EK-OKL10-MG-004

KL10 Maintenance Guide

Volume II

Prepared by Educational Services
of
Digital Equipment Corporation
Marlboro, MA

FOR INTERNAL USE ONLY

1st Edition, February 1979
2nd Edition, December 1983
3rd Edition, April 1985
4th Edition, June 1986

Printed in U.S.A

ORDERING PROCEDURES

1. KL10/KS10 Maintenance Guides

To order one or both of these guides (if you are not on the distribution list), send your order by:

MEMO: Publishing and Circulation Services
ATTN: Order Admin NRO3/W3
10 Forbes Road
Northboro, MA 01532

DECMAIL: ORDER @NRO

Specify the following:

Name
Badge Number
Cost Center
Location
Quantity: 1 (only one copy of each guide per name)
EK-OKL10-MG (KL10 Maintenance Guide)
EK-OKS10-MG (KS10 Maintenance Guide)

2. Deletions and Transfers

To be deleted from the distribution list or to transfer your subscription to someone else, send a memo to:

Publishing and Circulation Services
ATTN: List Maintenance NRO3/M1
10 Forbes Road
Northboro, MA 01532

Specify the following:

Remove: Name
Badge Number
Cost Center
Location

Add: New Name
Badge Number
Cost Center
Location

State which Maintenance Guide you want to stop receiving or to transfer:

EK-OKL10-MG (KL10 Maintenance Guide)
or
EK-OKS10-MG (KS10 Maintenance Guide)

To the Reader:

IMPORTANT - The information in this guide is for internal use only. Is intended for DIGITAL Field Service engineers only. Refer to the Field Service Methods and Procedures Manual for company policy on internal information.

OBJECTIVE - The objective of this guide is to organize and present the maintenance information needed to resolve 80 percent of all KL10 hardware malfunctions.

To properly maintain and improve this guide in future revisions, we need feedback concerning accuracy and clarity. This information is very helpful to your fellow engineers. Please forward any corrections, suggestions, and comments that would improve this guide to:

Customer Services Systems Engineering (CSSE)
RE: KL10 Maintenance Guide
MR01-1/S35

ORGANIZATION - Volume I contains general maintenance information for the KL10. The volume is divided into tabbed sections with separate tables of contents as follows.

1. GENERAL INFORMATION contains miscellaneous maintenance information that cannot be classifed and filed in the other hardware sections.

2. SWITCHES AND JUMPERS contains information on hardware switch positions and jumper connections.

3. TABLES AND MAPS describes the process tables and bit maps associated with the KL10 mainframe and peripheral equipment.

4. CHECKS/ADJUSTMENTS contains check and adjustment procedures performed during preventive and corrective maintenance.

5. DIAGRAMS AND MULs contains block diagrams, power supply layouts, and module usage lists associated with KL10-based systems.

6. MULTI-CPU contains maintenance and diagnostic information specific to multiprocessor systems.

7. DECnet-10/20 contains system hardware and software information.

The information in each hardware section is arranged according to unit and subsystem (that is, CPU, memory, disk, tape, and I/O).

Volume II contains additional hardware and software information on the KL10. The volume is divided into tabbed sections with separate tables of contents as follows.

1. COMPUTER INTERCONNECT contains descriptions of the card cage, module locations, switch settings, bit/error formats, diagnostics, and label information.

2. NETWORK INTERCONNECT describes the card cage, module locations, switch settings, bit/error formats, diagnostics, and label information.

3. HSC SUBSYSTEM contains RA81, RA60, and HSC50 Error Codes.

4. CLUSTER contains procedures for fault isolation on the cluster level.

5. RP07/RP20 describes RP07 registers and RP20 FSC, jumper, routines, error stops, and other information.

6. S/X BUS gives a general description of the S/X Bus and installation and operation information.

7. ARM-10LS contains installation, operation, and memory fault isolation information.

8. MAINTENANCE SOFTWARE contains information on the DIACON, KLDCP, KLDCPU, MEMCON, TRACON, DIAMON, DDT, D20MON programs.

9. SYSTEM SOFTWARE contains information on typical operating systems and command formats.

10. RSX-20F contains information on programs SYSLIB-20F and PARSER, in addition to stop/error codes.

11. TOPS-10 contains information on TOPS-10, DECnet-10, GALAXY-10, and PIP programs.

12. TOPS-20 contains system program, command summary/format, and error message information.

13. NOTES provides blank pages for note taking.

System Software

Table of Contents

SYSTEM SOFTWARE
System software is one of the three major categories of software. Refer to Figure 1.

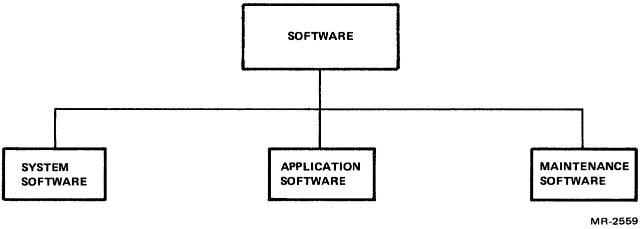


Figure 1 Three Major Categories of Software

System software consists of an operating system or monitor and a library of Commonly Used System Programs (CUSPs). Refer to Figure 2.

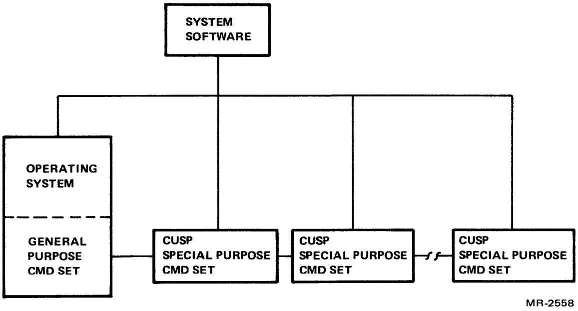


Figure 2 Component Parts of System Software

The operating system directs and monitors the overall performance of the system and supports a general purpose command set. The CUSPs, in effect, extend the general purpose command set by supporting individual special purpose command sets.

SYSTEM MONITORING - Directing and monitoring the overall performance of the system is the most complex aspect of an operating system. It involves tasks such as scheduling jobs for execution, directing I/O operation, handling interrupts, and managing system resources. Although field maintenance personnel should have an overall understanding of this aspect of operating systems, an in-depth knowledge is not generally required.

COMMONLY USED SYSTEM PROGRAMS (CUSPs) - The number and type of CUSPs associated with a given system program library depends largely on the intended use of the system. Regardless of the intended use of the system, however, the relationship between the operating system and the CUSPs in the corresponding system program library will remain the same. That is, the operating system will support a set of general purpose commands and each CUSP will support a unique set of special purpose commands. Refer to Figure 3.

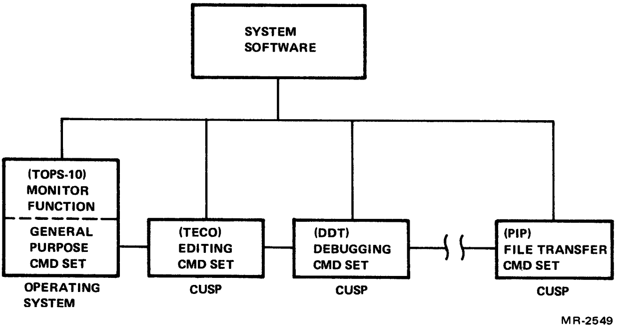


Figure 3   Typical Operating System and CUSP Relationship

Figure 3 uses the TOPS-10 operating system and three CUSPs from the TOPS-10 system program library to illustrate the relationship between operating systems and CUSPs.

The general purpose command set supported by the operating system enables system programmers, operators and users to perform the following functions: gain access to the system, run existing system and application software, communicate with system operators or other users on the system, request system resources and operator services as needed, and gather information concerning job and system performance.

Three of the CUSPs which extend or supplement the TOPS-10 general purpose command set are described below. Note that the CUSP command set is selected for use via one of the general purpose commands, usually GET or RUN (e.g., RUN TECO<CR>).

The Text Editor and COrrector (TECO) supports commands which enable the user to build and edit an ASCII text file. Later, this file may be transformed into a usable program via an assembler or compiler-type CUSP.

The Dynamic Debugging Technique (DDT) supports a command set which allows the user to test and debug his program on-line before putting it into operation.

The Peripheral Interchange Program (PIP) supports commands which enable a user to copy or transfer files between standard peripheral devices.

For field maintenance personnel, command sets are the simplest and most important aspect of system software. Some skill and proficiency in using system software is essential to field maintenance personnel because system software must be used to maintain on-line file storage areas, run on-line (user mode) utility and diagnostic programs, and compile and print system error logs.

System Software Command Format
Operating systems and system library programs use a command format similar to the one illustrated in Figure 4.

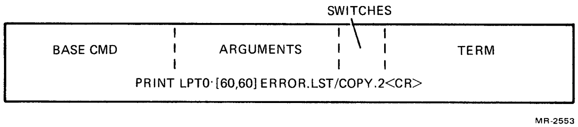


Figure 4   Typical System Software Command Format

BASE CMD - The base command is usually a verb which describes the task that the command will accomplish (e.g., GET, RUN, PRINT,etc.).

ARGUMENTS - The arguments specify the base command parameters. For example, the arguments supplied to the PRINT command illustrated in Figure 4 specify LPT0: as the output device, [60,60] as the project programmer numbers, and ERROR.LST as the file to be printed.

SWITCHES - Switches cause a minor modification to the basic action of the command. For example, the COPY:2 switch illustrated in Figure 4 will cause two copies of the file ERROR.LST to be printed instead of one, which is the default. For example, DIRECT [60,60]/FAST<CR>. The FAST switch associated with the DIRECTORY command will cause an abbreviated form of the directory area to be printed.

TERM - The command terminator, usually a carriage return <CR>, line feed <LF> or altmode <$>, directs the operating system or CUSP to execute the command. As a result of executing the command illustrated by Figure 4, line printer 0 will print two copies of the file ERROR.LST, which is stored in the [60,60] project programmer area of the default input device (in this case the system disk).

Although some system software commands do not require all of the command elements described above, and some will prompt for missing arguments, the basic format (BASE CMD ARGUMENTS SWITCHES TERM) will generally remain the same for all system software. Thus, learning to use system software is a relatively easy task.

Tips on Learning to Use System Software
The following are some tips you may find helpful when learning to use new system software.

1. Study the file structure and organization used by the operating system. This is important because many system software commands are related to file generation, modification and manipulation.

2. Think of system software in terms of command sets. Do not become overly concerned with the monitoring function.

3. Think of each command individually in terms of what task it will accomplish. Do not become overly concerned with how the command achieves the task.

4. Review the general purpose command set supported by the operating system. Become familiar with the type of commands that are available.

5. Review the abstract and command set associated with each CUSP in the system program library. Determine which CUSPs you are most likely to use on a regular basis.

6. Design some exercises which will help you develop skill and proficiency in using the system software. Remember perfect practice makes perfect.

7. Finally, and most important - DON'T be intimidated by system software. It is designed to be easy to use and there are a lot of people using it that know far less about computers than you do.

RSX-20F

KL10 Maintenance Guide
Volume II

RSX-20F

Replace all text pages following the **RSX-20F** tab with the following sheets (pages 1 through 31).

Discard this sheet after completing directions.

Table of Contents

RSX-20F SYSTEM PROGRAM LIBRARY

The RSX-20F System Program Library consists of four kinds of files.

Microcode files. These files are for the KL10 and are listed and described in Table 1.

Boot files. These files are used by the front-end subsystem to boot the KL10. They are listed and described in Table 2.

Automatic task files. These files are used by RSX-20F for various housekeeping tasks and are not normally loaded by the user. They are listed and described (for reference purposes) in Table 3.

User task files. These files are listed and described in Table 4.

Table 1 RSX-20F System Program Library Microcode Files

Task	Description
KLA.MCB	Microcode file for KL10 model PAs.
KLX.MCB	Microcode file for KL10 model PVs.

Table 2 RSX-20F System Program Library Boot Files

Task	Description
BOOT.EXB	Boot Boots KL10 monitor system image into KL's core from RIGID disk; is written in executable binary KL code.
MTBOOT.EXB	Magtape Boot Allows transfer of a program's core image from magtape into KL10's core; is written in executable binary KL code.

Table 3 RSX-20F System Program Library Auto Tasks

Task	Description
F11ACP.TSK	Files-11 Ancillary Control Processor File handler for front-end disk files (performs file access, management, and control functions).
KLE.TSK	KL Error Error processing of KL10 errors. Uses diagnostic DTE functions. Produces "snapshot" of KL10 error conditions for troubleshooting. Calls KLINIT when done.
KLI.TSK	KL Initialization Initializes the KL10 processor (produces installation dialogue, loads microcode, runs bootstrap, etc.). Called whenever system comes up.
KLR.TSK	KLINIK Request Checks KLINIK time window and KLINIK password when KLINIK line rings. If they are correct, it then enables KLINIK.

Table 3 RSX-20F System Program Library Auto Tasks (Cont)

Task	Description
KLX.TSK	KL Transfer Transfers KLEER.SNP to SYSERR file in KL10. (Not to be confused with KLX.MCB, which is the filename of the KL10-PV microcode.)
MIDNIT.TSK	Midnight Roll over time of day at midnight.
SETSPD.TSK	Set Speed Sets line speed table for -10 after restart and sets the time in the -10. **NOTE** **Do not confuse this with the TOPS-20 program SETSPD.EXE. SETSPD.TSK is a front-end task and it does not access CNFG.CMD.**
TKTN.TSK	Task Termination Program Outputs task termination notification and provides orderly termination for front-end tasks. Interfaces between KLINIT and KLERR (lets KLE call KLI).
T20APC.TSK	TOPS-20 Ancillary Control Processor File handler for files to be transferred to and from the KL10's disk area. Interacts with TOPS-20 area in terms compatible with FILES-11 operations.
UFD.TSK	User File Directory Sets up directories in FILES-11 area. Directories are "named" by a UIC (user identification code) and enclosed in brackets: [X, Y].

Table 4 RSX-20F System Program Library User Tasks

Task	Description
COP.TSK	Copy Floppy disk copy utility. Also allows verification of physical state of the disk, as well as verification of successful copying.
DMO.TSK	Dismount Removes a device from the front-end system's knowledge, making its contents inaccessible to the user.
FEDDT.TSK	Front-End DDT Symbolic debuger for RSX-20F. Permits user to read and print selected portions of front-end crashes.
INT.TSK	Initialize Initializes FILES-11 devices to be recognizable FILES-11 "VOLUMES". Sets up master directory space, index and home blocks, etc.
MOU.TSK	Mount Makes a device known to the system so that it can be accessed by a given user.
PARSER.TSK	Command Parser Primary means of access to front-end programs. Provides access to KL10's memory for diagnostic functions, as well as debugging tools. Will interface with KLINIK in future versions.
PIP.TSK	Peripheral Interchange Program Performs general file transfer and some maintenance functions among FILES-11 devices and other peripherals (e.g., floppy-to-disk file transfers, file deletions, typing directories at console, etc.).
RED.TSK	Redirect Changes front-end system's "home" from one FILES-11 device to another, and tells system where it resides presently.
SAV.TSK	Save Saves core image of front-end on RIGID disk in FILES-11 area.
ZAP.TSK	Zap Permits direct examination and modification of files on a FILES-11 volume. Patch task images and data files in an interactive environment.

GENERAL INFORMATION
The command PARSER runs as a task under the RSX-20F executive. Its primary function is to receive ASCII command strings, usually from the console terminal, and perform console functions on the KL10 or PDP-11 computer.

^\	Control Backslash - Command to RSX-20F to load and run PARSER
PAR>	Prompt - Indicates PARSER is ready to accept commands, and the KL10 clock and run flip-flop are on
PAR%	Prompt - Indicates PARSER is ready to accept commands, the KL10 run flip-flop is off, and the KL10 clock is on
PAR#	Prompt - Indicates PARSER is ready to accept commands, and the KL10 clock is off. This may indicate an error condition
QUIT or ^Z or SET CON/USER	Exit PARSER - Return to RSX-20F command mode. The CTY is connected to the program running in the KL10

Note

1. Commands and arguments may be abbreviated to the simplest form that uniquely identifies them; e.g., the EXAMINE command may be typed as E since no other commands begin with E.

2. The maximum number of characters in a command line is 280.

3. Numeric arguments default to decimal unless they are address or data arguments. Then they default to octal.

COMMAND CONVENTIONS
The command conventions and special characters used by PARSER are described in Table 1.

COMMAND SUMMARY
The command PARSER has four modes of operation. The mode is set by the SET CONSOLE command.

Maintenance Mode - Enables the commands described in Table 2.

User Mode - Connects the console to the program running in the KL10. No PARSER commands are in effect.

For a description of the commands listed in Table 3 and Table 4, refer to Table 2.

Table 1 Command PARSER Special Characters

Character	Meaning
?	PAR>?<CR> or PAR>SET?<CR> A question mark typed at PARSER command, subcommand, or argument level will cause a brief help message to be displayed.
;	PAR>E PC;E 20;SH<CR> Used to separate individual commands within a command line.
!	PAR>REP 5;E PC! SEE IF CPU IS IN HALT LOOP<CR> Indicates a comment line.
<CR>	PAR>SH<CR> Command line terminator - causes the command line to be executed.
-<CR>	PAR>ST M0-<CR> Nullifies the <CR> terminator - allows the command line to be continued on the next line. The continuation line will prompt with another dash.

Table 1 Command PARSER Special Characters (Cont)

Character	Meaning
^C	PAR<DE T 100:^C5<CR> Digits preceded by an up arrow and a C are interpreted as 1's complement.
^D	PAR>DE E 200:^D5<CR> Digits preceded by an up arrow and a D are interpreted as decimal.
^B	PAR>DE T 200:^B1010<CR> Digits preceded by an up arrow and a B are interpreted as binary.
^O	PAR>DE T 200:^O5252<CR> Digits preceded by an up arrow and an O are interpreted as octal (default).
^O	A control O can also be used to suppress printouts.
^Z	A control Z causes PARSER to exit. The console is connected to the program running in the KL10.
'	PAR>E E 34'<CR> A single quote adds the current value of the relocation switch to the number. See SET OFFSET.
"	PAR>E E 34"CR> A double quote subtracts the current value of the relocation switch from the number. See SET OFFSET.
-	PAR>DE T 30:-1<CR> A string of digits preceded by a hyphen (minus sign) is interpreted as the 2's complement of the value of the string.
+-*/	Two numeric expressions separated by plus, minus, asterisk, or slash are evaluated by applying the operations of addition, subtraction, multiplication or division, respectively.
_	Two numeric expressions separated by underscore are evaluated by shifting the first left by the second. Example: 1_3 is 10 octal.
(2*8)/4	Parentheses may be used to enclose expressions. Thus parentheses can be used to change the implicit order of arithmetic operations.

Table 2 PARSER Maintenance Mode Command Summary

Command	Description	Cross Ref.
ABORT	PAR>A<CR> Force the KL10 into the HALT loop. See HALT.	1
CLEAR	PAR>CL arg<CR> The CLEAR command accepts the following arguments. See SET commands. CLOCK e.g., PAR>CL CL CON<CR> The CLEAR CLOCK command accepts the following arguments. CONTROL e.g., PAR>CL CL CON<CR> Disable the control logic clock. CRAM e.g., PAR>CL CL CR<CR> Disable the CRAM clock. DATA-PATH e.g., PAR>CL CL D<CR> Disable the data path clock. EXTERNAL e.g., PAR>CL CL E<CR> Select the internal KL10 clock source. Same as SET CLOCK INTERNAL.	

Table 2 PARSER Maintenance Mode Command Summary (Cont)

Command	Description	Cross Ref.
	FULL e.g., PAR>CL CL F<CR> Set the KL10 clock rate to full speed. Same as SET CLOCK FULL.	
	HALF e.g., PAR>CL CL H<CR> Set the KL10 clock rate to full speed. Same as SET CLOCK FULL.	
	INTERNAL e.g., PAR>CL CL I<CR> Select the internal KL10 clock source. Same as SET CLOCK INTERNAL.	
	MARGIN e.g., PAR>CL CL M<CR> Select the internal KL10 clock source. Same as SET CLOCK INTERNAL.	
	NORMAL e.g., PAR>CL CL N<CR> Set the KL10 clock parameters to internal source and full rate with the CRAM, DATA-PATH and CONTROL clocks enabled.	
	QUARTER e.g., PAR>CL CL Q<CR> Set the KL10 clock rate to full speed. Same as SET CLOCK FULL.	
	SLOW e.g., PAR>CL CL S<CR> Set the KL10 clock rate to full speed. Same as SET CLOCK FULL.	
	CONSOLE e.g., PAR>CL C<CR> Put the console front end into operator mode. Equivalent to SET CONSOLE OPERATOR.	
	DATE e.g., PAR>CL D<CR> Clear the date validity bit and prompt for a new date and time. This command is invalid if RSX-20F is in primary protocol; i.e., if the public structure (PS) is mounted. See SET DATE.	
	FS-STOP e.g., PAR>CL FS<CR> Disable the field service clock error stop feature. Same as CLEAR PARITY-STOP FS-STOP.	
	INCREMENT e.g., PAR>CL I<CR> Set the KL10 increment factor to 0. See SET INCREMENT.	
	KLINIK e.g., PAR>CL K<CR> Clear KLINIK parameters (only).	16
	MEMORY e.g., PAR>CL M<CR> Make KL10 memory the default for deposits and examines. Not to be confused with zeroing memory. See SET MEMORY and ZERO.	
	NOT e.g., PAR>CL NO REL<CR> Used with CLEAR to negate the clear function. It is equivalent to SET.	
	OFFSET e.g., PAR>CL O<CR> Set the value of the PDP-11 relocation counter to 0. See SET OFFSET.	

Table 2   PARSER Maintenance Mode Command Summary (Cont)

Command	Description	Cross Ref.
	PARITY-STOP e.g., PAR>CL P ALL<CR> The CLEAR PARITY-STOP command accepts the following arguments.	
	ALL e.g., PAR>CL P ALL<CR> Disable all parity stop features.	
	AR e.g., PAR>CL P AR<CR> Disable the AR and ARX parity stop features.	
	CRAM e.g., PAR>CL P C<CR> Disable the CRAM parity stop feature.	
	DRAM e.g., PAR>CL P D<CR> Disable the DRAM parity stop feature.	
	ENABLE e.g., PAR>CL P E<CR> Clear all parity stop enables. Same as CLEAR PARITY-STOP ALL<CR>	
	FM e.g., PAR>CL P FM<CR> Disable the fast memory (FM) parity stop feature.	
	FS-STOP e.g., PAR>CL P FS<CR> Disable the field service clock error feature. Same as CLEAR FS-STOP.	
	RELOAD e.g., PAR>CL REL<CR> Disable the automatic reloading of the KL10 following a fatal error condition.	
	REPEAT e.g., PAR>CL REP<CR> Set the repeat counter to 0. All subsequent command lines will be repeated once. See SET REPEAT.	
	RETRY e.g., PAR>CL RET<CR> Clear the PARSER RETRY flag. Every KEEP-ALIVE-CEASED error will cause a KLERR snapshot before reloading the KL10.	
	TRACKS e.g., PAR>CL T<CR> Clear the KL10 tracking function. See SET TRACKS.	10
CONTINUE	PAR>CO<CR> Continue the KL10 running if it is continuable (i.e., the KL10 has not been reset). See START.	2
DEPOSIT	PAR>DE T N:500<CR> The DEPOSIT command accepts the following arguments. Default: see SET MEMORY. The previous contents of the location or argument specified will be displayed.	
	AR e.g., PAR>DE A:777777777777<CR> Load data (777777777777) into the AR.	
	ELEVEN e.g., PAR>DE E 2000:500<CR> Deposit data (500) into PDP-11 location specified (2000).	
	DEPOSIT ELEVEN accepts the following arguments. Default: THIS.	
	DECREMENT e.g., PAR>DE E D:500<CR> Deposit data (500) into the last PDP-11 location referenced minus two (-2).	
	INCREMENT e.g., PAR>DE E I:500<CR> Deposit data (500) into the last PDP-11 location referenced plus two (+2).	

Table 2 PARSER Maintenance Mode Command Summary (Cont)

Command	Description	Cross Ref.
	NEXT e.g., PAR>DE E N:500<CR> Same as DE E I:500<CR> (INCREMENT)	
	PREVIOUS e.g., PAR>DE E P:500<CR> Same as DE E D:500<CR> (DECREMENT)	
	THIS e.g., PAR>DE E T:500<CR> Deposit data (500) into the last PDP-11 location referenced. THIS is the default.	
	TEN e.g., PAR>DE T 30000:500<CR> Deposit data (500) into PDP-10 location specified (30000). All references are to a physical address. Paged (user) deposits are not supported by PARSER. DEPOSIT TEN accepts the following arguments. Default: THIS	
	DECREMENT e.g., PAR>DE T D:500<CR> Deposit data (500) into the last PDP-10 location referenced minus the increment value. See SET INCREMENT.	
	INCREMENT e.g., PAR>DE T I:500<CR> Deposit data (500) into the last PDP-10 location referenced plus the increment value. See SET INCREMENT.	
	NEXT e.g., PAR>DE T N:500<CR> Deposit data (500) into the last PDP-10 location referenced plus one (+1).	
	PREVIOUS e.g., PAR>DE T P:500<CR> Deposit data (500) into the last PDP-10 location referenced minus one (-1)	
	THIS e.g., PAR>DE T T:500<CR> Deposit data (500) into the last PDP-10 location referenced. THIS is the default.	
DISCONNECT	PAR>DI<CR> Disconnect the KLINIK link by running KLDISC.TSK. The existing KLINIK parameters are not affected. See CLEAR KLINIK.	
EXAMINE	PAR>EX T 3000<CR. The EXAMINE command accepts the following arguments. Default: see SET MEMORY.	
	ELEVEN e.g., PAR>EX EL 3000<CR> Display the contents of the PDP-11 location specified (3000). EXAMINE ELEVEN accepts the following arguments. Default: THIS.	
	DECREMENT e.g., PAR>EX EL D<CR> Display the contents of the last PDP-11 location referenced minus two (-2).	
	INCREMENT e.g., PAR>EX EL I<CR> Display the contents of the last PDP-11 location referenced plus two (+2).	
	NEXT e.g., PAR>EX EL N<CR> Same as EX EL I<CR> (INCREMENT)	
	PREVIOUS e.g., PAR>EX EL P<CR> Same as EX E D<CR> (DECREMENT)	

Table 2 PARSER Maintenance Mode Command Summary (Cont)

Command	Description	Cross Ref.
	THIS e.g., PAR>EX EL T<CR> Display the contents of the last PDP-11 location referenced. THIS is the default.	
	TEN e.g., PAR>EX T 30000<CR> Display the contents of the PDP-10 location specified (30000). All references are to a physical address. Paged (user) examines are not supported by PARSER. EXAMINE TEN accepts the following arguments. Default: THIS.	
	DECREMENT e.g., PAR>EX T D<CR> Display the contents of the last PDP-10 location referenced minus the increment value. See SET INCREMENT.	
	INCREMENT e.g., PAR>EX T I<CR> Display the contents of the last PDP-10 location referenced plus the increment value. See SET INCREMENT.	
	NEXT e.g., PAR>EX T N<CR> Display the contents of the last PDP-10 location referenced plus one (+1).	
	PREVIOUS e.g., PAR>EX T P<CR> Display the contents of the last PDP-10 location referenced minus one (-1).	
	THIS e.g., PAR>EX T T<CR> Display the contents of the last PDP-10 location referenced. THIS is the default.	
	AB e.g., PAR>EX AB<CR> Display the contents of the Address Break register.	
	AD e.g., PAR>EX AD<CR> Display the state of the ADder.	
	ADX e.g., PAR>EX ADX<CR> Display the state of the ADder Extended	
	AR e.g., PAR>EX AR<CR> Display the contents of the Arithmetic Register.	
	ARX e.g., PAR>EX ARX<CR> Display the contents of the Arithmetic Register eXtended.	
	BR e.g., PAR>EX BR<CR> Display the contents of the Buffer Register.	
	BRX e.g., PAR>EX BRX<CR> Display the contents of the Buffer Register eXtended.	
	CRADDR e.g., PAR>EX CRADDR<CR> Display the contents of the Cram ADDRess register.	
	CRLOC e.g., PAR>EX CRLOC<CR> Display the contents of the CRAM LOCation register.	

Table 2 PARSER Maintenance Mode Command Summary (Cont)

Command	Description	Cross Ref.
	DRADDR e.g., PAR>EX DRADDR<CR> Display the contents of the DRAM ADDRess register.	
	DTE-20 e.g., PAR>EX DTE<CR> Display the contents of the three DIAG registers and the status register in the DTE20.	
	EBUS e.g., PAR>EX EBUS<CR> Display the contents of the EBus.	
	FE e.g., PAR>EX FE<CR> Display the contents of the Floating Exponent register.	
	FLAGS e.g., PAR>EX FLAGS<CR> Display the state of the flag bits (00-12) in the left half of the PC: OVF, CY0, CY1, FOV, BIS, USR, UIO, LIP, AFI, AT1, AT0, FUF and NOV.	
	FM e.g., PAR>EX FM<CR> Display the contents of the Fast Memory register.	
	KL e.g., PAR>EX KL<CR> Perform, in order, an EX PC, EX VMA, EX PI and EX FLAGS.	
	MQ e.g., PAR>EX MQ<CR> Display the contents of the Multiplier Quotient register.	
	PC e.g., PAR>EX PC<CR> Display the contents of Program Counter.	
	PI e.g., PAR>EX PI<CR> Display the state of the Priority Interrupt system.	
	REGISTERS e.g., PAR>EX REG<CR> Display the contents of the following registers: AD, ADX, AR, ARX, BR, BRX, EBUS, FM, MQ, and PC.	
	SBR e.g., PAR>EX SBR<CR> Display the contents of the Subroutine Return register.	
	SC e.g., PAR>EX SC<CR> Display the contents of the Shift Count register.	
	VMA e.g., PAR>EX VMA<CR> Display the contents of the Virtual Memory Address register.	
	VMAH e.g., PAR>EX VMAH<CR> Display the contents of the Virtual Memory Address Held register.	
FREAD	PAR>FR 110<CR> Display the result of a diagnostic function read using the function code specified (110). The function code must be in the range of 100 to 177.	
FWRITE	PAR>FW 77:252525777777<CR> Perform a diagnostic function write using the function code (77) and data (252525777777) specified. The function code must be in the range of 40 to 77.	

Table 2 PARSER Maintenance Mode Command Summary (Cont)

Command	Description	Cross Ref.
FXCT	PAR>FX 0<CR> Perform a diagnostic function execute using the function code specified (0). The function code must be in the range of 00 to 37.	4
HALT	PAR>H<CR> Halt the KL10. See ABORT and SHUTDOWN.	5
INITIALIZE	PAR>I<CR> Check the state of the KL10 clock, run flip-flop and opcode enable.	6
JUMP	PAR>J 30000<CR> Start the KL10 at the address specified (30000) and exit. The address is in the executive space and the processor mode is not affected. See START TEN.	
MCR	PAR>M BOOT<CR> Load and start the specified task file (BOOT.TSK). Same as RUN.	
QUIT	PAR>Q<CR> Exit from PARSER. Same as SET CONSOLE USER<CR> or ^Z.	
REPEAT n	PAR>REP 2;EX T N<CR> Cause the command(s) in the remainder of the line to be repeated n(2) times.	7
RESET	PAR>RES ALL<CR> The RESET command accepts the following arguments. Default: <CR>.	
	<CR> e.g., PAR>RES<CR> Cause a master reset of the KL10. The state of the clock enables and parity stops are not affected. This is the default.	
	ALL e.g., PAR>RES AL<CR> Perform a RES APR, RES DTE-20, RES PAG and RES PI command. The KL10 must be halted.	
	APR e.g., PAR>RES AP<CR> Execute a CONO APR,267760. The KL10 must be halted.	
	DTE-20 e.g., PAR>RES D<CR> Reset the DTE20.	8
	ERROR e.g., PAR>RES E<CR> Execute a CONO APR,27760 clearing the error flags in the Arithmetic Process Register (APR).	
	INITIALIZE e.g., PAR>RES IN<CR> Perform a KL10 master reset and return clock enables and parity stops to their default. The KL10 must be halted.	
	IO e.g., PAR>RES IO<CR> Execute a CONO APR,200000 which causes an I/O reset.	
	PAGE e.g., PAR>RES PAG<CR> Execute a CONO PAG,0 followed by a DATAO PAG,X (where the contents of X = 100). This will reset the KL10 paging box.	
	PI e.g., PAR>RES PI<CR> Execute a CONO PI,10000 which resets the Priority Interrupt system.	

Table 2 PARSER Maintenance Mode Command Summary (Cont)

Command	Description	Cross Ref.
RUN	PAR>RU PIP<CR> Load and run the specified task file (PIP.TSK). Same as MCR.	
SET	PAR>SET MEM TEN<CR> The SET command accepts the following arguments.	
	CLOCK e.g., PAR>SET CL N<CR> The SET CLOCK command accepts the following arguments.	
	CONTROL e.g., PAR>SET CL CON<CR> Enable the control logic clock.	
	CRAM e.g., PAR>SET CL CR<CR> Enable the CRAM clock.	
	DATA-PATH e.g., PAR>SET CL D<CR> Enable the data path clock.	
	EXTERNAL e.g., PAR>SET CL E<CR> Set (select) the KL10 external clock source. PARSER will request confirmation.	9
	HALF e.g., PAR>SET CL H<CR> Set the KL10 clock rate to one half of the standard (divide by 2).	
	INTERNAL e.g., PAR>SET CL I<CR> Set (select) the KL10 internal clock source.	
	MARGIN e.g., PAR>SET CL M<CR> Set (select) KL10 clock margins.	
	NORMAL e.g., PAR>SET CL N<CR> Set the KL10 clock rate to the standard (internal source, full rate with CRAM, data-path and control logic clocks enabled).	
	QUARTER e.g., PAR>SET CL Q<CR> Set the KL10 clock rate to one quarter of the standard (divide by 4).	
	SLOW e.g., PAR>SET CL S<CR> Set the KL10 clock rate to one eighth of the standard (divide by 8).	
	CONSOLE e.g., PAR>SET CON M<CR> The SET CONSOLE command accepts the following arguments.	
	MAINTENANCE e.g., PAR>SET CON M<CR> Set the console to maintenance mode. The command set is unrestricted. Refer to Table 2.	
	OPERATOR e.g., PAR>SET CON O<CR> Set the console to operator mode. The command set is restricted to those listed in Table 3.	
	PROGRAMMER e.g., PAR>SET CON P<CR> Set the console to programmer mode. The command set is restricted to those listed in Table 4.	
	USER e.g., PAR>SET CON U<CR> Exit PARSER. Leave the CTY connected to the program running in the KL10.	

Table 2 PARSER Maintenance Mode Command Summary (Cont)

Command	Description	Cross Ref.
	DATE e.g., PAR>SET D<CR> Set the date and time to be used by the front-end executive, RSX-20F. This command is illegal if RSX-20F already has a valid date from a previous SET DATE command or a reload of the KL10.	
	FS-STOP e.g., PAR>SET F<CR> Enable the Field Service Clock Error Stop feature in the KL10. This requires backplane jumper wires to be meaningful. Same as SET PARITY-STOP FS-STOP.	
	INCREMENT e.g., PAR>SET I 10<CR> Set the increment and decrement value for KL10 deposit and examine commands to the value specified (10).	
	KLINIK e.g., PAR>SET K<CR> Set the KLINIK link for remote console operation.	15
	MEMORY e.g., PAR>SET M T<CR> The SET MEMORY command accepts the following arguments.	
	ELEVEN e.g., PAR>SET M E<CR> Set the PDP-11 as the default memory for deposits and examines.	
	TEN e.g., PAR>SET M T<CR> Set the KL10 as the default memory for deposits and examines.	
	NOT e.g., PAR>SET NO RELOAD<CR> Used with SET to negate the SET function. It is equivalent to CLEAR.	
	OFFSET e.g., PAR>SET O 101204<CR> Set the PDP-11 relocation counter to the value specified (101204). The relocation counter is initially set to the address of the PARSER root overlay.	
	PARITY-STOP e.g., PAR>SET P ALL<CR> The SET PARITY-STOP command accepts the following arguments.	
	ALL e.g., PAR>SET P ALL<CR> Set the parity stop enable to on and enable the following parity stop features. AR, CRAM, DRAM, FM and FS-STOP.	
	AR e.g., PAR>SET P AR<CR> Add stop on AR and ARX parity error to the parity stop features.	
	CRAM e.g., PAR>SET P C<CR> Add stop on CRAM parity error to the parity stop conditions.	
	DRAM e.g., PAR>SET P D<CR> Add stop on DRAM parity error to the parity stop conditions.	
	ENABLE e.g., PAR>SET P E<CR> Enable (turn on) the selected PARITY-STOP features.	
	FM e.g., PAR>SET P FM<CR> Add stop on a fast memory (FM) parity error to the parity stop conditions.	
	FS-STOP e.g., PAR>SET P FS<CR> Enable the Field Service Clock Error Stop feature in the KL10. This requires backplane jumper wires. Same as SET FS-STOP.	

Table 2 PARSER Maintenance Mode Command Summary (Cont)

Command	Description	Cross Ref.
	RELOAD e.g., PAR>SET REL<CR> Enable the automatic reload of the KL10 by the PDP-11 front end. This is the default. See CLEAR RELOAD.	
	REPEAT e.g., PAR>SET REP 5<CR> Set the repeat counter to the decimal value specified. All subsequent command lines will be repeated that number of times. The value will also be used as a multiplier by the REPEAT command.	7
	RETRY e.g., PAR>SET RET<CR> Set the PARSER RETRY flag. See CLEAR RETRY.	17
	TRACKS e.g., PAR>SET T<CR> Display all FR, FW, FX, Examine, Deposit, and DTE-20 operations.	10
SHUTDOWN	PAR>SH<CR> Gracefully shut down the TOPS-10 or TOPS-20 operating system. This is done by depositing a minus 1 in location 30. Timesharing ceases.	11
START	PAR>ST M 0<CR> or PAR>ST T 2000<CR> The START command accepts the following arguments. START with no arguments or an argument of 0 is illegal. If neither TEN nor MICROCODE is specified, TEN is assumed.	
	MICROCODE e.g., PAR>ST M 0<CR> Start the microcode at the address specified (0).	12
	TEN e.g., PAR>ST T 3000<CR> Start the KL10 at the address specified (3000). See CONTINUE and JUMP.	13
WHAT	PAR>W CL<CR> The WHAT command accepts the following arguments.	
	CLOCK e.g., PAR>W CL<CR> Display the current clock state. See SET CLOCK.	
	CONSOLE e.g., PAR>W CON<CR> Display the current console mode. See SET CONSOLE.	
	DATE e.g., PAR>W D<CR> Display the state of the validity flag and the current date and time held by RSX-20F.	
	INCREMENT e.g., PAR>W I<CR> Display the current increment/decrement value. See SET INCREMENT.	
	KLINIK e.g., PAR>W K<CR> Display the current status of the KLINIK link. See SET KLINIK.	15
	MEMORY e.g., PAR>W M<CR> Display the current default memory. See SET MEMORY.	
	OFFSET e.g., PAR>W O<CR> Display the current value of the PDP-11 relocation counter. See SET OFFSET.	

Table 2 PARSER Maintenance Mode Command Summary (Cont)

Command	Description	Cross Ref.
	PARITY-STOP e.g., PAR>W P<CR> Display the current state of the parity stop feature. See SET PARITY-STOP.	
	RELOAD e.g., PAR>W REL Display the current state of the KL10 automatic reload feature (ON or OFF). See SET RELOAD.	
	REPEAT e.g., PAR>W REP<CR> Display the current value of the repeat counter. See SET REPEAT.	
	RETRY e.g., PAR>W RET<CR> Display the state of the PARSER RETRY flag. See SET RETRY.	17
	TRACKS e.g., PAR>W T<CR> Display the current state of the trace enable feature (ON or OFF). See SET TRACKS.	10
	VERSION e.g., PAR>W V<CR> Display the current version of PARSER and RSX-20F.	
XCT	PAR>X 254200000000<CR> Execute the argument (245200000000) as a PDP-10 instruction. The KL10 must be in executive mode.	14
ZERO	PAR>Z 200>277<CR> Zero PDP-10 physical memory from first argument (200) through second argument (277). Note: depending on the amount of memory this may take a while.	

Table 3 PARSER Operator Mode Command Summary

Command	Description
ABORT	PAR>A<CR>
CLEAR	PAR>CL C<CR> or PAR>CL R<CR> etc. The CLEAR command accepts the following arguments. CONSOLE KLINIK NOT INCREMENT MEMORY REPEAT
DISCONNECT	PAR>DI<CR>
EXAMINE	KL e.g., PAR>EX KL<CR> PC e.g., PAR> EX PC<CR> ELEVEN e.g., PAR>EX EL adr<CR> DECREMENT e.g., PAR>EX EL D<CR> INCREMENT e.g., PAR>EX EL I<CR> NEXT e.g., PAR> EX EL N<CR> PREVIOUS e.g., PAR> EX EL P<CR> THIS e.g., PAR> EX EL T<CR> TEN e.g., PHR>EX T adr<CR> DECREMENT e.g., PAR>EX T D<CR> INCREMENT e.g., PAR>EX T I<CR> NEXT e.g., PAR>EX T N<CR> PREVIOUS e.g., PAR>EX T P<CR> THIS e.g., PAR>EX T T<CR>

Table 3 PARSER Operator Mode Command Summary (Cont)

Command	Description
JUMP	PAR>J 30000<CR>
MCR	PAR>MCR BOOT<CR>
QUIT	PAR>Q<CR>
REPEAT	PAR>REP 2:EX T N<CR>
RUN	RU PIP<CR>
SET	CONSOLE e.g., PAR>SET CON M<CR>
	The SET console command accepts the following four arguments: USER, OPERATOR, PROGRAMMER and MAINTENANCE.
	INCREMENT e.g., PAR>SET I 10<CR>
	KLINIK e.g., PAR>SET K<CR>
	MEMORY e.g., PAR>SET M E<CR> or PAR>SET M T<CR>

Table 4 PARSER Programmer Mode Command Summary

Command	Description
ABORT	PAR>A<CR>
CLEAR	PAR>CL C<CR> or PAR>CL T<CR> etc.
	The CLEAR command accepts the following arguments.
	CONSOLE, DATE, INCREMENT, KLINIK, MEMORY, NOT, OFFSET, RELOAD, REPEEAT, RETRY, TRACKS
CONTINUE	PAR>CO<CR>
DEPOSIT	AR e.g., PAR>DE A:data<CR>
	ELEVEN e.g., PAR>DE E adr: data<CR>
	DECREMENT e.g., PAR>DE E D:data<CR>
	INCREMENT e.g., PAR>DE E I:data<CR>
	NEXT e.g., PAR>DE E N:data<CR>
	PREVIOUS e.g., PAR>DE E P:data<CR>
	THIS e.g., PAR>DE E T:data<CR>
	TEN e.g., PAR>DET adr:data<CR>
	DECREMENT e.g., PAR>DE T D:data<CR>
	INCREMENT e.g., PAR>DE T I:data<CR>
	NEXT e.g., PAR>DE T N:data<CR>
	PREVIOUS e.g., PAR>DE T P:data<CR>
	THIS e.g., PAR>DE T T:data<CR>
DISCONNECT	PAR>DI<CR>

Table 4 PARSER Programmer Mode Command Summary (Cont)

Command	Description
EXAMINE	PAR>EX AB<CR> or PAR>EX PC<CR> etc. The EXAMINE command accepts any of the following arguments. AB CRLOC MQ AD DRADDR PC ADX DTE-20 PI AR EBUS REGISTERS ARX FE SBR BR FLAGS SC BRX FM VMA CRADDR KL VMAH ELEVEN e.g., PAR>EX EL adr<CR> DECREMENT e.g., PAR>EX EL D<CR> INCREMENT e.g., PAR>EX EL I<CR> NEXT e.g., PAR>EX EL N<CR> PREVIOUS e.g., PAR>EX EL P<CR> THIS e.g., PAR>EX EL T<CR> TEN e.g., PAR>EX T adr<CR> DECREMENT e.g., PAR>EX T D<CR> INCREMENT e.g., PAR>EX T I<CR> NEXT e.g., PAR>EX T N<CR> PREVIOUS e.g., PAR>EX T P<CR> THIS e.g., PAR>EX T T<CR>
HALT	PAR>H<CR>
INITIALIZE	PAR>I<CR>
JUMP	PAR>J 30000<CR>
MCR	PAR>MCR BOOT<CR>
QUIT	PAR>Q<CR>
REPEAT	PAR>REP 2;EX T N<CR>
RESET	PAR>RES ALL<CR> or PAR>PAG<CR> etc The RESET command accepts the following arguments. ALL ERROR PAG APR INITIALIZE PI DTE-20 I/O
RUN	PAR>RU PIP<CR>
SET	CONSOLE e.g., PAR>SET CON M<CR> The SET CONSOLE command accepts four arguments; USER, OPERATOR, PROGRAMMER and MAINTENANCE. DATE e.g., PAR>SET D<CR> INCREMENT e.g., PAR>SET I 10<CR> KLINIK e.g., PAR>SET K<CR> MEMORY e.g., PAR>SET M E<CR> or PAR>SET M T<CR> The SET MEMORY command accepts two arguments: ELEVEN and TEN. NOT e.g., PAR>SET NO arg<CR> OFFSET e.g., PAR>SET O 101204<CR>

Table 4 PARSER Programmer Mode Command Summary (Cont)

Command	Description
	RELOAD e.g., PAR>SET REL<CR>
	REPEAT e.g., PAR>SET REP 5<CR>
	RETRY e.g., PAR>SET RET<CR>
	TRACKS e.g., PAR>SET T<CR>
SHUTDOWN	PAR>SH<CR>
START	PAR>ST M<CR> or PAR>ST T 3000<CR>
	The START command accepts two arguments: MICROCODE and TEN.
WHAT	PAR>W CL<CR> or PAR>W V<CR> etc.
	The WHAT command accepts the following arguments.
	CLOCK MEMORY RETRY
	CONSOLE OFFSET TRACKS
	DATE PARITY-STOP VERSION
	INCREMENT RELOAD
	KLINIK REPEAT
XCT	PAR>X 254200000000<CR>
ZERO	PARZ 200>277<CR>

COMMAND DESCRIPTION
This section describes in detail the commands listed in Table 2.

1 A<CR> - The ABORT command stops the KL10 by trying to force it into the HALT loop. If this fails after a reasonable number of EBox clock ticks, the command tries to START MICROCODE at CRAM address 0, which implies a master reset of the KL10 processor.

NOTE
This is the best way to get the KL10 into a known state when the previous state left it hung.

2 CO<CR> - The CONTINUE command takes the KL10 out of the HALT loop, causing it to execute the instruction pointed to by the PC. If single instruction mode was not set, the KL10 should continue running. If single instruction mode was set via the FXCT 12 function, the instruction is executed, and the KL10 is returned to the HALT loop.

3 FLAGS<CR> - The PC flag mnemonics displayed are defined as follows.

AFI - Address Failure Inhibit (bit 08)
ATO - Trap 1 (bit 10)
AT1 - Trap 2 (bit 09)
BIS - First Part Done (bit 04)
CY0 - Carry 0 (bit 01)
CY1 - Carry 1 (bit 02)
FOV - Floating Overflow (bit 03)
FUF - Floating Underflow (bit 11)
LIP - Public (bit 07)
NDV - No Divide (bit 12)
OVF - Overflow/Previous Context Public (bit 00)
UIO - User In-Out/Previous Context User (bit 00)
USR - User (bit 05)

4 FX<CR> - The FXCT command accepts a number as a function write code, performs the function write, and displays the result. Useful values are 0 (stops the KL10 clock), and 1 (starts the KL10 clock). Random use of FXCT can cause false CRAM parity errors. (Use the HALT or ABORT commands first.)

5 H<CR> - The HALT command tries to put the KL10 into the HALT loop by clearing RUN, and waiting. If the KL10 is unable to go into the HALT loop, the HALT command tries to force it in by using BURST mode. If this does not work, an error message is displayed.

6 I<CR> - The INITIALIZE command (re)initializes PARSER, and checks the state of the KL10, sets up the KL10 state flag word with default values and restarts the KL10 based on those values. The following KL10 conditions are checked: clock running, run flip-flop set, and opcode enabled. INITIALIZE also checks to see if this PDP-11 is running on a privileged DTE20.

7 REP 2:EX T P<CR> - The REPEAT n command causes the command(s) in the remainder of the line to be repeated n (2) times if the SET REPEAT value is set to 1. See SET REPEAT. If the SET REPEAT value is greater than 1 then it is multiplied by the REPEAT n value and the commands are repeated that many times.

8 RES D<CR> - The RESET DTE-20 command resets the DTE20 by depositing a 1 in bit 6 of DIAG WORD 2 in the DTE20. Then bit 0 in DIAG WORD 1 of the DTE20 is set to 1 indicating word mode transfers.

9 SET CL E<CR> - The SET CLOCK EXTERNAL command selects the external clock source for the KL10. If no external clock source is connected, the KL10 is stuck and can only be reset by powering the system down and then up again.

10 SET T<CR> - The SET TRACKS command causes changes in the internal state of the KL10 to be displayed after each clock tick. This is done via diagnostic reads and is primarily used for debugging hardware or front-end software. This will result in a lot of wasted paper if you are not careful.

11 SH<CR> - The SHUTDOWN command deposits a -1 (minus one) into KL10 executive virtual location 30 (octal). It is used to gracefully bring down the KL10 timesharing systems. It will cause PARSER to exit if the deposit was successful, which will cause the console terminal to be connected to either EDDT (if loaded), or to the dead KL10. If EDDT is not loaded, the KL10 will execute a HALT instruction (TOPS-20 only) as soon as the clock interrupt is serviced.

12 ST M 0<CR> - The START MICROCODE command performs a KL10 master reset and starts the microcode at the microcode address specified. Starting the MICROCODE at addresses other than 0 is probably not helpful for most users.

13 ST T 30000<CR> - The START TEN command starts the KL10 at the address requested using an algorithm determined by the version of the microcode. It puts the KL10 into the HALT loop, loads the address onto the AR, and does a function CONTINUE, causing the KL10 to start at the address requested in EXEC KERNAL mode. To start the KL10 without losing the old processor mode, use the JUMP command, which will accept an address, EXECUTE a JRST (opcode 254) to that address (in EXEC Virtual Space), and continue in whatever mode the processor was in.

14 X 254200000000<CR> - The XCT command takes a 36-bit octal argument and executes it as a KL10 instruction.

NOTE

Executing an instruction with an opcode of zero may cause random results because the microcode uses op-code zero coming out of the HALT loop for START and CONTINUE.

15 SET KLINIK<CR> - The RSX-20F KLINIK link is enabled by issuing a SET KLINIK command to PARSER from the local console (CTY). PARSER will then request and validate the following parameters.

PARSER will request the KLINIK mode desired with the following prompt.

KLINIK MODE:

The acceptable response to this prompt is either USER or REMOTE.

USER indicates that the KLINIK link is to be used as a timesharing terminal line (only). See SET CONSOLE USER.

REMOTE indicates that the KLINIK link is to be used as a remote console line in either Maintenance, Operator or Programmer mode. See SET CONSOLE.

There is no default response to this prompt. If any other response is supplied, the command will abort and the local operator will receive one of the following error messages:

PAR [SET] NSK NO SUCH KEYWORD "XXX"
PAR [SET] ILC ILLEGAL CHARACTER "C"

where "XXX" and "C" are the offending keyword and character, respectively.

Next PARSER will request the KLINIK ACCESS WINDOW parameters by printing the following prompts and accepting responses in sequence.

ACCESS WINDOW OPEN DATE:
ACCESS WINDOW OPEN TIME:
ACCESS WINDOW CLOSE DATE:
ACCESS WINDOW CLOSE TIME:

The possible date formats are as follows.

DD-MMM-YY
DD-MMM-YYYY
DD MMM YY
DD MMM YYYY

DD is the decimal day, MMM is the alphabetic representation of the month, and YY or YYYY is the decimal year in which the KLINIK WINDOW is to open or close. The default response to a date prompt is a <carriage return>. This will set the Window Open Date to TODAY, and the Window Close Date to TODAY + 1. TODAY is the current date obtained from RSX-20F. See WHAT DATE.

The day specified must be within the range of 1-31. Date for months having less than 31 days will be validated. This includes a special check for February in a leap year. The month MMM is composed of the first three letters of the month to be entered. The year may be specified as either a Gregorian year, 19XX, or as a year relative to 1900, (00 through 99) where the first two digits are assumed to be the first two digits of the current century. Failure to adhere to this syntax will cause the command to abort, and one of the following error messages to be printed.

PAR [SET] DOR DAY OUT OF RANGE - If the day specified does not exist in the month specified.

PAR [SET] NSK NO SUCH KEYWORD "XXX" - If the keyword specified for the month cannot be matched.

PAR [SET] AMB AMBIGUOUS KEYWORD "XXX" - If that keyword is ambiguous. "XXX" is the offending keyword.

PAR [SET] YOR YEAR OUT OF RANGE - If the year has been improperly specified.

PAR [SET] DBT DATE BEFORE TODAY - If the entire window open or close date is prior to TODAY.

The Window Open Time and Window Close Time may be specified in either of the following formats.

HHMM
HH:MM

HHMM is a representation of the hour and minute. In both formats, HH is the hour and must be within the range of 00 to 23, and MM is the minute and must be within the range of 00 to 60. The default response is a <carriage return>. This will set the Window Open Time and the Window Close Time to NOW. NOW is the current time of day obtained from RSX-20F. See WHAT DATE.

Specifying a time which does not conform to this syntax will cause the command to abort and the following error message to be printed.

PAR [SET] TOR TIME OUT OF RANGE

Finally, when the complete specifications for both the Window Open and Window Close times and dates have been specified, the Window Open time and date will be checked to ensure that it does precede the Window Close time and date. If this is not the case, the command will abort and the following error message will be printed.

PAR [SET] KWE KLINIK WINDOW ERROR

If the KLINIK mode specified was USER, the dialogue will terminate at this point, as all necessary parameters have been input. If the specified KLINIK mode was REMOTE, two more parameters will be solicited from the operator. PARSER will first request a password with the following prompt.

PASSWORD:

The local operator must communicate this password to the remote KLINIK user in order that he be allowed access to the KLINIK link.

The password must be at least one and not more than six numeric or uppercase alphabetic characters, with no imbedded or trailing blanks. There are no default responses. The operator's response to this prompt will be echoed on the local console (CTY).

Failure to provide a password in this form will cause the command to abort and one of the following messages to be printed.

PAR [SET] NPI NULL PASSWORD ILLEGAL - If no password was specified.

PAR [SET] PTL PASSWORD TOO LONG - If more than six characters were typed.

PAR [SET] IPC ILLEGAL PASSWORD CHARACTER "C" - If a nonalphanumeric character was typed as a password character. "C" is the offending character.

PARSER will next request that the operator specify the highest PARSER console mode to be allowed while the KLINIK link is active with the following prompt.

HIGHEST CONSOLE MODE:

The acceptable responses to this prompt areas follows (See SET CONSOLE).

