device it should use:

MAGTAPE DEVICE -

Respond by typing a logical or physical name for a magnetic tape device. A check will be made that the device selected is available and is a magtape device. If the tape unit has the capability of reading and writing at 1600 BPI, the following is asked:

WHAT DENSITY? (800 or 1600) -

Either 800 or 1600 must be typed.

The MAGTAP program will now be ready to accept any commands and will prompt with:

COMMAND -

MAGTAP COMMAND SUMMARY

The MAGTAP program performs all operations in response to commands typed by the user. Commands consist of a verb followed by optional arguments. A separator character (space, tab or comma) must separate the command from any arguments that may follow. A comma must be used to separate arguments. Every command line must end with a carriage return.

The command verb can be abbreviated to any extent as long as the abbreviation is unique. In this version of MAGTAP, no two commands start with the same letter. Therefore, all commands may be abbreviated by typing only one letter. This may not be true in later versions, however.

Table 1 summarizes the MAGTAP commands.

Table 2 summarizes software switches which may be used with the CHECK, GET MAKE and VERIFY commands.

Table 1 MAGTAP Command Summary

Command	Description	Cross Ref.
CHECK	C,file.ext, file ext<cr> Read the specified files on magtape and check for errors. C/TYPE:USER<CR> or C/GROUP:COMM<CR> Read the specified types or groups of files on magtape and check for errors. C<CR> Read entire magtape and check for errors.	1
DIRECT	D<CR> or D dev:<CR> List the magtape directory on the console terminal or specified device.	2
EXIT	E<CR> The exit command causes the program to exit to monitor command level without unloading the tape.	N/A
FDIR	F<CR> or F dev:<CR> List an abbreviated magtape directory on the console terminal or specified device.	3
GET	G, file.ext, file.ext<CR> Copy the specified files from magtape to disk. G/TYPE:USER<CR> or G/GROUP:COMM<CR> Copy the specified types or groups of files from magtape or disk. G<CR> Copy all files from magtape to disk.	4

Table 1 MAGTAP Command Summary (Cont)

Command	Description	Cross Ref.
HELP	H<CR> The help command will cause a brief description of MAGTAP commands to be printed on the user's terminal. All arguments are ignored.	
MAKE	M DDRPI.A10,DDRPD.SAV<CR> or M/GROUP:ALL<CR> Make a diagnostic magtape containing the specified files or types or groups of files.	5
MAKE	M @dev:file.ext[P,PN]<CR> Make a diagnostic magtape as described in the control file.ext specified.	6
NUMBER	N file.ext v.v dd-mmm-yyy<CR> Change the version and creation date for the file specified to the version and date specified in the command.	7
OPTION	O name<CR> Execute the command line identified by "name" in the ASCII file SWITCH.INI.	8
REWIND	R<CR> The REWIND command will cause the magtape to be rewound and the in-core directory to be cleared, forcing the directory to be read from the magtape on the next GET, CHECK, VERIFY, DIRECTORY or FDIRECTORY command. This command is provided for the user's convenience. It should never be required for normal operations.	N/A
SEARCH	S dev:[P,PN], dev:[P,PN]...<CR> The SEARCH command is used to specify the disk area which is to be used for reading files by the MAKE, VERIFY and NUMBER commands.	9
TAPE	T MTA4:<CR> The TAPE command is used to select a new magtape device.	10
UNLOAD	U<CR> The UNLOAD command will cause the tape to be unloaded. The program then exits to monitor command level.	N/A
VERIFY	V file.ext,file.ext<CR> or V/GROUP:COMM<CR> The VERIFY command reads the files, types, or groups of files specified in the command string and compares them to the same files on disk.	11

Table 2 MAGTAP Software Switch Summary

Switch	Description	Cross Ref.
/LIST:	VERIFY/LIST:<CR> Used with the CHECK, GET, MAKE, or VERIFY command. Causes the file name to be printed when the file is processed.	N/A
/GROUP:arg	CHECK/GROUP:KLBDIAG<CR> Used with the CHECK, GET, MAKE, or VERIFY command. Specifies the groups of files to be processed. ALL KACPU KLBTIC SYSEXR UNDEF3 COMM KICPU KLCPU TAPE UNDEF4 DIASUP KLADIAG KLDIAG TOPS10 UNDEF5 DISK KLAISO KLISO TOPS20 UNDEF6 FEDIAG KLATIC KLTIC UNIREC FESUPP KLBDIAG MEMORY UNDEF1 FLDSRV KLBISO SPECIAL UNDEF2	12
/TYPE:arg	GET/TYPE:EXEC<CR> Used with the CHECK, GET, or VERIFY command. Specifies the types of files to be processed. Note: The numberic value associated with each argument is used by the MAKE command. ALL 0 USER 2 MONITOR 10 EXEC 1 SPECIAL 4 UNDEF 20	13

COMMAND AND FILE DESCRIPTIONS
This section describes in detail the commands and switches summarized in Table 1 and Table 2.

1 CHECK,DFSXA.A10<CR> - The CHECK command provides a simple method of determining whether the magtape can be read. The CHECK command may be followed by a list of file specifications. If just the CHECK command is typed, the entire tape is checked. If one or more files is specified, only those filed will be checked.

In addition, the optional /GROUP:arg and /TYPE:arg switches may be used to check only the selected file groupings and/or selected file types.

The check process involves reading the magtape directory and spacing to the file to be checked. The data blocks are then read to determine that there are no hard read errors. Files with extensions .A8, .A10, .A11, .TIC and .RAM are checked for proper data format and internal checksums are verified.

2 DIRECTORY LPT:<CR> - The DIRECTORY command should be used to retrieve the directory of the magtape. The directory file MAGTAP.DIR will be read from the magtape and printed on the user's terminal.

Optionally, a device name can be specified immediately after the directory command. If present, the directory will be copied to that device. Any device that can accept ASCII mode output can be specified.

3 FDIRECTORY LPT:<CR> - The FDIRECTORY command should be used instead of the directory command when a faster directory is desired. This command will print only the file names and extensions of the files on the magtape. Four names will be printed on each line.

An optional device name can be specified as in the FDIRECTORY command.

4 GET DGKAA.A10<CR> - The GET command is used to copy files from magtape to the disk. The GET command may be followed by a list of file specifications in the form file.ext. Both the filename and extension must be specified. If just the GET command is typed, MAGTAP will copy all of the files from the magtape to the disk. If one or more files are specified, only those files will be copied to disk.

In addition, the optional /GROUP:arg and /TYPE:arg switches may be used to copy only the selected file groupings and/or selected file types to the disk.

All files copied to disk are given their original file name and extension. The original creation date and version number also are maintained.

5 Make DDRPI.A10,DDRPD.SAV<CR> - The MAKE command is used to create a magtape. MAGTAP requires that the following files be accessible on disk.

MAGTAP.SAV
MAGMON.SAV
MAGTAP.RDI and/or DXMPA.A8

To specify the files that are to be written onto the magtape, a list of file specifications in the form file.ext separated by a comma must follow the MAKE command. The MAGTAP program will immediately create a file called MAGTAP.TMP on DSK containing the following.

MAGTAP.RDI
MAGTAP.SAV
MAGMON.SAV
MAGTAP.MTA
MAGTAP.DIR
MAGTAP.TMP
file.ext first file specified
file.ext second file specified, etc.

6 MAKE<CR> or MAKE @file <CR> or MAKE @dev:file.ext[P,PN]<CR> - Causes the MAKE command to use the contents of an existing control file to control the generation of the magtape.

MAKE<CR> defaults to the user's disk area and an .INP file extension

MAKE @file <CR> defaults to the system disk (DSK), a filename and extension of MAGTAP.INP and the user's disk area.

7 N DDTUA.A10 0.2 20-JAN-79<CR> - The NUMBER command is used to change the version number and the date of a file on disk. This version number and date will then appear in the magtape directory for all succeeding MAKE commands. The number command may be used on any file on disk that does not contain this information internal to the file. Files with a .SAV extension have a version number in location 137, so unless this location is 0, only the date will have any effect on the magtape directory file.

Files with extensions of .A8, .A10, .A11, .TIC and .RAM normally begin with a file ID line. This ID line contains the file name and may contain the version and date. The version and date in this ID line (if present) will be used rather than those in the disk directory by the make command.

The above example would set the version to 0.2 and the date to 20-JAN-79. If it is desired to change only one of the file's parameters, only that parameter need be specified. The months should be specified as JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC.

8 O RP04/5/6<CR> - Any number of command line entries may be placed in a file called SWITCH.INI. The OPTION command directs MAGTAP to read and execute the command line specified as the option "name." The format of a command line in SWITCH.INI is

MAGTAP:NAME dev:[P,PN],dev:[P,PN]....

If no name is specified in the command, a line with no :NAME will be read. An option command with no name specified is automatically called whenever the MAGTAP program is started or restarted.

9 **S FS:[6,10],FS:<CR>** - The SEARCH command is used to specify the disk areas from which the files are to be read. Up to 20 disk areas may be specified. This search list will be used by the MAKE, VERIFY, and NUMBER commands. The general form of the SEARCH command is:

SEARCH dev:[P,PN],dev:[P,PN]....<CR>

Individual disk areas must be separated by commas. If only the dev: is specified, [P,PN] defaults to the job's P,PN. If only the [P,PN] is specified, dev: defaults to DSK:. An implicit DSK: always follows the search list specified.

For example, to make or verify a magtape with files on FS:[6,10] and [1,2] from a job logged into [1,2], the SEARCH command shown in the example would be used. The command will cause the MAGTAP program to look for each file in FS:[6,10]. When a file is not found there, MAGTAP will look in FS:[1,2] next. Finally DSK: will be searched.

10 **T MTA4:<CR>** - The TAPE command is used to select a new magtape device. The TAPE command must be followed by a device name. If the device specified is not a magtape device,

MAGTAPE DEVICE -

will be asked. Now respond with the correct device name.

If the tape unit selected has the capability of reading and writing at 1600 bits/in, the following will be asked.

WHAT DENSITY? (800 or 1600) -

either 800 or 1600 (or an abbreviation) must be typed.

A TAPE command is automatically called whenever the MAGTAP program is started or restarted.

If an altmode character is typed as the device name, no magtape device will be selected. This feature was added for debug purposes, but may be useful at times. The HELP, NUMBER, and MAKE commands may be used with no magtape device selected. The MAKE command will create the directory files on DISK, then will exit before attempting to write the tape.

11 **V/TYPE:EXEC<CR>** - The VERIFY command can be used to verify beyond a doubt that the magtape can be read correctly. The VERIFY command may be followed by a list of file specifications. If just the VERIFY command is typed, the entire tape is verified. If one or more files are specified, only those files will be verified.

In addition, the optional /GROUP:arg and /TYPE:arg switches may be used to verify only the selected file groupings and/or selected file types.

The verify process performs the check process but also reads the file from disk and verifies that the data matches exactly. In order to perform a verify process, the files to be verified must exist on the disk.

12 **/GROUP:KLBDIAG<CR>** - The GROUP:argument switch may be appended to the CHECK, GET or VERIFY command and is used to selectively restore or check specific groups of files on magtape. The arguments for the /GROUP: switch are as follows.

ALL	Overrides groups on the magtape
COMM	Communications diagnostics
DIASUP	Diagnostic support
DISK	Disk diagnostic
FEDIAG	KL10 front-end diagnostic
FESUPP	Additional front-end diagnostics
FLDSRV	Field service support

KLCPU	KL10 central processor diagnostics
KICPU	KI10 central processor diagnostics
KLADIAG	KL10 model PA diagnostics
KLAISO	KL10 model PA isolation routines
KLATIC	KL10 model PA clock scanout files
KLBDIAG	KL10 model PV diagnostics
KLBISO	KL10 model PV isolation routines
KLBTIC	KL10 model PV clock scanout files
KLCPU	KL10 CPU functional diagnostics
KLDIAG	KL10 hardware diagnostics
KLISO	KL10 hardware isolation routines
KLTIC	KL10 hardware clock scanout files
MEMORY	Memory diagnostics
SPECIAL	Special programs
SYSEXR	System exerciser diagnostics
TAPE	Magtape diagnostics
TOPS10	TOPS-10 monitor support
TOPS20	TOPS-20 monitor support
UNIREC	Unit record diagnostics
UNDEF	UNDEF1 through UNDEF6 are undefined groups.

13 /TYPE:EXEC<CR> - The /TYPE:argument switch may be appended to the CHECK, GET or VERIFY command and is used to selectively restore or check specific types of files on magtape. The type switch may be used as many times as necessary to specify the selected types of files (e.g., GET /TYPE:EXEC /TYPE:USER<CR>). The arguments for the /TYPE: switch are as follows.

ALL	Overrides types on magtape
EXEC	Exec mode diagnostics
MONITOR	Monitor support programs
SPECIAL	Special programs
UNDEF	Undefined type
USER	User mode diagnostics

ERROR SUMMARY

All errors detected by this program are reported to the user by a message on the user's terminal. All errors that prevent continuation of the command will abort and ask for a new command.

All error messages begin with a question mark (?) so that this program can be run under a BATCH job, if desired. Error messages that are due to a device error include an octal copy of the monitor device status word.

DIAGNOSTIC MAGTAPES

Diagnostic magtapes are distributed by SDC, the Software Distribution Center and contain all the utility programs, diagnostic programs and control files used in the 10-Based 10 Maintenance Library. The magtapes are generated using the MAGTAP utility program and may be read using either the MAGTAP or MAGMON utility program.

The diagnostic magtapes are written in standard DIGITAL-compatible format at either 800 or 1600 bits/in. Each file is divided into 512 word records and ends with a single end-of-file mark. A double end-of-file mark is placed at the end of the last file on

tape. Any magtape subsystem which uses a DX10, RH10 or TM10 controller may be used to process a diagnostic magtape.

Each diagnostic magtape consists of the six files summarized in Table 3. These files are immediately followed by the remaining programs and files in the 10-Based 10 Maintenance Library.

Table 3 Diagnostic Magtape File Summary

File	Description	Cross Ref.
MAGTAP.RDI	The magtape readin bootstrap loader	1
MAGTAP.SAV	The diagnostic magtape generation utility program	2
MAGMON.SAV	The magtape diagnostic monitor	3
MAGTAP.MTA	A directory of the magtape readable by MAGTAP and MAGMON	4
MAGTAP.DIR	A printable directory of the magtape	N/A
file.ext	The name and extension of the control file used by MAGTAP to make the diagnostic magtape	5

Once an input file has been established or written, the MAGTAP program will see if the files MAGTAP.RDI and DXMPA.A8 are on the disk. If the MAGTAP.RDI does not exist or DXMPA.A8 has a more recent creation date, a new MAGTAP.RDI file will be created. This will be the file read by the hardware readin operation.

The MAGTAP program will then look up each file specified in the input file to obtain the version number and creation date. This information will be written into two files, MAGTAP.MTA and MAGTAP DIR. The file MAGTAP.MTA will contain three words for each file plus a header and a checksum word in each record. The MAGTAP.DIR file will be written in ASCII characters to become the printable directory of the magtape. A line will be inserted before the first file name containing the date and time of creation of the directory file.

All other lines of the file will be copied directly from the input file except for the first 34 characters of each line containing a printable character other than a semicolon in column 1. These 34 characters will be interpreted as the file name and extension and will be replaced with the file number on the magtape, the file name, extension, version and creation date. For example, an input file containing the following line:

```
DDXXX.A10                                   ;DEC-SYSTEM-10 DIAGNOSTIC
```

will produce a line similar to the following.

```
  8.  00  DDXXX A10  0.1  23-OCT-75  128 DEC-SYSTEM-10 DIAGNOSTIC
```

The magtape is then rewound and files are written onto the magtape exactly as read from the disk. After the entire tape has been written, the magtape will be rewound and the program will ask for another command.

DIAGNOSTIC MAGTAPE FILE DESCRIPTIONS
The following is a detailed description of the files summarized in Table 3.

1 **MAGTAP.RDI** - A readin bootstrap file which is placed on the front of every diagnostic magtape. The first record of this file is the executable PDP-10 code to perform the load and start the MAGMON program when activated by the hardware readin on a KA10 of KI10. The remaining records of the file are a core image of the DX10 microcode. During the generation of a diagnostic magtape, the MAGTAP program reads the DX10 microcode file DXMPA.A8 and produces the core image before generating the readin file.

2 **MAGTAP.SAV** - Refer to the first part of this summary.

3 **MAGMON.SAV** - Refer to the MAGMON summary module.

4 **MAGTAP.DIR** - The magtape directory file is generated by the MAGTAP program each time the MAKE operation is performed. This file contains all the necessary information about each file on the magtape. This file is written in 512-word records and can consist of as many records as necessary to contain information about each file.

The first word of each directory record is the file number of the first file named in that record. The first directory record will always have a first word of 0. Following the first word is a 3-word block for each file on the tape, in the order they appear on the tape. The right half of the last word in the record contains an 18-bit checksum of all the words in the record. This 18-bit checksum is computed by adding all the 36-bit words in the record together and then adding together the left and right halves of the result. The left half of the last word will contain a 1 if the information on the last file on the tape is contained in this directory record. Otherwise, the left half will always be 0.

In the 3-word block for each file, the file name is in the first word. The left half of the second word contains the extension, and the time the file was created is in the right half. The third word contains the version number in the left half and the creation date in the right half. The version number contains a 3-digit octal edit number followed by a 3-digit octal version number.

The group code information is contained in the second word in bit positions 20 through 24 and the type code information is contained in the second and third word with the low-order three bits in the third word in bit positions 18 through 20 and the high-order two bits in the second word in bit positions 18 and 19. Refer to the GROUP and TYPE switch described in Table 2.
The date and version of the file are obtained from the file itself, if so contained. If this information is not contained in the file, the extended disk directory lookup information is used. The following files may contain this information.

Files with a .SAV extension are read to determine the contents of location 137. If this location is nonzero, its contents are used as the version number.

Files with an extension of .A8, .A10, .A11, .TIC and .RAM are read to see if an ID line is present at the beginning of the file. If present, a version and date are searched for in this line and are then used.

5 **INPUT FILES** - An input file is an ASCII text file used by MAGTAP to control the generation of the diagnostic magtape. An input file is built using a standard text editor program. The following rules apply to constructing input files.

a. Comments may be inserted anywhere within the input file. However, they must be preceded by a semicolon.

b. The first five command lines in the file must be as follows.

```
MAGTAP.RDI
MAGTAP.SAV
MAGMON.SAV
MAGTAP.MTA
MAGTAP.DIR
```

c. The sixth command line must be the name and extension of the input file itself. Usually, the file.ext MAGTAP.INP is used.

d. The remaining command lines will specify what files, types of files or groups of files are to be put on the diagnostic magtape. They must begin at column 1 and use one of the following formats.

file.ext - Only one file name may be used per command line.

- A numeric code which specifies the file type. Refer to the /TYPE: switch (Table 2) for code descriptions.

/GROUP:arg - Refer to the /GROUP: switch (Table 2) for a description of the group arguments.

GENERAL INFORMATION

Code	SUBRTN, SAV, SUBKL.SAV, SUBUSR.SAV
Title	DECSYSTEM Diagnostic Subroutine Package
Abstract	A subroutine package consists of the following commonly used "service type" subroutines. Program Subroutine Initialization Control Switch Initialization Interrupt Handler UUO Handler Console Data Switch Input Teletype Input Print Sixbit Type-in/Print DF10 Control Word Print Memory Mapping KI10 Margin Printout Routine Device Code Change Subroutine Diagnostic Error Handler The subroutines in the package are stored in core locations 1000 through 10000 and are used by 10-based 10 utility and diagnostic programs as needed. The use of subroutines in this fashion eliminates redundency, facilitates program development, and inhances standardization. SUBRTN.SAV is used by exec and user mode diagnostic programs run on KA10- and KI10-based systems. SUBKL.SAV is used by exec mode diagnostic programs run on KL10-based system. SUBUSR.SAV. is used by user mode diagnostic run on KL10- and KS10-based systems. SUBUSR.SAV may also be used with KA10 and KI10 user mode diagnostics.
Notes	This summary is provided for information purposes only. The user of the 10/10 library has no control over the use of the subroutines package.

GENERAL INFORMATION

Code	DDCPA.A10
Title	PDP-10 CP10 Card Punch Diagnostic The PDP-10 card punch reliability test is a maintenance program designed to indicate malfunctions in the card punch and card punch interface.
Hardware Required	KA10, KI10 or KL10 mainframe/32K of core (minimum)/CP10 (card punch interface and a card punch) or CP10-D (card punch interface and a Documation P100 card punch). CR10 card reader interface and a card reader (optional).
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	The card device (punch or reader) must be READY before starting the program.

Notes

1. The program can be run both in the exec and user modes.

2. Unless switch 23 is set the program will punch up to 400 cards and exit or it will read up to 10,000 cards and exit.

3. The test punch patterns used in the reliability section of the diagnostic (phase 2) are compatible with the test patterns used in the card reader diagnostic (MD-10-DDCRA-A). The test patterns are:

   a. random data
   b. invalid data (validity pattern)
   c. floating 1s
   d. expanded floating 0s pattern.

4. Normal device code CR = 150
   Device code for CR1 is = 154

5. If this diagnostic is being used with card reader model CR10D (Documation No. 1000) or models CR10E-F (Documation No. 200), it is possible to have picked errors when reading cards that have the majority of the card punched. This is due to the vacuum drive on the card picker. To avoid this condition, shuffle the cards being read with less densely punched cards to eliminate consecutive identical punched cards.

6. If in exec mode, the card that was read incorrectly will be offset in the stacker so that it can be located and examined at any later time. CR10 models D through F have no offset features. Set SW07 to HALT on a card error if using one of these card readers.

7. Loading the Card Hopper

   a. Riffle the card deck and flex deck on both ends.

   b. Square the deck on a smooth surface.

   c. Take a 1 inch stack of cards and place in the card hopper so that the bottom of the stack (bottom cards) are resting against the card picker throat block. Allow the remaining cards of the deck to fall into place in this position.

   d. Add the remaining cards to this 1 inch stack.

   e. Turn the power ON.

f. Set the TEST/NORMAL or REMOTE/LOCAL switch to NORMAL or REMOTE (device dependent on make and model).

g. CP10-D - If either the ON-LINE or TEST switch indicators are on or blinking, shut off by pressing the relevant switch.

h. Eliminate any trouble conditions to the device.

i. Press CLEAR then START or RESET (device-dependent).

j. (CP10-D) - press the ON-LINE switch/indicator. The device drive motor should now be running.

Loading and Starting Procedure: Standard (Refer to the 10/10 STD module.)

Control Switches: Refer to Table 1.

OPERATIONAL CONTROL
Other than the switches, the user has no control over the operation of this diagnostic.

DDCPA TEST SUMMARY
DDCPA has two parts; Phase 1 and Phase 2. Phase 1 (Static test) runs in exec mode only and tests part of the control and status logic.

Phase 2 - with switch 18 = 0 (punching cards). Data patterns punched are controlled by switch 19 and switches 24-35.

19 = 0 Punch 100 cards (user mode) of each data pattern not inhibited by switches 32-35.

19 = 1 Punch all 80 columns of 100 cards and use the data in switches 24-35.

NOTE
Set switch 23 to continuously punch cards.

Phase 2 - with switch 18 = 1 (reading cards). Data cards are read in binary image and examined for correct data. The data is 11-bit pseudorandom numbers with an odd parity bit in row 12.

The order in which the cards are read is unimportant because the program calculates the cards' contents from the first column read. The assumption is that the first column is correct.

ERROR MESSAGE SUMMARY
All errors are self-explanatory. Upon each occurrence of an error the card is offset and another card is picked. If the error is persistent, so will be the printing. Switch 3 = (1) can be used to stop the printing of errors.

ERRORS WHEN PUNCHING CARDS
The following messages may appear in exec mode when punching cards.

CDP IS HUNG WAITING FOR BLKO TO COUNT OUT...STATUS WAS XXX
CDP EJECT FAILURE...CAN'T GET RID OF CARD IN PUNCH
CARD PUNCH TROUBLE...CHECK THE CDP
CDP RESET FAILED TO CLEAR BITS...STATUS WAS XXX
CARD PUNCH ERROR WAS DETECTED
CDP TEST MODE IS ACTIVE...CDP IS OFF LINE
CARD PUNCH OPR SERVICE REQUESTED (HOPPER-STACKER-CHIP BOX)

ERRORS WHEN READING CARDS
Data errors give the following message.

ERROR NO. XX
PARITY ERROR
XXXXXXX XXXXXXX DATA CARD
DATA IN XX COLUMNS HAVE ERRORS
DATA IN XX COLUMNS HAVE SUPER IMAGE ERRORS

	SUPER IMAGE READ INTO BITS 10 TO 17		READ INTO BITS 24 TO 35	
COLUMN	ERROR	CORRECT	ERROR	CORRECT
NUMBER	V9+ 08421	V9+ 08421	+ 0123456789	+ 0123456789
XX	XXXXXXXXX	XXXXXXXXX	XXXXXXXXXXX	XXXXXXXXXXX
XX	XXXXXXXXX	XXXXXXXXX	XXXXXXXXXXX	XXXXXXXXXXX
XX	XXXXXXXXX	XXXXXXXXX	XXXXXXXXXXX	XXXXXXXXXXX

etc.

DATA CARD - Incorrect data is printed with the number of errors in the left and right side of card data. The data itself is printed in a format described.

SUPER IMAGE refers to a data-condenser built into the BA10. When reading a card, the hardware will compress the data from 12 columns read to 8 columns with a VALIDITY BIT.

The validity column V will equal a 1 if more than one row has been punched in rows 1-7.

NOTE

"80 errors in the left" might appear if the diagnostic has been initialized in SUPER IMAGE mode and no hardware exists to input this data.

The + column is used to determine parity errors by containing the odd parity bit for the column punched.

A random data card error in row 12 (+) will yield correct data as 000000000000. Random cards with data errors in the beginning columns will have the same results. Also, any test pattern card with an error in column number 1 will not be able to determine the correct data and will yield all 0s.

A random data card with 0s in the "correct" field indicates either a parity error (in which case PARITY ERROR will be printed) or the actual CORRECT data cannot be determined.

The number of lines of data errors is limited to twelve unless the line printer is being used.

The following messages may occur in case of unusual status conditions during phase 2:

CARD MOTION ERROR-CARD IS JAMMED OR WRONG LENGTH
PICK FAILURE-CARD FAILED TO ENTER READ STATION
LIGHT-DARK CURRENT PHOTO CELL ERROR
READER TROUBLE OF UNKNOWN CAUSE, STATUS WAS XXXXXX
[THIS ERROR IS COMMON ONLY TO USER MODE AND WILL CAUSE NO DATA PRINTOUT UPON DETECTION.]

INTERRUPT FROM UNKNOWN CAUSE ON CHANNEL 7, CR STATUS WAS XXXXXXXXXXX

DATA WAS MISSED (SHOULDN'T HAPPEN)
READY-TO-READ CAME WHILE CARD WAS BEING READ
DATA-READY PI NEVER CAME
DATA-READY HAPPENED XXX TIMES, NOT 80 TIMES
END-OF-CARD-PI NEVER CAME
END OF FILE HAS OCCURRED

Table 1 DDCPA Control Switch Summary

Switch	Mnemonic	State	Description
0-17			Standard (refer to Section 4.2)
18	PUNCHRD	0	Punch cards
		1	Read and verify cards
19	GENDTA	0	Punch preselected data patterns (FLOAT ZEROS, FLOAT ONES, VALIDITY, RANDOM)
		1	Punch or read data pattern contained in switches 24-35.
20	CRDCNT	0	Normal operation
		1	Print number of cards punched per minute.
21	INHSTA	0	Normal operation
		1	Inhibit phase 1 (Static test)
22	INHREL	0	Normal operation
		1	Inhibit phase 2 (Reliability test)
23	CONT	0	Normal operation
		1	Continuous card punching in phase 2 (reliability)
24-35	SELDAT		If switch 19=0, switches 32 through 35 determine the data pattern punched. 32-35 = 0s punch all data patterns 32-35 = 1s punch blank cards 32 = 1 inhibit floating 0s 33 = 1 inhibit validity pattern 35 = 1 inhibit random data pattern
24-35	GENPAT		If switch 18=0 and switch 19=1 the pattern contained in switches 24-35 is punched. If switch 18=1 and switch 19=1 the pattern contained in switches 24-35 is compared against the card read.

GENERAL INFORMATION

Code	DDCRA.A10
Title	PDP-10 Card Reader Diagnostic
Abstract	PDP-10 card reader diagnostic (DDCRA) is designed to verify the operational status of the card reader. The program may be run in exec or user mode. It will diagnose most component or wiring failures in the control logic when in exec mode. It will read and verify the data in a deck of cards containing pseudorandom numbers, floating 1s pattern, floating 0s pattern, and a validity pattern.
Hardware Required	KA10, KI10 or KL10 mainframe/32K of core (minimum)/CR10A, CR10B, CR10D, CR10E or CR10F
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	Use an EOF card if running this program in user mode. This program is written for use only with the SOROBAN engineering compact card reader model ERD (CR10A and CR10B) and Documation models #1000, #1200, #200 (CR10D through CR10F).

Notes

1. When testing card reader models CR10D through CR10F, it is possible to get pick errors when reading many cards with the majority of the card punched (i.e., floating 0 pattern test cards). Avoid this condition by shuffling the card test deck.

2. A prepunched deck of cards containing the necessary data test patterns is required for phase 3.

3. An alternative set of cards may be prepared for phase 2. The data on the cards must meet these requirements.

   a. Every column on every card must have at least one but less than 12 holes punched.

   b. Every row on every card must have at least one but less than 80 holes punched.

      For example, repeat this series of characters across the card:

      +_0123456789+_123...

      (This is a floating 1s pattern)

      The order in which the cards are read is unimportant. A deck of identical cards copied from a card which makes the equipment fail may be made. This can be used for scoping the error.

4. Loading Cards

   a. Riffle (fan) both ends and flex the deck (CR10A and B).

   b. Jog the cards on a smooth surface to square up the deck.

   c. Place a 1 inch portion of the deck in the hopper, holding the right side of the deck higher than the left so that the leading edge of the bottom cards rests against the picker throat block. Allow the cards to fall in place in this position (CR10A and B).

   d. Add the rest of the deck to be read.

e. Turn on power.

f. Put TEST/NORMAL switch in the NORMAL position.

g. Press CLEAR or RESET.

h. Eliminate any trouble conditions as indicated by the indicators.

i. Press START or RESET. The motor should not be running.

Loading and Starting Procedures — Standard (Refer to the 10/10 STD module.)

Control Switches — Refer to Table 1.

OPERATIONAL CONTROL

Set the control switches as required.

If the BA10 has the ECO pertaining to the validity checking answer YES to this question:

TEST 'SUPER IMAGE' LOGIC IN THE BA10? Y OR N ?

SUPER IMAGE refers to an ECO made to the BA10 so that 12 columns of data can be compressed into 8 columns. The validity bit V is set when more than one row in rows 1 through 8 are punched in a column. The data tested from this compressor originates from the BA10 not the card reader.

DDCRA TEST SUMMARY
This diagnostic has three phases.

Phase 1 (switch 35 off, in exec mode) - Part of the control and status logic will be tested, and errors will be printed. No cards are actually read. This phase is never executed in a time-sharing environment.

Phase 2 (switch 34 off, in exec mode) - The remainder of the control logic status, and data buffers are tested, errors are printed, and cards are read.

Phase 3 (switch 33 off, in either user or exec mode) - The prepunched data cards are read and examined for exactly correct data, in image binary. The data is 11-bit pseudorandom numbers with an odd parity bit in row 12.

If the BA10 has the ECO pertaining to validity checking, run the following three special punched card decks:

1. Validity check - checks all combinations of double punching in each row with the validity bit on

2. Floating 1 - cards contain a floating 1s pattern repeating through all columns and rows.

3. Floating 0 - same as above except with a floating 0 pattern.

ERROR SUMMARY

Phases 1 and 2 (exec mode only)

If no match occurs between the operator's CR10 TYPE selection of switches 30-31 and the actual hardware status return on a CONI CR, an error message will be printed giving the actual status returned. The card reader status will be the contents of "AC" C(AC).

A fault in the logic of the card reader will cause a printout of this general form.

```
PC=XXXXXX
C(AC)=XXXXXXXXXXXX
E=XXXXXX
```

PC is the program counter at the point where the fault was detected. At that point in the listing the fault condition is given in the comments, as well as the events that caused failure. C(AC) is the contents of AC0, and the status of the card reader control at the time of failure. E is a sequential error number.

Phase 3

If in exec mode, the card that was read incorrectly will be offset in the stacker, so that it can be located and examined at any later time. CR10 models D through F have no offset features - set switch 7 to halt on a card error if using one of these card readers. Incorrect data is printed with the number of errors in the left and right side of card data. The data itself is printed in a format described below.

```
ERROR NO. XX
PARITY ERROR
XXXXXXX XXXXXXX DATA CARD
DATA IN XX COLUMNS HAVE ERRORS
DATA IN XX COLUMNS HAVE SUPER IMAGE ERRORS

                SUPER IMAGE
          READ INTO BITS 10 TO 17     READ INTO BITS 24 TO 35
COLUMN     ERROR         CORRECT      ERROR            CORRECT

NUMBER     V9+ 08421     V9+ 08421    + 0123456789     + 0123456789
    XX     XXXXXXXXX     XXXXXXXXX    XXXXXXXXXXX      XXXXXXXXXXX
    XX     XXXXXXXXX     XXXXXXXXX    XXXXXXXXXXXX     XXXXXXXXXXXX
    XX     XXXXXXXXX     XXXXXXXXX    XXXXXXXXXXXX     XXXXXXXXXXXX
ETC...
```

DATA CARD - Incorrect data is printed with the number of errors in the left and right side of card data. The data itself is printed in a format described.

SUPER IMAGE refers to a data-condenser built into the BA10. When reading a card, the hardware will compress the data from 12 columns read to 8 columns with a VALIDITY BIT.

The validity column V will equal a 1 if more than one row has been punched in rows 1-7.

NOTE

"80 errors in the left" might appear if the diagnostic has been initialized in SUPER IMAGE mode and no hardware exists to input this data.

The + column is used to determine parity errors by containing the odd parity bit for the column punched.

A random data card error in row 12 (+) will yield "correct" data as 000000000000. Random cards with data errors in the beginning columns will have the same results. Also, any test pattern card with an error in column number 1 will not be able to determine the "correct" data and will yield all 0s.

A random data card with 0s in the "correct" field indicates either a parity error (in which case PARITY ERROR will be printed) or the actual CORRECT data cannot be determined.

The number of lines of data errors is limited to twelve unless the line printer is being used.

The following messages may occur in case of unusual status conditions during phase 3:

```
CARD MOTION ERROR-CARD IS JAMMED OR WRONG LENGTH
PICK FAILURE-CARD FAILED TO ENTER READ STATION
LIGHT-DARK CURRENT PHOTO CELL ERROR
READER TROUBLE OF UNKNOWN CAUSE, STATUS WAS XXXXXX
[THIS ERROR IS COMMON ONLY TO USER MODE AND WILL CAUSE NO DATA
PRINTOUT UPON DETECTION.]

INTERRUPT FROM UNKNOWN CAUSE ON CHANNEL 7, CR STATUS
WAS XXXXXXXXXXX

DATA WAS MISSED (SHOULDN'T HAPPEN)
READY-TO-READ CAME WHILE CARD WAS BEING READ
DATA-READY PI NEVER CAME
DATA-READY HAPPENED XXX TIMES, NOT 80 TIMES
END-OF-CARD-PI NEVER CAME
END OF FILE HAS OCCURRED
```

Table 1 DDCRA Control Switch Summary

Switch	Mnemonic	State	Description
0-17			Standard (Refer to the 10/10 STD module.)
18-28			Not used
29	RANSPD	0	Normal operation
		1	Use a random card reader read speed.
30-31	CRTYP		Operator selection of CR10 type under test
			30 31 CR10 TYPE 0 0 CR10A or CR10B 0 1 CR10D (model #1000) 1 0 CR10E (model #1200) 1 1 CR10F (model #200)
32	RD100	0	Normal operation
		1	Read only 100 cards in phase 3
33	INHP3	0	Normal operation
		1	Inhibit phase 3 card reader reliability test
34	INHP2	0	Normal operation
			Inhibit phase 2 basic card reader read test
35	INHP1	0	Normal operation
		1	Inhibit phase 1 card reader static test

GENERAL INFORMATION

Code	DDDFA.A10
Title	PDP-10 DF10(C) with RP10(C), TM10B, RC10, RH10 Test
Abstract	This maintenance program is designed to detect and diagnose malfunctions in the DF10(C) data channel. It may test any DF10(C) which may be connected to an RP10(C), TM10B, RC10 or TM10B.
Hardware Required	KA10, KI10 or KL10 CPU/32K of core (minimum)/DF10 OR DF10C data channel RP10(C) disk pack controller with at least 1 drive or TM10B tape controller with at least 1 tape unit (TU10, TU20, TU30 or TU40) or RC10 fixed head disk or drum controller with at least 1 disk or drum or RH10 controller with at least 1 RS04 disk or RP04/06 disk pack drive.
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	1. All units to be tested must be on line and ready. 2. Disk packs must be in PDP-10 format. 3. If testing all subsystems [switch 18(0)], the unit numbers in each subsystem must be identical; i.e., DPA1, MTA1, etc.
Notes	4. If testing on a KI10 in 22-bit address mode (DF10C), the control word address test may not be completely done. In order to fully test the control word address logic it is suggested that at least one pass of the test be done in 18-bit address mode. This allows testing of control words up to 256K. 5. If testing with all possible memory (4096K on a KI10 or 256K on a KA10) it is advised to deselect at least some portion of memory in order to test the nonexistent memory logic.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	Refer to Table 1.

OPERATIONAL CONTROL

Make sure the console switches are set for the controller and DF10C that is to be tested.

Make sure that the drive to be tested is equal to the switches that select it.

If testing all devices, make sure that all devices are using the same unit number.

Starting Addresses

30000 Normal starting address

30004 Starting address for DF10C switch test

30007 Starting address of DDT

DDDFA TEST SUMMARY

Table 2 summarizes the DF10C switch test. This test can only be run on DF10Cs.

Table 3 summarizes the TM10B tests.

Table 4 summarizes the RP10(C) tests.

Table 5 summarizes the RC10 tests.

Table 6 summarizes the RH10 tests.

ERROR MESSAGE SUMMARY
Standard (Refer to the 10/10 STD module.)

Table 1 DDDFA Control Switch Summary

Switch	State	Description
0-15		Standard (Refer to the 10/10 STD module.) Exceptions - The following switches are not implemented: 1, 2, 10, 11, 13, 14, and 15.
16	0	60 Hz power (KA10/KI10 only)
	1	50 Hz power (KA10/KI10 only)
17		Not used
18	0	Test all devices regardless of 21-23 setting
	1	Test device specified by 21-23
19	0	Normal operation
	1	Disable control word chain typeout
20	0	Normal operation
	1	Use surface, cylinder or track 1 if disk or disk pack
21-23		Device test select 000 TM10B 001 RP10(C) 010 RC10 011 RH10 100 Future Device 101 Future Device 110 Future Device 111 Future Device
24-25		Not used
26	0	Normal operation
	1	Rewind magtape after each test. This switch only has effect if magtape is being tested.
27-29		Unit select test 000 Test Unit 0 001 Test Unit 1 010 Test Unit 2 011 Test Unit 3 100 Test Unit 4 101 Test Unit 5 110 Test Unit 6 111 Test Unit 7
30-35		Not used

Table 2 DF10C (only) Switch Test

Test	Description
DF10SW	This test checks the operation of the parity error (P.E.) stop switch. To run this test start DDDFA at address 30004. 1. Manually set the P.E. stop switch on the DF10C to ON. Set the write-even-memory lockout switch to OFF. Clear the DF10C (manually) and type on the terminal. 2. Set up and transfer 1 sector (64 words) of data to the disk or tape. Read this data back to memory with even parity. The data in memory will have even parity. Write this even parity data back on the disk or tape. The DF10C should halt and a data word parity error indication will appear. The DF10C MB will contain the incorrect data word and the DWPE indicator should be on. 3. After examining the the DF10C indicators, set the P.E. stop switch to OFF and manually clear the DF10C. The program waits for the operator to type on the terminal. 4. The program will proceed to the control word parity error test. The DF10C command list is modified to cause the first word to jump to the even parity buffer in memory (should cause a control word parity error). The DF10C should be halted with the MB equal to 0 and the control word register containing the incorrect control word (the even parity data in buffer). After visually inspecting the DF10C indicators, set the P.E. stop switch to OFF, press manual clear on the DF10C, and type on the terminal. 5. The program will proceed to the write even lockout test. Set the write-even-memory lockout switch to the ON position. The one sector of test data on the disk or tape will be written into memory with write-even-memory selected - the lockout switch should prevent even parity data from being written. Check memory for parity errors caused by even parity data.

Table 3 DDDFA Test Summary (TM10B)

Test	Description
TM1	TM10 Basic Static Test This test tries to set all TM10 PIA bits, then gets the TM10 status and determines if any PIA bits or any other status from the TM10 can be read. It also tells what type of DF10 the TM10 is connected to [only once - unless the program is restarted at 4000(8)].
.DF10T	TM10-DF10 Basic Jump Test Loads a DF10 command list with a jump to a terminate - then tests. This is done with all initial addresses from 20-776 (except 40-57).
TMCON	TM10 Memory Connection Test for DF10 Checks that the DF10 can transfer 8K words from each memory bank that the CPU is connected to. A bank is considered to be 8K words. This takes into account that the MA10 has upper and lower 8K deselection switches and that they may be set to deselect.
TMNXM	TM10-DF10 Nonexistent Memory Test

Table 3 DDDFA Test Summary (TM10B) (Cont)

Test	Description
.SKPTST	TM10-DF10 Skip Read Test Sets up control word (-17,,0) and starts DF10. Shadow (core 0-17) ACs are set to 0 and DF10 started. DF10 should skip 17 words and terminate without writing any information received from the device into memory. If the DF10 does write, it will probably start with location 0 so check that the shadow (core 0-17) ACs are still 0. The test also checks shadow (core 0-17) memory with all 1s because the previous test could have passed if the channel read 0s from the device and wrote 0s into shadow (core 0-17) memory (check for being all 1s.)
TMFET	TM10-DF10 Control Word Fetch Test Fetches control words from all of memory.
TMMUL	TM10-DF10 Multiple Control Word Fetch This test fetches multiple control words from memory. Control word is a skip of 17 words done 12 times. The test checks that the shadow (core 0-17) ACs are not written because a skip should never write anything - if it does, it will probably go into core 0-17. The test also checks that the written control word is correct; i.e., the DF10 fetched all control words and did not terminate early. Causes for early termination are: the DF10 did not fetch a control word, the DF10 had a control word parity error, or possibly the DF10 jumped to location 0 which has a 0 in it and is a DF10 terminate, or the DF10 had a nonexistent memory which could occur when the DF10 had a jump to 0 and a -1 was in location 0.
.JSKP	TM10-DF10 Jump-Skip Test Fills memory with a JUMP.+1,SKIP,JUMP.+1,SKIP.....etc. up to a 64(10) record skip, then starts the DF10. Each successive skip results in skipping one more record than the previous skip until the command list reflects a 64(10) record skip. The DF10 written control word is then checked. Incorrect termination can result from nonexistent memory, control word parity errors, and tape errors.
.SKRD	TM10-DF10 Skip-Read Test A control word chain is set up so that the DF10 does a skip followed by a read and checks that a skip followed by a read does not continue to skip.
.DXFER	TM10-DF10 Data Transfer Test
TMPAR	TM10-DF10 Parity Error Detection Test To test the parity error detection logic of the DF10, this program reads from the tape drive and writes even parity into memory. The test checks that the DF10 did in fact write even parity into memory, then uses the data written as a control word and expects a control word parity error. It then uses a correct control word pointing to buffer and expects a data word parity error.
TMWCT	TM10-DF10 Word Count Register Test Using magtape, this test proceeds in the following manner. First it fills the buffer with a count pattern for as many words as the channel is set up to transfer. The rest of the buffer is filled with 0s. Then it writes a file consisting of a 1-word record, backspaces to the beginning of the file, fills the buffer with the null pattern, and reads the file. Next it checks the buffer to make sure that only one word was read. Finally, the word count is shifted one place left and the whole process is repeated until it runs out of word count register or sufficient memory buffer for the entire word count.

Table 4 DDDFA Test Summary (RP10)

Test	Description
RP1	RP10 Basic Static Test This test attempts to set all RP10 PIA bits, then gets the RP10 status and determines if any PIA bits or any other status from the RP10 can be read. It also tells what type of DF10 the RP10 is connected to [only once - unless the program is restarted at 4000(8)].
DF10T	RP10-DF10 Basic Jump Test This test loads a DF10 command list with a jump to a terminate - then tests. This is done with all initial addresses from 20-776 (except 40-57).
RPCON	RP10 Memory Connection Test for DF10 This test checks that the DF10 can transfer 8K words from each memory bank that the CPU is connected to. A bank is considered to be 8K words. This takes into account that the MA10 has upper and lower 8K deselection switches that may be set to deselect.
RPNXM	RP10-DF10 Nonexistent Memory Test
SKPTST	RP10-DF10 Skip Read Test Control word (-17,,0) is set up and the DF10 is started. Shadow (core 0-17) ACs are set to 0 and DF10 started. DF10 should skip 17 words and terminate without writing any information received from the device into memory. If the DF10 does write it will probably start with location 0, so check that the shadow (core 0-17) ACs are still zero. The test also checks shadow (core 0-17) memory with all 1s because the previous test could have passed if the channel read 0s from the device and wrote 0s into shadow (core 0-17) memory (check for being all ones).
RPFET	RP10 Control Word Fetch Test Fetches control words from all of memory.
RPMUL	RP10-DF10 Multiple Control Word Fetch This test fetches multiple control words from memory (control word is a skip of 17 words done 12 times). It checks that the shadow (core 0-17) ACs are not written because a skip should never write anything - if it does it will probably go into core 0-17. The test also checks that the written control word is correct; i.e., the DF10 fetched all control words and did not terminate early. Causes for early termination are: the DF10 did not fetch a control word, the DF10 had a control word parity error, the DF10 jumped to location 0 which has a 0 in it and is a DF10 terminate, or the DF10 had a nonexistent memory which could occur when the DF10 had a jump to 0 and a -1 was in location 0.
JSKP	RP10-DF10 Jump-Skip Test JUMP.+1,SKIP,JUMP.+1,SKIP.....etc. is loaded into memory up to a 64(10) sector skip, then the DF10 is started. Each successive skip results in skipping one more sector than the previous skip until the command list reflects a 64(10) sector skip. The DF10 written control word is checked. Incorrect termination can result from nonexistent memory, control word parity errors, and disk errors.
SKRD	RP10-DF10 Skip-Read Test A control word chain is set up so that the DF10 does a skip followed by a read and checks that a skip followed by a read does not continue to skip.

Table 4 DDDFA Test Summary (RP10) (Cont)

Test	Description
%DXFER	RP10-DF10 Data Transfer Test Data of various types are transferred to each memory bank connected to the system. Each pattern is done ten times in each bank connected.
RPPAR	RP10-DF10 Parity Error Detection Test To test the parity error detection logic of the DF10, this program reads from the RP03 and writes even parity into memory. It checks that the DF10 did in fact write even parity into memory, then uses the data written as a control word and expects a control word parity error. Then it uses a correct control word pointing to buffer and expects a data word parity error.
RPWCT	RP10-DF10 Word Count Register Test Using disk packs, this test proceeds in the following manner. First, it fills the buffer with a count pattern for as many words as the channel is set up to transfer. The rest of the buffer is filled with 0s. Then the test writes a file consisting of one word, fills the buffer with the null pattern, and reads the file. Next, it checks the buffer to ensure that only one word was read. Finally, the word count is shifted one place left and the whole process is repeated until it runs out of word count register or sufficient memory buffer for the entire word count.

Table 5 DDDFA Test Summary (RC10)

Test	Description
RC1	RC10 Basic Static Test This test tries to set all RC10 PIA bits, then gets the RC10 status and determines if any PIA bits or any other status from the RC10 can be read. It also tells what type of DF10 the RC10 is connected to.
DFXT	RC10-DF10 Basic Jump Test A DF10 command list is loaded with a jump to a terminate - then tested. This is done with all initial addresses from 20-776 (except 40-57).
RCCON	RC10 Memory Connection Test for DF10 This test checks that the DF10 can transfer 8K words from each memory bank that the CPU is connected to. A bank is considered to be 8K words. This takes into account that the MA10 has upper and lower 8K deselection switches and they may be set to deselect.
RCNXM	RC10-DF10 Nonexistent Memory Test
SKXTST	RC10-DF10 Skip Read Test This test sets up control word (-17,,0) and starts DF10. Shadow (core 0-17) ACs are set to 0 and DF10 started. DF10 should skip 17 words and terminate without writing any information received from the device into memory. If the DF10 does write it will probably start with location 0, so check that the shadow (core 0-17) ACs are still 0. The test also checks shadow (core 0-17) memory with all 1s because the previous test could have passed if the channel read 0s from the device and wrote it into shadow (core 0-17) memory (check for being all 1s).
RCFET	RC10 Control Word Fetch Test Fetches control words from all of memory.

Table 5 DDDFA Test Summary (RC10) (Cont)

Test	Description
RCMUL	RC10-DF10 Multiple Control Word Fetch This test fetches multiple control words from memory. Control word is a skip of 17 words done 12 times. The test checks that the shadow (core 0-17) ACs are not written because a skip should never write anything - if it does it will probably go into core 0-17. The test also checks that the written control word is correct; i.e., the DF10 fetched all control words and did not terminate early. Causes for early termination are: the DF10 did not fetch a control word, the DF10 had a control word parity error, the DF10 jumped to location 0 which has a 0 in it and is a DF10 terminate, or the DF10 had a nonexistent memory which could occur when the DF10 had a jump to 0 and a -1 was in location 0.
JSKX	RC10-DF10 Jump-Skip Test JUMP.+1,SKIP,JUMP,+1,SKIP,....etc. is loaded into memory up to a 64(10) sector skip, then the DF10 is started. Each successive skip results in skipping one more sector than the previous skip up until the command list reflects a 64(10) sector skip. The DF10 written control word is then checked for being correct. Incorrect termination can result from nonexistent memory, control word parity errors, and disk errors.
SKRDX	RC10-DF10 Skip-Read Test A control word chain is set up so that the DF10 does a skip followed by a read and checks that a skip followed by a read does not continue to skip.
$DXFER	RC10-DF10 Data Transfer Test
RCPAR	RC10-DF10 Parity Error Detection Test To test the parity error detection logic of the DF10, this program reads from the disk (drum) and writes even parity into memory. The test then checks that the DF10 did in fact write even parity into memory, then uses the data written as a control word and expects a control word parity error. It then uses a good control word pointing to buffer and expects a data word parity error.
RCWCT	RC10-DF10 Word Count Register Test Using the disk, this test proceeds in the following manner. First, it fills the buffer with a count pattern for as many words as the channel is set up to transfer. The rest of the buffer is filled with 0s. Then the program writes a file consisting of one word, fills the buffer with the null pattern, and reads the file. Next, it checks the buffer to make sure that only one word was read. Finally, the word count is shifted one place left and the whole process is repeated until it runs out of word count register or sufficient memory buffer for the entire word count.

Table 6 DDDFA Test Summary (RH10)

Test	Description
DF10T	RH10 DF10 Basic Jump-Terminate Tests This test loads a DF10 command list with a jump to a terminate - then tests. This is done with all initial addresses from 20-776 (except 40-57).
DF10T2	RH10-DF10 Terminate Test In RH10/DF10 diagnostic mode, one word is written. Word should load into RH10 AR register. Channel should terminate, and RH10 CB register should be clear. RH10 status should indicate AR FULL and BUSY.
RHCON	RH10-DF10 Memory Connection This test checks that the DF10 can transfer 1K words from each memory bank that the CPU is connected to. A bank is considered to be 8K words. This takes into account that the MA10 has upper and lower 8K deselection switches that may be set to deselect.
RHNXM	RH10-DF10 Nonexistent Memory Test The DF10 is checked for connection to more memory than the CPU. DF10 is started with a control word pointing to nonexistent memory. The NON-EX MEM flag should appear in the controller. If not, then the DF10 is connected to a bank which the processor is not connected to. The increment is by 8K for each test because of MA10 logic which allows a deselect of either upper or lower 8K of the stack. This test will not be executed for a KA10 with 256K of memory or a KI10, KL10 with 4096K of memory. It is suggested that the user deselect at least 1 16K bank (or 8K if MA10) in order to test NON-EX-MEM flag.
PSKTST	RH10-DF10 Skip-Read Test This test sets up control word (-17,,0) and starts DF10. Shadow (core 0-17) ACs are set to 0 and DF10 started. DF10 should skip 17 words and terminate without writing any information received from the device into memory. If the DF10 does write it will probably start with location 0, so check that the shadow (core 0-17) ACs are still 0.
RHMUL	RH10-DF10 Multiple Control Word Fetch This test fetches multiple control words from memory. Control word is a skip of 17 words done 12 times. It checks that the shadow (core 0-17) ACs are not written because a skip should never write anything - if it does it will probably go into core 0 to 17. The test also checks that the written control word is correct; i.e., the DF10 fetched all control words and did not terminate early. Causes for early termination are as follows. 1. The DF10 did not fetch a control word. 2. The DF10 had a control word parity error. 3. The DF10 jumped to core 0 which has a zero in it and is a DF10 terminate. 4. The DF10 had a nonexistent memory which could occur when the DF10 had a jump to 0 and a -1 was in location 0.
RHXFER	RH10-DF10 Data Transfer Test This test transfers data of various types to each memory bank connected to the system. Each pattern is done ten times in each bank connected.

Table 6 DDDFA Test Summary (RH10) (Cont)

Test	Description
RHPAR	RH10-DF10 Parity Error Detection Test This program tests the parity error detection logic of the DF10. It reads from the RS04 and writes even parity into memory. Then it checks that the DF10 did in fact write even parity into memory. Then it uses the data written as a control word and expects a control word parity error. Then it uses a good control word pointing to buffer and expects a data word parity error.
RHFET	RH10-DF10 Control Word Fetch Test This test fetches control words from all of memory starting at the first free memory address beginning at the end of the program. It fetches a control word (the control word is a terminate command for the DF10). The test will call control words up through memory in 1K jumps until all available memory has been tested.
KSJP	RH10-DF10 Jump-Skip Test JUMP,+1,SKIP,JUMP,+1,SKIP,....etc. is loaded into memory up to a 64(10) sector skip, then the DF10 is started. Each successive skip results in skipping one more sector than the previous skip until the command list reflects a 64(10) sector skip. The DF10 written control word is then checked. Incorrect termination can result from nonexistent memory, control word parity errors, and disk errors.
RKSD	RH10-DF10 Skip-Read Test A control word chain is set up so that the DF10 does a skip followed by a read and checks that a skip followed by a read does not continue to skip.
RHWCT	RH10-DF10 Word Count Register Test The DF10 word count register test uses RS04 drives. This test proceeds in the following manner. First, it fills the buffer with a count pattern for as many words as the channel is set up to transfer. The rest of the buffer is filled with 0s. Then it writes a file consisting of one word, fills the buffer with the null pattern, and reads the file. Next, it checks the buffer to make sure that only one word was read. Finally, the word count is shifted one place left and the whole process is repeated until it runs out of word count register or sufficient memory buffer for the entire word count.

GENERAL INFORMATION

Code	DDDXA.A10
Title	PDP-10 DX10 Data Channel Diagnostic (Part 1)
Abstract	This is the first of a two part diagnostic designed to test a DX10 data channel. This diagnostic does not require a TX01 magnetic tape controller or any TU70 tape units, although the diagnostic will run if they are present. This part of the diagnostic tests the I/O bus interface, memory interface, PDP-8A interface and the PDP-8A itself. The PDP-8A is tested first by test routines in this diagnostic and then by running the PDP-8A diagnostics.
Hardware Required	KA10, KI10 or KL10 mainframe/32K of core (minimum)/DX10
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).

Restrictions

1. In user mode, a monitor version 6.02 or greater is required, or an equivalent which supports the TAPOP, UUO operation codes 23 and 24. A monitor which supports TAPOP, UUO operation code 1034 is required for configurations where one DX10 accesses more than one tape controller.

2. The PDP-8A diagnostics (DJKAA, DJMSA, and DJEXA) must be on the same load medium as DDDXA.

3. There are several operations that are not tested in user mode because of the possibility that these operations may affect the performance of the system and/or other users' jobs. The following is a list of the operations not tested in user mode.

   a. The IOB reset signal

   b. Priority interrupts

   c. Memory addressing beyond the physical core assigned to the job.

4. The feature register switches must be set up correctly before starting the diagnostic. The ECO bits that are used by this diagnostic are listed below.

   a. Bit 19 - This switch must be in the OFF position to allow the entire diagnostic to run. It should only be set to the ON position when the DX10 is connected directly to a KA10-style memory bus (18-bit memory addressing). This switch should never be set if more than 256K of memory is on the system. When it is set, the diagnostic will never attempt a memory reference with address bits 14 to 17 set.

   b. Bit 26 - This switch is used to determine if the slow clock feature is available on the M8599 module. The feature is first available on Rev. E of the module. This bit should be on if the slow clock feature is installed. It should always be off if the slow clock feature is not installed.

Notes

1. This diagnostic will run in both exec and user mode. When in user mode, the diagnostic will be locked in core and will run with the user IN-OUT bit set in the processor status word.

2. User mode privileges

   a. The following privileges are required to run this diagnostic in user mode.

      POKE
      PEEK
      LOCK
      TRPSET
      SET CPU (on dual processor systems)

   b. All magtape units connected to the DX10 to be tested must be assigned or assignable to the job running the diagnostic.

3. This diagnostic can also be run with a channel bus loopback plug installed in the DX10 to provide a better test of the DX10's connection to each of the channel bus lines in production.

4. If it is necessary to look into the listing to determine exactly how a routine is exercising the DX10, an understanding is required of the several instructions that control the DX10 but are not normal PDP-10 I/O instructions. Refer to the listing on microfiche.

Loading and Starting Procedures

This diagnostic must be loaded and started via the DIAMON program.

Control Switches

Refer to Table 1.

Table 1 DDDXA.A10 Control Switch Summary

Switch	State	Description
0-17		Standard (Refer to the 10/10 STD module.) Exceptions - The following switches are not implemented: 11, 13, 14, 15, 16, and 17
18	0	Normal operation
	1	Reads microcode from DX10 memory and returns to DX10 memory on test end. This switch should always be a 1 when the diagnostic was loaded from a TU70 magtape.
19		Not used
20	0	Tests DX10 no. 1 (device code 220).
	1	Tests DX10 no. 2 (device code 224).
21, 22		These switches control the operation of the PDP-8A diagnostics as follows.
	0	If switch 9=0, run each of the PDP-8A diagnostics on pass 0 and then once every eighth pass. If switch 9=1, run each of the PDP-8A diagnostics every pass.
	1	Do not run the PDP-8A diagnostics.
	2	This switch setting should be used when more extensive PDP-8A testing is desired. When DDDXA is started, RUN WHICH PDP-8A DIAGNOSTIC? (DJKKA, DJMSA, DJEXA) - will be asked. Answer with one of the diagnostic names. The selected diagnostic will then be started and allowed to run continuously. Halts will be reported in the normal manner.

Table 1 DDDXA.A10 Control Switch Summary (Cont)

Switch	State	Description
	3	Run each of the PDP-8A diagnostics every pass. This is similar to setting switch 9 but the rest of DDDXA runs in quick-verify mode.
23-34		Not used
35	0	Normal operation
	1	Channel bus loopback plug is installed in DX10 and the diagnostic will expect all outgoing lines to be looped back to the input lines. Switch 35 must not be set if the loopback plug is not installed. If this switch is set, TESTING WITH CHANNEL BUS LOOP BACK PLUG INSTALLED will be printed near the start of the diagnostic.

OPERATIONAL CONTROL

Starting Addresses

30000 Normal starting address

30006 or REENTER command in user mode, abort test. If in user mode, release the DX10.

30014 Start PDP-8A console program.

Verify that memory map printed agrees with the memory configuration of the system. If console data switch 35 is set, TESTING WITH CHANNEL BUS LOOP BACK PLUG INSTALLED will be printed to warn of accidental setting of this switch.

The diagnostic will run to completion and report any errors when they occur. If the thumb wheel switches on the DX10 maintenance panel are set to 17, a decrementing pass count will be displayed in the lights.

To force this diagnostic to stop and return the microcode to the DX10 memory, the abort must be requested by either setting switch 0 or typing an altmode.

Terminal Control Commands
Terminal control commands are described in Table 2.

PDP-8A Console Control Program
This program will enable the operator to control the PDP-8A microprocessor in the DX10 similar to operating a PDP-8 console. Control is through the terminal on the PDP-10.

This program is started by setting the address switches on the PDP-10 to 30014 and pressing START. The commands are described in Table 3.

DDDXA TEST SUMMARY
The tests performed by DDDXA are summarized in Table 4.

Table 2 DDDXA Terminal Control Commands

Command	Description
$	<ALTMODE> An altmode character can be typed at any time to cause the diagnostic to halt at the end of the current test. This is the recommended method of stopping the diagnostic. An altmode typed to the PDP-8A console program will restart the diagnostic. A second altmode must then be typed to abort the diagnostic.
↑C	A control C in user mode will release the DX10 and abort the test.
↑O	A control O can be typed to suppress the printing of an error message. All typeout except forced typeout will be suppressed until a new error is encountered. The program continues and the suppressed typeout is lost.
↑Q	A control Q can be typed to continue the printing of a message stopped via a control S.
↑S	A control S can be typed to stop the printing of an error message on a video display terminal without losing the rest of the typeout. Type a control S before the first line of the message disappears off the top of the screen and read the message. Then type a control Q to allow the printing to continue. While typeout is stopped with a control S the program will not run.

Table 3 PDP-8A Console Control Commands

Command	Description
$	<ALTMODE> An altmode may be typed at any time to restart the diagnostic.
A	A<CR> The contents of the AC is typed as data.
^	^<CR> The address in the CPMA is typed.
adr.	177.<CR> Load the CMPA with the address specified.
adr/	200/ Print the data in the specified address. If no address is typed, the address in the CPMA is assumed. The terminal remains in this position waiting for another command. The four options follow. <CR> No operation. Exit the examine routine. 7402<CR> Deposit the data in the selected ADR. Then exit the examine routine. <LF> Examine the next sequential location in PDP-8 memory and again wait for another command. 2014<LF> Deposit the data in the selected ADR. Then examine the next sequential ADR in PDP-8 memory and wait for command. Any other command may be typed. The program will exit the examine routine, then process the new command as normal.

Table 3 PDP-8A Console Control Commands (Cont)

Command	Description
adrl:adr2/	200:250/ The contents of every address starting with adrl through adr2 is typed. If adrl is omitted (only the colon typed), adrl is assumed to be 0000. If adr2 is not typed (no number between the colon and slash) adr2 is assumed to be 7777.
adrZ	272Z<CR> Clear the address specified.
adrl:adr2Z	100:300Z<CR> Clear all locations starting with adrl through adr2. Unspecified addresses are treated as in the adrl:adr2/ command. Also see the Z command.
C	C<CR> The position of the continue switch is typed as data (0 or 1).
0C	0C<CR> Clear the continue switch.
1C	1C<CR> Set the continue switch. If the switch is already set, it is cleared and set again.
Code I	4I<CR> The contents of the specified IBus register is typed as data. If no code is typed, IBus selection code 0 is assumed. The terminal waits here for another command. The options are the same as for the slash command.
D	D<CR> The contents of the data bus are typed as data.
H	H<CR> The position of the halt switch is typed as data. Data will either be 0 or 1.
0H	0H<CR> Clear the halt switch.
1H	1H<CR> Set the halt switch.
L	L<CR> Clear all of PDP-8A memory, clear all PDP-8A switches and, if this program was loaded with DIAMON, ask FILE NAME? -. At this time the name of a file must be typed. The file will be read from the same device from which DIAMON loaded this program. If this program was loaded from paper tape on a KA10 or KI10, the paper tape in the paper tape reader will be read. The file is loaded into PDP-8A memory and PDP-10 memory, then the two memories are compared. Any differences are reported as verify errors.
M	M<CR> The contents of the MQ register are typed as data.
P	P<CR> The contents of the processor status register are typed as data.
R	R<CR>

Table 3 PDP-8A Console Control Commands (Cont)

Command	Description
	Reset the DX10.
S	S<CR> The position of the single-step switch is typed as data (0 or 1).
0S	0S<CR> Clear the single-step switch.
1S	1S<CR> Set the single-step switch.
Z	Z<CR> Clear all of PDP-8A memory.

Table 4 DDDXA Test Summary

CONO/CONI CONNECTIONS TEST

PIA Register - Check connection of DX10 to I/O bus using PIA register. Check all combinations setting and clearing the PIA.

Interrupt Test - Test that all interrupt flags and PI request lines are inactive.

Device Codes - Check device selection logic. Make sure the DX10 will respond to only one device code by setting PIA register to 7 and then issuing (CONO 0) to devices with selection codes that differ from channel. The channel PIA should still be equal to 7 unless it responded to the CONO.

CCR - Check all CCR bits that can be set and cleared with a CONO instruction. Check that all bits which should be cleared by a CONO instruction are clear.

DATAO/DATAI CONNECTIONS TEST

IBus Selection - Check IBus selection logic using unused register 17 and ICPC register (register select 16).

ICPC - Check that each bit in the ICPC can be read correctly over the IBus.

Memory Register - Check that each bit in the memory register can be written and read correctly over the IBus.

MP Register - Check that each bit can be written and read in the selected microprocessor register.

DAC - Check that each bit in the data address counter can be written and read correctly.

CPC - Check that each bit in the channel program counter can be written and read correctly.

BC - Check that all bits in the byte counter can be written and read correctly. First test bits 18 to 31 for all combinations. Then test bits 32 to 35 for all combinations; first when selecting DAC bits 14 to 17, then when selecting CPC bits 14 to 17. Then check that both DAC and CPC can be selected.

DR - Check that each bit in the data register can be written and read correctly. This is a 36-bit register that is written and read by a pair of DATAO or DATAI commands.

CHN Bus - Check that CHN BUS can be read over the IBus. Check that each bit in the bus out register can be written and read correctly. Check that the BUSO can be read by BUSI with loop enable set. If loopback plug is installed, check that BUSO is always read by BUSI.

CHN TAG - Check that CHN TAG register can be read on IBus. Check that only TAG bits can be changed and that unused bits are read as 0s. Check that each bit in the tag out register can be set and cleared.

Table 4 DDDXA Test Summary (Cont)

Feature Register - Check that the feature register will not change when written. Type contents of the feature register the first time this routine is executed.

PDP-8A CONTROL TEST

8A Switches - Test that the switches for the PDP-8A can be set and cleared using the IBus MP control register. Check that setting the continue switch will cause a MEM start pulse to trigger the 8A error flag timer by waiting to see if the 8A error flag sets. Check that no interrupt occurs when the 8A error flag sets if the PIA register equals 0.

Priority Interrupts - Cause an interrupt on each interrupt channel using 8A error flag. Check that an interrupt can be caused on each channel and that the proper channel is interrupted.

CPMA - Check IBus register 5 for all 0s in unused bit positions, and for run bit to be 0.

8A Memory - Check that data can be written into 8A memory. Check that each location can be loaded and read back with the proper data pattern.

IBUS REG 15 - Select IBus register 15 and check that unused bits are all 0. Select the AC register and look for all 0s after initialize.

Run Indicator - Check that the 8A's run indicator will turn on and off when the 8A is started and stopped.

PDP-8A I/O INSTRUCTION TEST

Initialize - Test that the PDP-8A can execute a INT instruction under control of the PDP-10. Check that the INT instruction causes an initiate pulse in the DX10.

Set STAT AVAIL - Test that the 8A can set STAT AVAIL flag with I/O instruction.

Clear CLR/CONT - Test that 8A I/O instruction clear CLR/CONT will clear the clear and continue flags.

Set Timer - Test the set timer instruction by computing how long the 8A can run before the timer flag sets and then having the 8A set the timer and see if the timer flag clears for four times the timer cycle.

8 REGISTERS

8A IBUS SELECTION

LOAD TAG OUT

10-MEMORY PARITY

See if DX10 can select a location by attempting a read of that address twice; first with 0s in it, then with 1s in it. Check if memory register reads any different data.

10-Memory Write - Test that 8A can write into memory.

Bus Parity - Check the parity bits and logic on CHN bus.

Enable Readin - Check that enable readin instruction will cause the proper data to appear on the I/O bus.

Set Readin - Test that set RDI instruction will cause the readin flip-flop to set. Check if the ROM code was placed into the RAM correctly.

ERROR SUMMARY
Errors are reported on the terminal or the line printer (if switch 4 is set). An error report contains the program pass count if not 0, the PC of the error call, the contents of the data switches, the module most likely to be causing the error, the test name, correct and actual data (if applicable to the error) and a description of the failure. The description of the failure and test routine is printed so that it will not be necessary to study the actual code in the listing.

The module most likely to be causing the failure is listed with each error report. If it is not possible to select only one module, more modules are listed in the order of decreasing probability. This is an attempt to isolate most failures to a module and is provided with no guarantees. It is recommended that each module be replaced with a known good module in the order specified in the list. If the failure does not change, then use the description of the test routine and failure results.

The following is a sample error report.

```
PC=XXXXXX SWITCHES=XXXXXX XXXXXX
EXPECTED FAILING MODULE:
    M8599 IN SLOT 9
ERROR IN TEST NAME
CORRECT: XXXXXX XXXXXX
ACTUAL:  YYYYYY YYYYYY
DISCREP: ZZZZZZ ZZZZZZ
DESCRIPTION OF TEST ROUTINE AND FAILURE.
```

PDP-8A Diagnostic Errors
The PDP-8A diagnostics have no error reporting or looping facilities. The only error indication is by halting. For this reason, the DDDXA diagnostic runs each of the PDP-8A diagnostics for a period of time and watches to see if they ever halt. If a PDP-8A diagnostic does halt, (with the exception of the one expected halt in DJKKA) all information about the state of the PDP-8A is printed. Console switch 6 is then examined. If switch 6 is set, the operator is asked if the console program should be started. From the console program the operator can perform the steps recommended in the PDP-8A diagnostic write-up. Refer to Table 3 for instructions on how to use the console program.

A sample PDP-8A diagnostic error message follows.

```
PC=XXXXXX SWITCHES=XXXXXX XXXXXX
EXPECTED FAILING MODULE:
    M8315 IN SLOT 3
    M8311 IN SLOT 6 ROWS A-3
ERROR IN PDP-8A DIAGNOSTIC - DJKKA
PDP-8A HALTED. CPMA = XXXX   MEMORY DATA = XXXX   DATA BUS = XXXX
               AC   = XXXX   MQ = XXXX
               LINK = X

WANT TO TROUBLE-SHOOT USING THE PDP-8A CONSOLE PROGRAM? Y OR N
<CR> -
```

GENERAL INFORMATION

Code	DDDXB.A10
Title	PDP-10 DX10 Data Channel Diagnostic (Part 2)
Abstract	This is the second of a two part diagnostic designed to test a DX10 data channel. This diagnostic does not require a TX01 magnetic tape controller or any TU70 tape units, although the diagnostic will run if they are present. This part of the diagnostic tests the channel bus control register and all associated logic, including memory transfers and silo. The last part of this diagnostic includes running a PDP-8A microdiagnostic (MAINDEC-10-DXDXA) which tests the DX10 at normal operating speed.
Hardware Required	KA10, KI10 or KL10 mainframe/32K of core (minimum)/DX10
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).

Restrictions

1. In user mode, a monitor version 6.02 or greater is required, or an equivalent which supports the TAPOP, UUO operation codes 23 and 24. A monitor which supports TAPOP, UUO operation code 1034 is required for configurations where one DX10 accesses more than one tape controller.

2. There are several operations that are not tested in user mode because of the possibility that these operations may affect the performance of the system and/or other users' jobs. The following is a list of the operations not tested in user mode.

   a. The IOB reset signal

   b. Priority interrupts

   c. Memory addressing beyond the physical core assigned to the job.

3. The feature register switches must be set up correctly before starting the diagnostic. The ECO bits that are used by this diagnostic are listed below.

   a. Bit 19 - This switch must be in the OFF position to allow the entire diagnostic to run. This switch should only be set to the ON position when the DX10 is connected directly to a KA10-style memory bus (18-bit memory addressing). This switch should never be set if more than 256K of memory is on the system. When this switch is set, the diagnostic will never attempt a memory reference with address bits 14 to 17 set.

   b. Bit 26 - This switch is used to determine if the slow clock feature is available on the M8599 module. The feature is first available on Rev. E of the module. This bit should be on if the slow clock feature is installed. This bit should always be off if the slow clock feature is not installed.

Notes

1. This diagnostic will run in both exec and user mode. When in user mode, the diagnostic will be locked in core and will run with the user IN-OUT bit set in the processor status word.

2. User mode privileges

   a. The following privileges are required to run this diagnostic in user mode.

      POKE
      PEEK
      LOCK
      TRPSET
      SET CPU (on dual processor systems)

   b. All magtape units connected to the DX10 to be tested must be assigned or assignable to the job running the diagnostic.

3. This diagnostic can also be run with a channel bus loopback plug installed in the DX10 to provide a better test of the DX10's connection to each of the channel bus lines in production.

4. If it is necessary to look into the listing to determine exactly how a routine is exercising the DX10, an understanding is required of the several instructions that control the DX10 but are not normal PDP-10 I/O instructions. Refer to the listing on microfiche.

Loading and Starting Procedures

This diagnostic must be loaded and started via the DIAMON program.

Control Switches

Refer to Table 1.

Table 1 DDDXA.A10 Control Switch Summary

Switch	State	Description
0-17		Standard (Refer to the 10/10 STD module.)
		Exceptions - The following switches are not implemented: 11, 13, 14, 15, 16, and 17.
18	0	Normal operation
	1	Reads microcode from DX10 memory and returns to DX10 memory on test end. This switch should always be a 1 when the diagnostic was loaded from a TU70 magtape.
19		Not used
20	0	Tests DX10 no. 1 (device code 220).
	1	Tests DX10 no. 2 (device code 224).
21, 22		These switches control the operation of the DXDXA program as follows.
	0	If switch 9=0, run DXDXA on pass 0 and then once every eighth pass for a maximum of 2000(8) passes. If switch 9=1, run DXDXA every pass.
	1	Do not run DXDXA.
	2	Run only DXDXA and allow it to run continuously.
	3	Run DXDXA for 2000(8) iterations on every pass. Similar to setting switch 9 but the rest of DDDXB runs in quick-verify mode.

Table 1 DDDXA.A10 Control Switch Summary (Cont)

Switch	State	Description
23-34		Not used
35	0	Normal operation
	1	Channel bus loopback plug is installed in DX10 and diagnostic will expect all outgoing lines to be looped back to the input lines. Switch 35 must not be set if the loopback plug is not installed. If this switch is set, TESTING WITH CHANNEL BUS LOOP BACK PLUG INSTALLED will be printed near the start of the diagnostic.

OPERATIONAL CONTROL

Starting Addresses

30000 Normal starting address

30006 or REENTER command in user mode, abort test. If in user mode, release the DX10.

30014 Start PDP-8A console program

Verify that memory map printed agrees with the memory configuration of the system. If console data switch 35 is set, TESTING WITH CHANNEL BUS LOOP BACK PLUG INSTALLED will be printed to warn of accidental setting of this switch.

The diagnostic will run to completion and report any errors when they occur. If the thumb wheel switches on the DX10 maintenance panel are set to 17, a decrementing pass count will be displayed in the lights.

To force this diagnostic to stop and return the microcode to the DX10 memory, the abort must be requested by either setting switch 0 or typing an altmode.

Terminal Control Commands
Terminal control commands are described in Table 2.

PDP-8A Console Control Program
This program will enable the operator to control the PDP-8A microprocessor in the DX10 similar to operating a PDP-8 console. Control is through the terminal on the PDP-10.

This program is started by setting the address switches on the PDP-10 to 30014 and pressing START. The commands are described in Table 3.

DDDXB TEST SUMMARY
The tests performed by DDDXB are summarized in Table 4.

Table 2 DDDXB Terminal Control Commands

Command	Description
$	<ALTMODE> An altmode character can be typed at any time to cause the diagnostic to halt at the end of the current test. This is the recommended method of stopping the diagnostic. An altmode typed to the PDP-8A console program will restart the diagnostic. A second altmode must then be typed to abort the diagnostic.
↑C	A control C in user mode will release the DX10 and abort the test.

Table 2 DDDXB Terminal Control Commands (Cont)

Command	Description
↑O	A control O can be typed to suppress the printing of an error message. All typeout except forced typeout will be suppressed until a new error is encountered. The program continues and the suppressed typeout is lost.
↑Q	A control Q can be typed to continue the printing of a message stopped via a control S.
↑S	A control S can be typed to stop the printing of an error message on a video display terminal without losing the rest of the typeout. Type a control S before the first line of the message disappears off the top of the screen and read the message. Then type a control Q to allow the printing to continue. While typeout is stopped with a control S the program will not run.

Table 3 PDP-8A Console Control Commands

Command	Description
$	<ALTMODE> An altmode may be typed at any time to restart the diagnostic.
A	A<CR> The contents of the AC is typed as data.
^	^<CR> The address in the CPMA is typed.
adr.	177.<CR> Load the CMPA with the address specified.
adr/	200/ Print the data in the specified address. If no address is typed, the address in the CPMA is assumed. The terminal remains in this position waiting for another command. The four options follow. <CR> No operation. Exit the examine routine. 7402<CR> Deposit the data in the selected ADR. Then exit the examine routine. <LF> Examine the next sequential location in PDP-8 memory and again wait for another command. 2014<LF> Deposit the data in the selected ADR. Then examine the next sequential ADR in PDP-8 memory and wait for command. Any other command may be typed. The program will exit the examine routine, then process the new command as normal.
adr1:adr2/	200:250/ The contents of every address starting with adr1 through adr2 is typed. If adr1 is omitted (only the colon typed), adr1 is assumed to be 0000. If adr2 is not typed (no number between the colon and slash) adr2 is assumed to be 7777.
adrZ	272Z<CR> Clear the address specified.
adr1:adr2Z	100:300Z<CR> Clear all locations starting with adr1 through adr2. Unspecified addresses are treated as in the adr1:adr2/ command. Also see the Z command.

Table 3 PDP-8A Console Control Commands (Cont)

Command	Description
C	C<CR> The position of the continue switch is typed as data (0 or 1).
0C	0C<CR> Clear the continue switch.
1C	1C<CR> Set the continue switch. If the switch is already set, it is cleared and set again.
Code I	4I<CR> The contents of the specified IBus register are typed as data. If no code is typed, IBus selection code 0 is assumed. The terminal waits here for another command. The options are the same as for the slash command.
D	D<CR> The contents of the data bus are typed as data.
H	H<CR> The position of the halt switch is typed as data. Data will either be 0 or 1.
0H	0H<CR> Clear the halt switch.
1H	1H<CR> Set the halt switch.
L	L<CR> Clear all of PDP-8A memory, clear all PDP-8A switches and, if this program was loaded with DIAMON, ask FILE NAME? -. At this time the name of a file must be typed. The file will be read from the same device from which DIAMON loaded this program. If this program was loaded from paper tape on a KA10 or KI10, the paper tape in the paper tape reader will be read. The file is loaded into PDP-8A memory and PDP-10 memory then the two memories are compared. Any differences are reported as verify errors.
M	M<CR> The contents of the MQ register are typed as data.
P	P<CR> The contents of the processor status register are typed as data.
R	R<CR> Reset the DX10.
S	S<CR> The position of the single-step switch is typed as data (0 or 1).
0S	0S<CR> Clear the single-step switch.
1S	1S<CR> Set the single-step switch.
Z	Z<CR> Clear all of PDP-8A memory.

Table 4 DDDXB Test Summary

PDP-8A I/O INSTRUCTION TEST
CBC - Check the operation of the CBC register and the block transfer logic.
SILO TESTS - Test that the silo will accept a byte of data and propagate it through to the output.
TEST BLOCK TRANSFER
TEST READ OPERATIONS
TEST WRITE OPERATIONS
TEST SILO PARITY
SLOW CLOCK TEST
MICRODIAGNOSTIC TEST

ERROR SUMMARY
Errors are reported on the terminal or the line printer (if switch 4 is set). An error report contains the program pass count if not 0, the PC of the error call, the contents of the data switches, the module most likely to be causing the error, the test name, correct and actual data (if applicable to the error) and a description of the failure. The description of the failure and test routine is printed so that it will not be necessary to study the actual code in the listing.

The module most likely to be causing the failure is listed with each error report. If it is not possible to select only one module, more modules are listed in the order of decreasing probability. This is an attempt to isolate most failures to a module and is provided with no guarantees. It is recommended that each module be replaced with a known good module in the order specified in the list. If the failure does not change, then use the description of the test routine and failure results.

The following is a sample error report.

```
PC=XXXXXX SWITCHES=XXXXXX XXXXXX
EXPECTED FAILING MODULE:
    M8599 IN SLOT 9
ERROR IN TEST NAME
CORRECT: XXXXXX XXXXXX
ACTUAL:  YYYYYY YYYYYY
DISCREP: ZZZZZZ ZZZZZZ
DESCRIPTION OF TEST ROUTINE AND FAILURE.
```

PDP-8A DXDXA Diagnostic Errors
While running the microdiagnostic program (DXDXA) the 8A sets the status available flag at each end of pass and halts when it detects an error. All errors stop by executing a HLT instruction in 8A location 21. Information pertaining to the error is made available to the PDP-10 in the AC, MQ and 8 register. A sample error report follows.

```
PX=XXXXXX SWITCHES=XXXXXX XXXXXX
ERROR IN MICRO-DIAGNOSTIC TEST
PDP-8A DETECTED AN ERROR AT CPMA = XXXX
CORRECT: XXXX
ACTUAL:  YYYY
DISCREP: ZZZZ
DESCRIPTION OF TEST ROUTINE AND FAILURE.
```

The PDP-10 controls error looping of the DXDXA diagnostic by depositing in 8A location 20 the address of the error to loop on. Console switch 6 is used to determine error looping.

While DXDXA is running, it also displays a decrementing count in the DX10 lights if the thumb wheels are turned to 16. When in an error loop, an incrementing count is displayed every time the error fails again.

When it is necessary to use the 8A console program to repair an error, halt the PDP-10 on the error by settig switch 7 and then start at 30014. The microdiagnostic can also be restarted at any time at PDP-8A location 200.

GENERAL INFORMATION

Code	DDLPA.A10
Title	PDP-10 Line Printer Diagnostics
Abstract	The line printer test program is designed to verify the operational status of the line printer and its control logic. The program consists of five major parts. Part A (exec mode only) will explicitly diagnose failures in the control logic or the printer. As specific faults are identified, the operator will receive an error message of further instructions. Part B (user and exec mode) will produce a series of printout patterns. The operator must inspect these printouts to detect any errors in the printing logic or the printing mechanism. Each of the tests in Part B produces a printout which exercises specific printer functions. Part C (exec mode only) consists of three tests that time line printer printing speed, using the clock interrupt. Part D (user and exec mode) consists of tests 23 and 24 and is used for troubleshooting. In test 23 any terminal input supplied by the operator is repeatedly printed out on the line printer. Using this test, the operator can send to the printer combinations of characters which were previously found to cause print errors. Only the dispatch routine has been written for test 24; i.e., the operator may call for this test and control will go to the beginning address of an area reserved for test 24. Presently this will result in a printed message informing the operator that test 24 must be written manually. The operator may write a test routine that exercises specific line printer faults which were discovered previously. Part E (exec mode only) is the VFU test which exercises the programmable vertical format unit of the LSP10. It allows the operator to specify vertical spacing to exercise the device.
Hardware Required	KA10, KI10 or KL10 mainframe/32K of core (minimum)/BA10 and LP10A or LP10F/LSP10 and LP05
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	None
Notes	1. The test was designed specifically for line printers that use the BA-10 control and that conform to DEC purchase specification number 20-05485. Presently most printout tests are meaningful only for 132-column printers. However, program location COLSET may be changed to MOVEI 0, 170 (201 000 000 170) if the test is to run on 120-column printers. The program may be called automatically by the diagnostic monitor. In that case, only test 1 is performed, as many times as directed by the monitor. 2. Placing switch 28 into the (1) position while test 1 is being performed will stop program execution after the end of the "last test" and return control of the test sequence back to the operator. Program types: (CR)(LF)*. Typing one altmode into the terminal while any one of the printout tests is being peformed will accomplish the same.

3. Two altmodes typed into the terminal at any time during line printer printout will return test sequence control to the operator immediately [program types: (CR)(LF)*].

Loading and Starting Procedures — Standard (Refer to the 10/10 STD module.)

Control Switches — Refer to Table 1.

OPERATIONAL CONTROL
Upon starting, DDLPA prints:

<CR>,<LF>*

Type N-M <CR> onto the terminal. N is the number of the first test to be performed. M is the number of the last test to be performed. Refer to Table 2 for a description of each test. If only one test is to be performed, -M may be omitted.

Typing only <CR> will repeat the tests that were previously performed.

Typing "A <CR>" will execute all possible tests with the exception of the timing and troubleshooting tests (1 to 17 in exec mode, 3 to 17 in user mode). Typing "D <CR>" will print the contents of the print drum.

DDLPA TEST SUMMARY
The individual tests performed by this diagnostic are summarized in Table 2.

Table 1 DDLPA Control Switch Summary

Switch	State	Description
0-17		Standard (Refer to the 10/10 STD module.)
18-23		Not used
24-25		Set to number of characters 24 25 0 0 64-character printer 0 1 95- or 96-character printer 1 1 128-character printer
26-27		Not used
28	1	The program will continually repeat the test sequences.
29	1	Print out message in test 23 explaining the exceptions to the test.
30	1	Modify the printouts of tests 3, 44, 12, and 13, such that spaces are inserted for easier detection of errors. The printer is exercised more severely without spaces.
31-33		Not used
34	1	Set for flash error checks (off-line).
35	1	Set this switch for line printers that are other than the normal 132 columns. With the switch set the program will ask for the number of columns to be inputted on the terminal.

Table 2 DDLPA Test Summary

Test	Description
PART A	Exec Mode Only
1	This test checks specific control logic faults. In case of a computer-detected error, program will type out an error message that indicates specifically the area of the fault. Part of this test requires operator intervention. The operator is told either to actuate switches or to inspect line printer printout.
2	Checks whether the logic associated with turn-on and turn-off of power works properly. Operator intervention is necessary.
PART B	User and Exec Modes
3	Produces a printout of all line printer characters, one line of the same character at a time. If switch 30 is set the characters are printed in groups of two lines, with two blank lines between groups for easier detection of printing errors. If the printer has a 64-character set, all lowercase letters (ASCII 140 to 176) will be printed as uppercase letters (ASCII 100 to 136). Rubout (ASCII 177) will be ignored on all but full 96-character set printers.
4	Produces a ripple printout; i.e., a sequence of different characters in each line, where the printout of the following line is offset by one column to the right with respect to the previous line. If switch 30 is set the characters are printed in groups of four with four spaces between groups. If the printer has a 64-character set, all lowercase letters (ASCII 140 to 176) will be printed as uppercase letters (ASCII 100 to 136).
5	Produces one page of E printout to check vertical alignment of the line printer.
6	Produces one page of M and a page of MEBF printout to check horizontal alignment of the line printer.
7	Sends horizontal tabs to the line printer which are preceded by 0, 1, 2, 3, 4, 5, 6, and 7 characters since the last tab stop. A TAB YARDSTICK is printed also, which enables the operator to check whether the character following the tab gets printed in the correct position.
10	Sends all allowable vertical format control characters to the line printer together with a message that specifies the number of vertical spaces these characters must produce. This test will work correctly only if a DEC standard format tape is used.
11	Sends character strings of 0s separated by + signs to the line printer to: a. Check the correct operation of the printer's overflow correction logic, and b. Indicate easily whether the printer has 120 or 132 columns.
12	Sends alternately the characters "SPACE" (ASCII code = 040 = 0100000) and "-" (ASCII code = 137 = 1011111) to the printer to check on errors in the character handling logic, mainly the shift buffers.
13	The purpose of this test is the same as that of test 12. However, the characters are U (ASCII code = 125 = 1010101) and * (ASCII code = 052 = 0101010)
14	Sends alternately RUBOUT (ASCII code 177) and NULL (ASCII code 000) to the line printer. Same purpose as test 12; however, since both are nonprinting characters, the printout should produce only blank lines.

Table 2 DDLPA Test Summary (Cont)

Test	Description
15	Exercises printing and spacing of the line printer. First a series of nine number strings separated by carriage returns, are sent to the line printer. They should be printed on the same line (print and space with inhibit space).
16	Exercises the column counter. The program supplies one character for the first line, two for the next up, etc. to 132 for the last line.
17	All illegal characters except RUBOUT are sent to the line printer, one line full of each character, together with the respective ASCII code printout. The page should be blank except for the ASCII code identifiers.
PART C	Exec Mode Only
20	Printing speed is timed in this test using the clock interrupt. The line printer is supplied with the full character set contained in each line. Printing and spacing is accomplished by carriage return, line feed (=DC3). A displayed message informs the operator of the printing and spacing method and the measured line printer speed.
21	Similar to test 20, except that printing and spacing is accomplished by a line feed (=DC3) only.
22	Similar to test 21, except that only 36 sequential characters are printed for a maximum print rate test.
PART D	User and Exec Mode
23	A troubleshooting test: Any input typed in by the operator is repeatedly printed out on the line printer, with the following exceptions: ↑Z — Stops input from terminal. The line printer then repeatedly prints the previous terminal input. <ALTMODE> — Executes a local <CR> <LF> only. <LF> — Although only <LF> is sent to the line printer, the character echoes as <LF> <CR> on the terminal because the line printer will add <CR> when needed. <CR> — In exec mode, echoes as <CR> and sends <CR> to the line printer. In user mode, because of the operation of the monitor, <CR> is echoed as <CR><LF>, <CR>. <CR><LF> is sent to the line printer. A <CR> only can be obtained by ↑A. ↑A — Sends <CR> to the line printer and terminal, does not echo ↑A on the terminal. Use (ALTMODE> for local <CR><LF> if desired. DC0 to DC4 — Must be typed in as ↑P to ↑T. Sends the requested format character to the line printer except that it cannot send DC0 in user mode. In exec mode, all five characters echo as ↑P to ↑T. In user mode, they do that also, but only after a break character has been typed (delayed echo). <ALTMODE> is suggested to get an echo as soon as possible. ↑J — Send a VT (vertical tab) to the line printer. ↑L — Send a FF (form feed) to the line printer. ↑N — Allows the octal equivalent of a character (in ASCII) to be sent to the line printer.

Table 2 DDLPA Test Summary (Cont) *

Test	Description
	Use this format: ↑N123,456,XXXXXXXX where 123 is the ASCII octal equivalent and 456 is the decimal equivalent of character iteration. XXXXXXXX is the continuation of normal keyboard input. Example - to print 50 characters of 0: ↑N060,50, Example - to print 1 character of 0: ↑N060,, Example - to send ABC and octal of 0: ABC↑N060,, Using this test, the operator can send to the printer combinations of characters which have been found before to cause printing errors. To terminate line printer printout, the operator must type an altmode. The program responds by typing <CR><LF>**. Then the following operator commands are accepted. R (R <CR> in user mode) will cause the line printer to repeat the previous printout. N (N <CR> in user mode) will repeat test 23 (new line printer printout as typed in on the terminal). F (F <CR> in user mode) will exit from test 23 and return control of the test sequences back to the operator (program types: <CR> <LF> *).
24	Must be written by the operator. Refer to CONSTRUCTING TEST 24 which follows the error message summary.
Part E	Exec Mode Only
25	When this test is started the following message is printed: INSURE THAT THE LPT PAPER IS AT TOP-OF-FORM, THEN PRESS ANY CONSOLE KEY When the paper is correctly positioned, pressing any console key causes the following message to be printed: TEST MODE (S OR I)<CR>: S is struck if the load standard format bit (CONO LPT, bit 26) is to be tested. I is struck if the operator desires to specify the channel spacing. If I is struck, the following message is printed: 6 OR 8 LINES/INCH: Respond with the desired printer setting. The following message is then printed: FORM LENGTH 1<18 INCHES: Respond with a decimal number determined by the length of the paper form. The program then prints one of two messages determined by the test mode (S or I). If in S (standard) mode, all parameters are predetermined and all 8 channels have standard spacing. The program now cycles on the following command:

Table 2 DDLPA Test Summary (Cont)

Test	Description
	CHANNEL (1-8)<CR>: Every legal response will cause the printer to output a pattern determined by the spacing for that channel. If in I (individual) mode, the program cycles on the following command: CHANNEL: This is answered by a 2-part response in one of four modes. Mode 1 is: A-S Where A is the channel to be tested, and S sets standard format for channel A only Mode 2 is: A-/N Where A is the channel to be tested, and N is the number to be divided into the page length to determine spacing. For example: channel: 3-/4 would print lines 1, 5, 9, 13, 17, etc. when channel 3 is selected (see mode 4). Mode 3 is: A-N1, N2, N3, etc Where A is the channel to be tested, and N1, N2,... are line numbers to be printed. Numbers must be in increasing order and not greater than the page length. Mode 4 is: A-G Where A is the channel to be tested, and G is the go command that starts channel A printing with the pattern preselected by modes 1, 2, or 3. NOTES 1. All channels to be tested must be filled by one of the three command modes. 2. Line 1 is always printed, even if not specified.

Table 3 ASCII Representation of Line Printer Characters

ASCII	Function Symbol	ASCII Uppercase		ASCII Lowercase	
011	Horizontal tab	100	@	140	@
012	Line feed	101	A	141	A
013	Vertical tab	102	B	142	B
014	Form feed	103	C	143	C
015	Carriage return	104	D	144	D
020	DC0	105	E	145	E
021	DC1	106	F	146	F
022	DC2	107	G	147	G
023	DC3	110	H	150	H
024	DC4	111	I	151	I
040	(Space)	112	J	152	J
041	!	113	K	153	K
042	"	114	L	154	L
043	#	115	M	155	M
044	$	116	N	156	N
045	%	117	O	157	O
046	&	120	P	160	P
047	\	121	Q	161	Q
050	(	122	R	162	R
051	)	123	S	163	S
052	*	124	T	164	T
053	+	125	U	165	U
054	/	126	V	166	V
055	-	127	W	167	W
056	.	130	X	170	X
057	/	131	Y	171	Y
060	0	132	Z	172	Z
061	1	133	[	173	}(Right brace)
062	2	134	\	174	|(Vertical bar)
063	3	135	]	175	{(Left brace)
064	4	136	^(Circumflex)	176	(Tilde)
065	5	137	_(Underbar)	177	←(Back arrow)
066	6				
067	7				
070	8				
071	9				
072	:				
073	;				
074	<				
075	=				
076	>				
077	?				

ERROR REPORTS
Only errors that occurred during the tests of part A (tests 1 and 2) can be detected by the computer. In case of such an automatic error detection, the program will take the following action:

Unless switch 3 is set an error message of the following general form will be typed out:

PC = XXXXXX
C(AC) = XXXXXXXXXXXX
E = XXXXXX

Where PC = program counter when error occurred. Using the listing, the section of the program can be found that caused the error to show up. The section will be headed by comments that explain in detail the possible line printer fault conditions for:

E = the specific error number

C(AC) = the contents of the AC specified in the instruction at location PC. Where this number provides useful information, it is so stated in the error comments.

If switch 6 is set and switch 7 is reset, the program will enter an error loop after the error message has been typed out. This loop will repeat the instructions that caused the error to show up. Thus errors can be traced to faulty modules using an oscilloscope.

To recover from errors, the program may either be restarted, after the cause of the error has been eliminated, or switch 6 may be brought into the (0) position, if the error is still present.

In order for any further error messages to be meaningful, the cause of the error must first be eliminated. It is advisable that switch 6(0) be used only to obtain an initial picture of the state of the line printer and its control logic.

A few other self-explanatory error messages may be printed on the terminal if some condition for successful continuation of the tests is not met.

Vertical Format Control Characters
All vertical format control characters space paper with the help of a vertical format tape. When the line printer detects any vertical format control character, it prints the contents of the line printer buffer - unless the column counter is 0 (line printer buffer is empty) - and then moves paper until a hole is seen in the channel of the vertical format tape that corresponds with the control character. The spaces produced by each format control character is therefore a function of the vertical format tape used. The standard tape, for which test 10 has been written, is described below.

Control Character (ASCII)	TAPE COLUMN	NUMBER OF VERTICAL SPACES TO NEXT HOLE IN TAPE SUPPLIED
Line Feed [012(8)]	8	1
Vertical Tab [(013(8)]	7	10
Form Feed [014(8)]	1	Top of Page
DC(0)[020(8)]	2	30
DC(1)[021(8)]	3	2
DC(2)[022(8)]	4	3
DC(3)[023(8)]	5	1 Punched Every Hole
DC(4)[024(8)]	6	10

Six blank lines are left on the control tape before the top-of-page position except in the DC(3) case. In the vertical format control column corresponding to DC(3), all holes are punched to allow printing across the fold for ease of photo reproduction.

Constructing Test 24
The following guidelines should be adhered to when writing a program for test 24.

If the possibility exists that the diagnostic program is run in either user or exec mode, and if test 24 cannot be performed in both modes, the mode should be checked at the beginning of test 24. AC 13 (=UMFLAG) is 0 if machine runs in exec mode. It is -1 in user mode.

In order to do user mode line printer output, the line printer must be initialized. An existing subroutine may be called to do that by PUSHJ P,INITL. If the routine is called in exec mode, it enables channel 6 to interrupt when the line printer goes off-line.

If the test to be written has a definite end, return control to ASTRSK when it is finished. It will then be possible to call further tests.

Before returning to ASTPSK, the PI system and the I/O processor should be reset in exec mode, if they were used during test 24. In user mode, the line printer should be released.

It is suggested that existing subroutines be used by test 24 when practical. See the comments preceding the respective subroutine listings for a short description and the correct calling instruction. Some notes on specific subroutines follow.

Subroutine LOADL is useful for sending characters to the line printer. The characters are accepted one at a time and are stored in the user's buffer (user mode) or packed five to a word and then sent to the line printer (in exec mode).

Subroutine LPTMSG can be employed for printing line printer messages stored in core in ASCII form.

Subroutines TYPEL and TTYMSG work in a similar manner for the terminal.

The following methods are suggested to interrupt line printer printout either immediately or in some delayed fashion, from the terminal if desired:

User Mode

Either send <CR><LF> to the line printer via subroutine CRLFAD, or print line printer message that contains carriage returns via subroutine LPTMSG, or perform instruction INTCHK=PUSHJ P,INTRAD occasionally. The contents of PRTOUT = AC15 must be saved in SAVPRT during INTCHK if it is to be preserved. INTCHK restores AC 15 before return. Then typing a break character (ALTMODE, CR, etc.), on the terminal once will set the interrupt flag INTRPT = AC 12 to -1. Typing such a character for the second time will release the line printer and return control to ASTRSK.

Exec Mode

Use subroutine LOADL to send characters to the line printer. To interrupt, an ALTMODE can be typed into the terminal.

If subroutine LOADS is used to send characters to the line printer, a JRST to RELEASE at the end of the test will:

1. Print out the last characters sent to the line printer, even if less than five characters have been packed
2. Send a Form Feed to the line printer
3. Release the line printer in user mode
4. Jump to ASTRSK (get program ready to accept new commands).

GENERAL INFORMATION

Code — DDMMC.A10

Title — DECsystem-10 Fast AC Diagnostic

Abstract — This diagnostic will test the fast accumulators in the PDP-10. The program consists of a relatively simple test using 1s and 0s. This routine is followed by a more complex routine which uses random numbers. Tests are constructed such that each AC must exist and that references to one AC will not affect another.

The general form of the program is to load all ACs, then repeatedly cause fetch and store cycles on one. All ACs are then checked for accuracy.

Hardware Required — KA10, KI10 or KL10 mainframe/32K of core (minimum)

Preliminary and Associated Programs — Refer to diagnostic hierarchy (10/10 STD module).

Restrictions — None

Notes

1. This diagnostic operates in both exec and user mode. In exec mode it serves as a complete diagnostic and will cycle continually until stopped or until an error occurs. In user mode this diagnostic will verify the operations of the fast ACs.

2. KA10 and KI10 only - The MI program disable should be set to 0. The MI is used by the program to display a decrementing iteration count and/or error information.

3. The cycle time of the program is in the millisecond range and is therefore suitable for taking margins, vibration tests, etc.

Loading and Starting Procedure — Standard (Refer to the 10/10 STD module.)

User Mode:
Type START - iterates 10 times.
Type START 30000 - iterates effectively forever.

Control Switches — Refer to Table 1.

Table 1 DDMMC Control Switch Summary

Switch	Mnemonic	State	Description
00	ABORT	0	Normal operation
		1	Abort at end of pass
1-4			Not used
5	DING	0	Normal operation
		1	Ring bell on error
6	LOOPER	0	Normal operation
		1	Loop on error
7-8			Not used
9	RELIAB	0	Quick-verify mode
		1	Reliability mode
10-17			Not used
18	INHROT	0	Allow ROTC 1 exercising of ACs
		1	Change ROTC 1 to a ROTC 0
19-35			Not used

OPERATIONAL CONTROL
The user has control over the operation of this diagnostic other than the switches described in Table 1.

DDMMC TEST SUMMARY
The internal organization of this diagnostic is such that no test summary is necessary.

ERROR MESSAGE SUMMARY

Exec Mode KA10 and KI10 Only

Errors are in the form of halt instructions. The first halt in the first portion of the program is used to ensure the proper loading of the UUO trap at 41. Press CONTINUE to try again or correct the malfunction associated with AC0.

A second halt will occur if the contents of an AC are found to be in error. The following procedure is recommended.

1. Record the contents of MI bits 5-8. This is the number of the AC which was repeatedly interrogated prior to the failure.
2. Record the contents of MI bits 9-12 (the AC field). This is the number of the AC whose contents are in error.
3. Examine the contents of the failed AC [C(MI 9-12)]. Record this number.
4. Press continue.
5. The machine will halt a second time. The contents of the MI right half points to the correct data for the AC in question.
6. Examine the word addressed by C(MI)R. Record this word under the value recorded for the failed AC.
7. The bits which differ in items 6 and 3 are in error.
8. The PC at this time contains the location of the error.
9. Press CONTINUE to enter error loop. The MI will continue to display the ACs involved in the left and a count for each error in the right half.
10. When the malfunction has been corrected restart the program at 30000(8).

Exec Mode KL10 Only

Errors are in the form of a halt instruction which will print KL10 HALTED, PC = ######, VMA = ######.

The PC printout points to the error UUO in the program which may be referenced to the listing to obtain more information about the failure. The previous instruction before the error UUO is a CAME instruction which points to the AC and to the correct data in memory. These may be examined to get the associated data.

If a failure occurs in this diagnostic it is a fatal type error and should be detected via the KL10 basic hardware diagnostics.

User Mode

If an error occurs while in user mode the program will print:

```
?DECSYSTEM10 FAST AC DIAGNOSTIC (DDMMC)
?FAST AC FAILURE, CHECK OUT IN EXEC MODE !
```

It will then exit and the monitor examine commands may be used to obtain information about the error.

GENERAL INFORMATION

Code	DDMMD.A10
Title	Memory Diagnostic
Abstract	This memory diagnostic tests all of memory unoccupied by the program to make sure that the memory modules and associated hardware function as a complete operating system. The program consists of a series of five tests. The first test verifies the ability of the memory to operate with various bit combination data patterns. The second test verifies the ability of the memory to uniquely address any and every memory location. The third test verifies the ability of memory to detect a 1 or 0 under maximum half-select core array noise conditions. The fourth test verifies the ability of memory to write and read a single 1 or single 0 bit in a memory array of all 1s or all 0s words. The fifth test verifies the ability of memory to continuously access and read the same location without core characteristic changes due to core array heating. Any one, or a combination, of the five tests is selectable from terminal input or from the console data switches. Also, all of memory or any selected module or portion of memory may be tested. The memory diagnostic operates in executive mode or in time-sharing user mode. In executive mode it provides a complete memory checkout diagnostic, while in user mode it provides an indication of the state of the memory system.
Hardware Required	KA10, KI10 or KL10 mainframe/MA10s, MB10s, MD10s, ME10s
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	None
Notes	None
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	Refer to Table 1.

Table 1 DDMMD Console [illegible] Summary

Switch	Mnemonic	Description
0-17		Standard (Refer to the 10/10 STD module.)
18	SWCON	Use switches for test control - testing parameters input from RH switches rather than from terminal.
19	EXTDIN	Extended input format - used to specify and run on selected MA bit in fast-rate addressing, plus if in switches mode allows typein selection of specific data patterns for the data patterns test and for the WCP test.
20	WCPITL	WCP interleave - sets up WCP for interleaved modules.
21 22 23	WCPSW3 WCPSW2 WCPSW1	Module type selection (WCP test). 0 = Run all module types 1 = MA10 2 = MA10 64 X 64 3 = MB10 128 X 128 4 = MD10 5 = ME10 6 = Alternate 1s and 0s 7 = 1s and 0s checkerboard
24 25 26	TSTSW3 TSTSW2 TSTSW1	Test selection. 0 = Data patterns, address, WCP, float 1/0 1 = Data patterns 2 = Address 3 = WCP 4 = Float 1s/0s 5 = Heating 6 = Address and WCP 7 = All tests
27	INHMSK	Inhibit error checking on mask. Do not report errors in bits masked by 1s in location MASK (4037).
28	INHCMP	Inhibit data complement (R/C/W) when executing fast-rate addressing. Do read/restore instead.
29	INHBLT	Inhibit block transfer cycles.
30	INHFR	Inhibit fast-rate addressing.
31	INHRR	Inhibit read/restore cycles. WCP - RCW cycles
32	NOPARP	Inhibit parity error typeout.
33	NOERPT	Inhibit data error typeout.
34	LOCKPG	User - lock program in core
35	PNTTLS	Print totals at completion of all selected tests.

OPERATIONAL CONTROL
The test parameters may be selected by typein from the console terminal. All of memory, any memory size up to the available memory, or selected segments of memory may be selected for testing. The number, size, and module type of the memory modules may be specified. Also, the patterns to be used for the data patterns test and the data to be used for the worst-case patterns test may be specified.

Terminal input typing specifications are as follows:

1. If question asked is Y or N, type only Y or N followed by a carriage return as a response.

2. If question asked is for a digit, Y or N, type Y, N or the required digit followed by a carriage return.

3. If question asked requires a number as a response up to 6 octals may be typed, or up to 5 decimals, followed by a <CR> to delimit the number.

4. Octal typeins are:

   a. Addresses
   b. Module types.

5. Decimal typeins are:

   a. Memory size
   b. Number of modules
   c. Module size
   d. Number of segments.

The following is a sample typein and the associated computer printout. (Operator responses are in bold and follow the -.)

```
PDP-10 MEMORY DIAGNOSTIC (DDMMD)

MEMORY MAP =
FROM     TO        SIZE/K
000000   077777    32

SEGMENT TESTING? Y OR N (CR) - N

SPECIFY MEMORY SIZE TO BE TESTED IN K, 0 = ALL - 32

SPECIFY NUMBER OF MODULES, 0 = ALL SAME OR 1 TO 16 - 2

SPECIFY SIZE IN K, MODULE TYPE (0 TO 17) OF EACH
MODULE SIZE 1 - 16
MODULE TYPE 1 - 1
MODULE SIZE 2 - 16
MODULE TYPE 2 - 1
TEST SELECTION
DATA PATTERNS - 1 TO 7, Y OR N (CR) - Y
ADDRESS, Y OR N (CR) - Y
WCP - 1 TO 4, Y OR N (CR) - Y
FLOATING ONES/ZEROS, Y OR N (CR) - Y
CORE HEATING, Y OR N (CR) - Y

SELECTION COMPLETED, IS IT OK, Y OR N (CR) - Y.
```

In the example the test header was printed, the memory was mapped and the operator was asked if segment testing was desired. Operator answered no; he was then asked for memory size to be tested. Operator answered 32(K) which in this example is all available memory. He was then asked for number of modules, 2, then size and module type of each, 16(K) and module type 1(MA10), and tests to be run.

At the completion the operator verifies that the selection was correct and then testing commences.

Test Description
This section describes operator dialogue test selections and data patterns.

MEMORY MAPPING - In the first pass of the program, a map of memory is printed as it is seen by the CPU. The purpose of this memory mapping is to determine maximum memory size for the test parameters and also to point out any missing memory addresses or noncontiguous memory modules.

The memory map is printed in the following format:

```
MEMORY MAP =
FROM     TO        SIZE/K
NNNNNN   NNNNNN    NN
XXXXXX   XXXXXX    XX
```

Where:

Ns = lowest contiguous segment of core memory. This should always be the only entry and equal to all memory.

Xs = some higher segment of memory. This will occur, for example, in a 32K system where the higher module has an address greater than 1.

If there is more than one segment of memory the following is also printed:

```
MAPPING ERROR     ;NONCONSECUTIVE MEMORY MODULES?
```

If there is any nonexistent memory below 8K the following is printed:

```
MAPPING ERROR     ;NON-X-MEM BELOW 8K?
```

If there is an interleaving problem the following is printed:

```
MEMORY MAP =
FROM     TO        SIZE/K    ADR SEQ
000000   377777    128       XXX7
000000   577777    192       XXX6
000000   577777    192       XXX5
000000   577777    192       XXX4

MAPPING ERROR; MEMORY INTERLEAVING?
```

This indicates that the memory interleaving is faulty.
The first map is done with addresses ending with 7.
The second map is done with addresses ending with 6.
The third map is done with addresses ending with 5.
The fourth map is done with addresses ending with 4.

Memory mapping for nonexistent memory is done with address multiples of 1000(8). Memory mapping of existent memory is done with address multiples of 100(8).

KI10 - KL10 Only

Memory mapping on the KI10 - KL10 may be limited to unpaged memory by setting the INHPAG switch, i.e., 112K-1.

If the switch is not set, the EPMP/UPMP is set up so that relocation equals actual for addresses 112K through 256K.

SEGMENT TESTING - Segment testing allows the operator to define up to seven areas of core (i.e., banks) for individual testing.

If the operator had answered yes for segment testing, the following would have been printed from that point.

```
SEGMENT TESTING? Y OR N (CR) - Y

NUMBER OF SEGMENTS? 1 TO 7-3
STARTING ADDRESS    1-10000
ENDING ADDRESS      1-10777
MODULE TYPE         1-1
STARTING ADDRESS    2-40000
ENDING ADDRESS      2-40777
MODULE TYPE         2-1
STARTING ADDRESS    3-70000
ENDING ADDRESS      3-70777
MODULE TYPE         3-1

TEST SELECTION
DATA PATTERNS - 1 TO 7, Y OR N (CR) - Y
ADDRESS, Y OR N (CR) - Y
WCP - 1 TO 4, Y OR N (CR) - Y
FLOATING ONES/ZEROS, Y OR N (CR) - Y
CORE HEATING, Y OR N (CR) - Y

SELECTION COMPLETED, IS IT OK, Y OR N (CR) - Y
```

In this example the operator selected three segments for testing. Each segment was defined by a starting and ending address. The dialogue following segment testing remains the same.

MEMORY SIZE - Memory size can be all, or any part (in 1K increments) of the total core memory available to the system.

NUMBER OF MODULES - This question is asking for the number of different types of memory modules to be tested. For a system with four MA10S, one MD10, and five ME10s the answer would be 3. This question and the question following it (module type) are a pair.

MODULE TYPE - The module type, or WCP checkerboard, to be used for a specific memory module may be specified in one of two ways:

1. As terminal input - The terminal requests operator input. There are 0 to 17 different module types, as follows.

Noninterleaved	Interleaved
0-Run all module types	10-Run all interleaved module types
1-MA10, 16K	11-MA10s, 32K
2-MB10, 16K 64 X 64	12-MB10s, 32K 64 X 64
3-MB10, 16K 128 X 128	13-MB10, 32K 128 X 128
4-MD10, 32K	14-MD10s, 64K or 128K
5-ME10, 16K	15-ME10s, 32K
6-Alternate 1s and 0s	16-Alternate 1s and 0s
7-1s and 0s checkerboard	17-1s and 0s checkerboard

   When the memory modules are interleaved, the module size is specified as the overall size of the interleaved modules (i.e., two 16K interleaved modules would be specified as one 32K module).

2. As console switches - When the input parameters are provided from the switches, switches 20 through 23 are used. They are set up in octal as described above.

DATA PATTERNS - The word patterns to be used for the data pattern test and for the worst-case patterns test may be specified by typein when the program asks for the test selection. Instead of answering Y or N the operator may type in one of the following.

DATA PATTERNS;

Y = Run test, all patterns

N = Do not run test

1 = Run test, all 1s pattern

2 = Run test, all 0s pattern

3 = Run test, alternate bit pattern. Alternate bits, 525252 525252 on the first pass, then 252525 252525 for the second pass. (The same data is used on each pass throughout the memory area tested.)

4 = Run test, floating 1s pattern. Floating 1s, 042104210421 for the first location of the first pass. This data pattern rotates word to word throughout the memory area tested. On the second pass the initial location word is rotated one and then this data word rotates word to word throughout the memory area. This continues on subsequent passes until all data bit combinations are tested (i.e., four passes).

5 = Run test, floating 0s pattern. Floating 0s, 735673567356 for the first location of the first pass. This data then rotates word to word throughout the memory area tested. On the next pass the initial word is rotated one and the operation continues on subsequent passes until all combinations are tested (i.e., four passes).

6 = Run test, both floating 1s and floating 0s patterns.

7 = Run test, pseudorandom, parity bit check pattern. Pseudorandom parity bit check, 123456701234 for the first location of the first pass. This data then rotates word to word and is 2's complemented throughout the memory area tested. On the next pass the initial word is rotated three and the operation continues on subsequent passes until all combinations are tested (i.e., twelve passes). This pattern causes the parity bit to change word to word and should be the most comprehensive test.

MEMORY ADDRESSING TEST - The memory addressing test verifies the ability of the memory system to uniquely address any and every memory location selected for test. This is done by writing the value of each memory location into the right half of itself and the complement of this into the left half. The memory is then exercised and tested. At the completion of this procedure, the address pattern word is reversed and the value of each location written into the left half word and the complement of this written into the right half word. The memory is then exercised and tested for correct contents.

The address pattern words used are:

1. C,ADR - LH, complement address /RH, address
2. ADR,C - LH, address /RH, complement address.

The testing sequence used is the same as the data patterns test except that the address pattern words are used.

Address Testing Scheme - Memory exercising is accomplished using two different addressing schemes. These are sequential and fast-rate. Sequential is the normal method of addressing where the address is incremented by 1. Fast-rate was developed to provide a method of exercising each memory address bit at the fastest possible rate, whereas in sequential only MA35 is exercised at the fastest rate. The purpose of this type of memory addressing is to find and diagnose problems that are MA-bit sensitive.

Fast-rate addressing is accomplished as follows: a fast-rate bit is set up which is then added to the basic memory testing address to provide the actual testing address. This fast-rate bit starts at bit 35 and is used then to access all testing addresses. Then the fast-rate bit is changed to bit 34 and used again to access all testing addresses. This continues until the fast-rate bit is bit 18 (MSB of address). Each time the actual testing address becomes greater than the upper memory testing limit, if all addresses have not been tested, the initial lower testing address is incremented by 1 and then this lower address is incremented by the fast-rate bit for the next memory exercising pass.

An example of the way this operates would be: on the first pass, the fast-rate bit is bit 35; therefore, the initial lower testing address would be incremented by 1 until all addresses are tested (this is the same sequential). On the second pass, the fast-rate bit is bit 34; therefore, the initial lower testing address would be incremented by 2 until it is greater than the upper limit. Then the lower testing address is incremented by 1 and then the fast-rate bit is added to this address and the resulting addresses used until the remaining addresses in the testing area are exercised. On the next pass the fast-rate increment would be by 4, the next 8, and so on until the fast-rate bit is bit 18.

As can be seen by this, each MA bit is exercised so that with each memory access the particular bit being exercised is changed from a 0 to a 1 or from a 1 to a 0.

NOTE

To check out a particular memory addressing bit during the fast-rate addressing exercising routine, switch 19 (EXTDIN) is set for extended input during the initial typein and the selected bit is typed in. Then if switch 19 (EXTDIN) is set during the test operation the specified MA bit will be the only MA bit fast-rate exercised.

The following are the typed out request and typed in responses.

SPECIFY FAST RATE ADDRESSING MA BIT (0 OR 18 TO 35) -

0 = TEST ALL MA BITS
18 TO 35 = TEST SPECIFIED BIT

WORST-CASE PATTERNS - The worst-case patterns test verifies the ability of the memory system to detect a 1 or 0 under maximum half-select core array noise conditions. This is done by filling the memory with a discrete core array checkerboard pattern for each different type of core memory (i.e., MA10, MB10, MD10, ME10, etc.). This checkerboard pattern consists of a pattern word (all 1s or alternate bits) and its complement. The specific arrangement of this checkerboard pattern in core depends on the actual physical core stack layout and its associated crossover points. The memory is filled with the checkerboard pattern and then exercised and tested. Following this the complement checkerboard pattern is written into core and the memory exercised and tested for correct contents.

The worst-case patterns data words used are as follows.

Y = Run test, all patterns.

N = Do not run test.

1 = Run test, all 1s words and complement.

2 = Run test, all 0s words and complement. Provides the complemented checkerboard pattern.

3 = Run test, alternate 1s and 0s. First pass provides checkerboard pattern in even data bit core planes and complement checkerboard pattern in odd data bit core planes. Second pass provides the opposite odd and even data plane arrangement.

4 = Run test, both all 1s and all 0s patterns, WCP and WCP.

The testing sequence used is the same as the data patterns test except that the worst-case pattern data words and checkerboards are used.

To fully test a memory's worst-case pattern operation, the worst-case pattern must be exercised in the complete core stack. The most exhaustive test will be done only in noninterleaved memory, as the core array duty cycle will be the greatest then.

FLOATING ONES/ZEROS TEST - The floating 1s/0s test verifies the ability of the memory system to write and read a single 1 or a single 0 bit in a memory array of all 1s or all 0s words. This is done by filling the memory array with all 1s and then writing a single 1 in the rightmost position of the first word. This word is then read and compared to what was written. The single bit is then rotated left one position and written, read, and compared. This is repeated until all bit positions in the word have been checked. The tested location is then advanced to the next word and the sequence repeated. This continues until all words in the array have been checked. The entire memory array is then tested for correctness.

This process is then repeated using a memory array of all 0s words and a floating 0 testing word.

CORE HEATING - The core heating test verifies the ability of the memory system to continuously access and read the same location without losing data due to core characteristic changes resulting from core array heating. This is done by filling the memory with all 1s and then continuously reading the test location for the period of a clock cycle, after which testing advances to the next location. After all selected locations have been exercised and tested in this manner, the entire selected memory area is again tested for correct contents of all 1s.

The data words used are: 1s, all 1s.

The testing sequence is as follows.

1. Fill memory with all 1s.
2. Exercise each location for clock cycle.
3. Test all selected memory for correct contents.
4. Advance to next test.

ERROR MESSAGE SUMMARY

Memory Data Error
When a memory data error occurs, the following header and sample error information is printed.

```
MEMORY DATA ERROR
TN AS  PAT    ADDRESS    CORRECT        ERROR     FAILED BITS   PAR
1  SEQ ALTB   070777  252525 252525 252565 252525 000040 00000
2  F34 C,ADR  013447  764330 013447 764332 013445 000002 000002
3A F31 ZEROS  070777  777777 777777 000000 000000 777777 777777 P
4  SEQ ONES   070777  777777 777777 777776 777777 000001 000000
```

Where:

TN = The current test which detected an error.

1 = Data patterns
2 = Address
3 = Worst-case patterns
4 = Floating 1s/0s
5 = Core heating

AS = The current addressing scheme being used to exercise memory (SEQ=sequential, BLT=block transfer, Fnn=fast-rate on bit NN)

PAT = The current data pattern mnemonic

ADDRESS = The memory location which contains the data in error.

CORRECT = What the data should have been in that location.

ERROR = The data as read from that location.

FAILED BITS = A binary 1 indicates that the particular data bit failed. Could be a bit pickup or dropout (printed in octal).

PAR = A "P" indicates that when the failed location was reread for printout it still contained a parity error.

In the example above:

During test 1 address 070777 was found to be in error. From the example, it can be seen that bit 12 was picked up while using the alternate bit data pattern and sequential addressing.

During test 2 address 013447 was found to be in error. From the example, it can be seen that what was actually read was the data that corresponds with address 013445 and from this it can be deduced that, while exercising memory with memory address bit 34 changing with every memory access, address 013445 was incorrectly read and then written into the correct location; i.e., bit 34 was slow to set up.

During test 3 address 070777 was found to be in error. From the example, it can be seen that all data bits were dropped or that the location failed to read at all. Also, a parity error was found at the failed location.

For a worst-case pattern test failure (test 3), both data and parity, the test number is printed with a letter indicating which module is affected, as follows.

Noninterleaved	Interleaved
3A-MA10	3I-MA10
3B-MB10 64 X 64	3J-MB10 64 X 64
3C-MB10 128 X 128	3K-MB10 128 X 128
3D-MD10	3L-MD10
3E-ME10	3M-ME10
3F-Alternate 1s and 0s	3N-Alternate 1s and 0s
3G-1/0 checkerboard	3O-1/0 checkerboard

During test 4 address 07077 was again found to be in error. This time bit 17 was dropped.

In user mode the program relocation value and actual physical address will be printed for each data error.

After each error printout the program continues with the next memory location to test.

Memory Parity Error
When a memory parity occurs the following header and sample error information is printed.

```
MEMORY PARITY ERROR
TN  PROG    PC      AS   PAT        ADRCON  DATA             PARITY
3   001057  000006  F32  ALTB       027637  525212 525252    P
```

Where:

TN = The current test being run when the parity error occurred.

PROG = The current subroutine entry point in the test program (the last PUSHJ entry).

PC = The current program counter when the parity error occurred.

AS = The current addressing scheme being used to exercise memory.

PAT = The current data pattern.

ADRCON = The address control word used to access memory - normally points to the current location being tested.

DATA = The contents of the location pointed to by ADRCON. Normally will be the erroneous data which caused the parity error. If ADRCON points outside memory this will be blank.

PARITY = A "P" indicates that when the location pointed to by ADRCON was reread for printout it still contained a parity error.

In the above example, during test 3 a memory parity error occurred. It can be seen that while exercising the memory using the alternate bit data pattern and fast-rate addressing, bit 32 being the current fast-rate bit, bit 12 was dropped, causing the parity error. Also, when the current testing location pointed to by ADRCON was reread for printout it still contained a parity error. By referring to the program listing it can be seen that the current subroutine was running in the ACs, and which subroutine it was can be determined by the program printout.

If a memory parity error occurs when the PC is not in the ACs the following is also printed.

```
**********
PARITY ERROR IN PROGRAM
PROGRAM OPERATION QUESTIONABLE FROM THIS POINT
**********
```

In this case ADRCON and data are probably not valid error indications.

After each memory parity error and printout the program continues the testing sequence.

Nonexistent Memory and Channel 1 Interrupt Errors
When either of these errors occurs the following associated heading and sample error information are printed.

```
NONEXISTENT MEMORY INTERRUPT
APR             PI              FLAGS   PC      PROG
011023 013001   000000 000000   300000  000004  000755
```

or:

```
ERROR INTERRUPT
APR             PI              FLAGS   PC      PROG
011023 013001   000000 000000   300000  000004  000755
```

Where:

APR = CONI from APR

PI = CONI from PI

FLAGS = Current program flags when interrupt occurred

PC = The current program counter when the interrupt occurred.

PROG = The last subroutine entry point in the test program (the last PUSHJ entry).

Illegal UUO
When an illegal UUO is executed the following heading and sample error information are printed.

```
ILLEGAL UUO EXECUTED
UUO             FLAGS     PC        PROG
064240 000000   300000    003677    000433
```

Where:

UUO = The illegal UUO that was attempted.

FLAGS = The program flags.

PC = The program counter at which the UUO occurred.

PROG = The last subroutine entry point in the test program (the last PUSHJ entry).

Program Checksum Error
At the completion of a testing sequence the program is checksummed and this checksum is compared to the checksum obtained before the program is executed. If the checksums agree, the program continues normal operation. If the checksums do not agree the following is printed.

```
**********
ERROR IN PROGRAM, CHECKSUMS DO NOT AGREE
PROGRAM OPERATION QUESTIONABLE FROM THIS POINT
**********
```

The program then continues operation or restarts depending upon switch 7, ERSTOP.

The program checksum is obtained by adding together all program code from BEGIN1 to ENDSLD of the program. All changeable words are at the end of the program after ENDSLD (all variable data). The purpose of this is to verify that the program is indeed valid, since when testing a memory module in which the program also resides, the program could get changed due to interaction between the testing area of core and the program area of core. This could cause erroneous error reporting.

Error Totals
At the completion of the testing sequence or after each individual test, as selected by the switches, the error totals are printed. This printout facilitates data bit error determination and also address failure correlation. The printout provides the error totals per test, the number of parity errors, the data bits with pickup and dropout errors, and the address bits with associated data bit pickup and dropout errors.

A sample printout follows.

```
TEST COMPLETION, PASS COUNT 1
ERROR TOTALS:
PATTERNS      ADDRESS       WCP           FLOAT         HEATING
0             0             7             0             0
PARITY ERRORS: 7

DATA BIT FAILURES
BIT           PICKUP        DROPOUT
3             5             0
12            2             0

ADDRESS BITS WITH DATA FAILURES
BIT           PICKUP        DROPOUT
18
19
20
21
22
23            7             0
24            5             0
25            5             0
26
27
```

```
28
20
30
31
32          1           0
33          7           0
34          4           0
35          7           0
```

The sample printout indicates that at the completion of one pass, the worst-case patterns test found seven errors and there were also seven parity errors. Data bit 3 was picked up as an error five times and data bit 12 was picked up as an error twice. No other data bits had any errors. The address bits with data failures printout indicates that the address with the most errors was probably 16005 or an address combination that gives this indication.

If there are no errors the total printout prints as per the following example (if selected by the switches).

```
TEST COMPLETION, PASS COUNT 2
ERROR TOTALS: NONE
```

After the totals print out, the error counters are cleared and the program checksummed. Test operation then continues with the first/next test.

GENERAL INFORMATION

Code	DDMME.A10
Title	PDP-10 BLT/Memory Exerciser Test
Abstract	This diagnostic will exercise and test the block transfer instruction, the address registers and memory. The program loads memory so that each location contains [=(end of memory = current location), current location]. A random FROM and TO address (with TO less than FROM) is then generated and the BLT E is made by subtracting the FROM address from "end of memory" and adding the TO address. The BLT instruction is then performed. If the RELIAB switch is set and the areas do not overlap, the area that was BLT'd will be BLT'd back up and then down again eight times. The BLT is then checked by making sure the C(FROM address) equals C(TO address) up to E and that E+1 equals what it did before the BLT was performed. The pattern is complemented every fourth pass.
Hardware Required	KA10, KI10, KL10 mainframe/32K of core (minimum)
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	None
Notes	1. The cycle time of the program depends on the memory size being used. Execution time of the program may be reduced by setting the RELIAB switch. 2. The iteration count of the program is displayed in the memory indicators (MI). This count is a decrementing count and initially starts at -1 in stand-alone operation. 3. In user mode the contents of JOBREL determines the core size tested.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	Refer to Table 1.

OPERATIONAL CONTROL

Other than the switches, the user has no control over the operation of this diagnostic.

1. To thoroughly test all hardware, all test control data switches should be set to 0.

2. If an error occurs, the error message, used in conjunction with the listing and scoping if necessary, should allow the failure to be isolated and repaired.

3. When taking margins, set data switches RELIAB, NOPNT and DING. This will inhibit printouts but will allow the terminal bell to be rung when an error occurs. If the margin obtained is unacceptable, the operator may revert to standard switch settings for repair purposes.

4. Error information may be obtained quickly by printing errors on the line printer.

ERROR MESSAGE SUMMARY
If a large number of errors is encountered, only the first three are printed.

Memory protection, nonexistent memory, illegal interrupts, parity errors, etc. are handled by printout of an error message and pertinent information.

BLT ERRORS
Error occurring as a result of the block transfer instruction will be printed as follows.

MOVEM/CAME ERROR
ERROR.....MOVEM FOLLOWED BY CAME FAILED.
C(CAC) C(EAC)
700776 000776 700776 000774

This sample printout indicates that bit 34 was dropped in memory.

The error address equals E of C(CAC).

BLOCK TRANSFER ERROR

BLOCK TRANSFER ERROR
FROM TO "E" EADRES COR WORD ERR WORD C(BLTAC)
030244 020463 015016 020463 730244030244 720463020463 030244020463

This sample indicates that the BLT went too far.

ERROR RECOVERY
If halted:

1. Press CONTINUE.
2. To continue (i.e., change FROM and TO and check BLT) switch LOOPER should be set to 0.
3. To enter an error loop - set switch LOOPER. This will continue to perform the BLT but not perform the checking. Allows program to be in a tighter scope loop.

Table 1 DDMME Control Switch Summary

Switch	Mnemonic	State	Description
0-17			Standard (Refer to the 10/10 STD module.)
19	LOCKPG	0	Normal operation
		1	Lock program in core (user mode)
20-26			Not used
27-35	KSIZE		Select core size to be tested. (high-order 9 bits, low-order 9 bits are appended as 1s.)
			If 27-35 = 0, use all available core.

GENERAL INFORMATION

Code	DDMMF.A10
Title	PDP-10 Floating Zero/One Memory Diagnostic
Abstract	This diagnostic exercises memory using floating 1s and floating 0 data patterns. In exec mode the program will detect data errors only. Error messages are printed when errors occur. Switch options allow holding the present pattern and error pattern, providing a means of marginning sense amplifiers.
Hardware Required	KA10, KI10, or KL10 mainframe/32K of core (minimum)
Prelininary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	None

Notes

1. This diagnostic operates in both exec and user mode
2. The cycle time of the program depends on the memory size being used.
3. The iteration count of the program is displayed in the memory indicators (MI). This count is a decrementing count and initially starts at -1 in stand-alone operation.
4. In user mode the contents of JOBREL determines the core size tested.
5. Recommended usage of switches 19 and 20
   a. Set both switches.
   b. Start program.
   c. Crank sense amplifier margins negative until failure.
   d. Pattern will advance 1.
   e. Repeat on all 36 bits.
   f. Repeat going positive.

Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	Refer to Table 1.

OPERATIONAL CONTROL

Other than the switches, the user has no control over the operation of this diagnostic.

ERROR MESSAGE SUMMARY

1. Memory Selection Error
   If the memories are incorrectly set up the following error messages may occur.

   a. MEMORY MAP =

FROM	TO	SIZE/K
400000	577777	64
000000	177777	64

   b. MAPPING ERROR; NONCONSECUTIVE MEMORY MODULES?

FROM	TO	SIZE/K	ADR SEQ
00000	377777	128	XXX7
00000	577777	192	XXX6
00000	577777	192	XXX5
00000	577777	192	XXX4

   MAPPING ERROR; MEMORY INTERLEAVING?

This indicates that the memory interleaving is faulty. The first map is done with addresses ending with 7; the second map is done with addresses ending with 6; the third map is done with addresses ending with 5; and the fourth map is done with addresses ending with 4.

Memory mapping for nonexistent memory is done with address multiples of 1000(8). Memory mapping of existent memory is done with address multiples of 100(8).

2. Data Errors
Up to ten data errors are printed. Thereafter the MI is counted for additional errors. In user mode the MI is not counted.

3. Parity Errors
Prints PARITY ERROR, usually followed by data error printout. This is on the assumption that a parity error must necessarily result in a data error, unless the parity bit itself is failing.

4. User Mode Errors
Data errors as above.

The monitor handles parity errors.

In heavy time-sharing the program is often relocated to various areas in memory. Running the program in user mode gives an indication of the state of memory. The core address in the data error printout is a relative address. The actual physical address is also printed.

5. To test that power fail operates correctly, turn off power at any point in the program. Turn power back on, wait for delay timeout, start program at 30005. The program will continue from the point of interrupt if everything is correct. If an interrupt did not occur on power fail, the program will print out: POWER INTERRUPT FAILED.

Table 1 DDMMF Control Switch Summary

Switch	Mnemonic	State	Description
0-17			Standard (Refer to the 10/10 STD module.)
18			Not used
19	HOLD		See below
20	PROCEED		See below
			19 20
			0 0 NO ERROR - Program runs free
			0 0 ERROR - Refill memory current pattern
			0 1 NO ERROR - Program runs free
			0 1 ERROR - Program runs free
			1 0 NO ERROR - Repeatedly tests pattern in core
			1 0 ERROR - Refill memory current pattern
			1 1 NO ERROR - Repeatedly tests pattern presently in core
			1 1 ERROR - Fill memory next sequential pattern, test and hold pattern.
21	LOCKPG	0	Normal operation
		1	Lock program in core (user mode)
22-26			Not used
27-35	KSIZE		Select core size to be tested. (High-order 9 bits, low-order 9 bits are appended as 1s.)
			If 27-35 = 0 use all available core.

GENERAL INFORMATION

Code	DDPCA.A10
Title	DECsystem-10 Paper Tape Reader/Punch Test
Abstract	The PDP-10 reader/punch test is designed to verify the operational status of the paper tape punch and reader. The program provides various timing measurements and random start/stop modes to the reader and punch.
Hardware Required	KA10, KI10 or KL10 mainframe/32K of core memory (minimum)/PC04 or PC09
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	None
Notes	1. This program runs in both exec and user mode. 2. Closed loops of tape may be constructed for all the reader test. In modes 0 and 1, no leader should be present in the loop tape. In modes 2 and 3, the loop should contain leader (zero characters) as this is necessary to initialize the number generator. The tape must be positioned in the leader for modes 2 and 3. 3. Characters read from the tape are buffered in exec mode in a 4-word buffer to prevent false errors when the end of tape is encountered. These last four words are then discarded when the EOT occurs. 4. Exec mode, KA10 KI10 only - The MI is down-counted as a visual indication of a running program.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	Refer to Table 1.

OPERATIONAL CONTROL

Exec Mode

If the program was read in via paper tape, the tape must be removed from the reader before the program is started.

The program tests status bits and the PI system, and allows for adjustment of the punch up to speed delay (flag test).

The program runs continuously until terminated by KEY STOP.

User Mode

Typing START<CR> will cause the program to perform one operation (punch or read) and then exit.

Typing START 30000<CR> will cause the program to run continuously, until terminated by a control °C.

Exec and User Modes

Tests and commands for this program are summarized in Table 2.

The program indicates its readiness to accept commands by printing an asterisk (*).

Type a digit (0-3) to indicate the data mode. Next press the ALTMODE key. The punch will begin punching data. When the desired amount of tape has been punched, press the ALTMODE key. This will terminate punching and respond with an "*" on the terminal.

Remove the tape from the punch and place it in the paper tape reader. Type a digit to select the data mode and a carriage return. The program will read the tape and respond with an "*" when the end of tape is detected.

To select a test to be executed, the following format must be followed:

DIGIT, ALTMODE or CARRIAGE RETURN. For example:

```
*3<ALTMODE> to punch a tape
*3<CR>      to read a tape
```

Table 1 DDPCA Control Switch Summary

Switch	State	Description
0-17		Standard (Refer to the 10/10 STD module.)
		Exceptions - The following switches are not implemented: 1, 2, 9, 12, 14, 15, and 17.
18	0	Normal operation
	1	Inhibit reader flag test
19	0	Normal operation
	1	Inhibit punch flag test
20	0	Normal operation
	1	Loop continuously through flag tests
21	0	Normal operation
	1	Specify block size
		When this switch is set the program will type SPECIFY BLOCK LENGTH (1-63). The operator responds by typing in a decimal number between 1 and 63.
22	0	Normal operation
		When this switch is set the program will type SPECIFY STALL TIME (1-63). The operator responds by typing in a decimal number which represents the stall time in milliseconds, between restarts on reader or punch blocks. If 0 is typed, then random times are used.
23	0	Normal operation
	1	Exit from punch delay test
24	0	Normal operation
	1	Double amount of tape punched (system exerciser/user mode feature)
25-27		Not used
28-35		Data to be punched when data mode 3 is selected

Table 2 DDPCA Control Command Summary

Character	Description
Digits 0-3	Digits 0-3 select the test to be performed. 0 = punch/read a tape of alternate 1s and 0s. The punch-reader is set to the alphanumeric mode (image mode if in a time-sharing system). 1 = punch/read a tape in alphanumeric mode. If punching, the contents of switches 28-35 are punched. If reading, the program checks that all characters are the same as the first non-0 character read. 2 = punch/read a tape of pseudorandom characters in the alphanumeric (image) mode. When reading, the tape must be positioned in the all-0s leader. 3 = punch/read a tape of pseudorandom, binary (image binary) words.
ALTMODE	If the ALTMODE key is preceded by a digit (0-3) the punch will be activated. A second ALTMODE will stop the punch.
<CR>	If the carriage return is preceded by a digit (0-3), the paper tape reader will be activated. If no other character is typed prior to the carriage return, another "*" will be printed.
↑X	Control X - A request for a graph of errors. Stall time is in the vertical axis and block length is in the horizontal axis.
↑Z	Control Z - A request for a timing printout. The printout represents the time required to obtain the next character after a stall has occurred. The first three items represent the longest, shortest and average time to next character for all the stall times selected. The values are most meaningful when random stall times are used. The next printed item is a distribution plot of stall time versus next character time. All times are represented in MS with the low value (-), average (*) and high (+). If the line starts with E, an error occurred at that stall time.
?	Type input format message

ERROR MESSAGE SUMMARY

FAIL TO SYNC

The reader has read 1300(8) characters without any data bits. Excessive leader on data tape or a reader failure.

TAPE IN READER FLAG NOT SET

If there is tape in the reader, press the advance tape switch, and try again. If a second message is typed, there is a logic malfunction.

Data modes 0, 1 do not tolerate trailer. If present, a false error condition will exist when first encountered. Error messages will be printed throughout the length of trailer.

Data errors have the following message in modes 0, 1, 2:

```
ALPHA ERR XXX
      COR XXX
```

This is the data read and the data expected. If an error occurs in alpha mode the character being checked is four tape frames past the tape read head.

Data errors have the following message in mode 3.

```
BINARY ERR XXXXXXXXXXX
       COR XXXXXXXXXXX
```

This represents the 36 bits read and the 36 expected. If an error occurs in binary mode the word being checked starts 24 frames (4 words, 6 frames/word) past the read head.

Exec Mode Only Errors

Status errors are reported in the following format

PC = XXXXXX
E = XXXXXX
AC = XXXXXXXXXXXX

Where PC is the location of the error, C(AC) is the status at the time of error, and E is an error number.

TIME IN MS XXX.X

This message is printed only if punch startup time is less than .9 second. Prints repeatedly, reflecting change in time as delay is adjusted. Data switch 23 or typing an altmode provides exit.

PUNCH POWER OFF OR OUT OF TAPE-CORRECT AND CONTINUE

Status bit 29 (OT) is set. Turn on power or install tape and continue

.5 SECONDS HAVE ELAPSED.
DONE FLAG IS NOT SET -
READER IS HUNG.

The PTR DONE flag failed to set following a DATAI. The reader is hung.

User Mode Only

PUNCH NOT AVAILABLE

The paper tape punch is assigned to another job.

READER NOT AVAILABLE

The paper tape reader is assigned to another job.

I/O TO UNASSIGNED CHANNEL

Some input/output statement was executed prior to an INIT for that device. An instruction or monitor failure.

ADDRESS CHECK FOR DEVICE X

Not enough core available for I/O buffers. An instruction or monitor failure.

GENERAL INFORMATION

Code	DDQCB.A10
Title	DECsystem Memory Port Interaction/System Exercisor
Abstract	The port interaction test provides a way of checking for multiport interference problems. The program runs in exec mode only and loads with exec DDT to provide operator/program communication.
Hardware Required	KA10, KI10, or KL10 mainframe/48K of core (minimum)/some combination of the following subsystems: CPB - a second CPU DF10/DF10C data channel DL10 with 1, 2, 3, or 4 PDP-11s (maximum of two PDP-11s run at a time) DTE20 master and up to 3 other restricted DTE20s DX10 controller with TU70 RC10 controller with RD10 disk or RM10B drum RH10 controller with an RS04 or RP04/05/06 disk RH20 controller with an RP04/05/06 disk RP10 controller with RP02 or RP03 drives TM10B controller with TU10 or TU20 or TU30 or TU40 drives
Preliminary and Associated Programs	All processor and memory diagnostics, all diagnostics for those devices to be tested. (Refer to the 10/10 STD module.)
Restrictions	If a second CPU is to be tested (CPB) it must be of the same type as the first (i.e., both KA10s, both KI10s or both KL10s)
Notes	1. Clocks - The program uses the CPU clock (for KA10, KI10), and the meter for KL10 to time all device transfers. 2. Data Transfers - The program will utilize the maintenance cylinder on disk pack and the system swapping space (if it exists) on the disk file or drum. 3. CPB Testing - If the user types CPB, indicating that he wants to cycle the second CPU, that CPU must have previously been started at location 30004. Failing to do the above will result in an error message. This message will report the initialization failure and give the user the option of starting or forgetting CPB. If CPB gets excessive CPU errors it will halt. Normally, it will report these errors through CPA and try to continue to run. If CPB halts, it can be reinitialized by starting at 30004, assuming that CPB's code has not changed.

4. Overruns

   a. If system is configured to have an odd number of mass storage devices (including drums) overruns may occur. This is due to drum termination at some place other than a sector gap. Symptoms of this kind of failure are:

      (1) Overrun error with no data error

      (2) Normal termination (correct control word stored)

   b. Random cycling may cause overrun errors on drums even if an even number of devices is being used.

   c. Switch 25 may be used. If, for example, all devices go through a MX10, this case may cause many overruns. With switch 25 set, only one device will run at a time. The program waits for the device to finish and then starts the next device.

   d. Use switch 29 if running with fast memory disabled (KA10) or address stop down.

   e. Use switch 29 if system has slow memories and a drum. This will eliminate overrun errors due to speed difference.

Loading and Starting Procedures: Standard (Refer to the 10/10 STD module.)

Control Switches: Refer to Table 1.

OPERATING PROCEDURE

Upon starting, DDQCB will report the size of memory and request the names of the devices to be tested. The legal mnemonic device names and their associated PI channels are listed in Table 2.

Table 3 describes DDQCB control commands.

Table 4 describes special DDQCB/DDT commands.

Starting Addresses

Program Starting Address - 30000

CPB Starting Address - 30004

Restart Address - 30006

Executive DDT Starting Address - 30007

TEST SUMMARY

DDQCB does the following:

1. Asks for and initialize the devices to be used. During device initialization, memory is mapped through the designated channel unit(s). The mapping is done by a series of device reads into memory, checking for nonexistent memory on termination.

2. Loads a pattern through one of the 16K modules. The program uses the following data patterns in the device data buffers. The patterns are used on a cyclic basis.

   a. all 1s
   b. all 0s
   c. 4-bit floating 1
   d. 4-bit floating 0
   e. 4-bit count
   f. address pattern (core memory address)
   g. DF10 W102 worst-case pattern
   h. DF10 parity logic worst-case pattern
   i. MX10 W102 worst-case pattern

NOTE
With switch 27 = 1, only patterns C, D, E, F are used. With switch 27 = 0 all of the patterns are used.

3. The memory module is segmented by the number of devices the test is going to use (i.e., two devices would each have an 8K data buffer). The DF10 control words are set up to split the memory module into as many segments as are needed to ensure that no device transfers data from another's buffer.

4. Start all of the devices writing the data pattern set up in step 2. The CPU will either block transfer all of core to itself or a MOVES instruction to move all of core to itself. The latter utilizes the read-modify-write type of memory cycle.

5. Detect and report any CPU or device errors.

6. Start all devices reading into the assigned data buffers.

7. Detect and report any CPU or device errors. Also report any data-compare errors that may result.

8. Swap the DF10 control words to cause each device to transfer into a different data buffer unless mixed mode DF10s are to be used (DF10s and DF10Cs).

9. Continue this write/read action until each device has transferred all of memory to and from its storage medium.

10. Increment to the next memory module when all data patterns have been utilized, and repeat steps 4 through 9.

Table 1 DDQCB Control Switch Summary

Switch	Mnemonic	State	Description
0-17			Standard (Refer to the 10/10 STD module.) Exception - The following switches are not implemented: 1, 5, 6, 7, and 9.
14	OPRSEL	1	If switch 14 (operator select) is set, the program asks which memory bank to run in.
18	EXMD	0	Normal operations
		1	Enter exerciser mode. Massbus magtape (TU16/45) will issue a read reverse instead of a backspace then read sequence. RP04/05/06-type drives will use other than the maintenance cylinders for data transfers. Assigned swapping space will be used on known pack structures and any cylinder will be used on undefined packs or packs with no swapping space assigned.
19	WRLP	0	Normal operation
		1	Loop on write errors
20	INRS	0	Inhibit restart of slow devices. Fast devices will be restarted until all slow devices are done. Slow devices are the DTE20, DL10 and TM10B. This keeps the devices very busy but the data is not checked until all devices are done.
		1	Do not inhibit restart.
21	DELDEV	0	Delete a device from device test table after 16 errors.
		1	Do not delete a device from device test table after 16 errors.
22	ABTMAG	0	Allow 5 magtape errors before aborting.

Table 1 DDQCB Control Switch Summary (Cont)

Switch	Mnemonic	State	Description
		1	Allow 200 (decimal) errors before aborting.
23	INHMOV	0	Normal operation
		1	Inhibit MOVES on memory
			(read-pause-write)
24	INHBLT	0	Normal operation
		1	Inhibit BLT self to self
25	SIGDEV	0	Normal operation
		1	Prevent all devices from running simultaneously. This can be used where many devices go through a MX10 which may cause overruns. Only one device will run at a time.
26	RANCYC	0	Normal operation
		1	Random cycle both memory and data patterns
27	DATPAT	0	Use all of the data patterns
		1	Use only the 4-bit data patterns
28	DEVMAP	0	Normal operation
		1	Loop on device map
29	CLRCOR	0	Normal operation
		1	Inhibit clearing core between read and write
30	INHNXD	0	Normal operation
		1	Inhibit drive IOB nonexistent device
31	INHCWW	0	Normal operation
		1	Inhibit requesting control word writes
32	DEVPAR	0	Normal operation
		1	Output all device status on memory parity error
33	SELDAT	0	Normal operation
		1	Select data patterns
34	CPUCYC	0	Normal operation
		1	Inhibit cycling CPU (applies to CPB if its SW34 is equal to 1)
35	DATAWT	0	Normal operation
		1	Inhibit check for data compare errors

Table 2 Device Mnemonics and Channel Assignments

Device	Address	Mnemonic	Channel
DL10	060, 064	DL0	6
DL10	160, 164	DL1	6
DTE20	200	FE0	1
DTE20	204	FE1	1
DTE20	210	FE2	1
DTE20	214	FE3	1
DX10	220	DXA	2
DX10	224	DXB	2
DX10	034	DXC	2
RC10	170	DSK	4
RC10	174	FHB	4
RH10	270	RHA	5
RH10	274	RHB	5
RH10	360	RHC	5
RH10	364	RHD	5
RH10	370	RHE	5
RH10	374	RHF	5
RH20	540	RH0	5
RH20	544	RH1	5
RH20	550	RH2	5
RH20	554	RH3	5
RH20	560	RH4	5
RH20	564	RH5	5
RH20	570	RH6	5
RH20	574	RH7	5
RP10	250	DPA	3
RP10	254	DPB	3
TM10B	340, 344	MTA	7
TM10B	350, 354	MTB	7

The second CPU is designated CPB.

Table 3 DDQCB Control Commands

Command	Description
$	ALTMODE - Exit DDQCB enter DDT. Refer to Table 4.
↑C	Control C - Exit DDQCB enter DDT. Refer to Table 4.
↑Z	Control Z - End device input stream and begin device testing.

Table 4 Special DDQCB/DDT Commands

Command	Description
$G	Restart program from the beginning.
$I	Start DIAMON.
R$G	Start program over using current devices and clear run time.
S$G	Start program over using current devices and reassign drive numbers.
C$G	Continue program from point of interruption.

ERROR MESSAGE SUMMARY
The errors that are detected are in one of three classes.

1. Basic initialization errors - these errors occur while the program is trying to initialize the device(s) it is using. Examples could be:

   a. Controller powered down
   b. Selected unit is not ready.

2. Operational errors - these errors occur while the program is running and usually contain more information than the type 1 errors. Examples are:

   a. Device parity errors
   b. Device hung errors.

3. CPU errors - examples of these types of errors are:

   a. Memory parity
   b. Nonexistent memory
   c. Illegal UUO.

GENERAL INFORMATION

Code	DDRHA.A10
Title	RH10 Deviceless Diagnostics
Abstract	The DDRHA diagnostic is a RH10 deviceless diagnostic for the RH10 system. This is the first and basic diagnostic for the RH10. This diagnostic provides the majority of the testing for the RH10 without an actual Massbus device being connected to the RH10. By utilizing the maintenance mode feature of the RH10, read and write transfers can be simulated and the CPU memory to DF10/DF10C/DAS33 to RH10 controller data path is throughly tested without using the Massbus. This program is an exec mode-only diagnostic.
Hardware Required	KA10, KI10 or KL10 mainframe/32K of core (minimum)/DF10, DF10C, DAS33 (18 or 22 bit)/RH10
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	Control Switch 20=0 and 21=1 is an illegal combination
Notes	1. The data channel WRITE EVEN PARITY ENABLE switch should be set to the ON or ENABLE position. 2. The data channel ADDRESS MODE switch may be set to either the 18-bit or 22-bit position. 3. For DF10s the STOP ON PARITY ERROR switch must be reset (i.e., do not stop on parity errors). 4. Individual RH10s are tested by changing the device code. Control switch 12. 5. When testing a DAS33 the STATUS WRITE ENABLE and WRITE CONTROL WORD REG switches on the local control panel must be set as follows. 20 21 SWE WCR 1 0 OFF OFF 1 1 ON ON 6. Read test 21 description
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	Refer to Table 1.

OPERATIONAL CONTROL
DDRHA is controlled via the control switches. All switches should be reset for normal operation.

DDRHA TEST SUMMARY
The individual tests performed by this diagnostic are summarized in Table 2.

ERROR MESSAGE SUMMARY
The following is an example of an error message

```
PC = 032474
SWITCHES = 000000 140000
ERROR IN DF10/DAS33 TERMINATE - TERMINATION ERROR
CORRECT:  000761 030602      (INAD + 1)
ACTUAL:   000000 000000
DISCREP:  000761 030602
STATUS INCORRECT
```

The listing (on microfiche) must be consulted at the PC address to determine the subtest which failed and the meaning of the CORRECT and ACTUAL data.

Table 1 DDRHA Control Switch Summary

Switch	State	Description
0-15		Standard (Refer to the 10/10 STD module.)
16	0	60 Hz power (KA10)
	1	50 Hz power (KA10)
17		Not used
18	0	Normal operation
	1	Use switches 30-35 for test selection
20	0	Test DF10/DF10C
	1	Test DA33
21	0	DF10/DF10C type status
	1	DA33 extended status writeback
22	0	Long printout for DAS33 extended status
	1	Short printout for DAS33 extended status
23-29		Not used
30-35		Select individual test to be executed. Switch 18 must be set (1)

Table 2 DDRHA Test Summary

Test	Description
Test 0 Subtests	RH10 BASIC STATUS TESTS
RH1	Response Test Try to set all RH10 PIA bits, then get the RH10 status and determine if any PIA bits or any other status from the RH10 can be read.
RH2	PI Select Set all RH10 PIA bits (PIA 33-35), then read back the RH10 status and verify that all PIA bits were set in the RH10.
RH3	PI Select Clear all RH10 PIA bits, then read the RH10 status back to verify that no PIA bits were set in the RH10.
RH4	Device Code Response Clear the system (APR and PI network), then set all RH10 PIA bits. Read the status from all other devices to ensure that the RH10 device code (270 or 274) causes no other device on the system to respond.
RH5	Device Code Response Clear the system and the RH10 PIA bits. Set all PIA bits on all other devices and check to ensure that the RH10 does not respond to any other device code.
RH6	PIA Bit Test Check all RH10 PIA bits individually to ensure that they all operate properly.
RH7	PS Fail Check the RH10 status for no indication of a power supply failure.
RH8	CONI Reset Test Set the RH10 PIA bits, then execute a system clear (APR and PI) to ensure that all RH10 PIA bits clear.
RH9	BUSY Clear Test Clear the system and check the RH10 status bits to ensure that the BUSY flag is not set.
RH10	DONE Set Send the stop bit to the RH10. Check for DONE to be set.
RH11	DONE Clear Send the stop bit to the RH10 (clear BUSY and set DONE then clear the DONE flag with the CONO DONE clear bit.

Table 2 DDRHA Test Summary (cont)

Test	Description
RH12	DONE Clear Send the stop bit to the RH10 (clear BUSY and set DONE) and execute a system clear. Check for DONE to be clear.
RH13	DONE Clear Clear the system then execute a CONO RH,0. This is done to verify that the DONE bit will stay clear.
RH14	DONE Set Send the RH10 stop bit (clear BUSY and set DONE) and execute a CONO RH, 0. Verify that the DONE bit remains set.
RH15	Status Test RH10 CONI status test. Clear RH10 and test for correct status.
RH16	RH10 CR Register Clear RH10 control register, then test - should be clear.
RH17	RH10 INT ADDR Register Clear RH10 interrupt address register, then test for clear status.
RH18	RH10 Data Buffer Register Clear RHDB register and check status.
RH19	RH10 RAE Register Clear RH10 register access error register, then test status - should be clear.
RH20	RH10 Channel Buffer Clear RH10 channel buffer register and test its status - should be clear.
Test 1 Subtests	RH10 INTERRUPT ADDRESS REGISTER TEST
RHIAT	RH10 INT ADDR REG Clear Send all 0s to the RHIA register. The only bit set should be the KA10 interrupt mode bit.
RHIA1	RH10 INT ADDR REG Set Send all interrupt address bits (777) to the RHIA register. KA10 and all address bits should be set.
RHIA2	RH10 INT ADDR REG Bit Test Send floating 1s through the RHIA register to check the bit interaction.
RHIA3	RH10 INT ADDR REG Bit Test Send a floating 0s pattern through the RHIA register to check bit interaction.
RHIA4	RH10 INT ADDR REG KIINT Bit Test Send the RH10 KI10 and KL10 interrupt mode control bit to the RH10, then read the register and check the bit.
RHIA5	RH10 INT ADDR REG KIINT Bit Test Send the KI10 and KL10 INT bit and all RHIA address bits - should see KI10 and KL10 bit and all bits when read.
RHIA6	RH10 INT ADDR REG Clear Set all address bits and KIND INT bit, then clear them with RHCLR. RHIA should indicate KAINT and vector address = 777.
RHIA7	RH10 INT ADDR REG Clear Set all address vector bits and KIINT bit, then clear RHIA with IOCLR. RHIA should indicate KAINT and vector address = 777.

Table 2 DDRHA Test Summary (Cont)

Test	Description
Test 2 Subtests	**RH10 PI SYSTEM TESTS**
RHPI0	Initialize Interrupt Vectors Set up the interrupt vectors for the following tests. Jump to RHPI0A if CPU = KA10. Save locations 760 to 777 with a BLT instruction. Set up locations 760 to 777 with KIMUO MUUOs.
RHPI0A	Set up the normal interrupt locations (42-56).
RHPI1	No Interrupts Clear the system (APR and PI network), then turn on the RH10 interrupt system. No interrupt should occur.
RHPI2	No Interrupts Clear the system (APR and PI) and enable the RH10 interrupt system. Turn on the PI system and check for no interrupts.
RHPI3	No Interrupts Clear the system (APR and PI) then turn on the PI system. Send the RH10 stop bit (clear BUSY and set DONE) to the RH10 with no PIA bits set - check for no interrupts.
RHPI4	Done Interrupt Clear the system (APR and PI) then turn on the PI system. Send the RH10 stop bit (clear BUSY and set DONE interrupt.
RHPI5	Done Interrupt Verify that the RH10 DONE interrupt can interrupt on all selected PIA channels.
RHPI6	SETUP KI10 and KL10 Vectors Jump to RHPIX (exit location) if CPU = KA10 RH10. RH10 interrupt vector test for KI10 and KL10 mode interrupts.
RHPI7	KI10, KL10 Vectored Interrupts The program causes KI10 or KL10 mode interrupts by sending a CONO RH, STOP!7 to the RH10 (clear BUSY and set DONE Interrupt). Interrupts will be vectored to memory locations 760 to 777. KIMUO is the MUUO handler for KI10 or KL10 mode interrupts. This routine determines if the interrupt was vectored to the correct memory address and will return to RHPI7B if an error was detected - or it will return to RHPI7B + 1 if no error was detected.
RHPI8	Restore the locations 760 to 777
RHPI9	Restore the location 1000 to 1007
Test 3 Subtests	RH10 DATA BUFFER REGISTER TESTS
RHDBT	RH10 Data Buffer Clear Tests on RH10 data buffer.
RDBT1	RHDB Set Set all data bits in the RH10 data buffer.
RDBT2	RHDB Set RHCLR should not affect the data buffer.
RDBT3	RHDB Clear Clear RH10 data buffer with a DATAO to the register with DATA = 0.
RDBT4	RHDB Alternate Data Bits Send an alternate data bit pattern (252525) to RH10 data buffer register.

Table 2 DDRHA Test Summary (Cont)

Test	Description
RDBT5	RHDB Complement Alternate Data Bits Send a complement alternate data bit pattern (525252) to RH10 data buffer register.
RDBT6	RHDB Floating 1s Data Bits Send a floating 1s data bit pattern to the RH10 data buffer.
RDBT7	RHDB Floating 0s Data Bits Send a floating 0s data bit pattern to the RH10 data buffer.
Test 4 Subtests	**RH10 CONTROL REGISTER TESTS**
RHCRT	RH10 Control Register Set + Clear With IOCLR Send all writable RHCB bits except GO bit, then check status to verify all bits can set.
RHCR1	RH10 Control Register Set + Clear With IOCLR Send all bits to RHCB and clear with IOCLR.
RHCR2	RH10 Control Register Set and Clear Set all RHCR bits, then clear the register with a RHCR DATAO with no data.
RHCR3	RH10 Control Register Alternate Bits Send alternate bits to RH10 control register and check results.
RHCR4	RH10 Control Register Complement Alternate Bits Send complement alternate bits to RH10 control register and check results.
Test 5 Subtests	**DF10/DAS33 BASIC JUMP-TERMINATE TESTS**
DF10T	DF10/DAS33 Tests - Basic JUMP Load a DF10/DAS33 command list with a JUMP to a TERMINATE, then test. Do this with all initial addresses from 760-776.
Test 6 Subtests	**DF10 TERMINATE TEST**
DF10T2	RH10/DF10 Diagnostic Mode - Write 1 Word Word should load into RH10 AR register, channel should terminate, and RH10 CB register should be clear. RH10 status should indicate AR FULL and BUSY.
Test 7 Subtests	**RH10 CHANNEL BUFFER MAINT MODE**
RHCBT	RH10 Channel Buffer Maintenance Mode Test - Write Set up DF10/DAS33 command list to write from memory two words to the RH10. RHCB should contain the second data word and the DF10/DAS33 should shut down and write a control word. The test data will be as follows. 2 words all 0s 2 words all 1s 1 word all 1s, 1 word all 0s 1 word all 0s, 1 word alternate bits 1 word all 0s, 1 word complement alternate bits Set up DF10/DAS33 for 2-word transfer Clear memory buffer
RHCB1	RHCB MAINT Mode - Write 2 Words of All 0s Start a write transfer to the RH10 in maintenance mode. Should see second word in RHCB and DF10/DAS33 should terminate.
RHCB2	RHCB MAINT Mode - Write 2 Words of All 1s Start a write transfer to the RH10 in maintenance mode. RHCB should = second word.