MAINTENANCE
OPERATOR
PROGRAMMER

While the KLINIK link is active, PARSER will not allow the remote or the local console to raise the command PARSER console mode, to a level higher than that specified in response to this prompt. There is no default response to this prompt.

Failure to provide the proper response to this prompt will cause the command to abort and the following error message to be printed:

PAR [SET] NSK NO SUCH KEYWORD "XXX"

where "XXX" is the offending keyword.

If all parameters have been properly input and validated, PARSER will return to command level after displaying the KLINIK enable parameters in the following format.

KLINIK [<ACTIVE> <INACTIVE> <DISABLED>]
ACCESS WINDOW OPEN: DD-MMM-YY HH:MM
ACCESS WINDOW CLOSED: DD-MM-YYY HH:MM
KLINIK MODE: [<REMOTE> <USER>]

ACTIVE indicates that the KLINIK user is connected to the RSX-20F KLINIK link.

INACTIVE indicates that the KLINIK parameters have been set, but access has not yet been allowed (i.e., the WINDOW is not open yet).

DISABLED indicates that no KLINIK parameters have been set.

If the KLINIK mode is REMOTE, one additional line will be displayed describing the highest PARSER console mode to be allowed.

CONSOLE MODE LIMIT: [<MAINTENANCE> <OPERATOR> <PROGRAMMER>]

Upon receipt of these parameters RSX-20F will log the SET KLINIK command and the parameters that were accepted. Further, RSX-20F will pass these parameters to the KL10 operating system (TOPS-20 or TOPS-20), to facilitate KLINIK recovery from a PDP-11 reboot.

16 CLEAR KLINIK<CR> - The RSX-20F KLINIK link is disabled via the CLEAR KLINIK command. This command does not accept arguments, it simply clears the KLINIK WINDOW. If the KLINIK link is active, the CLEAR KLINIK command will cause the following message to be printed on both the local and the remote consoles.

KLD KLINIK ACCESS TERMINATED BY OPERATOR

The current KLINIK enable parameters will be reset and passed to the KL10 operating system (TOPS-10 or TOPS-20). The KLINIK ACCESS WINDOW will close and RSX-20F will log the KLINIK mode termination on the CTY. The modem will not be hung up; however, all input from and output to the remote console will be ignored and all subsequent calls made to the KLINIK LINK will be acknowledged and rejected until such time as a new KLINIK WINDOW is set by the local operator. The rejection message will be in the following format.

KLR--KLINIK RING KLINIK-WINDOW CLOSED

This rejection message will appear on both the local and remote consoles.

17 CL RET<CR> - When the RETRY flag is set, the occurrence of a KEEP-ALIVE-CEASED error will result in the execution of the instruction in location 71. The instruction typically branches to a routine that will cause the KL10 operating system (TOPS-10 or TOPS-20) to dump memory and request a reload. If this can not be accomplished before the end of the keep-alive period (5 seconds), then RSX-20F assumes that the KL10 is incapacitated. In this case KLERR is called to take a KL10 hardware snapshot and then reload the KL10.

If the RETRY flag is clear (CLEAR RETRY command) every occurrence of a KEEP-ALIVE-CEASED error will result in a KLERR snapshot and reload of the KL10.

RSX-20F STOP CODES AND I/O ERROR CODES
This appendix contains two lists of error codes. The first list contains RSX-20F stop codes. Associated with each code is the name of the module that issued the stop code, a short explanation of the error, and a possible cause of the error. The second is a list of I/O error codes that are produced by the device handlers and file control primitives. These error codes have associated messages that are listed along with them; however, due to the many different situations in which these errors can arise, no attempt is made to describe recovery algorithms for these errors.

Code	Module	Meaning
BO3	SCOMM	BUFFER OVERFLOW 3 The PDP-11 was not able to obtain the buffer space necessary for data it wanted to send to the KL. Possible Cause: Software error.
BO5	TTYDRR	BUFFER OVERFLOW 5 The Front-End does not have the buffer space to send an XON or an XOFF to a line.
CBR	PF	CROBAR ERROR DTE-20 power did not return after a power-fail restart. RSX-20F allows it 30 seconds to reappear. Possible Cause: Malfunctioning hardware in the KL.
DTB	QPRDTE	TO-11 DTE TRANSFER FAILURE A TO-11-done interrupt has occurred, but the TO-11 address in the DTE TO11AD register (register 22) did not have the expected value. Since TO11AD is incremented for each byte transferred, it should point to the first word following the buffer into which the TO-11 data was written. Possible Cause: The PDP-11 received the wrong byte count or, more likely, the DTE has a hardware malfunction. TO11BC contains the negative count of data that was actually transferred. TO11AS contains address of data node. R1 contains expected termination address and CR$DTB-2 contains the actual termination address for transfer.
DTD	COMTRP	UNIBUS TIMEOUT Reference to the DTE-20 caused a UNIBUS timeout. Possible Cause: Malfunction of the hardware in the KL.
DTF	QPRDTE	TO-10 DTE TRANSFER FAILURE A TO-10-done interrupt has occurred but the TO-10 address in the DTE TO10AD register (register 20) did not have the expected value. Since TO10AD gets incremented for each byte transferred, it should point to the first word following the packet that was sent to the KL. Possible Cause: The PDP-11 gave the KL the wrong byte count or, more likely, the DTE has a hardware malfunction. TO10SZ contains the size of the transfer and TO10AS the start address. The expected termination address is in R4.

RSX-20F STOP CODES AND I/O ERROR CODES (Cont)

Code	Module	Meaning
ETE	QPRDTE	TO-11 TRANSFER ERROR A DTE interrupt occurred with the TO11ER bit set in the DTE status register (register 34). Possible Cause: Hardware malfunction along the data path between the KL and PDP-11 (MBOX, EBOX, EBUS, DTE-20, through to 11-memory).
FTA	LC	FILES-11 TASK ABORTED A task occupying F11TPD partition has aborted and the task termination notification task (TKTN) cannot be started since it too runs in the F11TPD partition. Possible Cause: .TKTN may have aborted. R5 and .CRTSK point to the Active Task List (ATL) node of the aborted task.
IAS	SCH	UNKNOWN SIGNIFICANT EVENT An unused bit in .SERFG has been set. Possible Cause: PDP-11 hardware malfunction or corrupted software in PDP-11. .SERFG has the bit set.
ILF	QPRDTE	ILLEGAL PROTOCOL FUNCTION The function code in a TO-11 protocol header specified a function that is outside the legal range or that is currently unimplemented. Possible Cause: KL software is corrupted or hardware malfunction along data path between KL and PDP-11. R1 contains the function code times two. R4 contains the address of the protocol header.
ILQ	QPRDTE	ILLEGAL QUEUE COUNT The KL and the PDP-11 disagree on the number of direct transfers that have thus far taken place from the KL to the PDP-11. You should take into account that indirect headers are sent across the DTE-20 as direct packets. Possible Cause: The PDP-11 is missing TO-11 doorbell interrupts, or the software of either the KL or the PDP-11 is corrupted. STATI+0 to STATI+2 contain the KL's TO-11 status word as read by RSX-20F at the last examine. STATI+4 is the count the KL expects, and TO10QC is the count the PDP-11 expects.
LRF	SCH	LOAD REQUEST FAILURE An attempt to load a nonresident monitor routine into the F11TPD partition failed. Possible Cause: The Files-11 system is incomplete or damaged.
MPE	LC	MEMORY PARITY ERROR A memory parity error has occurred in the PDP-11 (trap to location 114). The memory status registers are stored starting at location PARSAVE. (Refer to the PDP-11 Processor Handbook for details.)

RSX-20F STOP CODES AND I/O ERROR CODES (Cont)

Code	Module	Meaning
PT1	QPRDTE	PROTOCOL BROKEN An illegal protocol device number was specified in TO-11 request. The number was found to be greater than the maximum allowed device number .DQPSZ (currently 10). Possible Cause: KL software is corrupted or hardware malfunction along the data path between the KL and PDP-11. The device number from the protocol header is in TO11DV.
PT2	QPRDTE	PROTOCOL ERROR 2 An illegal protocol function was specified in a TO-11 request. The function was found to be greater than the allowed maximum BC.FNM (currently 34). Possible Cause: Same at PT1 above. The function code from the protocol header is in TO11FN.
PT3	QPRDTE	PROTOCOL ERROR 3 The PDP-11 has received a doorbell interrupt from the KL. The indirect bit in the KL's TO-11 status word indicates that an indirect transfer is to be initiated. The function code, however, sent in the last protocol header, does not indicate that an indirect request is in progress (the most significant bit of the function code was not set). Possible Cause: Same as PT1 above. TO11FN contains the function code and STATI contains the TO-11 protocol status word.
PT4	QPRDTE	PROTOCOL ERROR 4 The KL wants to send a packet to the PDP-11, but the packet size is greater than the maximum allowed size of 100. Possible Cause: Same as PT1 above. The size in in EQSZ.
RED	RED	REDIRECT ERROR A fatal error has occurred during an MCR REDIRECT command. The file control service is corrupted. Call your Software Support Specialist.
RES	LC	RESERVED INSTRUCTION TRAP This is the PDP-11 trap to location 10. An attempt was made to execute an illegal or reserved instruction. Refer to the PDP-11 Processor Handbook for further details. Possible Cause: PDP-11 software is currupted or a PDP-11 hardware malfunction occurred.
TBT	LC	T-BIT TRAP This PDP-11 trap to location 14 occurs when the BPT instruction (not used by RSX-20F) is executed or when the T-bit is set. (See the PDP-11 Processor Handbook for further details.)

RSX-20F STOP CODES AND I/O ERROR CODES (Cont)

Code	Module	Meaning
		Possible Cause: Corrupted PDP-11 software or PDP-11 hardware malfunction.
TET	QPRDTE	TO-10 TRANSFER ERROR A DTE-20 interrupt has occurred with either TO10ER (TO-10 error) or MPE11 (PDP-11 parity error) bit set in the DTE-20 status register (register 34). Possible Cause: DTE-20 hardware error, PDP-11 memory parity error, or hardware malfunction along the data path between the PDP-11 and KL.
T04	LC	TRAP AT LOCATION 4 The PDP-11 traps to location 4 when it makes a word reference to an odd address or when a bus timeout occurs. (See the PDP-11 Processor Handbook for further details.) Possible Cause: PDP-11 software is corrupted, or a PDP-11 peripheral device is malfunctioning or has gone away.
UIE	QPRDTE	UNIMPLEMENTED PROTOCOL FUNCTION The KL uses bits 0-2 of its TO-11 status word in the communications region to inform the front end of any disaster occurring in the KL. These bits are read by the front end on receipt of a TO-11 doorbell. The currently implemented functions are KL-RELOAD REQUEST and KL POWER FAIL. Any other bits that are set cause this halt. Possible Cause: Corrupted KL software, a KL hardware malfunction or any hardware malfunction along the data path between KL and PDP-11 could be the cause of this error.

The following is a list of possible I/O error codes. Since these codes are returned by the device handlers and file control primitives in RSX-20F, they are global in the sense that they can come from any utility in the system. That is, a code of -33 means the same thing when it comes from PIP that it means when it comes from SAV. Because of the global nature of the error codes, it is not possible to describe the exact problem; the situation is different with different utilities. Therefore, the following list does not attempt to explain the error code other than to list the message associated with it.

Note that there are two messages associated with the code -2. This is legitimate; a message code of -2 is produced in two types of situations.

Code	Message
-1	Bad parameters
-2	Invalid function code
-2	EBOX stopped
-3	Device not ready
-4	Parity error on device
-5	Hardware option not present
-6	Illegal user buffer
-7	Device not attached
-8	Device already attached
-9	Device not attachable

Code	Message
-10	End of file detected
-11	End of volume detected
-12	Write attempted to locked unit
-13	Data overrun
-14	Send/receive failure
-15	Request terminated
-16	Privilege violation
-17	Sharable resource in use
-18	Illegal overlay request
-19	Odd byte count or virtual address
-20	Logical block number too large
-21	Invalid UDC module
-22	UDC connect error
-23	Caller's nodes exhausted
-24	Device full
-25	Index file full
-26	No such file
-27	Locked from write access
-28	File header full
-29	Accessed for write
-30	File header checksum failure
-31	Attribute control list format error
-32	File processor device read error
-33	File processor device write error
-34	File already accessed on LUN
-35	File ID, file number check
-36	File ID, sequence number check
-37	No file accessed on LUN
-38	File was not properly closed
-39	Open - no buffer space available for file
-40	Illegal record size
-41	File exceeds space allocated, no blocks
-42	Illegal operation on file descriptor block
-43	Bad record type
-44	Illegal record access bits set
-45	Illegal record attributes bits set
-46	Illegal record number - too large
-47	Multiple block read/write - not implemented
-48	Rename - two different devices
-49	Rename - new file name already in use
-50	Bad directory file
-51	Cannot rename old file system
-52	Bad directory syntax
-53	File already open
-54	Bad file name
-55	Bad device name
-56	Bad block or device
-57	Enter duplicate entry in directory
-58	Not enough stack space (FCS or FCP)
-59	Fatal hardware error on device
-60	File ID was not specified
-61	Illegal sequential operation
-62	End of tape detected
-63	Bad version number
-64	Bad file header
-65	Device off-line
-66	File expiration date not reached
-67	Bad tape format
-68	Not ANSI "D" format byte count

Tops-10

KL10 Maintenance Guide
Volume II

TOPS-10

Replace all text pages following the **TOPS-10** tab with the following sheets (pages 1 through 27).

Discard this sheet after completing directions.

Table of Contents

TOPS-10 SYSTEM PROGRAM LIBRARY
The programs in the TOPS-10 System Program Library are listed and described in Table 1.

Table 1 TOPS-10 System Program Library

Program	Description
AID	Algebraic Interpretive Dialogue. Each command occupies one line and can be executed immediately or stored as part of a routine for later execution. This interpreter requires no previous programming experience.
ALCFIL	A program used for allocating space for a new file or reallocating space for an existing file in one contiguous region on the disk.
ALGOL	ALGOrithmic Language. A scientifically oriented language that contains a complete syntax for describing computational algorithms.
BACKUP	A program used to save disk files on magnetic tape, and later to restore any or all of these files to disk. Magnetic tape is the medium used for backup storage of disk files and for transporting files between sites.
BASIC	Beginner's All-purpose Symbolic Instruction Code. A time-sharing computer programming language that is used for direct communication between terminal units and computer centers. The language was developed at Dartmouth College.
BATCON	The Batch controller. This program reads a job's control file, starts the job, and controls the job by passing commands and data to it.
BLISS	A programming language that enables users to write programs consisting only of declarations, which establish structure, and expressions, which compute values. It is specifically designed for implementing system software.
BOOTS	A bootstrap program whose main functions are to load a program into core from a SAVE file on a disk unit and/or to dump core as a SAVE file for later analysis.
CHKPNT	A program used to gather the information on the utilization of the DECsystem-10 for accounting and billing purposes.
COBDDT	The COBOL Dynamic Debugging Technique. With COBDDT the user can: 1. Change data-name contents, 2. Set breakpoints, 3. Continue the program, 4. Display the contents of a data-name, and 5. Trace paragraphs and sections.
COBOL	COmmon Business Oriented Language. A programming language used in programming data processing applications.
COMPIL	A utility program that allows the user to type a short, concise command string in order to cause a series of operations to be performed. COMPIL deciphers the command and constructs new command strings for the system program that actually processes the command. Several of the commands that invoke COMPIL are EDIT, COMPILE, CREF, and EXECUTE.
CREF	A program which produces a sequence-numbered assembly listing followed by tables showing cross references for all operand-type symbols, all user-defined operators, and/or all operation codes and pseudo-op codes.
DAEMON	A program for writing all or parts of a job's core area and associated monitor tables onto disk.
DATDMP	A program for dumping the core data base.

Table 1 TOPS-10 System Program Library (Cont)

Program	Description
DDT	The Dynamic Debugging Technique program used for on-line checkout, testing, examination, modification, and program composition of object programs.
DIRECT	A program for producing directory listings of disks and DECtapes.
DSKLST	A program which gives status and statistics of all user disk files at a given time.
DSKRAT	A damage assessment program that scans a file structure and reports any inconsistencies detected.
DTBOOT	A bootstrap program used to save and restore core images on DECtape or magnetic tape. It operates only in executive mode.
DUMP	A program that outputs selected portions of a file in one of the various formats that can be specified by the user.
EDDT	Executive DDT (Dynamic Debugging Technique). A version of DDT used for debugging programs, such as the monitor, in executive mode.
EDIT	A program used to build and edit ASCII text files.
FAILSAFE	A program used to save the contents of the disk on magnetic tape and later restore the saved contents back onto disk.
FILDDT	File DDT (Dynamic Debugging Technique). A version of DDT used for examining and changing a file on disk instead of in core memory. This program is used to examine a monitor for debugging purposes.
FILEX	A general file transfer program used to convert between various core image formats and to read and write various DECtape directory formats and standard disk files.
FORTRAN	FORmula TRANslator. A procedure-oriented programming language designed for solving scientific-type problems by expressing the procedure for their solution as arithmetic formulas. The language is widely used in many areas of engineering, mathematics, physics, chemistry, biology, psychology, industry, military, and business.
FUDGE 2	A program used to update libraries containing one or more relocatable binary modules and to manipulate modules within these libraries.
GLOB	A program used to read collections of relocatable binary modules which have been loaded together (from both library files and separate files) in order to generate an alphabetical cross-referenced list of all the global symbols encountered. When a program is composed of many modules which communicate via global symbols, it is useful to have an alphabetical list of all global symbols with the names and modules in which they are defined and referenced.
GRIPE	A program that accepts text from the user and records it in a disk file for later examination by the operations staff.
INITIA	A program for performing standard system initialization for a particular terminal. It is used to initiate specific programs, such as the spooling programs, on the designated terminal.
LINK	A program that provides automatic loading and relocation of binary programs, producing an optional storage map, and performs loading and library searching. Also, the program loads and links relocatable binary programs and generates a symbol table in core for execution under DDT.

Table 1 TOPS-10 System Program Library (Cont)

Program	Description
LINKER	A program that combines many input modules into a single module for loading purposes. Thus, it allows for independent compilations of modules. Typically, it satisfies global references and may combine control sections.
LINKING LOADER	A program that provides automatic loading, relocation, and linking of compiler- and assembler-generated object modules.
LOGIN	The system program by which the system users gain access to the computing system.
LOOKFL	A program for typing the characteristics of a single disk file, such as creation date and number of words written, on the terminal.
MONEY	A program for reading the system's time accounting file and assigning a monetary charge for each user according to the time and resources that he has used on the system.
MONGEN	The monitor generator dialogue program that enables the system programmer to define the hardware configuration of his individual installation and the set of software options that he wishes to select for his system.
OMOUNT	A program that interfaces with the operator in order to handle requests concerning removable media.
OPSER	The OPerator SERvice program that facilitates multiple job control from a single terminal by allowing the operator or user to initiate several jobs from his terminal.
PIP	The Peripheral Interchange Program which transfers data files from one standard I/O device to another and performs simple editing functions, such as sequencing, trailing blank suppression, and compressing blanks into tabs, and magnetic tape control functions.
PLEASE	A program that provides the user with two-way communication with the operator via an operator's terminal that is reserved for PLEASE commands and the user's terminal.
QMANGER	The Batch queue manager. QMANGR is called by BATCON to schedule jobs by computing and dynamically revising job priorities.
QUEUE	The system program that allows users to add, delete, list, or modify queue entries in the various system queues.
QUOLST	A program that prints the user's quotas for each file structure in his search list and the number of free blocks available in each file structure.
REACT	A program for maintaining administrative control files. It can be used to create, modify, delete or list entries in a file.
RUNOFF	A program that facilitates the preparation of typed or printed manuscripts by performing formatting, case shifting, line justification, page numbering, titling, and indexing.
SCRIPT	A program that sends predetermined sequences of characters over multiple pseudoterminals in order to simulate a load on the system for testing, measurement, and analysis.
SETSRC	A program that allows the user to list or change his search list.
SOUP	The SOftware Updating Package that consists of a set of programs for facilitating the updating of system or user source files.

Table 1 TOPS-10 System Program Library (Cont)

Program	Description
SPRINT	The Batch input stacker. SPRINT reads any sequential input stream, sets up the job's control file and data files, and enters the job into the Batch input queue.
SYSDPY	A variation of the SYSTAT program which runs on a keyboard display terminal (at up to 2400 baud). SYSDPY maintains a dynamic display of system status by periodically altering lines of the display to replace old information with the latest information.
SYSERR	SYSERR is the report generating portion of the DECsystem-10 and DECSYSTEM-20 error detection, recovery, and reporting system. As an error is detected by the monitor, various pieces of information describing pertinent hardware and software status are gathered and appended to a disk file. SYSERR is a user-mode program which lists the contents of this file at the direction of the command string.
SYSTAT	A program that outputs to the user's terminal status information on the system as a whole, on selected aspects of the system, or on a selected job or set of jobs.
TECO	A sophisticated Text Editor and Corrector program that allows simple editing requests, character string searches, complex program editing, command repetition, and text block movement. TECO editing is performed on files consisting of ASCII characters.
UMOUNT	A program for user interfacing for the handling of requests concerning removable media.

TOPS-10 COMMAND LANGUAGE
The TOPS-10 Operating System supports approximately 96 commands. The conventions used to illustrate these commands are described in Table 1. The individual commands are arranged in alphabetical order in Table 2.

Note that the complete command format has been shown for the commands. Depending on the circumstances, only part of this format may be required. Refer to the DECsystem-10 Operating System Commands manual to determine the arguments required for a particular task. In addition, the commands can be abbreviated as long as the abbreviation does not conflict with any other command abbreviation.

Many command strings allow wild-card characters to be used in place of alphanumeric characters. These characters permit more than one file or directory to be referenced by a single specification. Two such wild-card characters are available:

1. * - The asterisk is a wild card for an entire field. When positioned in the appropriate context, it means:

	Examples	
a. any filename or extension	*.EXT	FILNAM.*
b. any project number or programmer number (also, any subfile directory)	[*,1164]	[27,*]

   Note that *.* and [*,*] are also possible.

2. ? - The question mark is a wild card for a single character. It can be used in any field mentioned above, provided the * does not share the field. It means: any character.

   Examples:

   .EX? FI???.EX? ?ILNAM. [27,116?] [*,11??]

In addition, the directory name can be specified with the project number, the programmer number, or both numbers missing.

ERROR MESSAGES
TOPS-10 operating systems use four types of stop codes.

DEBUG - If a priority interrupt is in progress, the condition is not immediately harmful to the system or any job. The monitor types out a message on the console terminal and continues. If no priority interrupt is in progress, a DEBUG stopcode acts the same as a JOB stopcode.

JOB - If no priority interrupt is in progress, the condition jeopardizes the integrity of the current job. The monitor sends a message to both the console terminal and the user's terminal and aborts the job. If a priority interrupt is in progress, then a JOB stopcode acts like a STOP stopcode.

STOP - This condition jeopardizes the integrity of the entire system. The monitor sends a message to the console terminal, aborts all jobs, and reloads the system.

HALT - This condition is so serious that the monitor is not going to do anything that might affect stored data. The system executes a HALT instruction and waits for the operator to initiate a reload.

Table 3 lists and describes the STOP CODES associated with a TOPS-10 operating system (6.03 release).

Table 1 TOPS-10 Command Conventions

Convention	Description
adr	An octal address.
arg	A letter or word specifying the desired function of the command.
control file	The name of the control file for the Batch System.
core	Decimal number of blocks (n or nK) or pages (nP) of core.
dev:	Any physical (or logical, normally) device name (e.g., MTA:). The colon must be included.
devn:	Any phsical device name of three characters followed by a unit number of one to three numerals (e.g., DTA3:). The colon must be included.
devSnn:	Any physical device name of three characters followed by the letter S and a station number (e.g., LPTS2:). The colon must be included.
[directory]	A designation identifying a particular disk area. This designation can be in the form [proj,prog] which identifies a UFD or [proj,prog,sfd,sfd, ...] which identifies a sub-file directory path branching from a UFD. The square brackets are required.
drives	The physical drives on which a unit is to be mounted.
file.ext	Any legal filename from one to six characters followed by a dot and an extension of zero to three characters.
file structure	The name of a particular disk. This name is usually in the form DSKA, DSKB, etc.
input specifications	File specifications for the disk files to be processed.
jobn	A user's job number assigned by the system.
jobname	A name of up to six characters of the job being entered into one of the system queues.
lh	Left half of a 36-bit word.
logdev:	Any logical device name from one to six alphanumeric characters. The colon should be included.
log file	The name to be given to the log file created by the Batch system.
n or m	A number.
x	A letter.
<nnn>	A three-digit octal code indicating the protection of a file. This code can appear only on the output side of the command string and must be enclosed in angle brackets.
prog	A program name of six or fewer characters.
rh	Right half of a 36-bit word.
/S	One or more switches used to modify the command string.
↑tape id↑	A one to six character identifying name recorded on a DECtape.
text	A message to be sent to the designated user or terminal.

Table 1 TOPS-10 Command Conventions (Cont)

Convention	Description
[user number]	A numeric identification assigned to the user for the purpose of gaining access to the system. It is usually two numbers separated by a comma.
=	An equal sign used in command strings to separate the output specification (left of the equal sign) from the input specification (right of the equal sign).

Table 2 TOPS-10 Command Summary

Command	Description
ALCFIL	R ALCFIL<CR> Allocates space for a new file or reallocates space for an existing file in one contiguous region on the disk.
ASSIGN	ASSIGN dev:logdev:<CR> ASSIGN devSnn:logdev:<CR> ASSIGN devn:logdev:<CR> Allocates an I/O device to the user's job without operator intervention.
ATTACH	ATTACH jobn [user number]<CR> Detaches the current job and connects the terminal to the specified detached job.
BACKSPACE	BACKSPACE MTAn:m FILES<CR> BACKSPACE MTAn:m RECORDS<CR> Spaces a magnetic tape backward the specified number of files or records.
CCONTINUE	CCONTINUE<CR> Continues the program from the point at which it was interrupted, but leaves the terminal in monitor mode.
CLOSE	CLOSE dev:<CR> Terminates I/O currently in progress on the specified device, performs the CLOSE UUO, but does not release the device.
COMPILE	COMPILE dev:file.ext [directory]/S,...<CR> Produces relocatable binary files (.REL files) for the specified source files.
CONTINUE	CONTINUE<CR> Continues the program from the point at which it was interrupted.
COPY	COPY dev: ↑tape id↑ file.ext [directory] <nnn> = dev:file.ext [directory], file.ext [directory], ...<CR> Transfers files from one I/O device to another.
CORE	CORE core<CR> Types or modifies the amount of core assigned to the user's job.
CPUNCH	CPUNCH jobname = dev:file.ext [directory]/s, ...<CR> Places entries into the card punch output spooling queue.

Table 2 TOPS-10 Command Summary (Cont)

Command	Description
CREATE	CREATE file.ext<CR> Opens a new file on disk for creation with LINED.
CREF	CREF<CR> Lists on LPT: any cross-referenced listing files generated by a previous COMPILE, LOAD, EXECUTE, or DEBUG command.
CSTART	CSTART adr<CR> Begins execution of a program that was either loaded with a GET command or interrupted, but leaves the terminal in monitor mode.
D(eposit)	D lh rh adr<CR> Deposits information in the user's core area.
DAYTIME	DAYTIME<CR> Types the current date followed by the time of day.
DCORE	DCORE dev:file.ext [directory]<CR> Writes a core image file of the user's core area.
DDT	DDT<CR> Copies the saved program counter and starts the program at the beginning address of DDT if DDT was loaded with the program (automatic in 6.01).
DEASSIGN	DEASSIGN dev:<CR> Returns devices assigned to the user's job to the monitor's pool of available devices and clears logical names.
DEBUG	DEBUG dev:file.ext [directory]/s, ...<CR> Produces relocatable binary files (.REL files) for the specified source files, loads the .REL files along with an appropriate system debugging program, and prepares for debugging.
DELETE	DELETE dev:file.ext [directory], ...<CR> Deletes files from DECtape or disk.
DETACH	DETACH<CR> Disconnects the terminal from the current job without affecting the status of the job.
DIRECT	DIRECT dev:file.ext [directory] = dev:file.ext [directory]/s, ...<CR> Lists the directory entries for the specified arguments.
DISMOUNT	DISMOUNT dev:/s, ...<CR> Returns, via the operator, devices assigned to the user's job to the monitor's pool of available devices.
DSK	DSK jobn<CR> Types disk usage for the combined structures of the specified job.
DTCOPY	R DTCOPY<CR> Copies còntents of one DECtape to another, clears the blocks on a DECtape and clears the directory, compares two DECtapes, and/or loads and writes a bootstrap loader.

Table 2 TOPS-10 Command Summary (Cont)

Command	Description
DUMP	DUMP/S ...<CR> Writes a core image file, analyzes the file written, and provides printed output.
DUMP	R DUMP<CR> Provides printable output of data files in specified forms and modes.
E(xamine)	E adr<CR> Examines the specified core location in the user's area.
EDIT	EDIT file.ext<CR> Opens the specified file already existing on disk for editing with LINED.
EOF	EOF MTAn:<CR> Writes an end-of-file mark on the specified magnetic tape.
EXECUTE	EXECUTE dev:file.ext [directory]/s, ...<CR> Produces relocatable binary files (.REL files) for the specified source files, loads the .REL files, and begins execution.
FAILSAFE	R FAILSAFE<CR> Saves and restores disk files.
FILCOM	R FILCOM Compares two versions of a file and outputs any differences.
FILE	FILE arg, ↑tape id↑, file.ext, file.ext, ...<CR> Provides remote control, via the operator, of DECtape-to-disk and disk-to-DECtape transfers.
FILEX	R FILEX<CR> Converts between various core image formats, and reads and writes various directory formats.
FINISH	FINISH dev:<CR> Terminates I/O in progress on the specified device and performs the RELEASE UUO and DEASSIGN command.
FUDGE	FUDGE<CR> Creates a library REL file by reading a temporary file generated by a previous COMPILE, LOAD, EXECUTE, or DEBUG command containing the /FUDGE switch.
FUDGE2	R FUDGE2<CR> Updates files containing relocatable binary programs, and manipulates the programs within these files.
GET	GET dev:file.ext [directory] core<CR> Loads a core image from the specified device, but does not begin execution.
GLOB	R GLOB<CR> Reads multiple binary files to produce an alphabetical cross-referenced listing of all global symbols encountered.

Table 2 TOPS-10 Command Summary (Cont)

Command	Description
GRIPE	R GRIPE<CR> Accepts text from a user and records it in a disk file for the operations staff.
HALT	HALT<CR> or ↑C Stops the job and stores the program counter in the job data area. Control C can be used at user level as well as at monitor level.
HELP	HELP dev:prog<CR> or HELP dev:*<CR> Outputs useful documentation on various system features.
INITIA	INITIA<CR> Performs standard system initialization for the terminal issuing the command.
JCONT	JCONT jobn<CR> Continues the specified job if it was in a ↑C state because of a call to the device error message routine (HNGSTP).
KJOB	KJOB logfile = file structures/s<CR> Gives up access to the system.
LABEL	LABEL DEV: ↑tape id↑<CR> Writes an identifier onto a DECtape.
LIST	LIST dev:file.ext [directory]/s, ...<CR> Lists the specified files on the line printer.
LOAD	LOAD dev:file.ext [directory]/s, ...<CR> Produces relocatable binary files (.REL files) for the specified files and loads the .REL files generated.
LOCATE	LOCATE nn<CR> Establishes, logically, the user's job at a specified station.
LOGIN	LOGIN user number/s ...<CR> Provides access to the system.
MAKE	MAKE dev:file.ext [directory]<CR> Opens a new file on disk for creation with TECO.
MOUNT	MOUNT dev:logdev:/s drives<CR> Allocates an I/O device to the user's job via the operator.
OPSER	R OPSER<CR> Provides multiple job control from a single terminal.
PJOB	PJOB<CR> Outputs the job number to which the terminal is currently attached.
PLEASE	PLEASE dev:prog! text<CR> Provides two-way communication between the user and the operator.
PLOT	PLOT jobname = dev:file.ext [directory]/s, ...<CR> Places entries into the plotter output spooling queue.

Table 2 TOPS-10 Command Summary (Cont)

Command	Description
PRESERVE	PRESERVE file.ext, file.ext, ...<CR> Renames the specified files with the standard protection inclusively ORed with 100.
PRINT	PRINT jobname = dev:file.ext [directory]/s, ...<CR> Pleaces entries into the line printer output spooling queue.
PROTECT	PROTECT file.ext<nnn>, file.ext<nnn>, ...<CR> Sets the specified files to the requested protections.
PUNCH	PUNCH jobname = dev:file.ext [directory]/s, ...<CR> Places entries into the paper tape punch output spooling queue.
QUEUE	QUEUE queue name:jobname = input specifications<CR> Enters items into the specified system queue.
QUOLST	R QUOLST<CR> Types the used, loggin-in quota, and logged-out quota for each file structure to which the user has access, followed by the number of free blocks left on that structure.
R	R file.ext core<CR> Loads a core image from the system device (SYS:) and starts it at the location specified within the file.
REASSIGN	REASSIGN dev:jobn<CR> Gives the specified device to the designated job.
REATTA	R REATTA<CR> Transfers the job from the current terminal to the specified terminal.
REENTER	REENTER<CR> Starts the program at an alternate entry point specified by the program.
RENAME	RENAME new = old, new = old, ...<CR> Changes the name and protection of one or more files on DECtape or disk.
RESOURCES	RESOURCES<CR> Outputs the names of all available devices (except for terminals and PTYs), all file structures, and all physical units not in file structures.
REWIND	REWIND dev:<CR> Rewinds a magnetic tape or DECtape.
RUN	RUN dev:file.ext [directory] core<CR> Loads a core image from the specified device and starts it at the location specified within the file.
SAVE	SAVE dev:file.ext [directory] core<CR> Writes a core image of the user's core area on the specified device.
SCHED	SCHED<CR> Outputs the schedule bits set by the last SET SCHED command.

Table 2 TOPS-10 Command Summary (Cont)

Command	Description
SEND	SEND dev:text<CR> SEND jobn text<CR> Provides a one-way interconsole line of communication.
SET BLOCKSIZE	SET BLOCKSIZE dev:nnnn<CR> Sets the default blocksize for the specified magnetic tape.
SET BREAK	SET BREAK AT adr ON arg, ...<CR> SET BREAK NO arg, ...<CR> SET BREAK NONE<CR> Sets address break in program according to specified conditions used with KI10 processors only.
SET CDR	SET CDR file<CR> Sets the filename for the next card-reader spooling intercept.
SET CPU	SET CPU CPxn<CR> SET CPU NO CPxn<CR> SET CPU ALL<CR> SET CPU ONLY CPxn<CR> Sets the CPU specification for the job. This command is only available on multiprocessor systems (1055, 1077) and requires certain bits be set in the privilege word.
SET DENSITY	SET DENSITY dev:nnn<CR> Sets the default density for the specified magnetic tape.
SET DSKFUL	SET DSKFUL ERROR<CR> SET DSKFUL PAUSE<CR> Controls the job when the user has exhausted his disk space.
SET DSKPRI	SET DSKPRI n<CR> Sets the priority for the job's disk operations (data transfers and head positionings). Requires certain bits to be set in the privilege word.
SET HPQ	SET HPQ n<CR> Sets the high priority scheduler run queue for the job. Requires certain bits to be set in the privilege word.
SET PHYSICAL	SET PHYSICAL LIMIT core<CR> SET PHYSICAL GUIDELINE CORE<CR> Specifies when the job will go virtual and specifies a guideline for the page fault handler if GUIDELINE is designated. Used with KI10 processors only.
SET SPOOL	SET SPOOL dev:, dev:, ...<CR> SET SPOOL ALL<CR> SET SPOOL NONE<CR> SET SPOOL NO dev:, dev:, ...<CR> Adds devices to or deletes devices from the list of spooled devices for this job.
SETSRC	R SETSRC<CR> Manipulates the job's search list or system's search list.
SET TIME	SET TIME n<CR> Sets the central processor time limit for the job.

Table 2 TOPS-10 Command Summary (Cont)

Command	Description
SET TTY	SET TTY NO arg<CR> SET TTY arg Sets properties to be associated with the terminal.
SET VIRTUAL LIMIT	SET VIRTUAL LIMIT core<CR> Specifies the limit on the virtual memory for a job. Used with KI10 processors only.
SET WATCH	SET WATCH arg, arg, ...<CR> SET WATCH ALL<CR> SET WATCH NONE<CR> SET WATCH NO arg, arg, ...<CR> Sets the output of incremental job statistics.
SKIP	SKIP MTAn:m FILES<CR> SKIP MTAn:m RECORDS<CR> SKIP MTAn:EOT<CR> Moves the specified magnetic tape forward the designated number of files or records or to the logical end of tape.
SSAVE	SSAVE dev:file.ext [directory] core<CR> Writes a core image of the user's core area on the specified device. When it is loaded with a GET (or RUN) command, the high segment will be sharable.
START	START adr<CR> Begins execution of a program either previously loaded with the GET command or interrupted while running.
SUBMIT	SUBMIT jobname = control file, log file/s<CR> Places entries into the Batch input queue.
SYSTAT	SYSTAT/S<CR> Prints information about the current status of the system.
TECO	TECO dev:file.ext [directory]<CR> Opens the specified file for editing with TECO.
TIME	TIME jobn<CR> Outputs the running time for the specified job.
TPUNCH	TPUNCH jobname = dev:file.ext [directory]/s, ...<CR> Places entries into the paper tape punch output spooling queue.
TTY	TTY NO arg<CR> TTY arg<CR> Sets properties to be associated with the terminal.
TYPE	TYPE dev:file.ext [directory]/s, ...<CR> Types the specified files on the user's terminal.
UNLOAD	UNLOAD dev:<CR> Rewinds and unloads the specified magnetic tape or DECtape.
USESTAT	USESTAT<CR> or ↑T Prints information on the terminal concerning the user's job. Control T can be used at user level also.

Table 2 TOPS-10 Command Summary (Cont)

Command	Description
VERSION	VERSION<CR> Outputs the version number of a program on the terminal.
WHERE	WHERE dev:<CR> Outputs the station number of the specified device.
ZERO	ZERO dev: [directory]<CR> Clears the directory of the specified device.

Table 3 TOPS-10 Stop Code Summary

Monitor Module	STOPCD Name	STOPCD Type	Comment
XTCSER	28B*	DEBUG	DA28 is broken
FSXKON	4IF	DEBUG	RS04 is not fancy
D85INT	5WE	DEBUG	DC75 wrong PDP-11 code
D6SINT	6DD	DEBUG	11 gave too much direct data
D6SINT	6DI	DEBUG	Unexpected TO10 DONE interrupt
D6SINT	6ID*	DEBUG	11 gave too much indirect data
D76INT	6MS	DEBUG	DC76 message is short
D76INT	6QF	DEBUG	DC76 queue full
D78INT	8BI*	JOB	????????????
D78INT	8IN*	JOB	Input character count is not 0
D78INT	8NC*	JOB	Not enough monitor free core
D78INT	8ON*	JOB	Output character count is not 0
D78INT	8PI*	JOB	Positive IOWD
D60INT	8VI	DEBUG	DN60 wrong PDP-11 code
D78INT	8VI*	DEBUG	Version incorrect
FILFND	AAD	DEBUG	A. T. already dormant
KISER	AAO	JOB	Access allowed off
KLSER	AAO	JOB	Access allowed off
COMMON	AD#	STOP	CPU n address parity error
FILFND	AES	JOB	Abnormal end of search list
FILIO	AHB	DEBUG	Already have buffer
ONCMOD	AHS	HALT	Already have structure
FILIO	ANU	DEBUG	AU not owned by us
FILFND	AOC	DEBUG	Already own CB
VMSER	APF	DEBUG	Allocated page free
ONCMOD	AR1	DEBUG	ASKDEC returned CPOPJ1
DTESER	ARD	STOP	Runaway driver
KISER	ARF	STOP	Attempt to return free page
KLSER	ARF	STOP	Attempt to return free page
FILFND	ARM	DEBUG	Access rings all messed up
QUESER	AVE	DEBUG	Already have EO
DTESER	BAA	STOP	Buffer already there
CORE1	BAC	DEBUG	Bit already clear
FILFND	BAD	JOB	Block already dormant
FILIO	BAO	DEBUG	Bit already one
FILIO	BAZ	DEBUG	Bit already zero
D85INT	BBS	STOP	Bad byte size
DTESER	BDN	STOP	Bad device number
TAPUUO	BFO	DEBUG	Better find one
NETSER	BFU*†	DEBUG	BUSY fouled up
FILIO	BIN	STOP	I/O to a negative block
FILUUO	BMR	JOB	Block missing from RIB
COMMON	BNF*	HALT	BOOTS not found
FILUUO	BNR	JOB	Block not RIB
FILFND	BNT	DEBUG	Block not there

*Deleted in Version 7.01
†Deleted in Version 6.03A

Table 3   TOPS-10 Stop Code Summary (Cont)

Monitor Module	STOPCD Name	STOPCD Type	Comment
CORE1	BNZ	DEBUG	Bit not zero
CP1SER	BPS	HALT	Both processors stopped
COMCON	BRC	DEBUG	Bad return from CMPBIT
SEGCON	BSN	STOP	Bad segment number
XTCSER	BSY*	DEBUG	DA28 busy
FILIO	BWA	JOB	Block went away
COMMON	C#P*	DEBUG	CPU n power failed?
CP1SER	C1N*	DEBUG	CPU 1 NXM
FILUUO	CAO	DEBUG	Cluster address odd
REFSTR	CAS	HALT	Could not allocate space
COMMON	CD#	STOP	CPU n cache directory parity error
FILIO	CDA	DEBUG	In core copy does not agree
MSGSER	CDD	JOB	Cannot disconnect device
CLOCK1	CFP	JOB	Cannot find PDB
ONCMOD	CGS	HALT	Cannot get STR data block
CLOCK1	CIB	CPU	CPU interlocks broken
FHXKON	CIF	DEBUG	RC10 is not FANCY
REFSTR	CIO	DEBUG	CFP is odd
SCNSER	CL0	STOP	Chunk links to 0
FILFND	CME	DEBUG	CFP modulo error
VMSER	CMS	DEBUG	CORE1 must skip
SEGCON	CMU	STOP	Core messed up
SCHED1	CNA	STOP	Core not available
FILUUO	CNE	DEBUG	Cluster not even
FILUUO	CNF	DEBUG	In core copy not found
KILOCK	CRW	STOP	CA resource wrong
COMCON	CSA	DEBUG	Cannot set access allowed
FILIO	CSE	STOP	Checksum error
SEGCON	CSP	JOB	Cannot store path
NETMCR	CU0	STOP	Cannot use zero dispatch
NETSER	CWN†	DEBUG	Core allocation went negative
FILIO	DBZ	DEBUG	DEPLPC bit zero
FILUUO	DCR	DEBUG	DELRIB CPOPJ return
FILUUO	DDS	DEBUG	DELRIB did not skip
FILUUO	DER	DEBUG	DELRIB error return
COMNET	DFU	DEBUG	Device unrecognized
FILIO	DHA	DEBUG	Do not have AU
FILIO	DHB	DEBUG	Do not have buffer
FILIO	DHD	DEBUG	Do not have DA
FILIO	DND	DEBUG	Drive not dual-ported
DTESER	DNE	STOP	Data count not even
FILUUO	DNF	DEBUG	DDB not found
DTESER	DNH	STOP	Driver not hungry
DTESER	DNI	STOP	DTE not ready
QUESER	DNL	DEBUG	DEO not interlocked
FILUUO	DNR	DEBUG	DELRIB nonskip return
FILUUO	DNS	DEBUG	
KLSER	DOM	STOP	Do not own MM resource
KSSER			
COMCON	DPL	DEBUG	Directory page lost
COMCON	DPN	DEBUG	Directory page nonexistent
NETMCR	DRN	STOP	Data request want negative
VMSER	DSS	DEBUG	DLTSP skipped
DTESER	EFI	STOP	Eleven function illegal
ERRCON	EPO	DEBUG	Exec PDL overflow
REFSTR	ERB	DEBUG	Error reading BAT block
ONCMOD	ERD	DEBUG	Error refreshing disk
TAPSER	ERF	STOP	ERP really fouled up
REFSTR	ERH	DEBUG	Error reading HOME.SYS
ONCMOD	ERM	DEBUG	Error reading MFD

*Deleted in Version 7.01
†Deleted in Version 6.03A

Table 3 TOPS-10 Stop Code Summary (Cont)

Monitor Module	STOPCD Name	STOPCD Type	Comment
REFSTR	ERP	HALT	Too many retrieval pointers
ONCMOD	ERS	DEBUG	Error reading SAT
FILFND	ESS	JOB	Empty system search list
ERRCON	EUE	DEBUG	Exec UUO error
REFSTR	EWB	DEBUG	Error writing block
REFSTR	EWH	DEBUG	Error writing HOME blocks
ONCMOD	EWR	DEBUG	Error while refreshing
FILUUO	FAD	DEBUG	File already dormant
VMSER	FCZ	DEBUG	Funny core bit zero
FILIO	FDP	DEBUG	Fixed head device positioned
ERRCON	FEM	HALT	Fatal error in monitor
NETSER	FFU	STOP	F fouled up
VMSER	FIP	DEBUG	Free page in use
SCNSER	FLE	STOP	Free list empty
DTESER	FNG	STOP	Function no good
VMSER	FON	STOP	Funny address overlaps next
VMSER	FOP	STOP	Funny address overlaps previous
VMSER	FPE	DEBUG	Funny page must exit
KILOCK	FPF	STOP	Page on free list is not free
KISER	FPI	STOP	Free page in use
KLSER	FPI	STOP	Free page in use
KILOCK	FPN	STOP	Free page not found
REFSTR	HBE	DEBUG	Error reading HOME blocks
XTCSER	HDS	STOP	?????????
FILIO	HIF	DEBUG	Hole in file
ONCE	HNF*	HALT	High segment not found
FILIO	HWU	JOB	Hard wrong unit
CLOCK1	IBI	JOB	Intercept block illegal
FILIO	IBZ	JOB	I/O to block zero
UUOCON	ICL	JOB	Illegal channel number
SEGCON	ICN	DEBUG	Incore count negative
ONCMOD	IDC	HALT	Impossible drum condition
KISER	IEZ	DEBUG	IOWD equals 0
KLSER	IEZ	DEBUG	IOWD equals 0
TAPSER	IFI	STOP	Illegal function at interrupt
NETSER	IFU*†	DEBUG	Interrupt flag unrecognized
FILIO	IIP	STOP	I/O in progress error
KISER	IME	JOB	Illegal memory reference from exec
KLSER	IME	JOB	Illegal memory reference from exec
COMMON	IOP	CPU	I/O page failure
DTESER	IPA	STOP	Illegal post address
KLSER	IPC	CPU	Illegal page failure trap code
KSSER			
VMSER	IPF	DEBUG	In use page free
VMSER	IPM	DEBUG	Illegal pointer in MEMTAB
VMSER	IPN	DEBUG	HIPC page not found
FILUUO	IUN	DEBUG	Invalid unit number
UUOCON	JAC	DEBUG	Job data area clobbered
ONCMOD	JDJ	DEBUG	JFFO did not jump
SYSINI	JIT	HALT	Job in transit
CORE1	JJW*†	STOP	Job's JDA is wrong
FILIO	JNC	DEBUG	Job not in core
CLOCK1	JNE*	STOP	JBTADR not equal to CORTAL
COMMON	KAF	STOP	Keep alive failure
DPXKON	KDS	DEBUG	KONEC2 did not skip
SYSINI	KID	HALT	Controller is down
XTCSER	KNF*	STOP	Control not free
D85INT	KR3*†	STOP	Message too large
TAPSER	KSW	DEBUG	Controller status wrong
TAPUUO	LDN*	DEBUG	Tape label DDB not found

*Deleted in Version 7.01
†Deleted in Version 6.03A

Table 3 TOPS-10 Stop Code Summary (Cont)

Monitor Module	STOPCD Name	STOPCD Type	Comment
ERRCON	LN1	STOP	Line not there
FILUUO	LND	DEBUG	Logical name not found
QUESER	LNF	DEBUG	Lock not found
FILIO	LNP	DEBUG	Last pointer not a pointer
SCNSER	LNS	STOP	Line not set up
ERRCON	LNT	STOP	Line not there
FILUUO	LPU	JOB	Last pointer unit change
CP1SER	MAU*	DEBUG	Master already unlocked
NETSER	MBE*+	DEBUG	Monitor buffer exists
METCON	MCM	DEBUG	MCDB is missing
FILFND	MCN	DEBUG	Mount count negative
DTESER	MDM	STOP	Master DTE missing
FILIO	MHB	DEBUG	Must have buffer
ONCE	MIW	STOP	Memory interleaving wrong
VMSER	MIZ	DEBUG	MEMTAB is zero
ERRCON	MMN*	HALT	Monitor memory NXM error
ERRCON	MMP*	HALT	Monitor memory parity error
KILOCK	MMR	STOP	Moving monitor page not requested
FILIO	MNA	DEBUG	Monitor buffer not available
SYSINI	MNM	STOP	Monitor in nonexistent memory
ERRCON	MNR	HALT	Master-11 not running
KILOCK	MPN	STOP	Monitor page not found
REFSTR	MSR	HALT	No second RIB
NETSER	MY1	STOP	Incorrect just gave some back
NETSER	MY2	DEBUG	Already checked this in FEKINT
NETSER	MY4	DEBUG	Garbage
NETSER	MY5	DEBUG	Garbage
CPNSER	N4C	JOB	Not four cached pages
FILUUO	NAP	JOB	Not address pointer
CLOCK1	NCA	STOP	No core assigned
IPCSER	NCM	JOB	No core for message
ONCMOD	NDC*	STOP	No DF10C code
SCNSER	NDJ	DEBUG	No DDB for job
CLOCK1	NDP	DEBUG	Not DDB pointer
CLOCK1	NDS	STOP	Null job did SAVEGET
ONCE	NED*	HALT	No exec DDT
FILUUO	NER	DEBUG	No extended RIB
FILUUO	NET	DEBUG	No extended RIB
UUOCON	NEV	STOP	No exec virtual memory
FEDSER	NFB	STOP	No front-end device block
DTESER	NFC	STOP	No free core
RPXKON	NFD	DEBUG	No front-end drive
VMSER	NFS	DEBUG	No first slot
SYSINI	NFU	DEBUG	No first unit
DTESER	NIS	STOP	DTE not in indirect state
TAPUUO	NIV	STOP	Null interrupt vector
ERRCON	NJT	STOP	Null job has TTY
FILIO	NMB	DEBUG	Need monitor buffer
ONCMOD	NMC	HALT	No more core
NETSER	NMF*	DEBUG	No monitor buffer
REFSTR	NMU	DEBUG	No more units
FILUUO	NNF	DEBUG	NMB not found
FILUUO	NNR	JOB	No next RIB
ONCMOD	NNU	DEBUG	Not new unit
SCNSER	NOT	DEBUG	No operator terminal
SCHED1	NPC	STOP	No PDB in core
FILIO	NPD	DEBUG	No pointer in DDB
KILOCK	NPF	STOP	Next page free
KLSER	NPI	HALT	Not parity instruction
DATMAN	NPJ	DEBUG	No PDB for job

*Deleted in Version 7.01
+Deleted in Version 6.03A

Table 3 TOPS-10 Stop Code Summary (Cont)

Monitor Module	STOPCD Name	STOPCD Type	Comment
KISER	NPN	STOP	Nonexistent page not free
KLSER	NPN	STOP	Nonexistent page not free
KISER	NPP	STOP	No PI in progress
KLSER	NPP	STOP	No PI in progress
ERRCON	NPU	STOP	Null PDL underflow
VMSER	NRF	DEBUG	SWPLST not really fragmented
FILUUO	NRM	JOB	Next RIB missing
ONCMOD	NRS	DEBUG	No RIB in SAT
VMSER	NSE	DEBUG	No SWPLST entry
FILFND	NSL	JOB	No such search list
REFSTR	NSS	DEBUG	No space for SAT
FILIO	NSU	DEBUG	No such unit
SCHED1	NTE	STOP	Not processor queue error
COMNET	NTF	STOP	NT resource mixed up
FILFND	NUB	JOB	No UFB block
FILUUO	NUE	DEBUG	No UFB error
XTCSER	NUI*	DEBUG	Nonexistent unit interrupt
FILUUO	NUN*	DEBUG	NMB use count negative
FILUUO	NUP	DEBUG	No unit change pointer
VMSER	NUS	DEBUG	No unit for swapping
NETSER	NVP	STOP	Not a valid PCB
NETMCR	NWA	STOP	No one wrote anything
DTESER	NWD	STOP	No doorbell
FILIO	NXU	DEBUG	Nonexistent unit
QUESER	OFU	JOB	O-blocks fouled up
VMSER	O1F	DEBUG	Only one fragment
D8SINT	OIP*	DEBUG	Output on progress
FILUUO	ONC	DEBUG	Odd-numbered cluster
ONCE	OVA	HALT	Out of virtual address space
VMSER	P2L	STOP	Page too low
COMCON	PAO	STOP	Page already out
DTESER	PCI	STOP	Previously checked function code illegal
IPCSER	PCN	DEBUG	Packet count negative
NETSER	PCW	STOP	PCB count wrong
FILIO	PDA	DEBUG	Pointers with different addresses
VMSER	PEW	DEBUG	PAGTAB entry wrong
KISER	PEZ	STOP	PAGPTR=0
KLSER	PEZ	STOP	PAGPTR=0
KILOCK	PFA*	STOP	Page free already
VMSER	PFC*	STOP	Page on free core list
VMSER	PFL	DEBUG	Piece on free list
VMSER	PFR	DEBUG	Piece out of free range
LOKCON	PFS	STOP	Page is free in segment
COMCON	PGL	STOP	Pages got lost
ERRCON	PIE	STOP	Priority interrupt error
VMSER	PIF	DEBUG	Page is free
VMSER	PIN	DEBUG	Page in working set
KISER	PIP	STOP	PI in progress
KLSER	PIP	STOP	PI in progress
VMSER	PIW	DEBUG	Page is not in working set
CLOCK1	PJ0	DEBUG	Requeue JOB 0
FILIO	PLP	DEBUG	Past last pointer
KISER	PMU	STOP	PAGTAB is messed up
KLSER	PMU	STOP	PAGTAB is messed up
FILIO	PNE	DEBUG	Pointers not equal
FILFND	PNM	DEBUG	Physical name mismatch
KILOCK	PNP	STOP	Page not present
VMSER	PNW	DEBUG	Page not in working set
CLOCK1	POP*	STOP	PI on progress
SEGCON	POR	STOP	Process out of range

*Deleted in Version 7.01

Table 3 TOPS-10 Stop Code Summary (Cont)

Monitor Module	STOPCD Name	STOPCD Type	Comment
FILIO	PQE	DEBUG	Position queue empty
KLSER	PRF	CPU	Page refill failure
KISER	PSF	STOP	Page in segment free
KLSER	PSF	STOP	Page in segment free
KLSER	PTH	HALT	Parity trap halt
DTESER	PTL	STOP	Packet too large
KLSER	PTP	HALT	Page table parity
CORE1	PTT	DEBUG	Past top of table
SEGCON	PUF	JOB	Path UUO failed
FILUUO	PUN	DEBUG	PPB use count negative
DTESER	QEF	STOP	Queue entry full
SCNSER	QWC	DEBUG	On wrong CPU
SCHED1	RBQ*	STOP	Requeuing to beginning of queue
SCNSER	RCC	STOP	Range checked chunk
TAPUUO	RDN	DEBUG	Regular DDB not found
FSXKON	RDP	DEBUG	RS04 does not position
SEGCON	RDS	STOP	Remap did not skip
ERRCON	REH	HALT	Recursion in error handler
TAPSER	RFU	STOP	Recovery fouled up
FILIO	RHN	DEBUG	Reread HOME block count negative
XTCSER	RIE*	DEBUG	Remote interrupt error
DPXKON	RIF	DEBUG	RP10 is not fancy
D8SINT	RIP*	DEBUG	Read in progress
SCHED1	RJZ	STOP	Requeue JOB zero
ONCMOD	ROU	HALT	Ran out of units
ONCMOD	RPM	DEBUG	Retrieval pointer mismatch
VMSER	RPZ	STOP	Returning page zero
CLOCK1	RSJ	DEBUG	Requeue same job
NETMCR	RTM	STOP	Requested too much
FILIO	RWD	DEBUG	Returning wrong unit's DA
ERRCON	SAC	DEBUG	Strange APR condition
CP1SER	SAU	DEBUG	Scheduler already unlocked
COMMON	SB#*	STOP	CPU n SBus error
FILUUO	SBT	DEBUG	Should not be truncating
VMSER	SBW	DEBUG	SWPLST bits wrong
XTCSER	SCB*	DEBUG	Spurious CONI bit
SEGCON	SCR*	DEBUG	Segment could not be read
FILUUO	SER	JOB	SETDDO error return
FILUUO	SFI*	JOB	STR free count inconsistent
FILIO	SFU	DEBUG	Swapper fouled up
SCHED1	SHU	DEBUG	Swapper hung up
VMSER	SIN	DEBUG	SWPCNT is negative
VMSER	SLF	DEBUG	SWPLST full
FILUUO	SLM	DEBUG	Search list missing
FILFND	SLO	JOB	Search list overflow
VMSER	SLZ	DEBUG	SLECNT is zero
SCHED1	SMU	DEBUG	SWPCNT messed up
SCHED1	SMU	DEBUG	Try to recover from error
KILOCK	SNF	STOP	Segment not found
SWPSER	SNI	DEBUG	Swapping not in progress
NETMCR	SNS	STOP	NTRPCB not set up
SCHED1	SOD	STOP	Space on disk
ERRCON	SOR	STOP	Segment out of range
FILUUO	SPM	JOB	Second pointer missing
CP1SER	SPS	HALT	Second processor stopped
ONCMOD	SRE	DEBUG	SAT read error
SWPSER	SRO	STOP	Space ran out
SWPSER	SSD	STOP	Swap space disappeared
KILOCK	SSO	STOP	Segment swapped out
SWPSER	SWN	DEBUG	SQREQ went negative

*Deleted in Version 7.01

Table 3 TOPS-10 Stop Code Summary (Cont)

Monitor Module	STOPCD Name	STOPCD Type	Comment
DTESER	T1E	STOP	TO11 error
XTCSER	TC0*	DEBUG	?????????
XTCSER	TC1*	STOP	?????????
XTCSER	TC2*	DEBUG	?????????
XTCSER	TC3*	DEBUG	?????????
XTCSER	TC4*	DEBUG	?????????
XTCSER	TC5*	DEBUG	?????????
XTCSER	TC6*	DEBUG	?????????
XTCSER	TC7*	STOP	?????????
FILUUO	TCI	DEBUG	Truncation check inconsistent
FILIO	TMP	DEBUG	Too many pointers
REFSTR	TMR	HALT	Too many retrieval pointers
ONCMOD	TMU	HALT	Too many units
TSKSER	TND*	DEBUG	Tasks not defined
DTESER	TNI	STOP	TO10 not idle
DTESER	TQP	STOP	Found TO11 queue point
DTESER	TXE	STOP	TO10 error
FILIO	UDE	DEBUG	Unit does not exist
FILUUO	UDM	JOB	UFD data missing
FILUUO	UFI	STOP	Unit free count inconsistent
D8SINT	UID	DEBUG	Unexpected input done
ONCMOD	UIF	HALT	Unit already in file STR
ERRCON	UIL	STOP	UUO at interrupt level
XTCSER	UIP*	DEBUG	Not a unique interrupt
FILUUO	UNF	DEBUG	UFB not found
COMMON	UNJ	DEBUG	Illegal null job UUO
VMSER	UNL	DEBUG	UPMP not last
D8SINT	UOD*	DEBUG	Unexpected output done
FILUUO	UPC	JOB	Unit change pointer clobbered
KLSER	UPF	HALT	Unexpected page fail
FILIO	UPI	DEBUG	Unit pointer illegal
TAPSER	USW	DEBUG	Unit status wrong
VMSER	WAD	DEBUG	WSBTBL and AABTBL discrepancy
DTESER	WCN	STOP	Wrong CPU number
NETSER	WEM	STOP	William E. Matson general network stop code
KLSER	WPT	HALT	Wrong parity trap
CLOCK1	WTP	DEBUG	Wrong type of PDL
LOKCON	XPW	STOP	Exchanged page went away
SCHED1	XTH	DEBUG	XJOB too high
REFSTR	ZBC	DEBUG	Zero blocks per cluster

*Deleted in Version 7.01

GENERAL INFORMATION
PIP (Peripheral Interchange Program) is a utility program which is used to transfer files between standard peripheral devices. PIP can also perform editing and magtape control functions during file transfers.