Table 2 DDRHA Test Summary (Cont)

Test	Description
RHCB3	RHCB MAINT Mode - Write 1 Word All 1s and 1 Word All 0s Transfer two words to RH10 and verify that all bits in RHCB clear after being set.
RHCB4	RHCB MAINT Mode - Write 1 Word All 0s 1 Word ALT Bits Transfer two words to RH10. Second data word is alternate bits. RHCB should = second data word.
RHCB5	RHCB MAINT Mode - Write 1 Word All 0s, 1 Word Complement Alternate Bits Transfer two words to RH10. Second data word is complement alternate bits. RHCB should = second data word.
Test 10 Subtests	RH10 DRIVE INTERFACE BUFFER REG - MAINTENANCE MODE
RHDIBT	RH10 DIB MAINT Mode - Set and Clear Set all writable bits in DIB register and check that they set. Clear RH10 with RHCLR and test DIB register for proper bits to clear.
RHDIB1	RH10 DIB REG MAINT Mode - Send 0s Send all 0s to DIB register and check DIB register for clear status.
RHDIB2	RH10 DIB REG MAINT Mode - Send ALT Bits
RHDIB3	RH10 DIB REG MAINT Mode - Send COMP ALT Bits
RHDIB4	RH10 DIB REG MAINT Mode - CBOV Send two DATAOs to DIB register to cause control bus overrun - CBOV.
RHDIB5	RH10 DIB REG MAINT Mode - CBOV Interrupt Enable interrupt system to verify CBOV will cause an interrupt.
RHDIB6	RH10 DIB REG MAINT Mode - CBOV Cause CBOV and clear the error with RHCLR.
DIB0	RH10 DIB REG MAINT Mode - RAE Tests Drive 0 Cause DIB register data late error (DLT) which causes register access error (RAE) in RHRA register for selected drive. RAE should not appear in RH10 status until RAE is enabled. When RAE is enabled, RAE should cause RH10 RAE interrupts. Cause drive 0 RAE error - RH10 status should be clear. Enable RAE interrupt - verify RAE appears in RH10 status. Enable interrupt system - RAE should cause interrupt.
DIB1	Same as subtest DIB0 except tests drive 1.
DIB2	Same as subtest DIB0 except tests drive 2.
DIB3	Same as subtest DIB0 except tests drive 3.
DIB4	Same as subtest DIB0 except tests drive 4.
DIB5	Same as subtest DIB0 except tests drive 5.
DIB6	Same as subtest DIB0 except tests drive 6.
DIB7	Same as subtest DIB0 except tests drive 7.
Test 11 Subtests	RH10 DIAGNOSTIC MODE TESTS - WRITE
RHMW	RH10 Diagnostic Mode - Write Seven Words Set up DF10/DAS33/RH10 to write seven words (36 bits) of data in RH10 diagnostic mode, single-stepped. Check data path from memory to DF10/DAS33 to RH10 channel buffer to RH10 data buffer.

Table 2 DDRHA Test Summary (Cont)

Test	Description
	Clear memory data buffer and set up with these 7 words: 1. One 36-bit word of all 0s 2. One 36-bit word of all 1s 3. One 36-bit word of all 0s 4. One 36-bit word of alternate bits 5. One 36-bit word of complement alternate bits 6. One 36-bit word of (123456,,654321) pattern 7. One 36-bit word of (654321,,123456) pattern Set up DF10/DAS33 command list to transfer 7 words. Start RH10 in maintenance mode - write.
RHMW2	Check RH10 status - first word (all 0s) should be in assembly register and second word (all 1s) should be in the channel buffer. Write the DF10/DAS33 control word and check.
PHMW3	Supply first RH10 maintenance mode sync clock and check the first data word in RHDB, then write and check the DF10/DAS33 control word. This also causes the third word to transfer to the RHCB register (third word = all 0s).
RHMW4	Supply one RH10 maintenance mode sync clock and check RHDB and RHCB. RHDB should contain left half of second data word. RHCB should still contain third data word.
RHMW5	Supply one RH10 maintenance mode sync clock to cause fourth word transfer. RHDB should contain right half of second data word. RHCB should contain fourth data word.
RHMW6	Supply one RH10 maintenance mode sync clock and check RHDB and RHCB. RHDB should contain left half of third data word. RHCB should still contain fourth data word.
RHMW7	Supply one RH10 maintenance mode sync clock to cause fifth word transfer. RHDB should contain right half of third data word. RHCB should contain fifth data word.
RHMW8	Supply one RH10 maintenance mode sync clock and check RHDB and RHCB. RHDB should contain left half of fourth data word. RHCB should still contain fifth data word.
RHMW9	Supply one RH10 maintenance mode sync clock to cause sixth word transfer. RHDB should contain right half of fourth data word. RHCB should contain sixth data word (different bytes).
RHMW10	Supply one RH10 maintenance mode sync clock to get left half of fifth data word. RHDB should contain left side of fifth word complement alternate bits). RHCB should still contain sixth data word (difference bytes).
RHMW11	Supply one RH10 maintenance mode sync clock to cause the seventh word transfer. RHDB should contain right side of fifth data word (complement alternate bit). RHCB should contain last word - seventh data word (complement different bytes). The DF10/DAS33 should write the termination control word in CRICW + 1.
RHMW12	Supply one RH10 maintenance mode sync clock to cause the left half of the sixth data word to transfer to RHDB (difference bytes). RHCB should still contain the seventh data word (complement difference bytes).
RHMW13	Supply one RH10 maintenance mode sync clock to cause the right half of the sixth data word to transfer RHDB (difference bytes). RHCB should go to 0.
RHMW14	Supply one RH10 maintenance mode sync clock to cause left half of seventh data word (complement different bytes) to transfer to the RHDB. RHCB should remain 0.
RHMW15	Supply one RH10 maintenance mode sync clock to cause right half of seventh data word (complement difference bytes) to transfer to the RHDB.

Table 2 DDRHA Test Summary (Cont)

Test	Description
RHMW16	Supply one RH10 maintenance mode sync clock to cause left half of first 0 fill word to transfer to the RHDB.
RHMW17	Supply one RH10 maintenance mode sync clock to cause right half of first 0 fill word to transfer to the RHDB. Also, supply RH10 maintenance mode END-OF-BLOCK to cause transfer complete.
Test 12 Subtests	RH10 DIAGNOSTIC MODE READ TEST
RHMR	RH10/DF10 Diagnostic Mode-Read 7 Words Set up DF10/DAS33/RH10 to read seven words (36 bits) of data in RH10 diagnostic mode-single-stepped. Check data path from the RH10 data buffer to RH10 channel buffer through the DF10/DAS33 into memory. The data words to be read are as follows. 1. one word all 0s 2. one word all 1s 3. one word all 0s 4. one word alternate bits (525252,,525252) 5. one word (12345,,654321) 6. one word (123456,,654321) 7. one word (654321,,123456) Set memory data buffer to unused pattern. Set up DF10/DAS33 command list to transfer seven words. Start RH10 in maintenance mode-read. Check RH10 status: CHNACT, CHNPLS and BUSY should be on.
RHMR2	Supply one RH10 maintenance mode sync clock with left half of first data word (all 0s) to RHDB. RHCB should remain clear and RH10 status should not indicate ARFULL, or CBFULL and should indicate BUSY.
RHMR3	Write a DF10/DAS33 control word to verify that the DF10/DAS33 word count and address have not changed.
RHMR4	Supply one RH10 maintenance mode sync clock to transfer right half of first data word (all 0s) to RH10 data buffer, RH10 channel buffer, and to memory through the DF10/DAS33. RHCB should contain full first data word after the transfer.
RHMR5	Supply one RH10 maintenance mode sync clock to transfer left half of second data word (all 1s) to RHDB. RHCB should still contain first data word.
RHMR6	Supply one RH10 maintenance mode sync clock to transfer right half of second data word (all 1s) to RHDB. The RH10 should transfer complete second data word through the RH10 assembly register to RH10 channel buffer to DF10/DAS33 and into memory. RHCB should contain the second data word after transfer is complete.
RHMR7	Supply one RH10 maintenance mode sync clock to transfer left half of third data word (all 0s) to RHDB. RHCB should still contain second data word.
RHMR8	Supply one RH10 maintenance mode sync clock to transfer right half of third data word (all 0s) to RHDB. The RH10 should transfer the complete third data word through the RH10 assembly register to the RH10 channel buffer to the DF10/DAS33 and to memory. RHCB should contain third data word (all 0s) after the transfer is complete.
RHMR9	Supply one RH10 maintenance mode sync clock to transfer the left half of the fourth data word (alternate bits) to RHDB. RHCB should contain third data word.
RHMR10	Supply one RH10 maintenance clock to cause right half of fourth data word (alternate bits) to transfer through RH10 to memory. RHCB should contain fourth data word.

Table 2 DDRHA Test Summary (Cont)

Test	Description
RHMR11	Supply two RH10 maintenance clocks to send the fifth data word (complement alternate bits) to memory. Check associated functions.
RHMR12	Transfer sixth data word through RH10 to memory. This data has different left-right side bytes to check proper byte assembly and transfer.
RHMR13	Transfer seventh data word through RH10 to memory. This data has different left-right side bytes to check proper byte assembly and transfer.
RHMR14	This should have completed transfer. DF10/DAS33 should have terminated and written termination control word. RH10 status should indicate BUSY.
RHMR15	Try to transfer one more word through RH10 into memory. DF10/DAS33 should not allow transfer as the DF10/DAS33 command list was for only seven words and the DF10/DAS33 has terminated.
RHMR16	Supply RH10 maintenance END-OF-BLOCK. This should complete transfer, clear BUSY and set DONE.
Test 13 Subtests	**RH10 MAINTENANCE MODE WRITE PARITY TESTS**
RHMWP	RH10 Maintenance Mode Parity Write Set up DF10/DAS33 command list to transfer two words of data from memory through the DF10/DAS33 and to the RH10 RHDB register with wrong parity selected (even parity). No parity error should be generated. DBODD bit in the RHBD register should operate opposite from normal. Start data transfer in maintenance mode-write two words. Supply a maintenance clock to cause right half of first data word (all 1s) to transfer to RHDB register. DBODD should be 0 and no error should be generated. Supply a maintenance clock to cause left half of second data word (1) to transfer to the RHDB register. DBODD should be 0 (odd parity). No parity error should be generated. Supply a maintenance clock to cause right half of second (and last) word (all 1s) to RHDB register. DBODD should be 0 and no error generated.
Test 14 Subtests	**RH10 MAINTENANCE MODE READ PARITY TESTS**
RHMRP	RH10 Maintenance Mode Parity Read Set up DF10/DAS33 command list to transfer one word of data (36 bits) from the RH10 RHDB register to memory with wrong parity selected (even parity). No parity error should be generated. Set up DF10/DAS33 list and clear memory buffer. Start data transfer in maintenance mode. Read one word. Supply one maintenance clock and left half data of first data word to be transferred to memory. No parity error should be generated. Supply one maintenance clock and right half data of first data word to be transferred to memory. This clock causes the word to be written into RHCB and memory and no parity error should be generated.

Table 2 DDRHA Test Summary (Cont)

Test	Description
RHMRP1	RH10 MAINT Mode DBPE - Read Set up to transfer one word to RHDB with wrong parity supplied. Should cause data bus parity error (DBPE). DXES (disable transfer error stop) = 0 for this test to enable transfer shutdown. CXR DBPE STOP should freeze the word causing DBPE in the RH10 data buffer register. A second word should be inhibited from modifing the RHDB. No memory transfer will occur and the DF10/DAS33 control word should be written. END-OF-BLOCK should cause transfer complete. Set up DF10/DAS33 command list for 2-word transfer. Clear memory buffer. Start RH10 in maintenance mode. Supply one maintenance mode sync clock with left half of first data word (all 1s) with incorrect parity supplied to the RHDB register. RH10 status should indicate CHNPLS, DBPE and BUSY. DF10 control word should have been written due to channel termination. Supply one maintenance sync clock with right half of first data word (all 0s) with correct parity supplied to the RHDB register. Supply two maintenance mode sync clocks to clock second data word (alternate bits). Supply END-OF-BLOCK to terminate transfer, then check RH10 status, DF10/DAS33 control word, memory, and RHDB for no modification.
RHMRP2	RH10 MAINT Mode DBPE - Read Set up to transfer one word with wrong parity supplied to RHDB and to memory. Should cause DBPE error but not shut down the transfer because DXES (disable transfer error stop) now equals 1. Two words should be transferred to memory and DBPE should appear after the first word transferred. Normal transfer termination (END-OF-BLOCK in maintenance mode) should cause transfer complete. Set up DF10/DAS33 command list for a 2-word transfer. Clear memory buffer. Start RH10 in maintenance mode - read. Supply one maintenance mode sync clock with left half of first data word (all 1s) with incorrect parity supplied to the RHDB register. RH10 status should indicate DBPE and CHNACT with BUSY. Check for no DF10/DAS33 control word written. Supply one maintenance mode clock with right half of first data word (all 0s) and correct parity supplied. First word should transfer to memory. Supply two maintenance clocks to transfer the second word (alternate bits) to memory. Supply END-OF-BLOCK to cause transfer completion. Check RH10 status, DF10 control word and memory for proper transfer completion.

Table 2 DDRHA Test Summary (Cont)

Test	Description
Test 15 Subtests	RH10 SELECTED DRIVE REGISTER ACCESS ERROR TEST
SDRAE0	RH10 SD RAE Tests Drive 0 Cause DIB DLT and RAE in RHRA register. Start a data transfer in maintenance mode using drive 0 to cause SD RAE and ILC error in the RH10 status register. The DF10/DAS33 should shut down before any data has been transferred and the RHCB register should remain cleared. Cause DIB data late for drive 0. Start a data transfer in maintenance mode to cause SD RAE. DF10/DAS33 is set for a 2-word transfer. Check the RH10 status for SD RAE and ILC and DONE. Check the RHCB - should be empty.
SDREA1	Same as subtest SDREA0 except tests drive 1. Same as subtest SDREA0 except tests drive 2. Same as subtest SDREA0 except tests drive 3. Same as subtest SDREA0 except tests drive 4. Same as subtest SDREA0 except tests drive 5. Same as subtest SDREA0 except tests drive 6. Same as subtest SDREA0 except tests drive 7.
Test 16 Subtest	RH10 MAINTENANCE MODE CONO STOP
RHMMS	RH10 Maintenance Mode Transfer Stop Set up DF10/DAS33 command list to transfer two data words to memory in maintenance mode. Single-step one data word to memory, then send a STOP command to the RH10 and check for proper shutdown. Set up DF10/DAS33 for a 2-word transfer. Clear memory buffer. Start a maintenance mode read transfer. Supply first data word to RH10 and transfer to memory buffer. Issue a CONO RH, STOP to terminate the transfer. Check DF10/DAS33 control word, memory buffer, and RH10 status for proper shutdown.
Test 17 Subtest	RH10 OVERRUN
RHOV	RH10 Overrun Test - Maintenance Mode Set up the DF10/DAS33 command list so that the DF10/DAS33, when started, will enter and remain in a JUMP loop after fetching one word from memory. Try to transfer data to memory (in maintenance mode) - this should cause an overrun condition which should cause DONE. Create a DF10/DAS33 command list to transfer one data word, then enter a JUMP loop to hang the channel. Start a maintenance mode write transfer. After the second and third data word and maintenance mode clocks are sent, there should be an overrun condition. OVERRUN and DONE should appear in the status, the DF10/DAS33 should terminate and write a control word, and memory should not be modified.

Table 2 DDRHA Test Summary (Cont)

Test	Description
Test 20 Subtest	MEMORY CONNECTION TEST FOR DF10/DAS33
RHCON	Check that the DF10/DAS33 can transfer 8K words from each memory bank that the CPU is connected to. A bank is considered to be 8K words. This takes into account that the MA10 has upper and lower 8K deselection switches and they may be set to deselect.
Test 21 Subtest	DF10/DAS33 NONEXISTENT MEMORY TEST
RHNXM	Check that the DF10/DAS33 is not connected to more memory than the CPU. DF10/DAS33 is started with a control word pointing to nonexistent memory, the nonexistent memory flag should appear in the controller. If it is not, then the DF10/DAS33 is connected to a bank which the processor is not connected to. The increment is by 8K for each test because of MA10 logic which allows deselect of either upper or lower 8K of the stack. This test will not be executed on a KA10 with 256K of memory or a KI10, KL10 with 4096K of memory. It is suggested that the user deselect at least one 16K bank (or 8K if MA10) in order to test nonexistent memory flag.
Test 22 Subtest	DF10/DAS33 SKIP TEST
SKPTST	Skip Read Test for DF10/DAS33 Set up control word [-17,,0] and start DF10/DAS33. Shadow (core 0-17) ACs are set to 0 and DF10/DAS33 started. DF10 should skip 17 words and terminate without writing any information received from the device into memory. If the DF10/DAS33 does write, it will probably start with location 0, so check that the shadow (core 0-17) ACs are still zero.
Test 23 Subtest	DF10/DAS33 MULTIPLE CONTROL WORD FETCH
RHMUL	Fetch Multiple Control Words From Memory Control word is a skip of 17 words done 12 times. Check that the shadow (core 0-17) ACs are not written because a skip should never write anything; if it does, it will probably go into core 0 to 17. Also check the written control word (i.e., that the DF10/DAS33 fetched all control words and did not terminate early). Cause for early termination are as follows. 1. The DF10/DAS33 did not fetch a control word. 2. The DF10/DAS33 had a control word parity error. 3. The DF10/DAS33 jumped to core 0 which has a 0 in it and is a DF10/DAS33 terminate. 4. The DF10/DAS33 had a nonexistent memory which could occur when the DF10/DAS33 had a JUMP to 0 and A -1 was in location 0.
Test 24 Subtests	DF10 PARITY ERROR DETECTION TEST
RHPAR	Test the Parity Error Detection Logic of the DF10/DAS33. Read from the RS04 and write even parity into memory. Then check that the DF10/DAS33 did in fact write even parity into memory, then use the data written as a control word and expect a control word parity error. Then use a good control word pointing to buffer and expect a data word parity error.

Table 2 DDRHA Test Summary (Cont)

Test	Description
	Now check that the processor can detect a memory parity error before checking with the DF10/DAS33 because the DF10 might not have written even parity into memory. This could be because the write even parity lockout switch is set to the on position.
RHPAR2	Now check to see if a control word parity error (CWPE) occurs when 760 contains a JUMP to buffer and buffer has an even parity word.
RHPAR3	Write a sector full of even parity data words into memory. Form a control word so that it fetches even parity data words and causes a data word parity error.
	Now check that the processor can detect a memory parity error before checking with the DF10/DAS33 because the DF10 might not have written even parity into memory. The cause may be that the write even parity lockout switch is set to the on position.
RHPAR5	Now check that a good control word (odd parity) which causes the DF10/DAS33 to fetch an even parity word results in a data word parity error and the RH10 data buffer register is clear.
Test 25 Subtest	**DF10 CONTROL WORD FETCH TEST**
RHFET	Starting at the first free memory address over the end of the program, fetch a control word (the control word is a terminate command for the DF10/DAS33). The test will call control words up through memory in 1K jumps until all available memory has been tested.
Test 26 Subtest	**MAINTENANCE MODE HIGH-SPEED READ TEST**
MHSRD	Set up DF10/DAS33 command list for a 64-word transfer to memory to simulate a 1-sector read from a Massbus device. The following patterns will be read into a memory buffer and verified. 1. All 0s 2. All 1s 3. Alternate bits 4. Complement alternate bits 5. Floating 0s 6. Floating 1s 7. DF10/DAS33 data patterns 8. DF10/DAS33 parity patterns **NOTE** **Set switch 3 (print inhibit switch) if error looping is desired on this test.**
Test 27 Subtest	**MAINTENANCE MODE HIGH-SPEED WRITE TEST**
MHSWRT	Set up the DF10/DAS33 command list for a 64-word transfer from memory to simulate a 1-sector write transfer to a Massbus device. These patterns will be written from a memory buffer through the DF10/DAS33 and RH10 back to memory in maintenance mode. 1. All 0s 2. All 1s 3. Alternate bits 4. Complement alternate bits 5. Floating 0s 6. Floating 1s 7. DF10/DAS33 data patterns 8. DF10/DAS33 parity patterns **NOTE** **Set switch 3 (inhibit print switch) if error looping is desired in this test.**

Table 2 DDRHA Test Summary (Cont)

Test	Description
Test 30 Subtest	DAS33 INITIAL ADDRESS OVERFLOW
RHIA0	This test is for DAS33S only. This test requires a DAS33 with status writeback enabled and the WRITE CW REQ. switch in the up position. The test gives the DAS33 an initial address of 776 which contains a jump to 100, which contains a terminate. The test checks to see that INAD + 1 through INAD + 9 (777 to 1007) are written into.

GENERAL INFORMATION

Code	DDRPD.SAV
Title	DECsystem-10 User Mode Disk Pack Performance Test
Abstract	DDRPD is a user-mode-only program to be used as performance test of all disk units on the system. Read-only operations are performed on the selected units, and errors are reported as they are encountered. For moving head disks DDRPD reads all cylinders sequentially until all units have been read, then randomly reads all cylinders until all units are done and then recycles. Two types of errors are detected: recoverable and nonrecoverable. Recoverable errors are detected by scanning the unit's DDB and checking its soft error count. Nonrecoverable errors failed to go away during the monitor's retry sequence, and the monitor took the error return after the read operation.
Hardware Required	A DECsystem-10 with RP02, RP03, RD10, RM10B, RS04, or RP04 disk units.
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	1. Unit types are currently restricted to: RP02-RP03 (Memorex or ISS), RD10 (Burroughs Disk), RM10 (Bryant Drum), RP04 (ISS), or RS04 (DEC). The unit must be on-line and ready. 2. The TOPS-10 monitor must be level C or later. 3. Masking of bad tracks on a disk or drum will not work if the bad track is above track 35(10). When bad surfaces are masked, the program will continue to count cylinders and sectors through that surface. These are complicated software problems and will probably not be corrected.
Notes	1. This diagnostic has been tested on 1055, 1070, 1077 monitors with RP02s, RP03s, RP04s, RS04s, RM10Bs, and RD10s. 2. When running more than one copy of the program on the same unit, a conflict may arise when reporting errors. 3. The memory indicators register (MI) will reflect the following information. Cylinder MI bits 0-8 Surface MI bits 18-23 Sector MI bits 30-35 4. The cycle time of DDRPD is dependent on the load on the system and the mode of operation. Typical times are as follows (multiblock mode and consecutive surfaces). RP02 20 minutes RP03 40 minutes RP04 80 minutes RD10 6 minutes RM10B 5 minutes RS04 5 minutes
Loading and Starting Procedure	Login under [1,2] or run WHEEL Type: RUN DDRPD<CR>
Control Switches	None

OPERATIONAL CONTROL
Upon starting, DDRPD prints a title message and asks the following questions. The questions must be answered with a Y (yes) or N (no). Any response other than Y or N will result in a more detailed explanation of the question.

RUN IN HIGH PRIORITY QUEUE? Y OR N <CR>-
A yes answer to this question will put DDRPD in the high priority run queue. This will enable a disk (or drum) to be tested quickly; however, it is not recommended for use during normal timesharing as it tends to degrade system performance.

LOCK IN CORE? Y OR N <CR>-
A yes answer will lock DDRPD in core. This is useful if the user may be swapped onto the device being tested (usually disk or drum), and wishes to prevent this. This mode will not normally degrade system performance but will decrease the amount of total system core available to other users.

RUN IN HIGH DISK PRIORITY? Y OR N <CR>-
This mode will enable the user's disk I/O to be scheduled before all other disk I/O for that unit. This is useful if the user wishes to quickly run the program and get out during heavy timesharing. It will degrade system performance and is not suggested for use during normal timesharing.

RUN IN SINGLE BLOCK MODE? Y OR N <CR>-
Single-block mode reads each disk block one block at a time. This mode is very slow but is a good check of clobbered sector headers. The alternative is multiblock mode which will read the disk in increments of 10 blocks. This mode is much faster and will uncover surface timing problems and most problems with clobbered sector headers, parity errors, and search errors.

SLEEP OPTION? Y OR N <CR>-
The sleep option is useful for delaying unit testing if other users are accessing the disk unit to be tested. If the question is answered yes and other users are accessing that unit, then the program will sleep for 1 second after each read cycle of the program.

CONSECUTIVE SURFACES (TRACKS)? Y OR N <CR>-
A no answer to this question will cause a random access of surfaces and sectors. This mode along with multiblock mode does a lot of positioning, thus checking out the positioning hardware of RP02, RP03, or RP04 units.

DATA ERROR CHECKING? Y OR N <CR>-
A yes answer to this question will cause the job's core area to be scanned for parity errors after each disk read error.

ENABLE SYSERR REPORTING? Y OR N <CR>-
Normally if SYSERR reporting is enabled, all errors which have occurred as a result of running this program will be recorded in a file ERROR.SYS[1,4]. If disabled, this will not occur.

ENABLE MONITOR RETRIES? Y OR N <CR>-
A yes answer to this question will cause monitor to attempt error recovery. Only nonrecoverable errors will be reported to DDRPD. A yes answer will cause any error detected to be reported to DDRPD.

The monitor defines read errors as nonrecoverable after 101 unsuccessful retries.

NAME THE UNIT TO BE TESTED (<CR> FOR ALL)
The unit name specified must be an actual disk unit name, such as DPA0, DPB3, FHA0. Names such as DSKA0, DSKB, DSKC are unacceptable. The unit name must also be a disk unit. Names such as PTR, PTP, LPT will be treated with great disrespect. (Try it and you'll find out.)

HOW MANY PASSES? (<CR> FOR INFINITE)
At the end of each pass, the iteration count is outputted to either the terminal or the external device unless the device under test is a fixed-head disk. Then the count is only typed on every 10 passes.

Eliminating Disk Pack Surface
If it is desirable to suppress printouts for certain surfaces on the current unit type:

```
↑C↑C      (two control Cs)
.REE<CR>  (reenter)
```

DDRPD will respond with:

```
NAME THE BAD SURFACES FOR X Y Z:
```

Type the surfaces separated by commas and terminated by <CR> (e.g., 1, 5, 17<CR>)

DDRPD will then continue at the point where it was interrupted.

Forced Status Checking
At any time after the program has been initialized and is running, the user may type ALTMODE or ESCAPE. The program will respond with the following information.

```
************************
CURRENT STATUS: DPA0
PASS COUNT: 0
READ MODE IS; 10 BLOCK - NONCONSECUTIVE SURFACES
BLOCK  CYL\TRK  SURF  SECT
(10)    (10)     (10)  (10
111     0        11    1
TOTAL ERRORS THIS UNIT: 1
TOTAL ERRORS THIS SYSTEM: 1
************************
```

This is useful in determining what the current status of the program and unit under test is. The information is self-explanatory.

DDRPD ERROR SUMMARY
Errors will be reported to the user's terminal unless another device is assigned as DEV, in which case the output will go to this device. If the device is a directory device; e.g., DSK, DTA, the output will be written as RPD (Job #).PNT. The heading is self-explanatory. The block number, surface (track), sector, cylinder, DATAI, CONI, and a character (either U or R to denote Recoverable or Unrecoverable error) is typed out. Also the unit ID and the data channel number that its controller is connected to. Included in the typeout for the CONI and DATAI is an expansion of the error listing to interpret the error bits. In the case of Massbus devices (RP04, RS04) all the DATAIs are typed and expanded if needed. Such things as the drive look-ahead register, and others which simply hold random data, are not expanded but simply printed.

NOTE

Even when printing on an external device, the user is notified of errors on the controlling terminal via a "#" being typed upon each occurrence of an error. Also, when the error count has exceeded 50(10) the user is queried as to whether or not the unit under test is formatted.

Example:

```
DPA0-RP02 SEQUENTIAL ADDRESSING MODE
BLOCKS MARKED WITH <> FAILED IN A 10 BLOCK READ BUT NOT ON A BLOCK
BY BLOCK REREAD!

BLOCK  CYL   SURF  SECT  CONI  DATAI  RECOVERY
(10)   (10)  (10)  (10)  (8)   (8)

1      0     0     1     000001  004015  000261  140000 U
CONI:  DISK WORD PARITY ERROR, PARITY ERROR, PIA 33, PIA 35,
DATAI: HEADS IN POSITION, DISK ON LINE, WRITE HEADER LOCKOUT.
```

There is also a secondary class of errors which are not related to the unit under test but do affect the operation of the program. All system-type errors are intercepted and handled according to their severity. These errors include the following.

Error	Response
ILLEGAL MEMORY REFERENCE	EXIT TO DDT
ADDRESS CHECK	EXIT TO DDT
ILLEGAL UUO	EXTT TO DDT

I/O TO UNASSIGNED CHANNEL	EXIT TO DDT
TIME LIMIT EXCEEDED	EXIT TO DDT
EXCEEDED QUOTA	INFORM USER AND CONTINUE
FILE STRUCTURE FULL	INFORM USER AND CONTINUE
DISK OFF LINE	WAIT 45 SECONDS AND CONTINUE
OPERATOR ACTION REQUIRED	WAIT 45 SECONDS AND CONTINUE

User parity errors are also detected and handled by this program. On a parity error occurrence, all of the user's core area is read for recurrence of the parity error. These errors are treated as nonfatal if only one occurs. If more than one occurs, the monitor parity error recovery code then takes over and the job is stopped.

GENERAL INFORMATION

Code	DDRPH.A10
Title	RH10/RP04 Basic Device Diagnostic
Abstract	DDRPH is the RH10/RP04 basic device diagnostic, designed to isolate solid RP04 faults to the faulty module or group of modules within the DCL or drive electronics. Excellent scope looping capabilities have been designed into each test. Testing is done on a start small and build up basis. The diagnostic starts by testing even the most trivial of control bus cycles and ends up by doing full speed data transfer operations to various disk addresses. Positioning logic is also tested. In addition to the straight line diagnostic tests, the diagnostic also contains a head alignment/verification subprogram that is invoked by setting the proper sense switch.
Hardware Required	KA10, KI10 or KL10 CPU/48K of core (minimum)/RH10/DF10 (up to 6)/up to 48 (single- or dual-port) RP04s
Preliminary and Associated Programs	Refer to diagnostic hierarchy (I0/10 STD module).
Restrictions	Examination of switch 13 takes place only once, immediately following the initial load of the diagnostic into core from its storage media. The cache option is selected only during the very first program initialization cycle. Attempting to alter the state of the switch from that point on will have no effect.
Notes	1. This diagnostic has been designed to run in both exec and user mode. 2. The diagnostic runs in user mode with monitors that support the DIAG feature, TOPS-20 REL-3 or newer, TOPS-10 REL 6.03 or newer. Do not run it under earlier versions of the monitor - this could cause a system crash. 3. This diagnostic reads and writes various pack areas. It is necessary to write on certain areas of the pack and when the diagnostic completes each pass, the areas which it has written are left in PDP-10 format. This implies that if a PDP-11 formatted scratch pack is used, it will have to be reformatted after a pass of this diagnostic. If the diagnostic is stopped in the middle of a pass, it is impossible to determine the state of the pack at that time. 4. An asterisk (*) in the test column of Table 2 indicates that the listing (on microfiche) contains a recommended troubleshooting procedure. 5. Table 2 documents the individual tests for DDRPH, DDRPK, DFRPH and DFRPK. In some cases individual tests will vary between diagnostics. In such cases a note will appear in the test column. 6. Testing Single-Ported Devices The following sequence must be followed to ensure complete testing of a single-ported device.

a. Cable the controller to the active port, lock the device on that port, and run one pass of the diagnostic with the manual intervention tests selected.

b. If multiple passes are desired, it will not be necessary to run the manual intervention tests following the first pass.

7. **Testing Dual-Ported Drives**
The following sequence must be followed to ensure complete checkout of dual-ported drives.

a. Cable the controller to either port, lock the drive on that port, and run a pass of the diagnostic with the manual intervention tests selected. This will test one port of the drive.

b. Cable the controller to the other port, lock the drive on that port, and run a pass of the diagnostic, including the manual intervention tests. This will test the other drive port.

c. Run the dual-port test as instructed by the diagnostic. This will test the dual-port arbitration logic in the drive.

8. **Run Time (Per Drive Basis)**

a. If no operator intervention tests are run, the diagnostic will complete a pass in approximately 2 minutes 20 seconds.

b. The dual port tests take approximately 2 minutes to execute once the drive is cabled up.

c. Excluding warm-up time and set-up time, head alignment verification takes approximately 5 minutes.

Loading and Starting Procedure: Standard (Refer to the 10/10 STD module.)

Control Switches: Refer to Table 1.

OPERATIONAL CONTROL
The diagnostic is designed to be self-documenting in that it furnishes all necessary instructions for operation as it runs. Sense switch options may also be printed at run time.

Exec Mode
The diagnostic will first poll the system and report which disk drive and controller configurations exist. After reporting system configuration, the user is allowed to select any number of controllers for test (out of those found).

Once controllers have been selected, the user is allowed to select drives for test on each of the controllers selected. A drive may be selected for test regardless of whether it was detected at configuration time. Following the completion of configuration, the diagnostic goes to its test dispatcher to determine which tests are to be run.

User Mode
The program will not report Massbus configuration. It will immediately ask for RH and drive numbers.

You can exit from the job with a ↑C and continue where you left off with the monitor continue commands. The ↑C normally happens rapidly; however, if the diagnostic is waiting for terminal input you may have to type the ↑C followed by a <CR> to activate the software.

Dispatcher
In order to understand the dispatcher it is important to understand the following facts about the test structure.

1. There are 363 (octal) in-line stand-alone tests, some of which require operator intervention. These tests may be selected for individual operation by SW 27-35.

2. The dual port test is a stand-alone series of 23 (octal) subtests that are self-documenting with complete instructions.

3. Head alignment-verification is a stand-alone subroutine, but requires the use of a CE pack and DDU tester.

   The dispatch algorithm works as follows.

   a. Read the console switches.

   b. If pause switch (002000) is set, go to step A.

   c. If the dual-port option switch is set (040000), run the dual-port tests on dual-ported drives only. Then return to step A.

   d. If the head alignment option switch is set (020000), perform head alignment-verification on the drives selected for test. Then return to step A.

   e. If the operator intervention switch is ON, set a flag that allows operator intervention tests to be executed when encountered (flag is checked locally at each test).

   f. If SW26 is set (001000) (manual test selection), execute the test whose number is specified in switches 27-35. If illegal, give an error message. The test is executed one time and then return is made to step A.

   g. Execute the next in-line test of the 363 available, one time, and then return to step A. When all tests have been executed, end of pass is typed and the sequence will be repeated until the diagnostic is stopped or the abort switch is raised.

NOTE
When an in-line test is being run, it is executed on all drives being tested before return is made to the dispatcher.

Use of the Diagnostic
A typical use of the diagnostic would be as follows.

Run the diagnostic until it fails on a test and then select that test from the switches so that it will be continuously run and the trouble can be isolated or modules replaced. After replacing a module or group of modules it is suggested that the entire diagnostic be rerun just in case symptoms have changed.

Due to the structure of the diagnostic, it is recommended that troubleshooting be done using the earliest possible test that failed. In the event that several tests indicate malfunctions, check them out one at a time.

The individual tests performed by this diagnostic are described in Table 2.

Table 1 DDRPH Control Switch Summary

Switch	State	Description
0-15		Standard (Refer to the 10/10 STD module.) Exceptions - The following switches are not implemented: 1, 2, 9, 11, 12, and 15.
16	0	60 Hz power KA10/KI10
	1	50 Hz power KA10/KI10
17		Not used
18	0	Normal operation
	1	Trace program's progress test by test
19	0	Normal operation
	1	List all possible modules if not running in text inhibit mode
20		Not used
21	0	Normal operation
	1	Run the dual-port tests
22	0	Normal operation
	1	Run the head alignment verification test
23		Not used
24		Not used
25	0	Normal operation
	1	Pause until this switch is reset
26	0	Normal operation
		Run the test specified by switches 27-35
27-35		Specify the test that will be continuously run if SW26 is set (1000).

Table 2 DDRPH Test Summary

Test	Description
1*	DUAL DRIVE SELECTION All drives on the bus are powered up except the drive being tested. The diagnostic will do a blind read of drive register number 6 (the drive type register) to guarantee that no other drive on the Massbus responds to this drive's device number. If the test is unsuccessful, the drive type and serial number registers of the responding device on the bus will be printed in order to aid in the device isolation.
2*	DEMAND/TRANSFER TEST The drive being tested is now powered up. The diagnostic does a blind read of register number 6 in the drive (drive type) looking only for proper operation of demand and transfer over the Massbus. Improper operation would result in a control bus timeout. Data and parity errors are ignored at this point in the testing.
3*	TEST FOR CONTROL BUS BITS STUCK ON ONE This test checks for stuck bits on the Massbus and the device's wire-ORed bus lines by reading a nonexistent register (register 20 octal) within the drive. Reading a nonexistent register should return all 0s over the control bus if control bus logic is working properly.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
4*	TRANSCEIVER ENABLE TEST This test reads the drive type register looking for a legal device type. It is the first time 1s and 0s are read. This test is actually to guarantee that the transceivers are enabled during a control bus read cycle and that the C to D (direction of transfer signal) is not incorrect during the control bus read cycle. In getting this test operational, drive-type register problems are debugged as a by-product. The program reads the drive-type register. Successful operation indicates that the entire control bus read cycle is operational.
5*	RESET/WRITE/READ 1S TO OFFSET REGISTER Master reset, then write/read/verify 1s from the low-order 7 bits of the offset register. This verifies that the receivers pass 1s.
6*	RESET/WRITE/READ 0S TO OFFSET REGISTER Write 1s in bits 0-6 of offset register. This is known to work from test 5. Reset and write 0s; read and verify that bits 0-6 have gone to 0.
7*	RESET/WRITE 0S/WRITE 1S/READ/VERIFY OFFSET REGISTER This test is similar to test 5, but it writes 1s to flip-flops that are guaranteed to contain 0s, thus verifying that transmitters pass 1s, that reset is not stuck in clear direction, and that flip-flops are being loaded.
10*	WRITE 1S/WRITE 0S/READ/VERIFY OFFSET REGISTER This test is similar to test 6 but this time it is guaranteed that the flip-flops are 1s before the 0s are written, thus verifying that the transmitters pass 0s to the flip-flops and the flip-flops clear via the data inputs.
11*	WRITE 1S/RESET/READ/VERIFY OFFSET REGISTER Write 1s to the flip-flops, issue a reset (Massbus clear) to verify that the reset pulse is clearing the proper bits. This also verifies that the reset pulse is getting over the Massbus to the DCL properly.
12*	RESET/WRITE/READ/VERIFY 1S TO ERROR REGISTER NO. 3 Master reset, then write/read/verify 1s to error register no. 3 (ERR-3). This verifies that the receivers pass 1s.
13*	RESET/WRITE/READ 0S TO ERROR REGISTER NO. 3 Write 1s to the register. This is known to work from test 12. Reset/write 0s/read/verify that the bits were cleared. This verifies that the flip-flops and receivers work with 0s.
14*	RESET/WRITE 0S/WRITE 1S/READ/VERIFY ERROR REGISTER NO. 3 This is similar to test 12 except that this test writes 1s to flip-flops that are guaranteed to contain 0s, thus verifying that the transmitters are passing 1s, that reset is not stuck in the clear direction, and that flip-flops are loading via their data inputs.
15*	WRITE 1S/WRITE 0S/READ/VERIFY ERROR REGISTER NO. 3 This test is similar to test 13 except that it is certain that the flip-flops are 1s when the 0s are written, thus verifying that the transmitters pass 0s to the flip-flops and the flip-flops clear via the data inputs.
16*	WRITE 1S/RESET/READ/VERIFY ERROR REGISTER NO. 3 Write the flip-flops with 1s. Issue a reset (Massbus clear) and verify that the flip-flops are cleared by the reset.
17*	RESET/WRITE/READ/VERIFY 1S TO ERROR REGISTER NO. 2 Master reset, then write/read/verify 1s from the high-order 2 bits of error register no. 2 (ERR-2). This verifies that the receivers pass 1s.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
20*	RESET/WRITE/READ 0S TO ERROR REGISTER NO. 2 Write 1s in the flip-flops. This is known to work from test 17. Reset/write/read/verify that the flip-flops are cleared. This verifies that the flip-flops and receivers work with 0s.
21*	RESET/WRITE 0S/WRITE 1S/READ/VERIFY ERROR REGISTER NO. 2 This is like test 17 except that this test writes 1s to flip-flops that are guaranteed to be 0, thus verifying that the transmitters are passing 1s, that reset is not stuck in the clear state, and that flip-flops are loaded via their data inputs.
22*	WRITE 1S/WRITE 0S/READ/VERIFY ERROR REGISTER NO. 2 This test is similar to test 20 except that this time, it is guaranteed that the flip-flops are 1s when the 0s are written, thus verifying that the transmitters pass 0s to the flip-flops and the flip-flops are clear via the data inputs.
23*	WRITE 1S/RESET/READ/VERIFY ERROR REGISTER NO. 2 Write 1s to the flip-flops, issue a reset and verify that the flip-flops are cleared by the reset pulse.
24*	TEST FOR PERSISTENT ERROR REGISTER NO. 3 BIT 15 ERROR Master reset, read and verify that bit 15 is not stuck on a 1. This verifies that there is not a persistent (off cylinder) error that invalidates further testing of the control bus.
25	TEST FOR PERSISTENT ERROR REGISTER NO. 3 BIT 14 ERROR Master reset, read and verify that bit 14 is not stuck on a 1. This verifies that there is not a persistent (SEEK INCOMPLETE) error that invalidates further testing of the control bus.
26	TEST FOR PERSISTENT ERROR REGISTER NO. 3 BIT 06 ERROR Master reset, read and verify that bit 06 is not stuck on a 1. This verifies that there is not a persistent (DC LOW) error that invalidates further testing of the control bus.
27	TEST FOR PERSISTENT ERROR REGISTER NO. 3 BIT 05 ERROR Master reset, read and verify that bit 05 is not stuck on a 1. This verifies that there is not a persistent (DC LOW) error that invalidates further testing of the control bus.
30* (DDRPH) (DFRPH)	TEST FOR PERSISTENT ERROR REGISTER NO. 3 BIT 04 ERROR Master reset, read and verify that bit 04 is not stuck on a 1. This verifies that there is not a persistent (DISABLE) error that invalidates further testing of the control bus.
30 (DDRPK) (DFRPK)	TEST FOR PERSISTENT ERROR REGISTER NO. 3 BIT 04 ERROR Master reset, read and verify that bit 04 is not stuck on a 1. This verifies that there is not a persistent (35 VF) error that invalidates further testing of the control bus.
31 (DDRPH) (DFRPH)	TEST FOR PERSISTENT ERROR REGISTER NO. 3 BIT 03 ERROR Master reset, read and verify that bit 03 is not stuck on a 1. This verifies that there is not a persistent (unsafe other than read/write) error that invalidates further testing of the control bus.
31 (DDRPK) (DFRPK)	TEST FOR PERSISTENT ERROR REGISTER NO. 3 BIT 01 ERROR Master reset, read and verify that bit 01 is not stuck on a 1 indicating a persistent WT-OFS condition.
32* (DDRPH) (DFRPH)	TEST FOR PERSISTENT ERROR REGISTER NO. 3 BIT 01 ERROR Master reset, read and verify that bit 01 is not stuck on a 1. This verifies that there is not a persistent (velocity unsafe) error that invalidates further testing of the control bus.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
32 (DDRPK) (DFRPK)	TEST FOR PERSISTENT ERROR REGISTER NO. 3 BIT 00 ERROR Master reset, read and verify that bit 00 is not stuck on a 1. This verifies that there is not a persistent (DC unsafe) error that invalidates further testing of the control bus.
33 (DDRPH) (DFRPH)	TEST FOR PERSISTENT ERROR REGISTER NO. 3 BIT 00 ERROR Master reset, read and verify that bit 00 is not stuck on a 1. This verifies that there is not a persistent (pack speed unsafe) error that invalidates further testing of the control bus.
33 (DDRPK) (DFRPK)	TEST FOR PERSISTENT ERROR REGISTER NO. 3 BIT 13 ERROR Master reset, read and verify that bit 13 is not stuck on a 1. This verifies that there is not a persistent OPE error that invalidates further testing of the control bus.
34	TEST THAT ERROR REGISTER NO. 3 CAN BE LOADED WITH 1S Master clear, write 1s/read/verify that all bits in the register are set. This just checks flip-flops in the register. All bits are known to clear from previous tests.
35	TEST THAT ERROR REGISTER NO. 3 CAN BE LOADED WITH 0S Write 1s/write 0s/read/verify that all register bits are 0. This just checks that all flip-flops in the register can be loaded with 0s. Bits are known to set from previous tests.
36*	SINGULARITY OF ERROR REGISTER NO. 3 BITS (FLOAT 1S) Float a 1 down the register to guarantee that no bits interfere with one another. Before each 1 is loaded, the register is cleared.
37*	SINGULARITY OF ERROR REGISTER NO. 3 BITS (FLOAT 0S) Float a 0 down the register to guarantee that no bits interfere with one another. Before each bit is loaded, the register is preset to 1s.
40	ERROR REGISTER NO. 3 DIRECT RESET TEST Load the register with 1s. Master clear the register (Massbus clear) and verify that the register is in fact cleared. Just tests ability of all flip-flops to clear.
41	TEST THAT PARITY NETWORK CAN GENERATE A ONE Offset register is cleared with a master reset and then read over the control bus. The parity bit should be a 1 since the data field is 0s.
42	TEST THAT PARITY NETWORK CAN GENERATE A ZERO A single bit is written to error register no. 3 (bit 00). The register is then read back over the control bus. Verify that the parity bit is a 0 since the data field contains an odd number of 1s.
43	TEST PARITY NETWORK WITH VARIOUS PATTERNS Load and read a series of seven patterns over the control bus using error register no. 3 register. These patterns have been selected to guarantee that the parity networks themselves have no internal failures that will cause them to make a wrong decision.
44	Deleted
45*	RESET/READ/VERIFY C01 ERROR REGISTER NO. 1 Master reset, read and verify that the illegal register (ILR) bit is not stuck on a 1. It is necessary to have the ILR bit operational for further register selection tests.
46*	WRITE/READ/VERIFY A 1 TO C01 ERROR REGISTER NO. 1 Write/read and verify that a 1 can be written to the ILR flip-flop. C01 of error register no. 1.
47	WRITE A 1/RESET/READ/VERIFY C01 ERROR REGISTER NO. 1 This test is similar to test 45 except that it guarantees the flip-flop is set when the reset is applied.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
50	WRITE A 1/WRITE A 0/READ VERIFY C01 ERROR REGISTER NO. 1 This test ensures that C01 error register no. 1 can be cleared via its data input.
51*	SEE THAT READING REGS 20-37 CAUSES ILR ERRORS Although only REG SEL 04 seems important in determining whether or not an illegal register is selected, testing all of the nonexistent registers may catch coupling between the register select bits.
52*	SEE THAT READING REGS 00-17 DOES NOT CAUSE ILR Although only REG SEL 04 seems important in determining whether or not an illegal register is selected, testing all the legal registers may catch coupling between the register select bits.
53*	VERIFY OPERATION OF CONTROL REGISTER C03 This test gets C03 checked out along with using the control register for the first time, which will check part of the register select logic. Four discrete subtests are tried. 1. Write/read/verify a 1 to C03 2. Write a 1/write a 0/read/verify C03 3. Write a 0/write a 1/read/verify C03 4. Write a 1/reset/read/verify that C03 did not clear
54*	RESET/READ/VERIFY A ZERO IN C03 ERROR REGISTER NO. 1 It is necessary to get this bit working in order to do some later register tests. This is a tricky bit to test since it is the parity error detection bit and we are writing registers to test the bit. This resets the bit and verifies that the flip-flop is not stuck on a 1.
55*	RESET/WRITE 0/VERIFY C03 ERROR REGISTER NO. 1 Writing 0s to error register no. 1 and then checking C03 ensures that the drive's parity detection network is not detecting a parity error when the drive receives a parity bit of 1.
56*	RESET/WRITE 1 TO C01 ERROR REGISTER NO. 1/READ/VERIFY C03 ERROR REGISTER NO. 1 Writing a single 1 to C01 of error register no. 1 and verifying that (the parity bit) C03 error register no. 1 did not set ensures that the drive's parity detection network is not detecting a parity error when the drive receives a parity bit of 0.
57*	RESET/WRITE A 1/READ/VERIFY C03 ERROR REGISTER NO. 1 It has been proven that writing a single 1 does not cause a parity error. Now reset and write a 1 to C03 error register no. 1 to ensure that the flip-flop can be set.
60*	RESET/WRITE A 1/WRITE A 0/READ/VERIFY C03 ERROR REGISTER NO. 1 This test ensures that the C03 error register no. 1 bit can transition from the 1 to 0 state (can be cleared by writing a 0).
61*	RESET/WRITE A 1/RESET/READ/VERIFY C03 ERROR REGISTER NO. 1 This test ensures that the C03 flip-flop clears with direct reset.
62*	CAUSE A PARITY ERROR. SEE IF IT'S DETECTED This tests the parity detection logic by having the controller write the register with incorrect parity. C03 error register no. 1 should set since it is the parity error flip-flop.
63*	GUARANTEE THAT GO BIT C00 CONTROL REGISTER NOT STUCK Master reset, read, verify that the go bit is not stuck at 1. This is a necessary prerequisite for testing C03 in maintenance register.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
64*	VERIFY OPERATION OF C00 MAINTENANCE REGISTER Testing this bit is a prerequisite test for testing C03 maintenance register. This bit (C00 maintenance register) is the diagnostic maintenance mode bit. It must be operational before other bits in the maintenance register can be tested. A group of 4 tests are performed to verify operation of this bit. 1. Write a 1/read/verify C00 is a set 2. Write a 1/write a 0/read/verify C00 is clear 3. Write a 0/write a 1/read/verify C00 is set 4. Write a 1/reset/read/verify C00 is clear
65*	RESET/WRITE 1S/READ/VERIFY C01-C04 This test will catch register select failures because writing C01-C04 should not cause any bits to set since the diagnostic bit is clear.
66	VERIFY COMPLETE OPERATION OF C01-C04 MAINTENANCE REGISTER These four subtests check complete operation of C01-C04 of maintenance register, in conjunction with the maintenance mode bit C00 maintenance register. 1. Set M-M bit (M-M = maintenance mode bit C00) Write 1s to C01-C04/read/verify 2. Set M-M bit Write 0s to C01-C04/read/verify 3. Set M-M bit Write 0s/write 1s/read/verify C01-C04 4. Set M-M bit Write 1s/reset/read/verify C01-C04
67*	PARTIAL TEST OF C03 DESIRED SECTOR/TRACK ADDRESS REGISTER This test is essentially to make C03 desired sector/track address register readable and writable so the bit can be used later in register selection tests. The reset test for this flip-flop will be done later. This test could turn up register select problems since this is the first time we are addressing desired sector/track address. This test comprises the following three subtests. 1. Write a 1/read/verify C03 2. Write a 1/write a 0/read/verify C03 3. Write a 0/write a 1/read/verify C03
70*	PARTIAL TEST OF C03 DESIRED CYLINDER ADDRESS REGISTER This test is to verify that C03 is readable and writable so the bit can later be used to test register selection logic. This test could identify register select problems since this is the first time desired cylinder address register is addressed. The test comprises the following three subtests. 1. Write a 1/read/verify C03 2. Write a 1/write a 0/read/verify C03 3. Write a 0/write a 1/read/verify C03
71*	VERIFY THAT C03 ERROR REGISTER NO. 2 IS NOT STUCK AT 1 Issue a reset and verify that C03 error register no. 2 is not stuck in the set state or held set by a persistent (current switch unsafe) error from the VENDOR logic.
72	VERIFY COMPLETE OPERATION OF C03 ERROR REGISTER NO. 2 REGISTER This test comprises a series of four subtests to ensure that the bit is working properly and can be used in register select tests. The subtests are as follows. 1. Write a 1/read/verify C03 2. Write a 1/write a 0/read/verify C03 3. Write a 0/write a 1/read/verify C03 4. Write a 1/reset/read/verify C03
73*	TEST SELECTION OF DCL WRITABLE REGISTERS The object of this test is to guarantee that no writable DCL register interferes with any other register because of a register selection problem or a register select decoding problem.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
74*	VERIFY THAT DRIVE CLEAR WORKS This test verifies that drive clear operates. 1. Put drive in maintenance mode (set diagnostic bit maintenance register) 2. Issue a drive clear with the go bit set. 3. Read maintenance register and verify that the drive is no longer in maintenance mode. Drive clear should cause the maintenance mode bit to clear.
75	VERIFY SOME GO BIT CONTROL LOGIC This tests some of the gating to the D-input of the STO GO flip-flop (maintenance register). It ensures that issuing a drive clear command without the GO bit does not cause a drive clear. It does the following: 1. Puts drive in maintenance mode 2. Issues a drive clear with the GO bit set, but writes the register with incorrect parity 3. Reads the maintenance register to see if drive is still in maintenance mode. It should be, because the parity error should inhibit the drive clear from taking the drive out of maintenance mode.
77*	TEST FOR POSSIBILITY OF MULTI-DRIVE RESPONSE This test is effectively the complement of test 1. This test ensures that the drive being tested responds to no other drive's address but its own.
100* (DDRPH) (DFRPH)	ENSURE THAT A PERSISTENT MSE DOES NOT EXIST Reset/write 0/read/verify that C04 error register no. 2 motor sequence error (MSE) is not stuck at 1.
100 (DDRPK) (DFRPK)	ENSURE THAT A PERSISTENT (RAW) DOES NOT EXIST Reset/write 0/read/verify that C04 error register no. 2 read and write (RAW) is not stuck at 1.
101 (DDRPH) (DFRPH)	VERIFY THAT 30 V UNSAFE IS NOT BEING DETECTED Reset/write 0/read/verify that C12 error register no. 2 is not stuck at 1. In normal operation the unregulated 30 volt supply feeds the 20.5 V and 12. 5 V supplies.
101 (DDRPK) (DFRPK)	Deleted
102*	VERIFY THAT A MULTIPLE HEAD SELECT PROBLEM DOES NOT EXIST Reset/write 0/read/verify that C10 error register no. 2 is not a 1.
103*	VERIFY THAT A NO HEAD SELECT PROBLEM DOES NOT EXIST Reset/write 0/read/verify that C10 error register no. 2 is not a 1.
104*	VERIFY THAT A PLO UNSAFE DOES NOT EXIST Reset/write 0/read/verify that C13 error register no. 2 is not at 1.
105*	VERIFY THAT PERSISTENT INDEX ERROR DOES NOT EXIST Reset/write 0/read/verify that C11 error register no. 2 is not stuck at 1.
106	VERIFY THAT C08 ERROR REGISTER NO. 2 WRTRU IS NOT STUCK AT 1 Reset/write 0/read/verify C08 error register no. 2.
107*	VERIFY THAT C05 ERROR REGISTER NO. 2 TRADF IS NOT STUCK AT 1 Reset/write 0/read/verify that there is not a persistent transition detector failure.
110*	VERIFY THAT C06 ERROR REGISTER NO. 2 TRANSU IS NOT STUCK AT 1 Reset/write 0/read/verify that C06 error register no. 2 is not at 1.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
111*	VERIFY THAT C02 ERROR REGISTER NO. 2 WSUNS IS NOT STUCK AT 1 Reset/write 0/read/verify C02 error register no. 2.
112*	VERIFY THAT C01 ERROR REGISTER NO. 2 CFAIL IS NOT STUCK AT 1 Reset/write 0/read/verify C01 error register no. 2.
113*	VERIFY THAT C00 ERROR REGISTER NO. 2 WCUNS IS NOT STUCK AT 1 Reset/write 0/read/verify C00 error register no. 2.
114* (DDRPH) (DFRPH)	CHECK THAT C07 ERROR REGISTER NO. 2 FAILSE IS NOT STUCK AT 1 Reset/write 0/read/verify C07 error register no. 2.
114 (DDRPK) (DFRPK)	CHECK THAT C07 ERROR REGISTER NO. 2 ABS IS NOT STUCK AT 1 Reset/write 0/read/verify C07 error register no. 2.
115	VERIFY OPERATION ERROR REGISTER NO. 2 AND ITS MULTIPLEXERS This test merely verifies that the register flip-flops and their associated multiplexers go through their various transitions properly. The following subtests are performed. 1. Reset/write 1s/read/verify 2. Reset/write 1s/write 0s/read/verify 3. Reset/write 0s/write 1s/read/verify 4. Reset/write 1s/reset/read/verify
116	SINGULARITY OF ERROR REGISTER NO. 2 BITS (FLOAT 1S) Float a 1 down the register to guarantee that no bits interfere with one another. Before each 1 is loaded, the register is cleared.
117	SINGULARITY OF ERROR REGISTER NO. 2 BITS (FLOAT 0S) Float a 0 down the register to guarantee that no bits interfere with one another. Before each bit is loaded, the register is reset to 1s.
120	ERROR REGISTER NO. 2/ERROR REGISTER NO. 3 MULTIPLEXER TEST This test ensures that the multiplexers for the error register no. 2 (register 14) and error register no. 3 (register 15) are not passing data regardless of their input control line states. The following algorithm is used. 1. Write 1s to register 14 2. Write 1s to register 15 3. Read a nonexistent register and make sure 0s are read back (register 37 is used).
121	DATA TEST OF THE IMPLEMENTED BITS OF OFFSET REGISTER The implemented bits are C00-C07 and C10-C12. Although C15 is also implemented, it is read-only and indeterminate and will be ignored during this sequence of tests. The following four subtests are used to data-test this register. 1. Reset/write 1s/read/verify 2. Reset/write 1s/write 0s/read/verify 3. Reset/write 0s/write 1s/read/verify 4. Reset/write 1s/reset/read/verify (C00-C07 only) C10-C12 are cleared by reading (reset) and will have to be tested later.
122	SINGULARITY OF OFFSET REGISTER (FLOAT 1S) Float a 1 down the implemented bits of offset register to guarantee that no bits interfere with one another. Before the 1s are loaded, the register is cleared.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
123	SINGULARITY OF OFFSET REGISTER (FLOAT 0S) Float a 0 down the implemented bits of offset register to ensure that no bits interfere with one another. Before each floating pattern is loaded, the register is set to 1s.
124	TEST MULTIPLEXERS ASSOCIATED WITH OFFSET REGISTER This verifies that the multiplexers associated are not passing data when not enabled. The algorithm is as follows. 1. Load offset register with 1s 2. Read a nonexistent register (register 37) and ensure that the register is returning all 0s.
125	DATA TEST IMPLEMENTED BITS OF DESIRED CYLINDER ADDRESS REGISTER The following three subtests do basic data tests of the desired cylinder address register implemented bits. 1. Reset/write 1s/read/verify C00-C09 2. Reset/write 1s/write 0s/read/verify C00-C09 3. Reset/write 0s/write 1s/read/verify C00-C09
126	SINGULARITY OF DESIRED CYLINDER ADDRESS REGISTER BITS (FLOAT 1S) Float a 1 down the register's implemented bits to guarantee that no bits interfere with one another. Before each 1 is loaded, the register is cleared.
127	SINGULARITY OF DESIRED CYLINDER ADDRESS REGISTER BITS (FLOAT 0S) Float a 0 down the implemented bits of desired cylinder address register to guarantee that no bits interfere with one another.
130	A DESIRED CYLINDER REGISTER MULTIPLEXER TEST This test verifies that the multiplexer for desired cylinder address register is not ignoring its inputs and arbitrarily gating some desired cylinder address register bits onto the control bus. The following algorithm is used. 1. Write 1s to C00-C09 of desired cylinder address register 2. Read a nonexistent register (register 37) and ensure that 0s come back.
131	DATA-TEST THE IMPLEMENTED BITS OF DESIRED SECTOR/TRACK ADDRESS REGISTER The implemented bits are C00-C04 and C08-C12. The following subtests are used to check this register and its associated multiplexers. 1. Reset/write 1s/read/verify 2. Reset/write 1s/write 0s/read/verify The clearing and incrementing of this register will be verified in another test.
132	SINGULARITY OF DESIRED SECTOR/TRACK ADDRESS REGISTER BITS (FLOAT 1S) Float a 1 down the implemented bits of desired sector/track address register to guarantee that no bits interfere with one another. Before the 1s are loaded, the register is cleared.
133	SINGULARITY OF DESIRED SECTOR/TRACK ADDRESS BITS (FLOAT 0S) Float a 0 down the implemented bits of desired sector/track address to guarantee that no bits interfere with one another. Before each pattern is written, the register is loaded with 1s.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
134	TEST MULTIPLEXERS ASSOCIATED WITH DESIRED SECTOR/TRACK ADDRESS REGISTER This test verifies that the multiplexers associated with desired sector/track address register are not passing data when not enabled. The algorithm is as follows. 1. Load desired sector/track address register with 1s. 2. Read a nonexistent register (register 37) and ensure that all 0s are returned.
135	VERIFY THAT ERROR REGISTER NO. 1 C15 IS NOT STUCK AT 1 Reset/write 0s/read/verify that a persistent data check error does not exist.
136	VERIFY THAT C14 ERROR REGISTER NO. 1 IS NOT STUCK AT 1 Reset/write 0s/read/verify that a persistent unsafe condition does not exist.
137	VERIFY THAT C13 ERROR REGISTER NO. 1 IS NOT STUCK AT 1 Reset/write 0s/read/verify that a persistent OPI does not exist.
140	VERIFY THAT C12 ERROR REGISTER NO. 1 IS NOT STUCK AT 1 Reset/write 0s/read/verify that a persistent DTE error does not exist.
141	VERIFY THAT C11 ERROR REGISTER NO. 1 IS NOT STUCK AT 1 Reset/write 0s/read/verify that a persistent WLE does not exist.
142	VERIFY THAT C10 ERROR REGISTER NO. 1 IS NOT STUCK AT 1 Reset/write 0s/read/verify that a persistent IAE error does not exist.
143	VERIFY THAT C09 ERROR REGISTER NO. 1 IS NOT STUCK AT 1 Reset/write 0s/read/verify that a persistent AOE does not exist. The desired address registers are set to 0 in order to localize the failure to a smaller number of modules.
144	VERIFY THAT C08 ERROR REGISTER NO. 1 IS NOT STUCK AT 1 Reset/write 0s/read/verify that a persistent HCRC error does not exist.
145	VERIFY THAT C07 ERROR REGISTER NO. 1 IS NOT STUCK AT 1 Reset/write 0s/read/verify that a persistent header compare error does not exist.
146	VERIFY THAT C06 ERROR REGISTER NO. 1 IS NOT STUCK AT 1 Reset/write 0/read/verify that a persistent hard ECC error does not exist.
147	VERIFY THAT C05 ERROR REGISTER NO. 1 IS NOT STUCK AT 1 Reset/write 0s/read/verify that a persistent write clock fail error does not exist.
150	VERIFY THAT C04 ERROR REGISTER NO. 1 IS NOT STUCK AT 1 Reset/write 0s/read/verify that a persistent format error does not exist.
151	VERIFY THAT C02 ERROR REGISTER NO. 1 IS NOT STUCK AT 1 Reset/write 0s/read/verify that a persistent register modify refused error condition does not exist.
152	VERIFY THAT C00 ERROR REGISTER NO. 1 IS NOT STUCK AT 1 1. Write 0s to the control register (a NOP) 2. Reset/write 0s/read/verify that a persistent ILF condition does not exist.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
153	DATA TEST OF FLIP-FLOPS AND MULTIPLEXERS OF ERROR REGISTER NO. 1 The flip-flops and multiplexers of error register no. 1 are data-tested using the following four subtests. 1. Reset/write 1s/read/verify 2. Reset/write 1s/write 0s/read/verify 3. Reset/write 0s/write 1s/read/verify 4. Reset/write 1s/reset/read/verify
154	SINGULARITY OF ERROR REGISTER NO. 1 BITS (FLOAT 1S) Float a 1 down the register to guarantee that no bits interfere with one another. Before each 1 is loaded, the register is cleared by writing 0s into it.
155	SINGULARITY OF ERROR REGISTER NO. 1 BITS (FLOAT 0S) Float a 0 down the register to guarantee that no bits interfere with one another. Before each floating pattern is loaded, the register is preset to 1s.
156	A TEST OF THE MULTIPLEXER FOR ERROR REGISTER NO. 1 AND THE MAINTENANCE REGISTER Test that the multiplexer for the two registers is not ignoring its controls and passing data onto the control bus. The following algorithm is used. 1. Reset 2. Write 1s to low-order 4 bits of maintenance register (register 03) 3. Write 1s to all bits of error register no. 1 4. Read a nonexistent register (register 37). It should return 0s
157*	TEST THAT A PERSISTENT COMPOSITE ERROR DOES NOT EXIST Master reset, then read the status register and ensure that composite error is not set.
160*	VERIFY THAT COMPOSITE ERROR CAN BE DECODED This test verifies that composite error can be read back successfully. It will accept any path at this point in time. Reset/write 1s to ER1, 2, 3/read/verify that C14 status register is set.
161*	VERIFY THAT EACH ERROR REGISTER NO. 1 BIT CAUSES COMPOSITE ERROR This test verifies that the setting of each bit in error register number 1 (register 02) produces a composite error indication in the status register (register 01).
162*	VERIFY THAT EACH ERROR REGISTER NO. 2 BIT CAUSES COMPOSITE ERROR This test verifies that the setting of each bit in error register number 2 (register 14) causes a composite error in the status register (register 01).
163	VERIFY THAT EACH ERROR REGISTER NO. 3 BIT CAUSES COMPOSITE ERROR This test verifies that the setting of each bit in error register number 3 (register 15) causes a composite error in the status register (register 01).
164	VERIFY THAT DRY IS COMPLEMENT OF THE GO BIT This test ensures that DRY and GO are complements when GO is reset. Reset/read status register/verify that C07 (DRY) is 1.
164*	TESTING FOR PROPER HEAD LOAD OPERATION This test verifies that the head load sequence completes and also that the drive status is correct at completion of the head load sequence.
166*	TESTING FOR PROPER HEAD UNLOAD OPERATION This test verifies that the head unload sequence will operate correctly and error free.
167	VV AND PACK ACK OPERATIONAL TESTS This is a series of four subtests designed to ensure that the volume valid status bit is working in conjunction

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
	with the pack acknowledge command. The subtests are: 1. Issue pack acknowledge with drive cycled up and ensure that VV gets set 2. Verify that issuing pack acknowledge does not cause errors 3. Take drive off-line with VV set and ensure that it stays set 4. With VV set, cycle drive down and up and ensure that VV gets cleared.
170	TEST VV/COMP ERR INTERLOCK This test verifies that VV will not set if a composite error exists when the command is issued. The following sequence is issued: 1. Cycle drive down and up to clear VV 2. Write an error register to force composite error 3. Issue a pack ACK command 4. Read status and control register 5. Ensure that the GO bit is clear.
171	SEE THAT READIN PRESET WILL SET VV This ensures that readin preset sets VV and does not cause composite error in doing so.
172*	TEST STO-GO RESET LOGIC This verifies that the GO and STO-GO reset logic is partially operational. The following sequence is used: 1. Clear previous errors 2. Issue pack ACK to set VV 3. Issue another pack ACK to transfer STO-GO to GO 4. Read the control register and ensure that GO is clear.
173	VERIFY THAT MAINTENANCE MODE DISCONNECTS SECTOR CLOCK This is a basic test to verify that going into maintenance mode actually does something effective. The look ahead register is used here as a tool. The following method is used: 1. Power up the spindle 2. Go into maintenance mode 3. Read look ahead register for reference 4. Read the look ahead register 20 more times to ensure that it does not change. This ensures that the register is disconnected from the drive's index and sector clock.
174	TEST THAT THE GO BIT CAN BE SET This test ensures that the GO bit can be set. The following sequence is used: 1. Issue master reset 2. Put drive in maintenance mode (no sector clocks can occur) 3. Issue a seek command (GO should stay hung because a seek command cannot complete without sector clocks) 4. Read the control register and verify that GO is set.
175	CHECK THAT RHCLR WILL CLEAR HUNG GO BIT A hung GO bit is simulated by issuing a seek while in maintenance mode. The following sequence is used: 1. Issue RHCLR to clear errors 2. Put drive in maintenance mode (stops sector clock from coming) 3. Issue a seek (to set the GO bit) 4. Issue another RHCLR (to reset the GO bit) 5. Read control register and verify that GO is clear.
176	TEST THAT GO AND DRY ARE COMPLEMENTS This test ensures that GO and DRY are complements when GO is set. We set GO by issuing a seek while in maintenance mode.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
177	FLOAT A ONE DOWN C01-C05 OF CONTROL REGISTER Test merely verifies that all bits can be set. Some have already been tested.
200	TESTING OF PIP WITH A SEEK COMMAND This test is designed to check operation of PIP in conjunction with a seek command issued in maintenance mode. Three subtests are required. 1. Read PIP with no command to ensure that it is not stuck at 1. 2. Issue a maintenance mode seek with no GO bit and ensure PIP does not set. 3. Issue a maintenance mode seek with GO bit and ensure PIP does not set. 4. Issue a maintenance mode seek with GO bit and ensure that the PIP gets set.
201	VERIFY THAT READIN CLEARS OFFSET REGISTER This test verifies that the readin preset command clears C10-C12 of the offset register.
202	VERIFY THAT READIN CLEARS DESIRED CYLINDER REGISTER This test verifies that the readin preset command clears C00-C09 of the desired cylinder address register.
203	VERIFY THAT READIN WILL CLEAR DESIRED TRACK/SECTOR REGISTER This test is to verify that issuing a readin preset command will clear C00-C04 and C08-C12 of the desired track/sector address register.
204	A TEST OF THE CONTROL REGISTER MULTIPLEXER This test verifies that the control register multiplexer (RG6) is actually returning 0s in the guaranteed-to-be-0 bit positions which are C06-C10 and C12-C15.
205*	TEST THAT ILLEGAL COMMANDS CAUSE ILF ERROR This test issues the illegal commands to guarantee that the drive recognizes ILF for each one. The GO bit is tested after each operation also to ensure that it got reset.
206	DESIRED SECTOR/TRACK ADDRESS MULTIPLEXER TEST Read desired sector/track address register and verify that the guaranteed 0 bits (C05-C07 and C13-C15) are 0s.
207	LOOK AHEAD REGISTER MULTIPLEXER TEST Read look ahead register and verify that the guaranteed 0 bits (C00-C03 and C11-C15) are 0.
210*	DRIVE TYPE REGISTER GUARANTEED 0S TEST Read the drive type register and verify that the guaranteed 0 bits are correct. They are: C00-C03, C05-C10, C12, C14-C15.
211* (DDRPH) (DFRPH)	DRIVE TYPE REGISTER GUARANTEED 1S TEST Read and verify that C04 and C03 of drive type register are 1.
211 (DDRPK) (DFRPK)	DRIVE TYPE REGISTER GUARANTEED 1S TEST Read and verify that C01, C04 and C13 of drive type register are 1.
212*	DESIRED CYLINDER ADDRESS REGISTER GUARANTEED 0 TEST Read desired cylinder address register and verify that C10-C15 are 0.
213*	CURRENT CYLINDER ADDRESS REGISTER GUARANTEED 0 TEST Read current cylinder address register and guarantee that C10-C15 are 0.
214	A BASIC RESET TEST A test to verify that index will clear the low-order seven flip-flops of the extension counter. The following sequence is used:

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
	1. Put in maintenance mode (20-sector mode) 2. Issue index (a general reset) 3. Issue 127. sector clocks to fill low-order seven bits). 4. Issue index to try and reset the low-order seven flip-flops 5. Issue 127. sector clocks to transfer low-order seven flip-flops to a readable area 6. Read sector counter 7. Verify that fraction is 00.
215	SECTOR CONTROL TEST Verify that the sector counter flip-flops do not count too rapidly using the following sequence: 1. Put in maintenance mode (20-sector mode) 2. Issue index (to clear the counter) 3. Issue 127. sector clocks 4. Read the look ahead register 5. Verify that the sector counter is still at sector 00 extension 00.
216	SECTOR COUNTER TEST Verify that the sector counter is not counting too slowly using the following test sequence: 1. Put in maintenance mode (20-sector mode) 2. Issue index (clears counter) 3. Issue 128. sector clocks (now into first extension) 4. Read look ahead register 5. Verify that the sector counter is at sector 00 extension 20.
217	SECTOR COUNTER TEST Verify that sector counter is counting properly using the following sequence: 1. Put into maintenance mode (20-sector mode) 2. Issue index (clears counter) 3. Issue 255. sector clocks (to end of extension 20) 4. Read sector counter register 5. Verify that the sector counter is at sector 00 extension 20.
220	SECTOR COUNTER TEST Verify that sector counter is counting properly using the following sequence: 1. Go into maintenance mode (20-sector mode) 2. Issue index (clear the register) 3. Issue 256. sector clocks. (to start of extension 40) 4. Read sector counter register 5. Verify that the sector counter is at sector 00 extension 40.
221	A SECTOR COUNTER TEST Verify that the sector counter is counting correctly using the following sequence: 1. Go into maintenance mode (20-sector mode) 2. Issue index (clear register) 3. Issue 511. clocks (to end of extension 40) 4. Read the look ahead register 5. Verify that the sector counter is at sector 00 extension 40.
222	A SECTOR COUNTER TEST Verify that the sector counter is counting properly using the following sequence: 1. Go into maintenance mode (20-sector mode) 2. Issue index 3. Issue 512. sector clocks (to start of extension 60) 4. Read the look ahead register 5. Verify that the sector counter is at sector 00 extension 60.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
223	A SECTOR COUNTER TEST Verify that the sector counter is counting correctly using the following sequence: 1. Go into maintenance mode (22-sector mode) 2. Issue index (to clear the register) 3. Issue 608. sector clocks (to end of extension 60) 4. Read look ahead register 5. Verify that the sector counter is at sector 00 extension 60.
224	A SECTOR COUNTER TEST Verify that the sector counter counts correctly using the following sequence: 1. Go into maintenance mode (22-sector mode) 2. Issue index (clear counter) 3. Issue 609. clocks (steps you out of extension 60) 4. Read the sector counter 5. Verify that the sector counter is at the sector 01 extension 00.
225	A SECTOR COUNTER TEST Verify that the sector counter is counting correctly using the following sequence: 1. Go into maintenance mode (20-sector mode) 2. Issue index (to clear the counter) 3. Issue 671. clocks (step to the end of extension 60) 4. Read the look ahead register 5. Verify that the sector counter is at sector 00 extension 60.
226	A SECTOR COUNTER TEST Verify that the sector counter is counting correctly using the following sequence: 1. Go into maintenance mode (20-sector mode) 2. Issue index (clear counter) 3. Issue 672. sector clocks (to start of sector 01 extension 00) 4. Read look ahead register 5. Verify that the sector counter is at sector 01 extension 00.
227	A SECTOR COUNTER TEST Verify that the low-order nine bits of the sector fraction counter can be cleared with index. This particular subtest will only test the 128. and 256. weighted flip-flops. The following sequence is used: 1. Go into maintenance mode (20-sector mode) 2. Issue index (to clear the counter) 3. Issue 511. sector clocks (sets low-order nine bits) 4. Issue index (should clear) 5. Read the register 6. Verify that the sector counter is at sector 00 extension 00.
230	A SECTOR COUNTER TEST Verify that the low-order nine bits of the sector fraction counter can be cleared by index. This test will pick up the resets of the 1 through 64 weighted bits. The following sequence is used: 1. Go into maintenance mode (20-sector mode) 2. Issue index pulse 3. Issue 511. sector clocks (sets low-order nine bits) 4. Issue second index (should clear) 5. Issue 127. sector clocks (moves invisible bits to where they can be seen) 6. Read sector counter 7. Verify that the sector counter is at sector 00 extension 00.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
231	A SECTOR COUNTER TEST Verify that index can clear all of the low-order ten bits of the sector fraction counter. This test picks up the high-order bit (512. weight). The following sequence is used: 1. Go into maintenance mode (20-sector mode) 2. Issue index (clears counter) 3. Issue 512. clocks (set high-order bit) 4. Issue index (should clear) 5. Read sector counter 6. Verify that the sector counter is 0.
232	A SECTOR COUNTER TEST Verify that the high-order three stages of the sector fraction counter, edge load to 0s properly at sector pulse time. The following sequence is used: 1. Go into maintenance mode (20-sector mode) 2. Issue index (clears sector counter) 3. Issue 672. clocks (takes you to first sector, causing sector pulse. The fraction counter should have edge loaded with 0s) 4. Read the look ahead register 5. Verify that the sector counter is at sector 01 extension 00.
233	A SECTOR COUNTER TEST Verify that the low-order seven flip-flops of the sector fraction counter edge load to 0s at the occurrence of sector pulse. The following sequence is used: 1. Go into maintenance mode (20-sector mode) 2. Issue index (to clear counter) 3. Issue 672. clocks (causes sector pulse which should edge load the low-order seven bits with 0s) 4. Issue 127. clocks (steps the low-order 7-bit magnitude to a visible area in the register) 5. Read the look ahead register 6. Verify that the sector counter is at sector 01 extension 00.
234	A SECTOR COUNTER TEST Verify that the sector counter and overflow decode work in 20-sector mode. The following sequence is used: 1. Go into maintenance mode (20-sector mode) 2. Issue index 3. Issue 12767. sector clocks (to end of second last sector) 4. Read the sector counter 5. Verify that the sector counter is at sector 22 extension 60.
235	A SECTOR COUNTER TEST Verify that the sector counter and overflow decode work correctly in 20-sector mode. The following sequence is used: 1. Go into maintenance mode (20-sector mode) 2. Issue index (clears counter) 3. Issue 12768. sector clocks (to start of last sector) 4. Read the sector counter 5. Verify that the sector counter is at sector 23 extension 00.
236	A SECTOR COUNTER TEST Verify that the sector counter and overflow decode are working correctly in 20-sector mode using the following sequence:

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
	1. Go into maintenance mode (20-sector mode) 2. Issue index (clear counter) 3. Issue 13567. (takes you to invalid sector if decode fails) 4. Read the sector counter 5. Verify that the sector counter is at sector 23 extension 00.
237	A SECTOR COUNTER TEST Verify that the sector counter and overflow decoder work properly in 22-sector mode using the following sequence: 1. Go into maintenance mode (22-sector mode) 2. Issue index (clear the counter) 3. Issue 12788. sector clocks (to end of second last sector) 4. Read the sector counter 5. Verify that the sector counter is at sector 24 extension 60.
240	A SECTOR COUNTER Verify that the sector counter and overflow decode work properly in 22-sector mode using the following sequence: 1. Go into maintenance mode (22-sector mode) 2. Issue index (to clear counter) 3. Issue 12789. sector clocks (to start of last sector) 4. Read the sector counter 5. Verify that the sector counter is at sector 25 extension 00.
241	A SECTOR COUNTER TEST Verify that the sector counter and overflow decode function properly in 22-sector mode using the following sequence: 1. Go into maintenance mode (22-sector mode) 2. Issue index (clear the counter) 3. Issue 13489. sector clocks (into invalid sector if overflow fails) 4. Read the sector counter 5. Verify that the sector counter is at sector 25 extension 00.
242	A SECTOR COUNTER TEST Verifies that the four low-order bits of sector flip-flops clear with index pulse. The following sequence is used: 1. Go into maintenance mode (22-sector mode) 2. Issue index (clear the register) 3. Issue 9135. clocks (set low-order four flip-flops in sector register) 4. Issue index (should clear the register) 5. Read the sector counter 6. Ensure that the sector counter is at sector 00 extension 00.
243	A SECTOR COUNTER TEST Verify that the high-order bit of the sector counter will clear with index. The following sequence is used: 1. Go into maintenance mode (22-sector mode) 2. Issue index (clears counter) 3. Issue 9744. clocks (sets high-order flip-flop) 4. Issue index (should clear flip-flops) 5. Read the sector counter 6. Verify that the sector counter is at sector 00 extension 00.
244	A SECTOR COUNTER MULTIPLEXER TEST This is a test to verify that the look ahead register multiplexer is not ignoring its select lines. The following sequence is used:

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
	1. Go into maintenance mode (22-bit mode) 2. Issue index (clears the register) 3. Issue 9647. sector clocks (gets maximum number of flip-flops set by putting us at sector 17 extension 60) 4. Read a register 37 (a nonexistent register) 5. Verify that 0s are returned (if they are not, assume that sector counter data is being dumped on the control bus).
245*	VERIFY THAT SECTOR CLOCKS COME DOWN FROM DRIVE Under program control, the only thing the diagnostic can tell about sector clock is that it is present or not present. This test checks for its presence using the following sequence. 1. Go into maintenance mode (stops look ahead register). 2. Issue a diagnostic index pulse (clears the look ahead register). 3. Take drive out of maintenance mode with a master clear. This allows sector clock to increment the look ahead register. 4. Read the look ahead register up to a maximum of 200 times (more than enough to see it increment). 5. If no change occurs (look ahead is still sector 00 extension 00) it is safe to assume that it is not incrementing, because either sector clock is not coming down or index coming down from the drive is stuck high.
246*	TEST FOR THE PRESENCE OF INDEX PULSE This test is merely to ensure that index pulse is coming down from the drive. The test is based on the fact that if index pulse is missing, the sector counter will count up and stick at sector 23 (octal). This is the maximum count possible in 20-sector mode. The following sequence is used: 1. Go into 20.-sector mode 2. Read the look ahead register up to 50 times looking for a nonsector 23 value indicating that index pulse has come.
247*	CHECK PIP AND ILF WITH POSITIONING COMMANDS Verify the seek, offset, return to centerline, recalibrate, cause the PIP bit to set and do not cause ILF to be set. The following sequence is used: 1. Go into maintenance mode 2. Set DA = 0 and DCY not equal to CCY 3. Issue a position command 4. Read the control, status, and error register no. 1 5. Test for presence of PIP and that ILF is not set.
250*	VERIFY THAT A LEGAL CYL ADDRESS DECODES CORRECTLY This test verifies that a maintenance mode seek can be issued to every legal cylinder without getting any invalid address errors. The following sequence is used: 1. Issue master clear (clear errors) 2. Go into maintenance mode 3. Load desired address register with 0s 4. Load the desired cylinder register with cylinder number 5. Issue a seek command (strobes the IAE flip-flop) 6. Read control, status, ER1 and DCY 7. Verify that IAE is not set.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
251*	VERIFY THAT ILLEGAL CYLINDER ADDRESS DECODES CORRECTLY Verify that issuing a seek to an invalid cyclinder causes an IAE error. The following sequence is used: 1. Issue master clear (to clear errors) 2. Go into maintenance mode 3. Load DA register with 0s. 4. Load desired cylinder register with xxx (illegal address) 5. Issue a seek command (to strobe IAE flip-flop) 6. Read error register number 1 7. Verify that IAE is set. xxx = 411. for RP04s xxx = 815. for RP06s
252	VERIFY THAT A LEGAL TRACK ADDRESS DECODES CORRECTLY Verify that issuing seeks to each of the legal track addresses does not cause an IAE to occur. The following sequence is used: 1. Issue master clear (to clear errors) 2. Go into maintenance mode 3. Clear desired cylinder address register 4. Load sector = 0 and desired track 5. Issue a seek command (strobe the IAE flip-flop) 6. Read ER1 and verify that IAE is not set.
253	VERIFY THAT AN ILLEGAL TRACK ADDRESS CAUSES IAE Verify that an attempt to seek to track 19. will cause IAE to set. The following sequence is used: 1. Master clear (to clear errors) 2. Go into maintenance mode 3. Write 0s to desired cylinder register 4. Write 0 to desired sector and set up for track 29. 5. Issue a seek command 6. Read ER1 7. Verify that IAE is set.
254	VERIFY THAT LEGAL SECTORS DO NOT CAUSE IAES Verify (in 20-sector mode) that issuing seeks to the legal sectors does not cause IAE to set. The following sequence is used: 1. Master reset (to clear errors) 2. Go into maintenance mode 3. Go into 20-sector mode 4. Load 0s to desired cylinder register 5. Load DA register with track = 0 current sector 6. Issue a seek command (strobe the IAE flip-flop) 7. Read ER1 8. Verify that IAE is not set.
255	VERIFY THAT ILLEGAL SECTOR CAUSES IAE Verify (in 20-sector mode) that issuing a seek to sector 20. causes IAE to be set. The following sequence is used: 1. Issue master clear (clears errors) 2. Go into maintenance mode 3. Go into 20-sector mode 4. Write 0s to desired cylinder address register 5. Write DA register with track = 0 and sector = 20. 6. Issue a seek command (strobe IAE flip-flop) 7. Read ER1 8. Verify that IAE is set.
256	VERIFY THAT LEGAL SECTORS DO NOT CHANGE IAE Verify (in 22-sector mode) that seeking to legal sectors does not cause IAE. The following sequence is used:

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
	1. Master clear (clear errors) 2. Go into 22-sector mode 3. Go into maintenance mode 4. Write 0s to desired cylinder register 5. Write DA register with track = 0 current sector 6. Issue a seek (strobe the IAE flip-flop) 7. Read ER1 8. Verify that IAF is not detected.
257	VERIFY THAT ILLEGAL SECTOR CAUSES IAE Verify (in 22-sector mode) that issuing a seek to sector 22. causes IAE to occur. The following sequence is used: 1. Issue master reset (to clear errors) 2. Go into 22-sector mode 3. Go into maintenance mode 4. Write 0s to desired cylinder address register 5. Write track = 0 and sector = 22. to DA register 6. Issue a seek command (strobe the IAE flip-flop) 7. Read error register no. 1 8. Verify that IAE is set.
260	SEEK CONTROL WITH IAE selected This test verifies that signal SS4 BAD ADR L does the following: 1. Causes SEEK GO CLEAR to clear the GO bit 2. Inhibits the setting of SS0 SEEK GO. The following test sequence is used: 1. Issue master clear (to clear errors) 2. Go into maintenance mode and set CCY to 0 3. Load DCY with illegal cylinder (xxx) 4. Issue a seek 5. Read ER1 and control register 6. Issue RHCLR 7. Read DCY and CCY 8. Verify that go is 0 and that CCY is not xxx. xxx = 412. for RP04s xxx = 815. for RP06s
261*	CURRENT CYLINDER ADDRESS REGISTER TEST Data-test the current cylinder address register by transferring various patterns into it from the desired address register. The following test sequence is used: 1. Issue master clear (clears errors) 2. Go into maintenance mode 3. Load pattern into desired cylinder address register 4. Issue a seek command 5. Issue master clear (terminates seek transfers DCY to CCY) 6. Read control register, DCY, CCY 7. Verify that DCY copied to CCY correctly.
262*	VERIFY THAT SUBTRACTOR WORKS CORRECTLY FOR CCY = DCY This test verifies that the subtractor does not come up with a difference when CCY = DCY (done for all cylinders). The following test sequence is used: 1. Issue master clear 2. Go into maintenance mode 3. Load DCY with cylinder 4. Issue a seek command 5. Issue master clear (terminate seek transfer DCY to CCY) 6. Go back into maintenance mode (DCY now equals CCY) 7. Issue a seek command 8. Read the control register 9. Verify that the GO bit is clear. It should be clear because the difference should be 0. Issuing a seek when DCY = CCY should inhibit the seek from occuring and generate SS0 SK GO CLR H (SS0) to clear GO.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
263* (DDRPH) (DFRPH)	VERIFY ACTION OF THE DISABLE SWITCH This test merely verifies that the disable switch does what it is supposed to do: 1. Cause MOL to go away 2. Cause disable error to occur.
263 (DDRPK) (DFRPK)	Deleted
264*	A FULL-SPEED RECALIBRATE TEST This test issues a recalibrate, waits for completion, tests for errors following completion. The recalibrate can only be tested at full speed (functionally).
265*	TEST RESET OF CURRENT CYLINDER ADDRESS REGISTER This test verifies that all flip-flops in the CCY register will reset with a recalibrate pulse. The following sequence is used [two patterns are required (377 and xxx octal)]: 1. Issue master reset (to clear errors) 2. Go into maintenance mode 3. Load pattern into desired cylinder address register 4. Issue a seek command 5. Issue master clear (to transfer DCY to CCY) 6. Issue a recalibrate (should clear CCY) 7. Read current cylinder register 8. It should be 0. xxx = 400 for RP04s xxx = 1400 for RP06s
266	TEST SUBTRACTOR LOGIC WITH A SERIES OF PATTERNS This test applies a set of patterns to CCY and DCY and looks for the fact that a difference was detected. The patterns come from a look-up table. The patterns chosen will catch a very high percentage of subtractor faults.
267*	BASIC SEEK TEST This test merely issues a 1-cylinder seek at full speed and tests for the following situations: 1. Drive does not hang 2. No errors are caused 3. DCY = CCY = 1.
270* (DDRPH) (DFRPH)	TEST OF SKI LOGIC This test verifies that the seek incomplete logic is capable of detecting such an error condition. The following sequence is used to test the SKI logic: 1. Clear errors 2. Recalibrate (positioner to cylinder 0) 3. Seek to cylinder 1 (positioner to cylinder 1) 4. Load 400 into the CCY (hardware now out of synch) 5. Issue a seek to cylinder 1. This should cause an SKI error. The positioner is already over cylinder 1 but it thinks a large reverse seek must be done because CCY is sitting at 400. 6. Read error register and CCY register. 7. Verify that SKI has occurred. 8. Verify that CCY is now clear. This is true because SKI causes the drive to initiate an automatic recalibrate pulse which causes a DRIVE TO INNER GUARD BAND to occur and reset CCY if the sequence works correctly.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
270 (DDRPK) (DFRPK)	TEST OF SKI LOGIC This test verifies that the seek incomplete logic is capable of detecting such an error condition. The following sequence is used to test the SKI logic: 1. Clear errors 2. Recalibrate (positioner to cylinder 0) 3. Seek to cylinder 1 (positioner to cylinder 1) 4. Load 400 into the CCY (hardware now out of synch) 5. Issue a seek to cylinder 1 This should cause an SKI error. The positioner is already over cylinder 1 but it thinks a 4-cylinder reverse seek must be done because CCY is sitting at 400. 6. Read error register and CCY register. 7. Verify that SKI has occurred. 8. Verify that CCY is now clear. This is true because SKI causes the drive to initiate an automatic recalibrate pulse which causes a RESET REGISTERS to occur and reset CCY if the sequence works correctly.
271*	A TEST OF SUBTRACTOR DIFFERENCE LOGIC This set of two subtests causes seeks with all possible positive and negative subtactor differences. These are functional tests designed to detect but not diagnose a class of seek control logic faults. Since the output of the difference logic is not visible, this test has to be very functional. If we slip position during the test sequence, the error will accumulate and at some point in the sequence, force the positioner to one of the guard bands causing a SKI to occur. Unfortunately, the only thing that can be said about a fault detected in this test is that a positioner malfunction exists. The following test sequence is used: Subtest 1 1. N = 1 2. RHCLR 3. Recalibrate 4. Seek to cyl-n and test for error 5. Seek to cyl-0 and test for error 6. N = N + 1 7. Done if N = xxx. If not done go to step 4. Subtest 2 1. N = yyy 2. RHCLR 3. Recalibrate 4. Seek to cyl-xxx 5. Seek to cyl-n and test for error 6. Seek to cyl-xxx and test for error 7. N = N - 1 8. Done if (N < 0) otherwise go to step 5. Due to the way in which this test could fail, the standard scope loop technique is not used. Upon detecting an error, this test will report, clear and remember the fact that an error was committed. After cycling through the entire test, it will be repeated if loop on error has been selected and an error was detected somewhere along the way. xxx = 410. for RP04s xxx = 814. for RP06s yyy = 409. for RP04s yyy = 813. for RP06s

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
272*	VERIFY THAT A NO-OP COMMAND IS OPERATIONAL Issue a NO-OP command in maintenance mode and ensure that the GO bit is not set. This gives some confidence that the command was not decoded incorrectly.
273*	VERIFY OPERATION OF OPI ERROR LOGIC The following sequence is used to verify part of the OPI error detection logic: 1. Issue master clear (to clear errors) 2. Issue recalibrate (synchronize positioner) 3. Go into maintenance mode 4. Issue a search mode for surface = 0 sector = 0 5. Issue index 6. Issue index 7. Verify that OPI is not set too early 8. Issue index 9. Verify that OPI did set.
274	VERIFY THAT RMR CAN OCCUR Cause an RMR condition and verify that the drive responds correctly using the following sequence: 1. Master clear 2. Issue a recalibrate 3. Go into maintenance mode (allows seek to hang) 4. Seek to cylinder 1 (sets GO bit) 5. Write 0 to DCY (should cause RMR) 6. Read R1 and DCY 7. Verify that RMR occurred 8. Verify that DCY was not loaded.
275*	VERIFY THAT WRITING ATA DOES NOT CAUSE RMR Write the attention summary register with the GO bit set and verify that RMR does not occur. The sequence is as follows: 1. Master clear 2. Recalibrate 3. Go into maintenance mode (allows seek to hang) 4. Seek to cylinder 1 (sets GO bit) 5. Write the ATA register with 0s 6. Read ER1 7. Verify that RMR did not set.
276*	BASIC SEARCH FUNCTION IN MAINTENANCE MODE This test is a basic verification of the search command decode. The following test sequence is used: 1. Master clear, (to clear errors) 2. Recalibrate (causes CCY = 0) 3. Set DA register for cylinder = 0 4. Go into maintenance mode (no sector pulse freezes command) 5. Issue a search 6. Read control, status and error registers. Test for correct status: GO bit = 1 PIP = 0 Composite error = 0.
277	OPERATION WHILE SEARCHING FOR ILLEGAL SECTOR This test issues a search for an illegal sector. The status of the drive is then checked to verify that performance was correct under this set of conditions. The search should terminate immediately because of IAE and the GO bit should reset via the class B error reset condition. The following test sequence is used: 1. Clear errors 2. Recalibrate (sets CCY = 0) 3. Set DCY = 0 4. Set DA register to track = 0 sector = 30 5. Go into maintenance mode 6. Issue a search command 7. Read device registers and verify status.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
300*	SEARCH FOR SECTOR 0 This is the first full-speed search command. A functional search for sector = 0, cylinder = 0, surface = 0. The following sequence is used: 1. RHCLR 2. Recalibrate 3. Set DA = DCY = 0 4. Search 5. Wait until done 6. Read device registers.
301	SEARCH COMBINED WITH IMPLIED SEEK Verify that search causes implied seek by looking at the current cylinder address register when completed. Except for the implied seek to cylinder number 1, this test is identical to the previous test.
302	VERIFY OPERATION OF THE SECTOR COMPARATOR This test cannot run in user mode because system scheduling will cause it to fail. Search for all possible sectors in 22-sector mode. This test searches for all sectors from every other sector on the disk. The following algorithm is used: 1. RHCLR 2. Recalibrate 3. Synch = 0 (a synch point to start search from) 4. .SECT = 0 (sector = 0, track = 0) 5. RHCLR (clear errors) 6. DCY = 0 (clear desired cylinder address) 7. DA register loaded with .SECT 8. Keep reading look ahead register until sector is found (synch) 9. Now in synch. Issue a search command. 10. Wait until done. Then read the look ahead register. If error, report and loop to step 5. 11. Verify that the look ahead register = .SECT. If error, report and loop to step 5. 12. .SECT = .SECT + 1 If .SECT $\leq$22., go to step 5. 13. Synch = synch + 1. Advance to next synch point. If synch $\leq$22. go to step 4 (next permutation). 14. Done with test (approximately 484 searches).
303*	TEST OPERATION OF WRITE LOCK SWITCH Verify with the status register and operator intervention that the write lock switch works correctly.
304*	TEST OPERATION OF UNLOAD/STAND-BY Verify through operator intervention and status information that the unload command works correctly.
305*	ISSUE A 0 MICROINCH OFFSET A full-speed offset operation to test for a hung device or a composite error caused by offsetting. The test sequence is as follows: 1. Master clear 2. Recalibrate 3. DCY = 0 4. DA = 0 5. OFFSET = 0 6. Issue an offset command 7. Wait until done and test for errors.
306*	FULL SPEED RETURN TO CENTERLINE TEST Test to verify that the return to centerline command does not cause a composite error or hung device. The sequence is as follows: 1. Master clear 2. Recalibrate 3. DCY = DA = OFFSET = 0 4. Issue (two) return to centerline commands. The first will take drive out of offset mode if necessary. 5. Wait until done and verify that device did not hang and composite error did not set.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
307*	ISSUE THE POSSIBLE OFFSETS Issue the possible offsets and return to centerline commands at full speed to check for hung device or composite error. The test sequence is as follows: 1. Master clear 2. Recalibrate 3. DCY = DA = 0 4. Load offset register with a value 5. Issue an offset command 6. Verify that no error occurred. If there is an error, loop back to step 1 after reporting. 7. Issue a return to centerline command. 8. Verify that no error occurred. If there is an error, loop back to step 1 after reporting. 9. Update offset value from table. If not finished, go back to step 1.
310*	ISSUING A SEEK WHILE IN OFFSET MODE If the drive is in offset mode when a seek is requested, it is supposed to postpone the seek, issue an automatic return to centerline, and then proceed with the seek. The logic to do this is on (SS0). There is no way to guarantee that this operation works correctly, but this test does the operation looking for hung devices or composite error being caused by this operation. The following test sequence is used: 1. Master clear 2. Recalibrate 3. DCY = 0 = DA 4. Offset = xxx microinches 5. Issue an offset command 6. Seek to cylinder number 1 7. Wait until done 8. Read device registers 9. Verify that a. Device did not hang b. Composite error did not set c. Seek took place (CCY = 1). xxx = 800 for RP04s xxx = 600 for RP06s
311*	VERIFY THAT ATA IS NOT STUCK SET Verify that the ATA flip-flop is not stuck at 1. The following test sequence is used: 1. Issue master clear (should reset ATA) 2. Write ATA register with 377 (should reset ATA) 3. Read the status register 4. Verify that ATA is clear.
312	VERIFY THAT ATA CAN BE SET The following test sequence is used: 1. Issue master clear 2. Write the ATA register with 377 3. Issue recalibrate (causes ATA) 4. Read the status register 5. Verify that ATA is set.
313	VERIFY THAT RHCLR WILL RESET ATA 1. Issue master clear (clear ATA) 2. Write the ATA register with 377 (clear ATA) 3. Issue recalibrate (sets ATA) 4. Issue master clear (RHCLR to clear ATA) 5. Read status register 6. Verify that ATA is clear.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
314	VERIFY RESITTING OF ATA BY WRITING REG-04 The following test sequence is used: 1. Master clear (clears ATA) 2. Issue recalibrate to set ATA 3. Write ATA with proper bit for drive being tested 4. Read the status register 5. Verify that ATA is clear.
315	VERIFY OPERATION OF PSEUDO-ATA POSITION It is already verified that writing ATA register with 377 will reset the ATA flip-flop. This is a test to verify the position decode. The idea is to cause an attention and then write all bits of the ATA register except the drive asserting attention. If position decoder is working, ATA will still be set. The following sequence is used: 1. Master clear 2. Recalibrate (causes ATA) 3. Write ATA bits for drives "not" under test 4. Read status register for drive being tested 5. Verify that ATA has not cleared.
316*	BASIC ATA REGISTER READ CYCLE The following sequence is used: 1. Master clear 2. Write ATA with 377 (clears all ATAs in all drives) 3. Read the status registers for all drives into cores 4. Read ATA summary register 5. Verify that ATA summary register is 0.
317	TEST FOR PROPER PSEUDO-ATA POSITION This test verifies that when the test drive asserts ATA, it comes back in one and only one position in the ATA register and also that the position is correct. The following test sequence is used: 1. Master reset 2. Recalibrate (to set ATA) 3. Read ATA register 4. Verify that it is correct: a. ATA is asserted for this drive in proper position b. No other ATAs asserted (multiple position decode).
320	VERIFY THAT SK GO CLR SETS ATA Verify that the completion of a seek sets ATA. The following sequence is used: 1. Master reset (clears errors) 2. Recalibrate 3. Master reset (resets ATA) 4. Seek to cylinder 0 (causes ATA) 5. Read status register 6. Verify that ATA is set.
321	VERIFY THAT SRCH COMPL WILL SET ATA Verify that the completion of a search will set ATA using the following sequence: 1. Master reset (clear errors) 2. Recalibrate 3. Master reset (clears ATA) 4. Issue a search for cylinder 0, sector 0, surface 0 5. Read status register 6. Verify that ATA is set.
322	VERIFY THAT SETTING GO WHILE ERR CAUSES ATA Verify that attempting to set the GO bit while the drive has a composite error causes ATA to be set. The following sequence is used:

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
	1. Master reset (clears errors) 2. Recalibrate 3. Write a DCK error into ER1 4. Write 377 to ATA register to clear ATA 5. Issue a NO-OP with the GO bit set (should set ATA) 6. Read status register and verify that ATA was set.
323	VERIFY THAT ATA CAN BE CLEARED WHILE ERROR PERSISTS The following test sequence is used: 1. Master reset (clears errors) 2. Recalibrate (sets ATA) 3. Write a DCK error to error register no. 1 4. Write the ATA register to reset ATA flip-flop 5. Read status register 6. Verify that ATA is cleared.
324	ATA BEHAVIOR FOR CYCLE ON/OFF, UP/DOWN This test comprises 3 subtests designed to check the response of the ATA logic to various cycle states of the drive. Subtest A 1. Ensure that device is on-line 2. Recalibrate 3. Issue an unload (uses OFFCHK routine) 4. Read status and verify that ATA was not set. Subtest B (Exec Mode Only) 1. Power the drive off, then on, but do not load heads 2. Read status register 3. Verify that ATA was set by power-up 4. Verify that AC-low is asserted in ER3 5. Issue a drive clear (uses Massbus clear) 6. Verify that AC-low is cleared. Subtest C (Exec Mode Only) 1. Verify that the heads are at an unloaded state (via OFFCHK) 2. Get heads loaded (via ONCHK) 3. Read status register 4. Verify that the transition of MOL has set ATA.
325	DATA TRANSFER COMMAND DECODE CHECKS This test is designed to verify that the data transfer commands are decoded reasonably in the drive. This test sequence is used for each for the six data transfer commands available: 1. RHCLR (to clear errors) 2. Recalibrate 3. DA = DCY = 0 (a valid address) 4. Go into maintenance mode (20-sector mode) 5. Issue a data transfer command 6. Snapshot appropriate drive registers 7. Abort the transfer 8. RHCLR 9. Verify that GO = 1 PIP = 0 ILF = 0 10. Repeat test for all data transfer commands.
326	TEST OF DATA BUS CONTROL SIGNALS This test issues a controlled sequence of events to check: 1. OPI logic 2. Run line and command decode for write header and data 3. Occupied line (RH10 only) 4. Exception logic 5. EBL logic. The test is generally structured as follows (10 subtests):

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
	1. Start a maintenance mode write data command 2. Issue index pulses to verify OPI logic. When OPI works correctly, the command is decoded correctly and run must have been asserted by the controller. 3. When the OPI condition occurs, a class B error is caused which will force the transfer to shut down. By observing the states of the DA register, done, and exception, the diagnostic is able to isolate control line faults to the proper modules.
327*	VERIFY OPERATION OF WRT HD + DAT UP TO DATA FIELD This is a series of eight subtests that verify operation of the write header and data command from the preheader gap to the start of the data field. The test completes at the start of the data field in order to minimize the size of the scope loop. The subtests consist of these procedures. 1. Check 39 bytes of 0s in the sector gap. 2. Check for proper sync byte. 3. Check that the eight bytes of header and key words are 0 since the transfer is to cylinder 0, surface 0, sector 0. 4. Check that header CRC bytes are correct = 0. 5. Check that the 11 bytes of 0s in the head gap are correct. 6. Check that the post header synch byte is correct. 7. Verify that the data envelope is not up too soon. 8. Verify that the data envelope is up on time.
330*	CHECK FOR STUCK 1S AND 0S ON DATA BUS Issue a maintenance mode write header and data command and check that the data bits on the Massbus going from controller to drive are not stuck at 1s or 0s. 1. Start the transfer. 2. Issue 496 shift clocks to step to data area. 3. Clock first 18 bits out and verify that they are 0. 4. Clock second 18 bits out and verify that they are 1s.
331	CHECK OF WRITE DATA PATH USING FLOATING PATTERNS Verify that there are no bits cross-coupled on write data bus by testing data integrity with floating patterns. 1. Start a maintenance mode transfer of 1 sector. 2. Issue 962 clocks to step to data area. 3. Issue 36 clocks to step over first PDP-10 word. 4. Start clocking PDP-10 half words out of memory (floating patterns), each time verifying that data was correct. 5. Repeat step 4 until 18 PDP-10 words have been tested. 6. Abort the transfer.
332*	TRANSFER STANDARD DATA PATTERNS CHECKING FOR PARITY This test passes 52 (18-bit) data patterns across the Massbus and checks for parity errors after each transfer.
333	VERIFY THAT DRIVE CAN DETECT INCORRECT PARITY This test does a maintenance mode write of 18 bits of 0s across the data bus with incorrect parity (PAR = 0) to verify that the drive can detect incorrect parity.
334	VERIFY SECTOR TIMING OF DATA ENV, ECC ENV, EBL Test clocks through an entire sector in maintenance mode to verify that the above signals transition at exactly the right time. Eight subtests do the job as follows.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
	Clocks DATA ENV ECC ENV DONE 1. 495. 0 0 0 2. 1. 1 0 0 3. 4607. 1 0 0 4. 1. 0 1 0 5. 31. 0 1 0 6. 1. 0 0 0 7. 15. 0 0 0 8. 1. 0 0 1 Testing around transition points optimizes test execution time but the diagnostic can not tell exactly how much timing is incorrect. This would make the test unreasonably long.
335	CAUSE A DRIVE TIMING ERROR (DTE) This test verifies that the drive can detect a drive timing error. The error is forced in the following way. 1. Start a 1-section maintenance mode, write header and data command. 2. Cause a sector pulse (which causes a drive timing error). 3. Read ER1 and verify that DTE is set.
336	CHECK HEADER LOGIC WHILE SUPPLYING CORRECT DATA Start a 1-sector maintenance mode read header and data operation. Clock through the header area supplying correct data and verify that: 1. Supply 312 0s for preheader gap 2. Supply 8-bit synch char 31 (octal) 3. Supply 80 0s for header field 4. Supply 88 0s for post-head gap 5. Supply 8-bit synch char 31 (octal) 6. Verify that hardware status is correct.
337	TEST HEADER CRC LOGIC Issues a maintenance mode read operation and supplies wrong CRC data. This should cause HCRC to be detected. 1. Send 312 0s for header gap 2. Send 8-bit sync byte 3. Send 64 0s for header and key words 4. Send 8 0s for first half of header CRC 5. Send 8 1 bits (wrong data) for rest of header CRC 6. Send 8 0s to move slightly into gap 7. Read status and verify that HCRC is set.
340	TEST HEADER COMPARE LOGIC Verify that the drive can detect header data errors. This is done by supplying the header data in a maintenance mode read header and data operation. The diagnostic supplies the wrong cylinder information. This will result in a HCE and a HCRC error. Here is the test outline: 1. Send 312 0s for preheader gap 2. Send 8-bit synch byte 3. Send 8 bits of 1s for wrong cylinder information 4. Send 72 bits of 0s for rest of header data and header CRC word. 5. Verify that HCE was detected. The HCRC error is incidental to this test.
341	TEST OF FER ERROR DETECTION LOGIC Verify that a format error can be detected. Start a maintenance mode read header and data operation in 20-sector mode, then supply the header data with the 22-sector FMT bit set. This should cause a format error (FER). HCRC and HCE are incidental to this test. Test outline: 1. Send 312 0s for preheader gap 2. Send 8-bit synch byte 3. Cause error in next two bytes

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
	a. 12 bits of 0s b. 1 bit of 1 for 11-format c. 3 bits of 0s to fill out last byte 4. Send 64 bits of 0s to complete header area 5. Verify that FER has occurred. HCE and HCRC may also appear but are insignificant in this test.
342	TEST OPERATION OF HCI LOGIC Test initiates a read header and data command in maintenance mode (20-sector mode) with the header-compare inhibit bit (HCI) set. The data supplied by the diagnostic supplies incorrect header, format, CRC information. After completing transfer through the header area, status is checked. HCE, FER, HCRC should all be 0. The test outline: 1. Send 312 0s for preheader area 2. Send 8-bit synch byte 3. Send 8-bit byte with wrong cylinder (binary = 00000001) 4. Send 8-bit byte with wrong FMT (binary = 00010000) 5. Send 64 0s which completes header area with wrong CRC 6. Verify that HCE = FER = HCRC = 0.
343	VERIFY THAT SYNCH DETECTOR NOT STUCK HIGH Start a maintenance mode read header and data operation. The diagnostic supplies all 0s for data area. After 512 clocks (data of 0s) and data ENV bit is examined. It should be at 0. If it is at a 1, the synch byte detector has erroneously detected a synch byte.
344	VERIFY THE READ DATA PATH INTEGRITY Supply necessary data patterns via a maintenance mode read header and data operation and check data that appears in the RH buffer register against the expected data (18 bits at a time). Data checking is not performed until we step to the data field. Once at the data field, four subtests are performed: 1. 1 Massbus transfer of 0s 2. 1 Massbus transfer of 1s 3. 36 Massbus transfers of floating patterns 4. 8 Massbus transfers of 0s and parity patterns.
345	CHECK DRIVE PARITY NETWORK Send various data patterns from drive to controller over the Massbus data path using a maintenance mode read header and data operation. There are two subtests. 1. Sends 1s and 0s. Possible faults include parity network and parity transmit faults. 2. Supply other data patterns to verify parity generation logic in drive. Faults only include parity generator.
346*	TEST ATA TRANSMITTER FOR STUCK AT 1 Verify that no drive on the bus has an attention transmitter stuck at 1, (constantly asserting ATA). 1. Issue RHCLR 2. Write ATA register with 377 to clear ATA flip-flops 3. Verify that ATA (CONI DATA) is not asserted.
347	TEST THAT DRIVE SENDS ATA LINE TO RH Issue a recalibrate and verify that ATA (CONI DATA) is asserted.
350	VERIFY THAT ECC GENERATION WORKS CORRECTLY In 18-bit mode, issue a write header and data command, issue enough clocks to step up to the ECC field, clock out and read the next 32 bits of ECC data and verify that it is correct.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
351*	FULL-SPEED WRITE HEADER AND DATA TEST IN 18-BIT MODE This is a full-speed write header and data operation to cylinder 0 surface 0 sector 0. This transfer has worked previously in maintenance mode. This test uses disk clocking and the MDLI interface logic for the first time. Because of the variety of faults that could turn up in this test, module callout may not be optimum.
352	VERIFY THAT EACH HEAD CAN BE SELECTED Issue a full-speed write header and data to the first sector (sector 0) on every surface of cylinder 0. This is the first time any head other than 0 has been selected and the hope is that only head-related problems show up (NHS, MHS, WCU, etc.). OPI errors could possibly come up during this test also. This would be an indication of a faulty track comparitor on SS5.
353*	DISK ADDRESS LOGIC (INCREMENTATION) TEST Using full-speed write header and data tests, verify that the various disk address registers can increment properly.
354*	VERIFY THAT LBT DETECTOR WORKS Do a 1-sector write header and data to the last block on the disk (cylinder xxx surface 18 sector 19). At the end of the transfer, verify that LBT has set and that DA = 0 and DCY = yyy. xxx = 410. for RP04s xxx = 814. for RP06s yyy = 411. for RP04s yyy = 815. for RP06s
355*	VERIFY THAT AOE CAN BE DETECTED Do a full-speed write header and data operation to the last disk sector (sector 23 surface 20 cylinder xxx). The transfer will be two sectors long, which will overflow the disk and cause AOE. xxx = 632 for RP04s xxx = 1456 for RP06s
356	WRITE/READ HEADERS AT FULL SPEED Write, then read headers and data at full speed using cylinder 0, surface 0, sector 0. This is a functional test, so module callout will be impossible. This is the first time headers and data have been read at full speed. The operation has worked previously in maintenance mode.
357	WRITE THEN READ HEADERS FROM VARIOUS ADDRESSES Write, then read/verify headers from various disk areas. During the read, HCI is set to allow header recovery.
360	WRITE THEN READ HEADERS FROM VARIOUS ADDRESSES This test after writing headers reads them with HCI = 0. The previous test has just checked to see that headers were read and written correctly. This test will look for header errors HCE, FER. Either of these would be an indication that comparitor logic on the SS board is faulty and cannot match headers and/or formats.
361*	FUNCTIONAL READ WRITE TEST Full speed exercise at first disk address: 1. Write headers and data 2. Read data.
362	VERIFY SECTOR TIMING OF DATA ENV, ECC ENV, EBL Test clocks through an entire sector in maintenance mode (16-bit) to verify that the above signals transition at exactly the right time. Eight subtests do the job as follows.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
	Clocks DATA ENV ECC ENV DONE 1. 495. 0 0 0 2. 1. 1 0 0 3. 4095. 1 0 0 4. 1. 0 1 0 5. 31. 0 1 0 6. 1. 0 0 0 7. 15. 0 0 0 8. 1. 0 0 1 Testing around transition points optimizes test execution time but the diagnostic cannot tell exactly how much timing is incorrect. This would make the test unreasonably long.
363 (DDRPH) (DFRPH)	Deleted
363 (DDRPK) (DFRPK)	LOGICAL ADDRESS PLUG TEST This test checks for the proper drive response when the logical address plug is removed and reinserted. The expected response is as follows. With plug removed: 1. Handshake fails. Can no longer access the drive. With plug reinserted: 1. ATA = 1 2. MOL = 1 3. OPE = 1 4. VV = 0
364*	RH10 READIN TEST This test requires operator action (to press KA/KI CPU READIN switch) to test the RH10 readin capabilities. This test can not be performed in user mode. The logic relation to READIN has been tested by previous tests. It is assumed that the RH10 controller diagnostic has been run and the channel is in operating condition. This test will write a read in BOOT block of data on logical block 0 of the selected drive. The operator will then be requested to press READIN on the CPU console. This should cause the drive to initialize and read the data on logical block 0 (cylinder 0, surface 0, sector 0). The first word of data is passed to the channel as a control word indicating a word count and data buffer address. When the channel has received the specified amount of data from the selected drive, it will request one more word to be read from the drive to be used as a program jump word.
DUAL 1	VERIFY THAT PROGRAMMABLE MODE OPERATES CORRECTLY Read and verify that both ports' status registers yield programmable bit. This test also fails if reading the status register happens to capture the port. It should not do so.
DUAL 2	TEST OPERATION OF CAPTURE AND RELEASE ON PORT A The test is designed to do the following. 1. Go into neutral. 2. Seize port A with read of control register. Verify with a read of port B status 3. Wait approximately 1/2 second and verify (by reading port B status) that drive has not yet dropped into neutral because of a 1-shot fault. 4. Wait approximately 1 second more and verify that the drive has gone into neutral (the 1-shot did time out) causing automatic release.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
DUAL 3	TEST OPERATION OF CAPTURE AND RELEASE ON PORT B This test is designed to do the following. 1. Go into neutral. 2. Seize port B with read of control register. Verify with a read of port A status. 3. Wait approximately 1.5 second and verify that the drive has gone into neutral (the 1-shot did time out) causing automatic release.
DUAL 4*	VERIFY ABILITY TO HOLD PORT A INDEFINITELY Repeatedly capture port A for some period of time greater than 1.5 second and verify that the port stays captured for the entire time. The following sequence is used. 1. Go into neutral. 2. Capture port A by reading the control register. 3. Read port B status and verify that drive has port A captured. 4. Repeat steps 2 and 3 for a period of time longer than the 1-shot time (approximately 2 seconds).
DUAL 5*	VERIFY ABILITY TO HOLD PORT B INDEFINITELY Repeatedly capture port B for some period of time greater than 1.5 seconds and verify that the port stays captured for the entire period of time. The following sequence is used. 1. Go into neutral. 2. Capture port B by reading the control register. 3. Read port A status and verify that port B has been captured. 4. Repeat steps 2 and 3 for a period of time longer than the 1-shot time (approximately 2 seconds).
DUAL 6*	VERIFY THAT READING REG-00 ONLY, CAUSES PORT A CAPTURE 1. Go into neutral. 2. Read each of the Massbus registers (excluding control register) and after each read, verify that port A has not been captured by checking status on port B.
DUAL 7*	VERIFY THAT READING REG-00 ONLY, CAUSES PORT B CAPTURE 1. Go into neutral. 2. Read each of the Massbus registers (excluding control register) and after each read, verify that port B has not been captured by checking the status on port A.
DUAL 10*	VERIFY THAT WRITE TO OTHER ATAS DOES NOT CAPTURE PORT Write ATA registers for all drives except the one under test and verify no port has been captured.
DUAL 11	CAPTURE PORT A BY WRITING ITS CONTROL REGISTER Writing any register will capture a port if drive is in neutral, the selection of the control register is arbitrary. 1. Get into neutral. 2. Write control register on port A. 3. Read status from port B and verify that port A was captured.
DUAL 12	CAPTURE PORT B BY WRITING ITS CONTROL REGISTER Writing any register will capture the port if drive is in neutral. The selection of the control register is arbitrary. 1. Get into neutral. 2. Write the control register on port B. 3. Read status from port A and verify that port B was captured.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
DUAL 13	VERIFY THAT RELEASE FROM PORT A DOES NOT SET ATA With no request on port B, a release from port A should not set the ATA flip-flop on port B. 1. Get into neutral with ATAs for both ports clear. 2. Read control register for port A to capture. 3. Wait until we flip back into netural. 4. Verify that ATA has not been set for port B.
DUAL 14	VERIFY THAT RELEASE FROM PORT B DOES NOT SET ATA With no request on port A, a release from port B should not set the ATA flip-flop on port A. 1. Get into neutral with ATAs for both ports clear. 2. Read control register for port B to capture. 3. Wait until we flip back into neutral. 4. Verify that ATA has not been set for port A.
DUAL 15	RELEASE, WITH RELEASE COMMAND, FROM PORT A Verify that we can release from port A with a release command and that issuing the release command does not cause composite error. 1. Go into neutral. 2. Read control register on port A to capture it. 3. Issue a release command. 4. Read status register on port B and verify that release worked correctly and did not cause composite error.
DUAL 16	RELEASE, WITH RELEASE COMMAND, FROM PORT B Verify that we can release from port B with a release command and that issuing the release command does not cause composite error. 1. Go into neutral. 2. Read control register from port B to capture the port. 3. Issue a release command. 4. Read status register on port B and verify that release command worked correctly and did not cause composite error.
DUAL 17	FIRST TEST TO VERIFY REQUEST HOLD LOGIC 1. Go into neutral and reset all ATAs. 2. Capture port B by reading port B control register. 3. Request port A by writing port A control register. 4. Read port A status and verify that it is not captured. Port B should be in possession for 1 second. 5. Wait for greater than a second to give auto-release from port B a chance to happen. 6. Read status from port A and verify that ATA has set (because of the request from step 3).
DUAL 20	SECOND TEST TO VERIFY REQUEST HOLD LOGIC 1. Go into neutral and reset all ATAs. 2. Capture port A by reading port A control register. 3. Request port B by writing port B control register. 4. Read port B status and verify that it is not captured. Port A should be in possession for 1 second. 5. Wait for greater than a second to give auto-release from port A a chance to happen. 6. Read status from port B and verify that ATA has set (because of the request from step 3).
DUAL 21	CHECK AUTO UNLOCK FEATURE FROM PORT B Verify that the test does not hang on a port if it is requested but not used.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
	1. Clear all ATAs and go into neutral. 2. Capture port A by reading the control register. 3. Request port B by reading its control register. 4. Wait approximately 3 seconds. Drive should time out on port A, flip to port B because of the request, timeout on port B and flip back to neutral. 5. Read status and verify that the tests are not hung on one of the ports.
DUAL 22	CHECK AUTO UNLOCK FEATURE FROM PORT A Verify that the test does not hang on a port if it is requested but not used. 1. Clear all ATAs and go into neutral. 2. Capture port B by reading the control register. 3. Request port A by reading its control register. 4. Wait approximately 3 seconds. Drive should timeout on port B, flip to port A because of the request, timeout on port A and flip back to neutral. 5. Read status and verify that the test is not hung on one of the ports.
DUAL 23	VERIFY THAT RHCLR WILL NOT RESET ATA IN NEUTRAL 1. Cause ATA on port A (recalibrate). 2. Cause ATA on port B (recalibrate). 3. Issue RHCLR. 4. Read status from both ports and verify that neither ATA bit has been reset.
ALIGN (DDRPH) (DFRPH)	HEAD ALIGNMENT VERIFICATION TEST This test will operate on all drives under test. The overall head alignment for a drive is verified in the following manner: 1. Recalibrate the positioner 2. Seek to cylinder 245 3. Verify alignment of heads 0 through 18 4. Seek to cylinder 4 5. Verify the alignment of heads 0 and 18 6. Seek to cylinder 400 7. Verify the alignment of heads 0 and 18 8. Seek to cylinder 245 9. Reverify alignment of heads 0 through 18. The following algorithm is used to verify the alignment of a particular head. 1. Offset the positioner to +1200 microinches. 2. Record the value of the sign bit. 3. Move the positioner towards the track centerline in 25 microinch increments until the sign bit changes. Upon detecting a sign change, record the offset value. 4. Offset the positioner to -1200 microinches and repeat steps 2 and 3 above. 5. Average the two values of offset and consider the head to be out of alignment if the average offset has turned out to be in excess of 150 microinches (absolute value) on cylinder 245, 350 microinches on cylinders 4 and 400. NOTES 1. Positive offsets indicate a head position toward the spindle side of the actual track centerline. 2. By default, normal printout mode (SW10 reset), the program will print a complete table of offsets for all heads. In short printout mode (SW10 set), information will be printed for only those heads which are out of alignment.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
	3. The off-line tester must be interfaced. 4. An industry-standard alignment pack is required. 5. The test has two parts: adjust mode and verify mode. If the question (adjust or verify) is answered with adjust, no head alignment testing is performed by the program. The program merely furnished a method (self-documenting) of positioning back and forth between cylinders 245. and 0, cylinder 245. being used to scope the DIE bits and cylinder 0 to adjust the head. The procedure for setting up the DDU is as follows. 1. Take RP04 off-line. 2. Write-protect the pack. 3. Mount A.C.E. pack. 4. Power down RP04 using ckt breaker (CB-1). 5. Connect DDU power cable to RP04 J2. 6. Interface DDU test card to RP04 slots A09-A10. 7. Set DDU function switch to head align. 8. Set DDU range switch to 2000 microinches. 9. Set DDU gate logic display switch to position 2. 10. Place a jumper between the following pins: A08-08 and A08-02 on the ISS card cage. This prevents the DCL from ever detecting an index error from the ISS logic. If this jumper is left out, the test may still function correctly. A brief explanation: some C.E. packs do not have an index pattern recorded on them and some do. If you have one that does not, the jumper will inhibit the test from not running because of the index error. If you happen to have a pack with index recorded on it, the jumper is not necessary but will not interfere with test operation. 11. Turn on RP04 ckt breaker (CB-1). 12. Put RP04 back on-line and ready. 13. Allow the drive to sit in this state for at least two hours. If this is not adhered to, erroneous results will occur. It takes this long for the C.E. pack to reach the internal ambient temperature of the RP04 and stabilize. 14. Run the alignment program.
ALIGN (DDRPK) (DFRPK)	HEAD ALIGNMENT VERIFICATION TEST This test will operate on all drives under test. The overall head alignment for a drive is verified in the following manner: 1. Recalibrate the positioner 2. Seek to cylinder 496 3. Verify alignment of heads 0 through 18 4. Seek to cylinder 8 5. Verify the alignment of heads 0, 1 and 17., 18 6. Seek to cylinder 800 7. Verify the alignment of heads 0, 1 and 17., 18 8. Seek to cylinder 496 9. Reverify alignment of heads 0 through 18. The following algorithm is used to verify the alignment of a particular head: 1. Offset the positioner to +600 microinches. 2. Record the value of the sign bit. 3. Move the positioner toward the track centerline in 25 microinch increments until the sign bit changes. Upon detecting a sign change, record the offset value.

*Listing on microfiche contains troubleshooting procedure.

Table 2 DDRPH Test Summary (Cont)

Test	Description
	4. Offset the positioner to -600 microinches and repeat steps 2 and 3 above. 5. Average the two values of offset and consider the head to be out of alignment if the average offset has turned out to be in excess of 75 microinches (absolute value) on cylinder 496, 150 microinches on cylinders 8 and 800. NOTES 1. Positive offsets indicate a head position toward the spindle side of the actual track centerline. 2. By default, normal printout mode (SW10 reset), the program will print a complete table of offsets for all heads. In short printout mode (SW10 set), information will be printed for only those heads which are out of alignment. 3. The off-line tester must be interfaced. 4. An industry-standard alignment pack is required. 5. The test has 2 parts: adjust mode and verify mode. If the question (adjust or verify) is answered with adjust, no head alignment testing is performed by the program. The program merely furnished a method (self-documenting) of positioning back and forth between cylinders 496. and 0, cylinder 496. being used to scope the DIE bits and cylinder 0 to adjust the head. SETUP PROCEDURES: 1. Install a CE pack with drive write-protected. 2. Slide the logic assembly forward. 3. Remove drive dc power. 4. Connect the head alignment unit (PN-211292) interface cable or tester's head alignment cable to the B04 slot of the logic assembly. 5. Turn drive dc power back on and load heads. 6. Press the calibrate switch on the head alignment unit or the tester's control panel. 7. Verify that the tester's ALIGN INVALID lamp is illuminated. If the setup does not work, refer to the appendix of the maintenance manual for possible faults and solutions. 8. To prevent index error, place a jumper between D04-R19 and D04-R39. This will prevent index errors. The jumper is to be removed when the head alignment test is complete. THERMAL EQUILIBRIUM REQUIREMENTS (MINIMUM): 1. With the wind tunnel in the closed position, the drive must operate in track following or seek mode for 20 minutes. Any pack may be used for the first 15 minutes but the CE pack must be mounted for the last 5 minutes. 2. The CE pack must reach equilibrium by being in the computer room for 1 hour, or by being used for the entire 20 minute cycle described in the first step.

ERROR MESSAGE SUMMARY
To be supplied.

*Listing on microfiche contains troubleshooting procedure.

```
>. DDRPI - RH10/RH20 - RP04/05/06 - RELIABILITY TEST
VERSION 0.7, SV=0.1, CPU#=3108, MCV=157, MCO=0, HO=30, 60HZ

SWITCHES = 000000 000000
CLK SOURCE = NORMAL, CLK RATE = FULL, AC BLK 0 , CACHE: 0 1 2 3

MEMORY MAP =
FROM      TO          SIZE/K
00000000 01777777        512


LIST THE PGM SWITCH OPTIONS? Y OR N <CR> - Y

SW[illegible]ILL INHIBIT THIS TEXT (KL SWITCHES IN "(XXXXXX)"
        DON'T USE "KL" SWITCHES IF IN USER MODE

LEFT SWITCHES
0  400000 ABORT
  (100000)
1  200000 RSTART
  (040000)
2  100000 TOTALS
  (020000)
3  040000 NO PRINT
  (010000)
4  020000 LPT PRINT
  (004000)
5  010000 BELL ON ERROR
  (002000)
6  004000 LOOP ON ERROR
  (001000)
7  002000 HALT ON ERRORS
  (000400)
8  001000 PRINT ALL ERRORS
  (000200)
9  000400 RUN RELIABILITY
  (000100)
10 000200 TEXT INHIBIT
  (000040)
11 000100 PAGING INHIBIT (N/A)
  (000020)
12 000040 MODIFY DEV CODE (N/A)
  (000010)
13 000020 CACHE OFF (KL10 ONLY)
  (000004)
14 000010 OPR SELECTION OF TESTS
  (000002)

RIGHT SWITCHES
18 400000 DELTRK - DELETION INHIBITED
19 200000 VARIAB - VARIABLES ENABLED
20 100000 ACTSEK - ACTUAL SEEKS ENABLED
21 040000 FREEZE - LOOP ON CMD LIST
22 020000 INHULD - INHIBIT DRIVE UNLOADS
23 010000 ALLADR - SELECT DRV ADR FIRST
24 004000 ALLDRV - SELECT ALL DRV FIRST
25 002000 ONESEC - SECTOR SELECT ENABLED
26 001000 INHRET - NO ERROR RECOVERY
```

ES 2
for opr sel of Tests
Run under DIAG on
LH 10
RH 0

GENERAL INFORMATION

Code

DDRPI.A10

Title

RH10/RH20 - RP04/5/6 - Reliability Diagnostic

Abstract

DDRPI runs in either exec or user mode and is designed to provide a flexible system reliability and diagnostic exercise. The operator can execute tests ranging from basic read or write operations up to complete diagnostic and reliability sequences.

DDRPI runs on any RH10/RH20 RP04/05/06 system and provides the following features.

1. Full-speed data transfers which utilize most of the drive's features (i.e., ECC correction, implied seeks, spiral transfers, etc.).

2. A random parameter data/mechanical reliability test (FRTEST) to provide a simulation of a drive in a system environment.

3. The ability to format an RP04/05/06 disk pack in either 16-bit (PDP-11) or 18-bit (PDP-10) mode.

4. The ability to loop on a particular sequence of instructions.

5. The capability of simultaneous operations to several drives across one or more Massbus controllers.

6. Dual-port operation of the RP04/05/06s.

7. On-line operation of the program using dump mode I/O to read and write. Special protection is given to guard against accidental data destruction in user mode. Pack formatting during time-sharing requires a pack mounted not in the system structure list.

Hardware Required

KA10, KI10 or KL10 CPU/48K of core (minimum)/ RH10-DF10 (up to 6) or RH20 (up to 8)/ up to 64 (single- or dual-port) RP04/5/6s

Preliminary and Associated Programs

Refer to diagnostic hierarchy (10/10 STD module).

Restrictions

This program has not been run on many of the possible system configurations. The system configurations used to date have been as follows.

1. KA10 with one RH10 and two RP04s and an RP06 drive (only one drive was dual-port)

2. KI10 (128K) with one RH10 (DF10 in 18- and 22-bit addressing mode), with different drive types but not mixed on the same RH10

3. KL10 (128K) with three RH20s containing an RP04, RP05 on one RH, an RP04, RP06 on the second, and a dual-port RP04 on the third

4. 1080 system (128K) with 2 RH10s and 10 RP04s

5. TOPS-10 (user mode) operation with six RP04s on one RH10 controller running a "6.02" monitor

6. TOPS-10 monitor has been run on the following system types: a 1055 system, 1070, 1077, and 1080.

7. TOPS-20 does not (yet) support user mode operation to the degree of a TOPS-10 monitor; thus, operation is extremely limited.

Notes

1. When the program reports drive "12" or "77" or whatever exists, it is referring to the first digit as the MBC unit number and the second digit as the drive unit number (i.e., drive 75 is Massbus control unit 7, drive unit 5).

2. DDRPI does not support simple test routines for use with DDT as was the practice with previous disk reliability diagnostics. The structure of this program dictates that it must be thoroughly understood before it would be possible to code simple write-read loops in DDT. The operator data test has been furnished to support simple write-read loops to a specific disk address.

3. There are several help files embedded in this program. In many cases typing H in response to a question will cause a short help file to be printed.

   When running under KLDCP it is advisable to set parity enable to 16 (PE16<CR>). This will disable parity errors (memory) from causing PAGE FAIL TRAP type errors when parity errors are detected.

Loading and Starting Procedure

Standard (Refer to the 10/10 STD module.)

Control Switches

Refer to Table 1.

OPERATIONAL CONTROL

This diagnostic is controlled via DDT. When started, the program will report the number of drives on line and ask WHAT TEST?

Typing a number of a test instead of the actual test name is acceptable and proper test dispatching will take place. The test name will be typed back to the operator before dispatching is executed.

Typing anything except a test number or name defined in Table 2 will cause an error message to be printed.

The following characters may be typed while a test is running:

A will report all drives available for testing.

H will cause a help message to be printed.

R will report program run time, number of disk blocks (sectors) read and written, along with the number of positions issued and test currently running.

S will report drives selected for testing.

NOTE

If the test is being run under KLDCP these characters must be terminated with a control T(↑T).

DDRPI TEST SUMMARY

The tests performed by this diagnostic are summarized in Table 2.

Table 1 DDRPI Control Switch Summary

Switch	Mnemonic	State	Description
0-10			Standard (Refer to the 10/10 STD module.)
11, 12			Not used
13-15			Standard (Refer to the 10/10 STD module.)
16	KAHZ50	0	60 Hz power
		1	50 Hz power
17			Not used
18	DELTRK	0	Pack DELETION/PROTECTION in effect
		1	Suspend any operator-selected pack DELETION/PROTECTION
19	VARIAB	0	Normal operation
		1	Allow operator to change test parameters
20	ACTSEK	0	Use implied seeks
		1	Force actual seeking instead of implied seeking
21	FREEZE	0	Normal operation
		1	Allow the operator to execute a script (up to 20 commands) repeatedly
22	INHULD	0	Normal operation
		1	Inhibit drive unloads on a hard error abort
23	ALLADR	0	Normal operation
		1	Select all disk addresses before changing drive numbers
24	ALLDRY	0	Normal operation
		1	Select all drives on an MBC before selecting a new MBC
25	ONESEC	0	Normal operation
		1	Limit transfer size to one sector
26	INHRET	0	Automatic error recovery enabled
		1	Inhibit error recovery procedures
27-29			Not used
30-35	TSTSEL		Test Selection Test No. Test No. 00 TOTAL 07 FRTEST 01 RONLY 10 FORMAT 02 WONLY 11 SEQGEN 03 OPDTST 12 ACCEPT 04 STEST 13 NEXTST 05 SEKTST 14 PTIME 06 INTTST 15 RCTEST Four tests-MAPOUT, DELETE, PAKINT, and CONFIG-cannot be selected via the switches.

Table 2 DDRPI Test Summary

Test	No.	Description
TOTAL	00	All tests which do not require operator intervention are run.
RONLY	01	RONLY is a test designed to read all areas of a disk pack. No data comparison can be made on the data read because the type of data stored on the pack is not known. No write data transfers will be issued during this test (i.e., no damage to the data on the pack should occur). Any detected data errors will be reported and an error recovery will be issued. Errors corrected by using the ECC logic will be reported with both the original data and the correct data given.
WONLY	02	WONLY is a data transfer write-only test. This test will write data on all unprotected disk pack areas and DESTROY any previous data on the pack. Either hardware or software (by an operator command) write-protection will inhibit a drive from being selected to run this test. Any type of data that happens to reside in memory will be used for this test unless the operator requests a specific data pattern to be used. Setting console switch 19 VARIAB, will enable the operator to specify the data pattern to be used.
OPDTST	03	OPDTST - OPERATOR SELECTABLE DATA TEST - This test allows the operator to specify the type of data to be used for writing and reading data to/from the drive pack. OPDTST will select all unprotected pack areas for testing. Data will be written, read, and verified for correctness on all selectable areas on all drives. Switch 19, VARIAB, allows the operator to select transfer parameters such as disk address, data pattern to be used, pass iteration count, and data transfer size.
STEST	04	STEST - SURFACE TEST - This test will select all unprotected pack areas and will use several different data patterns to verify that the disk pack can hold data of different types reliably. The data will be written, read, and verified for correctness. Data patterns used for this test are: Floating 1s Floating 0s Alternate bits pattern Random data Binary up-count Disk worst-case data Channel pattern (DF10 type channel) Channel parity (DF10) STEST checks the reliability of the hardware to perform spiral-type data transfers. This section starts by writing data at sector 11 of the selected test track to the middle of the next track. The hardware should keep a record of the surface number and update it when the spiral occurs. The read and verify section issues two read transfer requests to check the entire write transfer. STEST uses the remaining test patterns with full track data transfers and no spiraling.
SEKTST	05	SEKTST MECHANICAL RELIABILITY TEST 1. Issues a recalibrate, then seeks to the maximum cylinder. Repeat 100 times. 2. Seeks between cylinder 000 and cylinder 128. Repeat 100 times.

Table 2 DDRPI Test Summary (Cont)

Test	No.	Description
		3. Perform one cylinder forward seek starting at cylinder 000 and ending at maximum cylinder. This is followed with a reverse 1 cylinder seek test starting at the maximum cylinder and ending at cylinder 000. Repeat 10 times. 4. Incremental seeks starting at cylinder 000 (i.e., 0 to 1, 0 to 2, 0 to 3, etc.). This is followed with an incremental reverse seek sequence starting at maximum cylinder. 5. Random seek test - seek to 500 random cylinders on the test drive. 6. Servo noise test - seek to a reference cylinder, then seek to cylinders n+4, n+1, n+3, n+2, and n+5. Repeat the sequence until all cylinders have been referenced. 7. Cylinder difference test - using cylinder 0 as a reference and with HCI set, issue a read headers and data command to cylinder n. Verify the cylinder position by checking the header data. Repeat until all cylinders have been checked in both the forward and reverse directions.
INTTST	06	INTTST - TRACK/DATA INTERACTION TEST - This test is designed to test for track interaction failures at the cylinders where the write current is stepped. The write current is stepped every 100 octal cylinders from cylinder 0 for RP04/05s and every 200 for RP06s. First seek to the reference cylinder n and write a worst-case data pattern on all tracks at ref cyl, ref cyl-1, and ref cyl+1. Then read/verify the data on those tracks. The reference cylinder is the cylinder where the write current is stepped. Next, write a parity pattern on all tracks of the reference cylinder and read/verify the data. Go back and reverify the data on ref cyl-1 and ref cyl+1. Repeat ten times. Repeat the sequence until all current change cylinders have been referenced. Next, get a reference cylinder and write a parity pattern on either side of it and verify the data. Reverify that the data on the reference cylinder has not changed. Repeat the sequence until all current change cylinders have been referenced. Pack DELETION/PROTECTION warning is inhibited during this test.
FRTEST	07	FRTEST - FAST RANDOM PARAMETER TRANSFER TEST - This test is designed to randomly select disk areas for random selected data patterns to keep the positioner in maximum motion. Data transfer size can be limited to one sector if program switch 25 ONESEC, is set. Setting this switch will keep the positioner moving.
FORMAT	10	FORMAT - DISK PACK FORMATTER/VERIFIER - This routine provides facilities for formatting/verifying disk packs in either 16-bit (PDP-11 mode) or 18-bit (PDP-10 mode) format. The operator has the option of formatting an entire disk pack or just a portion of it. Formatting/verifying is done on a track by track basis (i.e., an individual sector cannot be processed; the entire track must be processed).

Run Time Format 9 min

Table 2 DDRPI Test Summary (Cont)

Test	No.	Description
SEQGEN	11	SEQGEN is a test script generator. SEQGEN is capable of creating a test script containing up to 10 tests from the test list or executing an existing test script that was previously generated from SEQGEN or preloaded by the program. The preloaded scripts are: TOTAL - All tests ACCEPT - Drive acceptance testing MECH - All mechanical tests (no data) TIMING - All timing tests RUNLST - Run the script generated list
ACCEPT	12	Automatically runs the ACCEPT script.
NEXTST	13	NEXTST - READ/WRITE NEXT TEST - The read next-write next test is designed to test the drive's ability to switch between a read and write operation in the sector gap time. Switching between cylinders and surfaces are inhibited because of the time involved and/or it serves no useful purpose. 1. First initialize the test track (cylinder = 0, surface = 0 and the starting sector = 0) by writing alternate bit data pattern (525252,,525252) on it and then doing a read/verify of the data written. 2. Then set the transfer size equal to two sectors and proceed to issue back-to-back write and read data transfers utilizing the secondary transfer command registers in the RH20 controller (write two sectors/read two sectors). 3. After every 200 disk revolutions of writing/reading, recheck the data on the test track. Normally the data read from the disk will rely on ECC to detect errors.
PTIME	14	PTIME - DISK MECHANICAL TIMING TEST - PTIME will measure the times of recalls, seeks and disk rotation. If VARIAB (switch 19) is set, the operator may select seeks to be timed. NOTE This test has been designed to perform a relative measurement of seek times from drive to drive or from a previous run of the drive selected for test. It is not designed to measure the drive's performance against the seek time specification.
RCTEST	15	RCTEST - RANDOM COMMAND TEST - This test is designed to cause an overlap of position and data transfer commands to selected drives. The commands issued are: READ - No data compare READ HEADERS WRITE - Any data to be used SEEK - Seek to selected random address SEARCH - Search to selected random disk address All data transfers are limited to two sectors in length.
MAPOUT	N/A	This routine will interrogate the pack BAT BLOCK storage area for all selected drives. A copy of this area will be kept in the drive DSB area to safeguard against accidental destruction while testing the drive. Upon system configuration, the pack BAT BLOCK area will automatically be read and stored. Hard data error spots on the pack will be deleted from testing.

Table 2 DDRPI Test Summary (Cont)

Test	No.	Description
		MAPOUT will report the pack BAT BLOCK contents and allow the operator to update the pack BAT BLOCK (no update possible in user mode). The operator may also use this service to eliminate known soft error spots from testing or to include/exclude any hard error spots on the pack from testing. NOTE A pack formatted in 16-bit format (PDP-11 mode) will not have a BAT BLOCK area and cannot contain a BAT BLOCK area.
DELETE	N/A	This service routine allows the operator to eliminate areas of a pack from testing or to use only specific areas of the pack for testing. This service is called from MAPOUT if the operator desires to remove "soft" spots from testing.
PAKINT	N/A	This test is actually a script which will automatically format, map and create the BAT BLOCKS for the selected RP04/05/06 disk drives. The script can be run in exec mode or in user mode under a TOPS-10 monitor. The purpose of this script is to limit the dialogue necessary to complete this operation and avoid user errors. An additional feature has been added to limit the program startup dialogue in user mode. A question is asked (short startup dialogue Y or N?). If a Y is typed the program will be initialized automatically.
CONFIG	N/A	This routine allows the operator to RE-MAP the Massbus and report the results. It also enables the operator to add or delete drives from the test list without retyping the entire list. Write-enabling a drive in user mode is accomplished with this service (in user mode, all drives are initially write-protected at run time). Typing H to WHAT DRIVE will give the operator a list of acceptable commands to CONFIG.
H	N/A	Typing H to WHAT TEST will type the complete list to the operator.

Run time 11 min

ERROR MESSAGE SUMMARY

The printing of data-compare errors will be delayed until all I/O has finished. The following information will be reported.

Memory address of data compared

Expected data on the compare

Actual data contained at the error address

Bits in difference from the compare

Drive number of failing unit and disk address (cylinder, surface, and sector) are reported as a logical block number. Printouts are limited to three detected errors unless PRINT-ALL-ERRORS switch is set in the program switches.

All disk interrupts initially serviced on PI level 4 request a software interrupt to level 5 for further processing. Channel 5 interrupts handle all normal and error processing for the disk. Channel 5 service is common for all types of PDP-10 processors and Massbus controllers. Upon data transfer error detection, level 5 service will request a data recovery procedure using ECC correction and offsetting (if necessary) on interrupt level 6.

All nonfatal drive errors detected during the normal processing of drive interrupts on channel 5 will request an error recovery procedure called RETRY. Once a drive is processing a recovery request, all other system data transfers and positioning commands are inhibited. Upon either successful or unsuccessful error recovery, normal program command execution is automatically enabled and normal program operation continues until another recoverable error is detected by interrupt service. All error recovery is initiated at interrupt level and is automatic.

POSITIONING RECOVERY ALGORITHM
Issue a recalibrate command to the drive, then try to position again. Retry limit is set to 3. This recovery is used for recalibrate, seek, and search.

READ HEADER/WRITE DATA RECOVERY ALGORITHM
Reissue the command and check for errors. Retry limit is set to 3.

READ DATA RECOVERY ALGORITHM
Try to recover the whole data transfer by splitting it into individual sector read/recover transfers. Apply the following algorithm to each sector of the transfer to be recovered.

1. Issue up to 16 rereads of the sector at track centerline (no offset) and apply ECC correction for data recovery if needed.

2. If recovery is unsuccessful at this time, issue up to two rereads at +400 microinches offset for RP04/05s or -200 for RP06s and apply ECC correction if necessary.

3. Issue up to two rereads at -400 microinches offset for RP04/05s or -200 for RP06s and apply ECC correction if necessary.

4. Issue up to two rereads at +800 microinches offset for RP04/05s or +400 for RP06s and apply ECC correction.

5. Issue up to two rereads at -800 microinches offset for RP04/05s or -400 for RP06s and apply ECC correction.

6. Issue up to two rereads at +1200 microinches offset for RP04/05s or +600 for RP06s and apply ECC correction.

7. Issue up to two rereads at -1200 microinches offset for RP04/05s or -600 for RP06s and apply ECC correction.

Failure to recover any sector of data within the complete transfer after all seven steps of recovery have been applied results in the program eliminating that pack area from further use and printing "UNSUCCESSFUL RECOVERY . . ."

If the soft or hard error limit for a drive is exceeded during error recovery, an appropriate error message will be issued and the drive will be eliminated from the testing sequence.

ECC errors are reported in the following format.

```
## ECC CORRECTION RECOVERY ##
ECC POSITION = XXXX     ECC PATTERN = XXXX
BUFFER ADDR    XXXXXXXX                XXXXXXXX
DATA BEFORE ECC XXXXXX XXXXXX          XXXXXX XXXXXX
DATA AFTER ECC  XXXXXX XXXXXX          XXXXXX XXXXXX
```

GENERAL INFORMATION

Code	DDRPK.A10
Title	RH10/RP06 Basic Device Diagnostic
Abstract	DDRPK is the RH10/RP06 Basic device diagnostic, designed to isolate solid RP06 faults to the faulty module or group of modules within the DCL or drive electronics. Excellent scope looping capabilities have been designed into each test. Testing is done on a start small and build up basis. The diagnostic starts by testing even the most trivial of control bus cycles and ends by doing full-speed data transfer operations to various disk addresses. Positioning logic is also tested along the way. In addition to the straight line diagnostic tests, the diagnostic contains a head alignment/ verification subroutine that is invoked by setting the proper sense switch.
Hardware Required	KA10, KI10 or KL10 CPU/48K of core (minimum)/RH10/DF10 (up to 6)/up to 48 (single- or dual-port) RP06s
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	Examination of switch 13 takes place only once, immediately following the initial load of the diagnostic into core from its storage media. The cache option is selected only during the very first program initialization cycle. Attempting to alter the state of the switch from that point on will have no effect.
Notes	Refer to DDRPH summary for documentation on this diagnostic. DDRPH tests RH10s-RP04s. This diagnostic tests RH10s-RP06s. The major difference between the two diagnostics is the code (instruction) used to implement the testing. Where differences in documentation occur between this diagnostic and DDRPH, both are listed in the DDRPH summary.

GENERAL INFORMATION

Code	DDTMH.A10
Title	Magtape Reliability Test
Abstract	The tape reliability test is designed to exercise the TM10A/B tape control and up to eight tape units. The units may be 7- or 9-channel. The program may be run in exec mode or in a time-sharing system. The purpose of this program is to measure the tape system reliability. DDT is provided for operator communication with the program. The program will operate with any mixture of transport types.
Hardware Required	KA10, KI10 or KL10 mainframe/32K of core (minimum)/TU10, TU20, TU30, TU40, TU45 or TU70 magtape subsystem.
Preliminary and Associated Programs	Refer to diagnostic hiararchy (10/10 STD module).
Restrictions	1. TU45 and TU70 are limited to time-sharing only; also, error status will not agree. 2. The DF10/DF10C write even lockout switch must be in the off position (allow even parity). 3. The DF10C parity error stop switch must be in the off position (continue on error).
Notes	1. Additional core will be used to expand the size of the data buffer up to 256K/4096K when running MTEST. 2. Aborting tests or subroutines quickly fills the pushdown list with saved ACs. The pushdown list can be reinitialized by typing (to DDT) JSR INT$X, where $ is an altmode.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.) Additional Procedures Exec Mode Operation 1. Mount tapes of known quality on the transports to be tested. 2. Place these transports to READY, WRITE ENABLE, and REMOTE. 3. After setting appropriate data switches, start the program at location 30000(8). The program will be initialized and control will transfer to DDT. User Mode Operation 1. Gain access to the time-sharing system. 2. Assign the tape transports to be tested. 3. Mount tapes of known quality. 4. Place the units to WRITE ENABLE, READY, and REMOTE. 5. Set the data switches if desired. 6. Type RUN DDTMH (core argument optional - program requires a minimum of 32K). If program cannot obtain at least 32K core, it will exit immediately.
Control Switches	Refer to Table 1.

OPERATIONAL CONTROL
When the program starts it will respond on the terminal with:

TELETYPE SWITCH CONTROL? TYPE 0,S,Y OR N -

0 - The program will see all switches as reset.

S - The program will use the switches as they were set previously. This response is only valid if the program was stopped and then restarted.

N - The actual data switches will be used.

Y - The program will respond with:

SPECIFY LH SWITCHES IN OCTAL -

Respond by typing in six octal digits, representing the left-hand data switches, and carriage return. Typing anything else will cause the above typeout to repeat.

The program will then respond with:

SPECIFY RH SWITCHES IN OCTAL -

Respond in the same fashion supplying the right-hand switches as six octal digits.

User Mode Operation Only

After the above, the program will respond with:

WANT MONITOR TO ATTEMPT ERROR RECOVERY?

Respond by typing a Y or N.

Y - The program itself will not attempt error recovery.

N - Will inhibit the monitor from trying any error recovery and all errors will be reported. The program will then use data switches 23 and 28 to determine whether it should try its own recovery.

The program will next respond with:

WANT ERRORS REPORTED TO SYSERR?

Y - Errors detected by the program will be logged for future reporting by SYSERR. [This assumed the time-sharing system (monitor) has the SYSERR reporting feature for magtape].

N - The monitor will not log errors for SYSERR use.

The program will initialize and assign each transport to a software I/O channel. These assignments are typed on the terminal. Control is then transferred to DDT.

Starting address of program = 30000(8).
Starting address of DDT = 30007(8).

DDTMH COMMAND SUMMARY
DDTMH uses UUOs and DDT to control test dispatching.

Table 2 describes some of data patterns used with the GEN and COMP commands. Consult the listing on microfiche for a complete list.

Table 3 lists the mnemonic names of commonly used registers and buffers.

Table 4 lists and describes the commands available under DDTMH.

Table 1 DDTMH Console Switch Summary

Switch	Description
0-17	Standard (Refer to the 10/10 STD module.)
18	With this switch a 0, 9-channel drives will be tested in the core dump (non-industry-compatible) mode. With this switch a 1, the industry-compatible mode will be selected for 9-channel drives. This switch overrides switches 20-22 by setting parity odd and density to 800 BPI.
19	Repeat last file of RTEST (used to repeat files that detected errors). In user mode type R to set switch or C to reset switch.
20	With this switch a 0 odd parity is used for all tests. With this switch set and switch 18 0, even parity is used. There is a magtape restriction that characters of 00 must not be placed on tape with even parity; therefore, data patterns of .ZEROS, .FLONE and .ADR must not be used. .RAND may be used with even parity. .WCP9 may be used with even parity only on a 9-channel tape. MTEST, MMTEST, EOFTST, EETEST, COMPAT and INTER ignore this switch, using odd parity. RTEST does use even parity if this switch is set.
21-22	Density select (if SW18 = 0) 21 22 DENSITY SETTINGS 0 0 = 800 BPI in exec mode, or system standard in user mode 0 1 = 200 BPI 1 0 = 556 BPI 1 1 = 800 BPI
23	With this switch set, no read retries will be performed upon detection of a parity error. Thus inspection of data should show the cause of the parity error. With the switch reset, up to 64 read retries will be peformed. The number of retries then becomes a figure of merit for the transport. (If ECO number 21 is installed, all retries past number 32 will be performed with the transport read amplifiers operating at the low threshold level). Retries are never attempted by this program if monitor is doing retries.
24	With this switch 0, the channel control word will be split up into a number of sequential control words. (TM10B - exec mode only.)
25	With this switch and switch 24 in the 0 position, the several control words will be spread about memory to a buffer defined by C(.CB) (channel buffer), and .CS (channel buffer size). (TM10B - exec mode only.)
26	Not used.
27	With this switch reset, RTEST alternates random record size between modulo 77(8) and modulo 7777(8). When set, RTEST will write random-length records which alternate in size between modulo 777(8) and 32K of memory.
28	With this switch set, no write retries will be performed. With the switch reset, up to 64(10) retries will be performed. For details of the write retry operation, see the program description of the write routine section. Retries are never attempted by this program if monitor is doing retries.
29	Do all read operations at low read threshold level (effective only if ECO number 21 is installed).
30	Terminate current pass of RTEST, rewind all drives and print statistical data for each tested drive. Print same statistical summary at the end of INTER and MTEST. Print statistical summary for drive under test at end of compare routine and at end of read portion of COMPAT.
29-35	These switches are not used.

Table 2 Patterns for GEN and COMP Routines

Pattern	Description
.ONES	All 1s
.ZEROS	All 0s
.FLZRO	Floating 0 777777777776, 777777777775, 777777777773, etc.
.FLONE	Floating 1s 000000000001, 000000000002, etc.
.RAND	Random numbers, from base register .RAN
.CNT7	A count pattern for 7-channel tapes 010101010101, 020202020202, etc.
.CNT9	A count pattern for 9-channel tapes 001 001 001 001 001, 002 002 002 002 002, etc.
.ALTB	Alternating bits 52525252525252 etc.
.ALTB7	A true alternating bit pattern for 7-channel tapes
.ALTB9	A true alternating bit pattern for 9-channel tapes
.SQAR7	A 7-channel square wave pattern of alternating frames of 111000 000111 111000, etc.
.SQAR9	A 9-channel square wave pattern of alternating frames of 111000 000111 111000, etc.
.OPR	Any operator-selected word
.ADR	C(CORE) = address of core
.WCP7	A worse-case pattern for 7-channel tapes
.WCP9	A worse-case pattern for 9-channel tapes

Table 3 Mnemonic Names of Commonly Used Registers

Register Mnemonic	Use
C(.SIZE)	The number of words to transfer.
C(.BUFF)	The first location in the data buffer.
C(.CS)	The size of the control word buffer.
C(.CB)	The first location of the control word buffer.

Table 4 DDTMH Exerciser Routine Summary

Command	Description	Cross Ref.
COMP	COMP.ONES<CR> Compare pattern specified with contents of memory buffer.	1
COMPARE	COMPARE 1,0<CR> Compare a file on magtape with another file on the system.	2
COMPAT	COMPAT<CR> Run the compatability test.	3
DUMP	DUMP 5,1<CR> Dump (prints) the contents of tape on the terminal or line printer.	4
EETEST	EETEST<CR> Run EOF test for 1000 passes.	5
EOFTST	EOFTST<CR> Run EOF test for approximately 5 minutes.	5

Table 4 DDTMH Exerciser Routine Summary (Cont)

Command	Description	Cross Ref.
GEN	GEN.ONES<CR> Fill the memory buffer with the data pattern specified.	6
INFO	INFO 3, arg<CR> The INFO routine is provided so that the operator may obtain information about the system status. The number indicates the unit number. INFO commands are described below. INFO CHAN<CR> - Print the current channel assignments of all available transports. This is available in user mode only. INFO CLEAR<CR> - Clear all statistical, pick/drop and retry information. This is also accomplished by restarting the diagnostic or by typing ↑C in exec mode. INFO D, PICK<CR> - Print accumulated pick/drop information for unit D. The print format is the same as after a data comparison error. INFO D, POS<CR> - Print the position of the tape on unit D. The value printed is the current record number of the number of records and file marks between the heads and BOT. INFO RET<CR> - Print read and write retry information for the tape system. This is primarily an engineering tool for the evaluation of tape manufacture and transports. If only one transport is used in a test it also provides information about that particular transport. WRITE & READ RETRY DATA # OF WRITE RETRIES / # OF READ RETRIES RETRY NO. 5 1 RETRY NO. 10 3 The first line of data gives information about a write error; namely, there were 5 retries before a record could be written without error, and this occurred on one record. The last line indicates that 3 records were reread 10 times before no parity error occurred. INFO D, STAT<CR> - Print the statistical data for unit D. A sample printout follows. STATISTICAL DATA - DRIVE #0 RECORDS WRITTEN 2883 / RECORDS READ 2818 WORDS WRITTEN 3504986 / WORDS READ 3504921 FILE MARKS WRITTEN 892 RECORDS SPACED FWD 0 / RECORDS SPACED REV 34 FILES SPACED FWD 2924 / FILES SPACED REV 2891 REWINDS 6 / READ DATA ERRORS 0 WR STATUS ERRORS 0 / RD STATUS ERRORS 0 WR PARITY ERRORS 1 / RD PARITY ERRORS 0 WRITE RECOVERIES 1 / READ RECOVERIES 0 NON-REC WRITES 0 / NON-REC READS 0	
INTER	INTER<CR> Run the interchange test.	7
MMTEST	MMTEST<CR> Run MTEST for 1000 passes.	8
MTEST	MTEST<CR> Run MTEST for approximately 5 minutes	8

Table 4 DDTMH Exerciser Routine Summary (Cont)

Command	Description	Cross Ref.
READ	READ D,Y<CR> Read a word into memory. if Y = 0, read only if Y = PATTERN (i.e., READ 1,.ONES) compare the contents of the memory buffer to the pattern after the read operation. READ 1D,Y<CR> This is another form of the READ UUO with bit 9 of the AC field set. When this bit is set, record length and file mark errors are suppressed. The routine will read up to N words defined by .SIZE.	9
RRTEST	RRTEST<CR> Run RTEST continuously, print statistical data each time all tapes have hit end of tape.	10
RTEST	RTEST<CR> Run RTEST for approximately two minutes and print statistical data before returning to DDT.	10
SPACET	SPACET 1, N<CR> Space tape as specified by N. N > 0 space forward N records N < 0 space reverse N records N = 0 rewind N = EOF space forward one end of file N = -EOF space reverse one end of file	11
TAPE	TAPE N<CR> Display how data looks on tape. N = 0 7 channel transports N = 1 9 channel transports (dump mode) N = 2 9 channel transports (industry-compatible mode	12
TEXIT	TEXIT<CR> Return control to monitor mode (time-sharing mode only).	N/A
WRITE	WRITE D,Y<CR> Write to tape. Y = 0 write the memory buffer Y = pat write the pattern specified Y = EOF write an EOF mark Y = EOR write with an extended record gap WRITE 1D,Y<CR> This is another form of the write UUO; namely, with bit 9 of the AC field set, the routine will not try to rewrite the record when an error is encountered.	13

COMMAND DESCRIPTION
This section describes the commands summarized in Table 4.

1. COMP.ONES<CR> - This routine will check the contents of memory for the pattern specified. A maximum of three errors will be printed each time the routine is called, unless switch 8 is set. The routine is otherwise similar to GEN.

2. COMPARE D,N<CR> - This routine is provided to examine and compare a file written on a magtape which might have errors when the correct data is known to be in some other file on the system. This function is available only in user mode and will exit immediately if called in exec mode.

   The letter D represents the channel number of the transport on which the file resides. The transport need not be write-enabled. N is an argument to position the tape before the compare.

N = 0 rewind the tape and read from BOT
N > 0 rewind the tape, space n records, then read
N < 0 rewind the tape, space n files then read
N = CUR do not position the tape, read the current position

The input file will then be asked for on the monitor. Respond with the name of the file along with the device on which it exists and the project, programmer number if different from your own. The files will be compared and any differences will be reported as in any other test. This function will also report any differences in length of the two files.

Additional executions of this command will assume the same input file. This allows the ability to use the command in a loop to continually read and compare the file for scoping or testing. The file name will be asked for after each initialization (JSR INT$X) or interruption with altmode.

This test does not know the size of the records on tape. Therefore, C(.SIZE) is set to the highest possible value and all read UUOs have bit 9 set to inhibit record-length error printouts. This will cause the record-length error flag to be set when any status error printout is given and should not be mistaken as an error.

3. COMPAT<CR> - The purpose of this test is to ensure that all drives can read and write on a single tape. This test will serve with the interchange test to ensure all tapes will be interchangeable. This test can be run on a time-sharing system or batch processing via submitting a control file to check a drive's compatiblity with the rest of the system while putting little strain on the system.

Only one drive will be tested at a time. If only one drive is available, the test will start immediately on that drive. Otherwise WHAT DRIVE - will be asked and a drive number should be typed followed by a carriage return.

The operator is asked if he wants to create a new compatiblity tape. If he responds with a Y the drive is checked for writability. Writing is disallowed if the write-enable ring is off. If the operator responds with an N, the tape is rewound and is read. Reading is still allowed even if write-enable ring is off. If switch 31 is set, the header information is printed. The operator is then asked whether or not he would like to read the full tape. If he responds with a Y, reading will continue until two EOFs in a row are seen (logical-end-of tape). If he responds with an N, bunch number will be asked and reading continues until the number of bunches expected are reached. The operator is asked WHAT NEXT (REWIND, WRITE OR ERASE)?.

REWIND will rewind the tape and exit the test. ERASE asks HOW MANY BUNCHES JUST READ? Type 0 or a positive number. The tape will be backspaced, if necessary, and an EOF written to form a logical-end-of-tape. WHAT NEXT? will be asked again. WRITE will ask the questions for the header information and write a bunch.

NOTES

Write or erase is strictly disallowed if the write-enable ring does not exist.

Compatibility does NOT write an end-of-file at the beginning of tape.

If the operator did want to create a new tape, he is asked if he wants to WRITE AN ENTIRE TAPE USING THE SAME HEADER DATA?. A NO response will write only one bunch, whereas a YES will fill the tape without repeatedly asking for drive type, etc.

The operator is also given the option of using either the reduced-size (half of the normal size) format or normal size to write or read bunches of records. To use the reduced-size format type Y to the question DO YOU WANT TO USE REDUCED-SIZE FORMAT (N = NORMAL FORMAT)?.

NOTE

The format and header information is described in detail in the listing on microfiche.

In user mode, regardless of reading or writing, the operator is asked if system standard density should be used. NO will write and read at 800 BPI, while YES will use the system standard density. This allows reading and writing 1600 BPI tapes (via RH10s and DX10s) if the user has previously done a SET DENSITY monitor command.

NOTE

Densities of 200 and 556 BPI cannot be used because the TM10 does not allow industry-compatible mode for anything but 800 BPI for a 9-track drive.

When writing a tape the user is asked for the transport type. Type the transport type (TU10, TU20, TU30, etc.) followed by a carriage return. This is a SIXBIT word which will be part of the header data. A maximum of six characters may be typed.

The operator will then be asked for the transport serial number; again, a maximum of six characters is allowed.

In exec mode, on a KA10, the processor serial number will be requested; and for all exec systems, the current date. Type the date as MM-DD-YY.