R PIP <CR>	Monitor commmand to load and start PIP
*	Prompt - indicates PIP is ready to accept commands
↑C	Exit PIP - return to monitor command mode

Notes

1. This module is a summary of PIP intended for use by field engineers. Refer to the Software Notebooks for a complete description.
2. Wild characters, the asterisk (*) and question mark (?) may be used in filename and extension construction.
3. Octal constants may be used in filenames and extensions. The octal constant must be preceded by a pound sign (#) and delimited by a nonoctal digit or a character.
4. Including the "/X" switch in a command string will cause PIP to transfer each file separately (file by file) to the destination device.
5. Excluding the "/X" switch from the command string will cause PIP to combine (concatenate) the specified source files into one large file on the destination device.

COMMAND CONVENTIONS AND SWITCHES
PIP command conventions and switches are described in the following tables.

Table 1 PIP Command Conventions
Table 2 PIP Command String Delimiters
Table 3 PIP Acceptable Device Mnemonics
Table 4 File Protection Codes
Table 5 UFD and SFD Protection Codes
Table 6 PIP Control Switch Summary
Table 7 PIP Magtape Switch Summary

PIP Command String Format
A PIP command string consists of two fields separated by an equal sign (=) and terminated by a carriage return <CR>.

A PIP command string which is used to transfer files between I/O devices has the following format:

```
DESTINATION = SOURCE <CR>

dev:file.ext/s/s[p,pn]<nnn>↑ident↑=dev:file.ext[p,pn]<CR>
```

A PIP command string which does not transfer files (i.e., move magtape) has the following format:

```
DESTINATION = <CR>

MTA3:(MU)=<CR>
```

The equal sign delimiter and a terminator are still required in commands formatted in this manner despite the fact that only the DESTINATION portion of the command is used.

The DESTINATION portion of a PIP command describes the device and file(s) which is to receive the transferred data. This portion of a command consists of one file specification.

The SOURCE side of the command describes the device from which the transferred data is to be taken. This portion of a command may contain one or more file specifications.

PIP command strings may be of any length; both upper and lower case characters may be used. PIP commands are normally terminated and the requested operation initiated by a carriage return. However, an ALTMODE, ESC, line feed, vertical TAB, or form feed can also be used as a command terminator.

Table 1 PIP Command Conventions

Convention	Description
dev:	Either a physical or a logical device name. Refer to Table 3.
[directory]	The identifier of a specific directory (i.e., UFD or MFD) within the system. This identifier may consist of a project, programmer number pair and Sub File Directory (SFD) names.
.ext	A 1 to 3 character alphanumeric extension assigned to the name of a file either by the user or by the system.
file	A 1 to 6 character alphanumeric identification which is either to be assigned to a new file (when on the destination side of the command) or which identifies an existing file (when on the source side of the command).
↑ident↑	A 1 to 6 character name which is to be given to the contents of a DECtape reel mounted on a specified DECtape unit.
<nnn>	A 3-digit protection code which is to be assigned to either one or more destination files or to a specified User File Directory. Refer to Table 4 and Table 5 respectively.
/s	Switches which affect the transfer. All switches in a PIP command string must be preceded by a slash - e.g., /sw/sw - or enclosed in parentheses - e.g., (sw/sw). Refer to Table 6 for a summary of PIP switches.

Table 2 PIP Command String Delimiters

Delimiter	Use and Description
:	The colon delimiter follows and identifies a device name. For example, the device DTA1 is specified as DTA1: in PIP commands.
[]	Square brackets are used to enclose the user DIRECTORY numbers and SFD names (if SFDs are used). For example [40,633] or [40,633,SFD1,SFD2...SFDn] represent the manner in which DIRECTORY numbers can be written.
<>	Angle brackets must be used to enclose a protection code (e.g. <057>) which is to be assigned to either a file or a user file directory (UFD).
,	Commas are used to separate user project and programmer numbers, and file specification groups. For example: dev:[40,633]=dev:file.ext,file.ext<CR>
↑↑	A name to be assigned as an identifier to a DECtape is enclosed within a set of up-arrows (e.g. ↑MACFLS↑).
.	A period delimiter must be the first character of a filename extension. The form on an extension is (.ext).
#	A number symbol is used as a flag to indicate the presence of an octal constant in a filename or a filename extension.
!	An exclamation symbol may be used to delimit a file specification. When used, the ! symbol causes control to be returned to the monitor from PIP and the specified file (or program) to be loaded and run. This function is provided as a user convenience to eliminate the need for several control entries.

Table 2 PIP Command String Delimiters (Cont)

Delimiter	Use and Description
=	The equal sign must be used to separate the destination and source portions of a PIP command.
()	Parentheses are used to enclose magnetic tape options, PIP control switches, and one or more PIP function switches. The form of a command employing parentheses to enclose a series of switches is: dev:file.ext(sw1sw2..swn)=...<CR>

Table 3 PIP Acceptable Device Mnemonics

Mnemonic	Device
CDP	Card Punch
CDR	Card Reader
CTY	Console TTY
DTA	DECtape
DSK	Disk
DPx	Packs
FXx	Fixed-Head
DIS	Display
LPT	Line Printer
MTA	Magnetic Tape
OPR	Operator Terminal
PTP	Paper Tape Punch
PTR	Paper Tape Reader
PLT	Plotter
PTY	Pseudo-TTY
SYS	System Library
TTY	Terminal
TMP	Pseudo-device TMPCO

Table 4 File Protection Codes

Code	Permitted Operations
0	Change protection, rename, write, update, append, read, execute.
1	Rename, write, update, append, read, execute.
2	Write, update, append, read, execute.
3	Update, append, read, execute.
4	Append, read, execute.
5	Read, execute.
6	Execute only.
7	No access privileges. File may be looked up if the UFD permits.

Table 5 UFD and SFD Protection Codes

Code	Permitted Operations
0	Access not permitted.
1	The directory may be read as a file.
2	CREATEs are permitted.
3	The directory may be read as a file and CREATEs are permitted.
4	LOOKUPs are permitted.
5	The directory may be read as a file and LOOKUPs are permitted.
6	CREATEs and LOOKUPs are both permitted.
7	The directory may be read as a file and both CREATEs and LOOKUPs are permitted.

Table 6 PIP Control Switch Summary

Switch	Description
/DX	Copy all but specified files
/F	List disk or DTA directory (filenames and ext. only).
/G	Ignore I/O errors.
/H	Image binary processing (mode)
/I	Image processing (mode)
/J	Punch cards in ASCII (output device must be CDP) or convert control characters on terminal output.
/L	List directory.
/N	Delete sequence numbers.
/O	Same as /S switch, except increment is by 1.
/P	FORTRAN output conversion assumed. Convert format control character for line printer listing. /B/P FORTRAN binary.
/Q	Print (this) list of switches and meanings.
/R	Rename file.
/S	Resequence, or add sequence number to file; increment is by 10.
/T	Suppress trailing spaces only.
/U	Copy block 0 (DTA).
/V	Match and count angle brackets (<>).
/W	Convert TABs to multiple spaces.
/X	Copy specified files. (The DX switch tells PIP to copy all but specified files.)
/Y	DECtape to paper tape - If extension is: RMT - A RIM10B paper tape (with terminating transfer word) is produced RTB - A RIM10B paper tape (with RIM loader and terminating transfer word) is produced SAV - A RIM10B paper tape is produced (with neither RIM loader nor terminating transfer word)
/Z	Zero out directory

Table 7 PIP Magtape Switch Summary

Switch	Description
(M2)	Select 200 BPI density.
(M5)	Select 556 BPI density.
(M8)	Select 800 BPI density.
(MA)	Advance MTA one file.
(M#nA)	Advance MTA n files.
(MB)	Backspace MTA one file.
(M#nB)	Backspace MTA n files.
(MD)	Advance MTA one record.
(M#nD)	Advance MTA n records.
(ME)	Select Even Parity.
(MF)	Mark EOF.
(MP)	Backspace MTA one record.
(M#nP)	Backspace MTA n records.
(MT)	Skip to logical EOT.
(MU)	Rewind and unload MTA or DTA.
(MW)	Rewind MTA or DTA.

Examples
The following are examples of commonly used PIP command strings:

EX1 - PIPing an ASCII file from the DISK to the line printer

LPT:=DSK:ERROR.SYS<CR>

EX2 - Combines two files on disk into one file on DECtape:

DTA1:FILCOM.MAC=DSK:FILA.MAC,FILB.MAC<CR>

EX3 - Copies a paper tape

PTP:=PTR:<CR>

EX4 - Specifies that the DECtape on DTA3 be given the identifier "MYFILE" and receive a copy of each file on DTA1.

DTA3:↑MYFILE↑/X=DTA1:*.*<CR>

Tops-20

Table of Contents

TOPS-20 SYSTEM PROGRAM LIBRARY
The programs in TOPS-20 System Program Library are listed and described in Table 1.

Table 1 TOPS-20 System Program Library

Program	Description
ACTGEN	ACTGEN is an account generator program used to create and install an account validation data base for use by TOPS-20 in validating accounts. It is intended primarily for use by the system manager and operator. Wheel or operator capabilities must be enabled to run ACTGEN. ACTGEN is documented in the DECSYSTEM-20 System Manager's Guide.
BOOT	BOOT is used to load the TOPS-20 monitor from disk into KL10 memory. On normal system startup, BOOT is automatically loaded and started by RSX20F, and will load the TOPS-20 monitor without operator intervention. BOOT is also responsible for dumping KL10 memory after system malfunction, for later analysis. BOOT is documented in the following documents: DECSYSTEM-20 Software Installation Guide DECSYSTEM-20 Operator's Guide
CHECKD	CHECKD checks TOPS-20 disk file structure and bit table for consistency. In the process of checking the directory structure, CHECKD finds all disk space which is in use; this allows CHECKD to compute the disk pages lost. CHECKD can optionally release this lost space. CHECKD can also be used to completely rebuild the disk bit table or to scan the directory structure for a specified disk address. CHECKD may also be used to create new file structures. CHECKD is documented in the DECSYSTEM-20 Operator's Guide.
CHKPNT	CHKPNT has three major functions: 1. Compile account statistics on disk space utilization 2. Set the monitor checkpoint interval 3. Copy system-generated accounting data into the accounting file. CHKPNT is documented in the TOPS-20 Operator's Guide.
CNVDSK	System Utility converts file system to permit archiving files
CREF	CREF takes the modified listing files produced by the language processors and produces a final, printable listing with cross reference tables appended. CREF is documented in the DECSYSTEM-20 User's Guide.
DDT	DDT is a symbolic assembly language debugger. DDT allows up to 8 breakpoints as well as symbolic patching and manipulation of various datatypes.
DLUSER	DLUSR is a program which obtains identifying information about each directory on a system and places it in a file. The program can then use this file to create the same directories later, in the event of a system rebuild. DLUSER is documented in the DECSYSTEM-20 Operator's Guide and DECSYSTEM-20 System Manager's Guide.
DUMPER	DUMPER is a program for saving and restoring disk files using magtape. It is used by operations personnel for file system maintenance, and may be employed by users who wish to keep certain files on magtape and/or transfer them between systems.

Table 1 TOPS-20 System Program Library (Cont)

Program	Description
DX20LD	DX20 Microcode Loader
EDIT	EDIT is a line-oriented editor which is used to create and edit text files. It resembles the TOPS-10 editor SOS in function and command structure.
FE	FE is a utility for file transfers between the TOPS-20 file system and the FILES-11 file system. It handles protocol for the FE device such that FE: can be addressed as a FILES-11 device, usually through 11 PIP. **CAUTION** **The FE device is intended for use only in software development and updating procedures by knowledgeable people. Use without proper caution may produce unpredictable results.** FE depends on the existence of the RSX-20F task T20ACP, which should reside on the -11 file system as T20ACP.TSK. Use of FE and file conversion procedures are described in the Guide To Using the FE Device, USEFE.MEM.
FILCOM	The FILCOM program compares two files and outputs the differences between them. With FILCOM you may compare both ASCII files and binary files. FILCOM compares ASCII files line by line and binary files word by word.
FORMAT	FORMAT provides the mechanism for formatting and/or verifying RP04, RP05, RP06 disk packs that are configured to RH20s. FORMAT produces a pack in the identical format to one that was created using the diagnostic, DDRPI. FORMAT runs during timesharing only, while DDRPI can FORMAT in stand-alone mode only.
GALAXY	GALAXY is the Batch and Spooling Subsystem for the DECsystem-10 and DECSYSTEM-20. GALAXY comprises all the software (excluding operating systems software) necessary to do batch processing and input and output spooling and all queue management and task scheduling required for those functions. GALAXY Release 3 consists of the following programs: **Program** — **What It Does** QUASAR — Central queue manager, task scheduler, and GALAXY system controller BATCON — Batch job processor LPTSPL — Lineprinter output spooler (unspooler) SPRINT — Card reader input stacker/spooler QUENCH — Timesharing users' interface to the GALAXY system QMANGR — Interface module for FOROTS, BASIC, etc.
INFO	System Utility for Inter-Program Communication
LINK	LINK is the linking loader for the DECSYSTEM-20. OVRLAY is the overlay handler for the DECSYSTEM-20. LINK and OVRLAY are documented in the DECSYSTEM-20 User's Guide and in the DECSYSTEM-20 LINK User's Guide.
MACRO	Symbolic Assembler
MACSYM	Symbol Parameter Files

Table 1 TOPS-20 System Program Library (Cont)

Program	Description
MAIL	MAIL is a program which allows users to send messages to other users. Messages sent by MAIL are stored in the receiver's disk directory so that they may be referenced when convenient. MAIL depends on the programs INFO and MAILER to perform its stated tasks. Also, the program RDMAIL is used by message recipients to read messages. MAIL is documented in the TOPS-20 User's Guide.
MAKDMP	Create DUMP.EXE file for memory system image on system crash.
MAKLIB	MAKLIB is used to update and index .REL files. MAKLIB will insert, delete or replace modules. It is also used to index FORLIB.REL and LIBOL.REL to speed up the loading process. MAKLIB is documented in the DECSYSTEM-20 User's Guide.
MAKRAM	MAKRAM is a program to generate LP20 translation RAM files. MAKRAM commands are described in MAKRAM.HLP.
MAKVFU	MAKVFU is a program to generate LP05 Direct Access Vertical Format files. MAKVFU commands are described in MAKVFU.HLP.
MAPPER	Performance tuning tool
MONSYM	Symbol Parameter Files
MOUNTR	Labeled tape handler
MTBOOT	Tape Bootstrap
OPLEAS	OPLEAS is the program that enables the operator to talk to users running PLEASE. Requests for contact with the operators are queued; thus the user can type a request for operator action and know that the request will be received even if the operator is currently busy. OPLEAS also handles structure and tape mount requests submitted via the EXEC TMOUNT and SMOUNT commands. OPLEAS is documented in the TOPS-20 User's Guide.
PA1050	RA1050 is the TOPS-10 UUO simulator produced from the file PAT.MAC. It gets mapped into the address space of any program that executes a TOPS-10 UUO. Its function is to intercept all TOPS-10 UUOs and simulate them with the appropriate TOPS-20 JSYSs.
PLEASE	PLEASE provides a facility for one user at a time to talk to an operator. Requests for contact with the operator are queued; thus the user can type a request for operator action and know that the request will be received even if the operator is currently busy. PLEASE runs in conjunction with OPLEAS. PLEASE is documented in the TOPS-20 User's Guide.
PTYCON	PTYCON is a pseudoteletype (PTY) controller. It allows a user multiple job control from a single terminal. PTYCON provides the means to converse with a number of subjobs and to control the manner and times when output is received from the subjobs.
REAPER	Disk space maintenance utility
RDMAIL	RDMAIL is a program which allows a user to read the messages which have been sent to him. It always reads the messages from the file MAIL.TXT. RDMAIL is documented in the DECSYSTEM-20 User's Guide.
RMS	Record Management Services for BASIC-PLUS-2, COBOL-74

Table 1 TOPS-20 System Program Library (Cont)

Program	Description
RUNOFF	RUNOFF is a text-processing program. RUNOFF will format input text, generate tables, build lists, handle page and section numbering. RUNOFF allows a user to make all sorts of changes to the text of a document and still produce a clean, well-formatted result. RUNOFF is documented in Getting Started with Runoff.
SETSPD	SETSPD is a privileged system program which processes the 3-CONFIG.CMD file and, in so doing, sets many initial parameters about the system such as initial line speeds, system logical names, and magtape logical to physical correspondences.
SPEAR	SPEAR is a library of functions that sorts, evaluates, and reports on events recorded in the local system event file.
SYSJOB	SYSJOB is a program for controlling system background programs. It is normally started only by job 0, and it creates additional processes and jobs as necessary. An operator or other privileged job may pass commands to SYSJOB via an exec command (↑E) SPEAK to affect the status of the background programs. SYSJOB is documented in the DECSYSTEM-20 Operator's Guide under the (↑E) SPEAK command.
TGHA	MF20 on-line diagnostic/utility
TV	Video Text Editor
UETP	User Environment Test Package
ULIST	ULIST provides a mechanism for listing user and directory information. The listing may be directed to the printer, the user's terminal, or to a file. ULIST will provide information on user and directory groups, directory numbers, quotas, and protections, and will list user passwords if desired.
VERIFY	Installation verifier
WATCH	WATCH is a system program which provides a list of various system statistics and job run times upon request. A user can thus periodically check system performance with this utility.

The following unsupported software (binary and source) is distributed with TOPS-20 and is provided on an "as is" basis without DIGITAL warranty express or implied.

USAG20	Accounting Utility for USAGE.OUT
USAH20	Requires FORTRAN/COBOL/SORT License
SYSDPY	System Performance Tool
DDT11	Debugging tool for FE communication software
SED	Screen Editor
ALU	Source Control Utility
REDIT	Source Edit Utility
REV	File Manipulation Utility
BLIS10	BLISS-10 Compiler
FEDDT	Debugger for front-end dumps

TOPS-20 COMMAND LANGUAGE
The TOPS-20 Operating System supports approximately 70 basic commands. These commands are described in Table 2.

Special symbols and control characters used by TOPS-20 are described in Table 1.

COMMAND FORMAT
TOPS-20 commands use the following format.

COMMAND$(guide word)ARG$(guide word)ARG$(...<CR>

The base command and each argument is delimited by an altmode (ESCAPE KEY). The command string is terminated by a carriage return <CR>.

ERROR MESSAGES
Table 3 lists and describes many of the most commonly used BUGCHKS and BUGHLTS associated with a TOPS-20 operating system. The list was taken from TOPS-20 BIG SYSTEM, TOPS-20 MONITOR 3A (2013). A complete list for any given TOPS-20 operating system may be printed by typing

PRINT PS:<SYSTEM>BUGSTRING.TXT<CR>

Table 1 TOPS-20 Symbols and Control Characters

Character	Description
^C^C	Two control C characters will return the terminal to monitor command level.
@	Prompt - A single @ sign indicates the monitor is at command level and ready to accept commands.
,<CR>	A command and carriage return typed following a command name causes the monitor to enter subcommand level for the command named.
@@	Prompt - A double @@ sign indicates the monitor is at a subcommand level and ready to accept subcommands only.
<CR>	A single carriage return terminates a command or subcommand.
<CR><CR>	A double carriage return terminates a subcommand and returns the monitor to command level.
?	A question mark typed at the command level or subcommand level will cause the monitor to print a list of the available commands. A question mark typed following a partially typed command will cause the monitor to print a list of all commands or subcommands which begin with the characters typed. A question mark typed following a guide word will cause the monitor to print a list of the possible arguments. A question mark printed by the monitor indicates the user has made an error in typing a command.
$ (altmode) (ESCAPE)	If there is no ambiguity in a partially typed command, pressing the ESCAPE key will cause the remaining characters and the first guide word of the command to be printed. If a partially typed command is ambiguous pressing the ESCAPE key will cause the terminal bell to ring. The ESCAPE key is also used to terminate an argument and causes the next guide word to be printed.

Table 1 TOPS-20 Symbols and Control Characters (Cont)

Character	Description
RUBOUT DELETE	The RUBOUT or DELETE key will cause the last DELETE character typed to be deleted.
↑W	Typing a control W will cause the last field typed to be deleted.
↑U	Typing a control U will cause the entire command line to be deleted.
↑R	Typing a control R will cause the current command line to be reprinted.
↑O	Typing a control O will stop the current printout.
!	The exclamation mark is used to delimit text following a command. This is useful for sending messages during a KLINIK linkup.

Table 2 TOPS-20 Command Summary

Command	Description
System Access Commands	
ATTACH	Connects your terminal to a designated job. See also: DETACH, UNATTACH
DETACH	Disconnects your terminal from the current job without affecting the job. See also: ATTACH, UNATTACH
DISABLE	Returns a privileged user to normal status. See also: ENABLE
ENABLE	Permits privileged users to access and change confidential system information. See also: DISABLE
LOGIN	Gains access to the TOPS-20 system. See also: LOGOUT
LOGOUT	Relinquishes access to the TOPS-20 system. See also: LOGIN
UNATTACH	Disconnects a terminal from a job; it does not have to be the terminal you are using. See also: ATTACH, DETACH
Information Commands	
DAYTIME	Prints the current date and time of day.
INFORMATION	Provides information about your job, files, memory, errors, system status, and many other parameters.
SYSTAT	Outputs a summary of system users and available computing resources.
Terminal Commands	
ADVISE	Sends whatever you type on your terminal as input to a job connected to another terminal. See also: BREAK, RECEIVE, REFUSE, TALK
BREAK	Clears terminal links and advising links. See also: ADVISE, RECEIVE, REFUSE, TALK
RECEIVE	Allows your terminal to receive links and advice from other users. See also: ADVISE, BREAK, REFUSE, TALK
REFUSE	Denies links and advice to your terminal. See also: ADVISE, BREAK, RECEIVE, TALK
SET	Declares certain action to be taken when errors are detected in TOPS-20 commands.

Table 2 TOPS-20 Command Summary (Cont)

Command	Description
TAKE	Accepts commands from a file, just as if you had typed its contents on your terminal.
TALK	Links two terminals so that each user can observe what the other user is doing, yet does not affect the other user's job. See also: ADVISE, BREAK, RECEIVE, REFUSE
TERMINAL	Declares the hardware type of terminal you have, and lets you inform TOPS-20 of any special characteristics of the terminal.
Device Handling Commands	
ASSIGN	Reserves a device for use by your job. See also: DEASSIGN, DEFINE
BACKSPACE	Moves a magnetic tape drive back any number of records or files. See also: REWIND, SKIP, UNLOAD
DEASSIGN	Releases a previously assigned device. See also: ASSIGN
EOF	Writes an end-of-file mark on a magnetic tape.
REWIND	Positions a magnetic tape backward to its load point. See also: BACKSPACE, SKIP, UNLOAD
SKIP	Advances a magnetic tape one or more records or files. See also: BACKSPACE, REWIND, UNLOAD
UNLOAD	Rewinds a magnetic tape until the tape is wound completely on the source reel. See also: BACKSPACE, SKIP, REWIND
File Systems Commands	
ACCESS	Grants ownership and group rights to a specified directory. See also: CONNECT, END-ACCESS
APPEND	Adds information from one or more source files to an existing disk file. See also: EDIT
CLOSE	Closes a file or files left open by a program.
CONNECT	Removes you from your current directory and connects you to a specified directory.
COPY	Duplicates a source file in a destination file.
CREATE	Starts EDIT for making a new file. See also: EDIT
DELETE	Marks the specified file(s) for eventual deletion (disk files only) or deletes the specified files (all other devices). See also: EXPUNGE, UNDELETE
DEFINE	Associates a logical name with one or more file names. See also: ASSIGN
DIRECTORY	Lists the names of files residing in the specified directory and information relating to those files. See also: FDIRECTORY, TDIRECTORY, VDIRECTORY
EDIT	Starts EDIT for changing an existing file. See also: APPEND, CREATE
EXPUNGE	Permanently removes any deleted files from the disk. See also: DELETE, UNDELETE

Table 2 TOPS-20 Command Summary (Cont)

Command	Description
END-ACCESS	Relinquishes ownership rights to a specified directory. See also: ACCESS
FDIRECTORY	Lists all the information about a file or files. See also: DIRECTORY, TDIRECTORY, VDIRECTORY
LIST	Prints one or more files on the line printer with or without formatting. See also: PRINT, TYPE
PRINT	Lists one or more files on the line printer. See also: LIST, TYPE
QUEUE	Places an entry into or examines a specified queue, for example, the line printer output queue.
RENAME	Changes one or more descriptors of an existing file specification.
SDISMOUNT	Notifies the system that the given structure is no longer needed. See also: SMOUNT, SREMOVE
TDIRECTORY	Lists the names of all files in the order of the date and time they were last written. See also: DIRECTORY, FDIRECTORY, VDIRECTORY
SMOUNT	Requests that a structure be made available to the user. See also: SDISMOUNT, SREMOVE
TYPE	Types the specified files on your terminal. See also: PRINT, LIST
SREMOVE	Makes a structure unavailable and requests its removal. See also: SDISMOUNT, SMOUNT
UNDELETE	Restores one or more disk files marked for deletion. See also: DELETE, EXPUNGE
TMOUNT	Requests that a magnetic tape be made available to the user.
VDIRECTORY	Lists the names of all files, as well as their protection, size, and date and time they were last written. See also: DIRECTORY, FDIRECTORY, TDIRECTORY
Program Control Commands	
COMPILE	Translates a source program using the appropriate compiler. See also: DEBUG, EXECUTE, LOAD, MERGE
CONTINUE	Resumes execution of a program interrupted by a control C. See also: REENTER, START
CREF	Runs the CREF program which produces a cross-reference listing and automatically sends it to the line printer.
CSAVE	Saves the program currently in memory so that it may be used by giving a RUN command. The program is saved in a compressed format. See also: SAVE
DDT	Merges the debugging program, DDT, with the current program and then starts DDT. See also: DEBUG, MERGE
DEBUG	Takes a source program, compiles it, loads it with DDT and starts DDT. See also: COMPILE, DDT, MERGE

Table 2 TOPS-20 Command Summary (Cont)

Command	Description
EXECUTE	Translates, loads, and begins execution of a program. See also: COMPILE, LOAD
FORK	Makes the TOPS-20 language work for a particular address space.
GET	Loads an executable program from the specified file. See also: LOAD
LOAD	Translates a program and loads it into memory. See also: EXECUTE
MERGE	Loads an executable program into memory and merges it with the current contents of memory. See also: DEBUG
POP	Stops a copy of the TOPS-20 Command Language and returns control to the previous copy of the Command Language. See also: PUSH
PUSH	Starts a new copy of the TOPS-20 Command Language. See also: POP
R	Runs a system program. See also: EXECUTE, GET, LOAD, RUN, START
REENTER	Starts the program currently in memory at an alternate entry point specified by the program. See also: CONTINUE, START
RESET	Clears the job to which your terminal is currently attached.
RUN	Loads an executable program from a file and starts it at the location specified in the program. See also: EXECUTE, GET, LOAD, START
SAVE	Copies the contents of memory into a file in executable format. If memory contains a program, you may now execute the program by giving the RUN command with the proper file specification. See also: CSAVE
START	Begins execution of a program at the location specified in the entry vector. See also: CONTINUE, EXECUTE, GET, LOAD REENTER
Batch Commands	
SUBMIT	Enters a file into the Batch waiting list. When it is your job's turn, the commands contained in the file are executed.

Table 3 lists the BUGHLT and BUGCHK codes. For more information refer to TOPS-20 BUGHLT Documentation of the TOPS-20 Software Notebooks.

Table 3 BUGHLT and BUGCHK Names and Descriptions

Name	Type	Description
ABKSKD	[HLT]	ADDRESS BREAK FROM SCHEDULER CONTEXT
ACTBBD	[CHK]	VERACT - ACCOUNT FILE CORRUPTED
APRAPE	[HLT]	ADDRESS PARITY ERROR
APRNX1	[HLT]	NXM DETECTED BY APR
APRNX2	[HLT]	NXM DETECTED BY APR
ARCASS	[CHK]	ARCF: File dir # and mapped dir # do not match
ARCVER	[CHK]	ARCMSG: NOUT failed
ARSTXX	[HLT]	ARRST: FDB disappeared for destination file
ASAASG	[CHK]	DSKASA - ASSIGNING ALREADY ASSIGNED DISK ADDRESS
ASGBAD	[CHK]	DSKASA - ASSIGNING BAD DISK ADDRESS
ASGBPG	[CHK]	INIBTB-FAILED TO ASSIGN BAD PAGE(S)
ASGFR0	[HLT]	ASGFRE - ILLEGAL TO ASSIGN 0 FREE SPACE
ASGINT	[CHK]	ASGFRE CALLED OKINT
ASGREP	[CHK]	ILLEGAL PRIORITY GIVEN TO ASGRES
ASGREQ	[CHK]	ILLEGAL POOL NUMBER GIVEN TO ASGRES
ASGSW2	[HLT]	SWPOMG-CANNOT ASSIGN RESERVED DRUM ADDRESS
ASGSWB	[CHK]	SWPINI-CANNOT ASSIGN BAD ADDRESS
ASOFNF	[HLT]	DELFIL: ASGOFN GAVE FAIL RETURN FOR LONG FILE XB
ASTJFN	[HLT]	GETFDB: CALLED FOR JFN WITH OUTPUT STARS
BADBAK	[CHK]	FILIN2 - BACKUP COPY OF ROOT DIRECTORY IS NOT GOOD
BADBAT	[CHK]	BAT BLOCKS UNREADABLE
BADDAC	[HLT]	INSACT - NULL ACCOUNT STRING SEEN
BADIDX	[CHK]	IDXINI: PARTIALLY UNSUCCESSFUL INDEX TABLE REBUILD
BADPTR	[HLT]	BAD SECTION POINTER - SECMAP
BADREC	[HLT]	FILINI - Reconstruction of ROOT-DIRECTORY failed
BADROT	[HLT]	FILIN2: ROOT-DIRECTORY IS INVALID
BADTAB	[CHK]	VERACT - SPURIOUS HASH TABLE ENCOUNTERED
BADTTY	[HLT]	TRANSFER TO NONEXISTENT TTY CODE
BADTYP	[HLT]	BAD LABEL FIELD DESC
BADXT1	[HLT]	INDEX TABLE MISSING AND CAN NOT BE CREATED
BADXT2	[CHK]	INDEX TABLE MISSING AND WAS CREATED
BADXTB	[HLT]	FILIN2: Could not initialize index table
BKUPDF	[HLT]	BKUPD - BAD CST1 ENTRY OR INCONSISTENT CST
BLKF1	[CHK]	BYTINA: BLKF SET BEFORE CALLING SERVICE ROUTINE
BLKF2	[CHK]	BYTOUA: BLKF SET BEFORE CALL TO SERVICE ROUTINE
BLKF3	[CHK]	CLZDO: BLKF SET BEFORE CALL TO SERVICE ROUTINE
BLKF4	[CHK]	.GDSTS: BLKF SET BEFORE CALL TO DEVICE ROUTINE
BLKF5	[CHK]	.MTOPR: BLKF SET BEFORE CALL TO DEVICE ROUTINE
BLKF6	[CHK]	.SDSTS: BLKF SET BEFORE CALL TO DEVICE ROUTINE
BOOTCR	[HLT]	GETSWM - NOT ENOUGH CORE FOR SWPMON
BOOTER	[HLT]	GETSWM - ERROR LOADING SWPMON
BOOTLK	[HLT]	GSMDSK - FAILED TO LOCK NEEDED PAGES
BOOTMP	[HLT]	GSMDSK - CANNOT MAP BOOTSTRAP PAGES
BTBCR1	[HLT]	FILINI - NO BIT TABLE FILE AND UNABLE TO CREATE ONE
BTBCRT	[HLT]	FILINI - COULD NOT INITIALIZE BIT TABLE FOR PUBLIC STRUCTURE
CDILVT	[HLT]	ILLEGAL DEVICE FUNCTION CODE
CKPLEN	[CHK]	USGINI - ILLEGAL CHECKPOINT ENTRY LENGTH
CLZABF	[CHK]	CLZFFW: SERVICE ROUTINE BLOCKED ON AN ABORT CLOSE
CRDBAK	[CHK]	CRDIR3: COULD NOT MAKE BACKUP COPY OF ROOT-DIRECTORY
CRDBK1	[CHK]	CRDIR4: COULD NOT MAKE BACKUP COPY OF ROOT-DIRECTORY
CRDNOM	[CHK]	CRDIR-FAILED TO MAKE MAIL.TXT FILE
CRDOLD	[CHK]	CRGDGB: OLD FORMAT CRDIR IS ILLEGAL
CRDSDF	[CHK]	CRDIR1: SETDIR FAILED ON NEW DIRECTORY
CRSPAG	[CHK]	VERACT - ACCOUNT DATA BLOCK CROSSES A PAGE BOUNDARY
CST2I1	[HLT]	PAGE TABLE CORE POINTER AND CST2 FAIL TO CORRESPOND
CST2I2	[HLT]	MVPT-CST2 INCONSISTENT
CST2I3	[HLT]	PAGE TABLE CORE POINTER AND CST2 FAIL TO CORRESPOND
DEABAD	[CHK]	DSKDEA - DEASSIGNING BAD DISK ADDRESS
DEAUNA	[CHK]	DEDSK-DEASSIGNING UNASSIGNED DISK ADDRESS
DELBDD	[INF]	DELDIR: BAD DIRECTORY DELETED. REBUILD BIT TABLE
DEVUCF	[CHK]	DEVAV - UNEXPECTED CHKDES FAILURE
DGUTPG	[HLT]	DIAG - LOCKED PAGE LIST PAGE LOCKED AT DIAG UNLOCK
DGZTPA	[HLT]	DIAG - LOCKED PAGE LIST PAGE WAS ZERO
DIRACT	[CHK]	ACTBAD: ILLEGAL FORMAT FOR DIRECTORY ACCOUNT BLOCK IN DIRECTORY:
DIRB2L	[CHK]	RLDFB2: DIRECTORY FREE BLOCK TOO LARGE IN DIRECTORY:
DIRB2S	[CHK]	RLDFB1: DIRECTORY FREE BLOCK TOO SMALL IN DIRECTORY:
DIRBAD	[CHK]	SETDI4: SMASHED DIRECTORY NUMBER:
DIRBAF	[CHK]	RLDFB5: BLOCK ALREADY ON DIRECTORY FREE LIST IN DIRECTORY:
DIRBCB	[CHK]	RLDFB3: DIRECTORY FREE BLOCK CROSSES PAGE BOUNDARY IN DIRECTORY:
DIRBLK	[CHK]	BLKSCN: ILLEGAL BLOCK TYPE IN DIRECTORY:
DIRDNL	[CHK]	ULKDIR-DIRECTORY NOT LOCKED, DIRECTORY NUMBER:
DIREXT	[CHK]	EXTBAD: ILLEGAL FORMAT FOR DIRECTORY EXTENSION BLOCK IN DIRECTORY:
DIRFDB	[CHK]	ILLEGAL FORMAT FOR FDB IN DIRECTORY:
DIRFKP	[CHK]	SETDIR-DIR PAGE 0 BELONGS TO FORK IN DIRECTORY:

Table 3 BUGHLT and BUGCHK Names and Descriptions (Cont)

Name	Type	Description
DIRFRE	[CHK]	FREBAD: ILLEGAL FORMAT FOR DIRECTORY FREE BLOCK IN DIRECTORY:
DIRIFB	[CHK]	RLDFB4: ILLEGAL BLOCK TYPE ON DIRECTORY FREE LIST IN DIRECTORY:
DIRNAM	[CHK]	NAMBAD: ILLEGAL FORMAT FOR DIRECTORY NAME BLOCK IN DIRECTORY:
DIRPG0	[CHK]	DR0CHK: ILLEGAL FORMAT FOR DIRECTORY PAGE 0 IN DIRECTORY:
DIRPG1	[CHK]	DRHCHK: DIRECTORY HEADER BLOCK IS BAD IN DIRECTORY:
DIRRHB	[CHK]	RLDFB6: ATTEMPTING TO RETURN A HEADER BLOCK IN DIRECTORY:
DIRSY1	[CHK]	DELDL8: DIRECTORY SYMBOL TABLE FOULED UP FOR DIRECTORY:
DIRSY2	[CHK]	MDDNAM: SYMBOL TABLE FOULED UP IN DIRECTORY:
DIRSY3	[CHK]	LOOKUP: SYMBOL SEARCH FOULED UP IN DIRECTORY:
DIRSY4	[CHK]	NAMCM4: DIRECTORY SYMBOL TABLE FOULED UP IN DIRECTORY:
DIRSY5	[CHK]	SYMBAD: ILLEGAL FORMAT FOR DIRECTORY SYMBOL TABLE IN DIRECTORY:
DIRSY6	[CHK]	RBLDST: PREMATURELY RAN OUT OF ROOM IN SYMBOL TABLE IN DIRECTORY:
DIRULK	[CHK]	ULKMD2: ATTEMPT TO UNLOCK ILLEGALLY FORMATTED DIR, DIR NUMBER:
DIRUNS	[CHK]	UNSBAD: ILLEGAL FORMAT FOR DIRECTORY USER NAME BLOCK IN DIRECTORY:
DLDEF	[INF]	LOGICAL NAME DEFINE FAILED FOR FE CTY
DN20ST	[INF]	DTESRV- DN20 STOPPED
DRMFUL	[CHK]	DRUM COMPLETELY FULL
DRMIBT	[HLT]	DRMASN-BIT TABLE INCONSISTENT
DRMNFR	[HLT]	DRMAM-CANNOT FIND PAGE WHEN DRMFRE NON-0
DSKBT1	[CHK]	DSK BIT TABLE FOULED, CANNOT FIND FREE PAGE ON TRACK WITH NON-0 COUNT
DSKBT3	[CHK]	DISK BIT TABLE ALREADY LOCKED AT LCKBTB
DST2SM	[HLT]	SWPINI-DST TOO SMALL
DTECAR	[CHK]	CARRIER FNC WITH NO LINE NUMBER
DTECDM	[INF]	DTESRV- TO -10 COUNTS DO Not MATCH
DTEDAT	[CHK]	TAKTOD- ILLEGAL FORMAT FOR TIME/DATE
DTEDEV	[CHK]	ILLEGAL DEVICE
DTEDIN	[INF]	DTESRV- TO -10 IN PROGRESS ON DOORBELL
DTEDME	[INF]	DTESRV- ZERO Q COUNT
DTEERR	[CHK]	DTESRV-DTE DEVICE ERROR
DTEIDP	[CHK]	BAD INDIRECT PACKET
DTEIFR	[CHK]	DTESRV-ILLEGAL FNC REQUEST
DTELPI	[INF]	DTECHK- DTE LOST PI ASSIGNMENT
DTEMCC	[CHK]	DOFRGM-DN20 DISAGRESS WITH COUNT
DTEODD	[CHK]	TAKLC-ODD BYTE COUNT FOR LINE CHARACTERS
DTEP2S	[CHK]	TO10DN-PACKET TOO SMALL
DTEPGF	[CHK]	DTE TRANSFER PAGE FAIL
DTEPNR	[INF]	DTESRV- INCORRECT INDIRECT SETUP
DTETIP	[CHK]	DTETDN-TO10 DONE RECEIVED WITH NO TRANSFER IN PROGRESS
DTETTY	[CHK]	NON-TTY DEVICE
DTEUIF	[CHK]	DTESRV-UNIMPLEMENTED FUNCTION FROM 11
DVCHRX	[CHK]	DVCHR1 - UNEXPECTED CHKDES FAILURE WITHIN .DVCHR
DX2DIE	[CHK]	PHYX2 - DX20 HALTED
DX2DNF	[INF]	PHYX2 - DRIVE NUMBER NOT FOUND IN UDBS
DX2FGS	[CHK]	PHYX2 - FAIL TO GET SENSE BYTES
DX2FUS	[CHK]	PHYX2 - FAIL TO UPDATE SENSE BYTES
DX2HLT	[INF]	PHYX2 - DX20 HALTED
DX2IDM	[CHK]	PHYX2 - ILLEGAL DATA MODE AT DONE INT
DX2IDX	[INF]	PHYX2 - ILLEGAL RETRY BYTE POINTER
DX2IEC	[CHK]	PHYX2 - ILLEGAL ERROR CLASS CODE
DX2IFS	[CHK]	PHYX2 - ILLEGAL FUNCTION AT START IO
DX2IRF	[INF]	PHYX2 - ILLEGAL FUNCTION DURING RETRY
DX2MCF	[CHK]	PHYX2 - DX20 MICROCODE CHECK FAILURE
DX2N2S	[INF]	PHYX2 - MORE TU70S THAN TABLE SPACE, EXCESS IGNORED
DX2NRT	[CHK]	DX2ERR - IS.NRT SET ON SUCCESSFUL RETRY
DX2NUD	[CHK]	PHYX2 - CHANNEL DONE INTERRUPT BUT NO UNIT ACTIVE
DX2NUE	[CHK]	PHYX2 - NO ACTIVE UDB AND DX20 COMPOSITE ERROR SET
DX2RFU	[CHK]	PHYX2 - ERROR RECOVERY CONFUSED
DX2UNA	[INF]	PHYX2 - ATTENTION INTERRUPT AND UDB NOT ACTIVE
DX2UPE	[CHK]	PHYX2 - FAIL TO UPDATE SENSE BYTES DURING INITIALIZATION
DXBASD	[CHK]	PHYP2 - ASYCHRONOUS STATUS FROM NON-POSITIONING DRIVE
DXBDIE	[CHK]	PHYP2 - DX20B MICROCODE HALTED
DXBDMI	[CHK]	PHYP2 - DX20B MICROCODE IS INVALID
DXBEUI	[CHK]	PHYP2 - ERROR TRYING TO INITIALIZE A UNIT
DXBEWC	[CHK]	PHYP2 - ERROR PRESENT WHEN CONNECTING TO A UNIT
DXBFEX	[HLT]	PHYP2 - ILLEGAL FUNCTION STARTING IO
DXBFGS	[CHK]	PHYP2 - FAILED TO GET SENSE BYTES
DXBFUS	[CHK]	PHYP2 - FAILED TO UPDATE SENSE BYTES
DXBHLT	[INF]	PHYP2 - DX20B CONTROLLER HALTED
DXBIEC	[CHK]	PHYP2 - UNKNOWN ERROR CODE FROM DX20
DXBIF2	[HLT]	PHYP2 - ILLEGAL FUNCTION STACKING IO
DXBILF	[HLT]	PHYP2 - ILLEGAL FUNCTION AT DONE INTERRUPT
DXBLTF	[HLT]	PHYP2 - LATENCY OPTIMIZATION FAILURE
DXBMSR	[HLT]	PHYP2 - MULTIPLE SECTORS INDICATED IN ECC RECOVERY
DXBNUD	[CHK]	PHYP2 - NO UNIT ACTIVE FOR DONE INTERRUPT
DXBTNF	[HLT]	PHYP2 - UNIT TYPE NOT FOUND IN TABLE
DXBTTS	[CHK]	PHYP2 - TABLES TOO SMALL FOR THIS MANY DRIVES
DXBUA1	[CHK]	PHYP2 - DONE INTERRUPT AND UNIT WAS NOT ACTIVE
DXBUNA	[CHK]	PHYP2 - ATTENTION INTERRUPT AND UNIT WAS NOT ACTIVE

Table 3 BUGHLT and BUGCHK Names and Descriptions (Cont)

Name	Type	Description
DXBZEC	[CHK]	PHYP2 - ZERO ECC BYTE RETURNED
EPTMPE	[HLT]	PFCDPE: PARITY ERROR OCCURRED IN THE EPT
EXILGO	[CHK]	EXECI - Interrupt during login or logout
EXPAFK	[HLT]	EXPALL: JOB 0 CFORK FAILED
EXPRCD	[CHK]	EXPALL: RCDIR FAILURE
FATCDP	[HLT]	FATAL CACHE DIRECTORY PARITY ERROR
FATMER	[HLT]	FATAL MEMORY ERROR
FEBAD	[CHK]	FEHSD-WRONG FE
FEBFOV	[CHK]	FEHSD-BUFFER OVERFLOW
FEOCPB	[CHK]	FEFSYS - FAILED TO BACKUP ROOT-DIRECTORY
FEUSTS	[CHK]	FESSTS-UNKNOWN STATUS
FILBAK	[CHK]	FILCRD: COULD NOT CREATE BACKUP OF ROOT-DIR
FILBOT	[CHK]	COULD NOT CREATE BOOTSTRAP.BIN FILE
FILBTB	[HLT]	UNABLE TO WRITE BIT TABLE FILE
FILCCD	[CHK]	Could not create directory
FILFEF	[CHK]	Could not create Front End File System
FILHOM	[CHK]	UNABLE TO REWRITE HOME BLOCKS IN WRTBTB
FILIRD	[HLT]	FILINW: COULD NOT INITIALIZE THE ROOT DIRECTORY
FILJB1	[CHK]	FILCRD: No room to create standard system directories
FILMAP	[HLT]	FILIN2: COULD NOT MAP IN ROOT-DIRECTORY
FILRID	[HLT]	FILINW: INDEX TABLE ALREADY SET UP FOR ROOT DIR
FIXBAD	[CHK]	Could not re-write Home Blocks to point to FE Filesystem
FIXBDB	[CHK]	COULD NOT RE-WRITE HOME BLOCKS TO POINT TO BOOTSTRAP.BIN
FKCTNZ	[CHK]	FORK LOCK NEST COUNT NON-ZERO
FKWSP1	[CHK]	LOADBS-UNREASONABLE FKWSP
FLKNS	[CHK]	FUNLK-LOCK NOT SET
FLKTIM	[CHK]	FLOCK-FORK LOCK TIMEOUT
FRKBAL	[CHK]	AGESET-FORK NOT IN BALSET
FRKNDL	[CHK]	FORK NOT PROPERLY DELETED
FRKPTE	[HLT]	BADCPG-FATAL ERROR IN FORK PT PAGE
FRKSLF	[HLT]	SUSFK - GIVEN SELF AS ARG
GIVTMR	[INF]	GIVOK TIMEOUT
GLFNF	[HLT]	GLREM - FORK NOT FOUND
GTFDB1	[CHK]	DSKINS: GETFDB FAILURE.
GTFDB2	[HLT]	NEWLFP: GETFDB FAILURE FOR OPEN FILE.
GTFDB3	[HLT]	DSKREN-GETFDB FAILURE FOR OPEN FILE
GTFDB6	[HLT]	CRDIOA: CANNOT DO GETFDB ON ROOT-DIRECTORY
HARDCE	[CHK]	HARD CACHE ERRORS--CACHE DESELECTED
HPSCHK	[CHK]	SCHEDULER - EXCESSIVE TIME IN HIGH PRIORITY
HSHERR	[CHK]	VERACT - HASH VALUE OUT OF RANGE
HSYFRK	[HLT]	HSYS-JOB 0 CFORK FAILED
IBCPYW	[HLT]	COPY-WRITE POINTER IN INDEX BLOCK
IBOFNF	[HLT]	FILINI: ASOFN FAILURE FOR ROOT DIRECTORY IB
IDXNOS	[HLT]	FILINI - COULD NOT ASSIGN FREE SPACE FOR IDXTAB
ILAGE	[HLT]	BAD AGE FIELD IN CST0
ILBOOT	[HLT]	GETSWM-ILLEGAL VALUE OF BOOTFL
ILCHS1	[HLT]	PHYSIO - ILLEGAL CHANNEL STATUS AT SIO
ILCHS2	[HLT]	PHYSIO - ILLEGAL CHANNEL STATE AT STKIO
ILCNSP	[HLT]	PHYSIO - ILLEGAL CALL TO CONSPW
ILCNST	[HLT]	PHYSIO - ILLEGAL CALL TO CONSTW
ILCST1	[HLT]	ILLEGAL ADDRESS IN CST1 ENTRY, CANNOT RESTART
ILDEST	[HLT]	ILLEGAL DESTINATION IDENTIFIER TO SETMPG OR SETPT
ILDRA1	[CHK]	DASDRM-ILLEGAL OR UNASSIGNED DRUM ADDRESS
ILDRA2	[HLT]	DRMIAD-ILLEGAL DRUM ADDRESS
ILFPTE	[HLT]	ILLFPT: ILLEGAL SECTION NUMBER REFERENCED
ILGDA1	[HLT]	GDSTX - BAD ADDRESS
ILGDA2	[HLT]	GDSTX - BAD ADDRESS
ILGOKM	[CHK]	ILLEGAL FUNCTION FOR GETOKM CALL
ILIBPT	[CHK]	BAD POINTER TYPE IN INDEX BLOCK
ILIRBL	[HLT]	PHYSIO - IORB LINK NOT NULL AT ONFPWQ
ILJRFN	[CHK]	JFKRFH - BAD JRFN, IGNORED
ILLDMS	[CHK]	BADDMS: ILLEGAL DMS JSYS FROM MONITOR CONTEXT
ILLFLT	[CHK]	KA10 FLT PT INSTRUCTION IN MONITOR
ILLGO	[HLT]	INVALID CHANNEL LOGOUT
ILLIND	[HLT]	ILLEGAL INDIRECT
ILLTAB	[CHK]	TABLK2: TABLE NOT IN PROPER FORMAT
ILLUUO	[HLT]	KIBADU: ILLEGAL UUO FROM MONITOR CONTEXT
ILMNRF	[HLT]	ILLEGAL REFERENCE TO MON ADR SPACE
ILOFN1	[HLT]	MSCANP-ILLEG IDENT
ILOKSK	[HLT]	OKSKED EXECUTED WHEN NOT NOSKED
ILPAG1	[HLT]	SWPOTO-INVALID PAGE
ILPAGN	[HLT]	MRKMPG-INVALID PAGE NUMBER
ILPDAR	[HLT]	PHYSIO - ILLEGAL DISK ADDRESS IN PAGEM REQUEST
ILPID1	[CHK]	CREPID: ATTEMPT TO CREATE ILLEGAL PID
ILPID2	[CHK]	DELPID: VALIDATED PID TURNED ILLEGAL
ILPLK1	[HLT]	MLKPG-ILLEGAL ARGS
ILPPT1	[HLT]	UPDOFN-BAD POINTER IN PAGE TABLE
ILPPT3	[HLT]	BAD POINTER IN PAGE TABLE
ILPSEC	[HLT]	ILLEGAL SECTION NUMBER

Table 3 BUGHLT and BUGCHK Names and Descriptions (Cont)

Name	Type	Description
ILPTN1	[HLT]	MRPACS-ILLEG PTN
ILRBLT	[HLT]	PHYSIO - IORB LINK NOT NULL AT CNF/STWQ
ILRFPD	[HLT]	PDL-OV IN ILLEGAL PAGE REFERENCE
ILSPTH	[HLT]	SETPT-SPTH INCONSISTENT WITH XB
ILSPTI	[HLT]	ILLEGAL SPT INDEX GIVEN TO SETMXB
ILSRC	[HLT]	ILLEGAL SOURCE IDENTIFIER GIVEN TO SETPT
ILSWPA	[HLT]	SWPIN - ILLEGAL SWAP ADDRESS
ILTWQ	[HLT]	PHYINT - TWQ OR PWQ INCORRECT
ILTWQP	[HLT]	PHYSIO - PWQ OR TWQ TAIL POINTER INCORRECT
ILULK1	[HLT]	MULKPG - TRIED TO UNLOCK PAGE NOT LOCKED
ILULK2	[HLT]	TRIED TO UNLOCK PAGE NOT LOCKED
ILULK3	[HLT]	MULKMP - ILLEGAL MONITOR ADDRESS
ILULK4	[HLT]	MULKCR - ILLEGAL CORE PAGE NUMBER
ILUST1	[HLT]	PHYSIO - UNIT STATUS INCONSISTENT AT SIO
ILUST2	[CHK]	PHYSIO - UNIT STATUS INCONSISTENT AT SPS
ILUST3	[HLT]	PHYSIO - SCHSEK - IMPOSSIBLE UNIT STATUS
ILUST4	[HLT]	PHYSIO - CONTROLLER ACTIVE AT SPS
ILUST5	[HLT]	PHYSIO - ILLEGAL CHANNEL OR CONTROLLER STATE AT STKIO
ILUST6	[HLT]	PHYSIO - ILLEGAL UNIT STATE AT STKIO
ILXBP	[HLT]	SETPT-BAD POINTER IN XB
IMINX1	[INF]	UNUSUAL ANI INTERRUPT, CONI ANI IS
IMINX2	[INF]	IMIERR CALLED, CONI ANI IS
IMPAB2	[CHK]	ASNTBF: ASNTBF FAILED WHEN NCPLCK SET
IMPABF	[INF]	ASNTBF FAILED
IMPAFB	[HLT]	IMPCQ: ATTEMPT TO UNLOCK BUFFER ON FREELIST
IMPALF	[HLT]	IMPLKB: ATTEMPT TO LOCK BUFFER ON FREELIST
IMPAUF	[HLT]	IMPEIN: ATTEMPT TO UNLOCK BUFFER ON FREELIST
IMPBLK	[CHK]	SNDRFC: Sending RFC for a bad NCP link number
IMPBSC	[INF]	Message has bad size or count
IMPCCF	[HLT]	CANNOT CREATE IMP FORK
IMPCTH	[INF]	IMPNCL TOO HIGH
IMPCUL	[INF]	RECD CTL MSG FOR UNKNOWN LINK
IMPHIF	[INF]	HSTINI FAILED TO FIND HOST NAME FILE
IMPHNW	[CHK]	LHOSTN DISAGREES WITH THE IMP
IMPIFC	[INF]	ILL FMT CTL MSG
IMPIFH	[CHK]	IMPGC-IMPOSSIBLE FAILURE OF IMPHFL
IMPIOP	[CHK]	AN20 CAUSED AN IO PAGE FAIL
IMPLAE	[INF]	IMPOPL: Link already exists
IMPLEO	[INF]	Cannot find LT entry for output message
IMPLTF	[CHK]	IMPLT FULL
IMPMMX	[INF]	MESSAGE STUCK OR OVERDUE TOO LONG
IMPMSL	[INF]	PKMSG - MSG TOO LARGE
IMPMSO	[INF]	MESSAGE STUCK IN OUTPUT QUEUE
IMPMUL	[INF]	RECEIVED MSG FOR UNKNOWN LINK
IMPNBC	[HLT]	PKMSG: NEGATIVE RESIDUAL BYTE COUNT
IMPNEA	[INF]	NVT RECEIVED BYTES EXCEEDING ALLOCATION
IMPNII	[HLT]	No IMP input buffers
IMPNMA	[INF]	PKBY1: NO MSG ALLOCATION
IMPREA	[INF]	RECD EXCESS ALL
IMPREM	[INF]	UPBRB: Received excessive messages
IMPRMI	[HLT]	IMP - REGULAR MESSAGE ON IRREG QUEUE
IMPRNE	[INF]	RECD NCP ERR
IMPRNO	[INF]	RFNM OVERDUE
IMPTMB	[CHK]	NVTXG1: TOO MANY BREAKS OUTSTANDING
IMPUBF	[HLT]	IMULKB: ATTEMPT TO UNLOCK BUFFER ON FREELIST
IMPUFB	[HLT]	IMIP1: ATTEMPT TO UNLOCK BUFFER ON FREELIST
IMPURT	[INF]	IMPDV received unexpected RET
IMPUUO	[HLT]	IMPOSSIBLE MUUO
IMPUX0	[CHK]	IMP JB0 FORK - UNEXPECTED INTERRUPT
IMPXBO	[INF]	IRREG MSG BUFFER OVERFLOW
IMPXUT	[INF]	Received irreg msg with unknown link or type
INCFLK	[CHK]	Fork lock set at return to user
INDCNT	[INF]	DTESRV- BAD INDIRECT COUNT
INVDTE	[HLT]	DTEQ- INVALID DTE SPECIFIED
IOPGF	[HLT]	IO PAGE FAIL
IPCFKH	[CHK]	CHKPDD: COULD NOT FIND LOCAL FORK HANDLE
IPCFRK	[CHK]	PIDINB: CANNOT CREATE FORKS FOR IPCF
IPCJB0	[CHK]	PIDINI: NOT IN CONTEXT OF JOB 0
IPCMCN	[CHK]	MESREC: MESSAGE COUNT WENT NEGATIVE
IPCOVL	[HLT]	PIDINI: PIDS AND FREE POOL OVERLAP, IPCF WILL NOT WORK!
IPCSOD	[CHK]	GETMES: SENDER'S COUNT OVERLY DECREMENTED
ITRLGO	[CHK]	ITRAP - Instruction trap while logging in or out
J0NRUN	[HLT]	JOB 0 NOT RUN FOR TOO LONG, PROBABLE SWAPPING HANGUP
JSBNIC	[HLT]	SETPPG-JSB NOT IN CORE
JTENQE	[HLT]	JTENQ WITH BAD NSKED
KLIOVF	[CHK]	DTESRV-KLINIK DATA BASE TOO LARGE
KPALVH	[HLT]	KEEP ALIVE CEASED
LCKDIR	[HLT]	ATTEMPT TO LOCK DIRECTORY TWICE FOR SAME FORK

Table 3 BUGHLT and BUGCHK Names and Descriptions (Cont)

Name	Type	Description
LNGDIR	[CHK]	LONG DIRECTORY FILE IN DIRECTORY:
LNMILI	[CHK]	LNMLUK: ILLEGAL VALUE OF LOGICAL NAME TABLE INDEX
LOKINT	[CHK]	LOCK BEING LOCKED WHILE OKINT
LOKODR	[HLT]	LOCK REQUESTED OUT OF ORDER
LOKWRG	[HLT]	WRONG FORK IS RELEASING LOCK
LUUMNO	[HLT]	LUUO IN MONITOR CONTEXT
LUUMON	[HLT]	.LBCHK: ILLEGAL LUUO FROM MONITOR CONTEXT
MAP41F	[HLT]	MAPF41 FAILED TO SKIP
MAPBT1	[HLT]	OFN FOR BIT TABLE IS ZERO
MAPCLF	[CHK]	FAILED TO CLEAR MAPS WHEN KILLING JOB
MDDJFN	[HLT]	GETFDB: CALLED FOR NON-MDD DEVICE
MNTLNG	[HLT]	MNTBTB - BIT TABLE IS A LONG FILE
MONPDL	[HLT]	STACK FAULT IN MONITOR
MPEUTP	[HLT]	PFCDPE-UNKNOWN TRAP ON TEST REFERENCE
MPIDXO	[CHK]	MAPIDX - No OFN for Index Table File
MTANOA	[CHK]	IRBDN2: IRBDON CALLED FOR AN ACTIVE IORB
MTANOI	[CHK]	GETUBF: NO QUEUED IORB'S FOR INPUT
MTANOQ	[CHK]	IRBDN1: IRBDON CALLED FOR NON-QUEUED UP IORB
MTAORN	[CHK]	MTDIR0: MAGTAPE IORB OVERRUN
MTARIN	[HLT]	MTAINT: INTERRUPT RECEIVED FOR NONACTIVE IORB
MTMSG	[INF]	FAILED TO SEND MT MESSAGE TO "TAPE" CONTROLLER
NCPFUN	[INF]	NCP FSM RECEIVED FUNNY INPUT
NEGJRT	[CHK]	UCLOCK: NEGATIVE JOBRT DETECTED
NETBAF	[HLT]	RLNTBF: ATTEMPT TO RELEASE BUFFER ALREADY ON FREE LIST
NETBAU	[HLT]	ASNTBF: ATTEMPT TO ASSIGN A BUFFER ALREADY IN USE
NETDET	[CHK]	NVTDET: COULD NOT CLOSE NVT
NETRBG	[CHK]	RLNTBF: ATTEMPT TO RELEASE BUFFER AT GARBAGE LOCATION
NETRBL	[HLT]	ASNTBF: REQUEST FOR BUFFER LARGER THAN MAXWPM
NETWNS	[HLT]	WATNOT: WAS CALLED FROM SCHEDULER LEVEL.
NEWBAK	[HLT]	FILRFS - NEWIB FAILURE FOR BACKUP ROOT-DIR
NEWROT	[HLT]	FILRFS - NEWIB FAILURE FOR ROOT-DIRECTORY
NOACB	[HLT]	MENTR - NO MORE AC BLOCKS
NOADXB	[HLT]	RELOFN-NO DSK ADR FOR XB
NOALCM	[CHK]	ALCMES: CANNOT SEND MESSAGE TO ALLOCATOR
NOARCS	[INF]	ARCMSG: PID for QUASAR is not valid
NOBAT1	[CHK]	FAILED TO WRITE PRIMARY BAT BLOCK
NOBAT2	[CHK]	FAILED TO WRITE SECONDARY BAT BLOCK
NOBTB	[CHK]	FILINI - UNABLE TO OPEN BIT TABLE FILE
NOBTBN	[HLT]	FILINI - UNABLE TO GET SIZE OF BOOTSTRAP.BIN FILE
NOCTY	[HLT]	UNABLE TO ALLOCATE DATA FOR CTY
NODIR1	[CHK]	SPLMES: DIRST FAILED ON EXISTING DIRECTORY NAME
NODTEN	[CHK]	DTESRV - NO DTE BUFFERS AVAILABLE IN CRITICAL CASE
NOFEFS	[HLT]	FILINI - UNABLE TO GET SIZE OF FRONT END FILE SYSTEM
NOFNDU	[HLT]	FNDUNT-CANNOT FIND DEVICE FOR JFN
NOINTR	[CHK]	ITRAP AND PREVIOUS CONTEXT WAS NOINT
NOIORB	[HLT]	SETIRB - MISSING IORB
NOLEN	[HLT]	UPDLEN: NO LENGTH INFO FOR OFN
NOMCCD	[HLT]	TTYSRV: ROUTINE CALLED FOR LINE TYPE NOT SUPPORTED
NOPGT0	[HLT]	OPNLNG: NO PAGE TABLE 0 IN LONG FILE.
NOPID	[CHK]	PIDKFL: PID DISAPPEARED
NORSXF	[HLT]	FAILED TO GET SPACE FOR MASTER DTE
NOSEB2	[HLT]	PGMPE-NO SYSERR BUFFER AVAILABLE
NOSERF	[CHK]	CANNOT GTJFN ERROR REPORT FILE
NOSKTR	[CHK]	ITRAP FROM OR CSKED CONTEXT
NOSLNM	[CHK]	SLNINI: CANNOT CREATE SYSTEM LOGICAL NAME
NOSPLM	[CHK]	RELJFN: COULD NOT SEND SPOOL MESSAGE TO QUASAR
NOTOFN	[HLT]	UPDOFO-ARG NOT OFN
NOUTF1	[CHK]	SPLOPN: NOUT OF DIRECTORY NUMBER FAILED
NOUTF2	[CHK]	SPLMES: NOUT OF GENERATION NUMBER FAILED
NOXADR	[HLT]	EXTENDED ADDRESSING CONFUSION
NPWQPD	[CHK]	PHYSIO - NULL PWQ AT POSITION DONE
NRFTCL	[CHK]	PHYSIO - NO REQUESTS FOUND FOR CYLINDER SEEKED
NSKDIS	[HLT]	DISMISS WHILE NOSKED OR WITH NON-RES TEST ADDRESS
NSKDT2	[CHK]	PGRTRP-BAD NSKED OR INTDF
NSPUDF	[HLT]	UNSUPPORTED NETWORK FUNCTION
NULQTA	[HLT]	QCHK - NO QUOTA INFO SETUP
NWJTBE	[CHK]	NO FREE JTB BLOCKS
OFFONX	[HLT]	ARRST: File marked offline has index blk ptr
OFFSPE	[HLT]	OFFSPQ- PAGE NOT ON SPMQ
OKSKBG	[HLT]	OKSKD0 - OKSKED WHEN NOT NOSKED
OPOPAC	[HLT]	MRETN - TRIED TO OVER-POP AC STACK
OVFLOW	[HLT]	ASOFN - ALLOCATION TABLE OVERFLOW
OVRDTA	[INF]	PHYSIO - OVERDUE TRANSFER ABORTED
P2RAE1	[CHK]	PHYH2 - RH20 REGISTER ACCESS ERROR READING REGISTER
P2RAE2	[CHK]	PHYH2 - REGISTER ACCESS ERR WRITING REG
P2RAE3	[CHK]	PHYH2 - REGISTER ACC ERR ON DONE OR ATN INTERRUPT
PAGLCK	[HLT]	DESPT-PAGE LOCKED
PAGNIC	[HLT]	GETCPP-PAGE NOT IN CORE
PGNDEL	[HLT]	REMFPB-PAGE NOT COMPLETELY DELETED

Table 3 BUGHLT and BUGCHK Names and Descriptions (Cont)

Name	Type	Description
PGUNDX	[HLT]	PGUNTD-IN NESTED TRAP
PH2DNA	[INF]	PHYH2 - DONE INTERRUPT AND CHANNEL NOT ACTIVE
PH2IHM	[CHK]	PHYH2 - ILLEGAL HDW MODE - WORD MODE ASSUMED
PH2PIM	[CHK]	PHYH2 - RH20 LOST PI ASSIGNMENT
PH2WUI	[HLT]	WRONG UNIT INTERRUPTED
PHYCH1	[HLT]	PHYSIO - HOME BLOCK CHECK IORB ALREADY ON TWQ
PHYCH2	[INF]	PHYSIO - HOME BLOCK CHECK IORB TIMED OUT
PHYCH3	[INF]	PHYSIO - HOME BLOCK CHECK IORB TIMED OUT BUT WAS NOT ON TWQ
PHYICA	[HLT]	PHYINI - ILLEGAL ARGUMENT TO CORE ALLOC
PHYICE	[INF]	PHYINI - FAILED TO ASSIGN RESIDENT STG
PHYLTF	[HLT]	PHYSIO - SCHLTM - UNEXPECTED LATOPT FAILURE
PHYNIR	[CHK]	PHYSIO - NULL INTERRUPT ROUTINE AT OPERATION DONE
PHYPOE	[HLT]	PHYALZ - PAGE 0 STORAGE EXHAUSTED
PI1ERR	[CHK]	UNEXPECTED UNVECTORED INTERRUPT ON CHANNEL 1
PI2ERR	[CHK]	UNEXPECTED UNVECTORED INTERRUPT ON CHANNEL 2
PI4ERR	[CHK]	UNEXPECTED UNVECTORED INTERRUPT ON CHANNEL 4
PI6ERR	[CHK]	UNEXPECTED UNVECTORED INTERRUPT ON CHANNEL 6
PIDFLF	[CHK]	CREPID: FREE PID LIST FOULED UP
PIDOD1	[CHK]	MUTCHO: PID COUNT OVERLY DECREMENTED
PIDOD2	[CHK]	DELPID: OVERLY DECREMENTED PID COUNT
PIITRP	[HLT]	INSTRUCTION TRAP WHILE PI IN PROGRESS OR IN SCHEDULER
PISKED	[HLT]	ENTERED SCHEDULER WITH PI IN PROGRESS
PITRAP	[HLT]	PAGER TRAP WHILE PI IN PROGRESS
PM2SIO	[CHK]	PHYM2 - ILLEGAL FUNCTION AT START IO
PM8SIO	[CHK]	PHYM78 - ILLEGAL FUNCTION AT START IO
PRONX2	[HLT]	NXM DETECTED BY PROCESSOR
PSBNIC	[HLT]	SETPPG-PSB NOT IN CORE
PSINSK	[CHK]	PSI FROM NOSKED OR CRSKED CONTEXT
PSISTK	[HLT]	PSI STORAGE STACK OVERFLOW
PTAIC	[HLT]	SWPIN - PT PAGE ALREADY IN CORE
PTDEL	[HLT]	DESPT-PT NOT DELETED
PTMPE	[HLT]	PAGE TABLE PARITY ERROR
PTNIC1	[HLT]	SWPIN - PAGE TABLE NOT IN CORE
PTNON0	[HLT]	SETPT0 - PREVIOUS CONTENTS NON-0
PTOVRN	[HLT]	UPDPGS-COUNT TOO LARGE
PVTRP	[HLT]	PROPRIETARY VIOLATION TRAP
PWRFL	[HLT]	FATAL POWER FAILURE
PWRRES	[CHK]	POWER RESTART
PYILUN	[HLT]	PHYSIO - ILLEGAL UNIT NUMBER
RCVNOE	[CHK]	RCVOK - NO ENTRY FOUND IN QUEUE
RCVTMR	[CHK]	RCVOK TIMEOUT - IGNORING ACCESS CONTROL JOB
RELBAD	[CHK]	RELFRE-BAD BLOCK BEING RELEASED
RELFRM	[HLT]	ILLEGAL TO DEASSIGN 0 FREE SPACE
RELINT	[CHK]	RELFRE CALLED OKINT
RELRNG	[CHK]	RELFRE: BLOCK OUT OF RANGE
RESBAD	[CHK]	RELRES: ILLEGAL ADDRESS PASSED TO RELRES
RESBAZ	[CHK]	RELRES: FREE BLOCK RETURNED MORE THAN ONCE
RESBND	[CHK]	RELRES: RELEASING SPACE BEYOND END OF RESIDENT FREE POOL
RESCHK	[HLT]	RELRES: RESIDENT FREE SPACE WAS OVERWRITTEN
RFILPF	[CHK]	REFILL ERROR PAGE FAIL
RH2ICF	[HLT]	PHYRH2 - INVALID CHANNEL FUNCTION
RP4FEX	[HLT]	PHYP4 - ILLEGAL FUNCTION
RP4IF2	[HLT]	PHYP4 - ILLEGAL FUNCTION AT STKIO
RP4IFC	[HLT]	PHYP4 - ILLEGAL FUNCTION AT CNV
RP4ILF	[HLT]	PHYP4 - ILLEGAL FUNCTION ON INTERRUPT
RP4LTF	[HLT]	PHYP4 - FAILED TO FIND TWQ ENTRY AT RP4LTM
RP4PNF	[HLT]	PHYP4 - DISK PHYSICAL PARAMETERS NOT FOUND
RP4SSC	[CHK]	PHYP4 - STUCK SECTOR COUNTER
RP4UNF	[HLT]	PHYP4 - UNIT TYPE NOT FOUND:
RPGERR	[HLT]	BADCPG-FATAL ERROR IN RESIDENT PAGE
RSMFAI	[HLT]	RESSMM-FAILED TO ASSIGN SWAP MON PAGE
SBSERF	[INF]	SBSERR-COULD NOT GET ERROR BLOCK
SCDUUO	[HLT]	UUO IN SCHEDULER
SCPT01	[HLT]	SCNPT - ENTRY IS NOT AN IMMEDIATE POINTER
SCPT02	[HLT]	SCNPT - PAGE WAS NOT DELETED
SEBISS	[CHK]	SEBCPY-INSUFFICIENT STRING STORAGE IN BLOCK
SEBUDT	[CHK]	SEBCPY-UNKNOWN DATA TYPE
SECEX1	[HLT]	SETMPG-ATTEMPT TO MAP NON-EX SECTION
SERFOF	[CHK]	CANNOT OPENF ERROR REPORT FILE
SERFRK	[HLT]	SERINI-CANNOT CREATE SYSERR FORK
SERGOF	[CHK]	SETOFI-CANNOT GTJFN/OPEN SYSERR FILE
SHRNO0	[HLT]	DESPT-SHARE COUNT NON-ZERO
SHROFD	[HLT]	DWNSHR-OFN SHARE COUNT UNDERFLOW
SHROFN	[HLT]	UPSHR-OFN SHARE COUNT OVERFLOW
SKDCL1	[HLT]	CALL TO SCHEDULER WHEN ALREADY IN SCHEDULER
SKDMPE	[HLT]	MPE IN SCHEDULER OR PI CONTEXT
SKDPF1	[HLT]	PAGE FAIL IN SCHED CONTEXT
SKDTRP	[HLT]	INSTRUCTION TRAP WHILE IN SCHEDULER

Table 3 BUGHLT and BUGCHK Names and Descriptions (Cont)

Name	Type	Description
SNPIC	[CHK]	SNPFN3: INSTRUCTION BEING REPLACED HAS CHANGED
SNPLKF	[CHK]	SNPFN0: CANNOT LOCK DOWN PAGE INTO MONITOR
SNPODB	[CHK]	SNPF4C: COUNT OF INSERTED BREAK POINTS OVERLY DECREMENTED
SNPUNL	[CHK]	SNPF5A: CANNOT UNLOCK SNOOP PAGE
SPTFL1	[HLT]	SPT COMPLETELY FULL
SPTFL2	[HLT]	SPT COMPLETELY FULL
SPTPIC	[HLT]	SWPIN - SPT PAGE ALREADY IN CORE
SPTSHR	[HLT]	UPSHR-SPT SHARE COUNT OVERFLOW
SPWRFL	[CHK]	SPURIOUS POWER FAIL INDICATION
SRQBAD	[CHK]	SCDRQ-BAD CALL TO SCDRQ7
STKOVF	[HLT]	MONITOR STACK OVERFLOW
STRBAD	[HLT]	ASOFN-ILLEGAL STRUCTURE NUMBER
STRTER	[HLT]	FATAL ERROR WHILE PROCESSING PREVIOUS STARTUP ERROR
STZERO	[HLT]	FILINI: STRTAB ENTRY FOR PS IS 0
SUMNR1	[CHK]	AJBALX-SUMBNR INCORRECT
SUMNR2	[CHK]	WSMGR-SUMNR INCORRECT
SWPASF	[CHK]	CHKBAT-FAILED TO ASSIGN BAD SWAPPING ADDRESS
SWPFPE	[CHK]	SWAP ERROR IN SENSITIVE FILE PAGE
SWPIBE	[CHK]	SWAP ERROR IN INDEX BLOCK
SWPJSB	[CHK]	SWAP ERROR IN JSB PAGE
SWPMNE	[HLT]	SWAP ERROR IN SWAPPABLE MONITOR
SWPPSB	[CHK]	SWAP ERROR IN PSB PAGE
SWPPT	[CHK]	SWAP ERROR IN UNKNOWN PT
SWPPTP	[CHK]	SWAP ERROR IN UNKNOWN PT PAGE
SWPSTL	[CHK]	SWAP SPACE TOO LOW AT STARTUP
SWPUPT	[CHK]	SWAP ERROR IN UPT, OR PSB
SWPXXX	[HLT]	UNRECOVERABLE SWAP ERROR FOR CRITICAL PAGE
SYMNOM	[CHK]	Unable to map symbol table page
SYMNOU	[CHK]	Unable to unmap symbol table page
SYSERF	[CHK]	LOGSST-NO SYSERR STORAGE FOR RESTART ENTRY
TM2CCI	[CHK]	PHYM2 - TM02 SSC OR SLA WONT CLEAR
TM2HER	[CHK]	TM2ERR - IS.HER SET ON SUCCESSFUL RETRY
TM2IDM	[CHK]	PHYM2 - ILLEGAL DATA MODE AT DONE INT
TM2IDX	[INF]	PHYM2 - ILLEGAL RETRY BYTE POINTER
TM2IF2	[CHK]	PHYM2 - ILLEGAL FUNCTION ON COMMAND DONE
TM2IRF	[INF]	PHYM2 - ILLEGAL FUNCTION DURING RETRY
TM2N2S	[INF]	PHYM2 - MORE DRIVES THAN TABLE SPACE, EXCESS IGNORED
TM2NUD	[CHK]	PHYM2 - CHANNEL DONE INTERRUPT BUT NO UNIT ACTIVE
TM2RFU	[CHK]	PHYM2 - ERROR RECOVERY CONFUSED
TM2UNA	[INF]	PHYM2 - DONE INTERRUPT AND UDB NOT ACTIVE
TM8AEI	[INF]	PHYM78 - ASYNCHRONOUS ERROR INTERRUPT
TM8N2S	[INF]	PHYM78 - MORE DRIVES THAN TABLE SPACE, EXCESS IGNORED
TM8NUD	[CHK]	PHYM78 - CHANNEL DONE INTERRUPT BUT NO UNIT ACTIVE
TM8SNS	[CHK]	CAN'T SENSE TU78 STATUS
TRPSIE	[CHK]	NO MONITOR FOR TRAPPED FORK
TTBAD1	[HLT]	BAD DEVICE DESIGNATOR FOR TERMINAL AT ATACH2
TTDAS1	[HLT]	HLTJB: UNABLE TO DEASSIGN CONTROLLING TERMINAL
TTFSMS	[INF]	Failed to send system message
TTICN0	[HLT]	TCI - NO BUFFER POINTER BUT COUNT NON-0
TTILEC	[CHK]	TTSND-UNRECOGNIZED ESCAPE CODE
TTLOKB	[HLT]	BAD TTY LOCK IN TTLCK
TTNAC1	[CHK]	LINE NOT ACTIVE AT PTYOPN
TTNAC3	[HLT]	CTY NOT ACTIVE AT FSIPBO
TTNAC4	[HLT]	CTY NOT ACTIVE AT FSIPBI
TTNAC5	[HLT]	CTY NOT ACTIVE AT FSIINI
TTNAC7	[CHK]	DEALLOCATING INACTIVE LINE
TTNAC8	[HLT]	CANNOT ASSIGN TERMINAL AT DEVINI
TTONOB	[HLT]	TTY OUTPUT - NO BUFFER BUT COUNT NON-0
TTQADX	[CHK]	TTYSRV-UNKNOWN FUNCTION REQUESTED
TTULKB	[CHK]	BAD TTY UNLOCK IN ULKTT
TTYBBO	[CHK]	TTYSRV-BIG BUFFER OVERFLOW
TTYNTB	[CHK]	RAN OUT OF TTY BUFFERS
TTYSTP	[INF]	TTYSRV - LINE HAS BEEN SHUT OFF BECAUSE OF EXCESSIVE INPUT RATE
TWQNUL	[HLT]	PHYSIO - PWQ OR TWQ WAS NULL AT A SEEK OR TRANSFER COMPLETION
UBANXM	[HLT]	I/O NMX FROM UNIBUS DEVICE
UIONIR	[HLT]	UDSKIO - NO IORB FOR NOSKED FORK
ULKBAD	[CHK]	UNLOCKING TTY WHEN COUNT IS ZERO
ULKINT	[CHK]	LOCK BEING UNLOCKED WHILE OKINT
ULKSTZ	[CHK]	OVERLY DECREMENTED STRUCTURE LOCK
UNBFNF	[CHK]	UNBLK1 - FORK NOT FOUND
UNFWSS	[HLT]	UNIT NOT FOUND CREATING SDB FOR STRUCTURE
UNPGF1	[HLT]	MEMPAR-PARITY ERROR DURING MEM SCAN
UNPGF2	[HLT]	UNKNOWN PAGE FAILURE TYPE
UNPIRX	[CHK]	UNPIR-NO PSI IN PROGRESS
UNXMPE	[HLT]	PFCDPE-UNEXPECTED PARITY ERROR TRAP
UPTMPE	[HLT]	PFCDPE: PARITY ERROR IN UPT
USGHOL	[INF]	LOST PAGE(S) IN USAGE FILE
UXXCKP	[HLT]	COULDN'T CREATE CHECKPOINT FILE

Table 3 BUGHLT and BUGCHK Names and Descriptions (Cont)

Name	Type	Description
UXXCL1	[CHK]	UNABLE TO CREATE NEW USAGE FILE
UXXCL2	[CHK]	UNABLE TO OPEN NEW USAGE FILE
UXXCL3	[CHK]	UNABLE TO CLOSE USAGE FILE
UXXCRE	[HLT]	CANNOT CREATE USAGE FILE
UXXFAI	[CHK]	USAGE JSYS FAILURE
UXXFIT	[INF]	CHECKPOINT FILE NOT IN CORRECT FORMAT FOR THIS SYSTEM, REBUILDING...
UXXILL	[HLT]	USGMES: ILLEGAL FUNCTION CODE
UXXMAP	[HLT]	USGMAP: CALL TO JFNOFN FAILED
UXXOPN	[HLT]	UNABLE TO OPEN USAGE FILE
UXXWER	[CHK]	WRITE ERROR IN USAGE FILE
WAITNI	[HLT]	WAIT JSYS while not interruptable
WRTBT4	[CHK]	ASOFN ON BIT TABLE FILE FAILED
WRTCPB	[CHK]	WRTBTB - FAILED TO BACKUP ROOT-DIRECTORY
WRTLNG	[HLT]	WRTBTB - BIT TABLE IS A LONG FILE
WSPNEG	[CHK]	SOSWSP-WSP NEGATIVE
WSSPNA	[HLT]	WSSFKP-FORK SPECIAL PAGE BAD AGE
WSSPNC	[HLT]	WSSFKP-FORK SPECIAL PAGE NOT IN CORE
XBWERR	[CHK]	UPDOFN-DSK WRITE ERROR ON XB
XSCORE	[HLT]	CST TOO SMALL FOR PHYSICAL CORE PRESENT

RP07

KL10 Maintenance Guide
Volume II

RP07

This is an entirely new section. Insert the new RP07 tab found at the end of this package. Then insert the following sheets (pages 1 through 41) after this tab.

Discard this sheet after completing directions.

Table of Contents

MASSBUS REGISTERS
This section describes the Massbus registers and gives detailed information on the status and error bits of each.

Massbus Registers

Massbus Register Number (Octal)	Register Name		Mode of Operation
00	(RPCS1)	Control and Status Register	Read/Write
01	(RPDS)	Drive Status Register	Read Only
02	(RPER1)	Error Register 1	Read Only*
03	(RPMR1)	Maintenance Register 1	Read/Write
04	(RPAS)	Attention Summary Pseudo-Register	
05	(RPDA)	Desired Track/Sector Address Register	Read/Write
06	(RPDT)	Drive Type Register	Read Only
07	(RPLA)	Look Ahead Register	Read Only
10	(RPSN)	Serial Number Register	Read Only
11	(RPOF)	Offset Register	Read/Write
12	(RPDC)	Desired Cylinder Address Register	Read/Write
13	(RPCC)	Current Cylinder Address Register	Read Only
14	(RPER2)	Error Register 2	Read Only*
15	(RPER3)	Error Register 3	Read Only*
16	(RPEC2)	ECC Position Register	Read Only
17	(RPEC2)	ECC Pattern Register	Read Only

*Drive resident microdiagnostic routines test set and reset capabilities.

Register 00_8 - Control and Status Register (RPCS1)
This Read/Write register is used to initiate all RP07 command operations. It is physically shared by RP07 Device Control Logic (DCL) and the RH20 Controller. RP07 uses seven of the control register's 16 bits (0-5 and 11); RH20 contains the remaining 9 bits.

Bit 0 (GO) - A command (bits 1-5 in RPCS1) is always transmitted with the GO bit set. When set, GO causes the RP07 DCL to do the following:

1. Decipher the function code (bits 1-5 in RPCS1).

2. Determine if the function code is illegal and, if so, set the appropriate error bit.

3. Determine if the command is a data transfer command.

   a. If the decoded command is a data transfer command, assert the OCC (Massbus Occupied) line within 50 microseconds and execute the function if the RUN (Massbus Run) line becomes asserted.

   b. If the decoded command is not a data transfer command, the RP07 executes the function called for (a microdiagnostic routine, for example).

Bits 1-5 (Function Code) - Function Code bits contain the particular commands to be executed by the RP07. Function codes are listed in the table below.

Function Codes

Command Code Octal	Command	Function Code and GO F4	F3	F2	F1	F0	GO
01	No-Op	0	0	0	0	0	1
03	Illegal	0	0	0	0	1	1
05	Seek	0	0	0	1	0	1
07	Recalibrate	0	0	0	1	1	1
11	Drive Clear	0	0	1	0	0	1
13	Release	0	0	1	0	1	1
15	Offset	0	0	1	1	0	1
17	Return to Centerline	0	0	1	1	1	1
21	Read in Preset	0	1	0	0	0	1
23	No-Op	0	1	0	0	1	1
25	Illegal	0	1	0	1	0	1
27	Illegal	0	1	0	1	1	1
31	Search	0	1	1	0	0	1
33	Illegal	0	1	1	0	1	1
35	Microdiagnostic	0	1	1	1	0	1
37	Illegal	0	1	1	1	1	1
41	Illegal	1	0	0	0	0	1
43	Illegal	1	0	0	0	1	1
45	Illegal	1	0	0	1	0	1
47	Illegal	1	0	0	1	1	1
51	Write Check Data	1	0	1	0	0	1
53	Write Check Header and Data	1	0	1	0	1	1
55	Illegal	1	0	1	1	0	1
57	Illegal	1	0	1	1	1	1
61	Write Data	1	1	0	0	0	1
63	Format Track	1	1	0	0	1	1
65	Write Track Descriptor	1	1	0	1	0	1
67	Illegal	1	1	0	1	1	1
71	Read Data	1	1	1	0	0	1
73	Read Header & Data	1	1	1	0	1	1
75	Read Track Descriptor	1	1	1	1	0	1
77	Illegal	1	1	1	1	1	1

Bits 6-10 - These bits reside in the RH20.

Bit 11 (DVA) - Device Availability (DVA) is always set when RPCS1 is read by Massbus. The input to the buffer for DVA (J42 Pin 8) is hardwired to ground. In dual access mode the controller that has access sees DVA set; the controller that does not have access sees all zeros plus the parity bit set. In single access mode, DVA is always set when read by the controller.

Bits 12-15 - These bits reside in the RH20 controller.

Register 01_8 - Device Status Register (RPDS)
This read-only register contains non-error indicators.

Bit 0 (OM) - Offset Mode (OM) sets when an Offset Command (Function Code and GO, bits 5-0 in RPSC1) is written.

OM Reset condition generating ATTENTION INTERRUPT:

- Return to Centerline command

OM Reset conditions - NO ATTENTION INTERRUPT generated:

- Any write command
- Seek
- Implied Seek
- Mid-transfer Seek
- Search
- Recalibrate
- Read in Preset
- Power on Reset
- Release

Bit 1 (EWN) - Early Warning (EWN) is set when RP07 thermal or air flow sensors detect an out-of-limit condition in the drive, and remains set as long as the condition persists.

EWN is a status bit only and will not cause an attention interrupt but will, when set, cause the UNSAFE indicator (located on the operator control panel) to blink.

If an early warning condition is detected before a spindle-start sequence, spindle-start is inhibited. Detection after a spindle-start sequence does not generate a spindle-stop sequence; however, the UNSAFE indicator blinks to inform the operator of an unsafe condition. Continued rise in temperature causes CB3 to trip, removing ac power to the drive.

Bit 2 (ILV) - Interleaved Sectors (ILV) is set by the logic that enables sector interleaving. ILV is field-programmable by a hardware jumper on the backpanel (J26). See Chapter 2 of the RP07 Service Manual for the jumper configuration.

Bits 3-5 - Always 0.

Bit 6 (VV) - Valid Volume (VV) is presented to the host as a 1.

Bit 7 (DRY) - When Drive Ready (DRY) is set, the drive is ready to accept commands. DRY is the complement (opposite state) of GO (bit 0 in RPCS1): if GO is set when RPDS is read, DRY is negated.

Bit 8 (DPR) - In a dual access drive, Drive Present (DPR) is set to the RH20 controller that has access to the RP07 and reset to the other RH20 controller. In single access mode, the DPR bit is always set when read by the RH20 controller.

Bit 9 (PGM) - In a dual access drive, the Programmable (PGM) bit is set when the drive is equally available to both controllers and the Access A, A/B, B switch on the operator control panel is in the A/B position at the time of drive transition from off-line to on-line. Transition occurs after successful completion of a start-spindle sequence or after the Online switch is placed from off-line to on-line position.

In single access mode, when DRQ (Drive Request Required, bit 11 in RPDT) is reset, PGM is negated (reset), the Access switch is ignored, and access "A" is forced true.

Bit 10 (LBT) - Last Block Transferred (LBT) is set by the RP07 DCL during a Data Transfer Command when data is being transferred to the last addressable sector of user media.

LBT resets under any of the following conditions.

- A new command is issued
- Massbus Initialize
- Drive Clear
- Power On Reset

Bit 11 (WRL) - Write Lock (WRL) reflects the true write protect condition of the drive logic as a result of assertion of the Write Protect switch on the operator control panel. WRL will not set if a write command is in progress; set is deferred until completion of the write operation.

Bit 12 (MOL) - Medium Online (MOL) is set when the drive is ready to accept commands after a successful spindle-start sequence and when the Online switch is in on-line position. MOL must be set prior to initiation of any command except when the RP07 is in microdiagnostic mode.

MOL is reset whenever the drive enters one of the following states in which commands cannot be executed.

- Power-down sequence
- Unsafe condition
- Change from on-line to off-line position at the operator control panel; off-line start is deferred until command completion

Bit 13 (PIP) - Position in Progress (PIP) is set whenever the drive positioner is in motion.

PIP is reset at completion of the movement.

The following table shows a list of relationships between PIP and the type of operation being performed.

DRY-PIP-ATA Status During Operations

Operation	DRIVE READY (DRY)	POSITION IN PROGRESS (PIP)	ATTENTION (ATA) AT END OF OPER
No operation	0	0	No
Recalibrate	0	1	Yes
Offset	0	0*	Yes
Drive Clear	0	0	No
Return to Centerline	0	0*	Yes
Seek (including 0 cylinder)	0	1	Yes
Write Check	0	0**	No
Write Data	0	0**	No
Write Header and Data	0	0**	No
Read Data	0	0**	No
Read Header and Data	0	0**	No
Implied Seek	0	1	No
Mid-Transfer Seek	0	1	No
Read in Preset	0	0	No
Search	0	0**	Yes
Microdiagnostic	0	0	Yes

*PIP sets if command execution time exceeds the current Massbus cycle time.
**PIP is set during the implied seek portion of the command.

Bit 14 (ERR) - Composite Error Status (ERR) is the OR (Inclusive OR) of all register error bits. ERR is reset by one of the following conditions - provided the error is not persistent.

- Drive Clear
- Massbus Initialize
- Power On Reset

A composite error set at initiation of a command other than a Drive Clear or a Microdiagnostic command will inhibit execution of the command and prevent the GO bit from being set.

Bit 15 (ATA) - Attention Active (ATA) indicates the state of the

Attention flip-flop for the switched/seized RH20 controller.

ATA is set under any of the following conditions.

- Any error in the error registers
  At occurrence if GO bit is reset
  At completion of a command if GO bit is set

- On a Write to any register when Composite error is set except the Attention Summary register (RPAS) or maintenance registers, writing Microdiagnostic command, or Drive Clear function codes with the GO bit in the Control register

- Completion of a Seek, Search, Recalibrate, Offset, Return to Centerline, or Microdiagnostic command

- Whenever MOL changes state

- In a dual access RP07 when access request flip-flop (DRQ, bit 11 in RPDT) is set for one controller and the other controller releases.

ATA is reset under any of the following conditions.

- Writing the GO bit when ERR is reset

- Drive Clear (if error is not persistant and GO is not set)

- Massbus Initialize

- Writing a 1 into the Attention Summary Pseudo-Register (RPAS) bit position that corresponds to the RP07 logical drive address.

Register 02_8 - Error Register 1 (RPER1)
This read only register contains individual error condition indicators.

The RP07 error conditions fit into one of two basic categories:

- CLASS A errors, which can be handled at the completion of a non-data transfer command, at a convenient block boundary

- CLASS B errors, which must be handled immediately; a class B error causes the drive to terminate command execution as soon as possible.

All nonpersistant error bits in Error Register 1 are reset (cleared) under any of the following conditions.

- Drive Clear
- Massbus Initialize
- Power On Reset

Bit 0 (ILF) - Illegal Function (ILF) is set when a function code and GO bit are written into the RPCS1 and the code does not correspond to an implemented command in the RP07.

ILF is a CLASS B error.

Bit 1 (ILR) - Illegal Register (ILR) is set when a read or write command is attempted to or from a nonexistent register. Trying to write into a read-only register does not set ILR.

ILR is a CLASS A error.

Bit 2 (RMR) - Register Modification Refused (RMR) is set when a write command is attempted to an existing drive register (except the RPAS) while the GO bit is set and an operation is in progress.

RMR is a CLASS A error.

Bit 3 (PAR) - The parity (PAR) error bit is set:

- By DPE (Data Parity Error, bit 3 in RPER3) when a parity error is detected on a Massbus data line when writing data on the media (CLASS A error), or

- When a parity error is detected on a Massbus control line when writing into a register (CLASS B error).

PAR applies to data or control information being transmitted only from the RH20 controller to the RP07; the RP07 checks for the presence of odd parity.

Bit 4 (FER) - Format Error (FER) is set after reading an entire header if bit 12 of the first header word does not match FMT (bit 12 in RPOF).

FER is a CLASS A error during a Read and/or Write Check Header and Data Command.

FER IS A CLASS B error for all others.

Bit 5 (WCF) - Write Clock Fail (WCF) is set during a write operation if the RP07 fails to receive a response to a request for data (write clock) from the RH20 within one word time.

WCF is a CLASS B error.

Bit 6 (ECH) - ECC Hard (ECH) error is set when a Data Check (DCK, bit 15 in this register) cannot be recovered by using ECC.

ECH is a CLASS B error.

Bit 7 (HCE) - Header Compare Error (HCE) is set while reading the header if one or more of the following occurs.

- The cylinder address bits, 0-9 in the first header word, do not match the contents of the Desired Cylinder Address register (RPDC) bits 0-9.
- The sector address bits, 0-6 in the second header word, do not match the contents of the Desired Sector/Track Address register, bits 0-6.
- The track address bits, 8-13 in the second header word, do not match the contents of the Desired Sector/Track Address register, bits 8-13.
- Bits 13, 11, and 10 in the first header word or bits 15, 14, and 7 in the second header word are not 0.

Bits 15, 14, and 12 of the first header word are ignored by the header compare logic.

HCE is a CLASS A error that causes termination of the command in progress after reading the entire header, unless the command in progress is a Read or Write Check Header and Data Command, or HCI (Header Compare Inhibit, bit 10 in RPOF) is asserted, in which case HCE is a CLASS A error.

Bit 8 (HCRC) - Header CRC (HCRC) is set when the CRC register is nonzero after reading the entire header and redundancy bytes.

In the above case HCRC is a CLASS B error, which causes termination of command in progress after reading the entire header, unless:

1. The command in progress is a Read or Write Check Header and Data, or
2. HCI (Header Compare Inhibit, bit 10 in RPOF) is asserted.

In the above two cases, HCRC is a CLASS A error.

Bit 9 (AOE) - Address Overflow Error (AOE) is set when the RH20 attempts to continue data transfer beyond the last user-available sector causing a cylinder address overflow. When AOE is set, the sector and track count in the RPDA and the cylinder value in the RPDC are incremented at EBL assertion.

AOE is a CLASS B error.

Bit 10 (IAE) - When the contents of the RPDC or the RPDA are invalid, Invalid Address Error (IAE) is set as a result of any of the following commands.

- Seek
- Search
- Read Header and Data
- Read Data
- Write Check Header and Data
- Write Check Data
- Format Track
- Write Data
- Write Track Descriptor
- Read Track Descriptor

See the following table for valid addresses.

Valid Addresses

Address	Functional Mode 16-Bit	Functional Mode 18-Bit	Diagnostic Mode 16-Bit	Diagnostic Mode 18-Bit
Desired cylinder	0-629	0-629	0-631	0-631
Desired head	0-31	0-31	0-31	0-31
Desired sector	0-49	0-42	0-49	0-42

IAE is a CLASS B error.

Bit 11 (WLE) - When a write operation is attempted on a drive that is in write lock mode, Write Lock Error (WLE) is set. During a write command, if the Write Protect switch on the operator control panel becomes asserted, no error condition results; the current write operation completes.

WLE is a CLASS B error.

Bit 12 (DTE) - Drive timing error bit is set if during a data transfer a timing failure is detected by the drive logic. The DTE error is a CLASS B error that causes immediate termination of the command in progress.

Bit 13 (OPI) - Operation Incomplete (OPI) is set under any of the following conditions.

1. If during an implied seek the RP07 does not find the correct sector within three revolutions from the start of a search while executing a search or data transfer command.

2. Failure to detect INDEX pulse for three revolutions on commands that are oriented on the index marker, following:

   - Format track
   - Read Track Descriptor
   - Write Track Descriptor
   - Search
   - Read Check Header and Data for sector 0
   - Write Check Header and Data for sector 0

OPI is a CLASS B error.

Bit 14 (UNS) - Unsafe (UNS) is the inclusive OR of the following errors that make the RP07 unsafe for normal operation.

1. R/W Unsafe #1 (RWU1, bit 10 in RPER2)
2. R/W Unsafe #2 (RWU2, bit 11 in RPER2)
3. R/W Unsafe #3 (RWU3, bit 12 in RPER2)
4. DC Unsafe (DCU, bit 5 in RPER3)
5. Tach Calibration Failure
6. CPU Unsafe
7 All other permanent error conditions

All above error conditions are CLASS B errors.

Bit 15 (DCK) - Data Check (DCK) is set at completion of reading data and the ECC (Error Correction Code) field of a sector if the ECC register bits 11-31 are nonzero.

DCK is a CLASS A error if the ECC Inhibit (ECI, bit 11 in RPOF) is set.

DCK is a CLASS B error if ECI is reset. The command is then terminated at completion of the error correction process.

Register 03_8 - Maintenance Register (RPMR1)
Host processor software gains access to the drive-resident microdiagnostics through the Massbus Maintenance Register. This read/write register allows the host to initiate RP07 microdiagnostic routines and monitor microdiagnostic results.

Bits 0-7 - The host enters parameters to be used during execution of a specific microdiagnostic routine.

Bits 8-14 - Routine number bits, written by the host, direct the RP07 to run a specific microdiagnostic routine.

Bit 15 (DMD) - The Diagnostic Mode (DMD) bit is written by the host to enable operation in microdiagnostic mode. When set, bit 15 will:

- Disable write operations on all cylinders except FE cylinders

- Enable execution of a specific microdiagnostic routine

- Enable execution of commands with MOL reset

- Enable access to FE cylinders

Register 04_8 - Attention Summary Pseudo-Register (RPAS)
This read/write register is called a pseudo-register because it is implemented as one bit in each drive. Each RP07 has one flip-flop that, depending on the logical location of the drive, corresponds to the appropriate line on the asynchronous Massbus control lines. The RPAS allows the host to see where the drive requesting attention is located. To see the cause of ATA, the RH20 controller will then read the RPDS.

To read RPAS does not require that Massbus DS (Device Select) address lines be used; all drives respond each time the Massbus addresses RPAS. RPAS is the only register that may be read in this manner.

To write RPAS requires that Massbus DS address lines be used.

Bits 0-7 - Bit 0 is the Attention Active (ATA) bit of drive 0; bit 1 is the ATA bit of drive 1, and so on through drive 7.

Bits 8-15 - Bits 8-15 are 0s.

READING the RPAS - Because the host does not have to specifically address a drive in order to read the RPAS, the RH20 controller will generally request Attention Summary status from all drives simultaneously by indicating a "Read from Register 04_8" on the Massbus register select lines and raising the Demand pulse.

When "register 04_8" is selected, each drive recognizes the 04 address and places the output of its ATA flip-flop in its assigned position on one of the control lines. For example, drive 0 places ATA0 on Massbus line 0. The parity line is ignored since, on a read, parity cannot be generated in the drive.

After placing the Register 04 address on the Register Select line, asserting Demand, and receiving an ATA from each drive, the RH20 strobes the ATA bits in order to read the results.

RP07 will inhibit displaying the RPAS when it senses the negation of Demand.

WRITING in the RPAS - The attention summary flip-flop status on each drive can be altered by the RH20. Each drive receives a bit from the Massbus control lines; if the bit is set, the drive resets its ATA bit. To clear the attention bit the unit must be selected in RPCS2 and a 1 must be written into the appropriate bit. See the following table for the effect of writing an ATA bit.

Writing ATA Bits

Bit Written	ATA Before	ATA After
0	0	0
0	1	1
1	0	0
1	1	0

Writing a 1 causes a set bit to be reset.
Writing a 0 has no effect.

This write operation allows for reset of ATA bits that have already been seen and acted upon without accidentally resetting other ATA bits that may have become set in the meantime.

On a write, the controller presents the Register 04 address (176716_8) on the Massbus Register Select and DS lines and raises the Demand pulse.

Following the rise of Demand, the Massbus control bus lines with Attention Summary information are strobed by the RP07 selected by DS 2-0. The information is valid until negation of Demand.

For a write operation, parity will be generated by the RH20 but will only be checked on the controller that is switched/seized to RP07.

The RP07 must respond with the Transfer pulse.

When the RP07 sets the Attention Line without ERR (Composite Error, bit 14 in RPDS) set, drive logic will accept any command and reset ATA.

In the event of a hard (persistent) error, ATA must be reset by writing a 1 in the appropriate bit position so that all drives on the Massbus are not rendered inoperable by the attention line's constant assertion. The drive error will remain set. If the RH20 attempts a write in any register except the RPAS or attempts a command other than a Drive Clear or a microdiagnostic, the ATA flip-flop will set again.

Register 05_8 - Desired Sector/Track Address Register (RPDA)
This read/write register provides spiral transfer capability; it increments automatically at EBL (End of Block) during a data transfer command, relieving software from updating the register on multiple block transfers.

The RPDA is cleared by:

- Read in Preset
- Power On Reset
- Writing 0 via the Massbus

RPDA data will not change during a Massbus read cycle.

Bits 0-6 (SA) - These bits compose the Desired Sector Address (SA) field.

Bit 7 - Always 0.

Bits 8-13 (TA) - These bits compose the Desired Track Address (TA) field.

Bits 14-15 - Always 0.

RPDA resets after the final sector and/or final track is transferred. See the following table for examples (assume 16-bit mode).

Example Final Transfers

	Current Track Address	Current Sector Address
During Transfer	00_8	00_8
After EBL	00_8	01_8
During Transfer	00_8	61_8
After EBL	01_8	00_8

RPDA increments at EBL pulse.

Register 06_8 - Drive Type Register (RPDT)
This read-only register is used to provide the software with information distinguishing the RP07 from other Massbus devices.

Reading the RPDT Register will cause the drive to send the drive type number and the appropriate parity bit (odd parity) to the unseized RH20 controller.

Bit 1 - Always 1.

Bits 0, 2, 3, 4 - Always 0.

Bit 5 - Always 1.

Bits 6-10 - Always 0.

Bit 11 (DRQ) - Drive Request Required (DRQ) is field-programmable via backpanel jumper (J26 pins 1 and 2). DRQ is set for dual access configuration (which must be requested before use and released after use). DRQ is reset for single access configurations.

Bit 12 - Always 0.

Bit 13 - Always 1.

Bits 14-15 - Always 0.

Register 07_8 - Look Ahead Register (RPLA)
This read-only register contains the exact rotational position of the heads in relation to the data track. Rotational position is monitored by a sector counter in the RP07. The counter is set to 0 at each index pulse, then is incremented each time a sector mark is encountered. The RPLA remains stable during a Massbus read cycle; however, the value may be incorrect if sampled at the time the counter changes value.

RPLA presents a sequential binary count regardless of the interleave state.

Maximum count is specified by FMT (bit 12 in RPOF): 49 in a 16-bit format; 42 in an 18-bit format. Changing FMT has an immediate effect on the SC field of RPLA.

Bits 0-5 - Always 0.

Bits 6-11 (SC) - These bits compose the Sector Counter (SC) field and are the only ones used in the RPLA.

Bits 12-15 - Always 0.

Register 10_8 - Serial Number Register (RPSN)
This read-only register displays the last four digits of the RP07 serial number in BCD. The drive serial number is factory hardwired on the backpanel at J26.

Bits 0-3 - Least significant BCD digit of serial number.

Bits 4-7 - Tens BCD digit of serial number.

Bits 8-11 - Hundreds BCD digit of serial number.

Bits 12-15 - Most significant BCD bit of serial number.

Register 11_8 - Offset Register (RPOF)
This read/write register is used for control information.

Bits 0-9 - These bits are presented by the RP07 as 0.

Bit 10 (HCI) - Header Compare Inhibit (HCI) is set by software to inhibit all header errors. When header errors occur with HCI set during Read or Write Check Data commands, they are classified as A errors.

HCI is reset by any of the following.

- Read in Preset
- Writing a 0 in bit 10 (HCI)
- Power On Reset

Bit 11 (ECI) - Error Correction Inhibit (ECI) is set by software to inhibit attempts by the RP07 to recover from a DCK (Data Check Error, bit 15 in RPER1) and to allow a Data Transfer command to continue beyond the sector where DCK occurred.

ECI is reset by any of the following.

- Read in Preset
- Writing a 0 in bit 11 (ECI)
- Power On Reset

Bit 12 (FMT) - The Format (FMT) bit, when set by software, enables the RP07 to operate in 16-bit mode. When reset, FMT enables 18-bit operation. Format is determined by FMT and maintained in RPLA.

FMT is reset by any of the following.

- Read in Preset
- Writing a 0 in bit 12 (FMT)
- Power On Reset

Bit 13 - Always 0.

Bit 14 (MTD) - Move Track Descriptor (MTD), when set, causes the Track Descriptor Record to be written an additional 64 bytes after the index pulse when a Write Track Descriptor command is initiated.

MTD is reset by any of the following.