Finally, the operator is asked to input a comment. The comment field is limited to 10 characters of the SIXBIT set.

At the end of the bunch, two end-of-file marks are written to form a logical-end-of-tape. The tape is then backspaced one file and verified.

When writing is finished, the operator is asked WHAT NEXT (REWIND, WRITE, OR ERASE)?. For responses, see above.

NOTE

In order to support the COMPAT test batch processing, a $ (dollar) sign is put in the beginning of each question for transport type, transport serial number, and comment. In addition, a % sign is put in front of the rewinding message after writing a full tape. Neither $ nor % sign has any effect on COMPAT test running under time-sharing.

4. DUMP D,N<CR> - This routine is provided for operator's convenience when it is desirable to see octal data of a record or records on tape. The letter D represents the transport number (a digit from 0 to 7), in exec mode and channel number in user mode. The letter N represents the starting point at which printing is to occur.

N = 0	rewind and print
N > 0	rewind, then space N records and print
N < 0	rewind, then space N file marks
N = CUR	print from current position

The command DUMP 5,1 would rewind unit 5, then space over the first record and print from record 2 on. The command DUMP 5,-2 would rewind unit 5, then space over two file marks and print further records.

After positioning the tape the routine will ask for the number of records to print. Acceptable answers are: +M to dump M records, -1 to dump until the end of tape is reached, <CR> to dump just one record. Either octal or decimal numbers may be entered but decimal numbers must be followed by a decimal point. (NOTE: M must be in octal.)

The routine then asks PRINT JUST 8 WORDS OF EACH RECORD? If you need to see the entire record, type N. However, if you just want to make sure each record contains the correct pattern, type Y.

Data may be printed on a terminal or line printer. On the terminal it is printed four binary words to the line. The heading for each record is the record number. This represents the number of records and file marks from load point. If N < 0, this number will initially be 0, as there is no way to count records when spacing to file marks. This number is followed by the number of words in the record. The first column of numbers represents the number of the word in the record.

Reading and printing, when dumping more than one record, will continue until a double file mark is encountered (logical end of tape). Printing may also be terminated by pressing the ALTMODE key (PREFIX in model 37 terminals) or control C (↑C).

This routine does not know the size of the records on tape. Therefore, C(.SIZE) is set to the highest possible value and all read UUOs have bit 9 set to a 1 to inhibit record-length error printouts. This will cause the record-length error flag to be set when any status error printout is given and should not be mistaken as an error.

5. EOFTST<CR> or EETEST<CR> - The end of file test checks for EOF reliability by continuously spacing forward and reverse across random numbers of EOFs and then verifying tape position.

The test first rewinds all available transports and writes an EOF on each unit Next a 1-word record which contains a 1 is written on all units followed by an EOF, then a 1-word record containing a 2, followed by an EOF, etc. This procedure is continued until record number 177(8) is written, after which a number of EOFs are written to provide a logical end of tape. All transports are then rewound. The program then goes to a random number generator to select the unit and target file. For example: if the first file is file number 50, the program will space to EOF 50 times, then do a read. The 1-word read should be a 50. If the next file picked for that unit is a number 30, then the program will cause the unit to space-reverse to EOF 20 times, do a space-reverse one record to get over the target file, then do a read which should be record number 30.

NOTE
DO NOT set switch 18 when running EOFTST or EETEST test on 9-channel tape.

6. GEN .ONES<CR> - This routine will generate data in memory. The number of words and their location is determined by the values in registers .SIZE and .BUFF. Refer to Table 2 for a list of patterns available.

7. INTER<CR> - The purpose of this test is to ensure the interchangeability of tapes written on the PDP-10. The routine interrogates the tape system by trying to rewind each transport. This will inform the routine which transports are ready.

   The operator is given the option of creating new tapes or verifying previously written tapes. To verify existing tapes, type N to the question DO YOU WANT TO CREATE NEW TAPES?.

   The operator is also given the option of using either the reduced-size (half of the normal size) format or normal size format to create new tapes or verify existing tapes. To use the reduced-size format, type Y to the question DO YOU WANT TO USE REDUCED-SIZE FORMAT (N= NORMAL FORMAT)?.

   NOTE

   Format and header information is described in the listing on microfiche.

   Interchange does NOT write an end-of-file at BOT.

   When creating new tapes, WRITE 3 BUNCHES?, is asked. Typing N will fill the tape.

   A header record is the first record on the tape and, for INTER only, will contain six question marks for the transport type and the drive number for the transport serial number. In exec mode on a KA10, the processor serial number will be requested and, on all exec mode systems, the current date, if it has not already been asked for. INTER does not ask for comments, leaving that field blank.

   The transports are written sequentially. Write retries are permitted for write errors (if switch 28 = 0). A record written badly will remain bad throughout the test.

   The tapes are written and verified sequentially, one bunch at a time. The last bunch of the tape is followed by a double end-of-file mark (logical end of tape).

   EOT messages are provided whenever reaching end of tape. In addition, option for statistical printout request is available.

   All tapes are then rewound and the operator is asked, DO YOU WANT TO ROTATE/READ?. A NO response will cause the test to exit. A YES response prints a message asking the operator to type a carriage return when he has rotated the tapes, and then the program will wait in a terminal input loop until a carriage return is typed.

   Since INTER does not count the number of times the tapes are rotated, each tape should be clearly marked before the first rotation. Probably the easiest way to keep track of tapes and rotations is to identify each tape with the number of the drive it starts out on, then rotate each tape to the next higher numbered unit until it ends up on the drive it started on.

8. MMTEST<CR> or MTEST<CR> - This test is useful in the TM10/B system. It is a test to ensure that the data channel can communicate to all of the memory modules in the system. It will move the data buffer up through memory until all of memory (256K for a KA10 or 4096K for a KI10 or KL10) has been utilized.

   The program will rewind and write a file mark on each transport in the system, remembering which are ready and not write-locked. Memory is then divided into 8K segments. Starting with the first segment, an address pattern will be generated in memory. Then all transports will be written sequentially. The transports are then all backspaced and the tapes are read and checked for accuracy. Because of the address data written on the tape, data errors would appear as wrong words, much like core memory address test. If the error data is 400000000021 it indicates that no data was read into those addresses. The next 8K segment is then selected and tested. When all 8K segments have been tested eight times data switch 2 is checked and, if it is set, error

totals are printed. If MTEST was called, the test is completed and control is returned to the test dispatcher. If MMTEST was called, the test continues until 1000(8) passes have been completed. Switch 2 is checked after each eight complete passes through memory.

Whenever one transport reaches EOT, all transports are rewound and the current segment of memory is retested. All memory will be tested even if not contiguous.

NOTE
Do not use switch 18 while running MTEST or MMTEST.

9. READ D,Y<CR> - The read subroutine is a UUO where D is the drive number in exec mode and the channel number in user mode. The Y argument in bits 18-35 of the UUO may contain the following:

Y = 0	read the record into the buffer
Y = pattern	read the record into the buffer, then compare the buffer with a pattern.

The size and location of the buffer is determined by the registers .SIZE and .BUFF. To read a 100-word record without error, the value of .SIZE should be 100. The first location for data to be stored in is the address contained in C.(BUFF). To read a 100-word record into memory at locations 44000(8) to 44077, the values for .BUFF and .SIZE should be 44000(8) and 100(8). If the record is shorter than the value of .SIZE, a short length record error message occurs. The value of .SIZE and .BUFF will remain unless changed by the initialize routine or the exerciser routines. If this is a TM10/B system, the registers .CB and .CS are used to define a buffer for channel commands. See the write UUO for further details of TM10/B operations. If a parity error is detected and data switch 23 is 1, the routine will not try to reread the record. If however, switch 23 is 0 and the monitor is not doing retries, the routine will try to reread the record up to 64(10) times. If the error persists after 32 retries, the remaining 32 retries will be attempted using the low threshold setting. When the routine is trying to reread the record, error messages are suppressed. They are never suppressed on the initial reading of a record. The read routine normally returns to the location +1. If the end of tape flag is encountered, an immediate return to +2 is given with no error checking or reporting.

Before reading, the buffer is filled with the contents of location fill (initially 400000000021).

10. RRTEST<CR> or RTEST<CR> - This is the prime test of data reliability. If the program is in exec mode, the program determines which transports are in the ready state. It then tests the write-lock ring. If the ring is out (transport write-locked), the message WRITE RING IS OUT--XPORT NUMBER X will be printed. If in user mode, the transports assigned by the operator are tested for a write-lock condition. If they are write-locked, a message will be printed. Each tape available for use by the test is rewound to load point at the beginning of the test; the conditions of the test are specified at that time.

A random number is generated, three bits of which are used to define a transport number. If the transport selected is not available to the program, another number is generated and tested until an existing transport is found. A second random number specifies 1 to 7 records between file marks. A random data pattern is then selected. If odd parity, the selection is from the patterns listed in Table 2. If even parity, the last nine patterns are never selected. The records are of random lengths (alternately modulo 777(8) and 7777(8) if switch 27 is reset, or 777(8) and up to 16K of memory if switch 27 is set), and at a density determined by switches 21-22.

When the records have been written, a file mark is also written. The tape is then spaced backward over the file just written and the file mark preceding the records. Tape is then spaced over the file mark preceding the records. This leaves the tape head just forward of the first record in the file.

The random numbers are restored to the values used when the file was written. The records are read and the data compared with a random number generator. At the completion of a read operation, any errors present in the memory buffer will be printed. When all the records in the file have been read, the file mark will be spaced over. The data switches are then read and bit 2 is tested. If it is set, data accumulated during the test is printed and the transports are rewound. The test continues. If switch 2 is 0, the units are tested for the end of tape flag. When all transports are at end of tape, the procedure specified by switch 2(1) is executed. If however, switch 2 is 0 and the tapes are not at end point, switch 19 is tested. If it is 0, the test continues with new random numbers. If it is a 1, the random numbers are repeated.

11. SPACET D,N<CR> - The SPACET subroutine is a UUO where D is the desired drive number in exec mode and the desired channel number in user mode. Argument N, bits 18-35 of the UUO is the command. These commands are available:

N > 0	space forward N records
N < 0	space reverse N records
N = 0	rewind
N = EOF	space forward one end of file
N = -EOF	space reverse one end of file

The SPACET 0 command has been defined, for DDT, as REWIND. The routines will exit when spacing is complete except for REWIND, which exits as soon as the job done flag is set.

The space commands attempt to keep track of the position of the tape by counting records and file marks between the heads and BOT. However, it is impossible to count records when spacing to a file mark; the SPACE file command therefore uses the contents of location SAVCNT as the number of records and file marks that pass the head. This figure must be supplied before each SPACE file command to maintain accuracy of the position counter.

12. TAPE N<CR> - This routine is provided to closely examine how the data should appear when written on tape. Data is taken from the data buffer as defined by the registers .BUFF and .SIZE. The argument N determines the drive type and mode as follows.

TAPE 0	display as written on seven-channel transport
TAPE 1	display as written on 9-channel transport in dump mode
TAPE 2	display as written on 9-channel transport in industry-compatible mode

If not industry-compatible mode, parity is determined from data switch 20. The display will first contain a description of drive type, mode and parity. This is followed by three columns. The first column contains the data byte as it appears in the buffer. The last byte of each 36-bit word is preceded by a decimal point. The second column contains a picture of the tape and a 1 in each position where a 1 is written on tape. The third column displays the direction of magnetic flux in each track. The second and third columns are printed in terms of track number.

13. WRITE D,Y<CR> - The WRITE subroutine is a UUO where D is the desired drive number in exec mode and the desired channel number in user mode. The Y argument in bits 18-35 of the UUO may contain the following:

Y = 0	write the buffer on tape
Y = pattern	generate a pattern then write it on tape
Y = EOF	write an end of file mark
Y = EOR	write with extended record gap.

The size and location of the buffer in memory is determined by the registers .SIZE and .BUFF. To write a 100-word record from memory location 44000, set C(.SIZE) = 100, C(.BUFF) = 44000. This setting would be applicable to other buffer operations such as GEN and COMP also. The values would remain until changed by the operator or such routines as RTEST where buffer size is a variable. If this is a TM10/B system being executed in exec mode, the registers .CB and .CS are used to define a buffer for channel commands. Data channel commands are permitted in the area between locations C(.CB) and C(.CB) +C(.CS). If the split inhibit switch (switch 24) is = 0, the control word defined by .SIZE and .BUFF is split up into N sequential control words. The number N is a function of the number of bits in C(.SIZE). If the spread inhibit switch (switch 25) is = 0, the several control words generated by split are spread throughout a buffer defined by C(.CB) and C(.CS). The control words generated are placed at random locations in the channel buffer as a function of the random base register .RAN. The only restriction imposed on the control words are that:

a. Their several values are equal to the values defined by .BUFF and .SIZE;

b. They are all within the area defined by .CB and .CS;

c. There are never two sequential channel jump instructions.

The WRITE routine will try to leave the tape written properly. Namely, if an error is detected, the routine will try to rewrite the record (if switch 28 = 0). The error condition in exec mode is bad tape or parity error. In user mode it is bit 19 of a STATO or STATZ. Each time the above error occurs, the routine will try to rewrite up to 64 times. Every eight retries the record will be written with an extended record gap. The routine backspaces over the previously written record and then writes forward. If write creep is a positive value, the record should move forward. If write creep is a negative value, the previous record may be destroyed or the transport may never be able to successfully read the record. When the routine is in a write retry procedure, all error messages are suppressed until the next record. The routine will always print the first error encountered in a given record however.

The WRITE routine normally returns to the instruction following it. If, however, end of tape is detected during the operation, the routine will return to the write instruction +2; i.e., the instruction following the write will be skipped. This return will occur immediately upon detecting end of tape. No write recovery will be attempted and no write errors will be reported. The following is a program to fill tape with records of all 1s.

```
PATCH!   REWIND        GO TO LOAD POINT ON XPORT ZERO
PATCH+1! GEN .ONES     FILL THE BUFFER WITH ONES
PATCH+2! WRITE         WRITE BUFFER ON TAPE
PATCH+3! JRST          DO IT AGAIN UNTIL DETECTED EOT FLAG
PATCH+4! JRST SRTDDT   FINISHED, EXIT TO DDT
```

ERROR SUMMARY

Errors are printed on the terminal or line printer as a function of data switch 4.

Status Errors

Status errors are reported upon occurrence. All functions (WRITE, READ, and SPACE) check all status flags after each operation. If any flag is not in its expected state, an error message will be printed. Errors are suppressed when doing error retries.

The first line of the printout attempts to point to the flag which is in error. The possibilities are as follows.

PARITY ERROR - A parity error or bad tape flag is set.
END OF FILE MARK - End of file was set on a read operation.
STATUS ERROR - All others.

This error description is followed by the unit number under test, the operation just performed, and the current record number. In exec mode the status bit returned by a CONI MTC, and a CONI MTS, are then printed. If data switch 10 is reset, these bits will be followed by a description of all bits in the 1 state.

In user mode, the physical name of the tape unit is printed followed by the status bits returned by GETSTS and DEVSTS UUOS.

A sample exec mode printout follows.

```
     PARITY ERROR DRIVE #0      (READ) - RECORD #1
     MTC = 000000 062200        WHICH EQUALS:
     CM FUNCTION = 'READ'
     DENSITY = 800 BPI
     MTS = 000001 020200        WHICH EQUALS:
     CHARACTER COUNTER = 1
     PARITY ERROR, BAD TAPE
```

Data Errors
Data errors print out a correct word, an incorrect word and a discrepency word. The 36-bit words are printed as six 6-bit bytes for 7-channel drives, or five 8-bit bytes for 9-channel drives in core dump mode, or four 8-bit bytes and a 4-bit byte for 9-channel drives in industry-compatible mode. The four-bit byte represents the parity error bits on a read. Only the first three error words in a record are printed unless switch 8 is set.

As data errors are encountered, a check is made to see if the errors are on only one or two tracks in the transport. If such a case is seen, a bit PICK/DROP summary will be printed after the data error message. This PICK/DROP summary is then added to an accumulative summary which can be printed via the INFO command.

This printout states the bit number followed by the number of times that bit was picked and dropped. The bit number is the position of the bit in the data byte, the left-most bit being 0. For example, the 9-track data byte 377 was read as 373. The number of the bit that was dropped is 5.

Numbers in error printouts are always in decimal except when the number is a representation of data as stored in memory, which is printed in octal.

EOF Errors
End of file errors are reported as follows:

```
     UNIT#  START  DEST.  FINISH  DIFFERENCE

     3      037    105    116     .-9
     0      026    020    023     +3
     2      005    020    020     +1

     UNIT# 2 HIT LOGICAL END OF TAPE

     UNIT#  START  DEST.  FINISH  DIFFERENCE
     2      153    174    000     -4
```

In the first example, unit number 3 was trying to go from record number 37 to number 105, which means the unit would have to space to EOF 46(8) times. Because it stopped in front of record number 116, it must have missed 9 EOFs along the way.

In the second example unit number 0 was trying to space reverse to EOF six times. It appears that every record is being detected as an EOF.

In the third example the correct record was found, but it must have taken two reads to get there.

Examples

The following are examples of using DDT and Exerciser Routines and Subroutines to constant-short programs for testing and debugging. To assure that the main code of DDTMH is not disturbed, begin the program at symbolic location PATCH.

Loop on Read
This program is suitable for scoping the data path from tape t
memory.

```
PATCH!      REWIND 5,0          ;REWIND UNIT 5
PATCH+1!    WRITE 5,.ONES       ;WRITE A RECORD OF ALL ONES
PATCH+2!    SPACET 5,0-1        ;CONTINUALLY READ THE RECORD
PATCH+3!    READ 5,.ONES        ;AND CHECK FOR ERRORS
PATCH+4!    JRST .-2
```

Write Ones
The following is a program to fill tape with records of all 1s:

```
PATCH!      REWIND              ;GO TO LOAD POINT ON XPORT ZERO
PATCH+1!    GEN .ONES           ;FILL THE BUFFER WITH ONES
PATCH+2!    WRITE               ;WRITE BUFFER ON TAPE
PATCH+3!    JRST .-1            ;DO IT AGAIN
PATCH+4!    JRST DDT            ;WRITE DETECTED EOT FLAG, FINISHED
```

DIAMOND
P 6,10

DDTUA

GENERAL INFORMATION

Code	DDTUA.A10
Title	PDP-10 TU70 Magnetic Tape Subsystem Diagnostic
Abstract	The TU70 Magnetic Tape Subsystem Diagnostic is a test of the DX10 data channel to TU70 subsystem interface. This is the first TU70 diagnostic that uses a TU70/TU71/TU72 tape transport and TX01/TX02 controller. Testing begins with the diagnostic program stepping through a selection sequence with the TX01/TX02 controller. This stepping sequence is performed using the IBus control of the TAG IN and TAG OUT lines. Test I/O and sense commands are checked in this manner. Then sense bytes are read from each tape transport until a unit is found to be ready and write-enabled. The DX10 is then loaded with its microcode (MAINDEC-10-DXMPA) and all operations are processed under its control. All command testing sequences are passed to the DX10 through DX10 channel programs written into PDP-10 memory, then status and data written by the DX10 are examined for accuracy.
Hardware Required	KA10, KI10 or KL10 mainframe/32K of core (minimum)/DX10/TX01 or TX02/ up to 16 TU70s, TU71s or TU72s
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module). DXMPA (microcode - version 4 or later) SPAR (Subsystem Program for Analysis and Repair)
Restrictions	1. The DX10 microcode file DXMPA.A8 must be present and accessible on the same device used to load this diagnostic. 2. There is one switch in the DX10 that will affect diagnostic operations. This is switch 2 in the feature register or feature register bit 19. This switch must be in the OFF position to allow the entire diagnostic to run. It should only be set to the ON position when the DX10 is connected directly to a KA10-style memory bus. The switch should never be set if more than 256K of memory is on the system. When this switch is set, the diagnostic will never attempt a memory reference with address bits 14 to 17 set. 3. There are several operations that are not tested in user mode because of the possibility that these operations may affect the performance of the system and/or other users' jobs. The following is a list of the operations not fully tested in user mode. The IOB reset signal Priority interrupts Memory addressing beyond the physical core assigned to the job
Notes	1. This diagnostic will run in both exec and user modes. When in user mode, the diagnostic will be locked in core and will run with the user IN-OUT bit set in the processor status word.

2. User Mode Privileges

   a. The following privileges are required to run this diagnostic in user mode.

      POKE
      PEEK
      LOCK
      TRPSET
      SET CPU (on dual-processor systems)

   b. All magtape units connected to the DX10 to be tested must be either assigned or assignable to the job running this diagnostic.

3. To force this diagnostic to stop and return the microcode to the DX10 memory, the abort must be requested by either setting switch 0 or typing an altmode.

Loading and Starting Procedure

This program must be loaded and started via DIAMON. It may be run as a single program or it may be chained with other programs using the DIAMON chain feature.

1. Mount a magtape of known good quality (with write ring) on each tape transport to be tested. Load the magtape and press start switch on transport. Make sure ONLINE/OFFLINE switch is in the ONLINE position.

2. Reset all tape transports not to be tested.

3. Set tape CONTROL UNIT ONLINE and ENABLE UNIT switch for each tape unit to be tested.

4. Set console data switches for desired program options as desired.

5. Load and start diagnostic. (Refer to the 10/10 STD module.)

6. Verify that memory map printed agrees with the memory configuration of the system.

7. Diagnostic will print unit address of tape transport it chooses to test. If this address is not the address intended to be tested, stop the diagnostic. Set data switch 31 and set switches 32-35 to the unit address intended. This will allow the diagnostic to report errors with the knowledge that the unit is supposed to be ready.

8. The diagnostic will load the microcode program from the load device or from PDP-8A memory (depending on switch 18) and continue. The microcode must be version 4 or greater.

Control Switches

Refer to Table 1.

OPERATIONAL CONTROL

Starting Addresses

30000 Normal starting address

30004 Force contents of DX10 memory to be read and saved as the microcode even if diagnostic already has the microcode in core, then start the diagnostic.

30006 In either mode, this address or REENTER command will abort test. If user mode, release the DX10.

30013 Read the microcode from the load device and save in memory even if diagnostic already has the microcode in core. Then start the diagnostic.

30014 Start PDP-8A console program.

Terminal Control Commands
Terminal control commands are described in Table 2.

PDP-8A Console Control Program
This program will enable the operator to control the PDP-8A microprocessor in the DX10 similar to operating a PDP-8 console. Control is through the terminal on the PDP-10.

This program is started by setting the address switches on the PDP-10 to 30014 and pressing start. The commands are described in Table 3.

Table 1 DDTUA Control Switch Summary

Switch	State	Description
0-17		Standard (Refer to the 10/10 STD module.)
		Exceptions - the following switches are not implemented: 9, 11, 13, 14, 15, 16, and 17.
18	0	Read microcode from the load device.
	1	Read microcode from DX10 memory and return to DX10 memory on test end.
19	0	Allow DX10 microcode to run its internal diagnostics when idle.
	1	Prevent microcode from running its diagnostics
20	0	Test tape units through DX10 #1 (device code 220).
	1	Test tape units through DX10 #2 (device code 224).
21	0	Normal operation
	1	Bypass end of tape test.
22	0	Print only three lines in data comparison error.
	1	Print all data comparison errors.
23	0	Print only status that is in error.
	1	Print all status.
24	0	Print status descriptions.
	1	Print status in octal numbers.
25-30		Not used
31	0	Test all tape units in system.
	1	Test only the tape unit whose address is selected by switches 32-35.
		NOTE **The use of this switch tells the diagnostic that the unit selected is in a ready and write-enabled state and to report an error if the unit's status does not agree.**
32-35		Address of tape unit to be tested if switch 31 is set.

Table 2 DDTUA Terminal Control Commands

Command	Description
$	<ALTMODE> An altmode character can be typed at any time to cause the diagnostic to halt at the end of the current test. This is the recommended method of stopping the diagnostic. An altmode typed to the PDP-8A console program will restart the diagnostic. A second altmode must then be typed to abort the diagnostic.
↑C	A control C in user mode will release the DX10 and abort the test.
↑O	A control O can be typed to suppress the printing of an error message. All typeout except forced typeout will be suppressed until a new error is encountered. The program continues and the suppressed typeout is lost.
↑Q	A control Q can be typed to continue the printing of a message stopped via a control S.
↑S	A control S can be typed to stop the printing of an error message on a video display terminal without losing the rest of the typeout. Type a control S before the first line of the message disappears off the top of the screen and read the message. Then type a control Q to allow the printing to continue. While typeout is stopped with a control S the program will not run.

Table 3 PDP-8A Console Control Commands

Command	Description
$	<ALTMODE> An altmode may be typed at any time to restart the diagnostic.
A	A<CR> The contents of the AC is typed as data.
^	^<CR> The address in the CPMA is typed.
adr.	177.<CR> Load the CMPA with the address specified.
adr/	200/ Print the data in the specified address. If no address is typed, the address in the CPMA is assumed. The terminal remains in this position waiting for another command. The four options follow. <CR> No operation. Exit the examine routine. 7402<CR> Deposit the data in the selected address. Then exit the examine routine. <LF> Examine the next sequential location in PDP-8 memory and again wait for another command. 2014<LF> Deposit the data in the selected address. Then examine the next sequential address in PDP-8 memory and wait for command. Any other command may be typed. The program will exit the examine routine then process the new command as normal.
adr1:adr2/	200:250/ The contents of every address starting with adr1 through adr2 is typed. If adr1 is omitted (only the colon typed), adr1 is assumed to be 0000. If adr2 is not typed (no number between the colon and slash) adr2 is assumed to be 7777.

Table 3 PDP-8A Console Control Commands (Cont)

Command	Description
adrZ	272Z<CR> Clear the address specified.
adr1:adr2Z	100:300Z<CR> Clear all locations starting with adr1 through adr2. Unspecified addresses are treated in the adr1:adr2/ command. Also see the Z command.
C	C<CR> The position of the continue switch is typed as data (0 or 1).
0C	0C<CR> Clear the continue switch.
1C	1C<CR> Set the continue switch. If the switch is already set, it is cleared and set again.
Code I	4I<CR> The content of the specified IBus register is typed as data. If no code is typed, IBus selection code 0 is assumed. The terminal waits here for another command. The options are the same as for the slash command.
D	D<CR> The content of the data bus is typed as data.
H	H<CR> The position of the halt switch is typed as data. Data will either be 0 or 1.
0H	0H<CR> Clear the halt switch.
1H	1H<CR> Set the halt switch.
L	L<CR> Clear all of PDP-8A memory, clear all PDP-8A switches and, if this program was loaded with DIAMON, ask FILE NAME? -. At this time the name of a file must be typed. The file will be read from the same device from which DIAMON loaded this program. If this program was loaded from paper tape on a KA10 or KI10, the paper tape in the paper tape reader will be read. The file is loaded into PDP-8A memory and PDP-10 memory, then the two memories are compared. Any differences are reported as verify errors.
M	M<CR> The content of the MQ register is typed as data.
P	P<CR> The content of the processor status register is typed as data.
R	R<CR> Reset the DX10.
S	S<CR> The position of the single-step switch is typed as data (0 or 1).
0S	0S<CR> Clear the single-step switch.
1S	1S<CR> Set the single-step switch.
Z	Z<CR> Clear all of PDP-8A memory.

DDTUA TEST SUMMARY
The following tests are executed in sequence by DDTUA.

SELECTION SEQUENCE

1. Reset all TAG OUT lines and check that all TAG IN lines are clear.
2. Set OPL OUT and check for no activity on TAG IN lines.
3. Set OPL OUT, HLD OUT, then SEL OUT and wait for SEL IN. No device should respond, so SRV OUT should return as SRV IN.
4. Attempt a device selection sequence and look for controller to respond correctly with OPL IN.
5. Attempt test I/O command on control unit.
6. Load and start the DX10 microcode.

STORING STATUS

1. Set STAT request and check that something is written into memory. Check that byte counter is 0.
2. Execute STORE and HALT command and check that CPC is stored correctly.
3. Execute STORE, JUMP and HALT instruction and check for correct CPC stored in status word.
4. Execute a JUMP command to a STORE and HALT command.
5. Set up channel program with two store commands. Check that the DX10 can be continued after the first.
6. Execute a transfer command and check that sequence error sets.
7. Test that status is stored correctly by a NOP command.

EXTENDED STORE

1. Test extended store using NOP command.
2. Test that extended store writes into word 1 showing 0 record length.
3. Test that extended store writes into word 2 and correct data is written (where possible).
4. Test that extended store writes into word 3 by looking at bits 0-13.
5. Test that extended store writes into word 4.
6. Test that a byte count greater than 24. will cause a sequence error in status word 1.
7. Test that requesting less than 24. sense bytes will terminate correctly and will not write into too many words.
8. Set inhibit length error in NOP command and test that it is stored in status word 1.
9. Test that SEL ERR will set if an illegal device address is requested.
10. Test that DX10 can store extended status anywhere in existent memory.
11. Test that DX10 will attempt to store extended status in all of nonexistent memory. Check that nonexistent memory sets.

CHANNEL PROGRAM COUNTER

1. Test that DX10 will JUMP to all existent memory addresses.
2. Test that DX10 will attempt to JUMP to nonexistent memory.

COMMAND CHAINING

1. Test that several NOP commands can be chained.
2. Test that several NOP and channel JUMP commands can be chained.

REWIND

Test rewind for two interrupts.

TIO COMMAND

Test TIO command.

ERASE GAP COMMAND

Test that erase gap command moves tape from load point.

REWIND STATUS

1. Test that rewind moves tape back to load point.
2. Test device busy by writing four gaps followed by a rewind and a NOP. Extended store will be requested but sense error should set.

BACKSPACE BLOCK AT LOAD POINT

Test that backspace block at load point sets unit check.

BACKSPACE FILE AT LOAD POINT

Test that backspace file at load point sets unit check.

WRITE TAPE MARK

Test write tape mark.

BACKSPACE BLOCK OVER FILE MARK

Test that backspace block over a file mark sets unit exception.

MODE SET 2 COMMANDS

Test mode set 2 commands.

END OF TAPE TEST

1. Write file marks on all of tape. Check that end of tape sets.
2. Test that backspace file will clear unit exception.

DATA SECURITY ERASE

Test that data security erase will move tape to EOT. Then test that entire tape was erased.

SENSE COMMAND

1. Test that SENSE command can read 24. bytes without error.
2. Test that length error sets when too few bytes are requested by sense.
3. Test SENSE command to all of existent memory.
4. Test SENSE command to all of nonexistent memory.
5. Test SENSE command with two transfer words.
6. Test SENSE command with three transfer words.
7. Test SENSE command with four transfer words.
8. Test SENSE command with too many transfer words.
9. Test SENSE command with JUMP between SENSE and XFER commands.
10. Test SENSE command with JUMP between XFER and STORE commands.
11. Test SENSE command with no transfer word.

LOOP WRITE TO READ TEST

1. Test loop write to read in PE mode and 7-track NRZI.
2. Test loop write to read in 9-track NRZI mode.

WRITE TEST

1. Test device write operations in PE mode.
2. Test device write operations in NRZI mode.

READ FORWARD IN PE

1. Test device read forward operation in PE mode .
2. BYTE MODE
3. CORE DUMP mode
4. ASCII mode
5. SIXBIT mode

READ BACKWARD PE

1. Test device read backward operation in PE mode.
2. BYTE mode
3. CORE DUMP mode
4. ASCII mode
5. SIXBIT mode

READ FORWARD IN NRZI

1. Test device read forward operation in NRZI mode.
2. BYTE
3. CORE DUMP
4. ASCII
5. SIXBIT

READ BACKWARD IN NRZI

1. Test device read backward operation in NRZI mode.
2. BYTE
3. CORE DUMP
4. ASCII
5. SIXBIT

PARTIAL WORD TRANSFER TEST

1. Check writing and reading records not ending on word boundaries.
2. BYTE mode
3. CORE DUMP mode
4. ASCII mode
5. SIXBIT mode
6. BYTE mode
7. CORE DUMP mode
8. ASCII mode
9. SIXBIT mode

LARGE TRANSFER TEST

1. Check the largest transfer possible with one transfer word in all modes.
2. Check the largest transfer possible with two transfer words in all modes.

SKIP READ TEST

Test that data is not written into memory by a skip read transfer command.

ERROR SUMMARY

Errors are reported on the terminal or the line printer (if switch 4 is set). An error report contains the program pass count if not 0, the PC of the error call, the console data switches and the test name. If the error occurs before the microcode in the DX10 is used, the error message may also contain CORRECT and ACTUAL data.

Tests that use the DX10 microcode to perform the magtape operation check if the operation was performed correctly by examining the status stored by the DX10 into PDP-10 memory. Any error in this compare may be reported in one of the following four ways.

1. If console switch 24 is set the status will be printed as octal numbers. The actual status received is printed and, if it is not correct, the expected follows it. Each group of status is clearly identified.

2. If data switch 24 is reset, the status received will be described as closely as possible to the DX10 and STC documention.

3. If console data switch 23 is set, all status that was stored in memory will be printed.

4. If data switch 23 is reset, only the information in error is printed.

The status information is followed by any data pattern or data comparison error information, if applicable to the test. Then a description of the data channel program is printed to describe exactly what the DX10 was instructed to do.

GENERAL INFORMATION

Code	DDTUB.A10
Title	PDP-10 Magnetic Tape Subsystem Reliability Diagnostics
Abstract	This diagnostic is designed to exercise the TU70 magnetic tape subsystem to determine its reliability in a controlled set of tests with extensive error reporting facilities. This diagnostic is NOT designed to be used as a trouble-shooting tool for corrective maintenance. The TU70 subsystem has a full set of such diagnostics which are assumed to have been run prior to the running of this diagnostic. However, this diagnostic does have the facilities to allow the operator to generate his own test loops when found necessary. This diagnostic assumes that the operator wants to know about every error that is detected by the diagnostic. Upon the occurrence of each error a full error description is printed. After the error report, an attempt is made to recover from the error unless such action is inhibited by switch option. The diagnostic will then continue even though the error may lead to other errors as the test proceeds, such as write errors will normally lead to read errors when the record is read.
Hardware Required	KA10, KI10 or KL10 mainframe/32K of core (minimum)/DX10/TX01/TX02/up to 8 TU70s, TU71s, or TU72s
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	1. Either the DX10 must contain the microcode program (MAINDEC-10-DXMPA) or the file DXMPA.A8 must be on the file storage device from which DDTUB was loaded. 2. The microcode must be version 2 or greater. The version is checked by looking for a key word in location 16. The key word is "70" in SIXBIT. 3. In USER mode, a monitor of version 6.02 or greater is required, or an equivalent which supports the TAPOP, UUO set function code 2030.
Notes	1. DDTUB runs in either user or exec mode. 2. DIAMON must be used to load this program. 3. Subroutine must be in core before this program is loaded. 4. The DX10 and TX01/TX02 must be on and in the ready state. 5. If a drive drops off-line while being tested, the diagnostic will attempt to continue the test with the rest of the drives that were under test. The drive that dropped off-line will be picked up again at the start of the next test if the problem has been corrected. 6. A test can be aborted at any time by typing an altmode on the terminal. When an altmode is typed, the test is restarted at the test selection routine which will either ask what test to run next or will run the test selected by the console switches.

Loading and Starting Procedures

Exec Mode Operation

1. Mount tape of known quality on the transports to be tested.

2. Place the transports to READY, WRITE ENABLE and REMOTE.

3. Set the appropriate data switches. Refer to Table 1.

4. Start the program at location:

   30000 Normal starting address

   30004 Starting the diagnostic at this address will cause the contents of the microprocessor memory to be read and used as the microcode program even if the diagnostic already has a copy of the microcode in PDP-10 core memory.

   30006 Abort the diagnostic and close any output list file in user mode.

   30013 Starting the diagnostic at this address will cause it to read a fresh copy of the microcode into PDP-10 memory. Switch 18 will determine from where to read the microcode.

User Mode Operation

1. Gain access to the time-sharing system.

2. Assign transports to be tested.

3. Mount tape of known quality.

4. Place the transports to READY, WRITE ENABLE and REMOTE.

5. Set the data switches if desired.

6. Type RUN DDTUB (core argument optional - program requires a minimum of 32K). If program cannot obtain 32K it will exit to DIAMON immediately.

Control Switches

Refer to Table 1.

OPERATIONAL CONTROL

In User Mode

The diagnostic assigns each drive to a software channel numbered 0-7. When the assignment of a drive to a software channel is made, the assignment is printed as follows.

```
DRIVE 0  MTB3  9-TRACK  WRITE-LOCKED
```

This example describes a connection to software channel 0, the magtape drive MTB3. The drive is 9-track and write-locked. Write-locked drives will not be tested or rewound unless specifically selected by the operator. Only TU70, TU71/TU72 tape drives will be assigned to a software channel. Any other drives assigned to the job will be ignored. If no TU70 tape drives are assigned to the job the diagnostic will print:

```
NO TU70 DRIVES ASSIGNED TO THIS JOB.
PROGRAM CANNOT CONTINUE.
```

And control will be returned to DIAMON.

When a user mode job is started and every time it is restarted the following questions are asked:

```
WANT MONITOR TO ATTEMPT ERROR RECOVERY?
Y OR N <CR>
```

The operator's answers will affect all drives to be tested. If monitor error recovery is selected, the monitor will attempt to recover from all errors before informing the diagnostic that an error occurred. Therefore, only nonrecoverable errors will be reported. If monitor error recovery is not selected, the monitor will do only the operations specified by the diagnostic and every error will be reported. Then the diagnostic will use console switches 25 and 26 to determine if any error recovery should be attempted.

WANT ERRORS LOGGED TO SYSERR? Y OR N <CR>

Y - will cause errors detected by the program to be logged for future reporting by SYSERR. This assumes the timesharing system (monitor) has the SYSERR reporting feature for magtape.

N - the monitor will not log error for SYSERR use.

In Both Exec and User Mode

When the test is started, the program name and version number will be printed, followed by a memory map of the system. The diagnostic will then ask for the time of day by printing:

TYPE TIME AS HHMM

Next the state of all the drives on the system will be printed. Testing will begin automatically unless test selection is to be specified by terminal input.

Unless test selection is via the switches, the following question is asked.

WHAT TEST?

Any test listed in Table 2 may be selected. Only the first letter of each test name need be typed, although up to six letters will be accepted. HELP may be typed to get a list of the test names.

The next question asked will be:

WHAT DRIVE(S)?

Respond by typing in the drive number of the drive to be tested. If more than one drive may be tested by selected test, more than one drive number may be typed if they are separated by commas, or ALL may be typed to test all ready and write-enabled drives.

Table 1 DDTUB Control Switch Summary

Switch	Description
0	Abort program on test completion When this switch is set at the end of a test, the diagnostic returns control to DIAMON instead of restarting the test or starting a new test. This is the recommended method of aborting this diagnostic.
1	Print totals immediately, then abort the current test Setting this switch while a test is being executed will cause the performance statistics of all the drives to be printed immediately. The performance statistics are identical to the statistics printed by the test. After printing the statistics, the current test is aborted.
2	Print totals immediately, then continue with test Setting this switch causes the same action as switch 1 except that the current test is then continued.
3	Inhibit all print/typeout (except forced) This switch will inhibit all printout except forced. Forced printout normally consists of questions that require operator response. This switch is useful when it is desired to force a test to run without printing error information. To suppress the remainder of a current error report, it is recommended that a control O be typed on the terminal. This will suppress the remaining error information for the current error and allow printing to start again for the next error.

Table 1 DDTUB Control Switch Summary (Cont)

Switch	Description
4	Print all data on line printer (in user mode, print on logical device DEV or DSK). When this switch is reset, all printouts from the diagnostic will be printed on the terminal. If this switch is set, only forced printouts will be printed on the terminal. All data, including forced printouts, will be printed on the line printer if in exec mode. In user mode, the data will be output to a file called DDTUB.PNT on device DEV which can be any output device on the system available to the job. DEV can be assigned to any device with the ASSIGN command. If DEV has not been assigned as the logical name of any device, the data will be output to the DSK.
5	Ring bell on error This switch will cause the terminal bell to be rung every time an error occurs.
6	Not used
7	Not used
8	Print all data comparison errors Normally only three lines of data comparison errors are printed for any one error. It is usually possible to determine the pattern of the failure by these first three lines. If it is desired to see more of the words in error, this switch should be set.
9	Increase the execution time of nonoperator intervention tests This switch will cause certain tests to alter from their normal testing procedures to increase the amount of testing performed by the test. The tests that use this switch are LOOP, DATA, MEMORY, EOP and RANDOM. Refer to the description of each test for specific information as to how this switch alters the testing sequence.
10	Not used
11	Not used
12	Modify DX10 device code on program startup This switch should be set when it is desired to test a DX10 in exec mode which has a device code other than 220 or 224. When the diagnostic is started with this switch set, questions are asked on the terminal to allow the operator to select a device code for the DX10 to test. The program is initially set to test a DX10 with a device code of 220. The use of this switch actually causes the program to be modified.
13	Not used
14	Not used
15	Not used
16	KA10 only. Line frequency is 50 Hz. This switch is used only by a KA10 processor to determine the frequency of the ac power supplied to the processor. The KI10 and KL10 processors supply this information automatically. This switch should always be 0 on a KA10 if the line frequency is 60 Hz, and 1 if the line frequency is 50 Hz.
17	Not used

Table 1 DDTUB Control Switch Summary (Cont)

Switch	Description
18	Retrieve the DX10 microcode from DX10 memory When the diagnostic is first started in exec mode, it must load the DX10 with its microcode. This switch is used to tell the diagnostic where to find the microcode program. If this switch is reset, the diagnostic will read a file called DXMPA.A8 which should always be on the file storage device from which this diagnostic was loaded. If this switch is set, the diagnostic reads the contents of the DX10 memory. The diagnostic requires a microcode of version 1, and if the contents of the DX10 do not reflect version 1 of the microcode, the DXMPA.A8 file is then read. This switch should be used when the DX10 is known to already have the microcode loaded into its memory. The diagnostic reads the microcode once, then it holds a copy of it in core for use if the diagnostic is ever restarted.
19	Disable microcode self diagnostics The DX10 microcode normally runs diagnostics on the DX10 while in its idle loop. If this switch is set, these self-diagnostics will be disabled by starting the microcode at the proper optional start address.
20	Test DX10 No. 2 (device code 224) This switch tells the diagnostic which DX10 to test in exec mode. When the switch is reset, the diagnostic will test a DX10 with device code 220. When the switch is set, a DX10 with a device code of 224 will be tested. If any device code other than 220 and 224 are desired, switch 12 should be set and this switch should remain reset.
21	Inhibit memory cycles during data transfers When a data transfer is in progress between the magtape and memory, the diagnostic tries to keep the memory module busy by cycling through the data buffer itself. On KI10 and KL10 processors, the diagnostic causes read-modify-write memory cycles in the data buffer area. A KA10 processor causes read cycles because the fast memory may not be enabled. This action may be inhibited by setting this switch.
22	Print headers in COMPAT, INTER and SKIP tests. Print elapsed time in OPR test. This switch is used by the COMPAT, INTER and SKIP tests to cause a description of the records being read from the tape to be printed. This switch is also used to cause the elapsed time of the selected sequence of operations to be printed. Refer to the description of the individual tests for specific information.
23	Print data comparison errors as 36-bit octal numbers Data comparison error printouts normally break the data word into the bytes that are written onto the magtape. This format makes it easier to analyze errors that occurred because of a magtape subsystem problem. The following is an example of how a data word of all 1s will be printed when read from the magtape in each mode: DUMP MODE 377 377 377 377 37 (LAST BYTE CONTAINS BITS 32 THROUGH 35) BYTE MODE 377 377 377 377 (BITS 32 TO 35 WILL BE PRINTED AS IN DUMP MODE IF NOT ZERO) ASCII MODE 177 177 177 177 377 (BIT 35 IS DISPLAYED AS THE 200 BIT IN THE LAST BYTE) SIXBIT MODE 77 77 77 77 77 77 Errors which occur between the DX10 and memory can best be analyzed by looking at the 36-bit word. Therefore, this switch will cause the data to be printed as a 36-bit word regardless of the data mode. The data word is printed in 12 octal digits or 777777777777 for a word of all 1s.

Table 1 DDTUB Control Switch Summary (Cont)

Switch	Description
24	Print DX10 status information as octal numbers The diagnostic normally attempts to describe the status bits that are stored or received from the DX10. Because there are many status bits stored by the DX10, this printout gets quite lengthy. If it is desired to receive the status as octal numbers, this switch should be set.
25	Inhibit write error retries Except when running in user mode and the monitor has been allowed to attempt error recovery, this switch determines if the diagnostic should attempt error recovery on a write error. If this switch is set, no error recovery will be attempted. If this switch is reset, the write operation will be retried up to 64 times, each time erasing an additional three inches of tape, in an attempt to write the record correctly. If a write retry is successful, the retry number is printed after the error description. If the diagnostic fails to write the record correctly after 64 retries, nonrecoverable will be printed.
26	Inhibit read error retries This switch causes the same action as switch 25 except that this switch is used when a read error occurs.
27	Change PI channel to be used by DX10 The DX10 is normally connected to the processor PI channel 4. This switch may be used to connect the DX10 to another channel. When each test is started, this switch is tested. If the switch is set, the diagnostic prints the following. SPECIFY PI CHANNEL FOR DX10: (2 - 7) - Type the channel number to which it is desired to connect the DX10. Any channel from 2 to 7 may be specified.
28	Restrict testing to specific density and/or data mode This switch can be used to specify one density and/or data mode to be used by a test. If this switch is reset, the test will use each density and data mode as described in the description for that individual test. When this switch is set and any 7-track drive is to be tested, the following is printed. SPECIFY 7-TRACK DENSITY: (800, 556, 200, <CR> FOR ALL) - Respond by typing your selection of 800, 556, or 200. If just a carriage return is typed, 7-track drives will be tested at all three densities. When this switch is set and any 9-track drive is to be tested, the following two lines will be printed. SPECIFY 9-TRACK DENSITY: (1600, 800, <CR> FOR ALL) - SPECIFY DATA MODE: (DUMP, BYTE, ASCII, SIXBIT, <CR> FOR ALL) - Respond to the first line by typing your selection of 1600 or 800 BPI. If just a carriage return is typed, 9-track drives will be tested at both densities. Respond to the second line with the data mode that all 9-track drives should use. 7-track drives always use SIXBIT mode. These selections remain for the current test only, and must be specified each time a new test is started.
29	Select memory buffer address for test This switch can be used to cause a certain portion of memory to be used as the data buffer. Normally the data buffer starts just above the program and will increment up through all of memory each time the same test is run in succession. When a test is started and this switch is set, the following will be printed.

Table 1 DDTUB Control Switch Summary (Cont)

Switch	Description
	SPECIFY MEMORY BUFFER ADDRESS - Respond with any existent memory address. The data buffer will start at the first line in the page address selected if at least 4K of memory buffer will be available, or at the highest address below the address specified that will give a 4K word buffer. If an address is specified that is below the end of the program, the buffer will be placed immediately after the program. If a nonexistent address is specified: ADDRESS NONEXISTENT will be printed and a new address will be asked for.
30	Inhibit all read backward operations This switch will inhibit all read backward operations. The read backward will be bypassed or substituted with a space forward operation. This switch will speed up the execution time of most tests.
31	Run nonoperator intervention tests until end of tape is reached This switch will cause each test to continue execution until physical end of tape is reached on each tape. Exceptions are listed in the individual test descriptions. When this switch is set, only one density will be used for each drive as the density cannot be changed in the middle of the tape. This switch is useful when long run time is desired.
32	Select test to be run via terminal questions When this switch is set, tests will be selected from terminal input. This allows the operator to run any test of his choice in any order he wishes. This is the only method by which the tests requiring operator intervention may be selected. Also, this method of test selection allows the operator to choose which drives are to be tested. When this switch is 0, test selection is determined by switches 33-35 and all ready and write-enabled drives are tested.
33-35	These three switches select a test number if switch 32 is not set. Test numbers are: 0 All of the following tests 1 Loop write to read test 2 Data test 3 Memory test 4 End of file mark test 5 Random test A test selected in this manner will use all the drives on the system that are ready and write-enabled. When a test completes, the switches are checked again and if the same test is selected the data buffer is moved upward in the available core area. These switches may be changed during the execution of one test to select a different test upon its completion.

DDTUB TEST SUMMARY

Table 2 summarizes the tests and routines available in DDTUB. Table 3 summarizes the commands which may be used with the OPERATOR routine.

Table 2 DDTUB Test Summary

Test	Description	Cross Ref.
ALL	Runs LOOP (exec mode only) DATA, MEMORY EOF RANDOM and TOTALS.	1
COMP	Test tape drive compatibility	2
DATA	Test data reliability of each tape drive	3
EOF	Test drive's ability to read EOF marks	4
INTER	Test the interchangeability of each tape drive on a given system	5
LOOP	Test the loop write to read function of the TX01	6
MEMORY	Test the DX10 ability to address all memory locations	7
RANDOM	Randomly selects and exercises each drive on the system	8
NOTE	The following are routines available in DDTUB.	
OPR	Used to write an operator-generated program. Refer to Table 3 for commands.	9
PRINT TAPE	Used to print the contents of tape	10
SKIP READ	Used to SKIP READ all records on tape	11
TOTALS	Prints a performance report for a drive or the entire tape system	12

Table 3 DDTUB Operator Generator Command Summary

Command	Description
BSF	Backspace file - space backward to the next file mark.
BSR	Backspace record - space backward over one record. Report an error if record is a file mark.
FSF	Forward space file - space forward to the next file mark.
FSR	Forward space record - space forward over one record. Report an error if record is a file mark.
LWR	Loop write to read - allowed in exec mode only. This command will be treated as a WRT command except that no tape is moved. This command is used to test the data path from memory to the tape drive. The data is looped back in the drive and then checked for correct parity.
REW	Rewind the drive to BOT.
RDB	Read backward - read one record in the reverse direction. Report a file mark as an error. Compare the data read as in RDF command.
RDF	Read forward - read one record and report an error if record is a file mark. If a data pattern other than 0 is specified, compare the data read with the specified pattern.
WRT	Write data on the tape. The data and record size will be specified in response to questions.
WTM	Write tape mark - write an end of file mark at the current position on the tape.
?	Print current command list.

DDTUB TEST DESCRIPTIONS

This section describes the tests and routines summarized in Table 2. The descriptions assume that no restrictions were placed on the density and data mode selections for any drives. This is done to simplify the description. However, if any restrictions are specified, the test sequence will differ only with respect to that particular parameter which is restricted. The same number of records will be written in the same order as described.

1. **ALL** - All Test is a dispatch control that calls a set of tests in a specified sequence. No testing is performed directly by this test and no start or end reports are given for it. ALL runs the following sequence of tests: LOOP (if exec mode only), DATA, MEMORY, EOF, RANDOM, TOTALS. The drives, densities, buffer address and PI channel selected at the start of ALL TEST remain in effect for all the tests called by ALL. This is the normal test to run to get complete reliability performance on the magtape subsystem.

2. **COMPATIBILITY** - The purpose of Compatibility Test is to determine the compatibility of each tape drive to other tape drives, regardless of drive type or system type.

   A known good magtape should be marked as the compatibility tape. This tape should be labeled 9-track or 7-track and what density. It is recommended that three tapes be used for full system compatibility testing, one for 9-track at 1600 BPI, another for 9-track at 800 BPI, and a third for 7-track at 800 BPI. The necessity for each tape must be determined by the system configuration.

   A compatibility tape consists of a number of "bunches" of records, each bunch containing a header descriptive record, a number of data records of different lengths, followed by a tape mark. If the bunch is the last bunch on tape, two tape marks will follow the bunch.

   Normal operation of the compatibility test should be to read and data compare all the bunches on tape (until two tape marks are seen in a row), erase the second tape mark, append another bunch, then a second tape mark.

   The bunch format differs between 7- and 9-track tapes. Both tapes contain identical header records as the first record of each bunch. This header record is always written in SIXBIT mode and each character is a DECsystem-10 type SIXBIT character. Each text string is six characters long, left justified, with unused bytes filled with space characters. If string 6 is nonzero, strings 6, 7, and 8 will be taken as a SIXBIT comment.

   The content of each text string is described below:

String No.	Text String	Example
1	Drive Type	TU70
2	DRive Number	3 or MTA3 in USER mode
3	PDP Type	PDP-10
4	Proc Ser NR	537
5	Date	021275 for FEB. 12, 1975
6	Comment	This i
7	Comment	s a co
8	Comment	mment.

   A 9-track tape bunch contains 151 records followed by one tape mark. The first record is a header record. The next 50 records contain 24 bytes per record. The first and 26th record contain the first 9-track data pattern, the second and 27th record contain the second 9-track data pattern, etc.

   The next set of 50 records contains 528 bytes per record, and the third set contains 12024 bytes per record.

   The 9-track data patterns are: 2, 1, 4, 5, 8, 9, 12, 14, 16, 18, 19, 20, 21, 22, 23, 24, 25, 32, 33, 34, 35, 36, 37, 38, 39, 46, 47, 48, 49, 50, 51, 52, 53.

   A 7-track tape bunch contains 121 records followed by one tape mark. The first record is a header record. The next 40 records contain 24 bytes per record. The next set of 40 records contains 528 bytes, and the third set contains 12024 bytes.

The 7-track data patterns are: 2, 1, 7, 10, 11, 13, 15, 17, 26, 27, 28, 29, 30, 31, 40, 41, 42, 43, 44, 45, 54, 55, 56, 57, 58, 59, 60, 61.

Only one drive may be tested at a time by compatibility test. Any attempt to select more than one drive will result in an error message. The tape drive need not be write-enabled, but for full compatibility testing it is recommended.

When compatibility test is started, the diagnostic will ask:

BEGIN NEW TAPE? Y OR N <CR> -

If this is the first time the tape is being used as a compatibility tape, answer YES and the test will begin writing on the tape immediately. Otherwise answer NO, and the diagnostic will ask:

READ ENTIRE TAPE? Y OR N <CR> -

A YES answer should normally be given here. This question is provided in case a record gets written onto the tape that cannot be read correctly. Answering NO will give the option of only reading a specified number of bunches from the beginning of the tape and then allow the bad bunch to be erased. The diagnostic asks

READ HOW MANY BUNCHES?

Type the number of bunches the diagnostic should read.

After reading the entire tape or the specified number of bunches, the diagnostic will ask:

WHAT NEXT? (REWIND, WRITE, ERASE) -

Only the first letter of one of the choices need be typed, followed by a carriage return. A REWIND will cause the tape to be rewound and the test will end. A WRITE answer will cause the diagnostic to write one bunch onto the tape, then backspace over it and read the bunch. WHAT NEXT? will then be asked again.

The erase option is available if it is desired to erase one or more of the records just read. This should be done whenever a bunch on the tape is found to be incompatible with the rest of the drives in the system and the problem has been corrected.

ERASE HOW MANY BUNCHES JUST READ? -

will be asked. Type the number of bunches you wish the diagnostic to backspace over. Once the tape is properly positioned a double file mark will be placed there to mark it as the end of the tape. WHAT NEXT? will then be asked again.

When a tape is being written for the first time (BEGIN NEW TAPE? was answered yes), the diagnostic must determine at which density the tape should be written. If a density was specified by setting switch 28, that density will be used unless the tape is 7-track and a density other than 800 BPI was selected. In such a case, the following warning will be printed:

7-TRACK TAPES ARE NORMALLY WRITTEN AT 800 BPI.
ARE YOU SURE YOU WANT THIS TAPE WRITTEN AT 556(200) BPI?

If answered NO, the tape will be written at 800 BPI. A YES answer will cause the tape to be written at the selected density.

If the drive is 9-track and no 9-track density has been specified, the diagnostic will ask:

WANT 9-TRACK TAPE WRITTEN AT WHICH DENSITY? (800, 1600) -

Respond by typing your choice. Only an 8 or 1 need be typed followed by a carriage return.

The first time a bunch is written on a tape in exec mode,

DATE:

will be printed. Type today's date for the header information. Type the date as MM-DD-YY. (Example: 6-17-75 for June 17, 1975). If the processor is a KA10, the processor serial number will be asked for also.

Every time a bunch is written on a tape,

TYPE COMMENT FOR HEADER -

will be printed. Type any message up to 18 characters long to be inserted into the header block of the bunch.

Switch 22 can be set while a compatibility tape is being read to cause the header data of each bunch to be printed as it is read. If this switch is not set, the header information will be printed along with any error report.

3. **DATA** - Data Test is designed to cause maximum data reliability testing on the tape drives. All drives are tested together and they can be modified by console data switches 9, 30 and 31.

4. **EOF** - This is a test of the ability of each drive to properly space forward and backward to file marks. The test starts at the initial densities and data mode.

5. **INTERCHANGE** - Interchange Test is a modification of the Compatibility Test for the purpose of more thoroughly testing the compatibility of many drives in a single system at one time.

   This test may be run on any number of drives at one time. All the drives must be write-enabled. This test will write and read compatibility type bunches on all the drives simultaneously. The normal sequence of operations should be to mount a good tape on each drive in the system. Then start the Interchange Test.

   FAST MODE? Y OR N <CR> -

   will immediately be asked. Fast mode will cause only 10 bunches to be read or written on each tape. A NO answer will cause the tapes to be written from BOT to EOT.

   WANT TO WRITE THE TAPES?

   will be asked. Answer YES to write on all the tapes. Answer NO only if the tapes have already been written by Interchange Test. On a YES answer, the diagnostic will ask what density you wish to write on each tape.

   After the tapes have been written,

   VERIFY TAPES? Y OR N <CR> -

   will be asked. A YES answer will cause the tapes to be read immediately.

   WANT TO ROTATE THE TAPES? Y OR N <CR> -

   will be asked. A NO answer would be given to end the test. Normally the tapes should be rotated from one drive to the next until all drives have read each tape.

   NOTE

   Tapes written on a 9-track drive should never be rotated to a 7-track drive, and 7-track tapes should never be rotated to 9-track drives. The diagnostic cannot differentiate between the tapes and many errors will undoubtedly result.

   Switch 22 can be set to report which tape is being read on which drive by printing the header records as they are read.

6. **LOOP** - Loop Test can be run in exec mode only because it performs only diagnostic functions not available in user mode. This test requires a magtape drive to be ready and write-enabled but does not write any data onto the tape. The loop write to read function of the TX01 tape controller is exercised, which is treated like a write operation by the DX10. Data is passed to the TX01, the TX01 causes the data to loop back internally from the write logic to the read logic and there it is checked for correct parity.

   This test runs in two parts. The first part tests only one drive at a time to create the maximum data transfer rate to a single tape drive. Each drive is selected in sequence starting from 0. After each drive has been exercised individually, then part 2 selects a data pattern and writes it onto all drives before selecting another data pattern.

7. **MEMORY** - Memory Test causes data transfers from all addresses of memory to ensure full memory connection to the DX10. This test ignores any memory buffer address specified by setting data switch 29 and uses only one density and data mode. If no density was specified by setting switch 28, 7-track drives are written at 800 BPI and 9-track drives are written at 1600 BPI in dump mode.
   This test writes a physical address pattern (number 47) from each 8K word buffer location from the end of the program to the end of memory. Each record is read backward then forward if switch 30 is reset, or spaced backward and read forward if switch 30 is set. One record is written from each buffer location unless switch 9 is set. Then four records are written.

   If switch 31 is set, the test is repeated until all tapes have reached EOT.

8. **RANDOM** - This test is designed to randomly select and exercise the drives in the system.

9. **OPERATOR GENERATOR ROUTINE** - This routine is provided solely for the purpose of generating sequences of tape operations on a magtape drive. This routine will be useful if a problem is detected in the magtape subsystem that is not detected by the basic magtape diagnostic and which cannot be easily diagnosed from the existing tests in this diagnostic. This routine will provide the necessary links into the diagnostic to generate any desired loop of operations. These loops can be for the purpose of isolating the error by changing the loop slightly and seeing if the error changes or remains. The loop might also be for scoping purposes, in which case switch 3 should be set to inhibit continuous error printing if desired.

   The routine starts by typing the list of acceptable commands that may be inserted into the command list. Refer to Table 3.

   The diagnostic asks for a command sequence to insert into the command list. Up to 35 commands may be typed, separated by commas, and terminated by a carriage return. For example, to cause the tape to write a data record, then a file mark, read the record backward then forward, then space over the file mark, the following would be typed:

   ```
   COMMAND SEQUENCE - WRT,WTM,BSF,RDB,RDF,FSF<CR>
   ```

   If just a carriage return is typed, the current contents of the command list is used. If it is desired to examine the command list, type a question mark (?) and the command list will be printed as follows:

   ```
   COMMAND SEQUENCE - ?
   CURRENT COMMAND LIST CONTAINS:
   WRT,WTM,BSF,RDB,RDF,FSF
   ```

   If the command list contains a WRT, RDF, or RDB command, two additional questions are asked. For each question, if only a carriage return is typed, the value is left unchanged. The first question is:

   ```
   DATA PATTERN: (0 - 70) -
   ```

   Any number from 0 to 70 is acceptable. If a 0 is typed, the current contents of the data buffer will be written by all WRT commands and no data will be compared by RDF and RDB commands.

Nine of the most commonly used numbers and their corresponding data value follow. A complete list can be found in the diagnostic abstract located on mirofiche.

62 RANDOM NUMBERS

63 PHYSICAL ADDRESS of buffer location (shifted left 6 places) e.g., address 277777 will store the data word 000027777700

64 000000000001, 000000000002, 000000000004, 000000000010 000000000020 . . .

65 777777777776, 777777777775, 777777777773, 777777777767 777777777757 . . .

66 IN DUMP FORMAT -
001 002 003 004 05, 006 007 010 011 12,
013 014 015 016 17 . . .

67 IN ASCII FORMAT -
001 002 003 004 005, 006 007 010 011 012,
013 014 015 016 017 . . .

68 IN SIXBIT FORMAT -
01 02 03 04 05 06, 07 10 11 12 13 14,
15 16 17 20 21 22 . . .

69 252525252525, 525252525252, 252525252525 . . .

70 This pattern can be used to specify any desired data. When this pattern is selected, the diagnostic will print:

TYPE DATA -

Type any octal number and this word will be placed in all buffer locations. If only a carriage return is typed, the pattern will remain unchanged.

The second question is:

BUFFER SIZE -

There are two ways to specify a buffer size: a positive number is used to specify the number of words in the data buffer; a negative number is used to specify the number of bytes in the data buffer. A 0 size is not permitted. The maximum size allowed is 8K words or the amount of core after the program, whichever is smaller. The size is initially set at this largest when the test is started.

After these questions have been answered, the diagnostic checks to see if the tape is at BOT. If not,

REWIND? Y OR NO <CR> -

is asked. A YES answer will rewind the tape before starting to execute the command list.

REPEAT TIMES -

will then be printed. Type the number of times to execute the command list. Just a carriage return, a 0 and a negative number will cause the loop to be executed forever. This is useful when a scope loop is desired. An altmode may be typed to stop the execution and input another command list.

Any time the command list causes the tape to write beyond the EOT marker on the tape, the tape is rewound and the command list is restarted. No error message is printed.

Operator-Generated Routines
The following examples show how the commands described in Table 2 can be used to write special test routines.

Example 1 - The following will cause the record of 128 words and all 1s data to be written continuously in a loop suitable for scoping.

COMMAND SEQUENCE - WRT<CR>

DATA PATTERN: (0 - 70) - 62<CR>

BUFFER SIZE - 128<CR>

REWIND? Y OR N<CR> - Y<CR>

REPEAT TIMES - <CR>

Example 2 - The following will cause a record of 1000 words to be written onto the tape and then it will be read both forward and backward 50 times.

COMMAND SEQUENCE - WRT<CR>

DATA PATTERN: (0 - 70) -70<CR>

TYPE DATA - 100010001000<CR>

BUFFER SIZE - 1000<CR>

REWIND? Y OR N<CR> - Y<CR>

REPEAT TIMES - 1<CR>

COMMAND SEQUENCE - RDF,RDB,BSR<CR>

DATA PATTERN: (0 - 70) -<CR>

TYPE DATA -<CR>

BUFFER SIZE -<CR>

REWIND? Y OR N<CR> - Y<CR>

REPEAT TIMES - 50<CR>

An altmode character can be typed at any time during this test to back up and enter another command sequence. This should be useful when an error is discovered in setting up a command sequence or when the operation is being performed but it is not desired to wait until the operation is repeated the specified number of times. If an altmode is typed as the first command in a command sequence, the test is stopped and any other test may then be run.

This test can also be used to measure performance of the TU70 tape subsystem by setting console data switch 22. With this switch set, every sequence of operations to which an exact number of repeat times was specified is timed. If the sequence is allowed to run to completion, the elapsed time will be printed as:

ELAPSED TIME 2:30.6 SECONDS

The above example states that the sequence of operation took 2 minutes and 30.6 seconds.

10. PRINT TAPE ROUTINE - This is a routine to print the contents of any tape. This test can be useful when there is any doubt as to what is on a tape, or when a better understanding of the format of a tape is necessary.

    Only one drive can be selected when running print test but the drive does not need to be write-enabled. The test starts by asking the following two questions to determine which part and how much of the tape should be printed.

    START PRINTING AT WHICH (FILE, RECORD)? -

    PRINT HOW MANY (FILES, RECORDS)? -

    Both questions can be supplied with two numbers separated by commas. If no comma is typed, the answer is the number of files. If a comma is typed first, the file number is 0.

Typing only a carriage return to the first question will cause the test to start printing at BOT. If any numbers are typed, the tape will be spaced over the number of file marks specified, then to the record number specified. For example, to cause the print to begin with the third record after the fifth file mark on the tape, type "5,3" in response to the first question. The test will never move a tape beyond a double file mark on the tape, and an error message will be printed if a double file mark is found while spacing to position.

The second answer specifies the number of files to print from the start point. After all the files have been printed, the records in that file are printed until either the number of records specified have been printed or another file mark is read. If only a carriage return is typed in response to the second question, printing will continue until a double file mark is found.

11. **SKIP READ ROUTINE** - This routine will skip read all the records on any tape and report any read errors that might occur in the process. A summary of the data on the tape may also be printed as the tape is read by setting console switch 22.

    The tape will be read from BOT until a double end of file mark is found. Only one drive may be selected at one time but the drive does not have to be write-enabled.

    If the drive is 7-track the density will be asked for unless specified by setting switch 28. If the drive is 9-track, the tape will be read in dump mode unless another mode was specified by setting switch 28.

    Data record summaries are begun with a header describing the record followed by the data in the record. Each word as read into memory is printed in octal. Four words are printed on each line if being printed on the terminal, eight words on each line if being printed on the line printer (switch 4 is set). Each line begins with the word number of the first word on the line.

    If the tape is 9-track, the data will be read in dump mode unless a data mode was specified by setting switch 28. If the tape is 7-track, the density will be asked for at the start of the test.

12. **TOTALS** - This is not really a test. No data is transferred to or from any tape drive. This is simply a drive performance reporter. Totals can be called at any time and any drive in the system can be selected. If more than one drive is selected, a report for each drive will be printed.

    Drive performance reports may be obtained by selecting the Totals Test or by setting console data switches 1 or 2 while a test is running. If the performance report is requested via the switches, a report of each drive on the system is given. When the Totals Test is run, any one or more drives may be selected.

    A sample performance report follows:

    PERFORMANCE STATISTICS FOR DRIVE 0 MTB3

REWINDS	40	FILE MARKS WRITTEN	2520
RECORDS WRITTEN	9878	BYTES WRITTEN	146767471
RECORDS READ FORWARD	72654	RECORDS READ REVERSE	63854
BYTES READ FORWARD	1277801430	BYTES READ REVERSE	1138463982
FILES SPACED FORWARD	34421	FILES SPACED REVERSE	34405
RECORDS SPACED FORWARD	29	RECORDS SPACED REVERSE	684
WRITE ERRORS	0	DX10/MEMORY ERRORS	0
STATUS ERRORS READ FOR	0	STATUS ERRORS READ REV	0
DATA ERRORS READ FOR	0	DATA ERRORS READ REV	0
WRITE RECOVERIES	0	NON-RECOVERABLE WRITES	0
READ RECOVERIES	0	NON-RECOVERABLE READS	0

ERROR REPORTS
Errors are reported as they occur. At that time all status stored by the DX10 is printed. This status is preceded by a descriptive line of the operation in progress at the time the error was detected, the drive being tested, the time of day the error occurred, the record number which the drive was operating on, the data pattern and mode.

When a tape error is detected during a read or write operation, the diagnostic first checks to see if error recovery is allowed and, if so, retries the operation up to 64 times. A line describing the success or failure of the retries is then printed following the error report.

Error reports will begin with one of the following headers.

```
NONEXISTENT MEMORY ERROR
MEMORY PARITY ERROR
MICROPROCESSOR ERROR
END OF FILE MARK ERROR
LENGTH ERROR
STATUS ERROR
DATA ERROR
```

NONEXISTENT MEMORY, PARITY and MICROPROCESSOR errors are reported as follows:

```
MICROPROCESSOR STATUS: CPMA=2345, MB=7000, MEMORY USER=DAC
CPC: 73674
DAC: 12525252
DR:  777777 777777

STATUS ERROR DRIVE 0 MTA0 (READ BACKWARD) 6:55:07
FILE # 1  RECORD # 1  DATA PATTERN # 1  DUMP MODE  1600 BPI

CONI STATUS: RUN, ICPC-450, STAT AVAIL, PAI=3
CHANNEL STATUS: PROG STAT, TYPE=ENDING, UNIT=0
DEVICE STATUS: CHAN END, DEV END
BYTE COUNTER: 0.
CHAN PROG COUNTER: 541
RECORD LENGTH: 20480.
TAG LINES: PAR OUT, OPL OUT
CHANNEL BUS IN: 000      BUS OUT: 014
DATA ADR: 103777
IDENTIFICATION: VERSION=1(7), SW 2, ECO LEVEL=04, SERIAL #04
SENSE  BYTES:  TU=READY,  TRACK  IN  ERROR=004,  1600  BPI,
BACKWARD,
3803 MODE, MODEL=5, CU FEATURES=7 AND 9 TRACK NRZI,
CO SERIAL=1232, CO EC LEVEL=5
```

The above example shows a read backward at 1600 BPI that read the record correctly but detected a single track in error during the read. This error is reported to show that the error correction in the TX01 was required to read the record correctly. For this particular error, no error recovery is attempted.

DATA COMPARISON errors are reported as follows:

```
DATA COMPARISON ERROR     BUFFER SIZE = 40 BYTES = 10 WORDS
BYTE    CORRECT DATA      ACTUAL DATA         ADDRESS
1       777 777 777 776   777 777 777 777     100000
5       777 777 777 776   777 777 777 777     100001
        BYTES IN ERROR = 2

        BIT               PICKS               DROPS
        001               2                   0
```

If more than three words are in error, only the first three words will be printed unless console switch 8 is set. If it is desired to have the data printed as 36-bit octal numbers as they appear in memory, set switch 23. The bit PICK/DROP information is useful only if the error involved a small number of the tracks on the tape. Therefore, this information is not printed unless only one or two tracks had any PICK/DROP errors.

Error Halts - This diagnostic is written to recover from all error conditions and continue testing. However, there are a few halts coded into the program to catch a completely lost program because of some unexpected malfunction:

ADDRESS	REASON
1010	Program not coded for exec mode operation
1011	Fatal push list pointer error
1012	Initial push list pointer error
1013	MUUO with LUUO handler wiped out
1014	DTE20 interrupt without DOORBELL
1015	DTE20 clock interrupt without flag set
1016	CPU initialization error
1017	End of program error
1020	Interrupt with LUUO handler wiped out

This diagnostic is coded with several fatal instructions which will cause

FATAL PROGRAM ERROR AT ADDRESS

to be printed and the diagnostic to halt. These fatal instructions are placed at the point where the diagnostic finds itself in a situation from which it does not know how to proceed. There are no known program deficiencies that allow a fatal error to be executed.

PROGRESS REPORTS
At the start and end of each test (including each individual test run by ALL) the test name and time is reported. A sample start report is:

```
**********************
* START * RANDOM TEST     7:44:39     TESTING 2 DRIVES
**********************
```

The test's name is printed followed by the time of day and the number of drives being tested. At the end of the test the number of errors detected during the test is reported. Any time a test is aborted before completion, either because all the drives under test dropped off-line or because the operator aborted the test with switch 1 or by typing an altmode, an abort report is printed as follows:

```
**********************
* ABORT * RANDOM TEST     7:45:02     2 ERRORS
**********************
```

EXECUTION TIMES
The time for execution of each test will vary greatly depending on the type of processor, the amount of memory available, the number of drives being tested, the number of errors detected, and whether running in exec or user mode.

Console data switches 9 and 31 can be used to increase the execution time of each test.

The following is a list of approximate execution times of each test run by the ALL routine in user mode with one drive being tested and 32K of core assigned. Switches 9 and 31 were 0.

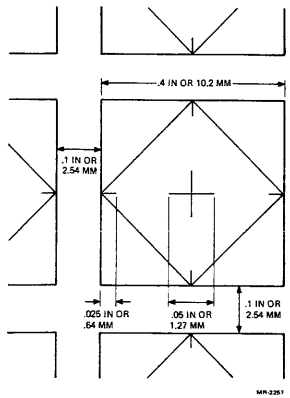
Data Test	32 minutes
Memory Test	8 seconds
EOF Test	8 minutes, 30 seconds
Random Test	1 minute, 10 seconds
Total time for all (including printing totals)	42 minutes, 10 seconds

GENERAL INFORMATION

Code	DDXYA.A10
Title	PDP-10 CALCOMP Plotter Diagnostic
Abstract	This program will test the PDP-10 BA10 plotter interface and a CALCOMP plotter. The program will run in user mode or exec mode. Switch register control has been provided for selecting various increment sizes, plotting speeds and CALCOMP models.
Hardware Required	KA10, KI10 or KL10 mainframe/32K of core (minimum)/plotter and controller.
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	If a proper figure (box) is to be drawn, the control switch settings must be correct (refer to Tables 1, 2 and 3).
Notes	1. Interface Margin Testing: set switch 27. The program will not perform the drawing test, but will test all logic except the motion flip-flops. The plotter motion flip-flops are best tested by drawing the small boxes and observing the output. 2. The plotter will position itself 1/2 inch from the right side and move the drum up approximately 3 inches before starting the test.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	Refer to Tables 2 and 3.

OPERATIONAL CONTROL
Other than the switches the user has no control over the operation of this diagnostic.

TEST SUMMARY
DDXYA draws a series of square boxes. The size of the boxes is dependent upon the control switch settings.



Typical DDXYA Box Plot

ERROR SUMMARY
Errors in the static test print the value of the PC, results of a CONI and an error number.

Errors in the drawing test are detected by visual observation of a failure to reproduce the figures shown below.

Table 1 DDXYA Control Switch Summary

Switch	Mnemonic	State	Description
0-17			Standard (Refer to the 10/10 STD module.)
18	INHSTA	0 1	Normal operation Inhibit static interface test
19-20	SELMDL		Select plotter model type 19 20 0 0 Model 502 CALCOMP 0 1 518 1 0 563 1 1 565
21-23	SELSTP		Select step size of plot. Switches Step Size 21 22 23 Inches Millimeters 0 0 0 .010 0 0 1 .005 .1 0 1 0 .0025 .05 0 1 1 .002 1 0 0 .00125 .025 1 0 1 .001 1 1 0 .010 spare-not used 1 1 1 .010 spare-not used
24-26	STPSEC		Select number of steps per second Switches 24 25 26 Steps/Second 0 0 0 200 0 0 1 300 0 1 0 350 0 1 1 400 1 0 0 450 1 0 1 700 1 1 0 900 1 1 1 Not used
27	INHLTL	0 1	Normal operation Inhibit little box
28	RETRAC	0 1	Normal operation Retrace last set of boxes NOTE Refer to Table 2 for further explanation of this switch.
29-35	RELSI%		Establish the relative size of the boxes to be drawn. NOTE Refer to Table 2 for further explanation of these switches.

Table 2 Switch 28 and 29-35 Usage

28	29-35	Description
0	= 1 (0000001)	Plot 4 rows of the smallest size boxes then double the dimensions of the boxes and plot 4 more rows of boxes. Continue this way until the largest size possible is drawn, then exit if in user mode or recycle if in exec mode. In order to save time and paper, the number of rows drawn is reduced to 2 if the multiplier is 16 or greater and reduced to 1 if 32 or greater.
0	NOT = 1	Use the number in SW 29-35 as a multiplier for the size boxes to be drawn, relative to the smallest size. If 0, the smallest size box is drawn. The smallest size boxes are drawn with 1/2 inch between centers. To draw boxes with 5 inches between centers the multiplier would be a decimal 10 or octal 12 in the switches. Once the size is determined, the program will draw 4 rows of boxes, then exit if in user mode or recycle if in exec mode.
1	= 1 (0000001)	Draw 4 rows of the smallest size box, then retrace the 4 rows. Next draw 4 rows of the next size box and retrace those 4 rows. The program continues this way until 4 rows of the largest size box have been drawn and retraced. The program will then exit if in user mode or recycle if in exec mode.
1	NOT = 1	Draw 4 rows of the specified size boxes, then continuously retrace those 4 rows.

Table 3 Plotter Specifications

Calcomp Model	Step Size	Steps/Second	Time Per Step
502	All	300	3.3 ms
518	.005 in .002 in .1 mm .05 mm	200 450 200 400	5 ms 2.2 ms 5 ms 2.5 ms
563	.010 in .005 in .1 mm	200 300 300	5 ms 3.3 ms 3.3 ms
565	All	300	3.3 ms
602	All	450/900	2.2 ms/1.1 ms
618	.005/.0025 in .00/.001 in .1/.05 mm .05/.025 mm	200/400 450/900 200/400 450/900	5 ms/2.5 ms 2.2 ms/1.1 ms 5 ms/2.5 ms 2.2 ms/1.1 ms
663	.010/.005 in .005/.0025 in .0025/.00125 in	350/700 450/900 450/900	2.9 ms/1.4 ms 2.2 ms/1.1 ms 2.2 ms/1.1 ms
665	All	450/900	2.2 ms/1.1 ms

GENERAL INFORMATION

Code	DEMMG.A10
Title	KI10 or KL10 4096K Memory Diagnostic
Abstract	This memory diagnostic tests all of memory unoccupied by the program to make sure that the memory modules and associated hardware function as a complete operating system. The program consists of a series of five tests. The first test verifies the ability of the memory to operate with various bit combination data patterns. The second test verifies the ability of the memory to uniquely address any and every memory location. The third test verifies the ability of memory to detect a 1 or 0 under maximum half-select core array noise conditions. The fourth test verifies the ability of memory to write and read a single 1 or single 0 bit in a memory array of all 1s or all 0s words. The fifth test verifies the ability of memory to continuously access and read the same location without core characteristic changes due to core array heating. Any one, or a combination, of the five tests is selectable from terminal input or from the console data switches. Also, all of memory or any selected module or portion of memory may be tested. The memory diagnostic operates in executive mode or in time-sharing user mode. In executive mode it provides a complete memory checkout diagnostic. In user mode it will provide an indication of the state of the memory system.
Hardware Required	KI10 or KL10 mainframe/32K of core (minimum)
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	NON-X-MEM and PARITY STOP must be reset
Notes	User Mode To lock the program in core, set the LOCKPG switch. This will lock the program at an extremity of core (if lock privileges are allowed). This can be used to test the last module of memory in user mode. In case of a bad module, set up the system so the bad module has the highest addresses and lock the program there for testing. Memory Indicators (MI) Displays During normal operation, the MI will display as follows. STAND-ALONE: LH = pass count/RH = test number DIAG MONITOR: decrementing program iteration count When an error occurs, the following will be displayed in the MI. DATA error - MI = failed bits (only if data error printout inhibited, switch NOERPT set) UUO error - MI = 707070, UUO address NON-X-MEM/INTERRUPT error - MI = 777000, interrupt address

Power Fail Test
A power fail interrupt routine exists in the program to be able to verify that there is no stored data loss during power turn-on or failure. This routine also verifies that the power failure interrupt feature does indeed operate.

The following procedure is used.

1. With the program operating, turn OFF power.
2. Turn power back ON.
3. Set address switches to 4005(8).
4. After waiting for power TURN-ON time delay, press START.
5. Program should continue without error.

If an interrupt did not occur on power fail, the program will print the following and halt.

POWER INTERRUPT FAILED

NOTE - KI10s
If an actual power failure occurs, when power is restored the program will auto-restart.

KI10 Inhibit Paging
To inhibit paging operation; i.e., limit maximum tested memory to 112K-1, set the INHPAG switch. Otherwise, the page map is set up to allow full 256K memory testing to occur (if there is that much memory). The INHPAG switch also limits memory mapping to 112K-1.

Recommended Method of Testing

1. Make several passes of the program with data switches 12, 13, and 14 set. Inhibit BLT exercising. Inhibit fast rate addressing and inhibit read/restore, to find any solid memory failures.
2. Make several passes of the program with data switch 9 set (12, 13, and 14 reset), fast cycle, to find any marginal memory failures.
3. Finally, make several passes of the program in the normal mode, no data switches set, to verify memory system reliability.