- Read in Preset
- Writing a 0 in bit 14 (MTD)
- Power On Reset
- Completion of any command (GO resets)

Bit 15 (CMD) - When set as a failsafe by software, the Command Modifier (CMD) bit allows the following header handling commands: Read Track Descriptor, Write Track Descriptor, and Format Track.

CMD is reset by any of the following.

- Read in Preset
- Writing a 0 in bit 15
- Power On Reset
- Completion of any command (GO resets)

Register 12_8 - Desired Cylinder Address Register (RPDC)
This read/write register is loaded by software with the address of the cylinder that the positioner will move to on a Seek, Search, or Data Transfer command.

The content of RPDC is subject to change while GO is set.

The RP07 provides spiral transfer capability. Spiral transfer means that the software can continue reading through data tracks on a normal read.

Spiral transfer capability is realized when the RP07 is transferring data from the final sector and final track providing that at EBL:

1. RUN line is active, and
2. No error condition exists.

The desired cylinder address will then increment, and a seek to that address is automatically initiated, provided the address is valid.

RPDC is reset by any of the following.

- Read in Preset
- Writing a 0 in this bit position
- Power On Reset

Bits 0-9 (DC) - Desired Cylinder (DC) bits compose the desired address field; LSB is 0.

Bits 10-15 - Always 0.

Register 13_8 - Current Cylinder Address Register (RPCC)
This read-only register reflects the address of the cylinder below the read/write heads.

The RPCC is updated at completion of a positioning operation.

The content of RPCC is subject to change while GO is set.

RPCC is reset by:

1. A recalibrate operation, or
2. An initial head load when the positioner is loaded.

Bits 0-9 (CC) - Current Cylinder (CC) bits comprise the current address field; LSB is 0.

Bits 10-15 - Always 0.

Register 14_8 - Error Register 2 (RPER2)
This read-only register contains error indicators associated with RP07 and its internal control logic.

Errors are classified into CLASS A and CLASS B:

- CLASS A errors can be handled at the completion of a non-data transfer command or at a convenient block boundary in a data command.

- CLASS B errors are handled immediately. The drive terminates command execution as soon as possible.

Provided that an error is not hard (persistent), RPER2 is reset by any of the following.

- Power On Reset
- Drive Clear
- Massbus Initialize

Bits 0-7 - These eight bits contain error status for errors that are processed by the drive's 8080 microprocessor. When drive-resident microdiagnostics are initiated by the host processor and an error results, an error code - the result of that particular microdiagnostic - will be:

1. Reflected in bits 0-7 of Error Register 2
2. Illuminated in the eight LEDs on A1A07 PCA
3. Displayed on the FE panel.

Bit 8 (WRU) - The Write Ready Unsafe (WRU) bit sets during a write operation if write current is active and drive logic determines that the positioner has moved beyond track centerline limits.

WRU set causes the RP07 to turn off write current immediately and abort the write command.

WRU is a CLASS B error.

Bit 9 (WOR) - The Write Over-run error bit will set if write current is active during both the leading and trailing edges on an index pulse. The detection of this condition will turn off write current immediately and subsequently abort the write command.

Bit 10 (RWU1) - The Read/Write Unsafe 1 (RWU1) error bit is set if no write transitions are detected by the drive read/write safety circuits during a write operation (write gate ON) within five microseconds.

Bit 11 (RWU2) - The Read/Write Unsafe 2 (RWU2) error bit is set if more than one head has been selected during a read or write operation.

RWU2 immediately:

1. Deselects all heads
2. Disables write current
3. Aborts the write command.

Detection of RWU2 causes illumination of the UNSAFE indicator on the operator control panel.

RWU2 is a CLASS B error.

Bit 12 (RWU3) - The Read/Write Unsafe 3 (RWU3) bit is set when drive read/write safety circuits detect write current when no write operation is in progress (write gate OFF).

Detection of RWU3 causes illumination of the UNSAFE indicator on the operator control panel.

RWU3 is a CLASS B error.

Bit 13 (CPU) - The CPU Unsafe (CPU) bit is asserted when the 8080 microprogram fails to retrigger the CPU Unsafe timer prior to time-out.

CPU is a CLASS B error.

Bit 14 (CPE) - CROM parity error.

Bit 15 (PGE) - The Program Error (PGE) bit is set if particular commands are attempted without the CMD (Command Modifier, bit 15 in RPOF) set. The following commands then set PGE.

- Write Track Descriptor
- Read Track Descriptor
- Format Track

PGE is a CLASS B error.

Register 15_8 - Error Register 3 (RPER3)
This read-only register contains error indicators that are classified CLASS A or CLASS B as in Error Registers 1 and 2:

- CLASS A errors can be handled at the completion of a non-data transfer command or at a convenient block boundary in a data command.

- CLASS B errors are handled immediately. The drive terminates command execution as soon as possible.

Provided that an error is not hard (persistent), RPER3 is reset by any of the following.

- Power On Reset
- Drive Clear
- Massbus Initialize

Bit 0 (RTO) - RUN Timeout (RTO) is set if after 30 milliseconds from assertion of GO the drive fails to detect the RUN line assertion.

Bit 1 (SCF) - Sync Clock Failure (SCF) sets if the RP07 sync clock counter has not gone to zero within the allocated time.

Bit 2 (SBE) - Sync Byte Error (SBE) is set if the sync byte associated with a data field or defect skip is not found.

Bit 3 (DPE) - Data Parity Error (DPE) sets during a write operation under either of the following conditions.

1. If a data parity error is detected (odd parity used)
2. If a buffer parity error is detected (odd parity used)

DPE sets causes PAR (bit 3 in RPER1) to set.

DPE is a CLASS A error.

Bit 4 (SDF) - SERDES Data Failure (SDF) is set as a result of timing failures relating to the drive data buffer. SDF is asserted under either of the following conditions.

1. An attempt to shift data into the buffer when it is not ready
2. An attempt to strobe buffer output when output data is not ready

Bit 5 (DCU) - DC Unsafe (DCU) sets when RP07 detects a low dc voltage.

DCU extinguishes the DC Safe Indicator and causes the UNSAFE indicator on the operator control panel to illuminate.

DCU is a CLASS B error.

NOTE
Overvoltage protection is provided by a crowbar circuit. DCU cannot be guaranteed to set for an overvoltage condition.

Bit 6 (IXU) - The Index Unsafe (IXU) error bit is asserted if an index error is detected during a Format Track command or a Write Track Descriptor command.

IXU set reflects either of two conditions:

1. Failure to detect an index pulse during Index Window
2. Detection of an index pulse outside the Index Window.

IXU is a CLASS B error.

NOTE
An index error condition cannot be reset until a valid index pulse is detected. IXU may therefore appear hard (persistent) for two complete media revolutions.

Bit 7 (DVC) - The Device Check (DVC) error bit is the inclusive OR of all error bits in RPER2 (8-15) and bits 0-15 in RPER3.

DVC does not cause an attention condition. ATA will be set in accordance with the specific error bit that causes DVC to assert.

Bit 8 (PHF) - The 8080 Processor Handshake Failure (PHF) error bit is set if the 8080 microprocessor fails to respond to a command.

Bit 9 (LCE) - Loss of Cylinder Error (LCE) is set if positioner movement is detected outside the cylinder boundary when no positioning operation is in progress. Detection of this condition causes the drive to automatically issue a recalibrate operation.

The Error bits (bits 0-7 in RPER2) and ATA will NOT become asserted until completion of the recalibrate operation. PIP (bit 13 in RPDS) is asserted during the positioning operation.

The proper error code for LCE is reflected in the eight LEDs on the A1A7 PCA (Servo Control) as well as in bits 0-7 of RPER2.

If a command is loaded with the GO bit (bit 0 in RPCS1) set during the processing of LCE, the command is deferred until completion of recalibration. Then the command is terminated and ATA is raised. If a data command is loaded, EXC (Exception) and EBL (End of Block) will be raised.

Bit 10 (LBC) - The Loss of Bit Clock (LBC) error bit is set under the following conditions.

1. A microcoded time-out occurred in which the word counter failed to attain the value loaded into the compare register within the time-out interval. This is detected by a failure of the end of branch condition to assert by the end of the time-out interval.

2. The end of branch condition failed to negate after the compare register (BC) was loaded with a new value, indicating that either:

   a. The word counter value is greater than or equal to the new BC register value at the time the end branch condition is tested.

   b. The end branch signal failed to reset with the writing of the BC register.

Bit 11 (CLF) - Control Logic Failure (CLF) is set by RP07 logic as a result of any of the following conditions.

- An attempt to write in the 8080 Communications Register when the register is full
- An interrupt failure in the 2901
- An invalid function code interrupt

Bit 12 (WSF) - Write Current Sense Failure (WSF) is set by the drive logic when the device fails to sense Write Current after write gate has been enabled.

Bit 13 (DSE) - Defect Skip Error (DSE) is set by the drive logic when an invalid value results during defect skip calculation.

Bit 14 (SKI) - Seek Incomplete (SKI) is set when the drive logic detects any of the following conditions.

Error Codes

Error Code	Condition
0A	Seek too long
0B	Guard band detected during seek
0C	Seek overshoot
44	Guard band detect failure during recalibrate
45	Reference gap or GB pattern; detection failure (recalibrate)
46	Seek error during recalibrate
4A	Attempt to land in guard band during recalibrate

Detection of a SKI error causes the RP07 to automatically initiate a recalibrate. The error bits (0-7 in Error Register 2) and ATA will NOT set until completion of the recalibration operation.

That particular error code for SKI is reflected in the eight LEDs on A1A07 PCA as well as in bits 0-7 of RPER2.

Bit 15 (BSE) - Bad Sector Error (BSE) is set at the completion of a CRC character check if either bit 14 or bit 15 of the first header word is found to be zero.

If the command is Read or Write Check Header and Data, command termination occurs at normal EBL (End of Block) time for the current sector.

If the command is not Read or Write Check Header and Data, command termination occurs at completion of the CRC check.

Register 16_8 - ECC Position Register (RPEC1)
This read-only register contains the binary address minus 1 of the first bit of an error burst in the data and ECC field. The contents reflect the completion of a Data Transfer that results in DCK (Data Check Error, bit 15 in RPER1), without ECH (ECC Hard Error, bit 6 in RPER1).

If ECH (ECC Hard Error, bit 6 in RPER1) or ECI (Error Correction Inhibit, bit 11 in RPOF) is set, the contents of RPEC1 are irrelevant.

Bits 0-12 - These bits are binary weighted.

Bits 13-15 - Always 0.

Register 17_8 - ECC Pattern Register (RPEC2)
This read-only register contains an 11-bit error burst that is XORed (exclusive ORed) with the data in main memory (located by the position count) to correct the error burst. The contents reflect the completion of a Data Transfer that results in DCK (Data Check Error, bit 15 in RPER1) without ECH (ECC Hard Error, bit 6 in RPER1).

If ECH (bit 6 in RPER1) or ECI (Error Correction Inhibit, bit 11 in RPOF) is set, the contents of RPEC2 are irrelevant. Valid counts include the entire ECC redundancy field.

RPEC2 is reset by any of the following.

- Drive Clear
- Massbus Initialize
- Power On Reset
- Initiation of a command Function Code and GO bit (bits 0-5 in RPCS1)
- Command continuance (RUN assertion at the fall of EBL).

Bits 0-10 (PAT) - The Pattern (PAT) bits compose the 11-bit error burst field; bit 0 is LSB.

Bits 11-15 - Always 0.

MICROPROCESSOR INTERFACE REGISTER
The 2901 and 8080 MPUs interact as Master and Slave depending on mode of operation. Interaction takes place through the Communications Register on A1A08 PCA (Command/Index/Sector).

The Communications Register is composed of four 8-bit latches that supply the data path from the 2901 to the 8080 via the Y Bus (16 bits) and from the 8080 to the 2901 via the S Bus (16 bits).

CONTROL AND STATUS (RPCS1)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R/W	0	0	0	0	DVA	0	0	0	0	0	F4	F3	F2	F1	F0	GO

MR-11393

DEVICE STATUS (RPDS)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R	ATA	ERR	PIP	MOL	WRL	LBT	PGM	DPR	DRY	VV=1	0	0	0	ILV	EWN	OM

MR-11394

ERROR 1 (RPER1)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R	DCK	UNS	OPI	DTE	WLE	IAE	AOE	HCRC	HCE	ECH	WCF	FER	PAR	RMR	ILR	ILF

MR-11395

MAINTENANCE 1 (RPMR1)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R/W	DMD	DIAG 64	DIAG 32	DIAG 16	DIAG 8	DIAG 4	DIAG 2	DIAG 1	P	A	R	A	M	E	T	E R

MR-11396

ATTENTION SUMMARY PSEUDO (RPAS)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R/W	0	0	0	0	0	0	0	0	ATA 7	ATA 6	ATA 5	ATA 4	ATA 3	ATA 2	ATA 1	ATA 0

MR-11397

DESIRED TRACK/SECTOR ADDRESS (RPDA)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R/W	0	0	TA 32	TA 16	TA 8	TA 4	TA 2	TA 1	0	SA 64	SA 32	SA 16	SA 8	SA 4	SA 2	SA 1

MR-11398

DRIVE TYPE (RPDT)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R	0	0	1	0	DRQ	0	0	0	0	0	1	0	0	0	DT 1	0

MR-11399

LOOK AHEAD (RPLA)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R	0	0	0	SC 64	SC 32	SC 16	SC 8	SC 4	SC 2	SC 1	0	0	0	0	0	0

MR-11400

SERIAL NUMBER (RPSN)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R	8K	4K	2K	1K	800	400	200	100	80	40	20	10	08	04	02	01

MR-11401

OFFSET (RPOF)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R/W	CMD	MTD	0	FMT 16	ECI	HCI	0	0	0	0	0	0	0	0	0	0

MR-11402

DESIRED CYLINDER ADDRESS (RPDC)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R/W	0	0	0	0	0	0	512	256	128	64	32	16	8	4	2	1

MR-11403

CURRENT CYLINDER ADDRESS (RPCC)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R	0	0	0	0	0	0	512	256	128	64	32	16	8	4	2	1

MR-11404

ERROR 2 (RPER2)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R	PGE	CPE	CPU	RWU 3	RWU 2	RWU 1	WOR	WRY UNS	E	R	R	—	C	O	D	E

MR-11405

ERROR 3 (RPER3)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R	BSE	SKI	DSE	WSF	CLF	LBC	LCE	PHF	DVC	IXU	DCU	SDF	DPE	SBE	SCF	RTO

MR-11406

ECC POSITION (RPEC1)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R	0	0	0	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

MR-11407

ECC PATTERN (RPEC2)

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R	0	0	0	0	0	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0

MR-11408

RP20 FSC LIST

FSC	Error Description
0000	FSC NOT GENERATED (LOAD ROUTINE 30)
0900	CHANNEL BUS OUT PARITY ERROR
0901	INVALID COMMAND
0902	INVALID SEQUENCE (CCW)
0903	CCW COUNT LESS THAN REQUIRED
0904	DATA VALUE NOT AS REQUIRED
0905	DIAG. WRITE INHIBITED BY FILE MASK
0906	CHANNEL ABORTED RETRY
0907	CHANNEL RETURNED WITH INCORRECT RETRY CCW
0908	MPL FILE NOT READY
0909	MPL FILE PERMANENT SEEK CHECK
090A	MPL FILE PERMANENT READ CHECK
090B	IMPROPER ALTERNATE TRACK POINTER
090C	PERMANENT DATA OVERRUN DETECTED
090D	INDEX DETECTED IN GAP OF RECORD
1000	FALSE DEVICE INTERFACE CHECK
1001	DEV INTF CHECK-TAG BUS PARITY CHECK
1002	DEV INTF CHECK-BUS OUT PARITY CHECK
1003	DEV INTF CHECK-TAG BUS AND BO PAR CHK.
11FF	HDA MODE PAR CHK-MULTI OR NOT FORMAT MODE
11XX	HDA SEQUENCE CHECK STATE 6
1200	TIMEOUT CK IN RECAL, ST 0-MOVE OUT
1201	TIMEOUT CK IN RECAL, ST 1-RESET
1206	TIMEOUT CK IN REZERO, ST 6-REZERO LIN MODE
1208	TIMEOUT CK IN SEEK, ST 8-DECELERATE
120A	TIMEOUT CK IN SEEK, ST A-ACCELERATE
120C	TIMEOUT CK IN SEEK, ST C-SEEK LIN MODE
120E	INV. T.O. CK IN SEEK, ST E-ON TRACK
1210	TIMEOUT CK IN REZERO, ST 10-MOVE OUT
1212	TIMEOUT CK IN REZERO, ST 12-TURN AROUND
1216	TIMEOUT CK IN REZERO, ST 16-MOVE IN
12XX	TIMEOUT CK IN AN INVALID CONTROL STATE
1301	SECTOR NON COMPR WITHIN TWO INDEX MARKS
1310	FALSE DRIVE CHECK
1400	FALSE R/W CHECK
1401	WRITE I CHECK
1402	TRANSITIONS CHECK
1404	CONTROL CHECK
1408	DELTA I/W CHECK
1410	INDEX CHECK SEL
1420	WRT OVERRUN LATCH
1440	CAPABLE ENABLE CHECK
1480	MULTICHIP CHECK
14F4	PAD GATE ERROR 5
14F8	HEAD SHORT LATCH
14XX	MULTIPLE R/W CHECKS
1500	OVERSHOOT CHECK DURING REZERO
1506	RECALIBRATE TRACK 0 OVERSHOOT CHECK
1508	OVERSHOOT CK IN SEEK, ST 8-DECELERATE
150A	OVERSHOOT CK IN SEEK, ST A-ACCELERATE
150C	OVERSHOOT CK IN SEEK, ST C-LIN MODE
150E	OVERSHOOT CK, LOST SERVO TRACK FOLLOWING
1510	OVERSHOOT CK DURING REZERO
1512	OVERSHOOT CK DURING REZERO
1516	OVERSHOOT CK DURING REZERO
15XX	OVERSHOOT CK IN AN INVALID STATE
160E	SERVO OFF TRACK ERR DURING ON TRACK STATE
16XX	SERVO OFF TRACK ERR DURING AN INV CTRL STATE OR SET R/W ACTIVE DURING ACCESS MOTION
1910	ERROR ALERT
1911	TRANSMIT TARGET ERROR
1912	MICROPROGRAM DETECTED ERROR (SENSE BYTE 18)
1913	DIFF COUNTER OR HAR FAILED TO REST ON A REZERO
1914	SYNC OUT TIMING CHECK
1915	UNEXPECTED FILE STATUS AT INTL SELECTION
1916	TRANSMIT CAR ERROR
1917	TRANSMIT HAR ERROR
1918	TRANSMIT DIFF COUNTER ERROR
1919	UNEXPECTED FILE STATUS IN READ IPL
191A	SEEK VERIFICATION CHECK
191B	SECTOR COMPARE CHK IF BYTE 9, BIT 1 ON TIMEOUT CHECK IF BYTE 16, BIT 0 ON OVERSHOOT CHECK IF BYTE 16, BIT 1 ON
191C	NO INTERRUPT FROM DRIVE (MISSING ATTENTION)
191D	DEFECT SKIPPING REORIENTATION ERROR
191E	UNABLE TO DETERMINE DEVICE FORMAT MODE
191F	RETRY REORIENTATION CHECK
2100	FILE INTERFACE CHECK

RP20 FSC LIST (Cont)

FSC	Error Description
2102	FILE INTERFACE TRANSFER CHECK
2104	FILE INTERFACE BUS OUT CHECK
2108	FILE INTERFACE TAG BUS PARITY CHECK
2110	FILE INTERFACE UNEXPECTED END CHECK
2120	FILE INTERFACE BUFFER PARITY CHECK
2140	FILE INTERFACE SELECT ACTIVE CHECK
2180	FILE INTERFACE LOGIC CHECK
2186	FILE INTERFACE LOGIC AND TRANSFER CHK
2188	FILE INTERFACE TAG BUS PARITY CHECK
21XX	IF BIT 0, 1, 2 OR 3 OF BYTE 20 IS ON, SUSPECT DEVICE
2202	COMPARE ASSIST CHECK
2204	LOAD S REG CHECK
2210	DATA TRANSFER CHECK
2220	INTERFACE CHECK CHANNEL B
2221	INTERFACE CHECK CHANNEL D
2240	INTERFACE CHECK CHANNEL A
2241	INTERFACE CHECK CHANNEL C
2280	CHANNEL BUFFER PARITY CHECK
2290	DATA TRANSFER CHECK
2920	CHECK 2 WITH NO BIT IN BYTE 11 OR 20
2923	S REGISTER LOAD ERROR
2924	CHECK 2-CI REGISTERS ARE VALID A CHECK 2 DETECTED IN A SEL SEQ WITH NO BITS IN BYTE 11 OR 20 REG TO 2920
3XXX	A FAILURE IN THE CHECK 1 REG CAN CAUSE ANY FORMAT 3 SYMPTOM
31XX	ERROR DETECTED IN CU CLOCK
3204	ERROR DETECTED IN CD DECODE CIRCUITRY
3220	ECC LOGIC FAILURE
3240	DOUBLE BIT ERROR
3260	ECC LOGIC AND DOUBLE BIT ERROR
3381	ERROR DETECTED IN SPEC. OP DECODE CIRC.
3382	ERROR DETECTED IN STATUS REG. OP CH/CL BRANCH CIRCUITRY
3402	MPL FILE NOT READY
3410	ERROR DETECTED IN CTRL STOR WRITE BUS 1/3
3420	ERROR DETECTED ON CTRL STOR WRITE BUS 0/2
3430	ERROR DET. ON CTRL STOR WRITE BUS 1/3 AND 0/2
3440	STORAGE ADR BUS 8-15 CHECK (IF BYTE 10, BIT 5 ON
3448	STORAGE ADR BUS 8-15 CHECK REF TO 3504)
3480	STORAGE ADR BUS 0-7 CHECK
3488	STORAGE ADR BUS 0-7 CHECK
34C0	STORAGE ADR BUS 0-7 AND 8-15 CHECK
34C8	STORAGE ADR BUS 0-7 AND 8-15 CHECK
3501	MPL READ CHECK, MPL PAR ERROR DETECTED
3502	ALU CHECK
3504	B REGISTER CHECK
3506	B REGISTER AND ALU CHECK
3508	A REGISTER CHECK
350A	A REGISTER AND ALU CHECK
3530	CHECK 1 ERROR BUT NO BITS ON IN BYTES 10 + 11
3930	CHECK 1 ERROR BUT NO BITS ON IN BYTES 10 + 11
4940	ECC DATA CHECK HA FIELD
4941	ECC DATA CHECK COUNT FIELD
4942	ECC DATA CHECK KEY FIELD
4943	ECC DATA CHECK DATA FIELD
4944	NO SYNC BYTE FOUND HA FIELD
4945	NO SYNC BYTE FOUND COUNT FIELD
4946	NO SYNC BYTE FOUND KEY FIELD
4947	NO SYNC BYTE FOUND DATA FIELD
4949	NO AM FOUND DURING RETRY
9001	MISSING TAG VALID ON R/W OPERATION
9002	NORMAL OR CHECK END MISSING FOLLOWING R/W OR ECC OPERATION
9003	NO RESPONSE FROM A CTRL MODULE ON A CONTROL OPERATION
9004	TIMEOUT WAITING FOR INDEX
9005	ECC HARDWARE CHECK
9006	MULTIPLE CONTROLLERS SELECTED
9007	PRESELECTION CHECK
9008	REPETITIVE CMD OVERRUNS ON G1 OPS.
9009	REP. CMD OVERRUNS ON G2 OR G3 OPS.
900A	PHYSICAL ADDRESS CHECK (WRONG ADR. RET.)
900B	BUSY MISSING AFTER SEEK START IS ISSUED
900E	DEVICE INTERFACE FAILURE
900F	ATTENTION CHECK (DEV ATTN FAILED TO RESET)
9101	REORIENT COUNTER CHECK
9102	TRACK COUNTER CHECK
9104	WRITE FAIL
9108	CONTROLLER BUS IN PARITY CHECK

RP20 FSC LIST (Cont)

FSC	Error Description
9110	DEVICE BUS IN PARITY CHECK
9118	DEV AND CONTR BUS IN PAR CHECK
9120	CHECK 1 OF 8
9140	BUS OUT PARITY CHECK
9180	TAG BUS PARITY CHECK
91FF	CONTR INTERFACE BUS IN ASSEMBLY FAILURE
91XX	SOME FAILURES CAUSE MULTIPLE FSC's
9200	FALSE CONTROLLER CHECK
9201	ECC 0 COMPARE (NORMAL COMPL OF R/W)
9202	ECC HARDWARE CHECK
9204	STATUS MONITOR CHECK
9208	WRITE DATA PARITY ERROR
9210	GAP COUNTER CHECK
9220	SHIFT REGISTER ERROR
9240	MISSING SERVO DATA
9280	VFO PHASE ERROR
92C0	MISSING READ DATA
92XX	SOME FAILURES CAUSE MULTIPLE FSC's
93XX	INVALID FAULT SYMPTOM CODE

MODULE ADDRESS JUMPERS RP20

CJ03 OR CJ04	
ON	1
C	2
OFF	3
ON	4
C	5
OFF	6
ON	7
C	8
OFF	9

RP20 COMMANDS

Command		MT OFF*	MT ON*	Count
Control	Orient(c)	28		Nonzero
	Recalibrate	13		Nonzero
	Seek	07		6
	Seek Cylinder	0B		6
	Seek Head	1B		6
	Space Count	0F		3(a); nonzero (d)
	Set File Mask	1F		1
	Set Sector (a,f)	23		1
	Restore (executes as a no-op)	17		Nonzero
	Vary Sensing(c)	27		1
	Diagnostic Load (a)	53		1
	Diagnostic Write (a)	73		512
Search	Home Address Equal	39	B9	4
	Identifier Equal	31	B1	5
	Identifier High	51	D1	5
	Identifier Equal or High	71	F1	5
	Key Equal	29	A9	KL
	Key High	49	C9	KL
	Key Equal or High	69	E9	KL
	Key and Data Equal (d)	2D	AD	Number of bytes (including mask bytes) in search argument
	Key and Data High (d)	4D	CD	
	Key and Data Equal or Hi (d)	6D	ED	
Continue Scan	Search Equal (d)	25	A5	
	Search High (d)	45	C5	
	Search High or Equal (d)	65	E5	
	Set Compare (d)	35	B5	
	Set Compare (d)	75	F5	
	No Compare (d)	55	D5	
Read	Home Address	1A	9A	5
	Count	12	92	8
	Record 0	16	96	Number of bytes to be transferred
	Data	06	86	
	Key and Data	0E	8E	
	Count, Key and Data	1E	9E	
	IPL	02		
	Multiple Count, Key, Data (b)	5E		> Max. track len.
	Sector (a,f)	22		1
Sense	Sense I/O	04		24 (a), 6 (d)
	Sense I/O Type (b)	E4		7
	Read, Reset Buffered Log (b)	A4		24
	Read Buffered Log (c)	24		128
	Device Release (e)	94		24 (a); 6 (d)
	Device Reserve (e)	B4		24 (a); 6 (d)
	Read Diagnostic Status 1 (a)	44		16 or 512
Write	Home Address	19		5, 7, or 11
	Record 0	15		8+KL+DL of R0
	Erase	11		8+KL+DL
	Count, Key and Data	1D		8+KL+DL
	Special Count, Key and Data	01		8+KL+DL
	Data	05		DL
	Key and Data	0D		KL+DL

* Code same as MT Off except as listed.
a. Except 2314, 2319
b. 3330-3340-3350 series only.
c. 2305/2835 only.
d. 2314, 2319 only.
e. String switch or 2-channel switch required.
f. Special feature required on 3340.

NONLINKED ROUTINES

Routine Number Hex	
D8	SERVO ADJUSTMENT
D9	INCREMENTAL SEEK
DA	CYL-CYL SEEK
DB	RANDOM SEEK
DC	PUMPED RESONANCE
DD	CRASH STOP
DE	SERVO MARGIN
E0	SYNC UTILITY
E1	READ UTILITY
E2	DISPLAY HA
E5	DISPLAY DRV CONFIG/SN
E6	DISPLAY MEMORY
E7	MEMORY SCAN
E8	FE PANEL
E9	HDA STATE ANALYSIS
EA	DISPLAY SENSE DATA
EB	TAG UTILITY
EC	STRING SW/DUAL PORT (FEATURE)
EE	MANUAL INTERVENTION
F2	TRACK ANALYSIS

LINKED SERIES ROUTINES

Routine Number Hex		
C1	CONTROL INTERFACE	
C2	DRIVE INTERFACE	
C3	BASIC SERVO	
C4	INDEX AND SECTOR	
C5	GAP COUNTER	
C6	BASIC READ-WRITE	
C7	PADDING	
C8	ECC LOGIC	
C9	REORIENT CTR/TR CTR	Drive must be ready
CA	COMPLEX SERVO	
CB	R/W RELIABILITY	
CC	R/W MARGIN	
CD	AM DETECTION	
CE	OVERWRITE	
CF	REFORMAT FE TRACKS	

PROGRAM CONTROL DATA DISPLAYS

Program Control		Data
82	ROUTINE LOADING	RTN NO.
8C	ROUTINE RUNNING	RTN NO.
8D	DYNAMIC ERROR DISPLAY REPEAT ERROR TEST AFTER ERROR	ERROR NO.
CO	INVALID ROUTINE OR SYSTEM RESET	RFTN NO. 00
CA	ROUTINE READY FOR EXECUTION	RTN NO.
CE	MANUAL INTERVENTION REQUIRED OR DISPLAY COMPLETE	RTN NO.
CF	NORMAL END	RTN NO.
DX	PARAMETER ENTRY REQUIRED	RTN NO.
E1	ERROR/MESSAGE STOP	ERROR/MSG NO.
EX	ERROR/MESSAGE BYTE BITS 4-7: BYTE NUMBER BEING DISPLAYED	BYTE

CONTROL OPTIONS

Hex	Entry
CX	DRIVE SELECTION
00	START/STOP ROUTINE EXECUTION
10	PARAMETER ENTRY
20	START OR ADVANCE ERROR/MESSAGE DISPLAYS
30	RESET DIAGNOSTIC CONTROL 8000 MOD-II AND ISC-RESTORE FAULT SYSTEM CODE GENERATOR IN OVERLAY AREA. MOUNT FUNCTIONAL FLOPPY IF ISC.

COMMON ERROR STOPS

Program Control Display: E1

Data Display:

01	INVALID PARAMETER ENTRY
02	DRIVE NOT ONLINE
03	NO PHYSICAL ADDRESS FROM FE DRIVE. CHECK SWITCHES
04	NOT WRITE ENABLED
05	MULTIPLE DRIVE SELECT CHECK FE MODE SWITCHES
06	INCORRECT DRIVE TYPE
07	HDA SEQUENCE ERROR, CHECK FOR DRIVE READY OR RUN ROUTINE E9
08	NO TAG VALID

RUN OPTIONS

Hex	Entry
01	DYNAMIC ERROR DISPLAY REPEAT TEST AFTER ERROR
02	LOOP ROUTINE(S)
03	DYNAMIC ERROR DISPLAY/LOOP ROUTINE(S)
04	INHIBIT LINKING
05	INHIBIT LINKING/DYNAMIC ERROR DISPLAY
06	LOOP SINGLE ROUTINE
07	LOOP SINGLE ROUTINE/DYNAMIC ERROR DISPLAY
08	RESET OPTION

BIT SIGNIFICANCE OF RUN OPTION ENTRY:

```
DYNAMIC ERROR DISPLAY
REPEAT TEST AFTER ERROR
LOOP ROUTINE(S)
INHIBIT LINKING
RESET OPTION

0   1   2   3   4   5   6   7

0   0   0   0   X   X   X   X
```

INFORMATION COMMON TO ALL FORMATS

BYTE \ BIT	0	1	2	3	4	5	6	7
0	COMMAND REJECT	INTERVENTION REQUIRED	BUS OUT PARITY	EQUIPMENT CHECK	DATA CHECK	OVERRUN	NOT USED	NOT USED
1	PERMANENT ERROR	INVALID TRACK FORMAT	END OF CYLINDER	NOT USED	NO RECORD FOUND	FILE PROTECT	WRITE INHIBIT	OPERATION INCOMPLETE
2	NOT USED	CORRECTABLE	NOT USED	ENVIRONMENTAL DATA BYTE 8-23	EMULATION	NOT USED	NOT USED	NOT USED
3	RESTART COMMAND BYTE 1 BIT 7=0 - LAST CHANNEL COMMAND IN THE CCW BYTE 1 BIT 7=1 - OPERATION IN PROGRESS WHEN OBR WAS GENERATED							
4	PHYSICAL DRIVE IDENTIFICATION							
	DRIVE 8 OR 0	DRIVE 9 OR 1	DRIVE A OR 2	DRIVE B OR 3	DRIVE C OR 4	DRIVE D OR 5	DRIVE E OR 6	DRIVE F OR 7
5	LOGICAL CYLINDER ADDRESS LOW							
	128	64	32	16	8	4	2	1
6	CYL ADD HIGH 1024	CYL ADD HIGH NAT/-11 512	CYL ADD HIGH NAT/-11 256	LOGICAL TRACK (HEAD)				
				16	8	4	2	1

MR-6083

FORMAT 0 MESSAGE ONLY SENSE BYTES 8-23 NOT USED SENSE BYTE 7 FORMAT/MESSAGE

	0	1	2	3	4	5	6	7
MESSAGE	NO MESSAGE	INVALID COMMAND	INVALID SEQUENCE	CCW COUNT LOW	DATA ARGUMENT INVALID	DIAG/W INHIBITED BY FILE MASK	CHANNEL ABORTED RETRY	CHANNEL CCW 1 INCORRECT ON RETRY
	8	9	A	B	C	D	E	F
MESSAGE	MPL FILE NOT READY	MPL FILE PERMANENT SEEK CHECK	MPL FILE PERMANENT READ CHECK	COMMAND OVERRUN	DATA OVERRUN	DEFECTIVE TRACK	ALTERNATE TRACK	NOT USED

MR-6084

FORMAT 1 DRIVE EQUIPMENT CHECK SENSE BYTE 7 FORMAT/MESSAGE

	0	1	2	3	4	5	6	7
MESSAGE	NOT USED	TRANSMIT TARGET ERROR	MICRO-PROGRAM DETECTED ERROR	TRANSMIT DIFFERENCE HIGH ERR	SYNC OUT TIMING ERROR	UNEXPECTED DRIVE STATUS AT INITIAL SELECTION	TRANSMIT CYL ADDR REGISTER ERR	TRANSMIT HEAD ERROR
	8	9	A	B	C	D	E	F
MESSAGE	TRANSMIT DIFFERENCE ERR	DRIVE STAT NOT AS EXPECTED DURING RD IPL	SEEK VER CHECK ON PHYSICAL ADDRESS	SEEK INCOMPLETE OR SECTOR COMPARE CHECK	NO INTR FROM DRIVE	DEFECT SKIPPING OR REORIENTATION CHECK	NOT USED	RETRY REORIENTATION CHECK

MR-6085

FORMAT 1 SENSE BYTES 8-13

BYTE \ BIT	0	1	2	3	4	5	6	7
DRIVE STATUS BYTE 19 BIT 0=0 8	CONTROLLER CHECK (REF BYTE 17&20) CC170	TAG BUS OR BUS OUT PAR DG050	ACCESS CHECK SECTOR NONCOMPARE DP050	RD/WR CHECK (REF BYTE 12&19) DE005	ONLINE DF030	ACCESS HDA ATTN DF030	BUSY DF040	SK CMPT SK SCTR PAD CMPT DF070
BYTE 19 BIT 0=1 WRITE OR READ		WRITE 1 DD010				PAD IN PROGRESS DE030	INDEX MARK DE015	3330 MODES DH005
9 CHECK STATUS	PAD STATUS DE030	SECTOR NON COMPARE DH060	MOTOR AT SPEED LTH DH010	AIR SWITCH ON LTH DH010	WRITE ENABLE DH010	FIXED HEADS DH005	3330-1 .... 10 8350 .... 01* 3330-11 .... 11* DH005	
10 HDA/SEQ CONTROL	FMT ERR/ FMT LTH ERROR	HDA SEQUENCE STATE LATCH 4	2	1	HDA SEQUENCE CHECK LTH	INHIBIT HDA RECYCLE	GEMINI HDA	ODD TRACK STAT 7
11 LOAD SW STATUS	DRIVE START LTH DH010	GUARD BAND LTH DA005	TARGET VELOCITY DB030	TRACK CROSSING DA025	NOT USED	AIR SWITCH ON DH085	GEMINI HDA DOUBLE DENSITY	MOTOR AT SPEED DH010
12	MULTICHIP CHECK DE010	CAPABLE ENABLE CHECK DE010	WRITE OVERRUN LTCH DE015	INDEX CHECK SEL DA005	DELTA I/W CHECK ** DE025	CONTROL CHECK DE010	TRANSITIONS CHECK DE035	WRITE 1 CHECK DE035
13 MESSAGE CODE 2 AND C	BUS OUT AT TIME OF ERROR WHEN SENSE BYTE 18=01,03,05,06 OR 0C							
13 MESSAGE CODE A OR B	LOGICAL CYL ADDRESS LOW PRIOR TO SENSE BYTE 5 128	64	32	16	8	4	2	1

MR-6091

FORMAT 1 SENSE BYTES 13-18

<table>
<tr><th>BYTE \ BIT</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th></tr>
<tr><td>13
MESSAGE CODE
1,3,5,6, 7,8,&9</td><td colspan="8">EXPECTED DRIVE STATUS/DATA</td></tr>
<tr><td>14
MESSAGE OTHER
THAN A&B</td><td colspan="8">CONTROL INTERFACE BUS IN AT TIME OF FAILURE</td></tr>
<tr><td rowspan="2">14
MESSAGE
CODE A&B</td><td colspan="2">IF BIT 0&1 = 11
FIXED HEAD
LOGICAL CYL ADDR HIGH PRIOR TO
BYTE 6</td><td colspan="6">LOGICAL TRACK PRIOR TO SENSE BYTE 6
TRACK = HEAD</td></tr>
<tr><td>512</td><td>256</td><td>32</td><td>16</td><td>8</td><td>4</td><td>2</td><td>1</td></tr>
<tr><td>15</td><td colspan="8">CONTROL INTERFACE TAG BUS AT THE TIME OF THE DETECTED ERROR</td></tr>
<tr><td>16</td><td>TIME
OUT
CHECK
DF040</td><td>OVER SHOOT
CHECK
DF030</td><td>SERVO
OFF-TRACK
DF010</td><td>REZERO
MODE
LATCH
DF050</td><td>SERVO
LATCH
DF010</td><td>LINEAR
MODE
LATCH
DF010</td><td>CONTROL
LATCH
DF010</td><td>WAIT
LATCH
DF040</td></tr>
<tr><td>17</td><td colspan="2">VFO PHASE CK
01 = MISSING SERVO DATA
10 = VFO PHASE ERR
11 = MISSING READ DATA CC140</td><td>SHIFT REG
ERR
CC140</td><td>GAP
CNTR
CHECK
CF120</td><td>WRT DATA
PARITY
ERROR
CC140</td><td>STATUS
MONITOR
CHECK
CJ160</td><td>ECC
HARDWARE
CHECK
CD100</td><td>ECC
O
COMPARE
CD100</td></tr>
<tr><td>18</td><td colspan="4">NOT USED</td><td colspan="4">CODED ERROR CONDITION
(BITS 4-7 HEX) LISTED BELOW</td></tr>
</table>

BYTES 18, BITS 4-7

0	1 !	2	3 !	4	5 !	6 !	7
NOT USED	NO TAG VALID ON R/W OPERATION	NO NORMAL OR CHECK END ON R/W OR ECC OPERATION	NO RESPONSE FROM CNT MOD ON CNT OPERATION	TIME OUT WAITING FOR INDEX	ECC HARDWARE CHECK	MULTIPLE OR NO CNT MOD SELECTED	PRESELECTION CHECKS
8	**9**	**A**	**B**	**C !**	**D**	**E**	**F**
REPETITIVE CMND OVERRUNS ON G1 OPERATIONS	REPETITIVE CMND OVERRUNS ON G2 OR G3 OPERATIONS	POLL OR 1 OF 8 DECODE ERROR	BUSY MISSING AFTER SEEK START IS ISSUED	DEVICE TYPE ERROR	CHANNEL SELECT ERROR	PRESELECTION DISK CONTROL INTERFACE BUS	UNRESETABLE INTERRUPT

NOTE
! SENSE BYTES 13,14,15 ARE VALID FOR THESE MESSAGES

MR-6092

FORMAT 1 SENSE BYTES 19-23

BYTE \ BIT	0	1	2	3	4	5	6	7
19	SET R/W OPERATION 85 CH100	NOT USED	NOT USED	NOT USED	HEAD SHORT LATCH DE015	PAD GATE ERROR 5 DE110	1,2 MB FILE	ALWAYS ON
20	TAG BUS PARITY CHECK CH120	BUS OUT PARITY CHECK LATCHED CH120	CHECK 1 OF 8 CJ150	DEVICE BUS IN PARITY CHECK LATCHED CJ150	CONTROLLER BUS IN PARITY CHK LATCHED CD180	CURRENT (I) WRITE CHECK CC170	TRACK COUNTER CHECK GTD CC170	REORIENT COUNTER CH150
20 MESSAGE A AND BYTE 0 BIT 3=1	128	64	LOGICAL CYLINDER ADDRESS LOW 32	16	8	4	2	1
21 MESSAGE A AND BYTE 0 BIT 3=1	BITS 0 & 1=11 FIXED HEAD; LOGICAL CYLINDER ADDRESS HIGH 512	256	LOGICAL TRACK (HEAD) 32	16	8	4	2	1
21	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	BUS OUT PARITY (BO PAR)	TAG BUS PARITY
22-23	FAULT SYMPTOM CODE							

MR-6093

FORMAT 2 DCU ERROR SENSE BYTE 7 FORMAT/MESSAGE

	0	1	2	3	4	
MESSAGE	NO MESSAGE	NOT USED	NOT USED	S REG LOAD CHECK	CTL INTF REG VALID SENSE BYTES 13-15	MESSAGE 5-F NOT USED

MR-6086

FORMAT 2 DCU CHECK SENSE BYTES 8-23

BYTE \ BIT	0	1	2	3	4	5	6	7
8-10	NOT USED							
11 CONTROL CHECK	CHANNEL BUFFER PARITY CHECK	INTERFACE CHECK CHANNEL A OR C	INTERFACE CHECK CHANNEL B OR D	DATA TRANSFER CHECK	CONTROL INTERFACE LOGIC CHECK	LOAD 5 REGISTERS CHECK	COMPARE ASSIST CHECK	CHANNEL C/D OR MULTI-CONNECT
12	SET TO 0							
13	CONTROL INTERFACE (CONTENTS OF TA REGISTERS. VALID ONLY IF SENSE BYTE 7 IS 24)							
14	CONTROL INTERFACE BUS-IN (CONTENTS OF MA REGISTER. VALID ONLY IF SENSE BYTE 7 IS 24)							
15	CONTROL INTERFACE BUS-IN (CONTENTS OF MD REGISTER, VALID ONLY IF SENSE BYTE 7 IS 24)							
16-19	NOT USED SET TO 0							
20	CONTROL MODULE CHECK	CONTROL MODULE ACTIVE CHECK	CONTROL MODULE BUFFER PARITY CHECK	CONTROL MODULE UNEXPECTED END CHECK	CONTROL MODULE TAG BUS CHECK	CONTROL MODULE BUS-OUT CHECK	CONTROL MODULE TRANSFER CHECK	NOT USED
21	NOT USED SET TO 0							
22-23	FAULT SYMPTOM CODE							

MR-6094

FORMAT 3 SENSE BYTE 7 FORMAT/MESSAGE

MESSAGE	FORMAT 3 SENSE BYTE 7/MESSAGE NOT USED SELECTIVE RESET

MR-6087

FORMAT 3 SENSE BYTES 8-23

BYTE \ BIT	0	1	2	3	4	5	6	7
8	FAILING ADDRESS (BACK-UP ADDRESS REGISTER BUS 0-7)							
9	FAILING ADDRESS (BACK-UP ADDRESS REGISTER BUS 8-13)							
10	BIT 0 = 1 EARLY ERROR BIT 0 = 0 LATE ERROR	CLOCK ERROR	0	0	0 A REG CHECK	0 B REG CHECK	0 ALU CHECK	SPECIAL OP ERROR MPL READ CHECK
11 SENSE BYTE 10 BIT 0 = 1	0	STORE MULTIPLE READ ERROR	STORE ECC LOGIC ERROR	0	0	CD DECODE ERROR	0	0

BYTE \ BIT	0	1	2	3	4	5	6	7
11 SENSE BYTE 10 BIT 0 = 0	STORE ADDRESS BUS 0-7 CHECK	STORE ADDRESS BUS 8-13 CHECK	STORE WRITE BUS 2/3 CHECK	STORE WRITE BUS 0/1 CHECK	0	0	MPL NOT READY	0
12	SYNDROME REGISTER							
13	TC REGISTER (THIS REGISTER IS RESET IF SELECTIVE RESET OCCURRED IN RESPONSE TO DISCONNECT IN)							
14	TG REGESTER (THIS REGISTER IS RESET IF SELECTIVE RESET OCCURRED IN RESPONSE TO DISCONNECT IN)							
15-23	NOT USED SET TO 0							

MR-6095

FORMAT 4 DATA CHECKS UNCORRECTABLE SENSE BYTE 7 FORMAT/MESSAGE

	0	1	2	3	4	5	6	7
MESSAGE	HA ECC DATA CHECK	COUNT FIELD DATA CHECK	KEY FIELD DATA CHECK	DATA FIELD DATA CHECK	HA FIELD NO SYNC BYTE FOUND	COUNT FIELD NO SYNC BYTE FOUND	KEY FIELD NO SYNC BYTE FOUND	DATA FIELD NO SYNC BYTE FOUND
	8	9						
MESSAGE	NOT USED	AM DETECTION FAILURE ON RETRY	A-F NOT USED					

MR-6088

FORMAT 4 DATA CHECKS NOT PROVIDING DISPLACEMENT INFORMATION SENSE BYTES 8-23

BYTE \ BIT	0	1	2	3	4	5	6	7
8	CYLINDER ADDRESS OF THE RECORD IN ERROR							
	0	0	0	0	0	0	512	256
9	CYLINDER ADDRESS OF THE RECORD IN ERROR							
	128	64	32	10	8	4	2	1
10	HEAD ADDRESS OF THE RECORD IN ERROR							
	0	0	0	0	0	0	0	0
11	HEAD ADDRESS OF THE RECORD IN ERROR							
	0	0	0	16	8	4	2	1
12	RECORD NUMBER (UNRELIABLE MESSAGE 0 OR 4, ERROR HA) (UNRELIABLE MESSAGE 1 OR 5, ERROR COUNT FIELD)							
13	SECTOR NUMBER OF THE RECORD IN ERROR							
	128	64	32	16	8	4	2	1
14-21	NOTE BYTE 15 = RETRY COUNT NOT USED							
22-23	FAULT SYMPTOM CODE							

MR-6096

FORMAT 5 DATA CHECKS CORRECTABLE FORMAT/MESSAGE

	0	1	2	3	
MESSAGE	NOT USED	NOT USED	NOT USED	DATA FIELD CORRECTABLE DATA CHECK	MESSAGES 4 - F NOT USED

MR-6089

FORMAT 5 DATA CHECKS PROVIDING DISPLACEMENT INFOMATION SENSE BYTES 8-23

BYTE \ BIT	0	1	2	3	4	5	6	7
8	CYLINDER ADDRESS OF THE RECORD IN ERROR							
							512	256
9	CYLINDER ADDRESS OF THE RECORD IN ERROR							
	128	64	32	16	8	4	2	1
10	HEAD ADDRESS OF THE RECORD IN ERROR							
11	HEAD ADDRESS OF THE RECORD IN ERROR							
	0	0	0	0	0	0	0	0
12	RECORD NUMBER SET TO 0 IF ERROR OCCURRED IN HA							
				16	8	4	2	1

13	SECTOR NUMBER OF THE RECORD IN ERROR
14	NOT USED
15,16,17	IDENTIFIES THE NUMBER OF BYTES PROCESSED BY THE DCU FROM THE INITIATION OF DATA TRANSFER AND THE END OF THE DATA FIELD
18-19	ERROR DISPLACEMENT, SPECIFIES THE FIRST BYTE IN ERROR WITHIN THE DATA FIELD WITH RELATIONSHIP TO THE END OF THAT DATA FIELD
20-21	ERROR CORRECTION PATTERN (EACH BIT IN ERROR WILL BE INDICATED BY A 1)
22	ALWAYS 0
23	NOT USED

MR-6097

FORMAT 6 USAGE AND OVERRUN ERROR STATISTICS SENSE BYTES 8-23

BYTE \ BIT	0	1	2	3	4	5	6	7
8-11	NUMBER OF BYTES READ OR SEARCHED (KEY AND DATA FIELD ONLY)							
12-13	NOT USED SET TO ZERO							
14-15	NUMBER OF DATA CHECKS SUCCESSFULLY RETRIED							
16-17	NUMBER OF ACCESS MOTIONS							
18	CHANNEL SELECT FOR SENSE BYTES 20-23 BIT 0=0 INFORMATION APPLIES TO INTERFACES A AND B IF BIT 0=1 INFORMATION APPLIES TO INTERFACES C & D BITS 1-7 NOT USED							
19	TOTAL SEEK ERRORS RETRIED							
20	COMMAND OVERRUNS A (C)							
21	DATA OVERRUNS A (C)							
22	COMMAND OVERRUNS B (D)							
23	DATA OVERRUNS B (D)							

MR-6098

FORMAT 6 USAGE AND OVERRUN ERROR STATISTICS

MESSAGE	FORMAT 6 - MESSAGE 0-F ARE NOT USED

MR-6090

S/X Bus

Table of Contents

GENERAL DESCRIPTION

Introduction

This section provides all the information required by an experienced Field Service engineer to use the S/X Bus Recorder to diagnose KL10 SBus errors. Included are instructions for connecting the recorder into a system, explanations of the functions of all controls and indicators, and error interpretation.

Physical Description

The S/X Bus Recorder is contained in a portable, suitcase-like aluminum housing. All controls and indicators are located on a console panel. The SBus input and output connectors, and the power connector are stored in a compartment at the back of the case. The S/X Bus Recorder is available in two models:

1. Part number 9307042-00; 117 Vac, 60 Hz
2. Part number 9307042-01; 234 Vac, 50 Hz

Console Description - All controls and indicators for the S/X Bus Recorder are located on the console panel. These controls and indicators consist of a function select switch, register and RAM address select thumbwheel switches, a 6-digit octal LED readout, a 2-digit octal LED readout, and a series of toggle switches for various control functions.

Logic Modules - All logic modules in the S/X Bus Recorder are plugged into a 9-slot hex backplane. The board complement consists of five double height modules and five hex modules in eight slots. One slot is unused. The module utilization is as follows (see Figure 1).

Slot	Module	Function
IA/F1	G5348 Hex	Recorder logic
ICD2	G5349 Double	Console
IA/F3	G5347 Hex	Translator
IA/F4	G5347 Hex	Translator
ICD5	M9006 Double	Cable connector
IEF5	M9006 Double	Cable connector
IA/F6	M8572 Hex	Cable connector
ICD7	M9005 Double	Terminator or cable connector
IEF7	M9005 Double	Terminator or cable connector
IA/F8	M8572 Hex	Terminator

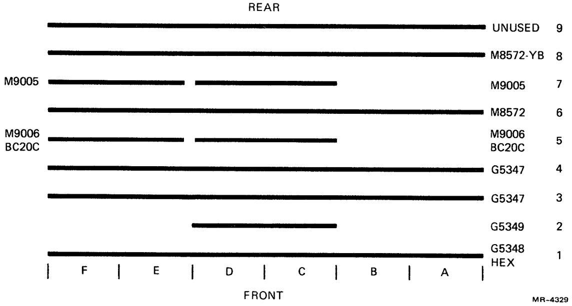


Figure 1 Backplane Layout (Top View)

Power Supply - DC operating voltages for the S/X Bus Recorder are provided by an L-H research model TM-34 power supply. The power supply outputs are as follows.

1. -5.2 Vdc @ 13 A
2. -2.0 Vdc @ 8 A
3. 5 Vdc @ 5.5 A
4. 5 Vdc @ 1.5 A

The power supply is contained on a subassembly that is 69.9 cm X 15.24 cm X 33 cm, 2.04 kg (2.75 in. X 6.0 in. X 13.0 in. and weighs 4.5 pounds).

The power supply is adjusted during its manufacture and should not normally require readjustment. If the voltages must be readjusted, proceed as follows.

1. Remove all screws from the bottom of the suitcase and remove eight screws from the recorder console top.

2. Rotate the two console panel locking screws 90 degrees and lift panel upward. Disconnect the Mate-N-Lok plug connecting the two cooling fans to ac power.

3. Remove the recorder assembly from the suitcase to gain access to the backplane for voltage measurements.

4. Verify that the following modules are present in the backplane to provide a load: 2-G5347, G5348, G5349.

5. Plug recorder into ac receptacle or set power switch to ON.

6. Connect a digital voltmeter to the test points listed below and adjust the appropriate potentiometer for the correct voltage. Access to the potentiometer is gained through a rectangular hole in the recorder housing. Refer to Figure 2.

Voltage	**Test Point**	**Potentiometer**
-5.20	C03B2	V1
-2.00	C03B1	V2
+5.00	C03A2	V3
+5.00	+5 V TAB (IND.BD)	V4

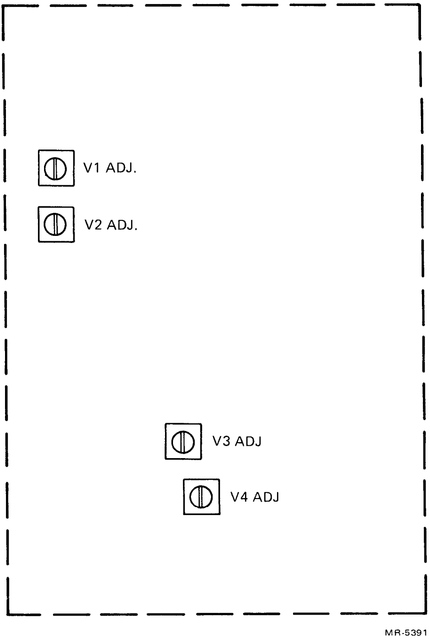


Figure 2 Potentiometer Locations

Cooling - Cooling for the S/X Bus Recorder is provided by two 4-inch fans located in the suitcase assembly. The fans are different for the 50 Hz and 60 Hz versions of the S/X Bus Recorder.

INSTALLATION

Introduction

This section provides all the information required to connect the S/X Bus Recorder into a system and disconnect it after testing is completed.

There are three variations of cabling, depending on the type of system being tested:

1. SBus using MA20 or MB20 memory
2. SBus using DMA
3. XBus using MF20 memory

After installation of the cables, the KL10 SBus recorder combination must be powered up and deskewed to align the clock in the S/X Bus Recorder to the clock in the KL10. This procedure is common to all cabling variations. After the deskewing procedure is completed, testing can proceed using the S/X Bus Recorder console panel and SBus diagnostic. On completion of testing, all recorder clocks must be checked and deskewed, cabling disconnected, and bus terminations replaced.

Cabling into System

Perform Procedures 1, 2, or 3 below, as appropriate, to connect the S/X Bus Recorder into the system. Set the bus and address switches as described in the Switch Settings section. Then proceed to the Deskewing Procedure section to perform the deskewing procedure.

Procedure 1: SBus with MA20/MB20 Memory

1. Power down the KL10.

2. Open the lid of the S/X Bus Recorder.

3. Remove the BC20 cables from the cable compartment in the recorder.

4. Rotate the two fasteners at the front of the console panel on the recorder one-quarter turn counterclockwise and hinge the panel upward.

5. Remove the two double height SBus terminator boards from MA20/MB20 memory in slots 1AB52 and 1CD52.

6. Install the terminator board removed from 1AB52 in slot 1CD7 of the S/X Bus Recorder.

7. Install the terminator board removed from 1CD52 in slot 1EF7 of the S/X Bus Recorder.

8. Plug one end of a BC20C-6C cable into slot 1CD5 of the recorder, and plug the other end into slot 1AB52 of the MA20/MB20.

9. Plug one end of the other BC20 cable into slot 1EF5 of the recorder, and the other end into slot 1CD52 of the MA20/MB20.

10. This completes cabling the recorder into the system. Proceed to the Switch Settings section and set the bus and address switches.

Procedure 2: SBus with DMA20

1. Power down the KL10.

2. Open the lid of the S/X Bus Recorder.

3. Remove the BC20C-6C cables from the cable compartment in the recorder.

4. Rotate the two fasteners at the front of the console panel on the recorder one-quarter turn counterclockwise and hinge the panel upward.

5. Remove the BC20C-6C cable from DMA slot 1AB01 and install it in recorder slot 1CD5.

6. Connect one end of a BC20C-6C cable into slot 1CD7 on the recorder, and the other end to slot 1AB01 on the DMA.

7. Remove the BC20C-6C cable from DMA slot 1CD01 and install it in recorder slot 1EF5.

8. Connect the remaining BC20C-6C cable from slot 1EF7 on the recorder, and slot 1CD01 on the DMA.

9. This completes cabling the recorder into the system. Proceed to the Switch Settings section and set the bus and address switches.

Procedure 3: XBus with MF20 Memory

1. Power down the KL10.

2. Open the lid of the S/X Bus Recorder.

3. Rotate the two fasteners at the front of the console panel on the recorder one-quarter turn counterclockwise and hinge the panel upward.

4. Remove the M8572YB board from slot 1A/F8 of the recorder.

5. Remove four terminations from PC22, PD22, PE22, and PF22 in the MF20.

6. Install these terminations on the M8572YB module.

7. Plug the M8572YB module into slot 1A/F8 of the recorder.

8. Connect the cables from the M8572 in slot 1A/F6 as follows.

From	To
P1	MF20/PC22
P2	MF20/PD22
P3	MF20/PE22
P4	MF20/PF22

9. This completes cabling the recorder into the system. Proceed to the Switch Settings section and set the bus and address switches.

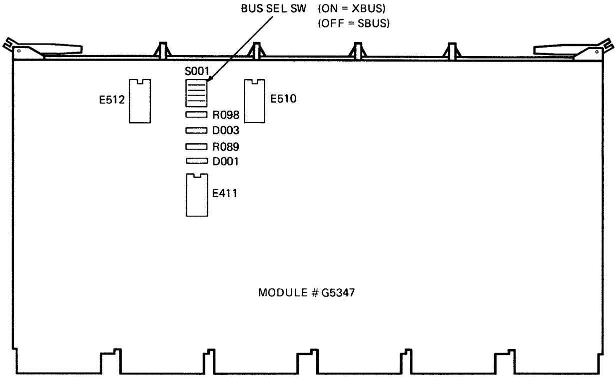
Switch Settings - Switches on the G5347 boards select whether the recorder is used with the SBus or XBus. A switch on the G5348 board selects the recorder address if two recorders are used.

The bus select switches are S001 on each of two G5347 boards located in slots three and four of the recorder. (Refer to Figure 3.)

Set both switches to ON if the recorder is to be used on the XBus, or OFF for the SBus.

If two recorders are used, one is assigned address 36 and the other 37. The address switch is located at S001 on the G5348 board in slot one of each recorder, as shown in Figure 4.

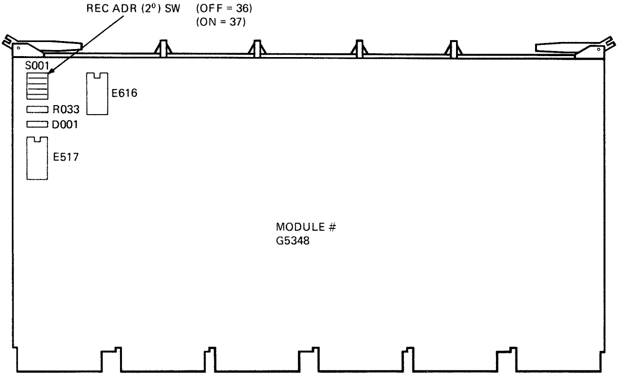
Set the switch to OFF if the recorder address is 36, and ON if the recorder address is 37.



NOTE
THE S/X BUS RECORDER CONTAINS TWO G5347 MODULES THE BUS SELECT SWITCH MUST BE IN THE SAME POSITION ON EACH MODULE

MR-4557

Figure 3 Bus Select Switch Location



MR-4558

Figure 4 Recorder Address Switch Location

Deskewing Procedure - Perform either the MA20/MB20 or DMA20 deskew procedure, as appropriate. These procedures are in the Checks and Adjustments section of the KL10 Maintenance Guide, Volume I (EK-0KL10-MG). Then perform the following procedure to align the recorder clock to the KL10 clock. This procedure requires a Tektronix 475 oscilloscope or equivalent (100 MHz). Use identical probes and short ground clips.

1. Power up the KL10 and select CR0 on the KL10.
2. Type MR and FX1 to turn on the clock.
3. Set oscilloscope for external trigger, negative edge, and trigger from A CHANGE COMING L, at 4E22F2 on the KL10 backplane.
4. Attach channel one probe to MTR BOX CLK C, 4D33P1. Set the scope to 0.5 V/cm with the ground reference 1.3 V above the centerline.
5. Press TRIGGER VIEW and observe that the relationship of MTR MBOX CLKC to A CHANGE COMING L corresponds to that shown in Figure 7.
6. Set the leading edge of the first A phase clock on the first division of the scope graticule.
7. Connect channel two probe of the scope to R31 on the G5348 in the recorder, as shown in Figure 5.
8. Adjust CLKA delay line on the G5349 in slot 1CD2 of the recorder (see Figure 6) so that the leading edge 50% point crosses the leading edge 50% point of MBox A phase clock (see Figure 7).
9. Connect channel two probe to R32 on the G-5348 board in the recorder, as shown in Figure 5.
10. Adjust CLKB delay line on the G5349 board in slot 1CD2 of the recorder (see Figure 6) so that the leading edge 50% point crosses the leading edge 50% point of MBox phase B clock (see Figure 7).
11. This completes the deskewing procedure for the S/X Bus Recorder.

Removal of Interconnecting Cables
At the completion of testing, remove all cables and replace terminations in the original locations. This procedure is the reverse of installation. Recheck clocks and deskew if necessary.

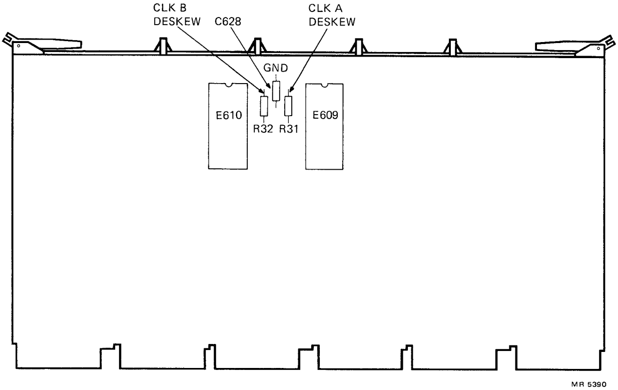


Figure 5 Oscilloscope Connection Points

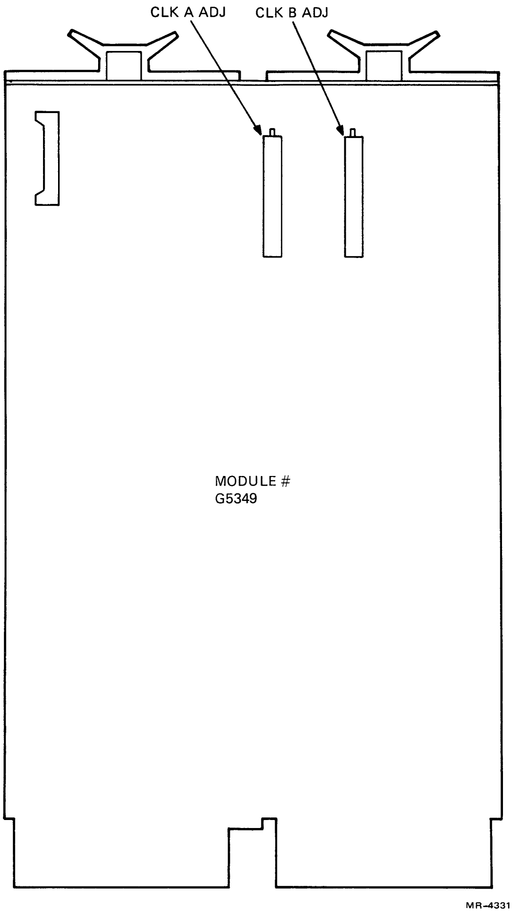


Figure 6   Deskewing Adjustments

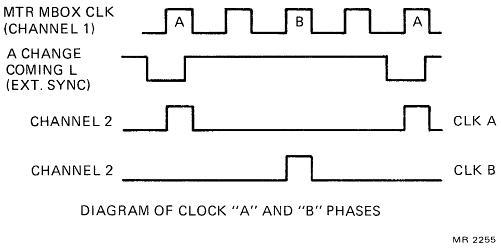


Figure 7   Deskewing Waveforms

OPERATION

Introduction

The following paragraph describes the controls and indicators on the recorder console panel. Refer to Figure 8.

Controls and Indicators

1. LOCKOUT switch - The LOCKOUT switch is a 2-position toggle. In the OFF position, the S/X Bus Recorder functions are controlled by the software and the START PULSE and STOP switches. In the ON position, the console panel functions are enabled for manual intervention, and the software is preempted.

2. STOP ON ERROR switch - The STOP ON ERROR switch is a 2-position toggle. In the OFF position, the S/X Bus Recorder continuously samples the S/X bus, and the REC decimal point remains lit. In the ON position, any error will cause the recorder to stop recording, and the REC decimal point goes out.

3. STOP switch - The STOP switch is a 2-position toggle. In the ON position, recording is terminated by clearing the record flip-flops at the next START "A" or START "B" from the S/X bus. In the OFF position, recording commences at the next START "A" or START "B" from the S/X bus, or START pulse from the recorder.

4. START PULSE - The START PULSE is a pushbutton switch. When the STOP switch is ON, depressing the START PULSE switch only clears the record flip-flop. When the STOP switch is OFF, depressing the START PULSE switch clears the recorder and begins recording on the next START "A" or START "B".

5. NXT/LD and FUNC SELECT switches - The NXT/LD switch is a pushbutton switch that is functional only when the recorder is not recording. The function of this switch is determined by FUNC SELECT. When FUNC SELECT is set to DEC, pressing the NXT/LD switch decrements the RAM ADDRESS by one. If the FUNC SELECT is set to INC, pressing the NXT/LD switch increments the RAM ADDRESS by one. When FUNC SELECT is set to LD MADR, pressing the NXT/LD switch loads the memory address into bits 13-35 of the DRR/MEMADR register.

   If the FUNC SELECT is set to LD RADR, pressing the NXT/LD switch loads the RAM register with the address selected by the REG SEL/RAM ADR thumbwheel switches.

6. REG SEL/RAM ADR thumbwheel switches - The low-order (right-hand) thumbwheel switch is the only switch to control the LED display (except for lamp test). Both thumbwheels are used to select a RAM address from 0-37. Address 40 is used for lamp test of LED D7. Address 42 and the lamp test option in the S/X Bus Recorder diagnostic program test all other LEDs (D1-D6 and D8).

7. MEM ADR indicator - When lit, the MEM ADR decimal point indicates that the DR register is loaded with memory address data. If this decimal point is not lit, the DR contains normal data.

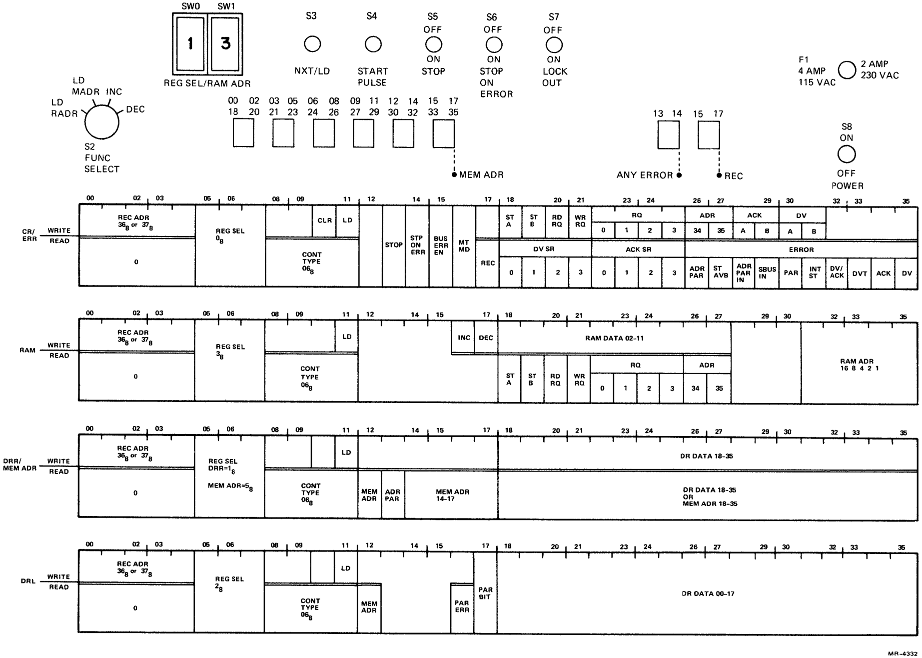


Figure 8 Console Panel Controls and Indicators

Register Descriptions

Control/Error Register (CR/ERR)

Bits 00-04 (REC ADR) - The S/X Bus Recorder is assigned a 5-bit discrete address (36 or 37 octal) for device selection when the SBus diag line is asserted. These bits are returned as zeros on the read portion of the SBus diag cycle. (Write-only bits)

Bits 05-07 (REG SEL) - Use a 3-bit field (zero octal) to select the CR/ERR register for writing and reading when the SBus diag line is asserted. (Write and read bits)

Bits 08-11 (CONT TYPE) - This is a hardwired controller type (06 octal) bit. If a recorder is connected to the SBus or XBus, these bits will be asserted for the read portion of the SBus diag cycle to indicate recorder present. The recorder's address (bits 00-04) must accompany the SBus diag cycle issuing a standard SBus diagnostic function zero to the recorder, which will read the recorder's controller type, and the CR/ERR register's read bits. (Read-only bits)

Bit 10 (CLR) - If the bit is a 1 and the console LOCKOUT switch is in the OFF position, clear and initialize the recorder's logic. (Write-only bit)

Bit 11 (LD) - If the bit is a 1 and the LOCKOUT switch is in the OFF position, allow the register's writable bits (18-35) to write and read all register bits. If the bit is a 0 and the LOCKOUT switch is in the OFF position, read all register bits. When the bit is a 1 or 0 and the LOCKOUT switch is in the ON position, no function is performed. (Write and read bits)

Bit 12 - Not used.

Bit 13 (STOP) - When this bit is a 1 and the console LOCKOUT switch is set to OFF, the REC flip-flop is cleared, the MAN STP flip-flop is set, its status is read, and the recording of bus signals is inhibited. If this bit is a 0 and the CLR bit is asserted, the MAN STP flip-flop is cleared. If the STOP and CLR bits are asserted simultaneously, the MAN STP flip-flop remains set. (Write and read bit)

Bit 14 (STP ON ERR) - If the bit is a 1 and the console LOCKOUT switch is in the OFF position, set the STP on ERR flip-flop and read its status. If the bit is a 0 and the CLR bit is asserted, clear the STP on ERR flip-flop. Simultaneous assertion of the STP on ERR and CLR bits leaves the STP on ERR flip-flop set. (Write and read bit)

Bit 15 (BUS ERR EN) - When the bit is a 1 and the console LOCKOUT switch is in the OFF position, set the BUS ERR EN flip-flop. This allows the recorder to assert SBUS ERR when an ERR is detected. If the bit is a 0 and the stop or CLR bit is asserted, clear the BUS ERR EN flip-flop. Simultaneous assertion of the BUS ERR EN, STOP, or CLR bits leaves the BUS ERR EN flip-flop set. (Write and read bit)

Bit 16 (MT MD) - When the bit is a 1 and the console LOCKOUT switch is in the OFF position, set the maintenance mode flip-flop and read its status. The MT MD flip-flop provides data paths from the SBus data lines to the selected register or to the RAM when loading data with the SBus diag line asserted. If the bit is a 0 and the CLR bit is asserted, clear the MT MD flop. Simultaneous assertion of the MT MD and CLR bits leaves the MT MD flip-flop set. (Write and read bit)

Bit 17 (REC) - When the bit is a 1, the REC (record) flip-flop is set. (Read-only bit)

NOTE

Writable bits 18-31: The MT MD flop must be set and the LD bit asserted for writable bits 18-31 of the CR/ERR register (refer to bits 11 and 16).

Bit 18 (ST A) - If the bit is a 1, load the required number of ACKNs and DVs per bits 20-25 into the ACK and DV shift registers. Also load the DV timeout counters, check for an INT ST ERR, and enable the ACK and DV shift registers to be shifted on the next and subsequent SBus diagnostics per bits 28-31.

Bit 19 (ST B) - When the bit is a 1, load the required number of ACKNs and DVs per bits 20-25 into the ACK and DV shift registers. Also load the DV timeout counters, check for an INT ST ERR, and enable the ACK and DV shift registers to be shifted on the next and subsequent SBus diagnostics per bits 28-31. When ST B is asserted, readable bits 18-35 are not guaranteed valid on the same SBus diagnostic cycle.

Bits 20-25 (RD RQ and WR) - RQ in conjunction with RQ 00/01/02/03 determine the number of ACKNS and data valids to be loaded when ST A or ST B is asserted.

Bits 26-27 - ADR 34 and ADR 35 in conjunction with RQ 00/01/02/03 are checked for an INT ST ERR when the ST A or ST B is asserted.

Bits 28-29 - ACK A and ACK B produces one shift of the ACK SR for either ACK A or ACK B. (Reference bit 18 - ST A and bit 19 - ST B)

Bits 30-31 - DV A and DV B produces one shift of the DV SR for either DV A or DV B. (Reference bit 18 - ST A and bit 19 - ST B)

Bits 32-35 - Not used.

NOTE

Readable bits 18-35: These bits are read during the read part of the SBus diagnostic cycle. (Refer to bit 11.)

Bits 18-21 (DV SR) - These bits reflect the status of the data valid shift register.

Bits 22-25 (ACK SR) - These bits reflect the states of the ACK shift register.

Bit 26 (ADR PAR ERR) - If the bit is a 1, the recorder has detected even parity in a memory address asserted on the SBus.

Bit 27 (ST A/B ERR) - When the bit is a 1, SBUS START "A" and START "B", or START "A", or START "B" and SBUS DIAG, were detected simultaneously.

Bit 28 (ADR PAR IN ERR) - If the bit is a 1, the SBUS ADR PAR ERR line was asserted.

Bit 29 (SBUS in ERR) - When the bit is a 1, the SBUS ERR line was asserted.

NOTE

Memory controllers (MF20) that generate SBUS ERR for a nonfatal error condition will stop the recorder.

Bit 30 (PAR ERR) - When the bit is a 1, even parity was detected in a SBus data word during the cycle. If MT MD is set, ACKN A or B, or DV A or B, does not clock the DR.

Bit 31 (INT ST ERR) - If the bit is a 1, an initial start error was detected and one of the following conditions occurred.

1. RQ OX = ADR 34/35 - The SBus request associated with SBUS ADR 34/35 was not asserted.

2. RMW = 1 WD RQ - More than one word was requested on the SBus for a read-modify-write cycle.

3. RD/WR RQ = 0 - No SBUS RD RQ or WR RQ was asserted for a valid SBus cycle.

Bit 32 (DV/ACK ERR) - If the bit is a 1, a data valid occurred before an ACKN.

Bit 33 (DV Timeout) - When the bit is a 1, a missing data valid has been detected. Timeout is produced when the memories' read-access time (ACKN to data valid) is exceeded.

Bit 34 (ACK ERR) - If the bit is a 1, ACK A and ACK B were detected simultaneously.

Bit 35 (DV ERR) - If the bit is a 1, data valid A and data valid B were detected simultaneously, or a data valid was detected when RD RQ was not asserted.

Data Register Left (DRL)

Bits 00-04 (REC ADR) - The S/X Bus Recorder is assigned a 5-bit discrete address (36 or 37 octal) for device selection when the SBus diag line is asserted. These bits are returned as zeros on the read portion of the SBus diag cycle. (Write-only bits)

Bits 05-07 (REG SEL) - Use a 3-bit field (02 octal) to select the DRL register for writing and reading when the SBus diag line is asserted. (Write and read bits)

Bits 08-11 (CONT TYPE) - This is a hardwired controller type (06 octal) bit. If a recorder is connected to the SBus or XBus, these bits will be asserted for the read portion of the SBus diag cycle to indicate recorder present. The recorder's address (bits 00-04) must accompany the SBus diag cycle issuing a standard SBus diagnostic function zero to the recorder, which will read the recorder's controller type, and the CR/ERR register's read bits. (Read-only bits)

Bit 11 (LD) - If the bit is a 1 and the LOCKOUT switch is in the OFF position, allow the register's writable bits (18-35) to write and read all register bits. Also set the MT MD flip-flop and read its status. This flip-flop provides data paths from the SBus data lines to a selected register or RAM when loading data with the SBus diag line asserted. If the bit is a 0 and the LOCKOUT switch is in the OFF position, read all register bits. If the bit is a 0 and the CLR bit is asserted, clear the MT MD flip-flop. Simultaneous assertion of the MT MD flip-flop and the CLR bit leaves the MT MD flip-flop set. When the bit is a 1 or a 0 and the LOCKOUT switch is in the ON position, no function is performed. (Write and read bits)

Bit 12 (MEM ADR) - If the bit is a 1, DR bits 14-35 contain a mem adr; DR bits 00-11 and 13 should be disregarded. This bit is set when the mem adr is read via the SBus diag or console and resets when other than a mem adr is clocked into the DR. (Read-only bit)

Bits 13-5 - Not used.

Bit 15 (PAR ERR) - If the bit is a 1, even parity was detected in the data register. If MT MD is set, loading the DRL will check the parity of the DR. (Read-only bit)

NOTE
The terms LOAD and NOT LOAD refer to the diagnostic and are used for diagnostic testing only.

Bit 17 (PAR BIT) - If the bit is a 1 and LOAD, set the DR parity flip-flop and read its status. If NOT LOAD, read the status of the DR parity flip-flop. (Write and read bit)

Bits 18-35 (DATA) - If LOAD, write bits 18-35 (SBus data) into the DRL (DR00-17) and read its contents. If NOT LOAD, read the contents of the DRL.

Data Register Right (DRR)

Bits 00-04 (REC ADR) - The S/X Bus Recorder is assigned a 5-bit discrete address (36 or 37 octal) for device selection when the SBus diag line is asserted. These bits are returned as zeros on the read portion of the SBus diag cycle. (Write-only bits)

Bits 05-07 (REG SEL) - This 3-bit field is used to select the DRR (01 octal) or the contents of the latches (05 octal). When the mem adr select (05 octal) is asserted, mem adr latches bits 14-35 are clocked into DR bits 14-35 on the write part of the SBus diag cycle and the mem adr flag (bit 13) is set.

Bits 08-11 (CONT TYPE) - This is a hardwired controller type (06 octal) bit. If a recorder is connected to the SBus or XBus, these bits will be asserted for the read portion of the SBus diag cycle to indicate recorder present. The recorder's address (bits 00-04) must accompany the SBus diag cycle issuing a standard SBus diagnostic function zero to the recorder, which will read the recorder's controller type, and the CR/ERR register's read bits. (Read-only bits)

Bit 11 (LD) - If the bit is a 1 and the LOCKOUT switch is in the OFF position, allow the register's writable bits (18-35) to write and read all register bits. Also set the MT MD flip-flop and read its status. This flip-flop provides data paths from the SBus data lines to a selected register or RAM when loading data with the SBus diag line asserted. If the bit is a 0 and the LOCKOUT switch is in the OFF position, read all register bits. If the bit is a 0 and the CLR bit is asserted, clear the MT MD flip-flop. Simultaneous assertion of the MT MD flip-flop and the CLR bit leaves the MT MD flip-flop set. When the bit is a 1 or a 0 and the LOCKOUT switch is in the ON position, no function is performed. (Write and read bits)

Bit 12 (MEM ADR) - If the bit is a 1, DR bits 14-35 contain a mem adr; DR bits 00-11 and 13 should be disregarded. This bit is set when the mem adr is read via the SBus diag or console and resets when other than a mem adr is clocked into the DR. (Read-only bit)

Bit 13 (ADR PAR) - If the bit is a 1, the address parity bit was asserted on the SBus/XBus.

Bits 14-17 (MEM ADR) - If mem adr (bit 13) is a 1, DR bits 14-17 reflect a mem adr. (Read-only bits)

Bits 18-35 (DATA) - If LOAD and DRR select, write bits 18-35 (SBus data) into DRR (DR18-35) and read its contents. If NOT LOAD and DRR select, read the contents of DRR (bits 18-35). If NOT LOAD and mem adr select, read the contents of the mem adr latches. The mem adr latches cannot be loaded via the SBus diag.

RAM Data Register - On each SBus transaction certain data is checked and other data is captured. The 10 bits that are captured are stored in a 32_{10} (37_8) word RAM. This data consists of START "A", START "B", qualifiers for read and write memory, four requests (0,1,2,3), and address bits 34 and 35 of the physical address. The RAM data is in bits 18-27 of the RAM register, as shown in Figure 8. During each RAM data cycle, either START "A" or START "B" must be asserted. RAM data should never be stored without START "A" or START "B" active. The bit definitions are as follows.

Bit 18 - START "A"
Bit 19 - START "B"
Bit 20 - Read Request
Bit 21 - Write Request
Bit 22 - Request 0
Bit 23 - Request 1
Bit 24 - Request 2
Bit 25 - Request 3
Bit 26 - Address Bit 34
Bit 27 - Address Bit 35

S/X Bus Recorder Operation and Dumping

S/X Bus Recorder Operation - After the deskewing of clock signals to the memories and S/X bus recorder, the LOCKOUT switch should be placed in the OFF position. Diagnostic DGSBA should be run in order to check out the functionality of the recorder. DGSBA is an 11-based diagnostic that can read and write registers contained within the recorder. After successful completion of DGSBA, the "B" command string should be run with the LOCKOUT switch in the ON position. All diagnostics should be run without errors, except for the following.

DHKBA	Test No. 19, Subtest 1, PC 31064
DHKBB	Test No. 1, Subtest 1, PC 22554
DGKBA	Test No. 19
DGKBB	Test No. 1

If no errors are found (except for those noted) the recorder is ready for system operation. To reset the recorder, proceed as follows.

1. Place the STOP switch in OFF position.

2. Place the LOCKOUT in ON position.

3. Enable the recorder to stop on error by placing the STOP ON ERROR toggle switch to the ON position.

4. Press the START PULSE button.

5. The recorder is now initialized. All errors are cleared and the ANY ERROR decimal point should now be OFF. The ACK and DV shift counters are initialized and recording will begin on the next START "A" or START "B" signal on the SBUS. Once the START "A" or START "B" signal is received by the recorder, the RECORD decimal point will light to indicate that the recorder is storing SBUS activity.

Steps 4 and 5 clear and arm the recorder. If the recorder detects an error, as described in the CR/ERR register bits 26-35 (refer to the Control/Error Register (CR/ERR) section), and the STOP ON ERROR switch is ON, recording will cease. Several methods are available in order to dump the contents of the recorder:

1. Manual (via switches on console).

2. Through the diagnostic DGSBA.

3. Through CCL file for KLDCP (another will be available with RSX-20F).

4. Through SBus DIAGs.

Dumping the Recorder via KLDCP or RSX20F - For RSX20F you must have Version 14. A new command has been added to Version 14. This command is the TAKE command, which will read and execute a command file. The available .CMD files for the S/X Bus recorder are

```
SB0.CMD   ;For SBus recorder addressed as #36
SB1.CMD   ;For SBus recorder addressed as #37
```

If you are using KLDCP, the I command for .CCL processing should be used. The .CCL files available are

```
SB0.CCL   ;SBus recorder addressed as #36
SB1.CCL   ;SBus recorder addressed as #37
```

Dumping the Recorder via DGSBA - To dump the recorder via DGSBA, perform the following steps.

1. To KLDCP type P DGSBA.

2. Ensure that the recorder's LOCKOUT switch is in the OFF position.

3. When KLDCP has returned with the >. prompt, type:

   SED/D

4. When the diagnostic starts, you will be required to select which SBus recorder you want. The diagnostic will print out the following information.

```
SBUS/XBUS RECORDER DIAGNOSTIC CONSOLE PACKAGE BEGINS

SBUS/XBUS RECORDER AVAILABLE:
RECORDER #0     ADDRESS 36

SELECT SBUS/XBUS RECORDER (0,1 OR B FOR BOTH) - 0

SBUS/XBUS RECORDER SELECTED:
REC #0     REC ADR 36

TYPE HLP<CR> IF YOU WANT INSTRUCTION
```

5. Once you reach SBA>, the command you give is RAL (READ ALL).

   This will give you a readout of all the registers.

```
SBA>RAL

CR REG = 000610 356140
DRL REG = 004602 351400
DRR REG = 002620 010316
MEM ADR REG = 012660 137144

RAM REG = 00660 336001
RAM REG = 00660 276000
RAM REG = 00660 336037
RAM REG = 00660 276036
RAM REG = 00660 276035
RAM REG = 00660 476034
RAM REG = 00660 276033
RAM REG = 00660 476032
RAM REG = 00660 276031
RAM REG = 00660 476030
RAM REG = 00660 276027
RAM REG = 00660 476026
RAM REG = 00660 276025
RAM REG = 00660 476024
RAM REG = 00660 276023
RAM REG = 00660 476022
RAM REG = 00660 276021
RAM REG = 00660 476020
RAM REG = 00660 276017
RAM REG = 00660 476016
RAM REG = 00660 276015
RAM REG = 00660 476014
RAM REG = 00660 276013
RAM REG = 00660 476012
RAM REG = 00660 276011
RAM REG = 00660 476010
RAM REG = 00660 276007
RAM REG = 00660 476006
RAM REG = 00660 276005
RAM REG = 00660 476004
RAM REG = 00660 276003
RAM REG = 00660 476002
SBA>EXT
CMD:
```

Manual Dumping of the S/X Bus Recorder - To manually dump the S/X Bus recorder, proceed as follows.

1. Adjust the following button and two switches.

   a. Put the LOCKOUT switch to the ON position.
   b. Put the STOP switch to the ON position.
   c. Press the ST PLS button, which will ensure that REC is reset.

   CAUTION
   Depressing the ST/PLS button with the STOP switch in the OFF position will clear the recorder.

2. Put a 00 in the REG SEL/RAMADR thumbwheel switch. This will gate out the CR/ERR register information, which is contained in bits 18-35, into the LEDs. Record this information.

3. Put a 01 in the REG SEL/RAMADR thumbwheel switch. This will gate out the contents of the Data Reg Right (DRR).

4. Put a 02 in the REG SEL/RAMADR thumbwheel switch. This will gate out the contents of the Data Reg Left (DRL).

5. Turn the rotary switch to the MEMADR position.

   CAUTION
   Before performing the next step make sure you have copied down the DRR, as it will be overwritten with MEMADR.

6. Press the NXT/LD switch. The MEMADR decimal point should light, indicating that memory address data is now loaded in the DRR.

7. Put a 05 in the REG SEL/RAMADR thumbwheel switch. This will gate out the contents of the MEMADR.

8. Put a 03 in the REG SEL/RAMADR thumbwheel switch.

9. Turn the function select rotary switch to DEC position and press the ST/PLS switch. The LEDs now display information for the SBus cycle in which the error(s) was (were) detected.

Operation with MOS Memory and TGHA - In order for the S/X Bus Recorder to run without detecting single-bit recoverable errors, TGHA must not be run.

Under TOPS-10, the OPR.ATO file must be edited so that TGHA will not run. This can be done by inserting an exclamation point before logging, defining, and running TGHA.

```
Example:  !  :SLOG
          !  :DEF  TGHA=
          !  TGHA-  R  TGHA
```

In the case of TOPS-20, rename the file <SYSTEM> TGHA.EX1 rather than <SYSTEM> TGHA.EXE.

NOTE

Once you are done using the S/X Bus Recorder, remove the exclamation point from OPR.ATO for TOPS-10 or rename the file <SYSTEM> TGHA.EXE (instead of <SYSTEM> TGHA.EX1).

Error Interpretation
The S/X Bus Recorder has the capability of detecting write parity errors on the other S/X Bus. This is possible because every time a data valid is received in the recorder, parity is checked. You can determine that the error occurred on the other bus by examining the CR/ERR register. If the CR/ERR register contains a 046040 in the right half, which indicates DV #3, ACK #2, ACK #3, and parity error, a parity error occurred on the write portion of a read-pause-write on the other S/X bus.

The following are five examples of errors that can occur on the S/X Bus, and how to interpret them using the recorder.

Internal Memory Failures - The following five examples of failures are internal memory failures and XBus errors.

Data Parity Error #1 - This error is an example of a data parity error on word #3, with the transfer starting on word zeros boundary.

The following program was put in location 100 and 101 of the internal memory:

```
100/ MOVE 10, 1000
101/ JRST 4,   101
```

Paging was set up so that page one (right half of location 600 of Exec Base register) was pointing at physical page 1000. The EBR was at page zero. The program was run one time in order to move the program to cache. Next, cache look and load was turned off (by executing a CONO PAG,20000). Bad parity was then put into loc 1003 by executing the following instructions.

```
CONO PI,200000    (700600 200000)  ;write even parity
MOVEM 1003  (202000 1003) ;store away AC0 in location 1,,001003
;with bad parity
CONO PI,0 700600 0 ;turn off write even parity
```

Cache look and load was then enabled with a CONO PAG,620000 (701200 620000). Next, the physical page 1001 was invalidated in cache to allow a 4-word memory request (the program at loc 100 and 101 was in cache and was valid).

When the machine was started at location 100 it executed a MOVE to AC0 location 1000. Since cache was enabled and those words were not in cache at the time, a 4-word request was generated on the SBus starting at location 1000. In analyzing the contents of the CR/ERR register note that all the data valid and ACK shift registers are cleared. The recorder disables the clearing of data valid and ACK shift registers when an error is detected. The fact that all the shift registers are cleared indicates that the error happened on word #3. By looking at the data contained in the memory address register (MEM ADR), determining the amount of requests that were sent and finding out which data valid was cleared out last, one can determine the failing word. The physical location of the failing word is 1,,001003.

The program was started at location 100. The following is the console printout and the execution of the CCL file called SB0.CCL This is at KLDCP level. An * indicates a comment inserted to explain data.

```
>.
KL10 HALTED PC /000100 VMA/000101
>.I SB0
;S/X BUS RECORDER DUMPER
;THIS WILL WORK IF RECORDER IS ADDRESSED AS 36
;
;******  THE LOCKOUT SWITCH MUST BE IN THE OFF POSITION***
;
;FIRST STOP THE RECORDER
DM10:740020 0
EX700500 10
;NEXT GET CR/ERR REGISTER
EM11
000011/000630 000040  *Bit 30 set (parity error).  Nothing was
                      *left in data valid and ACK shift registers.
;
;NOW GET DATA REG LEFT (DRL)
DM10:744000 0
EX700500 10
EM11
000011/004603 275500  *Bit 16 indicates parity error.
;
;NOW GET DATA REG RIGHT (DRR)
DM10:742000 0
EX700500 10
EM11
000011/002600 000400
;
;NOW GET MEMORY ADDRESS REGISTER (MEMADR)
DM10:752000 0
EX700500 0
EM11
000011/012641 001000  *Indicates address as being 1,,001000
;
;NOW GET ALL THE RAMS
DM10:746001 0
EX700500 10           *This command is executed twice in order
EX700500 10,EM11      *to get into the correct RAM address.
000011/006600 336007  *This is the RAM that was valid at time of
                      *error.  It shows that it was started on
                      *START "B", was a READ REQUEST, and was a
                      *4-word request.
EX700500 10,EM11
000011/006600 336006
EX700500 10,EM11
000011/006600 243405
EX700500 10,EM11
000011/006600 336004
EX700500 10,EM11
000011/006600 503403
EX700500 10,EM11
000011/006600 243402
EX700500 10,EM11
000011/006600 503401
EX700500 10,EM11
000011/006600 536000
EX700500 10,EM11
000011/006600 520037
EX700500 10,EM11
000011/006600 520036
EX700500 10,EM11
000011/006600 520035
EX700500 10,EM11
000011/006600 520034
EX700500 10,EM11
```

```
000011/006600 520033
EX700500 10,EM11
000011/006600 520032
EX700500 10,EM11
000011/006600 520031
EX700500 10,EM11
000011/006600 520030
EX700500 10,EM11
000011/006600 520027
EX700500 10,EM11
000011/006600 520026
EX700500 10,EM11
000011/006600 520025
EX700500 10,EM11
000011/006600 520024
EX700500 10,EM11
000011/006600 260023
EX700500 10,EM11
000011/006600 320022
EX700500 10,EM11
000011/006600 336021
EX700500 10,EM11
000011/006600 536020
EX700500 10,EM11
000011/006600 443417
EX700500 10,EM11
000011/006600 336016
EX700500 10,EM11
000011/006600 310415
EX700500 10,EM11
000011/006600 320014
EX700500 10,EM11
000011/006600 450413
EX700500 10,EM11
000011/006600 305012
EX700500 10,EM11
000011/006600 510411
EX700500 10,EM11
000011/006600 320010
;NOW RESET THE RECORDER AND SET STOP ON ERROR
DM10:740210 0
EX700500 10,EM11
000011/000610 000000
```

Data Parity Error #2 - This error is an example of a combination data parity error and data valid timeout (starting on word zero boundary).

In this example, the same program was initialized as in Data Parity Error #1 except that the parity error was put into physical location 1,,001002, or paged location 1002. In analyzing the contents of the CR/ERR register, two errors can be found: 1) a parity error, and 2) a data valid timeout. Once the data valid timers are started they will continue even after an error is detected. In this case, the real error is the parity error. Since the one data valid is still left, the recorder will inhibit clearing of any ACK or DV bits on detection of an error. The error occurred prior to receiving the data valid #3. If you look at the RAM at the time of the error, it had a 536000, which is a 4-word request beginning on word zero boundary. Since the recorder stopped on error, the data valid shift register has DV #3 left; it was a 4-word request, the error happened on word #2. The address contained in the MEM ADR register is 1,,001000. This is the initial address requested. By knowing the error happened on word two, the parity error occurred at location 1,,001002.

The printout is as follows.

```
CMD:
>.
>.I SBO
;S/X BUS RECORDER DUMPER
;THIS WILL WORK IF RECORDER IS ADDRESSED AS 36
;
;******  THE LOCKOUT SWITCH MUST BE IN THE OFF POSITION***
;
;FIRST STOP THE RECORDER
DM10:740020 0
EX700500 10
;NEXT GET CR/ERR REGISTER
EM11
```

```
000011/000630 040044  *Bit 30 = parity error,
                      *Bit 33 = data valid timeout
                      *Bit 21 = data valid #3 is not shifted out
;
;NOW GET DATA REG LEFT (DRL)
DM10:744000 0
EX700500 10
EM11
000011/004602 777777  *Bit 16 indicates parity error
                      *Bit 17 indicates parity bit
;
;NOW GET DATA REG RIGHT (DRR)
DM10:742000 0
EX700500 10
EM11
000011/002600 777777
;
;NOW GET MEMORY ADDRESS REGISTER (MEMADR)
DM10:752000 0
EX700500 0
EM11
000011/012641 001000  *Indicates address as being 1,,001000
;
;NOW GET ALL THE RAMS
DM10:746001 0
EX700500 10
EX700500 10,EM11
000011/006600 536000 *This is the RAM that was valid at time of
                     *error.  It shows that it was started on
                     *START "B", was a READ REQUEST and was a
                     *4-word request with address bit 34 and 35 =
                     *0. (This 4-word request begins on even quad
                     *word boundary, i.e., word zero.)
EX700500 10,EM11
000011/006600 520037

;
;
```

Data Parity Error #3 - This error is an example of the combination of a data parity error and a data valid timeout (starting in other than a word zero boundary).

In this example, the starting location for the 4-word request was 1,,001001. The contents of the CR/ERR register contain the same data as in Data Parity Error #2. In this example, however, the beginning of the 4-word request was not on word zero of the 4-word request, but word one of the 4-word request. Determine what was the original word requested and how many requests were asked for. After you have established this, see what is left in data valid and ACK shift registers in the CR/ERR register. In this example, data valid #3 is left. Again, as in example #2, it is the cause of the DVT (data valid timeout). The parity error is the real error on word #2. Since the quad word fetch started on word one of a 4-word fetch, the address that failed was 1,,001003.

REMEMBER: The data valid and ACK shift registers are modulo four type registers and that the data contained in them are relative to the beginning address of the transfer (RAM ADR bits 34 and 35). SBus address bits 34 and 35 are contained in RAM register bits 26 and 27.

The printout is as follows.

```
CMD:
>.
>.I SBO
;S/X BUS RECORDER DUMPER
;THIS WILL WORK IF RECORDER IS ADDRESSED AS 36
;
;******  THE LOCKOUT SWITCH MUST BE IN THE OFF POSITION***
;
;FIRST STOP THE RECORDER
DM10:740020 0
EX700500 10
;NEXT GET CR/ERR REGISTER
EM11
000011/000630 040044  *Bit 30 = parity error
                      *Bit 33 = data valid timeout
                      *Bit 21 = one data valid not shifted out
```

```
;
;NOW GET DATA REG LEFT (DRL)
DM10:744000 0
EX700500 10
EM11
000011/004603 000000  *Bit 16 indicates parity error
;
;NOW GET DATA REG RIGHT (DRR)
DM10:742000 0
EX700500 10
EM11
000011/002600 000001
;
;NOW GET MEMORY ADDRESS REGISTER (MEMADR)
DM10:752000 0
EX700500 0
EM11
000011/012641 001001  *Indicates address as being 1,,001001
;
;NOW GET ALL THE RAMS
DM10:746001 0
EX700500 10
EX700500 10,EM11
000011/006600 536400 *This is the RAM that was valid at time of
                     *error.  It shows that it was started on
                     *START "B", was a READ REQUEST, and was a
                     *4-word request with address bit 34 = 0
                     *and address bit 35 = 1.
EX700500 10,EM11
000011/006600 520037
EX700500 10,EM11
000011/000610 000000
;
```

Data Parity Error #4 - This error is similar to Data Parity Error #2, except that the word that failed was word zero of the transfer.

In this example, the parity error was put in location 1,,001000. The instruction in location 100 was a MOVE 10,1000. This would create a 4-word request (with cache enabled). At the CR/ERR register, notice that three data valids were not shifted out, and that the last error occurred on the first word of the transfer.

The address in the MEM ADR register is 1,,001000. Since it was the first word transferred that got the error, address 1,,001000 is the bad one.

The printout is as follows.

```
>.
>.I SBO
;S/X BUS RECORDER DUMPER
;THIS WILL WORK IF THE RECORDER IS ADDRESSED AS 36
;
;******  THE LOCKOUT SWITCH MUST BE IN THE OFF POSITION***
;
;FIRST STOP THE RECORDER
DM10:740020 0
EX700500 10
;NEXT GET CR/ERR REGISTER
EM11
000011/000630 340044  *Data valid 1, 2 and 3 still left parity
                      *error and data valid timeout.
;
;NOW GET DATA REG LEFT (DRL)
DM10:744000 0
EX700500 10
EM11
000011/004602 777777
;
;NOW GET DATA REG RIGHT (DRR)
DM10:742000 0
EX700500 10
EM11
000011/002600 777777
;
;NOW GET MEMORY ADDRESS REGISTER (MEM ADR)
DM10:752000 0
EX700500 10
EM11
000011/012641 001000  *Initial address latched = 1,,001000
```

```
;NOW GET ALL THE RAMS
DM10:746001 0
EX700500 10
EX700500 10,EM11
000011/006600 536000  *START "A", read request, 4-word request
                      *(RQ0,1,2,3), address 34 = 0, address 35 = 0
EX700500 10,EM11
000011/006600 520037
EX700500 10,EM11
000011/006600 520036
EX700500 10,EM11
                      *The rest of the RAMs are not displayed in
                      *this example since they are not needed.
```

Read-Pause-Write Failure (Write Portion) - On Read-Pause-Write (RPW) operations, a 1-word request is always generated. On receiving a RPW operation, the recorder will load the ACK/DV shift register with two ACKs and two data valids. This is done to identify which portion of the RPW cycle had the problem. Since the memory will respond with one data valid on the read portion of the cycle, this will clear one of the data valids that was loaded in the ACK/DV shift register. When the CPU sends the data out on the write portion, the CPU will send data valid to the memory, which will also clear the other data valid contained in the ACK/DV shift register. Therefore, if the ACK/DV shift register contained one data valid, the error happened on the READ portion. If there are no data valids left, the error occurred on the write portion.

In the following example an AOS instruction was executed after enabling write even parity. This latched an error on the write portion of the cycle.

```
>.
>.I SB0
;SB0.CCL -- CCL FILE FOR DUMPING RECORDER # 0
;LSGMEG 9-NOV-79
;THIS WILL WORK IF THE RECORDER IS ADDRESSED AS 36
;
;***** THE LOCKOUT SWITCH MUST BE IN THE OFF POSITION***
;
;FIRST STOP THE RECORDER
DM10;740020 0,EX700500 10
;NEXT GET CR/ERR REGISTER
EM11
000011/000630 002040  *This indicates a parity error and one ACK
                      *left to be shifted.
                      *This ACK is normal case on RPW.
;NOW GET DATA REG LEFT (DRL)
DM10:744000 0,EX700500 10,EM11
000011/004603 000000
;NOW GET DATA REG RIGHT (DRR)
DM10:742000 0,EX700500 10,EM11
000011/002620 000001
;NOW GET MEMORY ADDRESS REGISTER (MEM ADR)
DM10:752000 0,EX700500 10,EM11
000011/012660 000100
;
;NOW GET ALL THE RAMS
DM10:746001 0,EX700500 10
EX700500 10,EM11
000011/006600 560000  *This is the RAM at the time of failure.
                      *It indicates the cycle started on
                      *START "A", had RD and WR asserted, and
                      *request zero.
```

Since there were no data valids left, this happened on the write portion.

Read-Pause-Write Failure (Read Portion) - In this example the AOS instruction was executed again. Since the previous example had written bad parity into core, that same location now failed on the read portion of the RPW cycle as follows.

```
>.
>.I SB0
;SB0.CCL -- CCL FILE FOR DUMPING RECORDER # 0
;LSGMEG 9-NOV-79
;THIS WILL WORK IF RECORDER IS ADDRESSED AS 36
;
***** THE LOCKOUT SWITCH ON THE RECORDER MUST BE IN THE OFF
POSITION ***
;
;FIRST STOP THE RECORDER
DM10;740020 0,EX700500 10
;NEXT GET THE CR/ERR REGISTER
EM11
000011/000630 042040  *Bit 21 indicates data valid #3 left.
                      *Bit 25 indicates ACK #3 left (normal on
                      RPW).
                      *Bit 30 indicates parity error.

;NOW GET THE DATA REG LEFT (DRL)
DM10:744000 0,EX700500 10,EM11
000011/004603 000000
;NOW GET THE DATA REG RIGHT (DRR)
DM10:742000 0,EX700500 10,EM11
000011/002620 000001
;NOW GET THE MEMORY ADDRESS REGISTER (MEM ADR)
DM10:752000 0,EX700500 10,EM11
000011/012660 000100
;
;NOW GET ALL THE RAMS
DM10:746001 0,EX700500 10
EX700500 10,EM11
000011/006600 560000  *This indicates the cycle started with a
                      *START "B", and a one word request with
                      *read and write asserted.
```

There is still one data valid left to be shifted out. This means that the error latched on the first half of the cycle (read).

DMA Errors - (This section will be supplied at a later date.)

NOTE

DMA errors involving an incomplete cycle may point to an incorrect address. Read or write errors point to the correct address.

Loading and Running Diagnostics

Program Abstract - DGSBA is a diagnostic for the S/X Bus Recorder only. It is not a diagnostic for the KL10 memory system. However, DGSBA should be used to verify that the recorder is properly installed.

DGSBA is only a functional diagnostic. It was not designed to do any gate- or board-level callout. The error messages indicate the symptoms of an error as a result of what type of operation occurs.

Requirements

Preliminary Software - DGKAA, DGKAB, and DGKBA should be run first. Next, the memory should be configured; then DGSBA should be run. If the memory could not be configured, DGSBA should be run anyway because it may be the recorder that is polluting the memory bus.

Operating Procedure

Loading Procedure - DGSBA is supplied as an ".All" file and can be loaded from DECtape, flexible diskette, or a front-end RP04 or RP06 pack by selecting the device (see KLDCP operating procedures, EK-OKL10-MG) and typing: "P DGSBA.All".

Operation Modes and Switches - If DGSBA is started with all switches 0, and no / switches, it will run all tests that do not require operator intervention. Any and all recorders will be tested.

If the memory is configured, the last set of tests will be run. If the memory is not configured, the last set of tests will be skipped, and a message stating that a test has been skipped due to no memory will be printed.

SED/H The /H switch will type out the help test. The printout of the test is as follows.

SBD	DO SBUS DIAG
LCE	LOAD CONTROL/ERROR REGISTER
LDL	LOAD DATA REGISTER LEFT
LDR	LOAD DATA REGISTER RIGHT
LRM	LOAD RANDOM-ACCESS MEMORY
RCE	READ CONTROL/ERROR REGISTER
RDL	READ DATA REGISTER LEFT
RDR	READ DATA REGISTER RIGHT
RRM	READ RANDOM-ACCESS MEMORY
RAL	READ ALL REGISTERS AND RANDOM-ACCESS MEMORY
IRM	INCREMENT RAM ADDRESS
DRM	DECREMENT RAM ADDRESS
EXT	BACK TO KLDCP
HLP	PRINT HELP MESSAGE
SWI	PRINT CURRENT STATE OF SWITCH
IDT	IDENTIFY SBUS/XBUS RECORDER NUMBER
CFG	CHANGE SELECTION OF SBUS/XBUS RECORDER
HLT	HALT PROGRAM OPERATION
LPT	LAMP TEST

SED/D The /D switch will put DGSBA into diagnostic debug mode. This mode allows the operator to examine and modify the registers in the selected recorder.

SED/S The /S switch will ask the operator which of the recorders that are on-line are to be tested. This switch forces the operator to select recorders even if the OPRSEL switch is not up.

Test Control Switches

Name	Switch	Function
ABORT =	100000	;Abort at program pass completion
RSTART =	40000	;Restart test
TOTALS =	20000	;Print test totals
NOPNT =	10000	;Inhibit all printout (except forced)
PNTLPT =	4000	;Print on line printer
DING =	2000	;Ring TTY bell on error
LOOPER =	1000	;Loop on error
ERSTOP =	400	;Halt on error
PALERS =	200	;Print all errors
RELIAB =	100	;Reliability run mode
TXTINH =	40	;Test inhibit
INHPAG =	20	;Inhibit paging
MODDVC =	10	;Modify device code
INHCSH =	4	;Inhibit cache
OPRSEL =	2	;Operator selection
CHAIN =	1	;Chain control switch

ARM-10LS

Table of Contents

INSTALLATION

Off-Line Operation
To energize and check out the ARM-10LS, perform the following procedures.

Power Connection - The power connection is made at the EPO assembly. Receptacles located at the rear of this assembly are illustrated in Figure 1. All receptacles are reached via the accessible side panel. The following paragraphs describe the power connection procedures.

NOTE
Refer to the OPERATION section for a complete description of all controls and indicators.

Before power is applied to the memory, ensure that all connectors are secured.

CAUTION
Applying incorrect power will severely damage the equipment. Use 208 V, 3-phase, 60 Hz, or 220/240 V, 1-phase, 50 Hz for standard order units.

Before plugging the power cable into the site power outlet, check the power at the outlet with an ac voltmeter to verify correct voltage levels and proper receptacle wiring. Once site power is verified, plug the male of the cabinet power cable into the site power outlet.

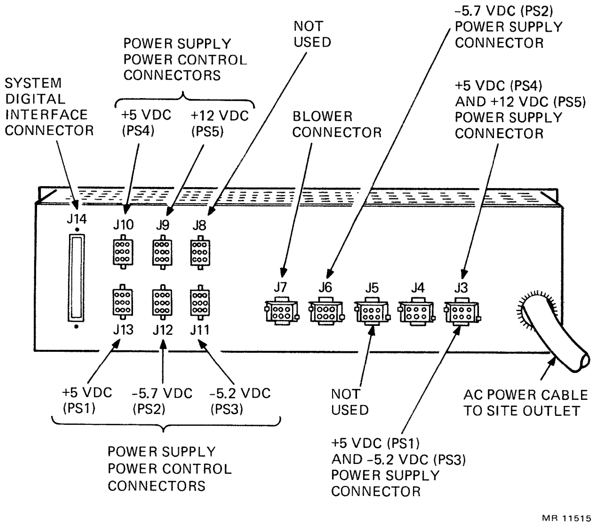


Figure 1   Rear View of EPO Assembly

The information on the ARM-10LS is included here by permission of Ampex Corporation.

Initial Power-On Procedure - Perform the following procedure to apply power to the memory. Controls and indicators are illustrated in the OPERATION section.

1. At the rear of the blower module, check that the circuit breaker button is pushed in.

2. On EPO front panel:

   a. Set MAIN POWER circuit breaker to OFF.

   b. Loosen the two inner thumbscrews on the panel and remove the EPO subpanel assembly. Ensure that the five DIP switches at the top of the EPO PWBA are set correctly, as shown in Figure 2. Replace the assembly.

   c. Set REMOTE/LOCAL switch to LOCAL.

   d. Set all VOLTAGE MARGIN switches to the center position.

   e. Set MAIN POWER circuit breaker to ON.

   f. Press POWER ON pushbutton and hold for three seconds. This step ensures proper power-up sequence.