To thoroughly test the first 16K memory module of a system, use the memory module switches to exchange modules 0 and 1, then reload the program and repeat the above sequence testing module 1.

Complete Address Testing
To test memory addressing greater than that which is available in consecutive memory, a memory module may be selected to whatever address is desired and then tested by using switch controlled testing or by using SEGMENT testing. Observe the memory map printout and use only memory that actually exists.

Inhibit Data Complement, Fast Rate Addressing
The fast-rate addressing exercise routine may be run in two different modes. The first, switch 11 (INHCMP) reset, is with read/modify/write memory accesses, (SETCMM). The second, switch 11 (INHCMP) set, is with read/restore memory accesses (CAM). These two different methods may be helpful in diagnosing memory problems and both should be tried to isolate marginal memory failures. A failure when using the (SETCMM) could be either a pick or drop.

Segment Testing
When a memory problem is isolated to a small section of memory, segment testing may be used to more exhaustively test this area. Also, to thoroughly test addressing between two modules or sections of memory, as in the case of between upper and lower 128K, segment testing may be set up so that the starting address is less than the crossover point and the ending address is greater than the crossover point.

Inhibit on Mask
When a solid bit failure is detected on a particular bit and it is desired to continue testing to find any less solid failures, a mask word may be used which will prevent that particular bit from being processed as an error. This is done by storing in location MASK, 4037, a word with 1s in the desired bit deletion positions. Setting switch 10, INHMSK, then provides this feature.

Loading and Starting Procedure: Standard (Refer to the 10/10 STD module.)

Control Switches: Refer to Table 1.

Table 1 DEMMG Control Switch Summary

Switch	Mnemonic	Description
0	CYCL50	N/A
1	RESTRT	Print operating totals, restart test.
2	TOTALS	Print totals after each test.
3	NOPNT	Inhibit all print/typeout (except forced).
4	PNTLPT	Print all data on line printer. Logical device in user mode.
5	DING	Ring bell on error.
6	LOOPER	Enter exercise/check loop on error. Remain in loop until switch reset.
7	ERSTOP	Stop on program error condition. Otherwise, program reinitializes and restarts from first test. However, operation may be invalid.
8	PNTTLS	Print totals at completion of all selected tests (nonstandard).
9	FSTCYC	Fast cycle - Perform memory exercising routines fewer than the normal reliability mode iterations (1 instead of 20). Speeds up testing by a factor of 20 (approximately).
10	INHMSK	Inhibit error checking on mask. Do not report errors in bits masked by 1s in location MASK (4037).
11	INHCMP	Inhibit data complement (R/C/W) when executing fast-rate addressing. Do read/restore instead.
12	INHBLT	Inhibit block transfer cycles.
13	INHFR	Inhibit fast-rate addressing.
14	INHRR	Inhibit read/restore cycles. WCP - R,C,W cycles

Table 1 DEMMG Control Switch Summary (Cont)

Switch	Mnemonic	Description
15	INHPAG	Exec - Inhibit paging KI10 - Limit memory at 112K-1
	LOCKPG	User - Lock program in core
16	NOPARP	Inhibit parity error typeout.
17	NOERPT	Inhibit data error typeout.
18	SWCON	Use switches for test control - testing parameters input from RH switches rather than from terminal.
19	EXTDIN	Extended input format - used to specify and run on selected MA bit in fast-rate addressing, plus if in switches mode allows typein selection of specific data patterns for the data patterns tests and for the WCP test.
20	WCPITL	WCP interleave - sets up WCP for interleaved modules.
21	WCPSW3	Module type selection (WCP test)
22	WCPSW2	0 = Run all module types
23	WCPSW1	1 = MA10 2 = MB10 64 X 64 3 = MB10 128 X 128 4 = MD10 5 = ME10 6 = Alternate 1s and 0s 7 = 1s and 0s checkerboard
24	TSTSW3	Test selection.
25	TSTSW2	0 = Data patterns, address, WCP, float 1/0
26	TSTSW1	1 = Data patterns 2 = Address 3 = WCP 4 = Float 1s/0s 5 = Heating 6 = Address and WCP 7 = All tests
27	MODUL	Test single module - 16K Must be set to enable selection of the functions controlled by switches 28 through 35.
28	HALF	Test half module - 8K
29	LOWER	Test lower half module
30	QTR	Test quarter module of selected half module - 4K
31	LWRQTR	Test lower quarter
32	MSW4	Module selection switches, 0 to 17(8)
33	MSW3	
34	MSW2	
35	MSW1	

OPERATIONAL CONTROL

User Mode
In user mode the program title will be printed and the following questions asked to select the operational switches.

TELETYPE SWITCH CONTROL? 0, S, Y OR N (CR) -

If 0, zeros are used for the switch settings.

If S, previously set switches are used. This is only valid on a program restart.

If YES, the following appears.

SPECIFY LH SWITCHES IN OCTAL -

SPECIFY RH SWITCHES IN OCTAL -

If NO, actual console switches are used.

System Exerciser/User Mode
In system exerciser mode, the starting address is 4003. The data switches used are prestored in SWTEXR location 4024. Data switches used are: FSTCYL, WCPSW3!WCPSW2!WCPSW1, (WCP 1s and 0s checkerboard) and SWCON. The program makes ten passes and then exits back to the system exerciser.

Exec Mode
Select operation console data switch settings. Set data switch 18 if testing parameters are to be selected by the switches. Reset switch 18 if testing parameters are to be input from the console terminal. The contents of the data switches, switches 1 through 17, are sensed at the outset of the program and then again before each program subroutine. The RH switches, 18 through 35, are only used at the beginning of the test to set up the testing parameters (if switch 18 is set).

Switch Control (18 = 1)
If testing parameters are from the switches and switch 18 is set, the test inputs the switches and commences operation.

Console Terminal Control (18 = 0)
If the testing parameters are to be from the console terminal, switch 18 reset, input the desired parameters as called for by the terminal.

Terminal input typing specifications are as follows.

1. If the question asked is Y or N, type only Y or N followed by a carriage return as a response.
2. If the question asked is for a digit, Y, or N, type Y, N, or the required digit followed by a carriage return.
3. If the question asked requires a number as a response, up to 6 octals may be typed, or up to 5 decimals, followed by a CR to delimit the number.
4. Octal typeins are:
   a. addresses
   b. module types
5. Decimal typeins are:
   a. memory size
   b. number of modules
   c. module size
   d. number of segments

The following is a sample typein and the associated computer printout (operator responses are in boldface and follow the -).

```
PDP-10 MEMORY DIAGNOSTIC (DEMMG)

MEMORY MAP =
FROM     TO        SIZE/K
000000   077777    32

SEGMENT TESTING? Y OR N (CR) - N

SPECIFY MEMORY SIZE TO BE TESTED IN K, 0 = ALL - 32

SPECIFY NUMBER OF MODULES, 0 = ALL SAME OR 1 TO 16 - 2

SPECIFY SIZE IN K, MODULE TYPE (0 TO 17) OF EACH
MODULE SIZE 1 - 16
MODULE TYPE 1 - 1
MODULE SIZE 2 - 16
MODULE TYPE 2 - 1

TEST SELECTION
DATA PATTERNS - 1 TO 7, Y OR N (CR) - Y
ADDRESS, Y OR N (CR) - Y
WCP - 1 TO 4, Y OR N (CR) - Y
FLOATING ONES/ZEROS, Y OR N (CR) - Y
CORE HEATING, Y OR N (CR) - Y

SELECTION COMPLETED, IS IT OK, Y OR N (CR) - Y
```

In the example the test header was printed, the memory was mapped and the operator was asked if segment testing was desired. Operator answered no; he was then asked for memory size to be tested. Operator answered 32(K) which in this example is all available memory. He was then asked for number of modules, 2, then size and module type of each, 16(K) and module type 1(MA10), and tests to be run.

At the completion the operator verifies that the selection was ok and then testing commences.

TEST DESCRIPTION
This section describes operator dialogue test selections and data patterns.

MEMORY MAPPING - In the first pass of the program, a map of memory is printed as it is seen by the CPU. The purpose of this memory mapping is to determine maximum memory size for the test parameters and also to point out any missing memory addresses or noncontiguous memory modules.

The memory map is printed in the following format.

```
MEMORY MAP =
FROM     TO        SIZE/K
NNNNNN   NNNNNN    NN
XXXXXX   XXXXXX    XX
```

Where:

Ns = lowest contiguous segment of core memory. This should always be the only entry and equal to all memory.

Xs = some higher segment of memory. This will occur for example in a 32K system where the higher module has an address greater than 1.

If there is more than one segment of memory the following is also printed.

```
MAPPING ERROR     ;NONCONSECUTIVE MEMORY MODULES?
```

If there is any nonexistent memory below 8K the following is printed.

```
MAPPING ERROR     ;NON-X-MEM BELOW 8K?
```

If there is an interleaving problem the following is printed.

```
MEMORY MAP =

FROM     TO        SIZE/K    ADR SEQ
000000   377777    128       XXX7
000000   577777    192       XXX6
000000   577777    192       XXX5
000000   577777    192       XXX4

MAPPING ERROR; MEMORY INTERLEAVING?
```

This indicates that the memory interleaving is faulty.
The first map is done with addresses ending with 7.
The second map is done with addresses ending with 6.
The third map is done with addresses ending with 5.
The fourth map is done with addresses ending with 4.

Memory mapping for nonexistent memory is done with address multiples of 1000(8). Memory mapping of existent memory is done with address multiples of 100(8).

KI10 - KL10 Only

Memory mapping on the KI10 - KL10 may be limited to unpaged memory by setting the INHPAG switch, i.e., 112K-1.

If the switch is not set the EPMP/UPMP is set up so that relocation equals actual for addresses 112K through 256K.

SEGMENT TESTING - Segment testing allows the operator to define up to seven areas of core (i.e., banks) for individual testing.

If the operator had answered yes for segment testing the following would have been printed from that point.

```
SEGMENT TESTING? Y OR N (CR) - Y

NUMBER OF SEGMENTS?     1 TO 7-3
STARTING ADDRESS        1-10000
ENDING ADDRESS          1-10777
MODULE TYPE             1-1
STARTING ADDRESS        2-40000
ENDING ADDRESS          2-40777
MODULE TYPE             2-1
STARTING ADDRESS        3-70000
ENDING ADDRESS          3-70777
MODULE TYPE             3-1

TEST SELECTION
DATA PATTERNS - 1 TO 7, Y OR N (CR) - Y
ADDRESS, Y OR N (CR) - Y
WCP - 1 TO 4, Y OR N (CR) - Y
FLOATING ONES/ZEROS, Y OR N (CR) - Y
CORE HEATING, Y OR N (CR) - Y

SELECTION COMPLETED, IS IT OK, Y OR N (CR) - Y
```

In this example the operator selected three segments for testing. Each segment was defined by a starting and ending address. The dialogue following segment testing remains the same.

MEMORY SIZE - Memory size can be all, or any part (in 1K increments) of the total core memory available to the system.

NUMBER OF MODULES - This question is asking for the number of different types of memory modules to be tested. For a system with four MA10S, one MD10, and five ME10s the answer would be 3. This question and the question following it (module type) are a pair.

MODULE TYPE - The module type, or WCP checkerboard, to be used for a specific memory module may be specified in one of two ways:

1. As terminal input - The terminal requests operator input of the module type. There are 0 to 17 different types, as follows.

Noninterleaved	Interleaved
0-Run all module types	10-run all interleaved module types
1-MA10, 16K	
2-MB10, 16K 64 X 64	11-MA10s, 32K
3-MB10, 16K 128 X 128	12-MB10s, 32K 64 X 64
4-MD10, 32K	13-MB10, 32K 128 X 128
5-ME10, 16K	14-MD10s, 64K or 128K
6-Alternate 1s and 0s	15-ME10s, 32K
7-1s and 0s checkerboard	16-Alternate 1s and 0s
	17-1s and 0s checkerboard

When the memory modules are interleaved, the module size is specified as the overall size of the interleaved modules (i.e., two 16K interleaved modules would be specified as one 32K module).

2. As console switches - When the input parameters are provided from the switches, switches 20 through 23 are used. They are set up in octal as described above.

DATA PATTERNS - The word patterns to be used for the data pattern test and for the worst-case patterns test may be specified by type-in when the program asks for the test selection. Instead of answering Y or N the operator may typein one of the following.

DATA PATTERNS;

Y = Run test, all patterns
N = Do not run test
1 = Run test, all 1s pattern
2 = Run test, all 0s pattern
3 = Run test, alternate bit pattern. Alternate bits, 525252 525252 on the first pass, then 252525 252525 for the second pass. (The same data is used on each pass throughout the memory area tested.)
4 = Run test, floating 1s pattern. Floating 1s, 042104210421 for the first location of the first pass. This data pattern rotates word to word throughout the memory area tested. On the second pass the initial location word is rotated one and then this data word rotates word to word throughout the memory area. This continues on subsequent passes until all data bit combinations are tested (i.e., four passes).
5 = Run test, floating 0s pattern. Floating 0s, 735673567356 for the first location of the first pass. This data then rotates word to word throughout the memory area tested. On the next pass the initial word is rotated one and the operation continues on subsequent passes until all combinations are tested (i.e., four passes).
6 = Run test, both floating 1s and floating 0s patterns.
7 = Run test, pseudorandom, parity bit check pattern. Pseudorandom parity bit check, 123456701234 for the first location of the first pass. This data then rotates word to word and is 2's complemented throughout the memory area tested. On the next pass the initial word is rotated three and the operation continues on subsequent passes until all combinations are tested (i.e., twelve passes). This pattern causes the parity bit to change word to word and should be the most comprehensive test.

MEMORY ADDRESSING TEST - The memory addressing test verifies the ability of the memory system to uniquely address any and every memory location selected for test. This is done by writing the value of each memory location into the right half of itself and the complement of this into the left half. The memory is then exercised and tested. At the completion of this, the address pattern word is reversed and the value of each location written into the left half word and the complement of this written into the right half word. The memory is then exercised and tested for correct contents.

The address pattern words used are:

1. C,ADR - LH, complement address /RH, address
2. ADR,C - LH, address /RH, complement address.

The testing sequence used is the same as the data patterns test except that the address pattern words are used.

ADDRESS TESTING SCHEME - Memory exercising is accomplished using two different addressing schemes. These are sequential and fast-rate. Sequential is the normal method of addressing where the address is incremented by 1. Fast-rate was developed to provide a method of exercising each memory address bit at the fastest possible rate, whereas in sequential only MA35 is exercised at the fastest rate. The purpose of this type of memory addressing is to find and diagnose problems that are MA-bit sensitive.

Fast-rate addressing is accomplished as follows: a fast-rate bit is set up which is then added to the basic memory testing address to provide the actual testing address. This fast-rate bit starts at bit 35, and is used then to access all testing addresses, then the fast-rate bit is changed to bit 34, and used again to access all testing addresses. This continues until the fast-rate bit is bit 18 (MSB of address). Each time the actual testing address becomes greater than the upper memory testing limit, if all addresses have not been tested, the initial lower testing address is incremented by 1 and then this lower address is incremented by the fast-rate bit for the next memory exercising pass.

An example of the way this operates would be as follows. On the first pass, the fast-rate bit is bit 35; therefore the initial lower testing address would be incremented by 1 until all addresses are tested (this is the same sequential). On the second pass, the fast-rate bit is bit 34; therefore the initial lower testing address would be incremented by 2 until it is greater than the upper limit. Then the lower testing address is incremented by 1 and then the fast-rate bit is added to this address and the resulting addresses used until the remaining addresses in the testing area are exercised. On the next pass the fast-rate increment would be by 4, the next 8, and so on until the fast-rate bit is bit 18.

Each MA bit is exercised so that with each memory access the particular bit being exercised is changed from a 0 to a 1 or from a 1 to a 0.

NOTE

To check a particular memory addressing bit during the fast-rate addressing exercising routine, switch 19 (EXTDIN) is set for extended input during the initial type-in and the selected bit is typed in. Then if switch 19 (EXTDIN) is set during the test operation the specified MA bit will be the only MA bit fast-rate exercised.

The following are the printed out request and typed in responses.

SPECIFY FAST-RATE ADDRESSING MA BIT (0 OR 18 TO 35) -

0 = TEST ALL MA BITS
18 TO 35 = TEST SPECIFIED BIT

WORST-CASE PATTERNS - The worst-case patterns test verifies the ability of the memory system to detect a 1 or 0 under maximum half-select core array noise conditions. This is done by filling the memory with a discrete core array checkerboard pattern for each different type of core memory (i.e., MA10, MB10, MD10, ME10, etc.). This checkerboard pattern consists of a pattern word (all 1s or alternate bits) and its complement. The specific arrangement of this checkerboard pattern in core depends on the actual physical core stack layout and its associated crossover points. The memory is filled with the checkerboard pattern and then exercised and tested. Following this the complement checkerboard pattern is written into core and the memory exercised and tested for correct contents.
The worst case patterns data words used are as follows.

Y = Run test, all patterns
N = Do not run test
1 = Run test, all 1s words and complement
2 = Run test, all 0s words and complement - provides the complemented checkerboard pattern.
3 = Run test, alternate 1s and 0s, first pass provides checkerboard pattern in even data bit core planes and complement checkerboard pattern in odd data bit core planes. Second pass provides the opposite odd and even data plane arrangement.
4 = Run test, both all 1s and all 0s patterns, WCP and WCP

The testing sequence used is the same as the data patterns test except that the WCP data words and checkerboards are used.

To fully test a memory's worst-case pattern operation, the worst-case pattern must be exercised in the complete core stack. The most exhaustive test will be done only in noninterleaved memory, as the core array duty cycle will be the greatest then.

FLOATING ONES/ZEROS TEST - The floating 1s/0s test verifies the ability of the memory system to write and read a single 1 or a single 0 bit in a memory array of all 1s or all 0s words. This is done by filling the memory array with all 1s and then writing a single 1 in the rightmost position of the first word. This word is then read and compared to what was written. The single bit is then rotated left one position and written, read, and compared. This is repeated until all bit positions in the word have been checked. The tested location is then advanced to the next word and the sequence repeated. This continues until all words in the array have been checked. The entire memory array is then tested for correctness.

This process is then repeated using a memory array of all 0 words and a floating 0 testing word.

CORE HEATING - The core heating test verifies the ability of the memory system to continuously access and read the same location without losing data due to core characteristic changes resulting from core array heating. This is done by filling the memory with all 1s and then continuously reading the test location for the period of a clock cycle, after which testing advances to the next location. After all selected locations have been exercised and tested in this manner, the entire selected memory area is again tested for correct contents of all 1s.

The data words used are: 1s, all 1s.

The testing sequence is as follows:

1. Fill memory with all 1s
2. Exercise each location for clock cycle
3. Test all selected memory for correct contents
4. Advance to next test.

ERROR MESSAGE SUMMARY

Memory Data Error
When a memory data error occurs, the following header and sample error information is printed.

```
MEMORY DATA ERROR
TN AS  PAT   ADDRESS    CORRECT        ERROR     FAILED BITS   PAR
1  SEQ ALTB  070777  252525 252525 252565 252525 000040 00000
2  F34 C,ADR 013447  764330 013447 764332 013445 000002 000002
3A F31 ZEROS 070777  777777 777777 000000 000000 777777 777777 P
4  SEQ ONES  070777  777777 777777 777776 777777 000001 000000
```

Where:

TN = The current test which detected an error.

1 = Data patterns
2 = Address
3 = Worst-case patterns
4 = Floating 1s/0s
5 = Core heating

AS = The current addressing scheme being used to exercise memory (SEQ=sequential, BLT=block transfer, Fnn=fast-rate on bit NN)

PAT = The current data pattern mnemonic

ADDRESS = The memory location which contains the data in error.

CORRECT = What the data should have been in that location.

ERROR = The data as read from that location.

FAILED BITS = A binary 1 indicates that the particular data bit failed. Could be a bit pickup or dropout (printed in octal).

PAR = A "P" indicates that when the failed location was reread for printout it still contained a parity error.

In the example above:

During test 1 address 070777 was found to be in error. From the example, it can be seen that bit 12 was picked up while using the alternate bit data pattern and sequential addressing.

During test 2 address 013447 was found to be in error. From the example, it can be seen that what was actually read was the data that corresponds with address 013445 and from this it can be deduced that, while exercising memory with memory address bit 34 changing with every memory access, address 013445 was incorrectly read and then written into the correct location; i.e., bit 34 was slow to set up.

During test 3 address 070777 was found to be in error. From the example, it can be seen that all data bits were dropped or that the location failed to read at all. Also, a parity error was found at the failed location.

For a WCP test failure (test 3), both data and parity, the test number is printed with a letter indicating which module these are as follows.

Noninterleaved	Interleaved
3A-MA10	3I-MA10
3B-MB10 64 X 64	3J-MB10 64 X 64
3C-MB10 128 X 128	3K-MB10 128 X 128
3D-MD10	3L-MD10
3E-ME10	3M-ME10
3F-Alternate 1s and 0s	3N-Alternate 1s and 0s
3G-1/0 checkerboard	3O-1/0 checkerboard

During test 4 address 07077 was again found to be in error. This time bit 17 was dropped.

In user mode the program relocation value and actual physical address will be printed for each data error.

After each error printout the program continues with the next memory location to test.

Memory Parity Error
When a memory parity occurs the following header and sample error information is printed.

```
MEMORY PARITY ERROR
TN  PROG    PC      AS   PAT        ADRCON  DATA           PARITY
3   001057  000006  F32  ALTB       027637  525212 525252  P
```

Where:

TN = The current test being run when the parity error occurred.

PROG = The current subroutine entry point in the test program. (The last PUSHJ entry)

PC = The current program counter when the parity error occurred.

AS = The current addressing scheme being used to exercise memory.

PAT = The current data pattern.

ADRCON = The address control word used to access memory. Normally points to the current location being tested.

DATA = The contents of the location pointed to by ADRCON. Normally will be the erroneous data which caused the parity error. If ADRCON points outside memory this will be blank.

PARITY = A "P" indicates that when the location pointed to by ADRCON was reread for printout it still contained a parity error.

In the above example, during test 3 a memory parity error occurred. It can be seen that while exercising the memory using the alternate bit data pattern and fast-rate addressing, bit 32 being the current fast-rate bit, bit 12 was dropped, causing the parity error. Also, when the current testing location pointed to by ADRCON was reread for printout it still contained a parity error. By referring to the program listing it can be seen that the current subroutine was running in the ACs, and which subroutine it was can be determined by the program printout.

If a memory parity error occurs when the PC is not in the ACs the following is also printed.

```
**********
PARITY ERROR IN PROGRAM
PROGRAM OPERATION QUESTIONABLE FROM THIS POINT
**********
```

In this case ADRCON and data are probably not valid error indications.

After each memory parity error and printout the program continues the testing sequence.

Nonexistent Memory and Channel 1 Interrupt Errors
When either of these errors occurs the following associated header and sample error information is printed.

```
NONEXISTENT MEMORY INTERRUPT
APR             PI              FLAGS    PC       PROG
011023 013001   000000 000000   300000   000004   000755
```

or:

```
ERROR INTERRUPT
APR             PI              FLAGS    PC       PROG
011023 013001   000000 000000   300000   000004   000755
```

Where:

APR = CONI from APR

PI = CONI from PI

FLAGS = Current program flags when interrupt occurred

PC = The current program counter when the interrupt occurred

PROG = The last subroutine entry point in the test program (the last PUSHJ entry).

Illegal UUO
When an illegal UUO is executed the following header and sample error information is printed.

```
ILLEGAL UUO EXECUTED
UUO             FLAGS     PC        PROG
064240 000000   300000    003677    000433
```

Where:

UUO = The illegal UUO that was attempted to be executed

FLAGS = The program flags

PC = The program counter at which the UUO occurred

PROG = The last subroutine entry point in the test program (the last PUSHJ entry).

Program Checksum Error
At the completion of a testing sequence the program is checksummed and this checksum is compared to the checksum obtained before the program is executed. If the checksums agree, the program continues normal operation. If the checksums do not agree the following is printed.

```
**********
ERROR IN PROGRAM, CHECKSUMS DO NOT AGREE
PROGRAM OPERATION QUESTIONABLE FROM THIS POINT
**********
```

The program then continues operation or restarts depending on switch 7, ERSTOP.

The program checksum is obtained by adding together all program code from BEGIN1 to ENDSLD of the program. All changeable words are at the end of the program after ENDSLD (all variable data). The purpose of this is to verify that the program is indeed valid, since, when testing a memory module in which the program also resides, the program could get changed due to interaction between the testing area of core and the program area of core. This could cause erroneous error reporting.

Error Totals
At the completion of the testing sequence or after each individual test, as selected by the switches, the error totals are printed. This printout facilitates data bit error determination and also address failure correlation. The printout provides the error totals per test, the number of parity errors, the data bits with pickup and dropout errors, and the address bits with associated data bit pickup and dropout errors.

A sample printout follows.

```
TEST COMPLETION, PASS COUNT 1
ERROR TOTALS:
PATTERNS      ADDRESS       WCP           FLOAT         HEATING
0             0             7             0             0
PARITY ERRORS: 7

DATA BIT FAILURES
BIT           PICKUP        DROPOUT
3             5             0
12            2             0

ADDRESS BITS WITH DATA FAILURES
BIT           PICKUP        DROPOUT
18
19
20
21
22
23            7             0
24            5             0
25            5             0
26
27
28
20
30
31
32            1             0
33            7             0
34            4             0
35            7             0
```

The sample printout indicates that at the completion of one pass the worst-case patterns test found seven errors and there were also seven parity errors. Data bit 3 was picked up as an error five times and data bit 12 was picked up as an error twice. No other data bits had any errors. The ADDRESS BITS WITH DATA FAILURES printout indicates that the address with the most errors was probably 16005 or an address combination that gives this indication.

If there are no errors the total printout prints as per the following example (if selected by the switches).

```
TEST COMPLETION, PASS COUNT 2
ERROR TOTALS: NONE
```

After the totals print out the error counters are cleared and the program checksummed. Test operation then continues with the first/next test.

GENERAL INFORMATION

Code	DFDTE.A10
Title	DECSYSTEM-10/20 KL10 DTE20 Interface Diagnostic
Abstract	This PDP-10 KL10 diagnostic tests the DTE20 functional operations of the KL10 central processor.
Hardware Required	KL10 mainframe/32K of core (minimum)
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restriction	The DTE20 must be set to the privileged mode
Notes	1. This program runs in exec mode only. 2. The program cycle time is several minutes. 3. The iteration count of the program is printed on the console terminal. 4. The program is usable with the diagnostic monitor to provide reliability tests, acceptance tests, and/or to provide a quick method of isolation of a fault to a particular area of the processor. To thoroughly test all hardware, all test control data switches should be set to 0. 5. When debugging hardware, set switches to 0. Allow the terminal to print the error messages. This allows the program to run a complete pass and then the error messages may be correlated to quickly diagnose the failure. If a hardware problem is such that the error messages, after the first one, have no meaning (first error causes all following tests to fail), set the loop on error switch and restart the test from the beginning. The first failure will then cause the program to enter a loop suitable for scoping. The error messages, used in conjunction with the listing and scoping if necessary, should allow the failing component to be isolated and replaced and/or repaired. 6. Error information may be obtained quickly by printing errors on the line printer.
Loading and Starting Procedures	Standard (Refer to the 10/10 STD module.)
Control Switches	Standard (Refer to the 10/10 STD module.) The following switches are not implemented: 11, 12, 13, 14, 15, 16 and 17. There are no right half switches.

OPERATIONAL CONTROL
Other than the switches, the user has no control over the operation of this diagnostic

DFDTE TEST SUMMARY
The individual tests performed by this diagnostic are summarized in Table 1.

ERROR MESSAGE SUMMARY
Errors are printed on the terminal or line printer. The error printout contains the test title, test pass count, PC of the failure, subtest title, subtest function, correct word, actual results, discrepancy word, and a diagnostic comment.

The following is a sample error message.

```
DECSYSTEM10 KL10 DTE20 FUNCTIONAL DIAGNOSTIC (DFDTE)
TEST PASS COUNT = #
ERROR IN (SUBTEST TITLE) - (SUBTEST FUNCTION)
CORRECT: XXXXXX XXXXXX
ACTUAL:  YYYYYY YYYYYY
DIFFER.  ZZZZZZ ZZZZZZ
(DIAGNOSTIC COMMENT)
```

The subtest title, subtest function, and diagnostic comment portion of the error printout may be inhibited by setting the TXTINH switch. This allows for shorter printouts on repetitive failures.

The test title is only printed when starting the program or on the first failure if it has not been printed yet.

The test pass count is only printed on errors which occur on passes after the first pass.

If an audio indication of a continuing error is desired the DING switch may be set.

Table 1 DFDTE Test Summary

Test	Description
1	Basic I/O testing
2	Interprocessor testing (basic)
3	DTE20 reset
4	PI test
5	0s deposit
6	1s deposit
7	Floating 1 deposit
8	Floating 0 deposit
9	Byte count test
10	Byte transfer test (basic)
11	Byte transfer error termination
12	Byte transfer Z stop
13	Byte transfer word mode reliability
14	Byte transfer byte mode reliability

GENERAL INFORMATION

Code	DFKAA.A10
Title	KL10 Basic Instruction Diagnostic 1
Abstract	This PDP-10 KL10 basic instruction diagnostic is the first and most basic in a series of PDP-10 KL10 processor diagnostics. This diagnostic assumes the halt instruction and the computer console to be operative. It makes no further assumptions except that it is loaded into memory correctly. The diagnostic tests the basic functionality of the processor and microcode.
Hardware Required	KL10 mainframe/32K of core (minimum)
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	None
Notes	1. If the diagnostic fails to start correctly, try starting at the first test instead of at the beginning of the control sequence. Refer to the listing on microfiche. 2. The cycle time of the program is in the millisecond range and is therefore suitable for taking margins, vibration tests, etc. 3. The iteration count of the program is printed by the console processor. 4. DFKAA does not use the SUBRTN package.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	None

OPERATIONAL CONTROL
Once started the program will cycle continuously until it is stopped or an error occurs.

DFKAA TEST SUMMARY
Refer to the listing on microfiche.

ERROR SUMMARY
Errors are in the form of halt instructions. The listing should be consulted to determine the cause of the error. A no operation (JUMP) instruction follows each halt. This may be useful in constructing a scope loop to cycle on the failing instruction.

GENERAL INFORMATION

Code	DFKAB.A10
Title	DECSYSTEM KL10 Basic Instruction Test (Part 2)
Abstract	This PDP-10 KL10 basic instruction diagnostic is the second in a series of PDP-10 KL10 processor diagnostics. The diagnostic performs logic testing of the processor and microcode shift/rotate functions.
Hardware Required	KL10 mainframe/32K of memory (minimum)
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	None
Notes	1. The nonexistent memory and parity stop switches should be reset (0). These errors, illegal UUOs and other errors of this type are handled by printout on the terminal. 2. The cycle time of the program is in the millisecond range and is therefore suitable for taking margins, vibration tests, etc.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	Standard (Refer to the 10/10 STD module.) The following switches are not implemented: 1, 2, 9, 12, 14, 15, 16, and 17. There are no right half switches.

OPERATIONAL CONTROL
Once started the program will run continuously until it is stopped or an error occurs.

TEST SUMMARY
Refer to the listing on microfiche.

ERROR SUMMARY
Standard (Refer to the 10/10 STD module.)

GENERAL INFORMATION

Code	DFKAC.A10
Title	DECSYSTEM KL10 Basic Instruction Test (Part 3)
Abstract	This PDP-10 KL10 basic instruction diagnostic is the third in a series of PDP-10 KL10 processor diagnostics. The diagnostic performs logic testing of the processor and microcode multiply/divide functions.
Hardware Required	KL10 mainframe/32K of memory (minimum)
Preliminary Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	None
Notes	1. The nonexistent memory and parity stop switches should be reset (0). These errors, illegal UUOs and other errors of this type are handled by printout on the the terminal. 2. The cycle time of the program is in the millisecond range and is therefore suitable for taking margins, vibration tests, etc.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	Standard (Refer to the 10/10 STD module.) The following switches are not implemented: 1, 2, 9, 12, 14, 15, 16, and 17. There are no right half switches.

OPERATIONAL CONTROL
Once started the program runs continuously until it is stopped or an error occurs.

TEST SUMMARY
Refer to the listing on microfiche.

ERROR SUMMARY
Standard (Refer to the 10/10 STD module.)

GENERAL INFORMATION

Code	DFKAD.A10
Title	DECSYSTEM KL10 Basic Instruction Test (Part 4)
Abstract	This PDP-10 KL10 basic instruction diagnostic is the fourth in a series of PDP-10 KL10 processor diagnostics. The diagnostic performs logic testing of the processor and microcode BYTE/BLT/JFFO/MISC functions.
Hardware Required	KL10 mainframe/32K of memory (minimum)
Preliminary Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	None
Notes	1. The nonexistent memory and parity stop switches should be reset (0). These errors, illegal UUOs and other errors of this type are handled by printout on the terminal. 2. The cycle time of the program is in the millisecond range and is therefore suitable for taking margins, vibration tests, etc.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	Standard (Refer to the 10/10 STD module.) The following switches are not implemented: 1, 2, 9, 12, 14, 15, 16, and 17. There are no right half switches.

OPERATIONAL CONTROL

Once started the program runs continuously until it is stopped or until an error occurs.

TEST SUMMARY

Refer to the listing on microfiche.

ERROR SUMMARY

Standard (Refer to the 10/10 STD module.)

GENERAL INFORMATION

Code	DFKBA.A10
Title	DECSYSTEM KL10 Basic Instruction Reliability Test (1)
Abstract	This PDP-10 KL10 basic instruction reliability test verifies the functionality of the KL10 basic instructions. This program tests all the basic instructions with the exception of fixed point multiply/divide, floating point and byte.
Hardware Required	KL10 mainframe/32K of memory (minimum)
Preliminary Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	None
Notes	1. The nonexistent memory and parity stop switches should be reset (0). These errors, illegal UUOs and other errors of this type are handled by printout on the terminal. 2. The cycle time depends upon memory size and increases as memory size increases. 1. Normal operation - approximately one minute in 32K. 2. Reliability mode - approximately 3 to 5 minutes.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	Refer to Table 1.

OPEATIONAL CONTROL
Normal operation with all switches set to 0 is quick-verify mode. For reliability test, set the RELIAB switch. Without either the OPRSEL switch or the RELIAB switch set or the memory size selected via the switches, the program will run using only 32K. Setting just the OPRSEL switch allows the program to run in quick-verify mode using all available memory.

TEST SUMMARY

In the majority of cases each instruction is tested by simulating the instruction, with simpler instructions, and then executing the instruction. Random numbers are used as the operands in AC and/or C(E). The results of the simulation and execution are compared and an error message is printed if the results are not equal.

In MEMORY and BOTH modes, a random number is placed in the AC and C(E) and a series of 4 or 5 instructions are executed. The answer in the AC and/or C(E) is checked and an error message is printed if the number is not correct. Each set of 4 or 5 instructions acts on each memory location from the end of the program to the designated end of memory.

Also tested are all of the PC-sensitive instructions. These instructions are: JSR, JSA, JSP, JRA, PUSH, POP, PUSHJ and POPJ. In the majority of cases a JRST or JSP back to the program is placed in every location from the end of the program to the designated end of memory. The program then does a PC-sensitive instruction to the first testing location. When the program returns, a check is made to see that the PC-sensitive instruction went to the right address. Memory address is then incremented and the PC-sensitive instruction repeated. The program also includes a defer test and both indirect and indexing are tested.

ERROR SUMMARY
Errors are printed on the terminal or line printer. The error printout contains the test title, the PC of the failure, AC number, error word and correct word.

The PC value is useful in relating the failure to the listing.

When the scope loop mode is used, the MI register will count for each occurrence of an error. If an audio indication of a continuing error is desired the DING switch may be set.

The following are the different error formats with their respective UUOs and error messages.

```
ERROR No. 1 - ERR  AC,E

    EXAMPLE:                        AC              E
    2053 / CAME   AC1,AC2           ;RESULT         CORRECT
    2054 / ERR    AC,RAN1           ;ORIG  C(AC)    ORIG C(E)

    AC1=5                           ;TEST DATA
    C(AC1)  = 201532107642
    C(AC2)  = 201432107642
    C(RAN1) = 777777777777
    C(AC)   = 576345670135

    ERROR MESSAGE:
                                    (SOURCE OF NUMBERS PRINTED)
    PC    = 002054                  ;PC OF ERROR UUO
    AC    = 05                      ;AC FIELD OF UUO-1
    C(AC) = 201532107642            ;C(C(AC)) OF UUO-1
    COR   = 201432107642            ;C(C(ADDRESS FIELD)) OF UUO-1
          ORIGINAL
    C(AC) = 777777777777            ;C(C(ADDRESS FIELD)) OF UUO
    C(E)  = 576345670135            ;C(C(AC)) OF UUO

ERROR NO. 2 - ERRM  AC,E

    EXAMPLE:                        AC              E
    2053 / CAME  AC2,MUD            ;CORRECT        RESULT
    2054 / ERRM  AC,RAN1            ;ORIG  C(AC)    ORIG C(E)

    MUD=5033                        ;TEST DATA
    C(MUD)  = 201532107642
    C(AC2)  = 201432107642
    C(RAN1) = 777777777777
    C(AC)   = 576345670135

    ERROR MESSAGE:
                                    (SOURCE OF NUMBERS PRINTED)
    PC   = 002054                   ;PC OF ERROR UUO
    E    = 5033                     ;BITS 18-35 (E) OF UUO-1
    C(E) = 201532107642             ;C(C(E)) OF UUO-1
    COR  = 201432107642             ;C(C(AC)) OF UUO-1
         ORIGINAL
    C(AC)= 777777777777             ;C(C(E)) OF UUO
    C(E) = 576345670135             ;C(C(AC)) OF UUO

ERROR NO. 3 - ERRI  AC,E

    EXAMPLE:                        AC              E
    2053 / CAME  AC1,AC2            RESULT          CORRECT
    2054 / ERRI  RAN1,(AC)          ;ORIG C(AC)     ORIG E

    AC1=5                           ;TEST DATA
    C(AC1)  = 107742670135
    C(AC2)  = 107642670135
    C(RAN1) = 777777777777
    C(AC)   = 576345670135

    ERROR MESSAGE:
                                    (SOURCE OF NUMBERS PRINTED)
    PC    = 002054                  ;PC OF ERROR UUO
    AC    = 5                       ;AC FIELD OF UUO-1
    C(AC) = 107742670135            ;C(C(AC)) OF UUO-1
    COR   = 107642670135            ;C(C(E)) OF UUO-1
          ORIGINAL
    C(AC) = 777777777777            ;C(C(AC)) OF UUO
    E     = 670135                  ;C(ADDRESS FIELD) OF UUO

ERROR NO. 4 - ERROR  AC,E

    EXAMPLE:                        AC              E
    2053 / CAME   AC,RAN
    2054 / ERROR  AC,RAN            ;RESULT         CORRECT

    AC=5                            ;TEST DATA
    C(AC)  = 201532107642
    C(RAN) = 201432107642
```

```
    ERROR MESSAGE:
                                        (SOURCE OF NUMBERS PRINTED)
    PC    = 002054                      ;PC OF ERROR UUO
    AC    = 5                           ;AC FIELD OF UUO
    C(AC) = 201532107642                ;C(C(AC)) OF UUO
    COR   = 201432107642                ;C(C(E)) OF UUO

ERROR NO. 5 - ER  AC,[ASCII/MESSAGE/]

    EXAMPLE:                            AC              E
    2053 / JFCL   10,,+2
    2054 / ER     AC,[ASCII/0V/]        ;RESULT         MESSAGE

    AC=5                                ;TEST DATA
    C(AC)   = 201432107642

    ERROR MESSAGE:
                                        (SOURCE OF NUMBERS PRINTED)
    PC   = 002054                       ;PC OF ERROR UUO
    AC   = 5                            ;AC FIELD OF UUO
    C(AC) = 201432107642 OV             ;C(C(AC)) OF UUO
                                        ;ADDRESS FIELD OF UUO POINTS TO
                                        ;AN ASCII MESSAGE

ERROR NO. 6 - ERM  AC,E

    EXAMPLE:                            AC              E
    2053 / SOJ   AC2,
    2054 / ERM   AC1,(AC2)              ;C(AC)          RESULT

    C(AC2)   = 5033                     ;TEST DATA
    C(AC)    = 740000005756
    C(C(AC2) = 254000004041

    ERROR MESSAGE:
                                        (SOURCE OF NUMBERS PRINTED)
    PC    = 002054                      ;PC OF ERROR UUO
    E     = 5033                        ;BITS 18-35 (E) OF UUO
    C(AC) = 740000005756                ;C(AC) OF UUO
    C(E)  = 254000004041                ;C(C(EE)) OF UUO

ERROR NO. 7 - ERMM  AC,E

    EXAMPLE:                            AC              E
    2053 / SOJ   AC2,
    2054 / ERMM  AC1,(AC2)              ;C(AC)          RESULT

    C(AC2) = 5033                       ;TEST DATA
    C(AC1) = 740000005756

    ERROR MESSAGE:
                                        (SOURCE OF NUMBERS PRINTED)
    PC    = 002054                      ;PC OF ERROR UUO
    E     = 5033                        ;BITS 18-35 (E) OF UUO
    C(AC) = 740000005756                ;C(AC) OF UUO

ERROR NO. 11 - EERR   ,E
ERROR NO. 12 - EERRM  ,E
ERROR NO. 13 - EERRI  ,E

    ERRORS 11, 12 AND 13 ARE THE SAME AS ERRORS 1, 2 AND 3 EXCEPT
    THAT THE AC OF THE UUO IS REPLACED BY C(RAN).  IN OTHER WORDS
    C(RAN) WILL BE PRINTED FOR THE ORIGINAL C(E).
```

Table 1 DFKBA Control Switch Summary

Switch	Mnemonic	State	Function
0	ABORT	0 1	Normal operation Abort at end of pass.
1	RSTART		Not used
2	TOTALS		Not used
3	NOPNT	0 1	Normal operation Inhibit printout (except forced)
4	PNTLPT	0 1	Normal operation Print errors on line printer
5	DING	0 1	Normal operation Ring bell on error
6	LOOPER	0 1	Normal operation Loop on test error
7	ERSTOP	0 1	Normal operation Halt on test error
8	PALERS	0 1	Print only first error when looping Print all errors
9	RELIAB	0 1	Normal - fast cycle operation Reliability mode - use all of memory
10	TXTINH	0 1	Print full error message Inhibit comment portion of error message
11	INHPAG	0 1	Allow paging and trap enable Inhibit paging and trapping
12	MODDVC		Not used
13	INHCSH	0 1	KL10 - Allow cache use KL10 - Inhibit cache use
14	OPRSEL	0	Run in 32K unless in reliability mode (SW 9 = 1)
		1	Use all available memory
15-26			Not used
27-35			Select size of memory to be used. Switches 27-35 correspond to the high-order nine address bits. The low-order nine bits are appended as 1s (e.g., 27-35 = 077 = 077777 or 32K). Switches 27-35 = 000 specify all available memory is to be used.

GENERAL INFORMATION

Code	DFKBB.A10
Title	KL10 Basic Instruction Reliability Test (2)
	This PDP-10 KL10 basic instruction reliability test is the second in a series of PDP-10 KL10 processor basic instruction random number diagnostics. This program tests every instruction in MEMORY and BOTH modes with the exception of fixed point multiply/divide, floating point, and byte instructions. This test is specifically set up to ensure reliability of the KL10 cache.
Hardware Required	KL10 mainframe/32K of core (minimum)
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	None
Notes	1. The nonexistent memory and parity stop switches should be reset (0). These errors, illegal UUOs and other errors of this type are handled by printout on the terminal. 2. The program runs within 32K of memory, but will test all of the first 256K of memory if the reliability or OPRSEL switches are set. 3. Cycle time - The cycle time depends upon memory size and increases as memory size increases. Normal Operation - approximately one minute in 32K. Reliability Mode - approximately 3 to 5 minutes.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	Refer to Table 1.

OPERATIONAL CONTROL

Normal operation with all switches set to 0 is quick-verify mode. For reliability test set the RELIAB switch. Without either the OPRSEL switch or the RELIAB switch set or the memory size selected via the switches, the program will run using only 32K. Setting just the OPRSEL switch allows the program to run in quick-verify mode using all available memory.

DFKBB TEST SUMMARY

In the majority of cases a random number is placed in the AC and C(E) and a series of 4 or 5 instructions are executed. The answer in the AC and/or C(E) is checked and an error message is printed if the number is not correct. Each set of 4 or 5 instructions acts on each memory location beginning at the last location of the program plus 1, and continuing to the designated end of memory.

Table 1 DFKBB Control Switch Summary

Switch	State	Description
0-17		Standard (Refer to the 10/10 STD module.) Exceptions - switches 2 and 12 are not used. Switches 9, 11 and 14 are redefined.
9	0	Quick-verify mode
	1	Reliability mode - use all available memory
11	0	Allow paging and trap enable
	1	Inhibit paging and trapping
14	0	Run in 32K unless in reliability mode (SW9 = 1)
	1	Use all available memory
18	0	Program runs unchached
	1	Run everything cached
27-35		Select size of memory to be used. Switches 27-35 correspond to the 9 high-order address bits. The low-order nine bit are appended as 1s (e.g., 27 - 35 = 077 equals 77777 or 32K). Switches 27 - 35 = 000 specifies all of memory is to be used.

ERROR SUMMARY

Memory parity interrupts caused by cache writebacks are intercepted and additional error information is printed giving the failing address, the testing address, the good data, the bad data and the XOR difference.

Errors are printed on the terminal or line printer. The error printout contains the test title, the PC of the failure, AC number, error word and correct word.

The PC value is useful in relating the failure to the listing.

When the scope loop mode is used the MI register will count for each occurrence of an error. If an audio indication of a continuing error is desired the DING switch may be set.

The following are the different error formats with their respective UUOs and error messages.

ERROR No. 1 - ERR AC,E

```
    EXAMPLE:                          AC              E
    2053 / CAME   AC1,AC2             ;RESULT         CORRECT
    2054 / ERR    AC,RAN1             ;ORIG  C(AC)    ORIG C(E)

    AC1=5                             ;TEST DATA
    C(AC1)  = 201532107642
    C(AC2)  = 201432107642
    C(RAN1) = 777777777777
    C(AC)   = 576345670135

    ERROR MESSAGE:
                                      (SOURCE OF NUMBERS PRINTED)
    PC     = 002054                   ;PC OF ERROR UUO
    AC     = 05                       ;AC FIELD OF UUO-1
    C(AC) = 201532107642              ;C(C(AC)) OF UUO-1
    COR    = 201432107642             ;C(C(ADDRESS FIELD)) OF UUO-1
         ORIGINAL
    C(AC) = 777777777777              ;C(C(ADDRESS FIELD)) OF UUO
    C(E)  = 576345670135              ;C(C(AC)) OF UUO
```

ERROR NO. 2 - ERRM AC,E

```
    EXAMPLE:                          AC              E
    2053 / CAME  AC2,MUD              ;CORRECT        RESULT
    2054 / ERRM  AC,RAN1              ;ORIG  C(AC)    ORIG C(E)

    MUD=5033                          ;TEST DATA
    C(MUD)  = 201532107642
    C(AC2)  = 201432107642
    C(RAN1) = 777777777777
    C(AC)   = 576345670135
```

```
ERROR MESSAGE:

                                        (SOURCE OF NUMBERS PRINTED)
    PC    = 002054                      ;PC OF ERROR UUO
    E     = 5033                        ;BITS 18-35 (E) OF UUO-1
    C(E)  = 201532107642                ;C(C(E)) OF UUO-1
    COR   = 201432107642                ;C(C(AC)) OF UUO-1
          ORIGINAL
    C(AC)= 777777777777                 ;C(C(E)) OF UUO
    C(E)  = 576345670135                ;C(C(AC)) OF UUO

ERROR NO. 3 - ERRI  AC,E

    EXAMPLE:                            AC              E
    2053 / CAME  AC1,AC2                RESULT          CORRECT
    2054 / ERRI  RAN1,(AC)              ;ORIG C(AC)     ORIG E

    AC1=5                               ;TEST DATA
    C(AC1)  = 107742670135
    C(AC2)  = 107642670135
    C(RAN1) = 777777777777
    C(AC)   = 576345670135

    ERROR MESSAGE:

                                        (SOURCE OF NUMBERS PRINTED)
    PC    = 002054                      ;PC OF ERROR UUO
    AC    = 5                           ;AC FIELD OF UUO-1
    C(AC) = 107742670135                ;C(C(AC)) OF UUO-1
    COR   = 107642670135                ;C(C(E)) OF UUO-1
          ORIGINAL
    C(AC) = 777777777777                ;C(C(AC)) OF UUO
    E     = 670135                      ;C(ADDRESS FIELD) OF UUO

ERROR NO. 4 - ERROR  AC,E

    EXAMPLE:                            AC              E
    2053 / CAME   AC,RAN
    2054 / ERROR  AC,RAN                ;RESULT         CORRECT

    AC=5                                ;TEST DATA
    C(AC)  = 201532107642
    C(RAN) = 201432107642

    ERROR MESSAGE:

                                        (SOURCE OF NUMBERS PRINTED)
    PC    = 002054                      ;PC OF ERROR UUO
    AC    = 5                           ;AC FIELD OF UUO
    C(AC) = 201532107642                ;C(C(AC)) OF UUO
    COR   = 201432107642                ;C(C(E)) OF UUO

ERROR NO. 5 - ER  AC,[ASCII/MESSAGE/]

    EXAMPLE:                            AC              E
    2053 / JFCL   10,.+2
    2054 / ER     AC,[ASCII/0V/]        ;RESULT         MESSAGE
    AC=5                                ;TEST DATA
    C(AC)   = 201432107642

    ERROR MESSAGE:

                                        (SOURCE OF NUMBERS PRINTED)
    PC   = 002054                       ;PC OF ERROR UUO
    AC   = 5                            ;AC FIELD OF UUO
    C(AC) = 201432107642 OV             ;C(C(AC)) OF UUO
                                        ;ADDRESS FIELD OF UUO POINTS TO
                                        ;AN ASCII MESSAGE

ERROR NO. 6 - ERM  AC,E

    EXAMPLE:                            AC              E
    2053 / SOJ    AC2,
    2054 / ERM    AC1,(AC2)             ;C(AC)          RESULT

    C(AC2)    = 5033                    ;TEST DATA
    C(AC)     = 740000005756
    C(C(AC2)) = 254000004041

    ERROR MESSAGE:

                                        (SOURCE OF NUMBERS PRINTED)
    PC    = 002054                      ;PC OF ERROR UUO
    E     = 5033                        ;BITS 18-35 (E) OF UUO
    C(AC) = 740000005756                ;C(AC) OF UUO
    C(E)  = 254000004041                ;C(C(E)) OF UUO
```

ERROR NO. 7 - ERMM AC,E

```
    EXAMPLE:                        AC              E
    2053 / SOJ   AC2,
    2054 / ERMM  AC1,(AC2)          ;C(AC)          RESULT

    C(AC2) = 5033                   ;TEST DATA
    C(AC1) = 740000005756

    ERROR MESSAGE:
                                    (SOURCE OF NUMBERS PRINTED)
    PC     = 002054                 ;PC OF ERROR UUO
    E      = 5033                   ;BITS 18-35 (E) OF UUO
    C(AC) = 740000005756            ;C(AC) OF UUO
```

ERROR NO. 11 - EERR ,E
ERROR NO. 12 - EERRM ,E
ERROR NO. 13 - EERRI ,E

ERRORS 11, 12 AND 13 ARE THE SAME AS ERRORS 1, 2 AND 3 EXCEPT THAT THE AC OF THE UUO IS REPLACED BY C(RAN). IN OTHER WORDS C(RAN) WILL BE PRINTED FOR THE ORIGINAL C(E).

GENERAL INFORMATION

Code	DFKCA.A10
Title	DECSYSTEM KL10 Advanced Instruction Diagnostic
Abstract	This PDP-10 KL10 advanced instruction diagnostic is the first in a series of PDP-10 KL10 processor floating point diagnostics. The diagnostic tests the functionality of the KL10 single-precision and double-precision floating point instructions.
Hardware Required	KL10 mainframe/32K of memory (minimum)
Preliminary Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	None
Notes	The cycle time of the program is in the millisecond range and is therefore suitable for taking margins, vibration tests, etc.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	Standard (Refer to the 10/10 STD module.) The following switches are not implemented: 1, 2, 9, 12, 14, 15, 16, and 17.

OPERATIONAL CONTROL
None - other than KLDDT and the control switches.

TEST SUMMARY
Refer to the listing on microfiche.

ERROR SUMMARY
Standard (Refer to the 10/10 STD module.)

GENERAL INFORMATION

Code	DFKDA.A10
Title	PDP-10 KL10 Arithmetic/Random/Interrupt/Memory Reliability Test
Abstract	This PDP-10 KL10 arithmetic/random/interrupt/memory reliability test is a comprehensive reliability test of the complete processor-memory subsystem. The program includes arithmetic instruction testing with random operands, random instruction testing, interrupt testing and memory reliability testing.
Hardware Required	KL10 mainframe/32K of core (minimum)
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	The OPRSEL switch must be set to allow the operator to be prompted and specify what test control switches are to be used on program startup.
Notes	1. This program is a reliability test using random numbers and as such should be run for several hours to ensure that enough random numbers are processed to verify system reliability. 2. To increase the duty factor and decrease the probability of simulator failures, switch FAST may be used. This, however, is not a complete test of the hardware and should be used with caution. 3. If it is desired to specify the initial base random number, set switch RANBAS (before starting) and respond to the following typeout accordingly. SPECIFY RANDOM NUMBER BASE - 4. The iteration count of the program is displayed in the memory indicators (MI). This count is a decrementing count and initially starts at -1 in stand-alone operation.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.) Special features start, 30004, clears all totals. In user mode under the diagnostic monitor the program will make two passes and return to the monitor. Type G to the monitor to continually run the program.
Control Switches	Refer to Table 1.

OPERATIONAL CONTROL

Operational control is via the console control switches. Switch 14, OPRSEL, set will cause a prompt for test selection.

DFKDA TEST SUMMARY

The arithmetic testing is done using pseudorandom numbers and comparing the machine results with each other and with a hardware operation simulation program.

The random instruction testing program is designed to execute random instructions (non-PC change) in core, in the fast ACs, and through software simulation. The results of the three groups of instructions are compared for equality. Upon finding a discrepency, the program prints all pertinent information and goes into a repetitive failure loop.

The interrupt portion of the diagnostic tests the priority interrupt system, the processor APR system and the interruptability of most classes of instructions. It also tests parity error interrupts, nonexistent memory interrupts, address breaks and BLT instruction interruptability. The DK20 meter is used to cause clock interrupts.

The memory reliability portion of the test is used to verify the operations of the memory subsystem. All of memory, up to 4096K, is used if available. A physical memory address test and a randomly selected data patterns test are performed. Memory addressing is verified by using a fast-rate addressing scheme whereby the selected address bit will change on every memory access. The fast-rate testing is randomly selected and, depending on the selection, will have all address bits fast-rated, any one address bit fast-rated, or no fast-rate testing performed.

Simulation Comparison. See control switch 34.
In this section instructions are executed and then simulated and the results compared. Actual execution of the instruction should give the same results as simulation of the instruction. Simulation is done by using software routines and pseudohardware registers to follow the hardware instruction sequences.

Instruction Comparison. See control switch 35.
In this section instructions are tested by performing a divide, then a multiply, and then add back in the division remainder. The original operands should be the same as the final results.

Table 1 DFKDA Control Switch Summary

Switch	Mnemonic	State	Description
0-17			Standard (Refer to the 10/10 STD module.) Exceptions - The following switches are not implemented: 11, 12, 15, 16, and 17.
18	ENTDDT	1	Enter DDT upon starting program
19	INHCLK	1	Inhibit clock interrupts
20	INHMEM	1	Inhibit memory testing
21	INHADB	1	Inhibit address break testing
22	INHI	1	Inhibit instruction interrupt testing
23	INHMI	1	Inhibit meter interrupt testing
24	INHBLT	1	Inhibit BLT interrupt testing
25	INHPAR	1	Inhibit parity interrupt testing
26	INHNXM	1	Inhibit NXM interrupt testing
27	INHNEW	1	Inhibit double-precision instructions
28	INHDFP	1	Inhibit double floating-point test
29	INHFP	1	Inhibit floating-point test
30	INHBYT	1	Inhibit BYTE test
31	INHFXD	1	Inhibit fixed-point test
32	INHRRAN	1	Inhibit random instruction
33	SNGFL	1	Run single fast loop
34	SLOW	1	Run just simulation comparisions
35	FAST	1	Run just instruction comparisions

ERROR MESSAGE SUMMARY

Program errors such as: illegal UUOs, parity errors, nonexistent memory, illegal interrupts, etc., are handled by printout of the type of error with as much information as is pertinent.

Instruction Error

If an instruction fails to give the correct results, the test title, pass count, type of failure (instruction comparision or simulation comparision), machine results, simulation results and instruction simulation are printed.

If something happens to the printout or you are not sure what the original and final operands are:

1. ACs are saved in locations SAVAC through SAVAC+17.
2. Original operands are in SAVAC+1, 2 and 3.
3. Results are in AC1, AC2 and AC3.
4. C(AC1)=C(AC)-, C(AC2)=C(AC+1), C(AC3)=C(E) or E.

Look in symbol table at end of listing for location of SAVAC.

Simulator Printout

When an error occurs the simulated machine states and registers are printed so that by use of hardware maintenance switches, i.e., single-pulse, console lights may be compared to the printout to determine the event time of the failure.

The printout contains the results of the instruction under test. This is AC, AC+1, E, or C(E) for fixed- and floating-point instructions; AC, pointer, C(E) for BYTE instructions. The following two lines indicate the instruction that failed, initial values, and simulated results. This is followed by the machine times and correct contents of the several registers, after the occurrence of the time pulse.

Scope Loop

When printing is complete, if the LOOPER switch is set, the routine will enter a loop suitable for scoping. At this time the MI register will display the results of the failing operand, or the flags register with machine results in C(left) and simulated results in C(right). If the failure is not flags, the MI will contain the first operand that failed in the order of AC, AC+1, and C(E).

GENERAL INFORMATION

Code	DFKFB.A10
Title	DECSYSTEM KL10 Instruction Timing Test
Abstract	The PDP-10 KL10 instruction timing diagnostic is designed to allow the execution times of the different classes of PDP-10 KL10 instruction to be timed. The time measurements are then used to ensure that the processor is operating correctly.
Hardware Required	KL10 mainframe/32K of memory (minimum)
Preliminary Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	None
Notes	1. The PDP-11 power line clock is used as the timing reference. 2. Program running time is less than 5 minutes.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	Refer to Table 1.

Table 1 DFKFB Control Switch Summary

Switch	Mnemonic	State	Description
0	ABORT	0 1	Normal operation Abort at end of pass
1			Not used
2			Not used
3	NOPNT	0 1	Normal operation Inhibit all printout (except forced)
4	PNTLPT	0 1	Normal operation Print on line printer
5-12			Not used
13	INHCSH	0 1	KL10 allow cache use KL10 inhibit cache use
14-35			Not used

OPERATIONAL CONTROL

The program will make one pass, printing out the timing information, and then halt.

Program may be restarted at location 30000 at any time.

TEST SUMMARY

A table is loaded with the instruction to be timed along with any initializing instructions needed. The table is repeatedly executed for one second while the number of iterations of the instruction sequence (the test instruction plus initializing instructions) is counted. The time for the instruction under test is, then, the time for the instruction sequence minus the time for the initializing instructions.

Assuming that the power line frequency is exactly 50 Hz or 60 Hz, the timing measurements are accurate to within 5 ns.

Any conclusions from the printed data must be drawn very carefully; the data is accurate but the nature of what was timed is sometimes misleading. For example, it would seem reasonable that the time necessary to do an index operation and an indirect operation separately, would be the same time required to do them in one instruction. Because the processor frequently waits for the memory cycle time, this is not always true.

Time Measurements
A description of the nonstraight-forward time measurements follows:

NOTE
The symbology T(X) means the time required for operation X.

ITEM 1: T(MAIN CLOCK) = <T(MOVEI 1, MUL [0[0 - T(MOVEI 1, MUL [252525252525]) >/18 THIS TIME IS THE SETTING OF THE MASTER CLOCK DELAYS.

ITEM 2: T(INDEXING) = T(MOVEI (3)) - T(MOVEI)

ITEM 3: T(INDIRECTING) = T(MOVEI @3) - T(MOVEI)

ITEM 4: T(INDEXING + INDIRECTING) = T(MOVEI @(3)) - T(MOVEI)

ITEM 12: T(PUSHJ) = T(PUSHJ + MOVEI) - T(MOVEI)

GENERAL INFORMATION

Code	DFKFD.A10
Title	DECSYSTEM KL10 I/O Bus Tester Diagnostic
Abstract	This PDP-10 KL10 diagnostic tests the advanced features of the KL10 API I/O bus tester.
Hardware Required	KL10 mainframe/32K of memory (minimum)/ I/O bus tester.
Preliminary Program	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	An I/O bus tester is required.
Notes	1. The nonexistent memory and parity stop switches should be reset (0). These errors, illegal UUOs and other errors of this type are handled by printout on the terminal. 2. The cycle time of the program is in the millisecond range and is therefore suitable for taking margins, vibration tests, etc.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	Refer to Table 1.

OPERATIONAL CONTROL

Reply to the following questions appropriately for selecting the I/O bus tester device code.

SPECIFY I/O BUS TESTER DEVICE CODE IN OCTAL -

Set I/O bus tester device code switches,
Type altmode when ready.

The I/O bus tester device cannot be the device code of any other device.

TEST SUMMARY

The diagnostic consists of the following tests.

1. Initialization
2. DATAO/DATAI register flip-flops
3. CONO/CONI register flip-flops
4. DATAO/DATAI - CONO/CONI interaction
5. I/O clear
6. Normal interrupts
7. Normal timed interrupts
8. DATAO functions
9. DATAO function addressing
10. DATAI functions
11. Increment functions
12. Increment function addressing
13. Dispatch functions
14. Dispatch function addressing

ERROR SUMMARY

Standard (Refer to the 10/10 STD module.)

Table 1 DFKFD Control Switch Summary

Switch	Mnemonic	State	Description
0-17			Standard (Refer to the 10/10 STD module.) Exceptions - The following switches are not implemented: 1, 2, 11, 12, 14, 15, 16, and 17.
18-26			I/O bus tester device code if running under the diagnostic monitor.
27-34			Not used
35	INHTIM	0 1	Normal operation Inhibit timed interrupts

DFRHB

GENERAL INFORMATION

Code	DFRHB.A10
Title	RH20 Fault Isolator
Abstract	DFRHB, the RH20 fault isolator diagnostic, is an executive mode program written in the MACRO-10 language. It is capable of testing from one to eight RH20 Massbus controllers. The program requires no drives to be attached to the tested RH20(s) but will operate successfully if there are drives present, providing that there are fewer than eight drives connected (i.e. there must be at least one nonexistant drive per tested controller). The program is incremental in nature and is made up of over 100 separate subtests. Scope looping, failing board callout, and comprehensive register and status printouts are provided.
Hardware Required	KL10 mainframe/32K of core (minimum)/RH20 (up to 8) and from 0 to 6 drives
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	There must be at least one nonexistent drive per controller.
Notes	1. The program does not guard against or attempt to clear any "attentions" which may be present from existing drives attached to the tested RH20(s). It may be necessary to power down existing drives for successful test execution. 2. Since the program is incremental in nature, relying on tested logic to check increasingly complex areas of the RH20, no provision has been made to allow operator selection of single subtests for execution. While such operation is feasible using DDT, it is not recommended. Program execution time is short enough that such operations should not be necessary.
Loading and Starting Procedure	Standard (Refer to the 10/10 STD module.)
Control Switches	Refer to Table 1.

OPERATIONAL CONTROL

Single RH20 Testing

To test a single RH20 (device code 540), load and start the program with OPRSEL switch reset (SW15). After identifying itself, the program will type:

TYPE NUMBER OF NONEXISTENT DRIVE ON ALL CONTROLLERS TESTED

and wait for operator input. A number from 0 to 7 should be typed. (Note: if the number typed corresponds to an existing drive, erroneous error reports will be indicated.)

Upon accepting the operator's drive selection, the program will proceed to incrementally test the RH20 Massbus controller. A single pass of the program takes less than a second, and the nonexistent drive request selection message is not printed on subsequent passes.

Multiple RH20 Testing
To test more than one RH20 or a single RH20 with a device code other than 540, set the OPRSEL switch (SW15) before starting the program. Upon starting, the program will type:

TYPE RH20S TO BE TESTED SEPARATED BY COMMAS. <CR> AT END

and wait for operator input. The operator should type the RH20 device codes (range 540-574 inclusive) in the order in which he wants them tested. The program will respond with:

TYPE NUMBER OF NONEXISTENT DRIVE ON ALL CONTROLLERS TESTED

and pause again. The drive number selected must be that of a drive which does not exist on any of the tested RH20s.

Upon accepting the operator's drive selection, the program will incrementally test each of the selected RH20s in the order indicated by the operator in response to the first query. Each RH20 is fully tested individually; i.e., all subtests are executed on the first RH20 selected before any testing is done on the second. The end of pass message will not be printed until all of the RH20s have been tested.

When testing multiple RH20s, error reports include the device code of the failing controller.

DFRHB TEST SUMMARY
The individual tests performed by this diagnostic are summarized in Table 2.

ERROR MESSAGE SUMMARY
Short printouts (SW10 set) include the PC of the failing subtest and generally a CORRECT, ACTUAL, and DIFFERENCE printout of a failing register.

Long printouts (SW10 reset) include in addition:

1. A diagnostic comment with failing board callout
2. A complete dump of all of the internal RH20 registers before and after execution of the failing subtest
3. An English language breakdown of the RH20 status register before and after subtest execution.

The device code of the failing RH20 is printed only if testing multiple RH20s or a single RH20 with device code other than 540(8).

Table 1 DFRHB Control Switch Summary

Switch	Mnemonic	State	Description
0-17			Standard (Refer to the 10/10 STD module.) Exceptions - The following switches are not implemented: 1, 2, 9, 11, and 12.
18	TRACE	0 1	Normal operation Report current test number
19-25			Not used
26	SWSEL	0 1	Normal operation Loop on test selected by switches 27 through 35
27-35	TEST No		Select specific test number to loop on if switch 26 set

Table 2 DFRHB Test Summary

Test	Description
	DEVICE CODE RESPONSE TEST
1	IOTCD1 - TEST FOR ILLEGAL DEVICE CODE RESPONSE 1. Set the selected RH20 PIA register to 7. 2. Attempt to clear the selected RH20 PIA bits using a CONO MBCN,0 in which one of the device select bits has been inverted. Do this for each bit of the device select code.
	PIA REGISTER AND ATTENTION ENABLE TESTS
2	PIA000 - PIA REGISTER AND ATTENTION ENABLE BIT (LOAD/READ TEST) 1. Load the PIA register and attention enable bit with test patterns. 2. Read the PIA register and attention enable bit after each load. 3. Verify that the expected data was read back.
3	PIA001 - PIA REGISTER AND ATTENTION ENABLE BIT (CLEAR - TEST) 1. Load PIA register and attention enable bit with test patterns. 2. Attempt to clear the PIA register and attention enable bit via a (CONO MBCN, RHCLR). 3. Read back the PIA register and attention enable bit after each attempted clear. 4. Verify that all the PIA register and the attention enable bits were cleared.
	PREP REGISTER LOAD/READ TEST BITS 0-6
4	PREP00 - PREP REGISTER BITS 0-6 (LOAD/READ TEST) 1. Load PREP register bits 0-6 with test patterns. 2. Read the PREP register after each load. 3. Verify that the expected data was read back.
5	PREP01 - PREP REGISTER 0-6 (CLEAR TEST) 1. Load the PREP register with a test pattern. 2. Attempt to clear the PREP register by executing a CONO MBCN, RHCLR. 3. Read the PREP register into ACTUAL. 4. Verify that PREP register bits 0-6 were cleared.
	PREP REGISTER BITS 9, 14-17 TEST
6	PREP02 - PREP REGISTER BITS 9, 14-17 1. Load each bit of the PREP register 9, 14, 15, 16, 17. 2. In turn, read back the PREP register and verify that bits 9, 14, 15, 16, 17 are as expected.
7	PREP03 - PREP REG CLEAR BITS 9, 14-17 (MBC0, RHCLR) 1. Load each bit of the PREP register bits 9, 14, 15, 16, 17. 2. Execute CONO MBC0, RHCLR and read back the PREP register. 3. Verify that bits 9, 14, 15, 16, 17 were cleared.
	INTERRUPT VECTOR INSTRUCTION REGISTER TEST
10	IVIR00 - IVIR REGISTER DATA TEST (BITS 27-35) 1. Load bits 27, 28, 29, 30, 31, 32, 33, 34, 35 with test data from the list. 2. Read the IVIR into ACTUAL. 3. Verify that the data is as expected.
11	IVIR01 - IVIR BASIC ADDRESSING TEST 1. Attempt to load 18 bits into the IVIR. 2. Read the IVIR into ACTUAL. 3. Verify that only the nine IVIR bits are read back.

Table 2 DFRHB Test Summary (Cont)

Test	Description
12	IVIR02 - IVIR CLEAR TEST (CONO MBC0, RHCLR) 1. Load all bits of the IVIR to one IVIR = 777. 2. Execute a CONO MBC0, RHCLR instruction. 3. Read the resulting contents of the IVIR into ACTUAL. 4. Test that all bits of the IVIR were cleared by the CONO.
	DRIVE EXCEPTION BIT TEST
13	EXEB00 - DRIVE EXCEPTION BIT TEST. (SET) This test will set both drive exception (DEXC) and end of block (EBL) bits in the diagnostic control register. This action will simulate the external drive doing the same thing. The program then checks status to determine if drive exception (DEXC) bit was set. 1. Set diagnostic register bits (DCREXC) and (DCREBL). 2. Read RH20 status into ACTUAL. 3. Test to see if DEXC was set.
14	EXEB01 - DRIVE EXCEPTION BIT TEST. (NOT SET) 1. Load only the diagnostic end of block (EBL) bit. 2. Read the RH20 status into ACTUAL. 3. Verify that the drive exception (DEXC) did not get set.
15	EXEB02 - DRIVE EXCEPTION BIT (NOT SET) 1. Load only the diagnostic exception (EXC) bit. 2. Read the RH20 status into ACTUAL. 3. Verify that the drive exception (DEXC) bit was not set.
16	EXEB03 - CLEAR DEXC BIT VIA A CMC 1. Set the drive exception (DEXC) bit using diagnostic (EBL) and (EXC). 2. Generate a clear Massbus control (CMC) command. 3. Read the RH20 status into ACTUAL. 4. Verify that the DEXC flip-flop was cleared.
17	EXEB04 - CLEAR (DEXC) VIA (TEC) 1. Set the drive exception (DEXC) flip-flop using EBL and EXC. 2. Generate a transfer error clear (TEC) command. 3. Read the RH20 status into ACTUAL. 4. Verify that DEXC flip-flop was cleared.
20	EXEB05 - TEST (SCRF) SECONDARY COMMAND REGISTER FULL (SET) 1. Load EBL and EXC to inhibit register swapping. 2. Load the STCR. 3. Read RH20 status into ACTUAL. 4. Verify that secondary command register full (SCRF) flag is set.
21	TASK57 - EBI CLEAR SCR FULL TEST 1. Set diagnostic bits (DCREBL and DCREXC) to inhibit register swaps. 2. Set secondary command register full (SCRF) flag. 3. Load the STCR. 4. Generate a clear Massbus controller (CMC) signal. 5. Read RH20 status into ACTUAL. 6. Verify that the SCRF flag was cleared.
22	TASK58 - STOP TRANSFER CLEARS SCR FULL TEST 1. Set the diagnostic bits (DCREBL and DCREXC) to inhibit register swapping.

Table 2 DFRHB Test Summary (Cont)

Test	Description
	2. Load the STCR to set the SCRF flag. 3. Generate a stop transfer (ST). 4. Read the RH20 status into ACTUAL. 5. Verify that the SCRF flag was cleared.
23	TASK59 - DELETE SCR CLEARS SCR FULL TEST 1. Load the diagnostic bits (DCREBL and DCREXC) to inhibit register swapping. 2. Load the STCR to set the SCRF flag. 3. Generate a clear secondary command register (CSCF) signal. 4. Read the RH20 status into ACTUAL. 5. Verify that the SCRF flag was cleared.
24	TASK60 - PCR FULL FLAG TEST 1. Load the STCR. This will swap and load the PTCR. 2. Read the RH20 status into ACTUAL. 3. Verify that the PCRF flag was set.
25	TASK61 - EBI CLEAR PCR FULL 1. Load the STCR. This will cause a register swap and load the PTCR and set the PCRF. 2. Generate a CMC signal. 3. Read the RH20 status into ACTUAL. 4. Verify that the PCRF was cleared.
26	TASK62 - SCR DELETE DOES NOT CLEAR PCR FULL 1. Load STCR. This will cause a register swap and load PTCR. 2. Generate a clear secondary command file (CSCF). 3. Read the RH20 status into ACTUAL. 4. Verify that the PCRF flag was not clear.
27	TASK63 - STOP TRANSFER CLEAR PCR FULL 1. Load STCR. This will cause a register swap and load PTCR. 2. Generate a STOP TRANSFER (ST). 3. Read RH20 status into ACTUAL. 4. Verify that the PCRF was cleared.
30	MBE001 - DOES CMC CLEAR BUS ENABLE 1. The Massbus enable (MBE) signal is set by (MBINI). 2. Generate clear Massbus controller (CMC). 3. Read RH20 status into ACTUAL. 4. Verify that MBE was cleared.
31	MBE003 - DOES RCLP CLEAR MBE 1. Massbus enable (MBE) is set by (MBINI). 2. Generate reset command list pointer (RCLP). 3. Read RH20 status into ACTUAL. 4. Verify that (MBE) was not cleared.
	FILE REGISTER TESTS
32	SBAR00 - SBAR DATA LOAD/READ TEST 1. Load each bit of the SBAR with test patterns from list FILE1T. 2. After each load read the contents of the SBAR into ACTUAL. 3. Verify that the readback is the same as the data sent.
33	SBAR01 - LOAD ALL BITS SBAR 1. Load all bits of the SBAR register to 1s regardless of whether or not they are implemented. 2. Read the SBAR into ACTUAL. 3. Verify that only the bits implemented in the SBAR (through ACTUAL) appear.

Table 2 DFRHB Test Summary (Cont)

Test	Description
34	STCR00 - STCR LOAD/READ DATA TEST 1. Set the DCREXC and DCREBL bits to inhibit the transfer of data from the STCR to the PTCR register. 2. Load test patterns through the STCR. 3. Read the STCR after each test pattern is loaded back into ACTUAL. 4. Verify that the correct data was returned.
35	STCR01 - STCR LOAD ALL BITS 1. Set DCREXC and DCREBL bits to inhibit the transfer of data from the STCR to the PTCR register. 2. Set all bits of the STCR to 1s regardless of whether they are implented or not. 3. Read back the STCR into ACTUAL. 4. Verify that only the bits implemented are read back through ACTUAL.
36	PBAR00 - PBAR DATA LOAD/READ TEST This test tries to verify that all four registers in the RAMS can be loaded with test patterns and read back. This test is complicated by the fact that loading the STCR makes the STCR-SBAR become the PTCR-PBAR. 1. Load the STCR to make the registers swap. 2. Load test data into the new SBAR (remember it used to be PBAR). 3. Read back the data into ACTUAL. 4. Verify that the expected data was read.
37	PBAR01 - LOAD ALL BITS INTO THE SBAR THAT WAS THE PBAR. 1. Load the STCR to make the registers swap. 2. Load all bits of the SBAR whether implemented or not. 3. Read the SBAR register into ACTUAL.
40	PTCR00 - PTCR DATA (LOAD/READ TEST) 1. Load the STCR to cause a register swap to occur. 2. Set diagnostic EBL (end-of-block) and EXC (exception) to inhibit any further register swaps. 3. Load the PTCR (remember it is now the STCR) with test patterns. 4. Read back the data from the current STCR into ACTUAL. 5. Verify that the data in ACTUAL is as expected.
41	PTCR01 - LOAD ALL BITS TO PTCR TEST 1. Load the STCR to cause a register swap. 2. Load the diagnostic EXC and EBL to prevent any further register swaps. 3. Load the current STCR with all 1s whether implemented or not. 4. Read the current STCR back into ACTUAL. 5. Verify that only implemented bits were read back.
42	PTB00 - PBAR REGISTER READ TEST 1. Load the SBAR register with test patterns. 2. Load the STCR to cause a register swap. 3. The SBAR that was loaded is now the PBAR, so read PBAR into ACTUAL. 4. Verify that PBAR contains the same data was loaded into SBAR.
43	PTB01 - PBAR ATTEMPT TO LOAD ALL BITS TEST 1. Attempt to load all bits in the SBAR whether implemented or not. 2. Load the STCR to cause a register swap. 3. The SBAR that was loaded is now the PBAR, so read PBAR into ACTUAL. 4. Verify that only implemented bits are read back.

Table 2 DFRHB Test Summary (Cont)

Test	Description
44	STR00 - LOAD STCR READ PTCR ADDRESSING TEST 1. Load test patterns into the STCR (this will cause a register swap). 2. Read the test data from the PBAR into ACTUAL. 3. Verify that the same data was read from PTCR as was loaded into STCR.
45	STR01 - LOAD ALL BITS STCR READ PTCR TEST 1. Attempt to load all bits in the STCR whether implemented or not. 2. Read the PTCR into ACTUAL. 3. Verify that only implemented bits were read back.
46	FLE000 - LOAD EACH FILE REG WITH A UNIQUE PATTERN AND READ 1. Load SBAR to 111111. 2. Load STCR to 222222. 3. The STCR and SBAR should have swapped and become PBAR and PTCR. 4. Load SBAR to 333333. 5. Load STCR to 444444. 6. The registers should have swapped again. 7. Read all four registers. 8. Verify that the following conditions exist. SBAR = 111111 STCR = 222222 PBAR = 333333 PTCR = 444444
	DATA PATHS TEST
47	DPA00 - TEST DATA PATH FOR STUCK AT HIGH 1. Load the read register (RR) to all 0s. 2. Read the read register into ACTUAL. 3. Verify that the read register appears to be all 0s.
50	DPA01 - DATA PATH END AROUND ENABLE GATES TEST (GATES ARE DISABLED) 1. Load the write register (WR) to 777777. 2. Disable the data path by setting DCRRWR in the diagnostic register to 0. 3. Read the read register into ACTUAL. 4. Verify that the read register is 000000.
51	DPA02 - DATA PATH TEST 1. Load test patterns from list WRTS1T into the write register. 2. Read the contents of the read register into ACTUAL. 3. Verify that the contents of read register equal the data that loaded into write register.
	DATA PATH PARITY TESTS
52	DPA03 - PARITY CHECKER DATA PATHS DBPE = 1 A test is made of the parity check logic. A simulated read command is executed and the data being read has an even number of bits set but does not have the parity bit set. This produces a parity error condition which should set the data bus parity error (DBPE) flip-flop. A sufficient number of error-causing patterns are used to ensure that all gates internal to the parity checking networks are tested. 1. Load the secondary transfer command register (STCR) with read command. This will allow the parity checking logic to be tested. 2. Load test patterns from list DPA2T into TPE write register. 3. Read the RH20 status into ACTUAL via a CONI command. 4. Check that the DBPE bit was set to a 1.