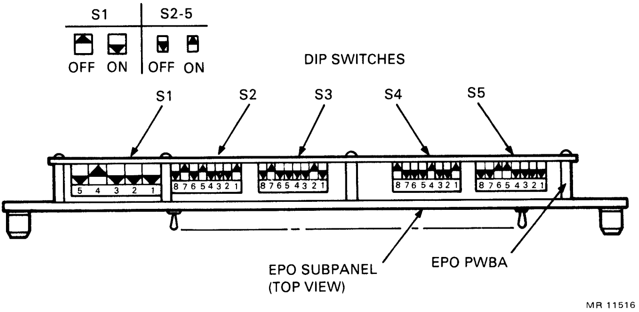


Figure 2 EPO PWBA Switch Settings

3. Check that the fans are operating.

NOTE
Memory will not turn on unless fans are operating.

4. Voltage Check: Verify/adjust power supply outputs to normal voltage defined in Table 1. Figure 3 shows the voltage test points, and Figure 4 shows the power supply voltage adjustment locations.

Table 1 DC Voltage Parameters

Power Supply	Nominal DC Output	Tolerance Low	High
PS1	+5 Vdc	+4.95 Vdc	+5.05 Vdc
PS2	-5.7 Vdc	-5.75 Vdc	-5.65 Vdc
PS3	-5.2 Vdc	-5.25 Vdc	-5.15 Vdc
PS4	+5 Vdc	+4.95 Vdc	+5.05 Vdc
PS5	+12 Vdc	+11.95 Vdc	+12.05 Vdc

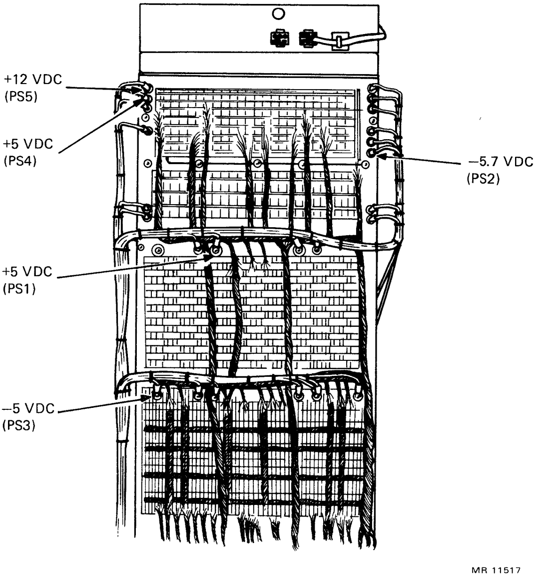


Figure 3 ARM-10LS Voltage Check Points

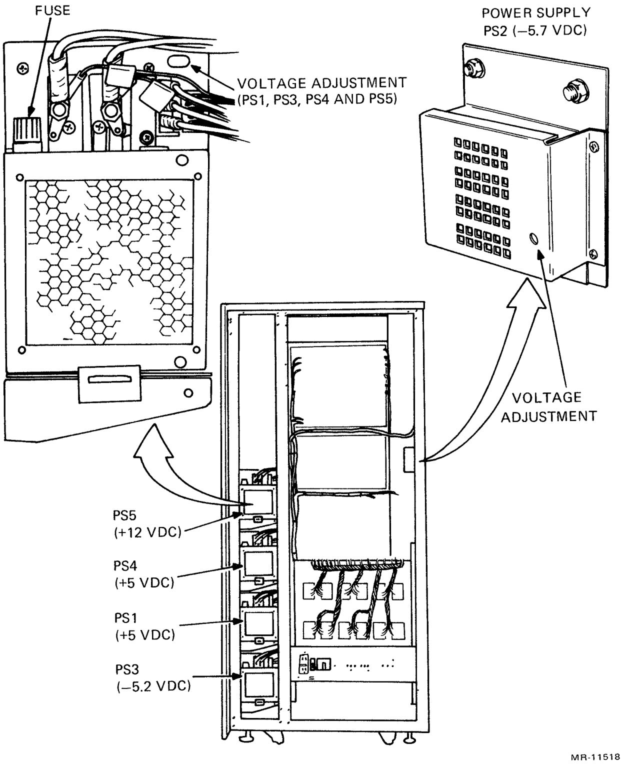


Figure 4   Power Supply Voltage Adjustment Locations

5. VTH Adjustments: The following adjustments must be made for all Transceiver PWBAs. (Card locations: CJ102-CJ109, CJ111-CJ118, CJ120-CJ127.)

   a. Set the digital multimeter to the lowest practical scale and measure dc voltage at test point lugs on the Transceiver PWBA. Refer to Figure 5 for location.

   b. Adjust potentiometer R4 until the voltage read on the digital multimeter is nominally -1.65 ± 0.01 Vdc.

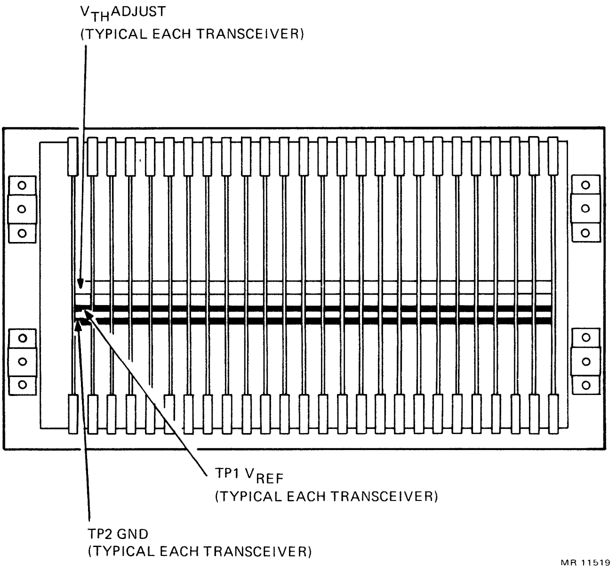


Figure 5 VTH Test Points

General Component Information
Figure 6 shows the placement of all PWBA boards in the rear of the ARM-10LS. (A similar chart appears on the inside of the unit's rear door.)

Tables 2, 3, and 4 cross reference the Ampex PWBA board part numbers with Digital's part numbers for the boards. Because Digital now supports the ARM-10 memory, you can order parts directly from Digital.

Figure 7 shows the locations of the various terminator and resistor assemblies and voltage test points on the backpanel.

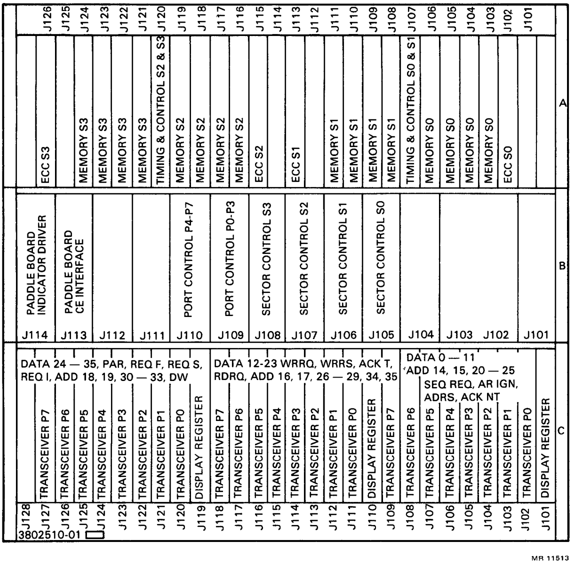


Figure 6   Card Placement Locations

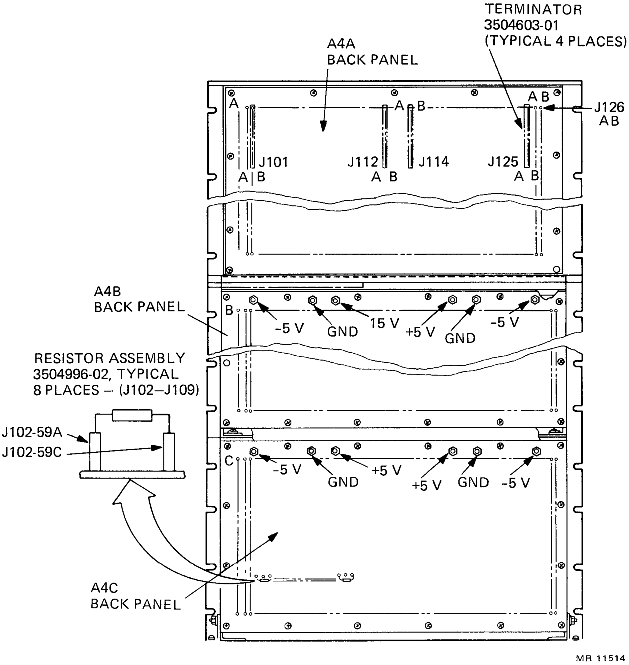


Figure 7   Terminator and Resistor Assemblies

Table 2 PWBA Table - 16 K Modules with Validate Storage

Abbr.	Title	Ampex Part No.	Digital Part No.
DR	Display Register	3506766-01**	29-80348
TR	Transceiver	3506195-01*	29-80350
CI	Control Interface	3506182-01**	29-80383
ID	Indicator Driver	3502283-01	29-80369
SC	Sector Control	3506180-01**	29-80352
PC	Port Control	3504400-__+	29-80371+
TC	Timing & Control	3502924-02	29-80374
M	Memory BD	3502915-02	29080372
ECC	ECC	3504406-03	29-80373

* Alternate part - 3281609-01
** Contains validate storage feature
\+ Part number depends on storage size:

Storage	Ampex Part No.	Digital Part No.
256 K	3504400-01	--
512 K	3504400-02	--
768 K or 1024 K	3504400-03	29-80371

Table 3 PWBA Table - 16 K Modules (No Validate Storage)

Abbr.	Title	Ampex Part No.	Digital Part No.
DR	Display Register	3280998-01	29-80366
TR	Transceiver	3281609-01*	
CI	Control Interface	3504408-01	
ID	Indicator Driver	3502283-01	
SC	Sector Control	3504402-01	
PC	Port Control	3504400-__+	
TC	Timing & Control	3502924-02	
M	Memory BD	3502915-02	
ECC	ECC	3504406-03	

* Alternate part - 3506195-01
\+ Part number depends on storage size:

Storage	Ampex Part No.
256 K	3504400-01
512 K	3504400-02
768 K or 1024 K	3504400-03

Table 4 PWBA Table - 64 K Modules

Abbr.	Title	Ampex Part No.	Digital Part No.
DR	Display Register	3506766-01	29-80348
TR	Transceiver	3506195-01	29-80350
CI	Control Interface	3506204-01	29-80349
ID	Indicator Driver	3506293-01	29-80351
SC	Sector Control	3506180-01	29-80352
PC	Port Control*	3506206-__+	29-803__+
TC	Timing & Control	3506678-02	29-80356
M	Memory BD*	3506685-02	29-80358
ECC	ECC	3506208-01	29-80359

* Refer to Tables 20 and 21 for external interleave card locations.
\+ Part number depends on storage size:

Storage	Ampex Part No.	Digital Part No.
1024 K	3506206-01	29-80353
2048 K	3506206-02	29-80354
3072 K or 4096 K	3506206-03	29-80355

Equipment Configuration

Assign Unit Starting Address - To set the unit starting address, determine the amount of memory below the Ampex unit; then refer to Table 5 for the value of addresses, bits 14-20. Set UNIT STARTING ADDRESS switches accordingly. Unit Starting Address switches are located on the ARM-10LS Control Panel.

NOTE
After setting the starting address switches, the reset switch must be pressed to properly configure system size and last address boundary.

Table 5 Unit Starting Address

Qty Of Memory Below Unit	Unit Starting Address Switches						
	A14	A15	A16	A17	A18	A19	A20
0 K	0	0	0	0	0	0	0
32 K	0	0	0	0	0	0	1
64 K	0	0	0	0	0	1	0
128 K	0	0	0	0	1	0	0
192 K	0	0	0	0	1	1	0
256 K	0	0	0	1	0	0	0
320 K	0	0	0	1	0	1	0
384 K	0	0	0	1	1	0	0
448 K	0	0	0	1	1	1	0
512 K	0	0	1	0	0	0	0
576 K	0	0	1	0	0	1	0
640 K	0	0	1	0	1	0	0
704 K	0	0	1	0	1	1	0
768 K	0	0	1	1	0	0	0
832 K	0	0	1	1	0	1	0
896 K	0	0	1	1	1	0	0
960 K	0	0	1	1	1	1	0
1024 K through 1984 K	0	1	*	*	*	*	*
2048 K through 3008 K	1	0	*	*	*	*	*
3072 K through 4096 K	1	1	*	*	*	*	*

* Repeat all of above.

Assign Logical Sectors to Physical Sectors - The two Sector Index switches on the ARM-10LS Control Panel determine the assignment of logical sectors to physical sectors. For normal operation, set both switches to the 0 position (down). In this position logical sector numbers correspond to physical sectors, i.e., when the CPU addresses Sector 0, it will select physical Sector 0 in the ARM-10LS. Table 6 lists the logical/physical sector assignments for all combinations of Sector Index switch settings.

Table 6 Sector Index Switch Selections

Sector Selected By CPU	Physical Sector Selected* (Memory Busy Indicator On)			
	SW11	SW10	SW01	SW00
S0	S1	S2	S3	S0
S1	S2	S3	S0	S1
S2	S3	S0	S1	S2
S3	S0	S1	S2	S3

*Sector Display Switches = 00

Establish Interleave Level - Two sets of switches, shown in Figure 8, are used to establish the interleave level of the unit. The INTERLEAVE switch determines the Internal level of interleave. The EXT INTERLEAVE switch and the SELECT (MSB, LSB) switches determine external interleave level.

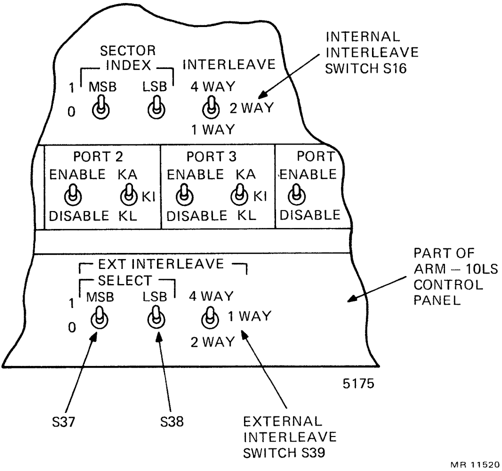


Figure 8 Control Panel Interleave Switch Locations

Internal Interleave Switch Settings - Set the internal INTERLEAVE switch as required by site conditions. (Set to 4-way for external interleave.) The EXT INTERLEAVE switch is set to the center (1-WAY) position for normal operation.

External Interleave Switch Settings - This mode is required for proper operation in multiprocessor applications such as shared multiprocessor (SMP) systems. Set the INTERLEAVE switch to the 4-WAY position.

When the EXT INTERLEAVE switch is set to 2-WAY, the unit memory size displayed at the control panel is twice the unit size. The unit can be operated as unit 0 or unit 1, depending on the position of the MSB SELECT switch.

When the EXT INTERLEAVE switch is set to 4-WAY, the unit memory size displayed at the control panel is four times the unit size. The unit can be operated as unit 0, 1, 2, or 3 depending on the positions of the MSB and LSB SELECT switches. Refer to Table 8 for EXT INTERLEAVE switch settings.

Table 8 External Interleave Switch Positions

Interleave Level	ARM 10LS Unit No.	EXT INTERLEAVE Switch Settings MSB	LSB	INTL SW S39
2-Way External Interleave	0	0	X	2-Way
	1	1	X	2-Way
4-Way External Interleave	0	0	0	4-Way
	1	0	1	4-Way
	2	1	0	4-Way
	3	1	1	4-Way

X = Not Used

NOTE
Set interleave switch (S16) to 4-WAY when using EXT INTERLEAVE. Each ARM-10LS unit must have the same starting address.

Enable Memory Modules and Set Memory Board Select Address - Switch locations are shown and identified in Figures 9 and 10. At each Memory PWBA location in Row A4A, set the enable toggle switch to the down position. Set the thumbwheel switch on each Memory PWBA as required by Table 9.

Note that board selection is not dependent upon slot location; therefore, the installer may set memory board thumbwheel switches in any orderly sequence, as long as sector and storage size requirements are observed. For example, in a 256 K system, four Memory PWBAs are required (one/sector). The Memory PWBAs may be inserted into any memory slot, when there is only one board per sector and the thumbwheel switch is set to 0.

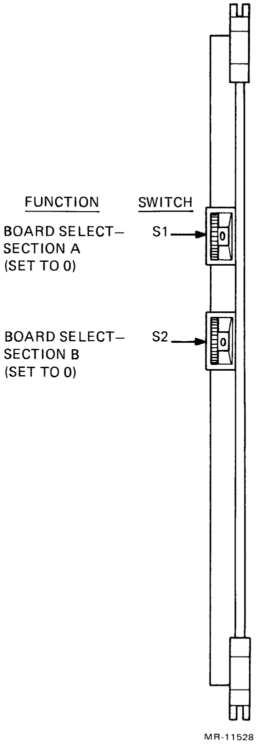


Figure 9 Timing and Control PWBA

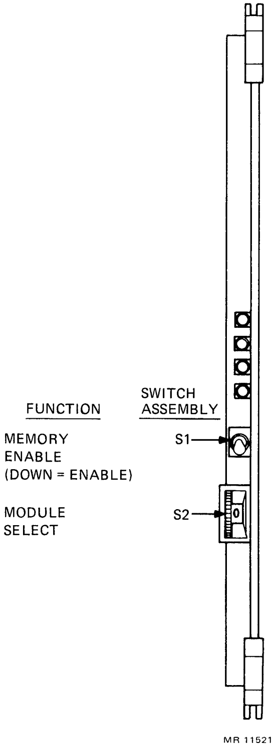


Figure 10 Memory PWBA Showing Switches

Table 9 Memory PWBA Thumbwheel Switch Settings

Unit Storage Size	Sector 0	Sector 1	Sector 2	Sector 3	Thumbwheel Switch Settings
256 K	J103	J108	J116	J121	0
512 K	J104	J109	J117	J122	1
768 K	J105	J110	J118	J123	2
1024 K	J106	J111	J119	J124	3

Set Memory Timing and Control Switch Settings - Memory Timing and Control PWBAs, located at AJ107 and AJ120, have card edge-mounted thumbwheel switches S1 and S2. (See Figures 9 and 10.) Both switches on each card must be set to the number 0.

Set ECC PWBA Switches - Each of the four ECC PWBAs (AJ102, AJ113, AJ115, and AJ126) has four card edge-mounted switches, SW1, SW2, SW3, and SW4. (See Figure 11 for switch locations.) Switches SW3 and SW4 are for maintenance display use. However, switches SW1 and SW2 must be set in the down position for normal operation.

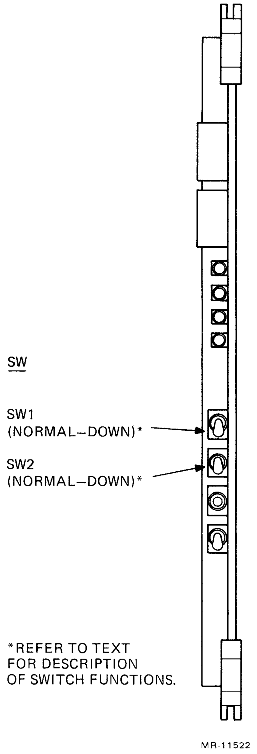


Figure 11 ECC PWBA Switches

Checks and Adjustments

RDRS Adjustment - Perform the following procedure to adjust the RDRS signal.

1. Make certain that the ARM-10LS address is above the first 20 (octal) system addresses. Run memory diagnostic scope loops, single-word read, in 4-bus mode. Use display panel indicators to determine the port-to-sector address, then move scope probe to the appropriate port for each sector. Observe RDRS and Data Warning signals at the following locations.

Signal	**IO Panel (A4D) Location**
DATA WARNING (DW)	PIN 7A
READ RESTART (RDRS)	PIN 2A

2. At PWBA locations B105-B108, adjust potentiometers K3 and K4. The timing must be set as shown in Figure 12. Potentiometer K4 is used to adjust pulse width T1, and potentiometer K3 is used to adjust pulse width T2.

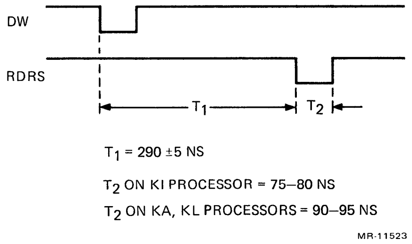


Figure 12 RDRS Timing Adjustments

Memory Request Adjustment

1. Load the Ampex diagnostic program and boot the system.

2. Observe the memory system control panel and note any control errors. If control errors are detected, proceed with step 3.; otherwise, continue with normal port testing procedures.

3. Locate potentiometer K8 on the failing Sector Control card.

NOTE
This component is a 22-turn potentiometer. Make certain it is initially full CCW; once the adjustment is made on one Sector Control, go to all other sectors and set potentiometers to the same setting.

4. Turn K8 three turns clockwise.

5. Repeat steps 1 through 5 unless the clockwise turns on K8 are greater than 22. If the number of turns is greater than 22 and control errors are still occurring, excessive electrical noise is present. To correct this condition, a "glitch protect" option may be added to the memory. To install this option, refer to the Glitch Protection Option section.

Glitch Protection Option - To install this option, proceed as follows.

1. Refer to Figure 13 for locations of backpanel pins for this option.

2. Turn off memory system power.

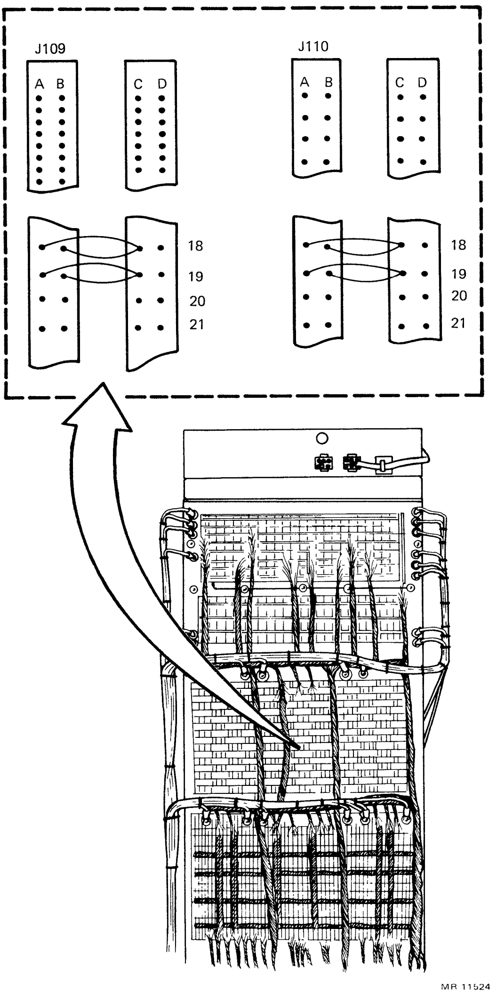


Figure 13 Glitch Protection Jumpers

3. Use wirewrap tool and 30 AWG Kaynar wire to make the following backpanel connections.

	From		To	
	Conn.	Pin	Conn.	Pin
PORT 0	BJ109	18A	BJ109	18C
PORT 1	BJ109	18B	BJ109	18C
PORT 2	BJ109	19A	BJ109	19C
PORT 3	BJ109	19B	BJ109	19C
PORT 4	BJ110	18A	BJ110	18C
PORT 5	BJ110	18B	BJ110	18C
PORT 6	BJ110	19A	BJ110	19C
PORT 7	BJ110	19B	BJ110	19C

4. Turn potentiometer K8 30 turns counterclockwise.

5. Continue with normal port testing procedures.

OPERATION

General Information

This section provides operating instructions for the ARM-10LS memory. When operating in a data processing system the memory is controlled by the CPU. Manual operation is generally limited to turning power on and off, and establishing the desired operating modes before on-line operation. Once the unit is on-line, further operator intervention should not be requried.

Controls and Indicators

Operator controls and status indicators are located on the EPO panel, blower assembly, control panel, and memory, memory timing and control, and ECC PWBAs. The following paragraphs describe these controls and indicators.

EPO Panel - Power for the memory is controlled at the EPO panel. The controls consist of switches that select remote or local power sequencing, activate and deactivate the memory, and control power supply operation. Related indicators provide a visual indication of the power status and the individual power supply voltage margins, and any other abnormal operating conditions. EPO panel controls and indicators are described in Table 10 and shown in Figure 14.

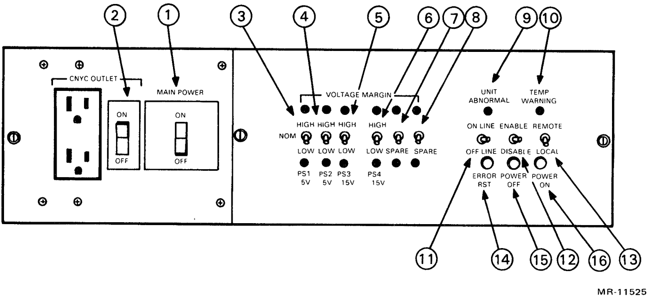


Figure 14 EPO Panel Controls and Indicators

Table 10 EPO Controls and Indicators

Fig. 14 Ref. No.	Control/Indicator Name	Type	Reference Designation	Function
1	MAIN POWER	Circuit Breaker Indicator	CB2	Controls ac input power to memory; provides automatic overload protection for the ac input circuit. Illuminates (white) when active.
2	CNVC OUTLET	Circuit Breaker Indicator	CB1	Controls ac input power to convenience outlet (J1); this outlet is intended to provide voltage for external test equipment used by service personnel. Illuminates (white) when active.
3	PS1(+5 V) VOLTAGE MARGIN	Toggle Switch (3-position)	S6	When in up position margins power supply 1 (+5 V) high. When in down position, margins power supply 1 low. Should be in center (NOM) position for normal operation.
	HIGH	LED (red) (above PS1 toggle switch)	DS1	Illuminates when PS1 is in high margin condition.
	LOW	LED (red) (below PS1 toggle switch	DS9	Illuminates when PS1 is in low margin condition.
4	PS2(-5.7 V) VOLTAGE MARGIN	Toggle Switch (3-position)	S7	Not used. Power supply 2 (-5.7 V) cannot be margined.
	HIGH	LED (red) (above PS2 toggle switch)	DS2	
	LOW	LED (red) (below PS2 toggle switch)	DS10	
5	PS3(-5.2 V) VOLTAGE MARGIN	Toggle Switch (3-position)	S8	When in up position, margins power supply 3 (-5.2 V) high. When in down position, margins power supply 3 low. Should be in center (NOM) positions for normal operation.
	HIGH	LED (red) (above PS3 toggle switch)	DS3	Illuminates when PS3 is in high margin condition.
	LOW	LED (red) (below PS3 toggle switch)	DS11	Illuminates when PS3 is in low margin condition.

Table 10 EPO Controls and Indicators (Cont)

Fig. 14 Ref. No.	Control/Indicator Name	Type	Reference Designation	Function
6	PS4 (+5 V) VOLTAGE MARGIN	Toggle Switch (3-position)	S9	When in up position, margins power supply 4 (+5 V) high. When in down position, margins power supply 4 low. Should be in center (NOM) position for normal operation.
	HIGH	LED (red) (above PS4 toggle switch)	DS4	Illuminates when PS4 is in high margin condition.
	LOW	LED (red) (below PS4 toggle switch)	DS12	Illuminates when PS4 is in low margin condition.
7	PS5(+12 V) VOLTAGE MARGIN	Toggle Switch (3-position)	S10	When in up position, margins power supply 5 (+12 V) high. When in down position, margins power supply 5 low. Should be in center (NOM) position for normal operation.
	HIGH	LED (red) (above PS5 toggle switch)	DS5	Illuminates when PS5 is in high margin condition.
	LOW	LED (red) (below PS toggle switch)	DS13	Illuminates when PS5 is in low margin condition.
8	PS6 VOLTAGE MARGIN	Toggle Switch	S11	Not Used
	HIGH	LED (red)	DS6	
	LOW	LED (red)	DS14	
9	UNIT ABNORMAL	LED (red)	DS7	Illuminates when a power supply is in high or low margin condition; also indicates that ac power is below required operating range.
10	TEMP WARNING	LED (red)	DS8	Not Used
11	ON-LINE/ OFF-LINE	Toggle Switch (2-position)	S12	Not Used
12	ENABLE/ DISABLE	Toggle Switch (2-position)	S13	Not Used
13	REMOTE/ LOCAL	Toggle Switch (2-position)	S14	Must be set to LOCAL position. NOTE If switch is in REMOTE position, memory power-on is inhibited.

Table 10 EPO Controls and Indicators (Cont)

Fig. 14 Ref. No.	Control/Indicator Name	Type	Reference Designation	Function
14	ERROR RST	Pushbutton Switch (white)	S15	Not Used
15	POWER OFF	Momentary Pushbutton Switch (red)	S16	Removes all dc power to the memory and ac power to the blower assemblies.
16	POWER ON	Momentary Pushbutton Switch (green)	S17	Applies ac power to the memory and ac power to the blowers. Should be held for 3 seconds when applying power.

Blower Assembly - A circuit breaker is located at the rear of the blower assembly, as illustrated in Figure 15. If the circuit breaker is tripped, press the center button to restore power. Note that if the breaker is tripped, power will be removed from the entire memory unit.

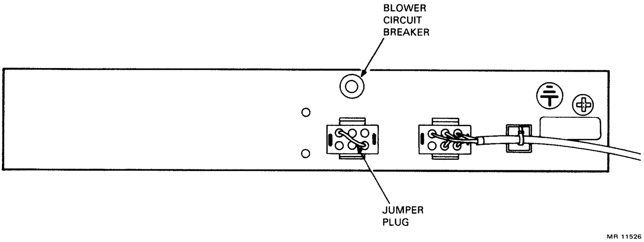


Figure 15 Blower Assembly Circuit Breaker Location

Control Panel - Controls and indicators located on the Control Panel (Figure 16) are functionally divided into Control, Error Status, and Maintenance Groups. Each group is described in the following paragraphs.

Control Group - The Control Group establishes memory operating parameters. Included are switches for enabling each of the ports, setting address boundaries, and establishing memory request type and interleave mode.

a. Port and Sector Enable switches. The three types of switches are described in Table 11.

b. Unit Starting Address. Seven switches are used to establish the memory starting address. The address may be set on 64 K boundaries, depending on the quantity of memory below the unit. Table 5 indicates the positions for this parameter.

c. Next Starting Address. Seven LED indicators display the starting address of the next unit. If the External Interleave switch is set to 1-WAY, the indicators display the ARM-10LS last address plus 1 (maximum storage size plus 1). If the External Interleave switch is set to 2-WAY, the next starting address indicators display twice the storage size plus 1. Four times the actual storage size plus 1 is displayed if the External Interleave switch is set to 4-WAY. When internal interleave (2-WAY or 4-WAY) is active, the next starting address indicators are not affected. The address range reflects system capacity only.

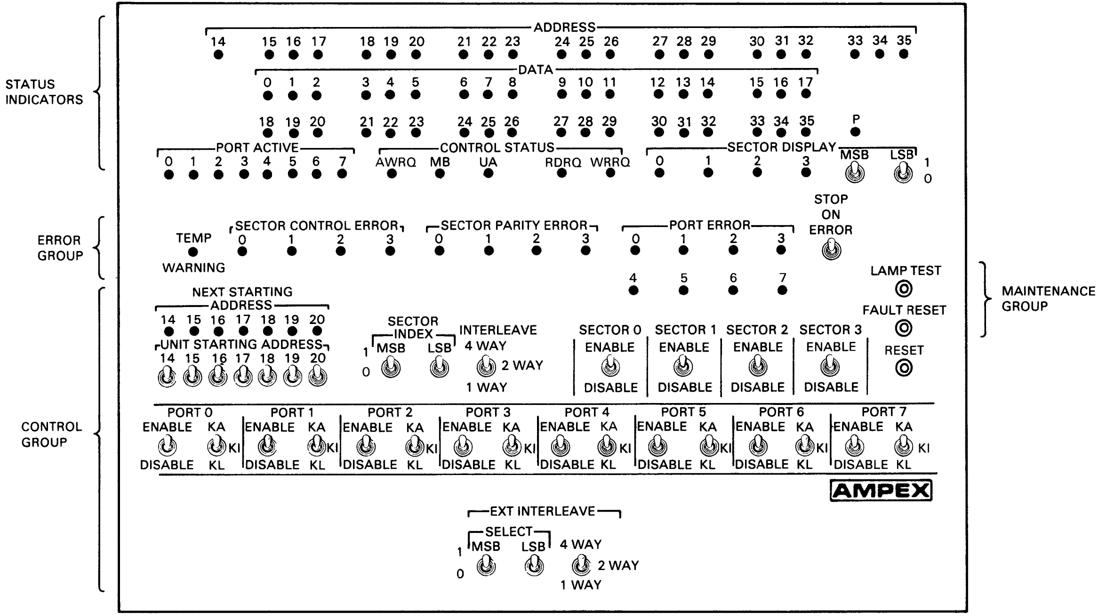


Figure 16 Main Control Panel

MR-11527

d. Sector Index. Two switches are used to determine the order in which sectors will be selected. Table 12 shows the sector index switch settings.

e. Interleave. A 3-position toggle switch is used to select the Memory Interleave mode. The 1-WAY position selects noninterleaved addressing. In the 2-WAY position, Sector 0 is interleaved with Sector 1, and Sector 2 is interleaved with Sector 3. In the 4-WAY position, all four sectors can be simultaneously addressed.

f. Reset. A single pushbutton switch that initializes the memory unit. When pressed, RESET presets control flip-flops in 6400S Memory, resets error latches in ECC PWBAs, and resets port request latches in all Sector Control PWBAs.

Every time RESET is pressed, circuitry in the Control Interface PWBA automatically configures the system memory capacity (last address) and the next unit starting address is displayed on the control panel indicators.

NOTE
Do not press reset during system operation.

Table 11 Port and Sector Enable Switches

Name	Type	Function
PORT ENABLE/DISABLE	2-position Toggle Switch	Places associated memory port on-line or off-line. In the ENABLE position, port is on-line (connected to CPU or channel); in DISABLE position, port is switched off-line.
PORT KA/KI/KL	3-position Toggle Switch	Determines the type of request to which the memory will respond. (Switches are set at the time of installation.)
SECTOR ENABLE/DISABLE	2-position Toggle Switch	Enables or disables associated sector. In ENABLE position, physical sector can be accessed by CPU. In DISABLE position, CPU access to sector is blocked.

Table 12 Sector Index Addressing

Memory Unit Sector	Sector Display Switches MSB	LSB	Sector Index Switches MSB	LSB	Sector Display Indicator ON	Electrical Address (CPU Sector Select)	Physical Address (Memory Sector)
0	0	0	0	0	0	S0	S0
	0	0	0	1	1	S1	S0
	0	0	1	0	2	S2	S0
	0	0	1	1	3	S3	S0
1	0	1	0	0	1	S1	S1
	0	1	0	1	2	S2	S1
	0	1	1	0	3	S3	S1
	0	1	1	1	0	S0	S1
2	1	0	0	0	2	S2	S2
	1	0	0	1	3	S3	S2
	1	0	1	0	0	S0	S2
	1	0	1	1	1	S1	S2
3	1	1	0	0	3	S3	S3
	1	1	0	1	0	S0	S3
	1	1	1	0	1	S1	S3
	1	1	1	1	2	S2	S3

Note that sector electrical bit assignment is a function of memory size and interleave settings.

g. External Interleave Switches. A 3-position toggle switch (S39) which sets external interleave level in multiunit installations. This switch controls SELECT switches MSB and LSB. In the 4-WAY position, both MSB and LSB are enabled. In the 2-WAY position, only MSB is enabled. In the 1-WAY position, both MSB and LSB are disabled (external interleave disabled).

h. SELECT MSB, LSB. Two 2-position toggle switches that determine the order or unit selection in a multiunit installation. The settings for these switches, in conjunction with EXT INTERLEAVE (switch S39), are defined in Table 8.

Error Group - The Error Group permits the operator to locate control and port errors.

a. STOP ON ERROR toggle switch. The 2-position STOP ON ERROR switch has the following functions.

   Up Position. If a Sector Control, Sector Parity, or Port Error is detected, the memory ceases operation; indicators display memory status at the time of error detection.

   Down Position. Memory continues operation under control of the CPU. Indicators accumulate and display errors.

b. FAULT RESET pushbutton switch. When pressed, FAULT RESET clears all control panel error displays (Sector Control, Sector Parity, and Port Error). In addition, the Fault Reset Function clears error displays on the four ECC PWBAs including UE, CE, DOPE, DIPE, and Card/Chip indicators. This switch may be pressed during system operation (also refer to the Control Group section).

c. Sector Control Error Indicators. During a Write or Read-Modify-Write cycle, if a Write Restart signal has not been received from the CPU within 25 microseconds, a control error occurs, and the Sector indicator lights. Control error indicators also light whenever an invalid request is detected. The condition also occurs when a cycle (Read or Write) has not been completed within 24 microseconds.

d. Sector Parity Error Indicators. A SECTOR PARITY ERROR indicator lights to identify the sector in which a Read or Write parity error has occurred. It should be noted that sector identification relates to physical locations, and is not influenced by interleaving or Sector Index addressing.

e. Port Error Indicators. These indicators identify I/O ports in which control or parity errors occur during a Write or Read-Modify-Write cycle. If the STOP ON ERROR switch is in the down position, accumulated errors will be displayed.

Refer to Table 13 for information on sector control error, sector parity error, and port error LED indicators.

Table 13 LED Error Indicators

Error Type	Sector Control Error	Sector Parity Error	Port Error
Read Error	LED OFF	LED ON	LED OFF
Write Error	LED OFF	LED ON	LED ON
Control Error	LED ON	LED OFF	LED ON

Status Group - The Status Group provides visual indicators relating to memory operations.

a. Sector Display. The Status Group indicators (Port Active, Control Status, and Address and Data) are shared by the four Memory Sectors. The status of only one sector can be displayed at a time. To view the status of any sector, the SECTOR DISPLAY switches must be set to the binary number representing the Sector. The corresponding SECTOR DISPLAY indicator will light.

b. Port Active. Illuminated LEDs in the port active group indicate operating ports.

c. Control Status. Status Control indicators described in Table 14 are used to monitor operating mode and status of the memory.

d. Address and Data. These status indicators provide a continuously updated display of transceiver address register and data register contents. An illuminated lamp indicates that the register contains a logical 1.

NOTE

The memory unit address is "normalized" to a number ranging from 0 to the value of the last address (256 K, 512 K, 768 K, or 1024 K). Address bits 14-17 on the control panel are not used. The selected physical sector is indicated by sector display indicators. Address bits 18, 19 always display the selected memory module.

Table 14 Control Status Indicators

Control/Indicator Name	Type	Function	Operation
AWRQ	LED	Monitors Sector Await Request Memory Status.	Lights to indicate interface is not busy and is awaiting an access request from one of the memory ports. Extinguishes when interface is busy. When sector is busy servicing a port, all other ports are locked out.
MB	LED	Monitors Memory Busy from Sector.	Lights to indicate Memory Sector is busy with a cycle. Always dimly lit due to refresh cycle.
UA	LED	Monitors Unit Available Status of Memory Sector.	Lights to indicate that sector is available to execute a cycle. Extinguishes when a sector is busy with a cycle, or has completed the Read portion of a Read-Modify-Write cycle and is awaiting WRRS (Write Restart).
RDRQ+	LED	Monitors Read Request input line.	Lights to indicate a read cycle requested.
WRRQ+	LED	Monitors Write Request.	Lights to indicate a write cycle is requested.

+If both RDRQ and WRRQ indicators are illuminated, the unit is in Read-Modify-Write mode.

Maintenance Group - The Maintenance Group consists of the LAMP TEST switch and the TEMP WARNING indicator. These are shown in Figure 16 and are described in Table 15.

Memory PWBA - Two switches and four indicators are located on each memory PWBA. These components are illustrated in Figure 17 (and Figure 10) and described in Table 16.

Table 15 Maintenance Controls and Indicators

Control/Indicator Name	Type	Function	Operation
LAMP TEST	Toggle Switch	Checks for faulty panel indicators.	Press this switch to test all indicators. Indicators that do not light are faulty.
TEMP WARNING	LED	Indicates over-temperature condition. Temperature within cabinet is 136^{o}F or greater.	Illuminates if overheating occurs. Extinguishes when temperature returns to normal.

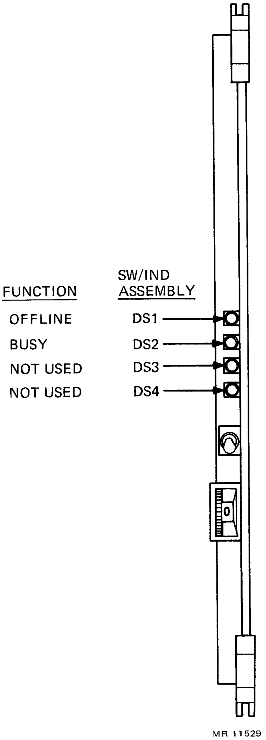


Figure 17 Memory PWBA Showing Lights

Table 16 Memory PWBA Controls and Indicators

Control/Indicator Name	Type	Reference Designation	Function
OFF LINE	LED (yellow)	DS1	Illuminates when Memory PWBA is disabled.
BUSY	LED (green)	DS2	Illuminates when Memory PWBA is performing Read or Write cycles.
MULTIBIT ERROR	LED (red)	DS3	Not Used.
SINGLE BIT ERROR	LED (red)	DS4	Not Used.
ENABLE	Toggle Switch	S1	In the down position, enables the Memory PWBA. In the up position, places the Memory PWBA off-line (disabled).
MODULE SELECT	Thumb-wheel Switch	S2	Provides four board select control settings (0-3). PWBA is selected when address line inputs match hex output of S2. Memory PWBAs in each sector must be set as follows. **Switch Setting** / **Memory PWBA Selected** 0 — 1st 64K (64K/Sector) 1 — 2nd 64K (128K/Sector) 2 — 3rd 64K (192K/Sector) 3 — 4th 64K (256K/Sector)

Memory Timing and Control PWBA - The Memory Timing and Control board shown in Figure 9 has two hexadecimal thumbwheels switches, S1 and S2. Each switch must be set to the "0" position.

ECC PWBA - Six indicators, visible through slots in the memory cover panel, reflect error conditions in the memory. ECC PWBA indicators and related controls are shown in Figure 18 and described in Table 17. Note that any detected error causes the indicators to light. From combinations of error indications (latched at the time of error), the following types of information can be determined.

1. Type of error: read data error or write data error.
2. Type of read error: single-bit (corrected) or double-bit (uncorrected).
3. Location of error: sector, card, chip group, and data bit number.

NOTE

The Chip Group is one of four physical groups of 43 memory chips located on the memory PWBA. Refer to the MEMORY CHIP FAULT ISOLATION section for the memory chip fault isolation procedure.

Power-On Procedure - Power-on sequencing is performed internally, eliminating the need for special precautions. Since power is applied to memory from the memory EPO panel, the REMOTE/LOCAL switch must be set to the LOCAL position. To power up the memory, set the MAIN POWER circuit breaker to ON and press the POWER ON pushbutton switch for approximately three seconds. The blower fans should operate, indicating that power is applied to the unit. Also check that VOLTAGE MARGIN, UNIT ABNORMAL, and TEMP WARNING indicators are extinguished.

On-Line/Off-Line Operation
For normal operation, set port ENABLE/DISABLE switches to ENABLE. To deselect a memory unit, set the port ENABLE/DISABLE switches to DISABLE.

Power-Off Procedure
Power may be removed at the memory EPO panel by pressing the POWER OFF pushbutton switch on the EPO panel.

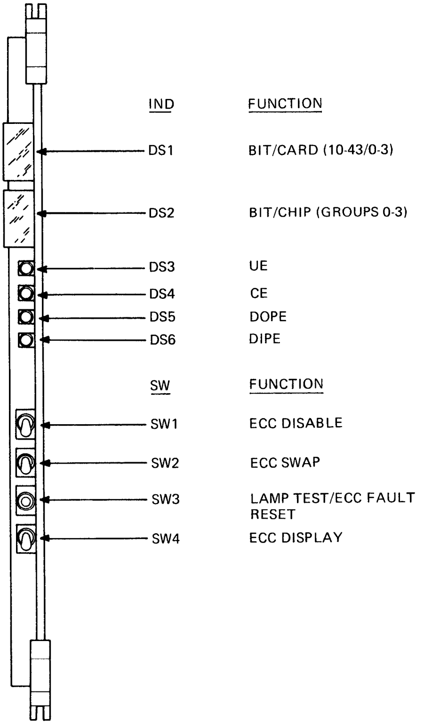


Figure 18 ECC PWBA Controls and Indicators

Table 17 ECC PWBA Controls and Indicators

Control/Indicator Name	Type	Reference Designation	Function
ECC DISABLE	Toggle switch (2-position)	SW1	Up position: Disables ECC (error checking and correction) Memory stores and checks single odd parity bit for entire data word. Down position: Enables ECC function. Memory corrects and reports single-bit data errors and reports multibit errors.
ECC SWAP	Toggle switch (2-position)	SW2	Swaps bits 29-35 with bits 36-42 so ECC bits can be checked. Switch is enabled in the up position for maintenance purposes only. This function is operational when SW1 (ECC DISABLE) is in the up position.
LAMP TEST/ ECC FAULT RESET	Push-button (momentary)	SW3	When pressed momentarily, lights all ECC LED indicators and clears ECC errors. NOTE: Switch S3 may be pressed when the system is running.
ECC Display	Toggle (2-position)	SW4	Controls hex displays DS1 and DS2. Up position: Causes DS1, DS2 to display failing Memory PWBA and failing 16 K chip group. Memory PWBA number is read on DS1 while 16 K chip group number is read on DS2. Down position: Causes DS1, DS2 to display Memory PWBA failing data bit number. MSD is read on DS1 and LSD is read on DS2.
Bit/Card	Hex Display	DS1	When SW1 is down and SW4 is up, DS1 displays the failing Memory PWBA as a decimal number 0-3. (Corresponds to module select number on Memory PWBA switch S2.)
Bit/Card	Hex Display	DS1	When SW1 is down and SW5 is down, DS1 displays the most significant decimal digit of the failing single data bit (MSD will be a number 0-4). If the failure is a multibit error, DS1 displays the hex digit "F".
Bit/Chip	Hex Display	DS2	Used in conjunction with SW1 and SW4 to display failing 16 K chip group, depending on position of SW4. When SW1 is down and SW4 is up, DS2 displays the failing 16 K group as a decimal number 0-3. When SW1 is down and SW4 is up, DS2 displays the failing 16 K group as a decimal number 0-3.

Table 17 ECC PWBA Controls and Indicators (Cont)

Control/Indicator Name	Type	Reference Designation	Function
			When SW1 is down and SW4 is down, DS2 displays the least significant decimal digit of the failing single data bit (LSD will be a number 0-9). If the failure is a multibit error, DS2 displays the hex digit "F". If the decimal point on DS2 is always ON, the indicated error is a "hard" failure. DS2 also blinks on any ECC error (both correctable and uncorrectable).
UE	LED (red)	DS3	Illuminates when a multibit (uncorrectable) error has been detected in data read from the Memory PWBA. Is cleared by pressing the Control Panel FAULT RESET or ECC TEST/FAULT RESET pushbutton. Indicator is active only when SW1 is down (ECC enabled.)
CE	LED (red)	DS4	Illuminates when a single-bit (correctable) data error has been detected in data read from the Memory PWBA. Is cleared by Control Panel FAULT RESET or ECC LAMP TEST/FAULT RESET pushbutton. Indicator is active only when SW1 is in down position (ECC enabled).
DOPE	LED (red)	DS5	Illuminates when bad parity is detected in read data (Data Out). Indicator is active only when SW1 is in the up position (ECC disabled). Indicator DS5 is cleared by pressing Control Panel FAULT RESET or ECC LAMP TEST/FAULT RESET pushbutton.
DIPE	LED	DS6	Illuminates when bad parity is detected in write data (data to memory). Indicator DS6 is cleared by pressing Control Panel FAULT RESET or ECC LAMP TEST/FAULT RESET pushbutton.

MEMORY CHIP FAULT ISOLATION

If a failure can be isolated to one data bit on a memory PWBA, the failing memory chip can be located and replaced. To isolate a failing memory chip, note the status of address bits 20 and 21 at the time of failure; then refer to Figure 19. The 172 memory chips are partitioned into four 16 K X 43 addressable blocks corresponding to the status of address bits 20 and 21.

Address Bit

20	21	
H	H	1st 16 K
H	L	2nd 16 K
L	H	3rd 16 K
L	L	4th 16 K

H - High Logic Level (Logical 0)
L - Low Logic Level (Logical 1)

Locate chip at the intersection of data bit and 16 K block; then remove and replace chip, as follows.

CAUTION
Memory ICs are MOS devices, which can be damaged by static electric charges.

1. Set PWBA on a flat surface and place one hand on PWBA.
2. Replacement memory ICs are normally packaged in a block of conductive foam; place the foam block on the PWBA.
3. Keep one hand in contact with the PWBA; remove and replace memory chip with free hand.

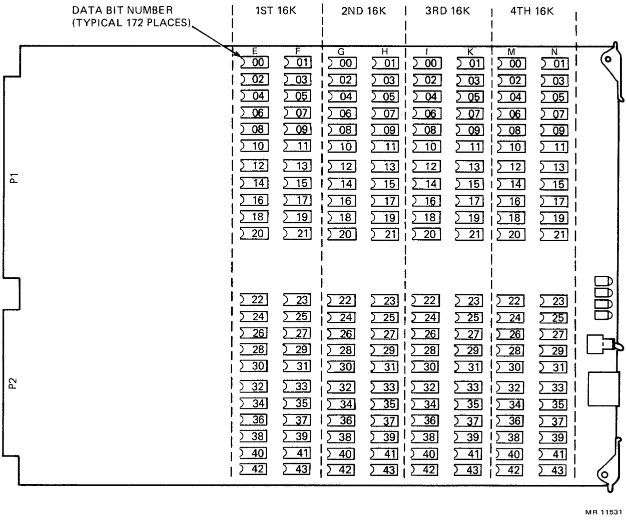


Figure 19 Memory PWBA Chip Locations

INSTALLATION REFERENCE INFORMATION
This section provides information that may be used in conjunction with memory system installation tasks.

Table 18 Memory Address Assignments

Interleave Level	Box Size	Unit Select	Sector Select	Module Select	Block Select	Intern Addr.
ONE-WAY EXTERNAL (NO INTERLEAVE)	1 M	14,15	16,17	18,19	20,21	22-35
	768 K	14,15	16,17	18,19	20,21	22-35
	512 K	14,15	16,17	18,19	20,21	22-35
	256 K	14-17	18,19	-	20,21	22-35
ONE-WAY EXTERNAL or TWO-WAY INTERNAL	1 M	14,15	16,35	17,18	20,21	22-34,19
	768 K	14,15	16,35	17,18	20,21	22-34,19
	512 K	14,16	17,35	18	20,21	22-34,19
	256 K	14-17	18,35	-	20,21	22-34,19
ONE-WAY EXTERNAL or FOUR-WAY INTERNAL	1 M	14,15	34,35	16,17	20,21	22-34,18,19
	768 K	14,15	34,35	16,17	20,21	22-33,18,19
	512 K	14-16	34,35	17	20,21	22-33,18,19
	256 K	14-17	34,35	-	20,21	22-33,18,19
TWO-WAY EXTERNAL or FOUR-WAY INTERNAL	1 M	14,34	15,35	16,17	20,21	22-33,18,19
	768 K	14,34	15,35	16,17	20,21	22-33,18,19
	512 K	14,15,34	16,35	17	20,21	22-33,18,19
	256 K	14-16,34	17,35	-	20,21	22-33,18,19
FOUR-WAY EXTERNAL or FOUR-WAY INTERNAL	1 M	34,35	14,15	16,17	20,21	22-33,18,19
	768 K	34,35	14,15	16,17	20,21	22-33,18,19
	512 K	14,34,35	15,16	17	20,21	22-33,18,19
	256 K	14,15 34,35	16,17	-	20,21	22-33,18,19

Address example: 01410010 (1 M No Interleave)

- -	15 0	18 1	21 0	24 0	27 0	30 0	33 0
- -	16 0	19 0	22 0	25 0	28 0	31 0	34 0
14 0	17 1	20 0	23 1	26 0	29 0	32 1	35 0
0	1	4	1	0	0	1	0

Bits 17, 18, 23, and 32 were on, therefore sector 1, module 2, chip 0 failed.

Table 19 Octal Storage Barrier Addresses
One Megabyte by 16 K Increments

OCT	DEC	OCT	DEC
0000000-0037777	00K- 16K	2000000-2037777	512K- 528K
0040000-0077777	16K- 32K	2040000-2077777	528K- 544K
0100000-0137777	32K- 48K	2100000-2137777	544K- 560K
0140000-0177777	48K- 64K	2140000-2177777	560K- 576K
0200000-0237777	64K- 80K	2200000-2237777	576K- 592K
0240000-0277777	80K- 96K	2240000-2277777	592K- 608K
0300000-0337777	96K-112K	2300000-2337777	608K- 624K
0340000-0377777	112K-128K	2340000-2377777	624K- 640K
0400000-0437777	128K-144K	2400000-2437777	640K- 656K
0440000-0477777	144K-160K	2440000-2477777	656K- 672K
0500000-0537777	160K-176K	2500000-2537777	672K- 688K
0540000-0577777	176K-192K	2540000-2577777	688K- 704K
0600000-0637777	192K-208K	2600000-2637777	704K- 720K
0640000-0677777	208K-224K	2640000-2677777	720K- 736K
0700000-0737777	224K-240K	2700000-2737777	736K- 752K
0740000-0777777	240K-256K	2740000-2777777	752K- 768K
1000000-1037777	256K-272K	3000000-3037777	768K- 784K
1040000-1077777	272K-288K	3040000-3077777	784K- 800K
1100000-1137777	288K-304K	3100000-3137777	800K- 816K
1140000-1177777	304K-320K	3140000-3177777	816K- 832K
1200000-1237777	320K-336K	3200000-3237777	832K- 848K
1240000-1277777	336K-352K	3240000-3277777	848K- 864K
1300000-1337777	352K-368K	3300000-3337777	864K- 880K
1340000-1377777	368K-384K	3340000-3377777	880K- 896K
1400000-1437777	384K-400K	3400000-3437777	896K- 912K
1440000-1477777	400K-416K	3440000-3477777	912K- 928K
1500000-1537777	416K-432K	3500000-3537777	928K- 944K
1540000-1577777	432K-448K	3540000-3577777	944K- 960K
1600000-1637777	448K-464K	3600000-3637777	960K- 976K
1640000-1677777	464K-400K	3640000-3677777	967K- 992K
1700000-1737777	480K-496K	3700000-3737777	992K-1008K
1740000-1777777	496K-512K	3740000-3777777	1008K-1024K

Table 20 Typical Memory PWBA Locations for Internal Interleave (One Box)

Interleave Level	Port Control PWBA Version	Box Size	S3 M3 AJ124	M2 J123	M1 J122	M0 J121	S2 M3 AJ119	M2 J118	M1 J117	M0 J116	S1 M3 AJ111	M2 J110	M1 J109	M0 J108	S0 M3 AJ106	M2 J105	M1 J104	M0 J103
Internal	3506206-01	1 M				X				X				X				X
	3506206-02	2 M			X	X			X	X			X	X			X	X
1-WAY	3506206-03	3 M					X	X	X	X	X	X	X	X	X	X	X	X
	3506206-03	4 M	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Internal	3506206-01	1 M				X				X				X				X
	3506206-02	2 M			X	X			X	X			X	X			X	X
2-WAY	3506206-03	3 M			X	X			X	X	X	X	X	X	X	X	X	X
	3506206-03	4 M	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Internal	3506206-01	1 M				X				X				X				X
	3506206-02	2 M			X	X			X	X			X	X			X	X
4-WAY	3506206-03	3 M		X	X	X		X	X	X		X	X	X		X	X	X
	3506206-03	4 M	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

NOTES: 1. BOX SIZE is memory capacity per ARM-10LS cabinet.
2. X indicates where Memory PWBA is to be installed for the corresponding BOX SIZE.
3. Memory locations for 3 M size vary depending on interleave level.