Table 2 DFRHB Test Summary (Cont)

Test	Description
53	DPR00 - PARITY CHECKER DATA PATH DBPE = 0 This test is similar to the previous parity test except the patterns used contain an odd number of bits so that no error condition should exist. 1. Load the secondary transfer command register (STCR) with a read command. 2. Load test patterns from list DPA1T into the write register. 3. Read the RH20 status into ACTUAL via a CONI command. 4. Verify that the DBPE bit was not set.
54	DPR01 - TEST FOR TEC CLEARS DBPE 1. Load write register to all 0s including a 0 parity bit. 2. Generate a transfer error clear (TEC). 3. Read the RH20 status into ACTUAL. 4. Verify that the data bus parity error (DBPE) bit was cleared.
55	DPR02 - TEST THAT CMC CLEARS DBPE 1. Load write register to all 0s including a 0 parity bit. 2. Generate a clear Massbus controller (CMC). 3. Read the RH20 status into ACTUAL. 4. Verify that the data bus parity error bit was cleared.
	FILE REGISTER VIA MASS CONTROL TESTS
56	SBMS00 - SBAR READ BACK FROM MASS CONTROL A sequence of test patterns are loaded into the STCR and caused to transfer over to the mass control board as if it were executing a read data transfer command. The low-order 16 bits of the STCR are stored in the mass control and can be read back under diagnostic control. These bits 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35 are read back and compared with the corresponding bits in the STCR. 1. Set diagnostic control bits BAR and CB (block address register test and control bus test bits.) 2. Load test patterns from list FILE2T into secondary block address register (SBAR). 3. Load the STCR to cause a register swap. 4. Read an external register; i.e., a register not in the RH20 control. In this way it can be seen what was sent to the MSS board. 5. Verify that the low-order 16 bits from the MSS are the same as were loaded into the SBAR. Note that when scoping this test, the DATAI read will occur after the first SBAR data is sent out and before the second (same) is sent. This is because lack of TRA causes a DRE delaying the second output. Ignore the second SBAR output as it is meaningless.
57	STMS00 - STCR READ BACK FROM MASS CONTROL A sequence of test data is loaded into the STCR. This causes a data transfer to the mass control board similar to the transfer caused in normal operation. The low-order 16 bits 20-35 are read back from the mass control board and checked. This test is more a test of the STCR SBAR control logic then a data test, since the data path has already been proven. 1. Set diagnostic control bits block address register test (BAR) and control bus test (CB). 2. Load test patterns from list FILE3T into STCR. 3. Read an external register; i.e., a register not internal to the RH20. 4. Verify that the low-order 16 bits from the external register are the same as were loaded into the STCR.

Table 2 DFRHB Test Summary (Cont)

Test	Description
60	PTMS00 - PTCR READ BACK FROM MASS CONTROL BOARD This test is very similar to the two previous tests except that the PTCR is transferred and read back. 1. Set diagnostic control bits BAR and CB. 2. Load test patterns from list FILE3T into STCR. 3. Read an external registe;, i.e., a register not internal to the RH20. 4. Verify that the low-order 16 bits are the same as were loaded into the STCR.
61	SBST00 - TRANSFER DATA FROM SBAR This subroutine tests for the ability to transfer data from the SBAR. Prior to the first transfer the STCR and PTCR are both empty. After the first transfer the register that was STCR becomes PTCR and the previous PTCR becomes STCR. It is this SBAR that is to be tested. 1. Set the diagnostic bits DCRBAR, DCRCB, and DCRTRA. 2. Load STCR to 0s with disable transfer error stop (DTES). 3. Generate a stop transfer (ST). 4. Generate transfer error clear (TEC). 5. Load the STCR with all 0s and DES set. 6. Read an external register into ACTUAL. 7. Compare the data loaded into the SBAR with that received back.
	EXTERNAL DATA CONTROL LOGIC TESTS
62	EXTN00 - EXTERNAL DATA PATH FOR THE CONTROL LOGIC This test will determine if data can be passed through the external control logic in end-around mode and be read back into the CPU. 1. Load test patterns into the external register from test list EXRG1T. 2. Read the contents of the external register into ACTUAL. 3. Verify that the expected data was read back from the external register.
	EXTERNAL CONTROL LOGIC PARITY
63	EXTN01 - EXTERNAL PARITY PE = 0 A pattern of test data in which the parity is always 0 is loaded and read through the external control logic. Both the parity and the data are tested; however, we are most interested in the parity bit. The number sequence used is designed to test every gate in the parity networks. 1. Load test data from test list CBPR1T into the external register. 2. Read the contents of the external register into ACTUAL. 3. Verify that the correct data including parity was returned.
64	EXTN02 - EXTERNAL PARITY PE = 1 This test is the same as the previous test except that the parity bit to be set is tested. 1. Load the external register with data from list CBPR2T. 2. Read the contents of the external into ACTUAL. 3. Verify that the correct data was read back and that the parity bit was set.
65	EXTN03 - EXTERNAL PARITY ERROR PE = 0 The diagnostic register test bit DCREP (diagnostic even parity check) is set. This will ensure that any pattern read will be read as if it were expected to be in even parity. Since all data was written in odd parity, parity errors will always appear, thus determining that the parity checking logic is working. 1. Load diagnostic bits DCREP, DCRCB, abd DCRTRA. 2. Load test patterns from list CBPR1T into the external register. 3. Read the external register into ACTUAL. 4. Verify that the parity error flip-flop is set.

Table 2 DFRHB Test Summary (Cont)

Test	Description
66	EXTN04 - EXTERNAL PARITY ERROR PE = 0 This test is the same as the previous except that a data test pattern is used such that the opposite parity is generated.
67	EXTN05 - GENERATE PARITY ERROR This parity test writes in even parity and reads in odd parity, thereby creating parity errors for each pattern. 1. Load STCR 2. Load an external register with test patterns from list CBPR1T. 3. Read the external register into ACTUAL. 4. Verify that the correct data was read.
70	EXTN06 - PARITY TEST FOR CONTROL PATHS 1. Load STCR. 2. Load an external register with test patterns from list CBPR2T. 3. Read the external register into ACTUAL. 4. Verify that the correct data was read back. FALSE REGISTER LOAD TESTS The following five load tests load each register in turn with a known test pattern using the expected register address etc. An attempt is then made to load over (i.e., zero out) the register under test by loading all other internal registers and those external registers deemed most likely to overwrite the register being tested. After each attempt at overwriting, the register is read to ascertain that in fact it was not overwritten.
71	FLSB00 - SBAR FALSE LOAD TEST (REG 0) 1. Set diagnostic DCREBL, DCRCB, DCRTRA, and DCREXC bits. 2. Load a test pattern into the secondary block address register (SBAR). NOTE On second pass an attempt will be made to load all 0s into a different register. Only on the first loop through this code is the SBAR register loaded. 3. Read the SBAR into ACTUAL. 4. Verify that the SBAR was not cleared.
72	FLST00 - STCR FALSE LOAD TEST (REG 1) Refer to description of Test 71.
73	FLDC00 - DCR FALSE LOAD TEST (REG 7) Refer to description of Test 71.
74	FLWR00 - WR FALSE LOAD TEST (REG 6) Refer to description of Test 71.
75	FLIV00 - IVIR FALSE LOAD TEST (REG 4) Refer to description of Test 71. FALSE REGISTER READ TESTS The following six false read tests are similiar to the false load tests except that a given register is loaded, then an attempt is made to read it using all other internal register addresses and several external register addresses deemed most likely to cause trouble.
76	FRAB00 - SBAR FALSE READ TEST 1. Load a test pattern into the register under test from list SBAR2T. 2. Attempt to read the register.
77	FRST00 - STCR FALSE READ TEST (REG 1) Refer to description of Test 76.

Table 2 DFRHB Test Summary (Cont)

Test	Description
100	FRPB00 - PBAR FALSE READ TEST (REG 2) Refer to description of Test 76.
101	FRPT00 - PTCR FALSE READ TEST (REG 3) Refer to description of Test 76.
102	FRRR00 - RR FALSE READ TEST (REG 5) Refer to description of Test 76.
103	FRIV00 - IVIR FALSE READ TEST (REG 4) Refer to description of Test 76.
	PREP REGISTER LOAD LOGIC TEST
104	DATA00 - DATAO INH TEST FOR PREP REGISTER LOAD A DATAO inhibit condition is created by attempting to load a nonexistent register (i.e., external register number 37). A test is made to determine that in fact a DATAO inhibit condition will prevent loading of the PREP register. 1. Clear the diagnostic control register (DCR). 2. Load the external register. This will set register access error (RAE) which will set DATAO INHIBIT. 3. Attempt to read the IVIR register into ACTUAL. 4. Verify that the interrupt vector index register (IVIR) cannot be read.
	INTERRUPT LOGIC TEST
	The next two tests will test the interrupt logic associated with the diagnostic register and the PIA logic. This will allow for further testing of the entire interrupt logic associated with each interrupt level.
105	EBI00 1. Set the attention interrupt enable (AIE) bit. 2. Set the diagnostic drive attention (DCRATT) bit. 3. Read RH20 status into ACTUAL. 4. Verify that the drive attention (DA) bit was set.
106	EBI02 - EBI ATT ENÀ +NOT MASS ATT 1. Load the attention interrupt enable (AIE) bit. 2. Read RH20 status into ACTUAL. 3. Verify that drive attention (DA) was not set.
	INTERRUPT TESTS
107	INTT00 - DOES INTERRUPT OCCUR ON PIA 0? 1. Set the interrupt vector index register (IVIR) to address 400. 2. Save all the SUBRTN package data in the storage area. 3. Set the interrupt logic for level 0. 4. Cause an interrupt condition. 5. If an interrupt occurs, it is an error.
110	INTT01 - INTERRUPT ALL PRIORITY LEVELS (DO NOT TEST FOR EXPECTED CHANNEL)
111	INTT02 - PIA ALL CHANNELS WITH NO INTERRUPT CONDITION
112	INTT03 - INTERRUPT TEST ONLY SELECTED CHANNEL ENABLED
113	INTT04 - INTERRUPT TEST ALL EXCEPT SELECTED CHANNEL ENABLED
114	TASK64 - TRANSFER BIT TEST Set the diagnostic transfer bit to simulate a real device transfer bit. Read back an external register, test for the presence of the transfer bit in bit 10. 1. Load the diagnostic DCRTRA bit. 2. Attempt to read an external register. 3. Save the transfer (TRA) bit read back from the external register. 4. Verify that the TRA bit was set.

Table 2 DFRHB Test Summary (Cont)

Test	Description
	ERROR BIT CHECKING
115	STP000 - DOES COMMAND DONE (CD) GET CLEARED BY CONO 1. Set command done (CD) by executing a read command. 2. Generate a (CONO MBCN, 0) (a NOP command). 3. Read RH20 status into ACTUAL. 4. Verify that (CD) was not cleared.
116	TASK68 - REGISTER ACCESS ERROR (RAE) ERROR BIT TEST 1. Clear the diagnostic register bits. 2. Attempt to load the nonexistent external register. 3. Wait for the (RAE) register access error to set. 4. Read RH20 status into ACTUAL. 5. Verify that the (RAE) register access error was set.
117	TASK69 - DOES CMC ZERO REGISTER ACCESS ERROR (RAE) BIT 1. Clear the diagnostic register bits. 2. Attempt to load the nonexistent external register. 3. Wait enough time for register access error (RAE) to set. 4. Generate clear Massbus controller (CMC) 5. Read RH20 status into ACTUAL. 6. Verify that the (RAE) was cleared.
120	TASK70 - DOES CRAE CLEAR REGISTER ACCESS ERROR (RAE) BIT 1. Clear the diagnostic register bits. 2. Attempt to load a nonexistent external register. 3. Wait enough time for the RAE to set. 4. Generate clear register access error (CRAE) 5. Read RH20 status into ACTUAL. 6. Verify that RAE was cleared.
121	TASK72 - DOES PREP9 INHIBIT DATAO INH 1. Clear all the diagnostic register bits. 2. Load the interrupt vector index register (IVIR) with (777). 3. Attempt to load a nonexistent external register. 4. Wait enough time for register access error (RAE) to set. 5. Attempt to load IVIR to (000). 6. Read the IVIR into ACTUAL. 7. The IVIR should still be (777).
122	TASK75 - CAN WE READ COMMAND DONE ON A ONE? 1. Load STCR with a read command. 2. Generate stop transfer (ST) 3. Read RH20 status into ACTUAL. 4. Verify that command done (CD) was set.
123	TASK76 - DOES CLRCD CLEAR COMMAND DONE? 1. Load STCR with a read command. 2. Generate ST to set command done (CD). 3. Generate clear command done (CLRCD). 4. Read RH20 status into ACTUAL. 5. Verify that command done (CD) was cleared.
124	TST76A - TEST PREP BIT 9 DTES = 0 1. Load the STCR with a read command. 2. Load the WR with bad parity to cause an error. 3. Read RH20 status into ACTUAL.
125	TST76B - TEST PREP BIT 19 DTES = 1
126	TST76C - TEST DTES FOR EXC
127	TST76D - TEST EXC WITHOUT DTES TO SET COMMAND DONE 1. Load STCR with a read command. 2. Set diagnostic DCREBL and DCREXC bits. 3. Read RH20 status into ACTUAL. 4. Verify that command done (CD) was set.

Table 2 DFRHB Test Summary (Cont)

Test	Description
130	TST76E - DOES REGISTER ACCESS ERROR (RAE) TIMEOUT SET CD DONE? 1. Set diagnostic DCRCB bit. 2. Load STCR with 0s. 3. Load (55) into external register. 4. Wait enough time for register access error (RAE) to set. 5. Read RH20 status into ACTUAL 6. Verify that command done (CD) was set.
	DRIVE RESPONSE ERROR TESTS
131	DRE000 - DRIVE RESPONSE ERROR. TEST CAN IT BE SET? 1. Clear the diagnostic control register. 2. Attempt to load a nonexistent external register. 3. Wait enough time for register access error (RAE) to set. 4. Clear (RAE) via a CONO MBCN, RAECLR command. 5. Load STCR again to set PCR OUT. 6. Clear transfer error via CONO MBCN, TEC. 7. Read RH20 status into ACTUAL. 8. Verify that DRE was set.
132	DRE001 - DOES TEC CLEAR DRE? 1. Clear the diagnostic bits. 2. Attempt to load a nonexistent external drive register. 3. Wait enough time for register access error (RAE) bit to set. 4. Clear RAE bit via a CONO MBCN, RAECLR command. 5. Reload the STCR. 6. Clear transfer error via CONO MBCN, TEC. 7. Verify that DRE was cleared.
133	DRE002 - DOES DRE SET COMMAND DONE? 1. Clear the diagnostic control register. 2. Attempt to load a nonexistent external register. 3. Wait enough time to allow register access error (RAE) to set. 4. Clear RAE flag. 5. Reload STCR. 6. Read RH20 status into ACTUAL. 7. Verify that command done (CD) was set.
	MASSBUS CMC TEST
134	TASK77 - DOES CMC CLEAR COMMAND DONE? 1. Load STCR. 2. Do a stop transfer (ST) to set command done (CD) 3. Generate CMC. 4. Read RH20 status into ACTUAL. 5. Verify that CD was cleared.
	VECTOR REGISTER TEST
135	TASK77A - ATTEMPT TO WRITE WITH A DATAI Load the internal register IVIR with 525. Attempt to disturb the IVIR by doing a DATAI with bit 6 L/R set. These two commands (i.e. DATAI, bit 6 = 1) are mutually exclusive; there should be no disturbing of the data in the IVIR by this. 1. Load the interrupt vector index register (IVIR) with 777. 2. Attempt to write into IVIR with a DATAI. 3. Read IVIR into ACTUAL. 4. Verify that the IVIR still had 777 in it.
136	TST77B - DO DATAI BIT 6 = 1 EXTERNAL 1. Load the interrupt vector index register (IVIR) with 777. 2. Attempt to load IVIR to 252. 3. Read the IVIR into ACTUAL. 4. Verify that IVIR still contains 777.

Table 2 DFRHB Test Summary (Cont)

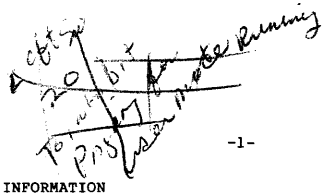
Test	Description
137	TST77D LOAD IVIR - LOAD EXTERNAL THEN TEST IVIR 1. Load IVIR to 525. 2. Attempt to load the external register with 252. 3. Read the contents of IVIR into ACTUAL. 4. Verify that the contents of the IVIR were not disturbed.
140	TST77E LOAD EXTERNAL - LOAD IVIR INTERNAL THEN TEST EXTERNAL 1. Load the external register with 1777777. 2. Load the interrupt vector index register (IVIR) with 525. 3. Read the contents of the external register into ACTUAL. 4. Verify that the external register was not disturbed by the IVIR.
	EXTERNAL REGISTER TEST
141	TST77F - LOAD 374000 THEN 370000 THEN READ 1. Load the external register with 052525. 2. Attempt to write 125252 over it without setting the load bit. 3. Read the contents of the external register into ACTUAL. 4. Verify that the external register still contains 052525.
142	TST77G - DATAO EXTERNAL DATAI EXTERNAL DO WE KILL DATA? 1. Load the external register with 052525. 2. Do a DATAI to see if it gets cleared. 3. Read the contents of the external register into ACTUAL. 4. Verify that the external register still contains 052525.
143	TST77H Load external, then SBAR, then test for all 1s from external and all 0s from SBAR (test for shorts in mixer). 1. Load the external register to 177777. 2. Load the SBAR to 000000. 3. Read the SBAR into ACTUAL. 4. Verify that 000000 is read.
144	TST77J - SAME AS ABOVE, REGISTERS REVERSED 1. Load the SBAR to 3617,,777777. 2. Load the external register to 000000. 3. Read the contents of the external register into ACTUAL. 4. Verify that the content of the external register is 000000.
	BLOCK COUNTER TESTS
145	TASK79 - BLOCK COUNTER TEST This test is run from a list that contains the load for the block counter (i.e. test patterns), and the number of EBL that must be generated in order to cause block counter overflow. The actual contents of the block counter are not readable, so we must check that it counts correctly (overflows with the correct number of EBLs). After the block counter is loaded, the program generates one less EBL than the proper number needed to cause overflow. A test is made to see if it overflowed prematurely. If it did this is detected as an error. One more EBL is generated and tested to see if it overflowed properly. 1. Load the STCR block counter. 2. Set up an EBL tally in a register. 3. Generate one less EBL than required. 4. Test for overflow - if overflow, then error. 5. Generate one more EBL pulse. 6. Test for overflow. If no overflow, then error. 7. Do this for all patterns in the list.

Table 2 DFRHB Test Summary (Cont)

Test	Description
146	TASK80 - CD DONE INTERRUPT TEST 1. Load STCR with a read command. 2. Set up IVIR to handle the expected interrupt. 3. Turn on CPU interrupt logic. 4. Wait enough time for an interrupt to occur. 5. Dismiss interrupt.
147	TASK81 - DATAO INHIBIT INTERRUPT TEST
150	STPX00 - DOES TRANSFER SET LONG WORD COUNT? If the stop transfer command is issued while the MBC is executing a command, the long word count will be set. This implies that there are still more words to be transferred, thereby allowing the program to take expected action. 1. Load STCR with a write command. 2. Do a stop transfer (ST) command. 3. Read RH20 status into ACTUAL. 4. Verify that LWE was set by ST.
	REGISTER NUMBER 4 (AS) DECODE TESTS
151	REG400 - DOES ATT REG NUMBER 4 DECODE AS NUMBER 4? The external register number 4 is a special status register which is made up of bits from all drives. Normally the TRA bit comes up to tell the logic that the data is ready. Since this register is made up of data from all drives, there is no way that the TRA bit can be made to refer to any particular drive. For this reason special logic exists to ensure that RAE will never come up while accessing register number 4. 1. Access an external register. 2. Read RH20 status into ACTUAL. 3. Verify that register access error (RAE) was set.
152	REG401 - DOES ANY REGISTER OTHER THAN NUMBER 4 DECODE AS A NUMBER 4 1. Clear diagnostic register bits. 2. Access a register from the test list (not register number 4). 3. Read RH20 status into ACTUAL. 4. Verify that register access error (RAE) was set.
	DATA TRANSFER TESTS
153	WRT000 - WRITE ONE WORD MBOX TO RH20 1. Set up channel command word. 2. Load the SBAR. 3. Load STCR with a write 1-word command. 4. Read in this word to ACTUAL. 5. Generate EBL to terminate transfer. 6. Test for proper data transfer. If data has changed, an error occurred. 7. Check status register for proper status.
154	WRT001 - WRITE 128 DATA WORDS FROM MBOX TO MBC 1. Set up channel command word. 2. Load STCR. 3. Transfer data and compare results. 4. Generate EBL to terminate transfer properly. 5. Check for proper status.
155	RAD000 - READ ONE WORD RH20 TO MBOX 1. Set up channel command word. 2. Load the STCR with the read command. 3. Transfer the word into ACTUAL. 4. Generate EBL to properly terminate the transfer. 5. Check the data transferred. If not the same, an error occurred. 6. Check for proper status.

Table 2 DFRHB Test Summary (Cont)

Test	Description
156	RAD001 - READ 128 WORDS FROM RH20 TO MBOX 1. Set up channel command word. 2. Load the STCR with the read command. 3. Transfer the word into ACTUAL and check it. 4. Check the data transferred. If not the same, an error occurred. 5. Generate EBL to properly terminate the transfer. 6. Check for proper status.
157	RAD002 - READ 128 WORDS FROM RH20 TO MBOX WITH FIRST WORD HAVING WRONG DATA PARITY COMPUTED. 1. Set up channel command word. 2. Load the STCR with the read command. 3. Transfer the word into ACTUAL and check it. 4. Check the data transferred. If not the same, an error occurred. 5. Generate EBL to properly terminate the transfer. 6. Check for proper status.
160	TSK104 - LONG WORD COUNT TEST 1. Set up channel command word for LWE. 2. Load STCR for LWE when a write operation is in progress. 3. Generate an EBL. 4. Read RH20 status into ACTUAL. 5. Verify that LWE was set.
161	TSK105 - SHORT WORD COUNT TEST Read 126 words from RH20 to MBox when clocking 128 words to cause short word count error (SWE). 1. Set up channel command word. 2. Load the STCR with the read command. 3. Transfer the word into memory. 4. Check the data transferred. If not the same, there is an error. Expect two words in error due to SWE. 5. Generate EBL to properly terminate the transfer. 6. Check for proper status.
162	TSK106 - SHORT WORD COUNT TEST WHILE DOING A WRITE TRANSFER Write 126 words from MBox to RH20 device when RH20 is expecting 128 words from the MBox (i.e. 2 words short). Check transfer termination status. Read transfer operation. The device supplies data to memory.



GENERAL INFORMATION

Code	DFRPH.A10
Title	RH20/RP04 Basic Device Diagnostic
Abstract	DFRPH is the RH20/RP04 basic device diagnostic, designed to isolate solid RP04 faults to the faulty module or group of modules within the DCL or drive electronics. Excellent scope looping capabilities have been designed into each test. Testing is done on a start small and build up basis. The diagnostic starts by testing even the most trivial of control bus cycles and ends up by doing full-speed data transfer operations to various disk addresses. Positioning logic is also tested along the way. In addition to the straight line diagnostic tests, the diagnostic also contains a head alignment/verification subroutine that is invoked by setting the proper sense switch.
Hardware Required	KA10, KI10 or KL10 CPU/48K of core (minimum)/RH20 (up to 6)/up to 48 (single- or dual-port) RP04s
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	Examination of switch 13 takes place only once, immediately following the initial load of the diagnostic into core from its storage media. The cache option is selected only during the very first program initialization cycle. Attempting to alter the state of the switch from that point on will have no effect.
Note	Refer to DDRPH summary for documentation on this diagnostic. DDRPH tests RH10s-RP04s. This diagnostic tests RH20s-RP04s. The major difference between the two diagnostics is the code (instruction used to implement the testing). Where differences in documentation occur between this diagnostic and DDRPH, both are listed in the DDRPH summary.

GENERAL INFORMATION

Code	DFRPK.A10
Title	RH20/RP06 Basic Device Diagnostic
Abstract	DFRPK is the RH20/RP06 basic device diagnostic, designed to isolate solid RP06 faults to the faulty module or group of modules within the DCL or drive electronics. Excellent scope looping capabilities have been designed into each test. Testing is done on a start small and build up basis. The diagnostic starts by testing even the most trivial of control bus cycles and ends up by doing full-speed data transfer operations to various disk addresses. Positioning logic is also tested along the way. In addition to the straight line diagnostic tests, the diagnostic contains a head alignment/ verification subroutine that is invoked by setting the proper sense switch.
Hardware Required	KA10, KI10 or KL10 CPU/48K of core (minimum)/RH20 (up to 6)/up to 48 (single- or dual-port) RP06s
Preliminary and Associated Programs	Refer to diagnostic hierarchy (10/10 STD module).
Restrictions	Examination of switch 13 takes place only once, immediately following the initial load of the diagnostic into core from its storage media. The cache option is selected only during the very first program initialization cycle. Attempting to alter the state of the switch from that point on will have no effect.
Note	Refer to DDRPH summary for documentation on this diagnostic. DDRPH tests RH10s-RP04s. This diagnostic tests RH20s-RP06s. The major difference between the two diagnostics is the code (instruction used to implement the testing). Where differences in documentation occur between this diagnostic and DDRPH, both are listed in the DDRPH summary.

GENERAL INFORMATION

Code — DFSXA.A10

Title — DECSYSTEM KL10 Channel/DTE20 Interaction Test

Abstract — DFSXA is the KL10 channel/DTE20 interaction test for the KL10 computer system. It provides the service engineer with a software tool that permits interactive testing of all data channels into and out of the KL10's internal memory. The program can test up to eight KL10 I/O channels (RH20 Massbus controllers) along with up to four front-end PDP-11 channels (DTE20 KL10 to PDP-11 interfaces).

The RH20 channels may be tested in one of the following two modes of operation:

1. In internal loopback mode (deviceless)

2. By writing/reading data to/from either an RP04/06 disk pack or a TU16/45 magtape drive.

The DTE20 channels may be tested in one of the following two possible modes of operation.

1. With only the minimum PDP-11 resident software required to transfer data between 11 and 10 core

2. With a preconfigured DEC/X11 systems exerciser load module resident in the PDP-11 front end and exercising the 11 front-end devices in parallel with the execution of DFSXA on the 10 side. (Note: This is only permissible for restricted front ends and does not apply to the master DTE that is running KLDCP.

The program provides comprehensive error and status reports that permit the service engineer to evaluate system performance and aid in detecting and isolating interactive system problems to the faulty subsystem. This is the only program within the set of DECSYSTEM-20 diagnostics that provides simultaneous testing of both the PDP-11 front ends and the KL10 I/O channels.

Hardware Required — KL10 mainframe/32K of core (minimum)/RH20s (up to 8) with or without devices.

Preliminary and Associated Programs — All processor and memory diagnostics (both 11/10 and 10/10). All diagnostics for those devices to be tested (refer to the 10/10 STD and 11/10 STD modules).

Restrictions — Any RH20s to be tested in loopback without devices connected must be properly terminated.

Notes

1. Execution Time - The time for one complete program pass may vary anywhere from one minute to greater than one hour depending on the system configuration being tested.

2. Trace Printout - Each time the program begins a new test it checks the PALERS switch on the PDP-11. If it is set the program types out an arrow followed by the test number.

3. Loading Secondary Front Ends - If the DECX11 switch is not being used (SW25=0) DFSXA will do two things.

   First, it will perform a BOOT/DUMP test on all selected front ends (except that front end containing KLDCP) at startup. The following printout occurs as each front end is tested

BOOT-DUMP TEST FE# n

Where n = front-end number.

Second, DFSXA will load each front end selected for test with a copy of DFSXAX (11 code resident in DFSXA). As each front end is loaded, the following message gets typed

LOADING DFSXAX INTO FE# n

Where n = front-end number.

4. Recommended Usage - To verify system integrity and detect interaction problems, DFSXA should be run in the following modes.

   a. Without the DEC/X11 switch set and all channels in loopback mode.

   b. Without the DEC/X11 switch set and all channels having devices using a device while the remaining RH20s operate in loopback.

   c. Repeat step 2 except operate the front ends in DEC/X11 mode. This will require loading and starting the DEC/X11 run time exerciser for each front end before starting DFSXA.

   Use the error and status reports provided by DFSXA to localize the problem to the minimum failing subsystem. (Use the S and R commands to selectively modify the test configuration dependent on the error information provided.)

   Once the problem has been isolated to a subsystem, use the individual subsystem diagnostics to isolate the problem to the field replaceable unit.

5. AC USAGE - Throughout most of the tests, the accumulators are used to perform specific program functions as described in Table 2.

Loading and Starting Procedure: Standard (Refer to the 10/10 STD module.)

Control Switches: Refer to Table 1.

OPERATIONAL CONTROL
Upon starting, DFSXA will print the following introductory message.

```
DECSYSTEM KL10 CHANNEL/DTE20 INTERACTION TEST (DFSXA)
VERSION #.#, SV=#.##, CPU#=####, MCV=###, MCO=#, HO=##, ##HZ

SWITCHES = ssssss ssssss
CLK SOURCE = NORMAL, CLK RATE = FULL, AC BLK 0, CACHE: 0 1 2 3

MEMORY MAP =
FROM        TO          SIZE/K
00000000    01377777    384

PI LEVEL (1-7)?

DATA MASK(0-777777777777)?

SYSTEM CONFIGURATION:
```

NOTE
The system configuration as determined by the program is printed here.

```
THE AVAILABLE COMMANDS ARE:
```

NOTE
A list of commands is printed here.

After printing the introductory message DFSXA will:

1. Enter console command mode if OPRSEL (SW14) is set. An asterisk (*) indicates DFSXA is ready to accept commands.

2. Auto-start if OPRSEL (SW14) is reset. Auto-start exercises the default system configuration (all RH20s in loopback mode and all front ends in non-DEC/X11 mode).

Command Summary
DFSXA commands are summarized in Table 3.

TEST SUMMARY
DFSXA consists of 27(8) separate tests carefully designed and logically sequenced to provide maximum fault resolution (to a failing subsystem) with the minimum active hardware configuration. The tests are summarized in Table 4.

ERROR MESSAGE SUMMARY
There are several different error message formats dependent upon the function being checked when the error was detected. The formats follow.

FAULT ERROR MESSAGE FORMAT

```
***TRANSMISSION ERROR DURING WRITE***
ERROR PC: 031133
<C0:>
CM:     CNR,MBE,CMD:    PIA: 1
PT: DTE,GO:     NBC: 1  MFC: WRITE
W1:             CLP:   57351
W2:     WC:     0       DAC:0
EA: SBUS,NXM,IOPF,MBPAR,CDIR,ADRP,PWR,  PIA: 0
ER:
MA:     00032354        WD: 1     RF: CPU MEM RD
CHN PRG
0/ CHN JMP TO 57347
57347 / XFER 20 WRDS TO 43245
57350 / SKIP 160 WRDS & LAST XFR
```

FAULT1 ERROR MESSAGE FORMAT

```
***NOT ALL RH20 STATUS BITS=0***
ERROR PC: 030775
<C0:>
CM:     CNR,MBE,        PIA: 0
PT: DTE;GO:     NBC: 1  MFC: READ
W1:             CLP:  57350
W2:     WC:     0       DAC:  44245
EA: SBUS,NXM,IOPF,MBPAR,CDIR,ADRP,PWR,  PIA: 0
ER:
MA:     00000002        WD: 2    RF: CPU MEM RD
```

FAULT2: ERROR MESSAGE FORMAT

```
***DATA ERROR IN BOOT-DUMP TEST***
PC:     062730
CH: 13  REC SIZE: 256.  WRDS BAD: 237.  BUF ADR: 0

ADR         BAD             GOOD            XOR
00043245    000000 000000   000000 000240   000000 000240
00043246    777777 777777   000000 000000   777777 777777
00043247    777777 777777   000000 000000   777777 777777
00043250    777777 777777   000000 000000   777777 777777
00043251    777777 777777   000000 000000   777777 777777
```

FAULT3 ERROR MESSAGE FORMAT

```
***     RANDOM EVENT TESTING     ***
ERROR PC: 031766
+++ CHANNEL 0 LOGOUT ERROR +++

GOOD DAC: WC=0 AND ADR= 0037456
BAD DAC: WC=1 AND ADR= 0037455
```

FAULT4 ERROR MESSAGE FORMAT

```
*** CHANNEL #N DEVICE #M (ERROR DESCRIPTOR) ***
ERROR PC: PPPPPP
TS: (DTE STATUS REG - ENCODED)
TC: (TO10 BYTE COUNT REG - ENCODED)
```

NOTE
This is followed by the run summary printout for all selected devices.

+++ THIS IS A FATAL ERROR - THE UNIT IS BEING DROPPED! +++

FAULT5 ERROR MESSAGE FORMAT

(NOT USED IN THIS REV. RELEASE)

FAULT6 ERROR MESSAGE FORMAT

```
*** CHANNEL #13 DEVICE # 0 FATAL BOOTSTRAP ERROR ***
ERROR PC: 062566
TS: RM,TOllDN,PI0ENB,   PIA:  0
TC:     CNT: 0
```

Performance Reports

There is only one performance report and that is the runtime summary. This printout is invoked each time the program enters a new test if the TOTALS switch is set to a 1. It is also printed as part of a FAULT4 error message when fatal errors are reported. The summary appears as follows.

```
TEST SUMMARY-RUNTIME: 0:0:37
<C0:>
WORDS WRITTEN: 45056.   WRITE RETRIES: 0.    WRITE ERRORS: 0.
WORDS READ: 4224.       READ ERRORS: 0.      DATA ERRORS: 0.
CHN ERRORS: 0.
<C1:>
WORDS WRITTEN: 45056.   WRITE RETRIES: 0.    WRITE ERRORS: 0.
WORDS READ: 4224.       READ ERRORS: 0.      DATA ERRORS: 0.
CHN ERRORS: 0.
<C2:>
WORDS WRITTEN: 45056.   WRITE RETRIES: 0.    WRITE ERRORS: 0.
WORDS READ: 4224.       READ ERRORS: 0.      DATA ERRORS: 0.
CHN ERRORS: 0.
<C3:>
WORDS WRITTEN: 45056.   WRITE RETRIES: 0.    WRITE ERRORS: 0.
WORDS READ: 4224.       READ ERRORS: 0.      DATA ERRORS: 0.
CHN ERRORS: 0.
```

Table 1 DFSXA Control Switch Summary

Switch	Mnemonic	State	Description
0-7			Standard (Refer to the 10/10 STD module.)
8	PALERS	0	Normal operation PRINT TEST #6
9	RELIAB	0	Normal operation
		1	Loop on tests 23 through 27 on passes after initial pass
10-12			Not used
13	INHCSH	0	Allow cache use
		1	Inhibit cache use
14	OPRSEL	0	Enter auto-test mode upon program load or restart (same as typing A to command interpreter)
		1	Enter command interpreter upon intial load or program errors
15	CHAIN		This switch used by DIAMON to control chaining operations
16-17			Not used
18	INHUSR	0	Run all tests
		1	Inhibit user mode tests (tests 26 and 27)
19-24			Not used
25	DEC/X11	0	Load DF SXA11 from DFSXA to test front ends

		1	Inhibit loading DFSXA11 from DFSXA to allow using DEC/X11 load module
26	LOOPTS	0	Sequence through all tests
		1	Loop on current test continuously
27	SPCTST	0	Normal operation
		1	Loop on test specified by switches 28-35
28-35	TSTNUM		Select test to loop on if switch 27 = 1

Table 2 DFSXA AC Usage

AC	Symbol	Use
AC17	P	Stack pointer
AC16	CNTRL	Generally contains the channel no. being tested
AC15	DEV	Contains the device no. if one is being tested
AC14	LIMIT	General-purpose counter for iteractive loops
AC13		General use
AC12	PNTR	Pointer
AC11	ERRDAT	Used to index error stack
AC10	PATRN	Used to index data patterns table
AC7	CCW	Used to index CCW word test table
AC6	GOOD	Contains what data should have been
AC5	BAD	Contains what data actually was
AC4	CLKCNT	Clock count index
AC3	DATAPT	Data pointer
AC2,1,0		General-purpose use

Table 3 DFSXA Command Summary

Command	Description
A	A<CR> This command instructs DFSXA to size the system, and start the exerciser testing the default configuration. This results in all available RH20s being tested in loopback mode and all front ends being tested in non-DEC/X11 mode.
BL	BL F2<CR> This command is used to load a selected front end with a PDP-11 binary file. This file is normally the preconfigured copy of the DEC/X11 exerciser program for the front end (other than the master front end). DFSXA will respond by typing: FILE: Type the NAME of the binary file followed by a <CR> DFSXA will then respond by typing: EXT: Type the file extension followed by a <CR> DFSXA will now call DIAMON and load the PDP-11 front end with the file named. After the PDP-11 has been loaded it will be self-started and ready to accept input from its console. NOTE DIAMON must be loaded and started. The file to be loaded must be on the load device and the load device must be selected. The front-end device must be selected. Refer to the S command.
BH	BH F2<CR> This command does the same as the BL command except that the program loaded into the PDP-11 does not self-start. When you are ready to run the PDP-11 program, press the CONTINUE switch on the PDP-11.
BT	BT F2<CR> This command causes DFSXA to transfer data to and from the selected front-end PDP-11 core. The data transferred is checked and any errors detected during transmission are reported.
CP	CP<CR> This command causes DFSXA to read the CPU status words and print them on the console terminal. DFSXA responds by printing: EA: XXX,XXX,XXX,XXX,XXX, etc. ER: XXX,XXX,XXX,XXX etc. where EA is the content of the left half of the CPU status and ER is the content of the right half. XXX indicate and identify which bits were asserted (set=1).
D	D<CR> This command transfers control from DFSXA to KLDDT to allow the user to use DDT commands.
E	E<CR> This command allows the user to print out the contents of certain registers within the RH20 or device interfaces. Refer to the HELP file (type H) to get a complete description of how to use the E command. Also refer to the N command.
G	G<CR> This command is used to instruct DFSXA to start exercising the system using those devices selected by the last set of select commands.

Table 3 DFSXA Command Summary (Cont)

Command	Description
H	H<CR> This command causes DFSXA to type out the help file which summarizes how to use the various commands.
MB	MB<CR> This command causes DFSXA to print out the contents of the MBox status. DFSXA responds by printing: MA: AAAAAAAA WD: N RF: XXX,XXX,XXX etc. where A, N, X indicate the status.
N	N<CR> This command causes DFSXA to type out a list of all the register mnemonics that can be used by the E command to examine device registers.
PC	PC<CR> This command causes DFSXA to size the system and print out a list of all devices found on the target system. DFSXA will respond by printing: SYSTEM CONFIGURATION: C0<T3,0,> C1<P0,> C2<> C3<> C4<> C5<> F0M: F3R: which indicates the following configuration: A tape unit on channel 0 as device #3 and slave device #0 A pack on channel 1 as device 0 Channels 2 through 5 in loopback A master front end #0 A restricted front end #3
PT	PT<CR> This command causes DFSXA to type out the contents of the test selection table. DFSXA responds by printing: CURRENT DEVICE SELECTION C0<3,0,> C1<0,> C2<> C3<> C4<> C5<> F0: F3: etc.
R	R C2:1<CR> This command is used to remove one or more devices from the current selection table. In this case device 1 on channel 2 is removed.
S	S C4:3<CR> This command is used to add one or more devices to the test selection table. In this case the third device on the fourth channel is added.
TM	TM<CR> This command instructs DFSXA to type out program runtime. DFSXA responds by printing: RUNTIME: HH:MM:SS

Table 3 DFSXA Command Summary (Cont)

Command	Description
Z	Z<CR> This command causes DFSXA to clear the device selection table so that no devices are selected to run.

Table 4 DFSXA Test Summary

Test	Description
TST01	This test verifies that the MBox is halted and then tests that the RH20 status bits of all selected channels have been properly initialized during program startup.
TST02	This test verifies that the Massbus enable bit can be set in all selected RH20s.
TST03	This test verifies the ability of each selected channel to perform a noninteractive write operation in noninterrupt mode.
TST04	This test verifies the ability of each selected channel to perform a noninteractive read operation in noninterrupt mode.
TST05	This test repeats the write operation of TST03 except it is executed in interrupt mode.
TST06	This test repeats the read operation of TST04 except it is executed in interrupt mode.
TST07	This test verifies the ability of the channel to execute a read reverse channel control word.
TST10	This test verifies the ability of each selected channel to jump through core (channel JUMP CCW), halt properly, and store the correct channel status.
TST11	This test verifies that all selected channels can write interactively.
TST12	This test verifies that all selected channels can read interactively.
TST13	This test verifies that all selected channels can read reverse interactively.
TST14	This test verifies that all selected channels can successfully perform a write while the other channels are reading.
TST15	This test verifies that each selected channel can successfully perform a write while the other channels are executing a read reverse.
TST16	This test verifies that each selected channel can successfully perform a read operation while the other channels are performing a write.
TST17	This test verifies that all channels can properly execute a SKIP channel control word (CCW).
TST20	This test verifies the RH20 ZERO FILL feature for all selected channels.
TST21	This test verifies that each selected channel can read from every core location above the program's first free location.
TST22	This test verifies that each selected channel can write into every core location above the program's first free location.

Table 4 DFSXA Test Summary (Cont)

Test	Description
	NOTE If the reliability mode switch option is selected, tests 01 through 22 are executed on the first pass only. Subsequent passes iterate the random tests 23 through 27 continuously until the program is stopped by the user (Control C).
TST23	This test verifies that all selected channels can write and read back using random variables as the test parameters. The random variables are: 1) data size, 2) data pattern, and 3) memory buffer address.
TST24	Same as TST23 except CPU interactive testing is performed in the background.
TST25	Same as TST23 except that memory interactive testing is performed in the background.
	NOTE If the INHUSR switch is set, the next two tests (26 and 27) are inhibited.
TST26	Same as TST23 except that a CPU interactive test is executed in user mode in parallel with the random channel operations.
TST27	Same as TST23 except that a memory test is executed in user mode in parallel with the random channel operations.

CONTENTS

DECAID	Contents
	Introduction
	Troubleshooting Flow Diagrams
AC/DC LO	Location of AC LO and DC LO signals on CPU backplanes
BUS	Quiescent Unibus levels Location of signals on Unibus slot backplane pins
CONSA	Quiescent console light displays
CONSB	PDP-11/05 console checkout procedure
HELP	What to do if all else fails
INT OPT	List of internal options including modules, location, required jumper and wire changes
LOADER	For paper tape systems: procedures for loading and running the bootstrap and absolute loader and for loading the MAINDECs For disk, DECtape, or magtape systems: procedures for using the hardware and software bootstrap loaders
MAINDA	List of available CPU, memory, and internal option MAINDECs
MAINDB	Acceptable times for running MAINDECs
MEMA	Module utilization and jumper configuration and location for MM11-S, MF11-L(P), and MF11-U(P) memories, and SSYN delay adjustment procedure for parity memory
MEMB	Strobe adjustment procedure for memories using the G110

CONTENTS (Cont)

DECAID	Contents
PROCA05	GPR program, module utilization, SCL clock adjustment, proc. clock adjustment
PROCA40	Proc. clock adjustment, jumper location and configuration, module utilization
PROCA45	Module utilization, jumper location, and configuration
PROCB	CPU modules that drive and receive Unibus signals
PWRB-1	PDP-11/40/45 CPU dc voltage checks
PWRB-2	PDP-11/40/45 regulator slot assignments
PWRB-3	PDP-11/40/45 crowbar problems
PWRB-4	PDP-11/40/45 distribution panel connectors
PWRB-5	PDP-11/40/45 power harnesses and cables
PWRB-6	PDP-11/40/45 reguators' connectors
PWRB-7	PDP-11/40/45 procedure for removal of regulators, fuse location
PWRB-8	PDP-11/40/45 bulk supply connectors
PWRB-9	PDP-11/40/45 fuse location on and removal procedure for bulk supply's power control board
PWRD	PDP-11/05/10 power connectors
REMOTE	Remote power control facility
START	Suggestions on what to do before actual hardware troubleshooting

THE PDP-11 MAINFRAME TROUBLESHOOTING GUIDE

INTRODUCTION

This guide is comprised of two major sections:

1. The Troubleshooting Flowchart
2. The DECAIDs.

It is intended that they be used concurrently by Field Service personnel to aid them in locating and repairing faults in PDP-11 systems that utilize PDP-11/05/10/35/40/45 CPUs.

The guide makes two basic assumptions.

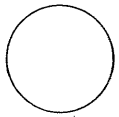
1. The Field Service technician has an excellent supply of replacement parts (e.g., modules, power regulators, cables, etc.).

2. There is a configured copy of DEC/X11 with the system which has been loaded and run error-free at some time prior to this failure. That is, the system has already been installed and was fully operational at one time. The guide does not provide the necessary information on how to install either systems or add-ons to a system.

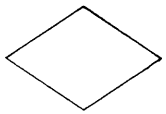
THE TROUBLESHOOTING FLOWCHART

The flows guide the Field Service technician through the steps he must perform to isolate a fault to either a peripheral device or the mainframe and if it is found to be in the mainframe, then to either the Unibus, power system, internal option, memory, or processor. Once the fault has been isolated, the flows then indicate the steps to follow to repair the fault on a field-replaceable-unit basis.

The symbology used in the flowchart is as follows:

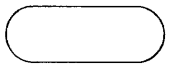


Circle
An entry from or exit to other pages of the flows with the pages listed in parentheses near the circle.



Diamond
A decision point.

Rectangle
A procedure or operation to be performed by the technician.



Bubble
A reference to one of the DECAIDs that provides the technician with the information necessary to perform the operation or make the decision.

The 13 pages of flows are organized as follows:

1.–2. Starting and quick-check; loading and running system exercisers

3.–4. AC power

5.–6. DC power

7. Mainframes

8. Unibus

9.–10. Consoles

11. Processors

12. Memories

13. Peripherals.

The flowchart is intended to fulfill two primary functions.

1. To aid the technician in determining a logical troubleshooting approach to a system failure.

2. To help the technician develop an effective troubleshooting technique.

THE DECAIDS

The second section of the quide provides the troubleshooter with all the information necessary to perform the operations, do the adjustments, and make the decisions to isolate and repair a problem in a PDP-11 system. These DECAIDs cover all the detailed aspects of troubleshooting from how to load a diagnostic to the procedure for adjusting core memory strobes to locating the output pins on a power regulator's connector.

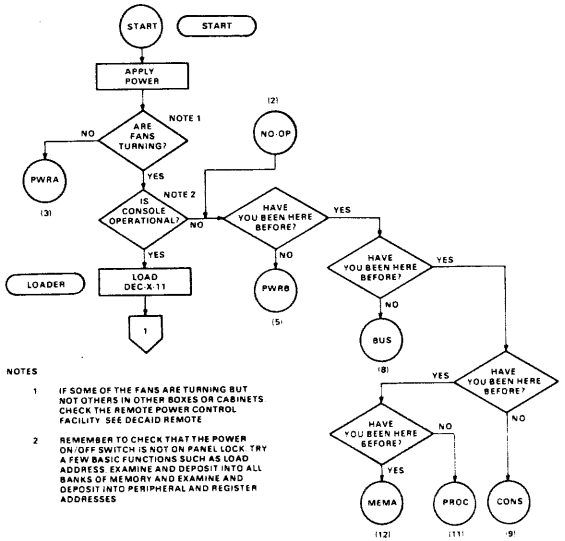
It is intended that these DECAIDs will be used by the technician on the job, providing information that is:

1. Much too detailed to memorize

2. Hidden away in one of more than a dozen manuals and engineering drawings

3. Not available anywhere else.

TROUBLESHOOTING FLOWCHART 1

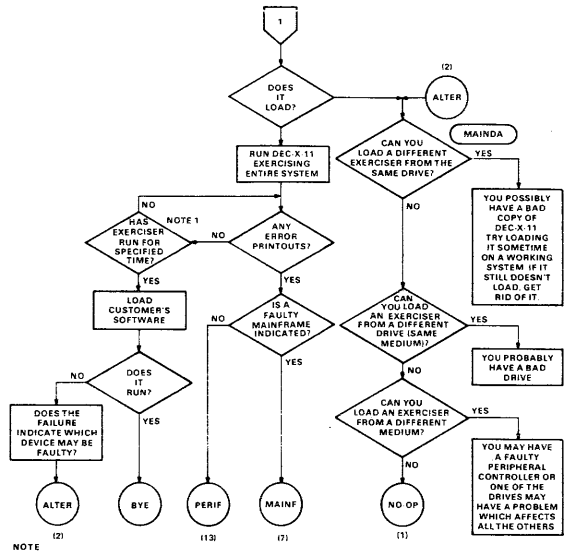


NOTES

1 IF SOME OF THE FANS ARE TURNING BUT NOT OTHERS IN OTHER BOXES OR CABINETS CHECK THE REMOTE POWER CONTROL FACILITY SEE DECAID REMOTE

2 REMEMBER TO CHECK THAT THE POWER ON/OFF SWITCH IS NOT ON PANEL LOCK TRY A FEW BASIC FUNCTIONS SUCH AS LOAD ADDRESS EXAMINE AND DEPOSIT INTO ALL BANKS OF MEMORY AND EXAMINE AND DEPOSIT INTO PERIPHERAL AND REGISTER ADDRESSES

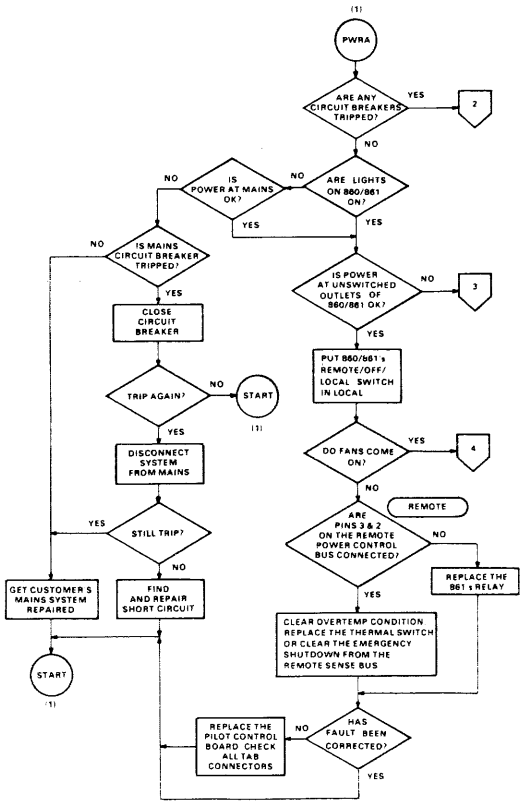
TROUBLESHOOTING FLOWCHART 2



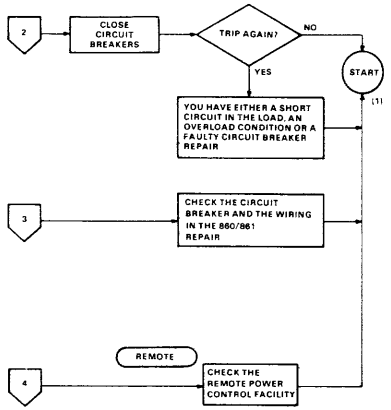
NOTE

1 WHEN RUNNING A MAINDEC IT IS SOMETIMES DIFFICULT TO DETECT WHETHER IT IS RUNNING PROPERLY OR WHETHER IT IS HUNG IN A LOOP. READ THE MAINDEC'S WRITE-UP TO SEE IF YOU CAN USE SUCH TECHNIQUES AS SETTING UP THE SWITCH REGISTER TO CAUSE A FREQUENT PRINTOUT, EXERCISING THE CONSOLE TERMINAL'S PRINTER, RUNNING IN SINGLE ITERATION MODE, ETC

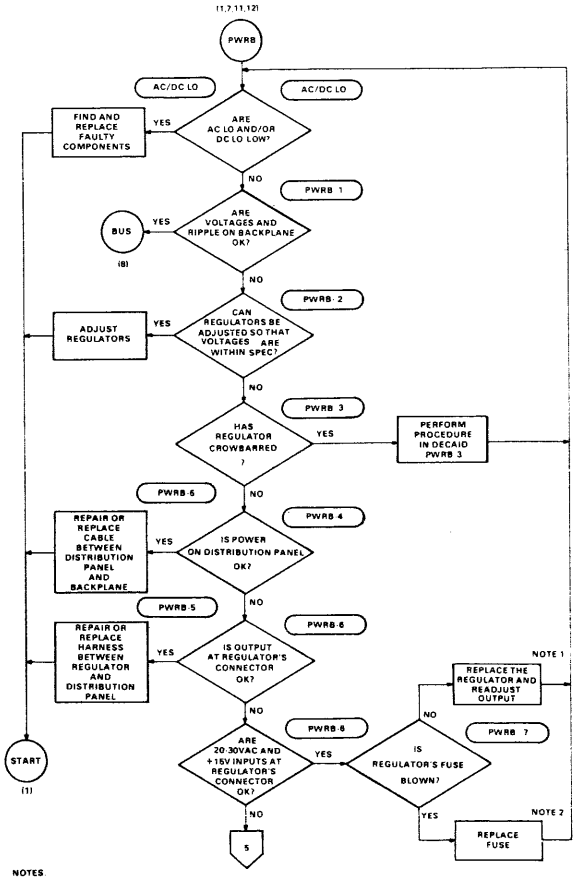
TROUBLESHOOTING FLOWCHART 3



TROUBLESHOOTING FLOWCHART 4



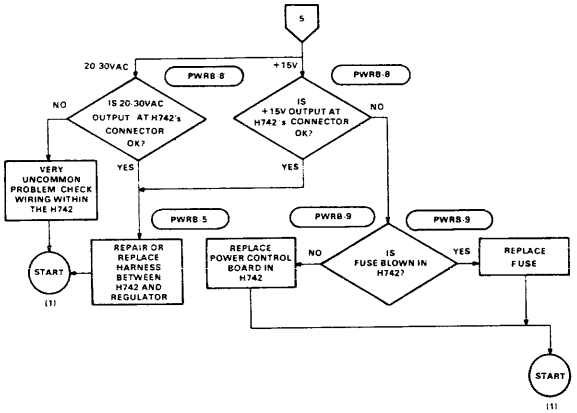
TROUBLESHOOTING FLOWCHART 5



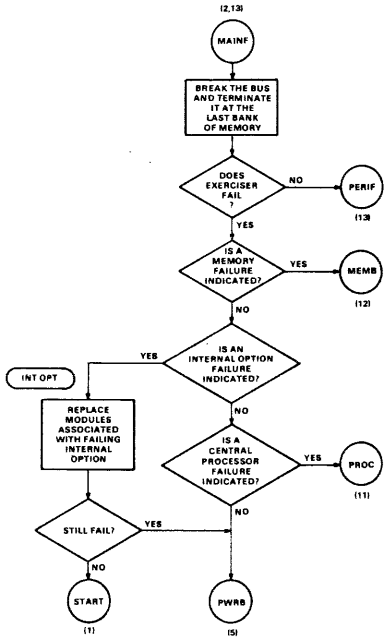
NOTES:

1. Before replacing the fuse or regulator, check for possible causes. Faulty components within the regulator itself can cause the fuse to blow. Check for scorching of the etch board and/or components. Check the backplane for a bent pin or a piece of wire which might be causing a short circuit.
2. If system is an 11/05/10, check the voltages as listed in DECAID PWRD.

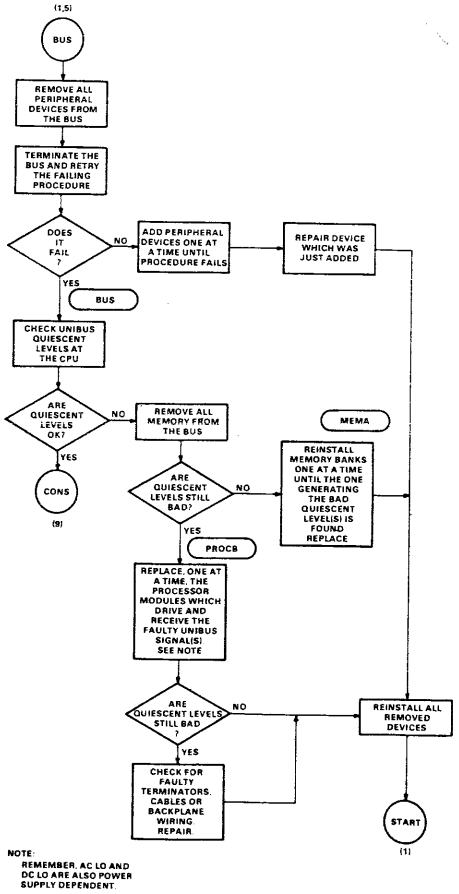
TROUBLESHOOTING FLOWCHART 6



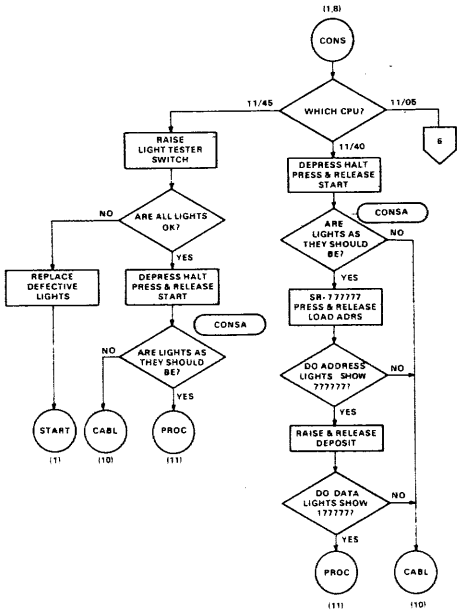
TROUBLESHOOTING FLOWCHART 7

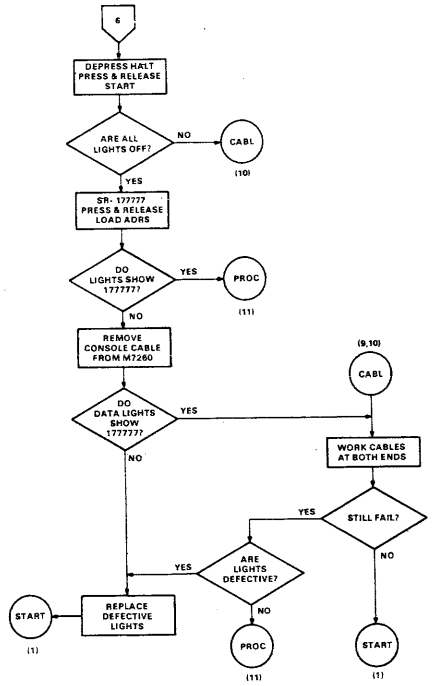


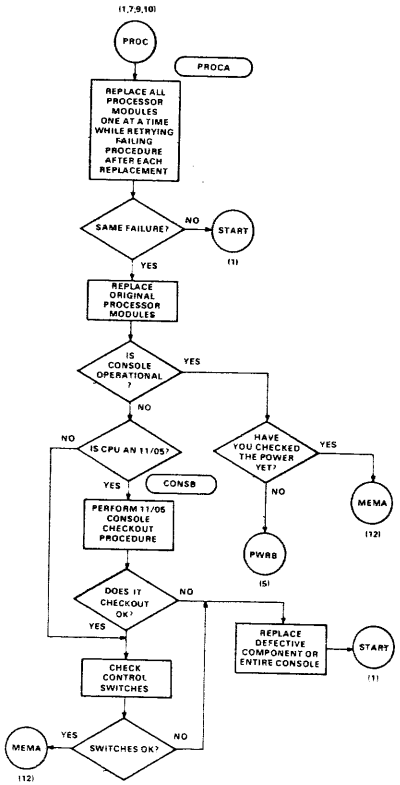
TROUBLESHOOTING FLOWCHART 8



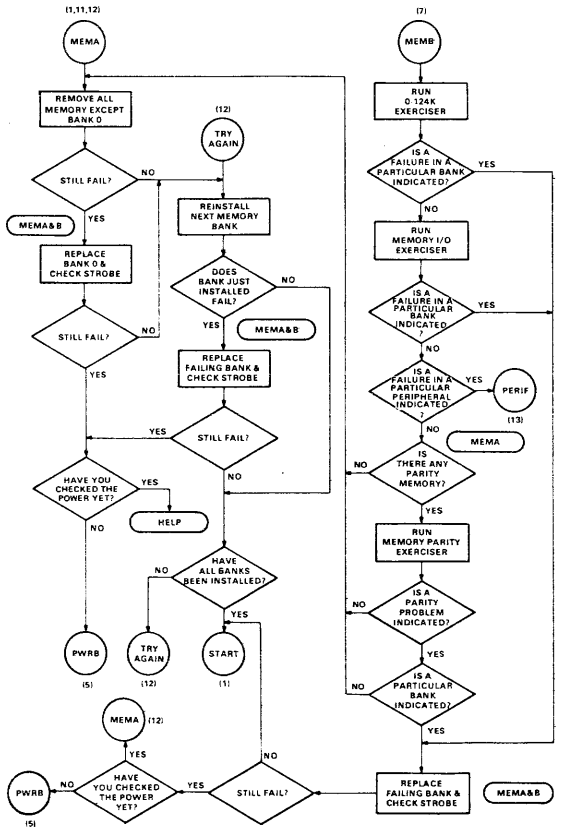
TROUBLESHOOTING FLOWCHART 9

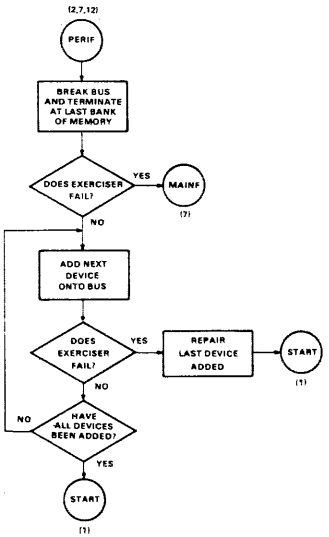



6
DEPRESS HALT
PRESS & RELEASE
START
ARE ALL
LIGHTS OFF?
NO
CABL
(10)
YES
SR- 177777
PRESS & RELEASE
LOAD ADRS
DO
LIGHTS SHOW
177777?
YES
PROC
(11)
NO
REMOVE
CONSOLE CABLE
FROM M7260
(9,10)
CABL
DO
DATA LIGHTS
SHOW
177777
YES
NO
WORK CABLES
AT BOTH ENDS
YES
STILL FAIL?
NO
YES
ARE
LIGHTS
DEFECTIVE?
NO
START
REPLACE
DEFECTIVE
LIGHTS
(1)
PROC
(11)
START
(1)


(1,7,9,10)
PROC
PROCA
REPLACE ALL PROCESSOR MODULES ONE AT A TIME WHILE RETRYING FAILING PROCEDURE AFTER EACH REPLACEMENT
SAME FAILURE?
NO
START
(1)
YES
REPLACE ORIGINAL PROCESSOR MODULES
IS CONSOLE OPERATIONAL ?
YES
NO
IS CPU AN 11/05?
NO
YES
CONSB
HAVE YOU CHECKED THE POWER YET?
YES
NO
MEMA
(12)
PWRB
(5)
PERFORM 11/05 CONSOLE CHECKOUT PROCEDURE
DOES IT CHECKOUT OK?
NO
YES
REPLACE DEFECTIVE COMPONENT OR ENTIRE CONSOLE
START
(1)
CHECK CONTROL SWITCHES
MEMA
(12)
YES
SWITCHES OK?
NO

TROUBLESHOOTING FLOWCHART 12




(2,7,12)
PERIF
BREAK BUS AND TERMINATE AT LAST BANK OF MEMORY
DOES EXERCISER FAIL?
YES
MAINF
(7)
NO
ADD NEXT DEVICE ONTO BUS
DOES EXERCISER FAIL?
YES
REPAIR LAST DEVICE ADDED
START
(1)
NO
NO
HAVE ALL DEVICES BEEN ADDED?
YES
START
(1)

LOCATION OF AC LO/DC LO SIGNALS

Signal	11/05	11/40*	11/45*
AC LO DC LO	M7261 B08F1 B08F2 (Very difficult to scope. Check on tab connector)	M7235 B06D2 B06C1	M8106 C12S1 C12U1

Quiescent levels should be +4.8 V ± 0.2 V

AC LO and DC LO are generated in the above CP modules, the bulk power supplies, and the power supplies of all peripherals. These are all connected to the Unibus in a wired OR. The processor receives them and makes itself and its memories totally inoperative if either one is true. It, therefore, will be necessary to physically disconnect the system in a logical fashion to determine which device is faulty.

* The following displays will be seen on the front panel if DC LO is true:

PDP-11/45 – The CPU's microaddress will be 200.

PDP-11/40 – The CONS and RUN lights will be off and the BUS and PROC lights will be on.

QUIESCENT UNIBUS LEVELS

Normal bus quiescent levels are listed below. Any level that deviates from normal should be looked upon as a potential failure. In most cases, this improper level will be caused by a defective bus receiver or driver. AC LO and DC LO are also power-supply dependent.

Signal	Quiescent Level
BG(7:4), NPG, BBSY (on PDP-11/40)	+0.4 V ± 0.4 V
AC LO, DC LO	+4.8 V ± 0.2 V
All others	+3.4 V ± 0.2 V

Procedure

To measure quiescent Unibus levels:

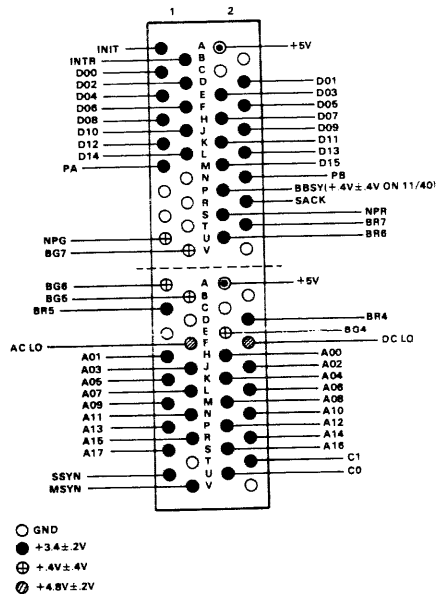
1. Turn the system on with the processor halted. Press the START key and release it with HALT down. (On older PDP-11/40s it is necessary to LOAD ADRS 000000 to clear the A<17:00> lines.)

2. Use a *calibrated* oscilloscope to measure the Unibus signal lines. (See the following chart of Unibus slot backplane pins.)

To obtain meaningful readings of bus grant lines (BG<7:4> and NPG), they should be measured at *each* device wired to them down the length of the bus.

All buses should be checked in multiple bus systems.

UNIBUS SLOT
(Viewed from the Backplane)



QUIESCENT CONSOLE LIGHT DISPLAY

Press and release the START key with the HALT key down.

PDP-11/40

ADRS = 000000
DATA = 000000
RUN = ON
BUS = ON
PROC = ON
CONSOLE = ON
All the rest = OFF

PDP-11/45

All ADRS = 000000
DISPLAY REG = 000000
BUS REG = 000000
DATA PATHS = Contents of R0
μADRS CPU = 170 (200 when START is held down)
KERNEL = ON (OFF when START is held down)
MASTER = ON
All the rest = OFF

NOTE

Address display may NOT be all zeros, depending on the FCO level of the CPU.

PDP-11/05 CONSOLE CHECKOUT PROCEDURE

1. Make certain the computer power is off.
2. Disconnect the console cable connector from the M7260 module and then turn on the computer power.
3. After step 2 is completed, the data pattern 177777_8 should be displayed on the console lights.
4. At the connector that plugs into the M7260 module, use a piece of small gauge wire and jumper pin F to the following pins and observe if the displayed pattern is correct. Remove the jumper each time before proceeding to the next step.
5. Jumper: GND; Pattern: 000000
6. Jumper: N; Pattern: 125252
7. Jumper: L; Pattern: 146314
8. Jumper: J; Pattern: 170360
9. Jumper: D; Pattern: 177400
10. Remove the jumper after completing the procedure.

WHAT TO DO WHEN ALL ELSE FAILS

Most of the cases in the flowchart that take you here are for problem symptoms that are very unusual. Make sure you have checked every possibility. If you have a solid problem and you just don't know where to go, start at NO-OP on flow page 1.

PROCESSOR'S INTERNAL OPTIONS

PDP-11/05

None

PDP-11/40

KE11-E	Extended Instruction Set M7238, slot 2 Requires jumper changes; see DECAID PROCA40
KE11-F	Floating Instruction Set M7239, slot 1 Requires jumper changes; see DECAID PROCA40
KT11-D	Memory Management M7236, slot 8 Requires jumper changes; see DECAID PROCA40
KJ11-A	Stack Limit Register M7237, slot 3E Requires jumper changes; see DECAID PROCA40
KW11-L	Line Clock M787, slot 3F Requires wire change; see DECAID PROCA40

PDP-11/45

FP11	Floating Point Processor M8114, slot 2 M8115, slot 3 M8112, slot 4 M8113, slot 5 No jumper changes required.
KT11-C	Memory Management M8108, slot 13 M8107, slot 14 Requires removal of M8116 from slot 14
KW11-L	Line Clock M787, slot 1C Requires removal of wire between pin C01R2 and pin C01V2.

FOR SYSTEMS EQUIPPED WITH PAPER TAPE

NOTE

In the following descriptions, the value XX reflects the system's memory size as shown below:

Memory Size	XX
4K	01
8K	03
12K	05
16K	07
20K	11
24K	13
28K or greater	15

LOAD ADRS	XX7744
DEPOSIT	016701
	000026
	012702
	000352
	005211
	105711
	100376
	116162
	000002
	XX7400
	005267
	177756
	000765
	177560 (TTY)
	or 177550 (PC11)

DECAID LOADER

Place the Absolute Loader tape in the specified reader with the special bootstrap leader code (ASCII code*351) over the reader sensors.

LOAD ADRS XX7744
START

If the system contains a BM792-YA Paper Tape Bootstrap Loader:

LOAD ADRS 773000
START

The Absolute Loader tape will pass through the reader and will stop when the last frame of data has been loaded into memory.

Place the MAINDEC tape in the specified reader with blank leader tape over the reader sensors.

Press CONT or
LOAD ADRS XX7500
START

The MAINDEC tape will pass through the reader and will stop when completely loaded into memory.

At this time R0 which is displayed in the DATA display* will contain all zeros in the low byte. If not, a checksum error has occurred indicating that some data was incorrectly loaded. The tape should be reloaded.

FOR SYSTEMS EQUIPPED WITH DISK, DECtape, AND/OR MAGTAPE

Hardware Bootstrap Loaders

The following table describes the procedures for loading the MAINDEC monitors. Ensure that your XXDP medium is on drive 0.

* For PDP-11/05: to read R0, load address 177700 and press EXAM switch.

DECAID LOADER

ROM	Device	Procedure
BM792-YA	PC11/PC05	LOAD ADRS.....773000 START
BM792-YB	RK11/RK05	LOAD ADRS.....773100 SWITCH REG.....777406 START
	TC11/TU56	LOAD ADRS.....773100 SWITCH REG.....777344 START
BM792-YC	CR11	LOAD ADRS.....773200 START
MR11-DB (Consists of an M792-YD and an M792-YE)	RK11/RK05	LOAD ADRS.....773110 START
	TC11/TU56	LOAD ADRS.....773120 START
	TM11/TU10	LOAD ADRS.....773136 START
	RP11/RP03	LOAD ADRS.....773154 START
BM792-YH	TA11/TU60	LOAD ADRS.....773300 START
BM792-YL	RX11/RX01	LOAD ADRS.....773400 START
BM873-YA (See note)	RK11/RK05	LOAD ADRS.....773010 START
	TC11/TU56	LOAD ADRS.....773030 START
	TM11/TU10	LOAD ADRS.....773050 START

DECAID LOADER

ROM	Device	Procedure
	TA11/TU60	LOAD ADRS .....773230 START
	RP11/RP03	LOAD ADRS .....773100 START
BM873-YB (See note)	RK11/RK05 (unit zero)	LOAD ADRS .....773030 START
	RK11/RK05 (Unit specified in Switch register)	LOAD ADRS .....773032 START
	TC11/TU56	LOAD ADRS .....773070 START
	TM11/TU10	LOAD ADRS .....773110 START
	TJU16/TU16	LOAD ADRS .....773150 START
	TA11/TU60	LOAD ADRS .....773524 START
	RP11/RP03	LOAD ADRS .....773350 START
	RJP04/RP04	LOAD ADRS .....773350 START
	RJS03/RS03	LOAD ADRS .....773000 START

NOTE

M873 is slot sensitive. Use in slot 1 or connect AC LO from B01F1 to C__V1 and DC LO from B01F2 to C__N1. Also, M873 must be on the CPU side of any bus switch or bus repeater.

Software Bootstrap Loaders

If the hardware bootstrap doesn't work, try one of the following software bootstraps depending on your particular syystem configuration.

Device	Procedure
RK11/RK05	LOAD ADRS.....010000 DEPOSIT.....012737 000005 177404 000001 LOAD ADRS.....010000 START Wait 1 second. LOAD ADRS.....000000 START
TC11/TU56	LOAD ADRS.....777342 DEPOSIT.....004003 DECtape will rewind. EXAMINE DEPOSIT.....000001 LOAD ADRS.....000216 DEPOSIT.....012737 000005 177342 000777 LOAD ADRS.....000216 START

DECAID LOADER

Device	Procedure
TM11/TU10	LOAD ADRS.....010000
	DEPOSIT.....005137
	172524
	012737
	060011
	172522
	000777
	012737
	060003
	172522
	000777
	LOAD ADRS.....010000
	START
	Wait 1 second.
	Press HALT switch.
	LOAD ADRS.....010014
	START
	Wait 1 second.
	Press HALT switch.
	LOAD ADRS.....000000
	START

DECAID LOADER

Device	Procedure
TJU16/TU16	LOAD ADRS.....010000
	DEPOSIT.....012737
	001300
	172472
	012737
	177777
	172446
	012737
	000031
	172440
	105737
	172452
	100375
	012737
	177400
	172442
	005037
	172444
	042737
	000007
	172452
	012737
	000071
	172440
	105737
	172440
	000100
	000375
	000137
	000000
	LOAD ADRS.....010000
	START

DECAID LOADER

Device	Procedure	
TA11/TU60	LOAD ADRS	001000
	DEPOSIT	012700
		177500
		005010
		010701
		062701
		000052
		012702
		000375
		112103
		112110
		100413
		130310
		001776
		105202
		100772
		116012
		000002
		120337
		000000
		001767
		000000
		000755
		005710
		100774
		005007
		017640
		002415
		112024
	LOAD ADRS	001000
	START	

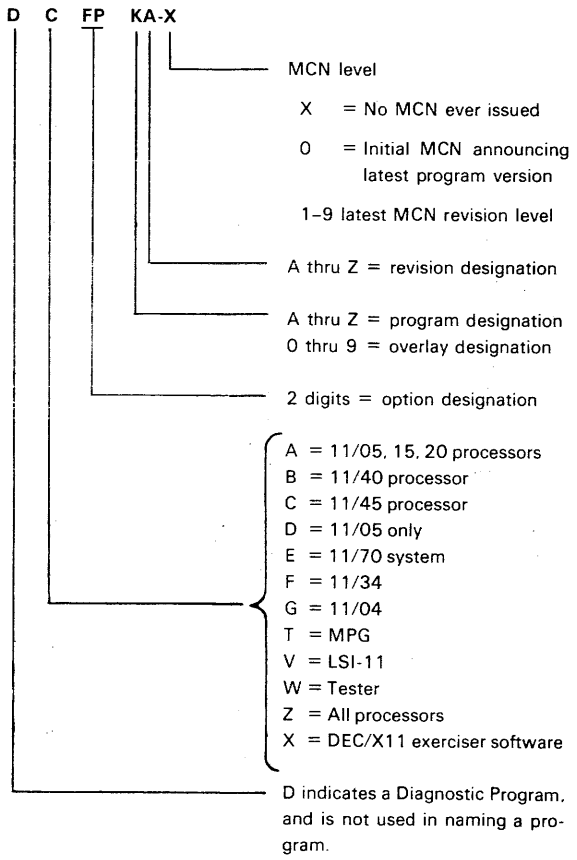
DECAID LOADER

Device	Procedure	
RX11/RX01	LOAD ADRS	001000
	DEPOSIT	005000
		012701
		177170
		105711
		001776
		012711
		000003
		005711
		001776
		100405
		105711
		100004
		116120
		000002
		000770
		000000
		005000
		000110
		000000
		000000
		000000
	LOAD ADRS	001000
	START	
RP11/RP03	LOAD ADRS	001000
	DEPOSIT	012705
		176716
		012715
		177400
		012745
		000005
		105715
		100376
		005007
	LOAD ADRS	001000
	START	

DECAID LOADER

Device	Procedure	
RJP04/RP04	LOAD ADRS	010000
	DEPOSIT	012700
		176700
		012710
		000023
		005060
		000034
		005060
		000006
		012760
		177400
		000002
		012710
		000071
		105710
		100376
		005007
	LOAD ADRS	010000
	START	
RJS04/RS04	LOAD ADRS	001000
RJS03/RS03	DEPOSIT	012705
		172044
		012745
		177400
		012745
		000071
		032715
		100200
		001775
		100762
		005007
	LOAD ADRS	001000
	START	

MAINDEC NAMING CONVENTION



MAINFRAME MAINDECS AND THE PROCESSORS THEY Run On

Name	Number	05	40	45
T17 4K, CPU, I/O	DZQKB	X	X	X
11 Family Instruction Exerciser	DZQKC	X	–	–
11/40 and 11/45 Instruction Exerciser	DCQKC	–	X	X
Multiplication–Division Exerciser	DCQKA	–	EIS	X
11/40 and 11/45 Processor Parity	DCKBR	–	PAR	PAR
11/45 Console Test	DCKBQ	–	–	X
States 11/45	DCKBO	–	–	X
PIRQ	DCKBN	–	–	X
Traps 11/45	DCKBM	–	–	X
T14 11/05, 20 Traps	D0NC	X	–	–
T14 11/40 Traps	DBKDM	–	X	–
Power Fail 11/45	DCKBP	–	–	X
Power Fail 11/05, 20, 40	DZKAQ	X	X	–
DIV	DCKBL	–	EIS	X
MUL	DCKBK	–	EIS	X
ASHC	DCKBJ	–	EIS	X
ASH	DCKBI	–	EIS	X
11/45 Registers	DCKBH	–	–	X
SPL	DCKBG	–	–	X
Stack Limit	DCKBF	–	KJ	X
RTT	DCKBE	–	X	X
MARK	DCKBD	–	X	X
XOR	DCKBC	–	X	X
SOB	DCKBB	–	X	X
SXT	DCKBA	–	X	X

BASIC MAINFRAME MAINDECs (Run On All Processors)

Name	Number
T13 JSR, RTS, RTI	D0MA
T12 Jump	D0LA
T11 Subtract	D0KA
T10 Add	D0JA
T9 Bit Set Clear Test	D0IA
T8 Move	D0HA
T7 CMP Non-Equality	D0GA
T6 CMP Equality	D0FA
T5 Rotate/Shift	D0EA
T4 Binary	D0DA
T3 Unary	D0CA
T2 Con Branch	D0BA
T1 Branch	D0AA

MEMORY MAINDECS AND THE PROCESSORS THEY RUN ON

Name	Number	05	40	45
0–124K Memory Exerciser	DZQMB	X	X	X
Memory I/O Exerciser	DZQMA	X	X	X
Up-Down Address Test	DZMMK	X	X	X
8K Special	DZMMJ	X	X	X
Random Data	DZMMI	X	X	X
Core Heating	DZMMH	X	X	X
Worst Case Noise	DZMMG	X	X	X
Ones Susceptibility	DZMMF	X	X	X
Moving Ones and Zeros	DZMME	X	X	X
Basic Test Patterns	DZMMD	X	X	X
No Dual Address Test	DZMMC	X	X	X
Basic Address Test Down	D1BA	X	X	X
Basic Address Test Up	D1AA	X	X	X
MA11, MF11, and MS11 Parity Test	DCMFA	–	PAR	PAR
MS11	DCMSB	–	–	MOS BIP

INTERNAL OPTION MAINDECS

PDP-11/45 Floating Point – FP11

Name	Number
GTP Overlay	DCQOA
Basic Instruction Exerciser	DCFPO
Division Exerciser	DCFPU
Multiplication Exerciser	DCFPT
Addition and Subtraction Exerciser	DCFPS
LDD/STD Exerciser	DCFPR
Maintenance	DCFPM
MODF, MODD	DCFPL
LDEXP, STEXP	DCFPK
LDCJX, STCXJ	DCFPJ
LDCDF, LDCFD, STCDF, STCFD	DCFPI
CLR, TST, ABS, NEG	DCFPH
DIVF, DIVD	DCFPG
MULF, MULD	DCFPF
CMPF, CMPD	DCFPE
ADDF, ADDD, SUBF, SUBD	DCFPD
LDF, LDD, STF, STD	DCFPC
STS, Illegal Instructions	DCFPB
Basic Test	DCFPA

PDP-11/45 Memory Management – KT11-C

Name	Number
Exerciser	DCKTG
Abort	DCKTF
MFPD/I	DCKTE
MTPD/I	DCKTD
Keys	DCKTC
Logic 2	DCKTB
Logic 1	DCKTA

PDP-11/40 Floating Point – KE11-F

Name	Number
GTP Overlay	DBKEO
Exerciser	DBKEB
Basic Instruction Test	DBKEA

PDP-11/40 Memory Management – KT11-D

Name	Number
Exerciser	DBKTG
Abort	DBKTF
States	DBKTD
MFPI/MTPI	DBKTC
Keys	DBKTB
Logic	DBKTA

Line Clock – KW11-L

Name	Number
Exerciser	DZKWA

PDP-11 SYSTEMS TESTS

1.	DEC/X11	*	2 hours or 2 passes
2.	DZQGA	GTP (for non-communication-oriented systems)* or	2 hours or 2 passes
	DZQCA	CTP (for communication-oriented systems)*	2 hours or 2 passes

* Mechanical devices, e.g., TTY, line printers, should only be run 5 minutes.

PDP-11/05/10 CUSTOMER ACCEPTANCE DIAGNOSTICS

Central Processor

1.	DZQKC	11 Family Instruction Exerciser, any switch 0–7 up	30 minutes
2.	DONC	T14 11/05, 20 traps	2 passes
3.	DZQKB	T17 Systems Test, only console device enabled	10 minutes
4.	DZKAQ	Power Fail	10 power fails

Read/Write Memories

1.	DZQMB	0–124K Memory Exerciser, with switch 11 up	1 minute/8K
2.	DZQMB	0–124K Memory Exerciser, 8K or greater	2 minutes/8K
3.	DZMMI	Random Data, for less than 8K only	5 minutes

PDP-11/35/40 CUSTOMER ACCEPTANCE DIAGNOSTICS

Central Processor

1.	DCQKC	11/40 and 11/45 Instruction Exerciser, any switch 0-7 up	30 minutes
2.	DBKDM	T14 11/40 Traps	2 passes
3.	DZQKB	T17 Systems Test, only console device enabled	10 minutes
4.	DZKAQ	Power Fail	10 power fails

Read/Write Memories

1.	DZQMB	0-124K Memory Exerciser, with switch 11 up	1 minute/8K
2.	DZQMB	0-124K Memory Exerciser, 8K or greater	2 minutes/8K
3.	DCMFA	Parity Test	2 passes

Stack Limit Option - KJ11-A

1.	DCKBF	Stack Limit Option Test	2 minutes

Extended Instruction Set - KE11-E

1.	DCKBI-DCKBL	ASH, ASHC, MUL, DIV Test	2 minutes each
2.	DCQKA	MUL, DIV Random Exerciser	20 minutes

Floating Instruction Set – KE11-F

1. DBKEB	KE11-F Exerciser	20 minutes
2. DBKEO	GTP Overlay	20 minutes

Memory Management – KT11-D

1. DBKTG	KT11-D Exerciser	4 minutes/8K

PDP-11/45/50 CUSTOMER ACCEPTANCE DIAGNOSTICS

Central Processor

1. DCQKC	11/40 and 11/45 Instruction Exerciser	30 minutes
2. DZQKB	T17 Systems Test, only console device enabled	10 minutes
3. DCKBP	Power Fail	1 pass

Read/Write Memories

1. DZQMB	0–124K Memory Exerciser, with switch 11 up	1 minute/8K
2. DZQMB	0–124K Memory Exerciser, 8K or greater	2 minutes/8K
3. DCMFA	Parity Test	2 passes
4. DCMSB	MOS Memory Test	2 passes

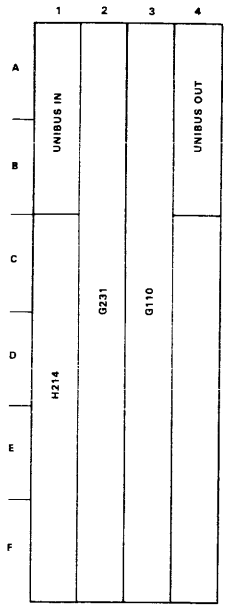
Floating Point Processor – FP11

1. DZFPO	FP11 Exerciser	20 minutes
2. DCQOA	GTP Overlay, all available devices enabled except line printer if any	20 minutes

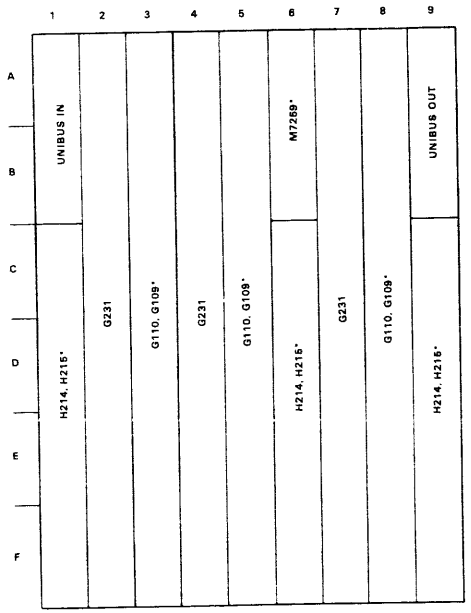
Memory Management – KT11-C

1. DCKTG	KT11-C Exerciser, all available devices enabled except line printer if any	1 pass

MM11-S MODULE UTILIZATION



MF11-L(P) MODULE UTILIZATION



* IF PARITY MEMORY.

G110 CONTROL MODULE JUMPERS

(Used on MM11-L type memories)

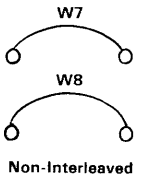
See the next page for a diagram showing the physical location of the jumpers.

Memory Bank (words)	Machine Address (words)	Device Address Jumpers*			
		W6 A14 or A01†	W4 A15	W3 A16	W2 A17L
0-8K	000000-037776	In	In	In	In
8-16K	040000-077776	Out	In	In	In
16-24K	100000-137776	In	Out	In	In
24-32K	140000-177776	Out	Out	In	In
32-40K	200000-237776	In	In	Out	In
40-48K	240000-277776	Out	In	Out	In
48-56K	300000-337776	In	Out	Out	In
56-64K	340000-377776	Out	Out	Out	In
64-72K	400000-437776	In	In	In	Out
72-80K	440000-477776	Out	In	In	Out
80-88K	500000-537776	In	Out	In	Out
88-96K	540000-577776	Out	Out	In	Out
96-104K	600000-637776	In	In	Out	Out
104-112K	640000-677776	Out	In	Out	Out
112-120K	700000-737776	In	Out	Out	Out
120-128K	740000-777776	Out	Out	Out	Out

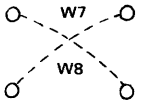
* W5 and W10 must be installed and W9 must be removed (for 8K stacks).

† The memory can be interleaved as 16K only, using two adjacent contiguously addressed 8K banks. When two 8K banks are interleaved, jumpers W7 and W8 must be in the configuration shown by the dotted lines. Bit A01 goes to the device selector gate controlled by jumper W6. One 8K banks must have W6 installed and the other must have W6 removed.

When not interleaved, jumpers W7 and W8 must be in the configuration shown by the solid lines. Bit A14 goes to the device selector gate controlled by jumper W6.



Non-Interleaved

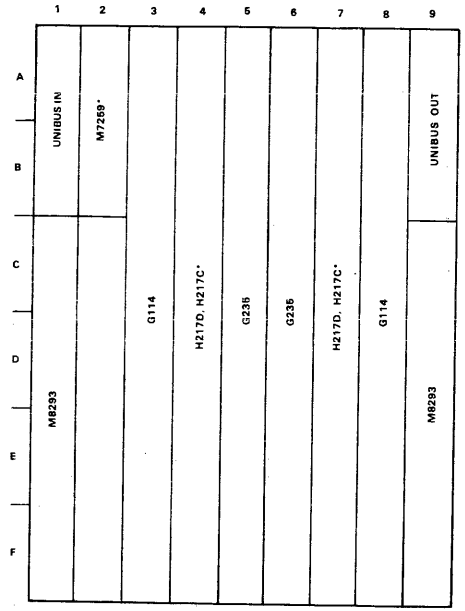


Interleaved

G110 CONTROL MODULE
(Used on MM11-L type memories)

NOTES:
1. Jumper W1 is for test purposes only. It must be installed for normal operation.
2. Jumper W11 should be removed for normal operation. When installed the memory responds to DATI only, regardless of state of control lines COO and CO1.
3. Jumpers W7 and W8 must remain in the factory installed positions.
4. When used as an 8k bank, jumpers W5 and W10 must be installed and jumper W9 must be removed.
5. When used as a 4k bank, jumper W10 must be removed and jumper W9 must be installed. Jumper W5 determines the location of the bank on the bus.

MF11-U(P) MODULE UTILIZATION



* IF PARITY MEMORY.

M8293 DEVICE ADDRESS JUMPERS
(Used on MM11-U type memories)

Memory Bank (Words)	Machine Address (Words)8	W7 A17	W6 A16	W5 A15	W4 A14	W3 A13
0-16K	000000-077776	IN	IN	IN	IN	IN
4-20K	020000-117776	IN	IN	IN	IN	OUT
8-24K	040000-137776	IN	IN	IN	OUT	IN
12-28K	060000-157776	IN	IN	IN	OUT	OUT
16-32K	100000-177776	IN	IN	OUT	IN	IN
20-36K	120000-217776	IN	IN	OUT	IN	OUT
24-40K	140000-237776	IN	IN	OUT	OUT	IN
28-44K	160000-257776	IN	IN	OUT	OUT	OUT
32-48K	200000-277776	IN	OUT	IN	IN	IN
36-52K	220000-317776	IN	OUT	IN	IN	OUT
40-56K	240000-337776	IN	OUT	IN	OUT	IN
44-60K	260000-357776	IN	OUT	IN	OUT	OUT
48-64K	300000-377776	IN	OUT	OUT	IN	IN
52-68K	320000-417776	IN	OUT	OUT	IN	OUT
56-72K	340000-437776	IN	OUT	OUT	OUT	IN
60-76K	360000-457776	IN	OUT	OUT	OUT	OUT
64-80K	400000-477776	OUT	IN	IN	IN	IN
68-84K	420000-517776	OUT	IN	IN	IN	OUT
72-88K	440000-537776	OUT	IN	IN	OUT	IN
76-92K	460000-557776	OUT	IN	IN	OUT	OUT
80-96K	500000-577776	OUT	IN	OUT	IN	IN
84-100K	520000-617776	OUT	IN	OUT	IN	OUT
88-104K	540000-637776	OUT	IN	OUT	OUT	IN
92-108K	560000-657776	OUT	IN	OUT	OUT	OUT
96-112K	600000-677776	OUT	OUT	IN	IN	IN
100-116K	620000-717776	OUT	OUT	IN	IN	OUT
104-120K	640000-737776	OUT	OUT	IN	OUT	IN
108-124K	660000-757776	OUT	OUT	IN	OUT	OUT
112-124K	700000-757776	OUT	OUT	OUT	IN	IN
116-124K	720000-757776	OUT	OUT	OUT	IN	OUT
120-124K	740000-757776	OUT	OUT	OUT	OUT	IN

NOTE: The memory may be interleaved in 32K increments, using two contiguously addressed 16K banks.

M8293 DEVICE ADDRESS JUMPERS
(Used on MM11-U type memories)

In the 16K interleaved memory configuration the following changes must be made:

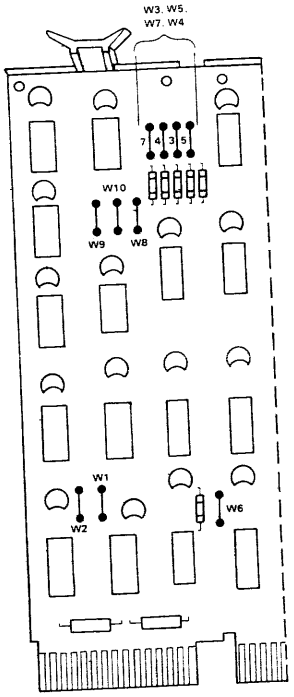
W1 is cut.
W2 is in.
W8 is in.
W9 is in. } For one 16K of the interleaved pair
W10 is cut. }
W9 is cut. } For the other 16K of the interleaved pair
W10 is in. }

Both interleaved memories should be cut for the same starting address.

Non-Interleaved

W1 is in.
W2. W8. W9. and W10 are cut.

M8293 16K UNIBUS TIMING MODULE
(Used on MM11-U type memories)

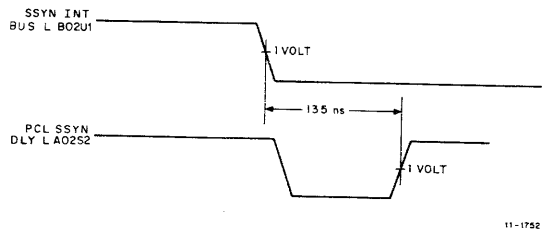


NOTE
W1, W2, W8, W9, AND W10 ARE SELECTED FOR INTERLEAVED OR NON-INTERLEAVED OPERATION. W3, W4, W5, W6, AND W7 ARE SELECTED FOR STARTING ADDRESS

PCL SSYN DLY L ADJUSTMENT PROCEDURE FOR M7269 PARITY CONTROLLER

This adjustment sets a 105-ns delay from the leading edge of SSYN INT BUS L to allow sufficient settling time for the parity checking logic. Too long a delay will result in increased cycle time and access time. The waveshapes should be as shown below. Adjust R16 to obtain the required delay.

SSYN DLY L TIMING RELATIONSHIP



CSR ADDRESS JUMPER SELECTION FOR M7259 PARITY CONTROLLER MODULE,
Etch Rev D, CS Rev E

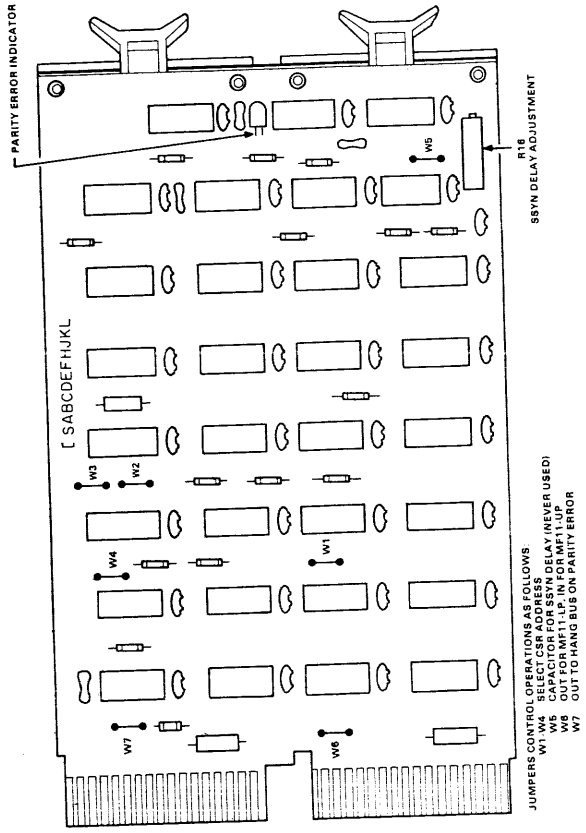
	CSR Jumpers	W4	W3	W2	W1
	Bus Address Line	A04	A03	A02	A01
Lower Memory Boundary	CSR Address				
0K	772100	X	X	X	X
4K	772102	X	X	X	O
8K	772102	X	X	X	O
12K	772104	X	X	O	X
16K	772104	X	X	O	X
20K	772106	X	X	O	O
24K	772106	X	X	O	O
28K	772110	X	O	X	O
32K	772110	X	O	X	X
36K	772112	X	O	X	O
40K	772112	X	O	X	O
44K	772114	X	O	O	X
48K	772114	X	O	O	X
52K	772116	X	O	O	O
56K	772116	X	O	O	O
60K	772120	O	X	X	X
64K	772120	O	X	X	X
68K	772122	O	X	X	O
72K	772122	O	X	X	O
76K	772124	O	X	O	X
80K	772124	O	X	O	X
84K	772126	O	X	O	O

(Continued)

	CSR Jumpers	W4	W3	W2	W1
	Bus Address Line	A04	A03	A02	A01
Lower Memory Boundary	CSR Address				
88K	772126	O	X	O	O
92K	772130	O	O	X	X
96K	772130	O	O	X	X
100K	772132	O	O	X	O
104K	772132	O	O	X	O
108K	772134	O	O	O	X
112K	772134	O	O	O	X
116K	772136	O	O	O	O
120K	772136	O	O	O	O

O = Jumper Removed
X = Jumper Installed
W6: Removed for MF11-LP, installed for MF11-UP

M7259 PARITY CONTROL MODULE
(Etch Rev. D)



STROBE ADJUSTMENT PROCEDURE FOR MEMORIES USING THE G110

When any module in the memory system is replaced, the strobe must be adjusted for that system.

To adjust the strobe, use DZQMBC or a higher revision of the 0-124K Memory Exerciser MAINDEC.

Start the diagnostic at ADRS 214, allowing you to enter test parameters. Cycle on the bank to be adjusted while running pattern 1. Check program procedures for further information.

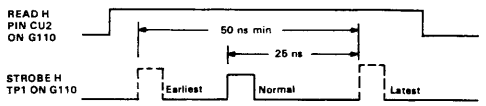
Attach and sync channel 1 to READ H pin CU2 of the G110 and channel 2 to STROBE H at test point 1 on the G110. (See the next page for the location of TP1.)

Adjust the strobe to the earliest (minimum) setting at which the memory will cycle error free and allow the test to run for three minutes.

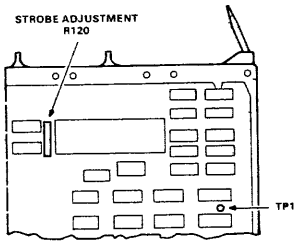
Adjust the strobe to the latest (maximum) setting at which the memory will cycle error free; this adjustment is most critical and there should not be more than 5 ns difference between the failing and passing points on this adjustment. Allow the test to run at this setting for three minutes or more.

This window should be 50 ns minimum. If it is not, replace modules within that bank of memory until it is.

Adjust STROBE H to be 25 ns (leading edge to leading edge) from the latest (maximum) setting.



G110 CONTROL MODULE



PDP-11/05 MAINTENANCE

The PDP-11/05 has the ability to run a program in its General Purpose registers. Therefore, disconnect all Unibus devices and all memory from the processor and if the console switches operate properly, attempt the program shown below.

General Purpose Register (GPR) Program

LOAD ADRS.....177700
DEPOSIT.....000240
000777
LOAD ADRS.....177700
START
The program is now being executed.
The run light should come on.
Press the HALT switch.
The ADDRESS/DATA display should contain either 177700 or 177701.

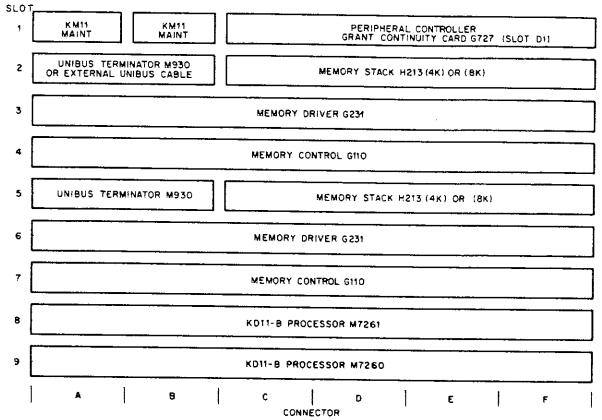
If a processor problem is indicated, replace the M7260 and M7261 modules, and attempt the failing procedure after each replacement.

Always replace *one* module at a time to ensure that you don't plug a module into the wrong slot. If you are not absolutely positive, check the module utilization diagrams in this DECAID.

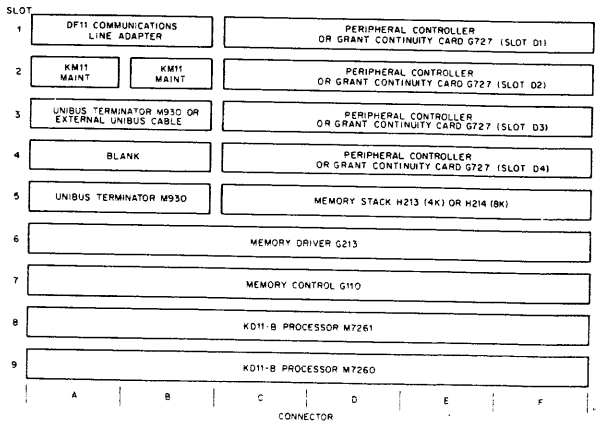
After replacing the M7260, it may be necessary to adjust the SCL clock as shown in this DECAID.

Remember, when you replace the M7261 module, check and if necessary, adjust the processor's clock rate as outlined in this DECAID.

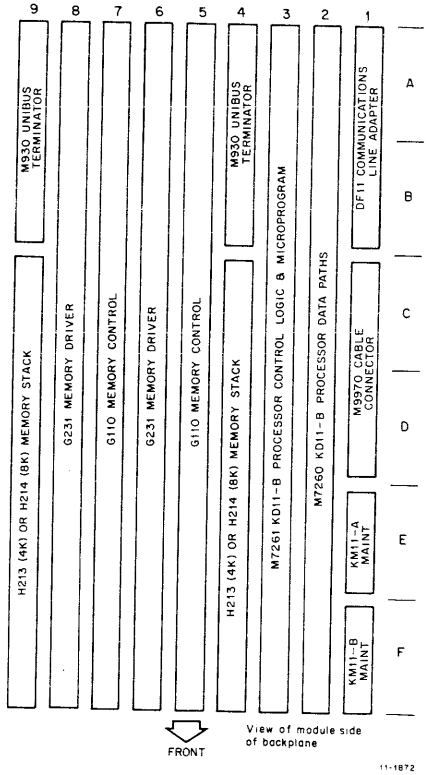
PDP-11/05-JA MODULE UTILIZATION (16K)



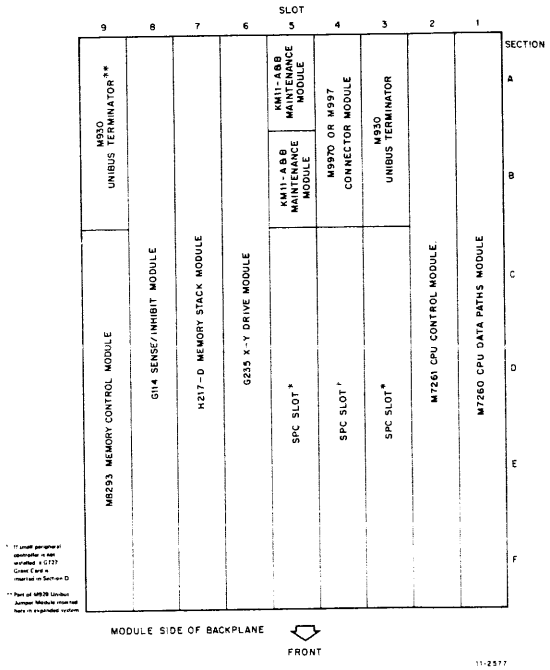
PDP-11/05-LA MODULE UTILIZATION (8K)



PDP-11/05-N MODULE UTILIZATION
(10-1/2-Inch Mounting Box)

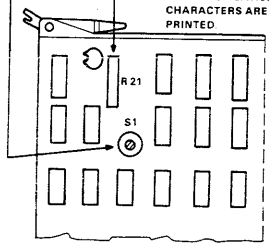


PDP-11/05/10-S MODULE UTILIZATION
(10-1/2-Inch Mounting Box)



M7260 DATA PATHS MODULE
(Etch Rev. C)

NOTES:
1. ARROW ON S1 MUST POINT TO:
   5 FOR 110 BAUD, 150 BAUD
   4 FOR 300 BAUD
2. RUN A PROGRAM TO PRINT A CONTINUOUS STREAM OF CHARACTERS. ADJUST R21 TO THE CENTRE OF THE RANGE FOR WHICH SATISFACTORY CHARACTERS ARE PRINTED.



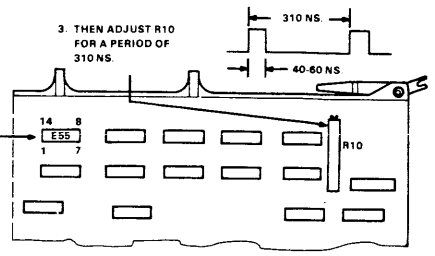
PROGRAM TO PRINT A STREAM OF CHARACTERS

500/005000	TOGGLE PROGRAM INTO CORE
502/105737	LOAD ADRS: 500
504/177564	START
506/100375	PROGRAM WILL LOOP
510/110037	CONTINUOUSLY UNTIL
512/177566	HALTED WITH HALT SWITCH
514/105200	
516/000771	

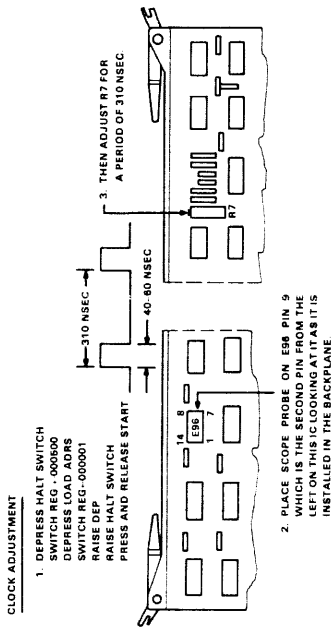
M7261 CONTROL LOGIC AND MICROPROGRAM MODULE
(Etch Rev. F)

CLOCK ADJUSTMENT

1. DEPRESS HALT SWITCH
   SWITCH REG ← 000500
   DEPRESS LOAD ADRS
   SWITCH REG ← 000001
   RAISE DEP
   RAISE HALT SWITCH
   PRESS & RELEASE START
2. PLACE SCOPE PROBE ON E55 PIN 8 WHICH IS THE FIRST PIN ON THE LEFT ON THIS IC LOOKING AT THE BOARD AS IT IS INSTALLED IN THE BACKPLANE
3. THEN ADJUST R10 FOR A PERIOD OF 310 NS.



M7261 CONTROL LOGIC AND MICROPROGRAM MODULE
(Etch Rev. E)



PDP-11/40 MAINTENANCE

The PDP-11/40 processor is comprised of the following modules:

M7231	Slot 4
M7232	Slot 3
M7233	Slot 5
M7234	Slot 7
M7235	Slot 6

Always replace *one* module at a time to ensure that you don't plug a module into the wrong slot. If you are not absolutely positive where a module should go, check this DECAID for the module location.

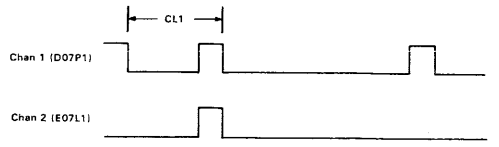
Be very careful that the jumpers on the boards are cut and inserted correctly for the particular system you are working on. See this DECAID for a list of jumpers, their locations, and configurations.

PDP-11/40 Processor Clock Adjustment Procedure

When replacing the M7234 module, check and if necessary, adjust the processor's clock rate according to the following procedure.

Clock adjustment requires a dual-trace oscilloscope with channel 1 (triggering) on D07P1 and channel 2 on E07L1. The computer should be in the CONSOLE mode waiting for a switch activation. This may be forced by a START in the HALT mode. Make sure that both the RUN and CONSOLE lights are on. The S1 switch (see the M7234 module diagram in this DECAID for its location) is adjusted (in 10-ns increments) until the CL1 interval shown below is:

$\geqslant 140$ ns $<$ 150 ns.



NOTE

As you are powering the machine on and off, one of the regulators may crowbar. You should suspect this if all the symptoms change for no apparent reason. If this should happen, determine which regulator has crowbarred, then turn its adjustment potentiometer a half revolution counterclockwise, and power the machine down and up. Then, with a scope, readjust the regulator so that it is producing the correct output. Remember that the machine must be powered down and up to turn the crowbar circuit off. Just turning the potentiometer down is not enough.

PDP-11/40 PROCESSOR MODULE'S JUMPERS

Option/ Module	KE11-E	KE11-F	KJ11	KT11-D	KW11-L	Parity Memory
M7231			Move: W2* (L→U)	Move: W10 (U→L) Cut: W1, W2, W3, W4, W5, W6, W7, W8, W9		
M7232						
M7233	Cut: W1†					
M7234			Move: W1 (U→L)	Move: W2 (L→U) Insert: C113, 680 pF C114, 560 pF		Cut: W5
M7235‡			Move: W1 (U→L)			Cut: W8
M7238		Cut: W1, W2, W3				
Backplane					Remove: Wire between pins F03R2 and F03V2	

Note: U = upper position and L = lower position between three split pins when looking at the module with the handles at the top. That is:

Upper position

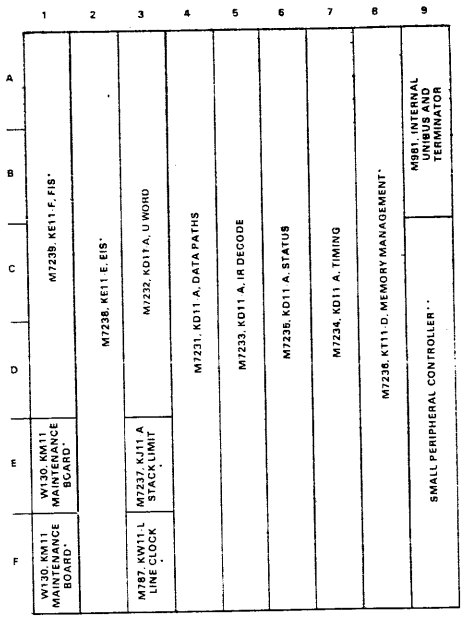
Lower Position

* If the KT11-D is installed along with the KJ11, cut the jumper out completely.

† Also install the three short cables between the M7232 and the M7238 modules.

‡ Connections for W7 through W2 are shown on the module for basic power-up vector address of 24_8. On the moudle's etch, the W numbers are located under these original jumper positions.

PDP-11/40 (KD11-A) MODULE UTILIZATION

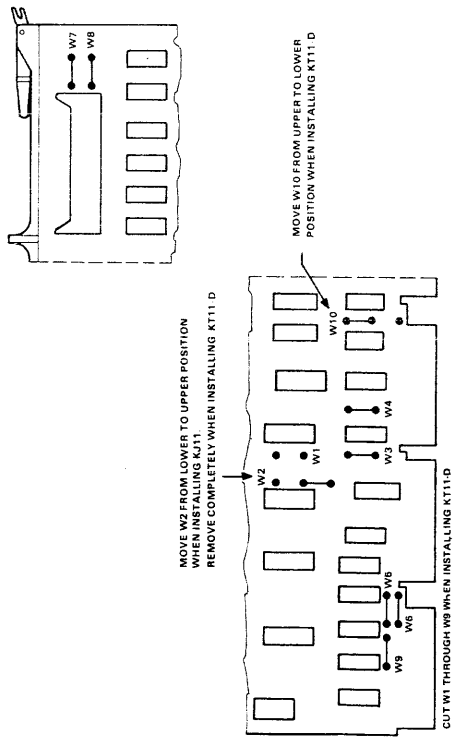


NOTES:

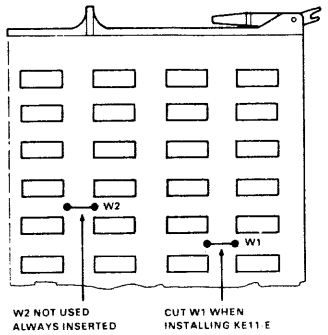
* PREWIRED MODULE SLOTS FOR NOTED OPTIONS.

** IF NO OPTION IS PRESENT IN THE SMALL PERIPHERAL CONTROLLER SLOT, A G727 GRANT CONTINUITY MODULE MUST BE INSERTED INTO ROW D.

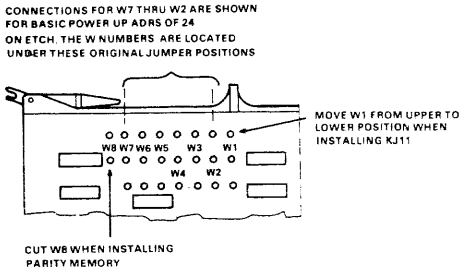
M7231 DATA PATHS MODULE
(Etch Rev. D)



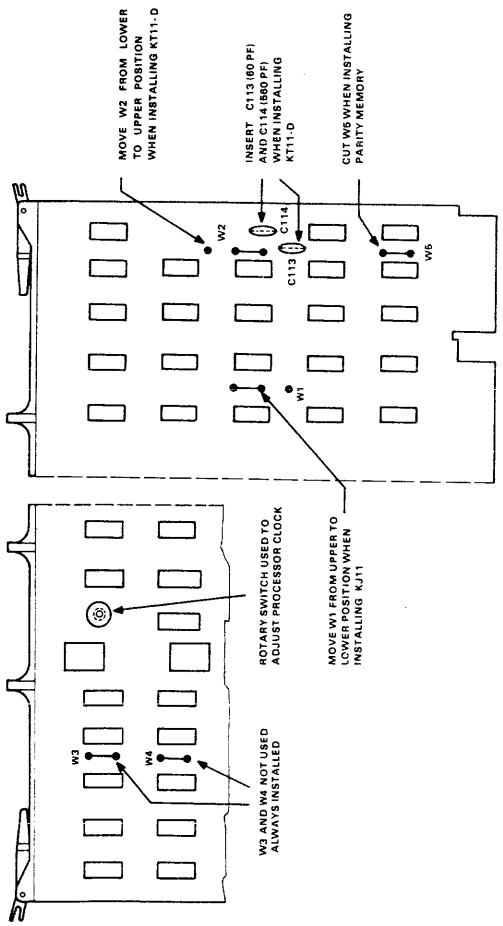
M7233 IR DECODE MODULE
(Etch Rev. D)



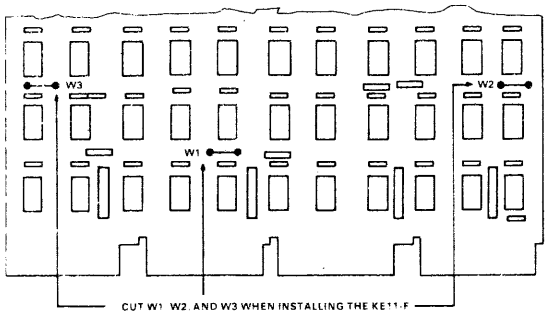
M7235 STATUS MODULE
(Etch Rev. D)



M7234 TIMING MODULE
(Etch Rev. D)



M7238 EIS MODULE
(Etch Rev. D)



PDP-11/45 MAINTENANCE

The PDP-11/45 processor is comprised of the following modules:

M8100	Slot 6
M8101	Slot 7
M8102	Slot 8
M8103	Slot 9
M8104	Slot 10
M8105	Slot 11
M8106	Slot 12
M8116	Slot 14 (if no Memory Management)
M8109	Slot 15

Always replace *one* module at a time to ensure that you don't plug a module into the wrong slot. If you are not absolutely positive where a module should go, check this DECAID for the module location.

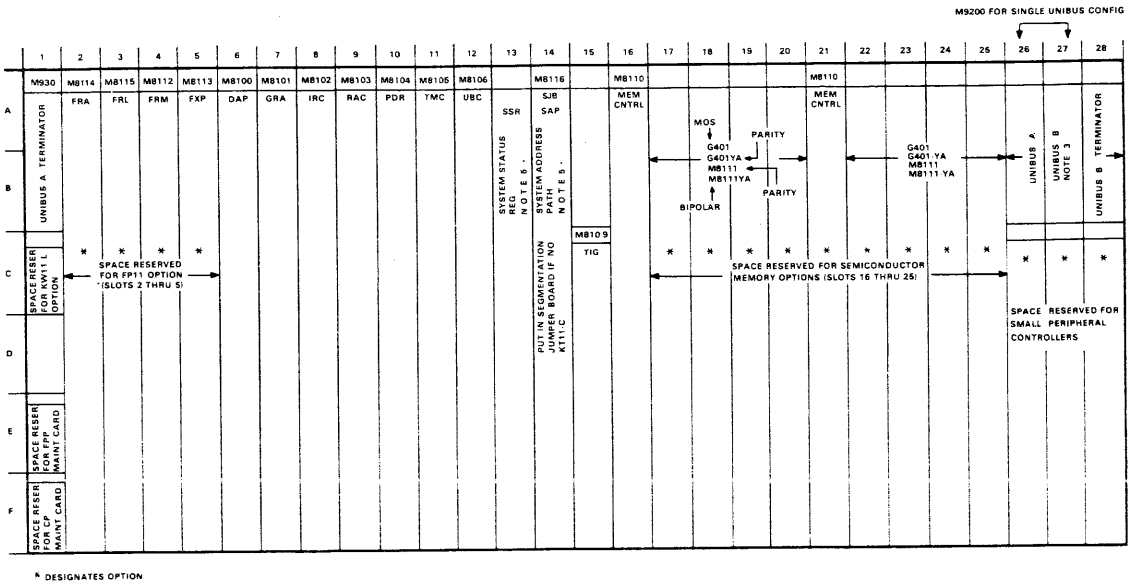
The PDP-11/45 has two 860 Power Controllers (one switched and one unswitched) or an 861 Power Controller with switched and unswitched outputs. Therefore, before you replace any modules, be sure to turn the power off at the power controller(s)' circuit breaker(s) not just at the front panel key. You might accidentally be changing a module that still has power applied to it.

When replacing the M8100, copy W1 through W6 from the old module onto the new module. See this DECAID for their location.

NOTE

As you are powering the machine on and off, one of the reulators may crowbar. You should suspect this if all the symptoms change for no apparent reason. If this should happen, determine which regulator has crowbarred, then rotate its adjustment potentiometer a half turn counter-clockwise and power the machine down and up. Then with a scope, readjust the regulator so that it is producing the correct output. Remember, the machine must be powered down to turn the crowbar off. Just turning the potentiometer down is not enough.

PDP-11/45 MODULE UTILIZATION



* DESIGNATES OPTION

NOTES:

1. Power to MOS and BIPOLAR memory remains on with the console on/off switch in the off position. This is indicated by the LED on the MOS Control Board.

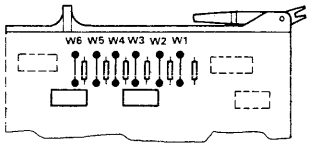
2. Caution must be observed when installing the boards into the backplane because of non-standard voltages present in slots 1, 2 and 15 thru 28.

3. If only one Unibus is used:
   The Unibus plugs into slot 28, plug an internal bus connector jumper (M9200) into slots 26 and 27, and plug a bus terminator (M930) into slot 1.
   If two Unibuses are used:
   Unibus A plugs into slot 26, terminator for Unibus A plugs into slot 1, Unibus B plugs into slot 27, and terminator for Unibus B plugs into slot 28.

4. Modules in slots 17-20 must be same type, modules in slots 22-25 must be same type.

5. If KT11 option is present, M8108 goes in slot 13 & M8107 goes in slot 14. If no KT11, M8116 goes in slot 14 and slot 13 is empty.

M8100 DAP MODULE
(Etch Rev. C)



W1 THRU W6 DETERMINE THE POWER UP VECTOR ADDRESS (USUALLY 24).
COPY FROM ORIGINAL MODULE.

CPU MODULES THAT DRIVE AND RECEIVE UNIBUS SIGNALS

CPU	Module	Unibus Signal
PDP-11/05	M7260	D<15:00>
	M7261	All the rest
PDP-11/40	M7231	A<17:00>, D<15:00>
	M7235	AC LO, DC LO, INIT, D<07:00>
	M7234	All the rest
PDP-11/45	M8104	D<15:00>
	M8116	A<17:00>
	M8105	BR<7:4>
	M8106	All the rest

BACKPLANE DC VOLTAGE CHECKS

With a calibrated scope, check the voltage levels and ripple on the backplane at the points indicated in the following tables. Notice that the tables indicate which regulator is providing the power being measured so that you can easily locate the correct one to adjust. (See DECAID PWRB-2 for the regulators' physical locations.)

Use a good scope probe with as short a ground lead as possible. Use "flag" type probe tips on both the scope probe and the ground lead and ground the probe on the backplane (not on the cabinet) as close as possible to the power point being measured.

CAUTION

Be extra careful that you put the ground lead on the correct backplane pin. Otherwise you may short something to ground which could cause considerable damage to the machine and your pride.

Only the voltage levels can be adjusted – not the ripple. If the magnitude of the ripple is greater than that specified, the offending regulator must be replaced.

PDP-11/40 CPU DC VOLTAGE CHECKS

Pin	Voltage (±5%)	Maximum Ripple (P-P Volts)	Regulator/ Slot
D09A2 (A01A2)	+5.0	0.20	H744/A
C09U1	+15.0	0.45	H742*
D09B2 (C09B2)	-15.0	0.45	H745/E

*No adjustment.

NOTE

If there is an MF11-U memory installed in the processor box, then all -15 V power must come from the H745 in slot D, since slot E contains the H754. Some configurations may have H754s in both slots D and E and then, of course, there cannot be anything installed in the box that requires -15 V, e.g., many small peripheral interfaces, MF11-L memories, etc.

PDP-11/45 CPU DC VOLTAGE CHECKS

Pin	Voltage	Maximum Ripple (P-P Volts)	Regulator/ Slot
A02A2	+5.0	0.20	H744/A (FPP)
F09V1	+5.0	0.20	H744/B (CPU)
F15V1	+5.0	0.20	H744/C (CPU)
F26A2	+5.0	0.20	H744/D (INT OPT)
E02B2	-15.0	0.45	H745/E (INT) (INT OPT and CPU)

(Continued)

Pin	Voltage	Maximum Ripple (P-P Volts)	Regulator/ Slot
A19A2	+5.0	0.20	H744/H (Bipolar)
A17V2	+23.2	0.70	H746/H (MOS)
A17U2	+19.7*	0.60	H746/H (MOS)
F17C1	-5.0	0.20	H746/H (MOS)
A16A2	+5.0	0.20	H744/J (MOS)
A21A2	+5.0	0.20	H744/K (Bipolar)
A24A2	+5.0	0.20	H744/L (Bipolar)
E15A1	+15.0† (13.5-16.5)	0.45	Top H742 (Switched)
E01B1	+8.0† (6.8-9.2)	0.24	Top H742 (Switched)
E16B2 E21B2	-15.0† (13.5-16.5)	0.45	Bottom H742 (Unswitched)

* 3 to 4 V less than +23.2.
† Not adjustable.

All voltages are to be adjusted to ±5% except +23.2 V, which is +3%, -5%.

REGULATOR SPECIFICATIONS

Regulator	Voltage and Tolerance	Output Current (Maximum)	P-P Ripple (Maximum)
H744	+5 Vdc ± 5%	25 A	200 mV
H745	-15 Vdc ± 5%	10 A	450 mV
H746	+23.2 Vdc +3, -5%	1.6 A (1)	700 mV
	+19.7 Vdc	3.3 A (1)	700 mV
	-5 Vdc	1.6 A (1)	150 mV
H754	+20 Vdc ± 5%	8 A	5% (3)
	-5 Vdc ± 5%	1 A-8 A (4)	5% (3)
H742	+15 Vdc ± 10%	3 A (2)	—
	+8 Vdc ± 15%	1 A (2)	—
	20-30 Vac (5 outputs)	300 W each output, 1 Kw max. total output.	—

Notes:

1. Refer to drawing D-CS-H746-0-1. Since the 19.7 V output is obtained by regulating down from the +23.2 V level, any combination of loads on the two outputs is acceptable as long as the sum does not exceed 5 A.

Notes (Cont):

Negative 5 V level is obtained by inserting a 5.1 V zener diode in series with the +23.2 and +19.7 loads, and using the zener cathode as GND. Therefore, maximum −5 V load current is equal to the greater of 1.6 A or the sum of the two positive load currents (+23 and +19).

2. Total not to exceed 3 A continuously.

3. At backplane. Typical ripple ≈±3%.

4. Maximum −5 V current is dependent upon +20 V current. It is equal to 1 A + I(+20) up to a total of 8 A. (I (+20) is the amount of +20 V current.)

PDP-11/40 REGULATOR/SLOT ASSIGNMENTS

Regulators					H742 Bulk Power Supply
E	D	C	B	A	
H745 −15 V or H754 +20 V −5 V if MM11-U/UP	H745 −15 V or H754 +20 V −5 V if more MM11-U/UP	H744* +5 V	H744 +5 V	H744 +5 V	AC LO DC LO 20–30 Vac +15 V +8 V LTC

* Some earlier versions with small power requirements had no regulator in slot C.

PDP-11/45 REGULATOR SLOT ASSIGNMENTS
(Serial Numbers Beow 2000)

H745	H744	H744	H744	H744	
REGULATORS					BULK SUPPLY A
E	D	C	B	A	REGULATORS A B C D E SWITCHED
+15V CENTRAL PROCESSOR −15V TO ROWS 1.2.15	+5V INTERNAL OPTIONS	+5V CENTRAL PROCESSOR	+5V CENTRAL PROCESSOR	+5V FLOATING POINT	+15V TO REGS E F
INTERNAL OPTIONS −15V TO ROWS 26.27.28 CONSOLE	+5V TO ROWS 26.27.28				+15V TO ROW 13 CONSOLE
	SYSTEM UNITS + 5V TO SYSTEM UNITS #1 #2 #3				+ 8V TO ROW 1 FOR MAINT MODULES
SYSTEM UNITS −15V TO SYS UNITS #1 #2		+5V TO ROWS 10.11.12.13.14.15	+5V TO ROWS 1.6.7.8.9	+5V TO ROWS 2.3.4.5	50/60 HZ SIG (0 TO +5V) TO ROW 1 FOR CLOCK MODULE

H742 Upper Power Supply

H744 or H746	H744	H744	H744 or H746	H745	
REGULATORS					BULK SUPPLY B
L	K	J	H *	F	REGULATORS F H J K L NOT SWITCHED
+5V BIPOLAR MEMORY	+5V BIPOLAR MEMORY	+5V IF BIPOLAR MEMORY IS INSTALLED	+5V IF BIPOLAR MEMORY IS INSTALLED	−15V SYSTEM UNITS	
+5V TO ROWS 24. 25		+5V TO ROWS 19.20	+5V TO ROWS 16.17.18		
+19V +23V −5V IF MOS MEMORY IS INSTALLED		+5V IF MOS MEMORY IS INSTALLED	+19V +23V −5V IF MOS MEMORY IS INSTALLED		
MOS VOLTAGES TO ROWS 22 - 25	+5V TO ROWS 21.22.23	+5V TO ROWS 16.17.18.19.20. 21.22.23.24.25	MOS VOLTAGES TO ROWS 17 - 20	−15V TO SYS UNIT #3	* REG H WILL BE EITHER A +5V OR MOS VOLTAGE REG

H742 Lower Power Supply

PDP-11/45 REGULATOR SLOT ASSIGNMENTS
(Serial Numbers 2000 and Higher)

H745 or H754 | H744 | H744 | H744 | H744

REGULATORS					
E	D	C	B	A	H742 BULK SUPPLY
15V TO SYSTEM UNITS 1 2 3	+5V INTERNAL OPTIONS +5V TO ROWS 26 27 28	+5V CENTRAL PROCESSOR	+5V CENTRAL PROCESSOR	+5V FLOATING POINT	-15V TO REGS E F ROW 13 & CONSOLE
					ACLO DCLO
+20V -5V ALTERNATE TO SYSTEM UNITS 1 2 3	-5V TO SYSTEM UNITS 1 2 3				+8V TO ROW 1 FOR MAINT MODULES
		-5V TO ROWS 10 15	-5V TO ROWS 1 6 7 8 9	+5V TO ROWS 2 3 4 5	50 60 HZ SIG (0 TO +5V) TO ROW 1 FOR CLOCK MODULE

H742 Upper Power Supply

H744 or H746 | H744 | H744 | H744 or H746 | H745

REGULATORS					
L	K	J	H	F	H742 BULK SUPPLY
+5V BIPOLAR MEMORY +5V TO ROWS 24 25	+5V BIPOLAR MEMORY	+5V IF BIPOLAR MEMORY IS INSTALLED +5V TO ROWS 19 20	+5V IF BIPOLAR MEMORY IS INSTALLED +5V TO ROWS 16 17 18	+15V CENTRAL PROCESSOR 15V TO ROWS 1 2 15	-15V TO CENTRAL PROCESSOR
+19V +23V -5V IF MOS MEMORY IS INSTALLED MOS VOLTAGES TO ROWS 22 - 25	+5V TO ROWS 21 22 23	+5V IF MOS MEMORY IS INSTALLED +5V TO ROWS 16-25	+19V +23V -5V IF MOS MEMORY IS INSTALLED MOS VOLTAGES TO ROWS 17 - 20	INTERNAL OPTIONS 15V TO ROWS 26 27 28 CONSOLE	ACLO DCLO

H742 Lower Power Supply

PDP-11/45 VOLTAGE REGULATOR CONFIGURATION DATA
CABINET SERIAL NUMBERS 2000 AND HIGHER

Type	Regulator Name	Quantity	Location	Comments
Basic System				
H744	+5 V	3	B	+5 V to CPU modules slots 6–9.
			C	+5 V to CPU and KT11-C modules, slots 10–15.
			D	+5 V to internal options, slots 26–28, system units 1, 2, and 3, and console.
H745	−15 V	1	F	−15 V to CPU and internal option modules. This supply is switched, even in the lower H742, because it is fed by +15 Vdc from the upper H742.
FP11-B Floating Point Processor				
H744	+5 V	1	A	+5 V to FP11 modules, slots 2–5.

PDP-11/45 VOLTAGE REGULATOR CONFIGURATION DATA CABINET SERIAL NUMBERS 2000 AND HIGHER (CONT)

Type	Regulator Name	Quantity	Location	Comments
MS11-C Bipolar Memory				
H744	+5 V	2	H, J	+5 V to control and matrix modules if no MOS memory is installed, or only 4K is used. H: slots 16–18; J: slots 19–20.
		2	K, L	If MOS memory is also installed, or if more than 4K of bipolar is used. K: slots 21–23, L: slots 24–25.
MS11-B MOS Memory				
H744	+5 V	1	J	+5 V to control and matrix modules, slots 16–25.
H746	MOS	2	H, L	+19.7 V, +23.2 V, and −5 V to MOS matrix modules; H slots 17–20; L slots 22–25.
MM11 Core Memories and Controls				
H745	−15 V	1	E	−15 V to system units 1–3.
H754	+20, −5 V	1	E	20 and −5 Vdc to MF11-U/UP. No −15 Vdc available for other system units.

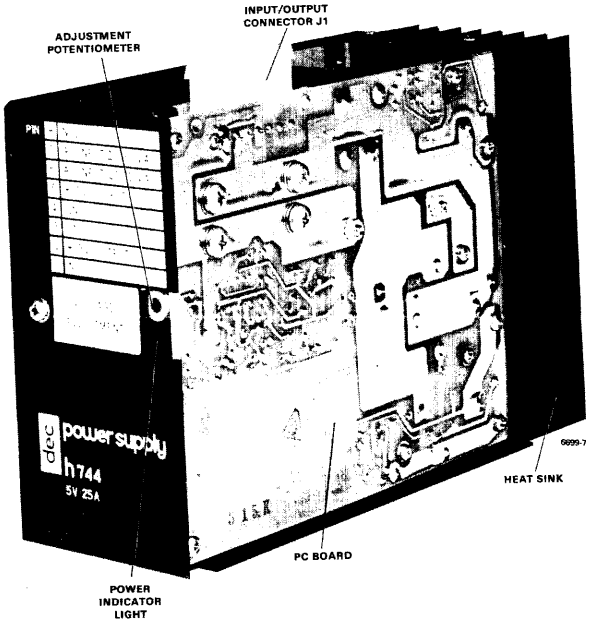
PDP-11/45 VOLTAGE REGULATOR CONFIGURATION DATA
CABINET SERIAL NUMBERS LESS THAN 2000

Type	Regulator Name	Quantity	Location	Comments
Basic System				
H744	+5 V	3	B	+5 V to CPU modules, slots 6–9.
			C	+5 V to CPU and KT11-C modules, slots 10–15.
			D	+5 V to internal options, slots 26–28, system units 1, 2, and 3, and console.
H475	−15 V	1	E	−15 V to CPU and internal option modules and system units 1 and 2.
FP11-B Floating Point Processor				
H744	+5 V	1	A	+5 V to FP11 modules, slots 2–5.
MS11-C Bipolar Memory				
H744	+5 V	2	H, J	+5 V to control and matrix modules if no MOS memory is installed, or only 4K is used. H: slots 16–18; J: slots 19–20.

PDP-11/45 VOLTAGE REGLATOR CONFIGURATION DATA CABINET SERIAL NUMBERS LESS THAN 2000 (CONT)

Type	Regulator Name	Quantity	Location	Comments
		2	K, L	If MOS memory is also installed, or if more than 4K of bipolar is used. K: slots 21–23, L: slots 24–25.
MS11-B MOS Memory				
H744	+5 V	1	J	+5 V to control and matrix modules, slots 16–25.
H746	MOS	2	H, L	+19.7 V, +23.2 V, and –5 V to MOS matrix modules; H slots 17–20; L slots 22–25.
MM11 Core Memories and Controls				
H745	–15 V	1	F	–15 V to system unit 3. H745 provided in basic system supplies system units 1 and 2. This supply is switched even in the lower H742, because it is fed by +15 Vdc from the upper H742.

H744, +5 VDC REGULATOR



CROWBAR PROBLEMS

As you are powering the machine on and off (or for various other mysterious reasons), it can happen that one or more of the regulators may crowbar. That is, the regulator short circuits its output to protect the load. The output of a crowbarred regulator would be seen as approximately 0 V.

To check for this, first power the machine down. (This is very important; otherwise the crowbar circuit will not reset.) Then rotate the adjustment potentiometer of the offending regulator(s) counterclockwise a half revolution. Turn the power back on and observing the voltage level with a scope on the backplane pin, readjust the potentiometer so that the regulator is providing the correct output.

If you can't adjust the regulator's output to the correct level, then something else is causing the fault. Continue on the PWRB flows.

If you find that the regulator crowbars again after this procedure, then you have something connected to the regulator's output that is causing it to go too high. This is often due to a poor connection in one of the connectors in the power harness. Work the connectors firmly into their sockets and manipulate the wires going into the connectors.

Also look for possibilities such as a loose piece of wire shorting out a couple of backplane pins or else a backplane pin that is bent and is touching a neighboring pin.

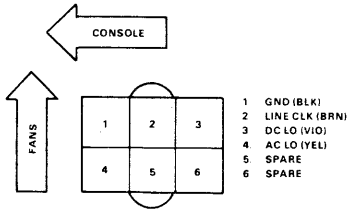
DISTRIBUTION PANEL CONNECTORS

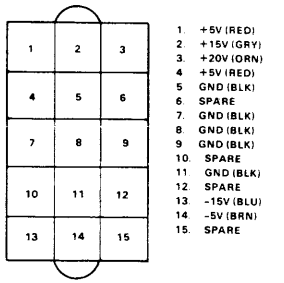
The following diagrams show the locations of the pins carrying the power and signals into and out of the various distribution panels used on all models of the PDP-11/40 and PDP-11/45 processors.

Check for these voltages by wedging the scope probe down into the connector's pin socket so that the probe makes connection with the pin inside the socket.

Note that the pin layout for the distribution panel supplying power to the system units on PDP-11/45s with serial numbers 2000 and higher is the same as the one for distribution panels used on PDP-11/40s with serial numbers 6000 and higher.

PDP-11/40 DISTRIBUTION PANEL CONNECTORS FOR SERIAL NUMBERS 6000 AND HIGHER

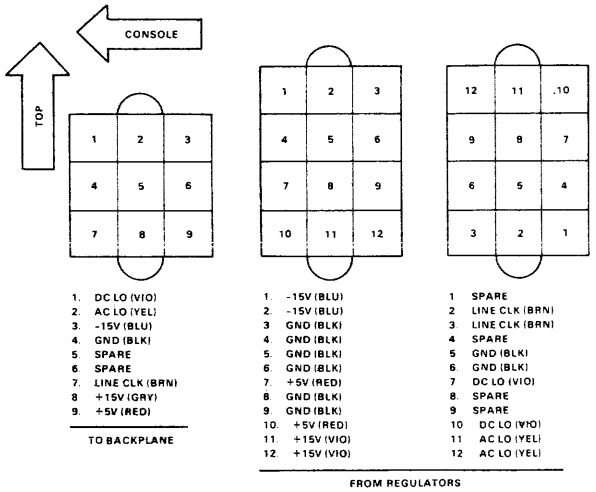




(Continued)

In the mainframe box's distribution panel there are four groups containing five pairs as shown above. Within a group, all pairs are bussed together pin-for-pin by the distribution panel. Each connector may or may not have all the above-listed wires connected to it.

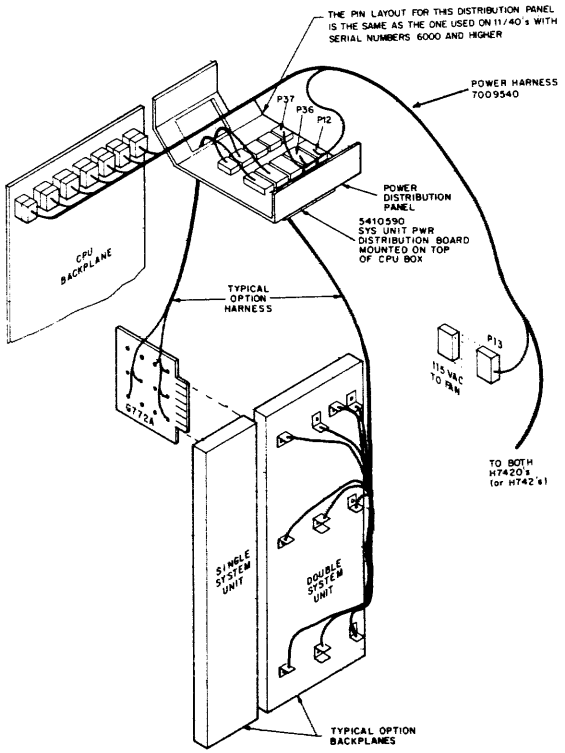
PDP-11/40 DISTRIBUTION PANEL CONNECTORS FOR SERIAL NUMBERS LESS THAN 6000



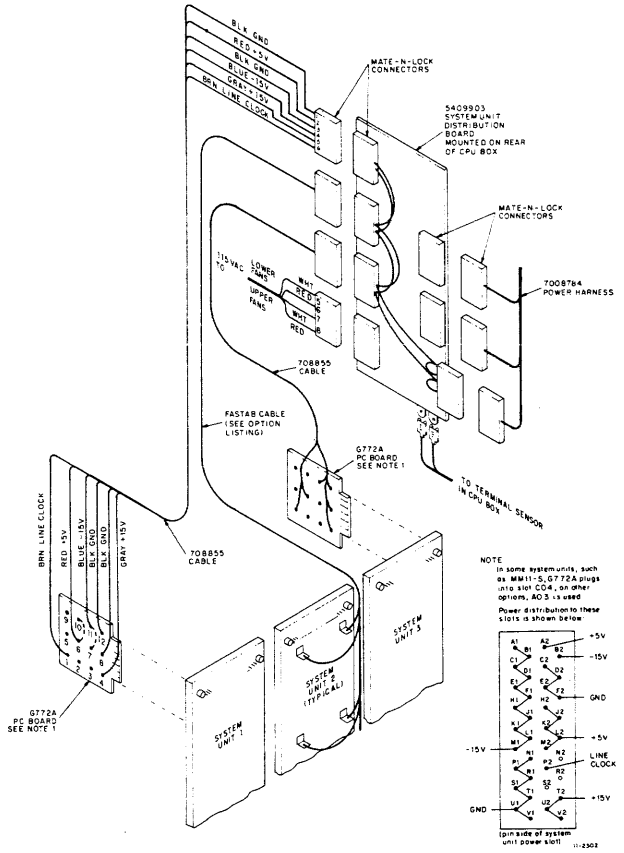
In the mainframe box's distribution panel, there are three groups each containing five connectors, that is, the two 12-pin connectors shown above and three of the 9-pin connectors shown above. Within a group, all three 9-pin connectors are bussed together pin-for-pin by the distribution panel. Each connector may or may not have all the above-listed wires connected to it.

PDP-11/45 BACKPLANE CONNECTORS AND PINS

INSTALLATION OF SYSTEM UNITS, LATER SYSTEMS, PDP-11/45 CABINET SERIAL NUMBERS 2000 AND HIGHER



INSTALLATION OF SYSTEM UNITS, EARLY SYSTEMS, PDP-11/45 CABINET SERIAL NUMBERS LESS THAN 2000



POWER HARNESSES AND CABLES

Before you replace any of the cables or harnesses, ensure that the connectors are making a solid electrical connection. Firmly work the connectors into their sockets and manipulate the wires at the connectors while looking for any changes in the problem symptoms. If this procedure causes a change, then it may only be necessary to reset the connectors in their sockets or the pins in their connectors.

If you find that it is necessary to replace any cables, refer to the following pages of this DECAID to make sure that you obtain the correct replacement.

Present and future power handling capabilities of the 11/40-11/45 CP expander boxes and H960D, E cabinets have been increased with the creation of two new main power harnesses.

Along with this comes a whole new set of option cables necessary to connect our existing options to this new power configuration.

New and old machines may be easily identified by visually inspecting their serial number affixed to the expander box or by noting the new main power harness that is installed within the expander box.

Serial numbers > 6000 indicate a new PDP-11/40 power configuration (< = old type).

Serial numbers > 6000 indicate a new H960D, E power configuration (< = old type).

Serial numbers > 2000 indicate a new PDP-11/45 power configuration (< = old type).

The following tables list the options and CPU power harnesses affected by this change. Also included are the part numbers for other power cables not affected along with remote sense and intermodule cables used in the PDP-11 world.

POWER DISTRIBUTION HARNESS
(Power Supply to Distribution Points)

Description	Cable
PDP-11/15, PDP-11/20	70-6518
BA11-ES	70-5894
PDP-11/05 NC, PDP-11/35 10-1/2-inch box	70-9208
PDP-11/45, PDP-11/50 (Old)	70-8784
PDP-11/40 and H960D/E (Old)	70-8754
PDP-11/45 and PDP-11/50 (New)	70-9540
PDP-11/40 and H960D/E (New)	70-9566

* There will be some units between serial numbers 6000 and 6542 that will still have the old configuration.

DC DISTRIBUTION TO CPU BACKPLANE

Description	Cable
PDP-11/05S	70-9918
PDP-11/05 NC PDP-11/10 10-1/2-inch box	70-9360
PDP-11/35	70-9209
PDP-11/40 (Old)	70-9046
PDP-11/45	N/A
PDP-11/40 (New)	70-9564
PDP-11/35S	70-10113

NOTE

If a new PDP-11/35-PDP-11/40 backplane is ordered (part no. 70-10230), you must order 70-9994 cable for new-style PDP-11/40s. For old-style PDP-11/40s, use original cable cut to length.

REMOTE SENSE CABLES

Description	Cable
861 Remote Sense Cable (11/40 to 861)	70-9053
861/860 Remote Sense Cable (861/860 to 861/860)	70-8288
861/860 Remote Sense Cable (861/860 to 1st H720E/F)	70-8964
H720E/F to H720E/F Remote Sense Cable	70-8288
H720E/F (J2) Remote Sense Terminator Plug	70-7006-1
H720E/F (J1) Local Sense Jumper Plug	70-7006-2

INTER-MODULE CABLES

CPU Type	Description	Cable
PDP-11/05 (5-1/4 inch)	Console Cable	BC08R-3
PDP-11/05 (10-1/2 inch)	Console Cable	BC08R-4
PDP-11/35 PDP-11/35-11/40	Console to K1 and K5 (2) KE11-E (M7238 to M7232) (3)	BC08R-3 BC08R-1
PDP-11/40 PDP-11/45	Console to K1 and K5 (2) Console Cable (2)	BC08R-6 70-8864

OPTIONS

CPU Type Option	11/05, 11/10 10-1/2-In. Box	11/35 10-1/2-In. Box	11/35S, 11/05S BA11-KE/F 10-1/2-In. Box
AA11-DA	70-9205	70-9205	70-9562
AA11-DB	70-9205	70-9205	70-9562
DA11-F	70-9099	70-9099	70-10117
DB11-A	70-9205	70-9205	70-9562
DC11-A	70-9205	70-9205	70-9562
DD11-A	70-9205	70-9205	70-9562
DD11-B	70-9099	70-9099	70-10117
DH11-AA	N/A	N/A	70-10118
DH11-AB	N/A	N/A	70-10118
DH11-AC	N/A	N/A	70-10118
DJ11-AA	70-9099	70-9099	70-10117
DJ11-AB	70-9099	70-9099	70-10117
DJ11-AC	70-9099	70-9099	70-10117
DN11-AA	70-9205	70-9205	70-9562
DP11-DA	70-9205	70-9205	70-9562
DR11-B	70-9205	70-9205	70-9562
MF11-L	70-9206	70-9206	70-10114
MF11-LP	N/A	70-9206	70-10114
MF11-U/UP	N/A	N/A	70-10115
MM11-S	70-9205	70-9205	70-9562
RH11	70-9099	70-9099	70-9570
RH11-AB	70-9099	70-9099	70-10117
RK11-D	70-8992	70-8992	70-10116
VT11	70-9099	70-9099	70-10117
CD11-A/B/E	70-9099	70-9099	70-10117
DQ11	70-9099	70-9099	70-10117

OPTIONS

CPU Type Option	11/45 (Old)	11/40 H960D, E (Old)**	11/40, 11/45 H960D/E (New)
AA11-DA	70-8855-1J	70-8909-XX	70-9562
AA11-DB	70-8855-1J	70-8909-XX	70-9562
DA11-F	70-9162	70-9099	70-9563
DB11-A	70-8855-1J	70-8909-XX	70-9562
DC11-A	70-8855-1J	70-8909-XX	70-9562
DD11-A	70-8855	70-9177 70-8909-XX	70-9562
DD11-B	70-9162	70-9099	70-9563
DH11-AA	N/A	70-9466	70-9561
DH11-AB	N/A	70-9466	70-9561
DH11-AC	N/A	70-9466	70-9561
DJ11-AA	70-9162	70-9099	70-9563
DJ11-AB	70-9162	70-9099	70-9563
DJ11-AC	70-9162	70-9099	70-9563
DN11-AA	70-8855-1J	70-8909-XX	70-9562
DP11-DA	70-8855-1J	70-8909-XX	70-9562
DR11-B	70-8855-1J	70-8909-XX	70-9562
MF11-L*	70-9242	70-9103	70-9565
MF11-L†	N/A	70-9174	70-9560
MF11-LP*	70-9242	70-9103	70-9565
MF11-LP†	N/A	70-9174	70-9560
MF11-U/UP‡	N/A	N/A	70-9535
MM11-S	70-8855-2B	70-8909-XX	70-9562
RH11	70-9162	70-9571	70-9570
RH11-AB	70-9162	70-9099	70-9563
RK11-D	70-8855-1J	70-8992	70-9559
VT11	70-9162	70-9099	70-9563
CD11-A/B/E	70-9162	70-9099	70-9563
DQ11	70-9162	70-9099	70-9563

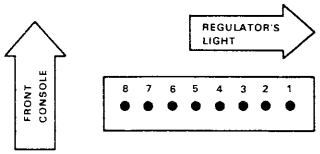
* PDP-11/40 only (1st MF11-L/LP).

† PDP-11/40 only (2nd MF11-L/LP).

‡ MF11-U/UP cannot be mounted in old style PDP-11/45 CPU box.

** 70-8909-XX cable has two variations – 11 inches and 17 inches. Use 70-9177 if due to new module guide layout; 70-8909-XX cables are too short.

PIN LOCATIONS FOR REGULATOR'S CONNECTORS



H744

1. Spare
2. +5 V output (red)
3. GND (black)
4. GND (black)
5. +5 V output (red)
6. 20–30 Vac input (red)
7. 20–30 Vac input (white)
8. Spare

H745

1. –15 V output (blue)
2. GND (black)
3. GND (black)
4. +15 V input (violet or gray)
5. +15 V input (violet or gray)
6. 20–30 Vac input (red)
7. Spare
8. 20–30 Vac input (white)

H754

1. Spare
2. GND (black)
3. –5 V output (brown)
4. Spare
5. +20 V output (orange)
6. Spare
7. 20–30 Vac input (red)
8. 20–30 Vac input (white)

H746

1. Spare
2. GND
3. –5 V output
4. +19.7 V output
5. +23.2 V output
6. Spare
7. 20–30 Vac input
8. 20–30 Vac input

PROCEDURE FOR REMOVAL OF REGULATORS

1. Turn the power off at the circuit breaker on the 860/861 Power Controller.

2. Disconnect the connector at the top of the regulator.

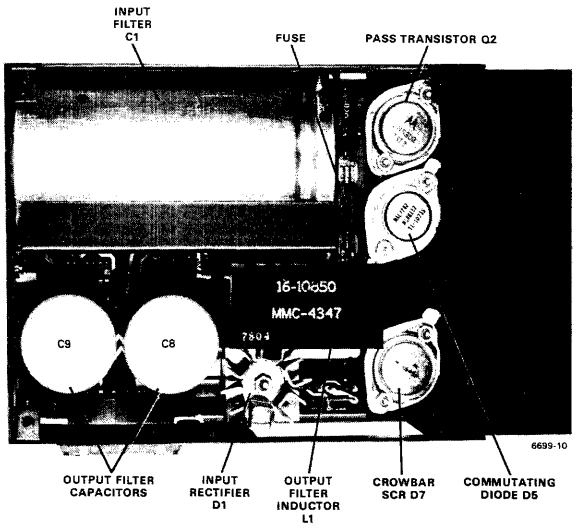
3. Remove the two Phillips screws on top of the regulator which hold the regulator in the H742 box.

4. Loosen the finger-tight (usually) screw on the bottom of the regulator.

5. Slide the regulator out horizontally away from the H742 box.

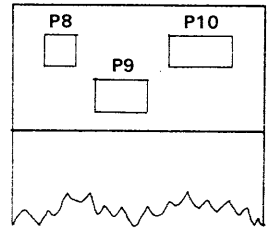
The following diagram shows the location of the fuse. Usually you can check it without taking off the clear plastic cover. It is an AGC 15-amp fuse.

+5 V REGULATOR, SIDE VIEW



H742 CONNECTORS FOR PDP-11/40

As seen from the back, the connectors on the H742 Power Supply are located as shown in the diagram below.



The following pages list the functions and locations of each of the pins on the above three connectors.

PDP-11/40 Pin Locations for H742's Connectors

P8

3	6	9
2	5	8
1	4	7

Pin(s)	Function (Vac)	Color	Dest.
1–2 (new)	20–30	Red, white	Slot B
1–2 (old)	20–30	Red, white	Slot A
3–4	20–30		
5–6	20–30		
7–8	20–30	Red, white	Slot D

PDP-11/40 Pin Locations for H742's Connectors (Cont)

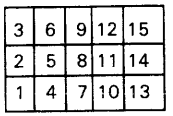
P9

3	6	9	12
2	5	8	11
1	4	7	10

Pin(s)	Function	Color	Dest.
1	+8 V		
2	+15 V	Gray or violet	Dist. Panel
3	+15 V	Gray or violet	Slot D
4	GND	Black	Slot D
5	GND	Black	Slot E
6	GND	Black	Dist. Panel
7	GND	Black	Dist. Shield
8	AC LO	Yellow or red	Dist. Panel
9	DC LO		
10	AC LO		
11	LINE CLK	Brown or black	Dist. Panel
12	DC LO	Violet or clear	Dist. Panel

PDP-11/40 Pin Locations for H742's Connectors (Cont)

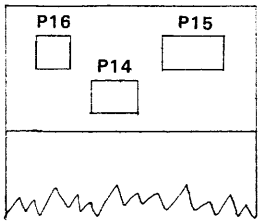
P10



Pin(s)	Function (Vac)	Color	Dest.
1–2 (new)	20–30	Red, white	Slot A
1–2 (old)	20–30	Red, white	Slot B
8–10	20–30	Red, white	Slot C
9–12	20–30	Red, white	Slot E
3–7	115	Red, white	H742 fan
5–6	115	Red, white	Proc. fans

UPPER H742 CONNECTORS FOR PDP-11/45

As seen from the back, the connectors on an upper H742 Power Supply are located as shown in the diagram below.



The following pages list the functions and locations of each of the pins on the above three connectors.

PDP-11/45 Pin Locations for Upper H742's Connectors

P16

3	6	9
2	5	8
1	4	7

Pin(s)	Function (Vac)	Color	Dest.
1–2	20–30		
3–4	20–30	Red, white	Slot B
5–6	20–30		
7–8	20–30	Red, white	Slot C

PDP-11/45 Pin Locations for Upper H742's Connectors (Cont)

P14

3	6	9	12
2	5	8	11
1	4	7	10

Pin(s)	Function	Color	Dest.
1	+8 V	White	Dist. Panel
2 (new)	+15 V	Gray	Slot F
2 (old)	+15 V	Gray	Slot E
3	+15 V	Gray	Dist. Panel
4	GND	Black	Slot E
5	GND	Black	Dist. Panel
6	GND	Black	Slot F
7	GND	Black	Dist. Panel
8	AC LO	Yellow	Dist. Panel
8	AC LO	Yellow	Lower H742
9 (new)	DC LO		
9 (old)			
10	AC LO	Yellow	Dist. Panel
10	AC LO	Yellow	Lower H742
11	LINE CLK	Brown	Expn. Dist.
12	DC LO	Violet	Dist. Panel

PDP-11/45 Pin Locations for Upper H742's Connectors (Cont)

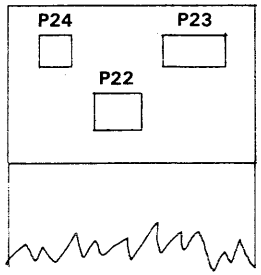
P15

3	6	9	12	15
2	5	8	11	14
1	4	7	10	13

Pin(s)	Function (Vac)	Color	Dest.
1–2	20–30	Red, white	Slot A
8–10	20–30	Red, white	Slot D
9–12	20–30	Red, white	Slot E
3–7	115	Red, white	H742 fan
5–6	115	Red, white	Time meter

LOWER H742 CONNECTORS FOR PDP-11/45

As seen from the back, the connectors on a lower H742 Power Supply are located as shown in the diagram below.



The following pages list the functions and locations of each of the pins on the above three connectors.

PDP-11/45 Pin Locations for Lower H742's Connectors

P24

3	6	9
2	5	8
1	4	7

Pin(s)	Function (Vac)	Color	Dest.
1–2	20–30	Red, white	Slot J
3–4	20–30		
5–6	20–30	Red, white	Slot H H746
7–8	20–30	Red, white	Slot H H744

PDP-11/45 Pin Locations for Lower H742's Connectors (Cont)

P22

3	6	9	12
2	5	8	11
1	4	7	10

Pin(s)	Function	Color	Dest.
3	GND	Black	Dist. Panel
4	-15 V	Blue	Dist. Panel
7	GND	Black	Dist. Panel
8	AC LO	Yellow	Upper H742
9	DC LO	Violet	Dist. Panel
10	AC LO	Yellow	Upper H742
12	DC LO	Violet	Dist. Panel

PDP-11/45 Pin Locations for Lower H742's Connectors (Cont)

P23

3	6	9	12	15
2	5	8	11	14
1	4	7	10	13

Pin(s)	Function (Vac)	Color	Dest.
1–2	20–30	Red, white	Slot L
8–10	20–30	Red, white	Slot K
9–12	20–30	Red, white	Slot F
5–6	115	Red, white	Proc. fans
3–7	115	Red, white	H742 fan

PROCEDURE TO CHECK THE FUSE AND TO REMOVE THE POWER CONTROL BOARD WITHIN THE H742 POWER SUPPLY

CAUTION

You are working with 115 Vac. Be very sure to disconnect the H742 from the power source and allow about 10 seconds for the capacitors to discharge before starting this procedure.

Lift off the H742's top and side cover (one piece) after removing the two screws located at the front and rear of the power supply box. Don't pull it away too far as the fan on the cover is connected to a terminal board in the box.

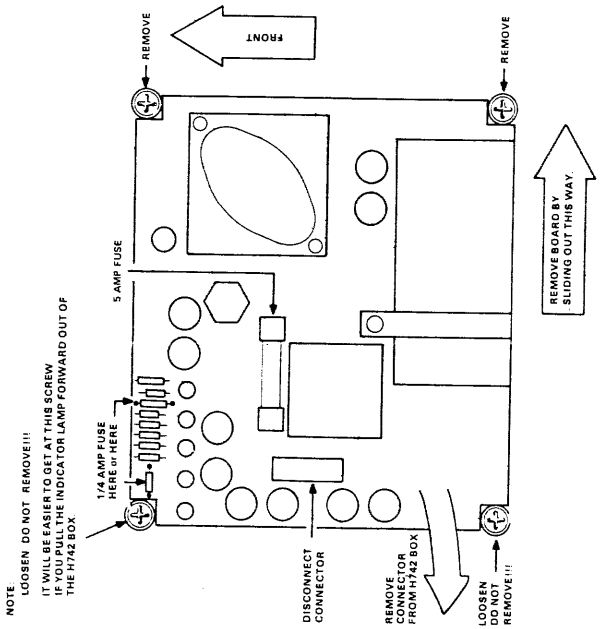
Check the fuses (5 amp and 1/4 amp) on the power control board. The following diagram shows their locations.

If it is necessary to replace the power control board:

- Loosen but **do not remove*** the two screws indicated in the diagram.
- Remove the screws indicated.
- Disconnect the connector from the board.
- Remove the other connector from the box.
- Slide out the power control board.

* Do not remove these screws because it is very likely that in trying to put them back in, you will drop them down into the H742 box.

H742'S POWER CONTROL BOARD



PDP-11/05/10 POWER CONNECTORS

See the next page for a diagram shownin the pin locations.

PDP-11/05/10 5-1/4-INCH BOX

Voltage	Tolerance	Allowable Ripple
+15 V	±5%	750 mV
+5 V	±5%	200 mV
-15 V	±5%	450 mV

PDP-11/05-N, 11/10-N 10-1/2-INCH BOX

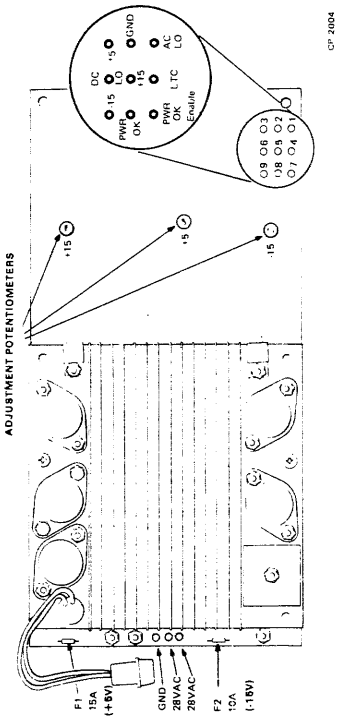
Voltage	Tolerance	Allowable Ripple	Regulator
+15 V	±5%	500 mV	5409728
+5 V	±5%	200 mV	5409728
-15 V	±5%	450 mV	5409728
+5 V	±5%	200 mV	H744

The two +5 V outputs must *not* be shorted together. The 5409728 regulator provides +5 V to connectors J1 and J2 on the power distribution board. The H744 provides +5 V to connectors J3, J4, and J5. The CPU backplane is normally connected to J1.

PDP-11/05-S, 11/10S

Voltage	Tolerance	Allowable Ripple	Wire Color	Pin Number
+5 V	±5%	200 mV	Red	1, 4
+15 V	±10%		Gray	2
+20 V	±5%	100 mV	Orange	3
-15 V	±5%	450 mV	Blue	13
-5 V	±5%	250 mV	Brown	14

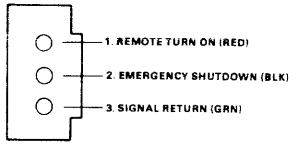
54-09728 POWER REGULATOR



REMOTE POWER CONTROL FACILITY

All PDP-11 computers have provisions to allow the console switch to control the operation of the cabinet mounted 860/861 Power Controllers and H720 Power Supplies.

These devices should be connected together with the remote power control bus which is a 3-wire system with the wires having the functions shown below:



POWER CONTROL OPERATION

Connections Between Control Lines	Switch Position		
	Local Switched Power Is	Off Switched Power Is	Remote Switched Power Is
None	On	Off	Off
1–3	On	Off	On
2–3	Off	Off	Off
1–3, 2–3	Off	Off	Off

See DECAID PWRB-5 for a list of the remote sense cables used to connect various power components.

SUGGESTIONS ON WHAT TO DO BEFORE ACTUAL HARDWARE TROUBLESHOOTING

Before you go to the customer's site:

1. Find out as much about the system configuration as you can.

   - Which CPU
   - How much memory, what type
   - What mass storage devices, how many of each
   - What software
   - What other peripherals, how many of each
   - What communications equipment, etc.

2. Read the last few Field Service Reports for this system to see if the problem is a recurring one or if it may be related to some work that was done recently.

When you get to the customer's site:

1. Question (gently) the user to find out as much about the failure as possible.

   - What was the system doing when it failed?
   - Was there any smoke, sparks, noises, etc.?
   - What software was running?
   - How many users were on the system?
   - Have there been any recent changes to the hardware or software?

2. Check the obvious and simple possibilities first.

   - Power plugs not inserted properly
   - Cables hanging loose
   - Physical damage