Table 21 Typical Memory PWBA Locations for External 2-Way Interleave (Two Box)

Box Number	Port Control PWBA Version	Box Size	S3	S2	S1 M3 AJ111	S1 M2 J110	S1 M1 J109	S1 M0 J108	S0 M3 AJ106	S0 M2 J105	S0 M1 J104	S0 M0 J103
BOX 0	3506206-01	512 K						X				X
	3506206-02	1 M					X	X			X	X
	3506206-03	1.5 M				X	X	X		X	X	X
	3506206-03	2 M			X	X	X	X	X	X	X	X
BOX 1	3506206-01	512 K						X				X
	3506206-02	1 M					X	X			X	X
	3506206-03	1.5 M				X	X	X		X	X	X
	3506206-03	2 M			X	X	X	X	X	X	X	X

NOTES:
1. Sectors 2 and 3 of each box not used.
2. X indicates where memory PWBA is installed for corresponding BOX SIZE.
3. 512 K and 1.5 M box sizes are nonstandard.

Table 22 Typical Memory PWBA Locations for External 4-Way Interleave (Four Box)

Box Number	Port Control PWBA Version	Box Size	S3	S2	S1	S0 M3 AJ106	M2 J105	M1 J104	M0 J103
BOX 0	3506206-01	256 K							X
	3506206-02	512 K						X	X
	3506206-03	768 K					X	X	X
	3506206-03	1 M				X	X	X	X
BOX 1	3506206-01	256 K							X
	3506206-02	512 K						X	X
	3506206-03	768 K					X	X	X
	3506206-03	1 M				X	X	X	X
BOX 2	3506206-01	256 K							X
	3506206-02	512 K						X	X
	3506206-03	768 K					X	X	X
	3506206-03	1 M				X	X	X	X
BOX 3	3506206-01	256 K							X
	3506206-02	512 K						X	X
	3506206-03	768 K					X	X	X
	3506206-03	1 M				X	X	X	X

NOTES:
1. Sectors 1, 2, and 3 of each box not used.
2. X indicates where Memory PWBA is installed for corresponding BOX SIZE.

Cluster

Table of Contents

COMMON-FILE SYSTEM (CFS)

INTRODUCTION

Common File System (CFS) operates at the TOPS-20 file system level. This is done by "distributing" the file system over the CFS network. This distribution implies several things.

1. Processors must agree on file operations previously of concern only to a single processor.

2. The existing file system data base must be extended to accommodate additional information.

3. Other peripherally related parts of TOPS-20 were affected, for example, job numbers.

This section describes the specifics of the CFS kernel, CFSSRV, as well as the extensions and modifications of the file system. Since this is a design document, it is assumed the reader is familiar with the functional specification for CFS.

CFSSRV MODULE

The CFSSRV contains the kernel, or low-level, routines that implement CFS. The important pieces of CFSSRV are:

1. The resource manager/creater
2. The "voter"
3. The SCA connection manager
4. Initialization.

Also, CFSSRV has "interface" routines that are used by other system modules and services for interfacing to the kernel routines.

Resource Manager

The CFSSRV manages a hash table where CFS resources are kept. The hash table contains "hash chains" built as collisions occur. Each entry represents some resource, but the manager has no inherent knowledge of the actual resource represented. Each resource block has identifying information, state information, and a number of co-routine addresses that are used when certain events occur.

There are two types of resource blocks: long blocks and short blocks.

A long block is the same as a short block except that it has a bit mask representing the forks in the system. Resources must use a long block if they need to keep track of blocked forks, and need to wake the forks when the resource is available. See figure 1.

Each resource's name is used to compute its hash address and to identify the resource. The name is seventy-two bits and is in the second and third words of the hash packet. Also, a resource can be a resource code used for various purposes including:

1. Matching during voting.
2. As a mask during garbage collection and searching.

The first word of the name, the root code, is usually a structure name. It represents a resource class, such as files. The second word of the name, the qualifier, distinguishes specific resources within the class defined by the root name. Therefore, each type of file resource has a unique qualifier that may be constructed based on the resource type.

For example, the file open token (representing the open state of the file) is simply the file's index block address. The pair of names defines a unique and unambiguous resource.

The resource requester must supply the various call-back routine addresses. The resource manager has not already gathered any information about the actions to be taken for each resource. Therefore, when the requester builds a prototype resource packet, it must fill in any co-routine addresses that it wants honored. In general, the code that builds the prototype resource packet is in CFSSRV and, therefore, is closely allied with the resource manager itself. Call-back addresses need not be provided. In fact, there are resources, for example, directory locks, that provide no call-back co-routines, but rely entirely on the default mechanisms of the manager and the voter.

HSHLNK	LINK WORD
HSHROT	ROOT CODE
HSHQAL	QUALIFIER
HSHTIM	TIME STAMP
HSFLAG	FLAGS
HSHCOD	RESOURCE CODE
HSHPST	CALL-BACK ADDRESS ON AVAILABLE
	VOTE CODE AND COUNT
HSHNBT	NODE BIT TABLE
	CALL-BACK ON DEASSIGN
	DEFERRED VOTE DATA
HSHMSK	DEFERRED WAIT MASK,,OWNING FORK
HSHOPT	OPTIONAL DATA WORD 1
HSHOP1	OPTIONAL DATA WORD 2
HSHOKV	CALL-BACK WHEN VOTE OK
HSHCDA	CALL-BACK WHEN OPT DATA PRESENT
HSHFCT	FAIRNESS TIMER
HSHWTM	REVOTE WAIT TIME
HSHBKP	BACK POINTER
	FORK BIT TABLE IF LONG PKT

MR-16411

Figure 1 CFS Resource Block

The resource manager has several entry points. The main entry is CFSGET, but other entries are used if the resource is known to be local (exclusive to this processor).

The manager is responsible for entering packets in the hash table. It is the only agent that places new entries in the hash table.

The flags word has an "access type" used to record the resource access level. The manager and the vote processor use this field to determine if an entry or a vote request may be honored.

The manager may delete an old, unused entry, if doing so removes a resource conflict.

VOTER

The voter is in two parts.

1. The agent that starts votes on behalf of the resource manager.

2. The agent that processes in-coming vote requests and makes replies.

The voter is called when a resource is created or whenever its state changes. The voter is called VOTEW.

A vote request is one of the messages that CFSSRV sends to other CFS systems. See figure 2.

Requests, or votes, use the opcode .CFVOT. Vote replies use the opcode .CFREP. Replies may have optional data if the resource is defined as supporting optional data. Examples of optional data are:

1. EOF values for file resources
2. Directory allocation for directory resources.

	SCA HEADER
SCALEN	FLAGS, OPCODE, UNIQUE VOTE CODE
CFROT	ROOT CODE
CFQAL	QUALIFIER
CFTYP	TYPE OR ANSWER
CFDAT	OPT DATA
CFDT1	OPT DATA
CFDSTO	STR FREE COUNT FOR BITTABLE
CFDST1	TRANSACTION COUNT FOR BITTABLE

MR-16412

Figure 2 General Format of a CFS Message

In general, the two OPT data words have a value in one of the words and a "transaction count" in the other. The transaction count "age" the data so the voter can determine which reported value is most current. Note that in a large CFS configuration (more than two nodes), a voter can receive different values for the optional data based on the resource's history. Optional data processing is done through a callback address, and if that address is not provided in the packet, any received optional data is ignored.

The voter sends one vote request to each connected host and waits for the replies. The interrupt level part of the voter uses the vote count word in the resource packet to record replies. The process level part of the voter examines this field to determine when the vote is complete. Also, the flags word has a flag (HSHYES) that records a "no" reply to the vote.

If the CFS configuration changes during the vote, a "vote restart" flag is set in the resource block directing the voter to restart the vote. This is done instead of keeping state information about the vote. The restarted vote implicitly cancels the other vote.

The vote processor (the companion routine to the voter) uses the state information in the resource packet to determine how to reply. A reply may be:

1. Unconditional yes. This can result in a resource being released.

2. Unconditional no.

3. Timed no. This reply has an accompanying retry time stored in the hash packet.

4. Conditional or delayed yes.

The vote processor may use one or more call-back routine in the resource packet while deciding on the reply. For example, the decision to reply "delayed yes" is made by the callback routine when the vote is to be OKed. The co-routine may direct the vote processor to change the reply from "unconditional yes" to "delayed yes." This is typically the case for the file access token resource.

SCA Connection Manager

The connection manager is responsible for maintaining the CFS "host tables" and for interfacing to SCA. Its duties are to:

1. Connect to all extant hosts at start-up
2. Connect to hosts newly on-line
3. Close duplicate connections
4. Send messages and queue messages on credit failures
5. Receive messages
6. Resend on credit available

The connection manager maintains four tables that describe connections to other CFS hosts.

1. CFSHST contains the SCA connect ID
2. CFSHNM contains the processor serial number
3. CFHSTS contains the state of the connection
4. CFSNAM contains the DECnet node name.

While a connection is opened, or while a listener exists, the following interpretations apply.

1. If CFSHST = -1, this is a listener.
2. If CFSHNM = -1 and CFSHST<>0, this is a connect request waiting for a reply.

All other combinations indicate an unused entry.

The connection manager also maintains a table of previously seen hosts, OLDTAB. When a connection to another processor is lost, the processor's serial number is placed in OLDTAB. Then, each time a new connection is established, the processors search OLDTAB to determine whether each was ever connected to the other. If both processors believe that each has seen the other, then the reconnection cannot be honored because the CFS resources on the processors were not correctly coordinated. One of the processors, therefore, has to crash.

To allow for expected KLIPA reload conditions, the monitors "pause" whenever there is configuration change. During the pause time, no CFS resources can be acquired or change state. Therefore, if a reconnection happens during the pause time, neither system has to crash. Pause time is 15 seconds.

Initialization

CFS manages its own buffer pool that it creates in its own section, CFSSEC.

CFS creates the buffer pool by touching pages and then locking them into memory. Therefore, CFS initialization must proceed in process context.

The other side of CFS initialization is that done when SCA initializes. CFSSRV is called at CFSINI by SCA when SCA initializes. However, if CFSCSC has not yet been called, CFSINI cannot proceed to make connections and "join" the CFS network.

CFSJYN is the routine called to complete initialization and to join the CFS network. CFSJYN only proceeds the second time it is called. That is, it proceeds only after both CFSINI and CFSCSC were called. This makes sure that both SCA is initialized and the CFS buffer pool was created.

Extensions to Existing Services

Many of the monitor services were extended to allow for distributed control. In all cases, the code changes were inserting calls to routines in CFSSRV and, possibly, some reorganizing to eliminate a dependency on NOSKED as a system-wide interlock. For the most part, the existing monitor services know little about the workings of CFS except that it is a means to acquire a global interlock or global resource.

The bulk of changes were to PAGEM and PAGUTL (PAGEM in pre-release six monitors). The next most significant set of changes were to DSKALC to allow for managing the bit table. Finally, there are scattered changes in various monitor routines, including MSTR, DISC and MEXEC, and DIRECT.

File System Changes

Each OFN acquired some new state. The word, SPT02 has CFS-specific flags as well as a file access state field. This field reflects the value of the file's state in the CFS data base and is stored in SPT02 as well for convenience and efficiency.

Each opened file has at least one, and perhaps two, CFS resources assigned on the accessing processor. Also, each active OFN has a CFS access token assigned as a CFS resource. Therefore, an opened file has one or two CFS file resources and at least one access token associated with it.

The file resources are:

1. File Open Token. This is defined by a qualifier word of the XB address. This is specified whether the file is opened, frozen, or thawed.

2. Frozen Writer Token. This is defined by a qualifier word containing the XB address + <DSKAB_1>. This resource is present whenever a file is opened for frozen write access on this processor. This is an exclusive resource and therefore may be owned by only one processor.

The OFN resource is:

The file access token. Each active OFN has one such resource assigned. It is defined by a qualifier word containing the file XB + <DSKAB_3>. The access of the file access token defines the types of references that processes on the owning processor may make. Therefore, if the access token specifies "read only," then only read references may be made to the data in the file section.

The access specified in the access token is also present in SPT02, which is the field SPTST. This is so because many of the file and memory management routines (for example, the page fault handler) need to make sure that the access is proper for the operation in question. Finding the CFS resource each time is too costly, so the CFS state is also kept in a per OFN data table. To minimize the chance of error, only the routines in CFSSRV set and change SPTST.

The access of an OFN is set or changed by the routine CFSAWT or CFSAWP. The former sets the proper access but does not "reserve" the resource to this processor. The latter both sets the access and reserves it. CFSAWT is the most commonly used entry point. CFSAWP is used by code that has to reference often over time and does not want to incur another page fault if the access is removed during the interval. Examples of this are:

1. During file opening
2. Bit table lookup and modification.

Crucial CFS Storage Locations

Table 1 lists CSF storage locations. When doing dump analysis it may be useful to check any or all of the following storage locations.

CFS Voter Summary

VOTEW is the routine that sends out votes on behalf of the resource manager when a resource has to change state. It sends out votes, checks HSHVCT to see when all votes are in, and then returns to the caller with the result of the vote in HSHYES. If HSHYES is set, then a node vetoed the vote and access is not granted. HSHVCT is incremented at interrupt level by CFSRVT as vote packets arrive.

CFSRTV is the interrupt level routine that responds to incoming vote requests. It is the routine responsible for determining the kind of vote reply to return for this node. Table 2 lists possible responses. In Table 21 "we" refers to the host system and "he" refers to the remote system in a two-node cluster.

Table 1 Storage for CFS

Location	Description
CFSSTK	Stack pointer
CFSSVP	Place to save old stack
CFSSTR	Space for STR transaction values
CFSOFN	Points to where the OFN table resides
OFNCFS	Bit mask used by DDMP for dumping OFNs
VOTNUM	Vote dialogue unique number
CFHSHT	Points to hash table
CFSHCT	Count of hosts in this network
CFSHCM	Count of full voting hosts
SCAILK	Interrupt level interlock
SCAQ	Queue of pending messages
SCTAIL	End of the queue
CFSPCQ	Normal hash-free queue
CFSPQL	Long packet queue
VOTQ	Queue of voting packets
HSTSIZ--:10	Max hosts on the same Cl, full or reduced
CFSHST,HSTSIZ	Host table - CID or -1 if listener
CFSHNM,HSTSIZ	Host serial number
CFHSTS,HSTSIZ	Status word - connection state
CFNNAM,HSTSIZ*2	DECnet node name
CFSSMQ,HSTSIZ	Send queue (waiting for credit)
CFSSQT,HSTSIZ	End of queue
CFSSKC	Scheduler flag for testing CFS
CFSCMC	Connection management count
CFGFSP	Count calls to ASGRES
CFGTLG	Count long block allocates
CFGTSH	Count short block allocates
CFBRDS	Count of broadcast messages
CFSENT,HSTSIZ	Count of messages sent to node
CFRECV,HSTSIZ	Count of messages received node

Table 2 CFSRTV

CONDITION	RESPONSE IN .CFTYP
HSHLOK fails to find block	0 - unconditional OK
HSHDWT set (we have a commit)	1 - No, HSHNBT set for release
CFVUC set and HSHCODs unequal	1 - No, HSHNBT set for release
We are voting:	
HSHYES set (already got "no") and	
HSHUGD set and we have .HTPLH	0 - unconditional OK
HSHUGD set and:	
we have .HTPLH	0 - unconditional OK
we have .HTOEX	1 - No, ^D50 = retry
access types disagree	1 - No, ^D50 = retry
access types agree	0 - unconditional OK
We are voting:	
HSHDLY set (we have a commit)	1 - No, HSHNBT set for release
access types agree	0 - unconditional OK
access types disagree and:	
we have larger serial number	1 - No, HSHNBT set for release
he has larger serial number and:	
HSHUGD set but we have .HTPLH	0 - unconditional OK
HSHUGD not set (not upgrading)	0 - unconditional OK
he want .HTOEX	1 - No, HSHNBT set for release
access types disagree	1 - No, HSHNBT set for release
he doesn't want .HTOEX and agree	0 - unconditional OK
We are not voting (own it):	
we have .HTPLH and HSHKPH	0 - unconditional OK
we have .HTPLH and only	0 - unconditional OK (release)
if either access is .HTOPM	0 - unconditional OK
access types agree	0 - unconditional OK
he want .HTOEX, or we have .HTOEX or access types disagree:	
HSHCNT not zero (in use)	1 - No, HSHNBT set for release
HSHFCT still valid (fairness)	1 - No, difference = retry time
HSHCNT 0 and HSHFCT invalid	0 - unconditional OK (release)

Note that if the resource is to be released, the co-routine can change the reply code. Currently, file access tokens change the reply from 0 to -1 (delayed commit).

Figure 3 shows how CFS fits into the TOPS 20 Monitor V6.1.

Figure 4 shows the hierarchy of the MSCP Server within the System Communication Architecture.

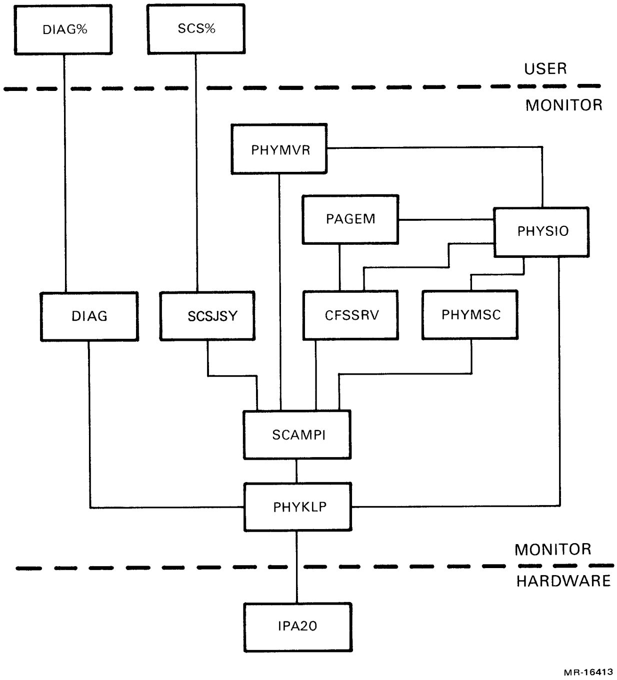


Figure 3 How CFS Fits into the TOPS 20 Monitor V6.1

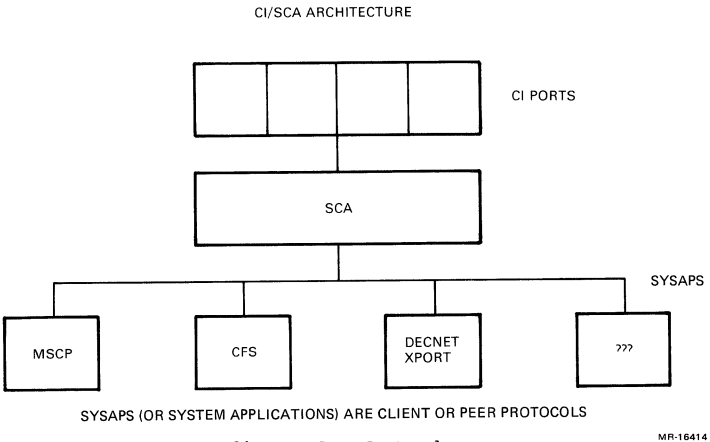


Figure 4 SYSAPS Are Client or Peer Protocols

Figure 5 shows how the major TOPS 20 monitor modules interact.

Figure 6 shows a page number look-up using the Mass Storage Control Protocol Server.

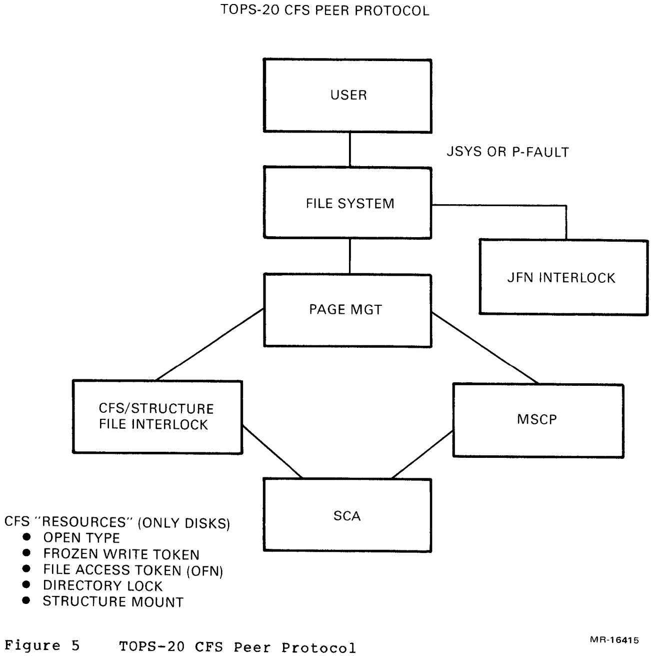


Figure 5 TOPS-20 CFS Peer Protocol

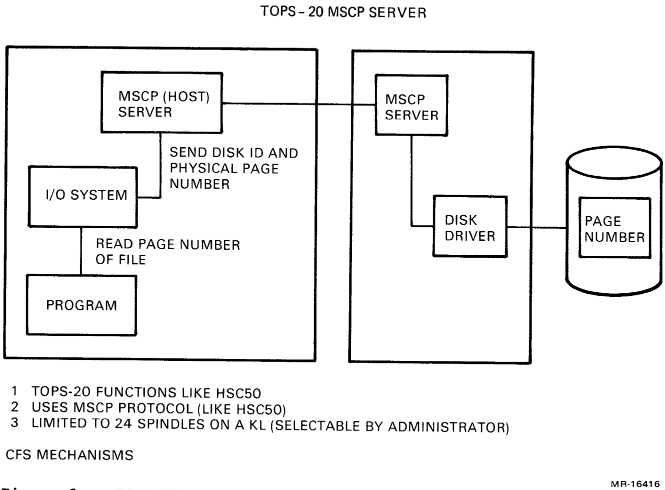


Figure 6 TOPS-20 MSCP

CFS MECHANISMS

Resource Manager

- CFS is a resource manager.
- Resources are locks that can represent any agreed upon entity.
- Data transfer is done by other TOPS-20 facilities, not CFS.
- Share updates are supported.
- Design allows for future resource definitions.

Implementation Devices

- Locks (72 bit resourced IDs)
- Resource table
- Resources modes
- Resource hold times
- Votes
- Broadcast messages
- Optional data
- Transaction counters
- SCA messages
- New microcode

To use a resource, such as a file, different ways of "locking" a resource are needed. Table 3 describes attributes of the different locking mechanisms.

Table 3 Attributes of Locking Mechanisms

TYPES	Directory Write	Allocation Table Structure	File Access ENQ
Content	Name (ID) Data	Owner Level	State
States	Owned Locally	Reside Locally, But Unowned	
Level	None	Read	Exclusive

Voting

- Purpose
  - To establish or change ownership of resource
- Consensus required
- Votes identified by node and vote number
- Voting may be restarted
  - Configuration Change
  - Timeout
- Answers from Votes
  - Yes

Figure 7 shows examples of the write token.

Transaction Numbers

- On broadcast messages
  - EOF changes
  - Bit table updates (disk allocation)
- Keeps all systems up-to-Date
- Algorithm
  - If

```
    Current-transaction-number < New-Transaction-Number.

  Then
    Use new data,
    Set current-transaction-number = new-transaction-number.

  Else
    Ignore data.
```

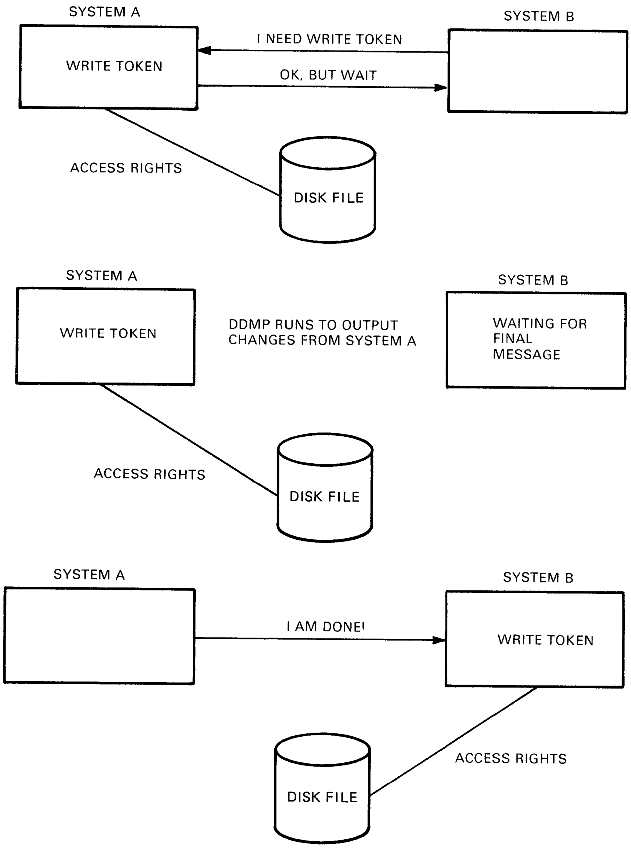


Figure 7 Write Token Examples

New KL MicroCode for CST "Write-Bit"

- Set by TOPS-20 to indicate page is writable.
- Microcode causes page fault if program writing into "unwritable page"
- CFS determines course of action

CFS Constraints

- ENQ on file
  - Exclusive on a single system
  - Will fail (no access) on other systems
- IPCF
  - No intersystem PID or delivery mechanism
- DBMS Files
  - Open with OF%DUD (don't update automatically)
  - Exclusive on a signal system
  - OPEN fails on other systems
- No CI DECnet

CFS Summary

- CFS is a fundamental requirement in a loosely coupled multiprocessor
- Natural extension of TOPS-20 file access mechanisms
  - No reprogramming
  - No retraining
- Constraints limit some muli-user applications to single system
- Homogeneous systems only
- Permits shared access to
  - HSC50 disks
  - Dual ported MASSBUS disks
  - Single ported MASSBUS disks via MSCP server
- Eases expansion from single system environment

TOPS20 V6.1 AC BLOCK 6 AND 7 INFORMATION

The following are the locations and the contents of Saved Data within ac blocks 6 and 7 after a crash.

AC BLK 6

Location	Contents
5	Save VMS register
10	Save BR register
11	Save SC and FE register
12	Save PFW (Page Fail Word)
16	Save AR register
17	Save ARX register

AC BLK 7

Location	Contents
0	Save Bad Parity Word
2	Save IOPF (IO Page Fail Word
3	Save IOP

NEW OPR INFORMATION

Displaying Disk Drive Status

You can display the status of the disk drives with the SHOW STATUS DISK-DRIVES command. With this command, you can specify three switches /ALL, /FREE, /MOUNTED. The default switch is /ALL. See the following.

Command	Description
OPR>SHOW STATUS DISK-DRIVES /ALL	/ALL switch displays both FREE and MOUNTED disk drives and their current status.
OPR>SHOW STATUS DISK-DRIVES /FREE	/FREE switch displays all disk drives that are FREE for mounting. See example 1.
OPR>SHOW STATUS DISK-DRIVES /MOUNTED	/MOUNTED switch displays those disk drives that are currently mounted and being accessed. See example 2.

Example 1 /FREE

The following is an example of the Show Status Disk-Drives command with the /FREE switch specified.

```
     OPR>SHOW STATUS DISK-DRIVE /FREE<RET>

     OPR>
                    12:11:15                   -- Disk Drive Status --
                              FREE DRIVES

DISK DRIVE INFORMATION                                DISK PACK INFORMATION
        Chan-Cont       Disk     Mount    Mount                    Usage
Type    Drive           Status   Status   Count       Name         Options

*RP06   0,    ,1        Avail    Offline
*RP07   1,    ,0        Avail    Free                 FIAT (1/1)
RP06    7,01 ,1         Avail    Free                 CHIP (1/1)
RA60    7,07 ,254       Avail    Offline
RA81    7,08,18         Avail    Free                 DRV18 (1/1)
RA81    7,08,5          Avail    Offline

                                 NOTE
             Channel 7 indicates CI channel
             (*) indicates potential external port.
```

Example 2 /MOUNTED

The following is an example of the Show Status Disk-Drive Command with the /MOUNTED switch specified.

```
OPR>SHOW STATUS DISK-DRIVE /MOUNTED<RET>

     OPR>
     12:11:28               -- Disk Drive Status --

                                 MOUNTED DRIVES

DISK DRIVE INFORMATION                                DISK PACK INFORMATION
        Chan-Cont       Disk     Mount    Mount                    Usage
Type    Drive           Status   Status   Count       Name         Options

RP06    0,    ,0        Avail    Mounted 0            DALE (1/1)
*RP06   0,    ,2        Avail    Mounted 1            AP20 (1/1)
RA81    7,07 ,5         Avail    Mounted 1            EXODUS (1/1)

                                 NOTE
             Channel 7 indicates CI channel
             (*) indicates potential external port
```

The output from the SHOW STATUS DISK-DRIVES command with the /ALL switch, displays the following information for disk drive and disk pack.

Disk Drive

1. Type -- The type of disk drive (RP04, RP06, RP07, RP20, RA60, or RA81). An asterisk (*) next to the type of drive indicates that the disk drive is potentially dual-ported.

2. Chan-Cont Drive -- The channel, controller, and drive number.

3. Disk Status -- The status of the disk drive (available/unavailable).

4. Mount Status -- The mount status of the disk drive (off-line/free).

5. Mount Count -- The mount count shows the number of users who requested a mount of the structure.

Disk Pack

6. Name -- The name of the structure on the drive and the number of packs in the structure.

7. Usage Options -- Any usage options, if these are specified.

Displaying Structure Status
You can display the status of structures with the SHOW STATUS STRUCTURE command. With this command, you can specify three switches, /ALL, /MOUNTED, /UNMOUNTED. STR = the name argument. The default switch is /ALL.

Command	Description
OPR>SHOW STATUS STRUCTURE /ALL	/ALL switch displays both mounted and unmounted structures and their current status.
OPR>SHOW STATUS STRUCTURE /MOUNTED	/MOUNTED switch displays all structures are currently mounted. See example 3.
OPR>SHOW STATUS STRUCTURE /UNMOUNTED	/UNMOUNTED switch displays all structures that are free for mounting. See example 4.
OPR>SHOW STATUS STRUCTURE STR:	The str: name argument displays information about a single structure. See example 5.

Example 3 /MOUNTED

The following is an example of the SHOW STATUS STRUCTURE command with the /MOUNTED switch specified.

```
OPR>SHOW STATUS STRUCTURE /MOUNTED <RET>
OPR>
20:14:04          -- Structure status --
                MOUNTED STRUCTURES
```

ATTRIBUTES:

Alias	Name	Mount State	Mount Count	File Count	Status	Access	Accounting
DALE	DALE	Mounted	0	0	Avail	Shared	## Primary Public Structure ##
AP20	AP2	Mounted	1	51	Avail	Shared	Foreign Regulated
EXODUS	EXODUS	Mounted	1	2	Avail	Shared	Foreign Regulated

Example 4 /UNMOUNTED

The following is an example of the SHOW STATUS STRUCTURE command with the /UNMOUNTED switch.

```
OPR>SHOW STATUS STRUCTURE /UNMOUNTED <RET>
OPR>
20:14:50          -- Structure status --

    UNMOUNTED STRUCTURES
```

ATTRIBUTES:

Alias	Name	Mount State	Mount Count	File Count	Status	Access		Accounting
AP20					Avail	Shared	Domestic	Regulated
DUMP					Avail	Shared	Domestic	Unregulated
FARK					Avail	Shared	Domestic	Unregulated
SIXOH					Avail	Shared	Domestic	Unregulated
WARPIG					Avail	Shared	Domestic	Unregulated
KL2136					Avail	Shared	Domestic	Regulated
BASIC					Avail	Shared	Domestic	Regulated
CBL74					Avail	Shared	Domestic	Regulated
DLM62					Avail	Shared	Domestic	Regulated

EXAMPLE 5 /STR

The following is an example of the SHOW STATUS STRUCTURE command with the str:name argument specified.

```
     OPR>SHOW STATUS STRUCTURE DALE: <RET>
     12:11:15        --Structure status

                              ATTRIBUTES:
                              File
 Alias  Name State   Count   Count  Status  Access     Accounting

 DALE   DALE Mounted 0              Avail   Shared     ## Primary Public
                                                          Structure ##

        Disk information pertaining to mounted structure

 DISK DRIVE INFORMATION                            DISK PACK INFORMATION

        Chan-Cont Disk     Mount    Mount                   Usage
     Type Drive     Status  Status   Count  Name            Options

     RP06 0,   ,0   Avail   Mounted  0      DALE (1/1)

     Users who have MOUNTED this structure:
     None.

     Users who are are ACCESSING this structure:
     KOVALCIN, WONG.

     Users who are CONNECTED to this structure:
     OPERATOR, KOVALCIN, WONG.
```

The output from the SHOW STATUS STRUCTURE command with the /ALL switch specifies the following.

1. Alias The alias name for the structure.

2. Name The physical name of the structure, if there is one.

3. Mount State The status of the structure (mounted or unmounted).

4. Mount Count The number of users who requested a mount of the structure.

5. File Count The file count of the structure.

6. Status The availability of the structure.

7. Access The access to the structure (SHARED/EXCLUSIVE, DOMESTIC/FOREIGN).

8. Accounting The accounting status of the structure (REGULATED/UNREGULATED).

The output from the SHOW STATUS STRUCTURE command with the str: name argument specified, displays the following.

1. Structure information about the single structure, including alias, name, mount status, mount count, file count, availability, access, and accounting.

2. Disk drive information for the drive(s) on which the structure is mounted. This includes drive type, channel controller drive, drive availability, mount status, mount count, structure name, number of packs in the structure, and usage options.

3. A list of users who have mounted, are accessing, or "have accessed" connection to the structure.

NOTE

If you need to remove information about a structure from the structure tables, use the UNDEFINE STRUCTURE command. For information on this command, refer to the TOPS-20 Operator's Command Language Reference Manual (AA-H600B-TM).

Examining Mount Requests in the Queue

When a user submits a request for a tape or disk mount, the request is placed in a queue where it waits for your response to either mount the tape or disk or delete the mount request. To examine the tape and disk mount queue at any time, do the following.

```
OPR>SHOW QUEUES MOUNT-REQUESTS
```

See example 6.

The output of the SHOW QUEUES MOUNT-REQUESTS command gives the following.

1. **Volume** The name of the tape or disk volume.
2. **Status** The status of the request.
3. **Type** The type of request (tape or disk).
4. **Dens** The density of the request.
5. **Write** The write access requested.
6. **Req #** The number of the request.
7. **Job#** The number of the user's job who made the request.
8. **User** The name of the user.
9. **Label status** the label status of a tape (labeled or unlabeled).

The status column displays the following.

1. **Waiting**, if a tape or disk request is waiting to be satisfied.

2. **A tape drive name** (for example, MTA3:), if the request is currently for a tape drive.

3. **Dismount**, if a tape or disk is being dismounted.

4. **Aborted**, if a tape mount request was deleted or aborted and the user has not given a dismount command to release the logical tape drive (MTn:).

Example 6 MOUNT-REQUESTS

The following is a example that uses the SHOW QUEUES MOUNT-REQUESTS command to display the current tape and disk requests in the queue.

```
     OPR>SHOW QUEUES MOUNT-REQUESTS<RET>

OPR>
     11:37:10             -- System Queues Listing --
Volume        Status    Type  Dens    Write    Req#    Job#    User

JCR            Waiting  Tape  Defa    Locked   1       143     JROSSELL
        Volume-set: JCR Tape is unlabeled
JCR            Waiting  Tape  Defa    Locked   2       143     JROSSELL
          Volume-set: JCR Tape is labeled
There are 2 requests in the queue

     OPR>
```

Creating a Structure During Timesharing

During timesharing, you can create a structure by running CHECKD. For more information about the CHECKD program, refer to the TOPS-20 CHECKD Specification (AA-V918B-RM).

NOTE

When you create a structure, you destroy any information that was on the disk packs in the structure. If you want to save information from an old structure, run DUMPER to save the files before you create the new structure on the same disk packs.

Procedures for Running CHECKD.

1. Get the number of formatted disk packs needed for the structure.

2. Place the packs on any available disk drives and put the drives on-line.

3. Type PUSH to get the OPR prompt.

4. Run CHECKD. The following is an example of the questions asked by CHECKD.

```
@ena
$checkd
CHECKD>create (NEW FILE SYSTEM FOR) DALE:
Enter alias: dale
How many units in this structure? 1
DECIMAL Channel, Controller, and Unit numbers for logical
unit 0:1,0,1
Number of pages to allocate for swapping? 10000
Number of pages to allocate for the Front End File
System? 950
Owner name? SYSTEM
Is this a system structure for start-up? Yes
Serial number of CPU started from this structure: 2102

[DALE: Mounted as DALE:]

[Dismounting structure - DALE]

CHECKD>Exit
$DISABLE
```

5. OPR>MOUNT STRUCTURE str: str=DALE

NOTE
If this structure is to be used on a system other than a KL-2060, perform steps 6 through 8 in addition to the full CHECKD procedure. Otherwise, go to step 9.

6. Run CHECKD.

7. CHECKD>DISABLE LARGE-DIRECTORIES

8. CHECKD>EXIT

9. $DISABLE.

10. Type POP to return to OPR command level.

NOTE
If CHECKD is a "subjob" of PTYCON at your installation, you need only to CONNECT to the CHECKD subjob and perform steps 4 through 5 above.

If your installation is part of a CFS configuration, CHECKD may not be able to set the structure exclusive. If this happens, you must dismount the structure from the other systems using it. Use the dismount procedures for CFS systems in Massbus Disk Cluster Mode Diagnostic Procedures section.

KLAD NAMING CONVENTIONS

Due to the operation of Common File System (CFS) the system structure of each KL host in the cluster cannot have the same name in its home blocks as the other system structures. For remote support requirements, the following naming convention is defined for multiple KLAD usage in a cluster environment. (That is, when bringing up a entire cluster under KLAD operation.)

Any KLAD intended for use in a cluster must have its home blocks changed to the following.

1st KLAD - PS:
2nd KLAD - PS1:
3rd KLAD - PS2:
4th KLAD - PS3:

Although TOPS20 version 6.1 supports only a two KL host cluster, future releases will support more hosts within the cluster.

MIC files PS1.MIC, PS2.MIC, PS3.MIC are in the <F-S> directory and should be used to change the name in the home blocks.

NOTES

A reboot of the KLAD monitor is needed to incorporate the change in the home blocks.

A carriage return at the BOOT prompt to load the monitor no longer works. You must type the structure name at the BOOT prompt. (That is PS1:return). The monitor, after it starts, also asks for the name of the system structure.

Clearly mark the pack and the pack cover after the name has been changed.

TOPS20 V6.0 AND V6.1 CI20 DISABLING INFORMATION

TOPS20 V6.0 CI20 Disabling Information at System Start-Up

The following procedure reloads the TOPS20 Version 6.0 Monitor on a KL10 that TOTALLY ignores (both user-mode diagnostically and from an on-line functional view) the CI20 option.

NOTE

This procedure starts at the BOOT prompt (after you have gone through the KLI dialog)!

```
Command                                   Description
--------------------------------------------------------------------------
BOOT V10.0(201)

BOOT>/e <CR>                              ;TELL BOOTS TO LOAD MONITR.EXE
[BOOT: LOADING] [OK]                      ;AND ENTER EDDT
EDDT
klpini+1/   CONI 574,T1   ret <CR>        ;REPLACE THE CONI WITH A RET
sysgo$go                                  ;START-UP MONITOR AT SYSGO

[PS MOUNTED]
--------------------------------------------------------------------------
```

NOTE

When you start up a KL10 as above, a system reload has to be scheduled to return the CI20 option.

TOPS20 V6.0 CI20 Disabling within Monitr.Exe
The following procedure creates a copy of the TOPS20 Version 6.0 Monitor that TOTALLY ignores (both user-mode diagnostically and from an on-line functional view) the CI20 option.

```
Commands                                  Description
------------------------------------------------------------------------
@enable <CR>                              ;ENABLE PRIVS
$get ps:<system>monitr.exe <CR>           ;GET A COPY OF THE MONITOR
$ddt <CR>                                 ;CALL IN DDT
DDT
klpini+1/   CONI 574,T1   ret <CR>        ;REPLACE THE CONI WITH A RET
^Z                                        ;CONTROL Z TO GET OUT OF DDT
$save monitr-noci20.exe <CR>              ;SAVE IT WITH A NEW NAME
 MONITR-NOCI20.EXE.1 Saved
$
```

NOTE
When you start up a KL10 with this "NOCI20" monitor, a system reload (using MONITR.EXE) has to be scheduled to return the CI20 option.

TOPS20 V6.1 CI20 Disabling at System Start-Up
The following procedure reloads the TOPS20 Version 6.1 Monitor on a KL10 that TOTALLY ignores (both user-mode diagnostically and from an on-line functional view) the CI20 option.

NOTE
This procedure starts at the BOOT prompt (after you have gone through the KLI dialog)!

```
Command                                   Description
------------------------------------------------------------------------
BOOT V11.0(306)

BOOT>/e <CR>                              ;TELL BOOTS TO LOAD MONITR.EXE
[BOOT: LOADING] [OK]                      ;AND ENTER EDDT
EDDT
noklip  0   1  <CR>                       ;REPLACE THE ZERO WITH A ONE
sysgo$go                                  ;START-UP MONITOR AT SYSGO

[UPS MOUNTED]
```

NOTE
When you start up a KL10 as above, a system reload has to be scheduled to return the CI20 option.

TOPS20 V6.1 CI20 Disabling Within Monitr.Exe
The following procedure creates a copy of the TOPS20 Version 6.1 Monitor that TOTALLY ignores (both user-mode diagnostically and from an on-line functional view) the CI20 option.

```
Command                                   Description
------------------------------------------------------------------------
@enable <CR>                              ;ENABLE PRIVS
$get ps:<system>monitr.exe <CR>           ;GET A COPY OF THE MONITOR
$ddt  <CR>                                ;CALL IN DDT
DDT
noklip/   0   1  <CR>                     ;REPLACE THE ZERO WITH A ONE
^Z                                        ;CONTROL Z TO GET OUT OF DDT
$save monitr-noci20.exe <CR>              ;SAVE IT WITH A NEW NAME
MONITR-NOCI20.EXE.1 Saved
$
```

NOTE
When you start up a KL10 with this "NOCI20" monitor, a system reload (using MONITR.EXE) has to be scheduled to return the CI20 option.

CI20 USER MODE DIAGNOSTICS PROCEDURES UNDER TOPS-20

Cluster Rules for Setting CI20 to Unavailable or Available

1. A CI20 option may be Set Unavailable under any cluster configuration at any time. Make sure you have permission or acknowledgment before you set the CIA20 to Unavailable.

2. You can only use the Set Available command to the CI20 option when it is the only host up and "running" ("running" means TOPS20 monitor) in that cluster. Failure to do this cause the system to crash. See step 6 in the next procedure.

3. If rule 2 is not true (that is, if more than one host is "running"), and you want to set the CI20 option available, schedule a system reload on the CPU that has the CI20 option Set Unavailable.

Use the following procedure to set the CI20 for User Mode Diagnostics (with the CI20 in the Unavailable State).

1. User Mode Diagnostic Restrictions -- First, make sure you set the CI port to unavailable before running the diagnostic. Type the following.

   OPR>SET PORT CI UNAVAILABLE (CR)

   WARNING

   The program OPR notifies you (with a message) as to which disk structures need to be dismounted. Have operations dismount those structures. Or, see Massbus Disk Cluster Mode Diagnostic section for dismount procedures. When the structures are dismounted you can respond to the message number with a proceed response. Type the following.

   OPR>RESPOND # PROCEED (CR)
   (# = The message number) OPR then sets the port to unavailable by stopping the CI20.

2. Mount the KLAD pack via OPR (Privileged Job). Load the KLAD pack on an available drive, and lock the port switch (if drive is dual ported between systems in a CFS configuration) on the system where you will run the CI20 diags. Type the following.

   OPR>MOUNT STRUCTURE KLAD:/STRUCTURE ID:PS: or PSx:
   (where"x" = 1, 2, or 3, derived from KLAD20 naming convention) OPR mounts your structure and calls it KLAD.

   OPR>EXIT (CR)
   OPR exits and returns to the monitor prompt.

3. Mount the KLAD to your user job. Connect the KLAD to the <2-Diagnostic> area and run D20MON. Type the following.

```
$MOUNT STRUCTURE KLAD: (CR)
$CONNECT KLAD:<2-DIAGNOSTICS> (CR)
$RUN D20MON (CR)
```

NOTE

DFPTA.A10 and DFCIA.A10 both run under User Mode with the supervision of D20MON.

DFCIA.A10 requires an additional file called CI20.ULD (CI20 diagnostic microcode file).

DFCIA loads CI20.ULD via D20MON'S control when you start the diagnostic.

If you wish to exit from D20MON prompt, type CTRL C twice. This gets you back to the Monitor prompt.

Also DFCIB.EXE (KL's exerciser) and DFCIC.EXE (KL's responder) run under user mode while the CI20 is active and serving general timesharing (the CI20 being in the Available state). The HSC50'S responder is built into CRONIC (HSC50's operating system). Both DFCIB.EXE and DFCIC.EXE do NOT run under D20MON supervision.

4. After running the diagnostic, dismount the KLAD pack from your user job by typing:

   $DISMOUNT STRUCTURE KLAD: (CR)

5. Start up OPR and dismount KLAD from the system.

   OPR>DISMOUNT STRUCTURE KLAD: REMOVAL (CR)

   - Spin down the drive and remove the KLAD pack.
   - Return the port switch to its original position.

6. Set the CI20 Port to Available after running user-mode Diagnostics from a Privileged Job. See Cluster Rules.

   OPR>SET PORT CI AVAILABLE (CR)

   The following is an example of the question asked by the program OPR after setting the Port CI to Available.

   <6> -- Port Available Procedure Notice --

Is there another TOPS-20 system currently running on the CI?

OPR>RESPOND <number> "Yes or No"

A "yes" response aborts the SET PORT CI AVAILABLE command so that you can schedule a system reload.

A "no" response either:

1. Allows the port to be reloaded and started.

2. Or if you answered "no" and you were lying and there really was another TOPS20 Monitor "running" within the Cluster then the following happens:

   OPR>res 6 no (CR)

   %DECSYSTEM-20 NOT RUNNING (with a CRFRCN BUGHLT)

DFCIB (KL-EXERCISER) INFORMATION

The following information is how to run the KL-EXERCISER (DFÇIB.EXE) with the KL-RESPONDER (DFCIC.EXE) between two KL10s configured in a cluster (linked via the same CI).

DFCIB.EXE Rules

1. DFCIB can exercise other CI nodes whether they are KL10 or HSC50.

2. DFCIB only exercises these other nodes if they were selected via the SELECT command.

3. You can select more than one CI node at a time, BUT you can NOT select the KL10 node from which you are running the program DFCIB.

4. You can start up a second DFCIB on the other KL10 in the cluster and then select the opposite KL10 from each DFCIB running on each host.

5. DFCIC.EXE is the KL's Responder program. After it is started and while it is running, the program does NOT print out messages.

6. The HSC50 has its responder built into Cronic (HSC50's operating system).

7. DFCIB use the COMMAND JSYS, so command prompting is available as it is under TOPS20's EXEC.

8. Certain tests are designed to run only with a KL10 and others to run with an HSC50. If you request ALL TESTS to be run against the selected nodes, expect to see a message stating that a test may not run against a selected node.

NOTE
The following is a description of the HELP text from DFCIB.HLP. It is in a different format than you would see on the system. For your viewing convenience see the following table.

Table 4 DFCIB Commands

Command	Description
*ACCEPT	Used only for writing your own tests. It causes a connection to be accepted from another exerciser program.
*BUILD	Used only for writing your own tests. It allows the building of CTP packets for later transmission using the SEND command.
*CLEAR	The complement of the SET command. For more information, see either CLEAR LOGGING or CLEAR SWITCH.
*CLEAR LOGGING	Used to close the log file, which is opened by the SET LOGGING command. The log file receives a copy of all output directed to the terminal.

Table 4 DFCIB Commands (Cont)

*CLEAR SWITCH

Used to clear one or more of the switches. It is the complement of the SET SWITCH command. The current switch settings are displayed with the SHOW SWITCHES command. The switches are:

BELL	Ring the terminal bell on errors.
HALT	Stop testing when any error is detected.
PALL	Print all bytes in buffer (TYPE command).
TRACE	Print trace messages as each test is executed.

*CONNECT

Used only for writing your own tests. It causes a connection to be made to the last node specified in the SELECT command.

*DDT

Used only for debugging the diagnostic. It causes execution to be transferred to DDT if it is loaded. If DDT is not loaded it to load it.

*DESELECT

Used to disable selection of a node for testing. It is followed by one or more decimal node numbers, separated by commas. To see what nodes are selected, use the SHOW SELECTED-NODES command.

*DISCONNECT

Used only for writing your own tests. It causes the connection created by the CONNECT command to be broken.

*DESELECT

The complement of the SELECT NODE command. It is followed by one or more decimal node numbers separated by commas. To see what nodes are selected, use the SHOW SELECTED-NODES command.

*HELP

The HELP command followed by question mark (?) provides the user a list of the entries in the on-line help file. HELP followed by the subject provides the user a short message on the selected subject. Help with no subject specified provides the user a help message on using the HELP command.

*LISTEN

Used only for writing your own tests. It causes the program to listen for a CTP connection from any node.

*MAP-BUFFER

Used only for writing your own tests. It causes a map of an internal buffer of the number of bytes specified in the command.

*PAUSE

Normally used only for a writing your own tests. It causes the program to "sleep" for the number of seconds furnished in the command.

Table 4 DFCIB Commands (Cont)

*QUIT	Causes the diagnostic to exit.
*READ	Used only for writing your own tests. It is followed by a keyword, DATAGRAM or MESSAGE, indicating the read action.
*REQUEST-DATA	Normally used only for writing your own tests. It is used to request DMA transfer data from either a RESPONDER-NODE or an EXERCISER-NODE, as specified in the command.
*REQUEUE	Normally used only for writing your own tests. It causes a buffer used by the READ command to be freed for later use.
*RETURN-MESSAGE	Used only for writing your own tests. It causes the diagnostic to return a message as would a responder program.
*RUN	Used to start the execution of a test or group of tests. The format of the RUN command is: RUN /PASSES:n test/ITERATIONS:n , test/ITERATIONS:n The optional /PASSES switch must be specified before any test name, and specifies how many passes of the test to run. The default number of passes if /PASSES isn't specified is 1. A test name or test script comes next, separated by commas. Each test may be executed a number of times using the optional /ITERATIONS switch. The default number of interations is 1, except for the EXERCISER and TST99, for which the default iteration count is 100. A RUN command with no arguments is used to rerun the last successfully entered RUN command. The tests are stored in the run table, which can be displayed using the SHOW RUN-TABLE command. A sample run command would be DFCIB>RUN /PASSES:100 BASIC/ITER:10, MESSAGE/ITER:5, COUNTER This runs 100 passes, where each pass consists of 10 iterations of the Basic Connect/Disconnect Tests, five iterations of the Message Tests, followed by one iteration of the Counter Test.

Table 4 DFCIB Commands (Cont)

For information on RUN command qualifiers, refer to any of the following later in this table.

ALL-TESTS	DATA-TESTS
BASIC-TESTS	DEFAULT
BUFFER-TESTS	EXERCISER
COUNTER-TEST	MESSAGE-TESTS

For information on RUN command test qualifiers, refer to any of the following later in this table.

TST01	TST02	TST10	TST11
TST12	TST13	TST14	TST30
TST31	TST32	TST33	TST40
TST50	TST51	TST52	TST53
TST54	TST55	TST56	TST57
TST58	TST59	TST80	TST99

*SELECT

Used to select a node for CTP testing. It is followed by node numbers separated by commas. The selected nodes are displayed with the SHOW SELECTED-NODES command.

*SET

Used to activate various program options. For more information see any of the following.

SET LOGGING
SET PATH-SELECTION
SET SWITCH
SET TIME-OUT
SET TYPE-OUT-LINE-LIMIT

*SET LOGGING

This command is followed by a filename into which all terminal output is placed. The default filename is DSK:DFCIB.LOG. To close the log file, either exit the diagnostic with the QUIT command or use the CLEAR LOGGING command.

*SET PATH-SELECTION

Used to select either Path A, Path B, or automatic path selection for all CTP packets sent over the CI. The default is to use automatic path selection, which lets the monitor pick the path to use, causing CTP packets to be split more or less evenly across both paths.

*SET SWITCH

Used to set one or more of the switches. It is the complement of the CLEAR SWITCH command. The current switch settings are displayed with the SHOW SWITCHES command. The switches are:

BELL	Ring the terminal bell on errors.
HALT	Stop testing when any error is detected.
PALL	Print all bytes in buffer (TYPE command).
TRACE	Print trace messages as each test is executed.

Table 4 DFCIB Commands (Cont)

*SET TIME-OUT	Followed by the maximum number of seconds to wait for any event to complete. The default time out is five seconds. The command is used to disply the current time out value.
*SET TYPE-OUT-LINE-LIMIT	Used to set the number of lines that are displayed with the TYPE command. The TYPE command is only used for writing your own tests.
*SHOW	Used to display various program settings. For more information see SHOW followed by any of the following. ALL CONNECTION-DATA-OF-ALL-NODES COUNTERS EVENT-QUEUE ID-OF-CONNECTED-NODE LOCAL-NODE-NUMBER MAPPED-BUFFER-NAME MINIMUM-BUFFER-SIZED PATH-STATUS-OF-NODE POLL-STATUS-OF-CONNECTED-NODE RUN-TABLE SELECTED-NODES STATUS-OF-CONNECTED-NODE SWITCHES TIME-OUT
*SHOW ALL-PROGRAM-PARAMETERS	Is the equivalent of the following SHOW commands. LOCAL-NODE-NUMBER PATH-SELECTION SELECTED-NODES TIME-OUT RUN-TABLE SWITCHES The other SHOW commands have to do with writing your own test or getting status of this node or other nodes on the CI. For more information, see SHOW followed by any of the commands.
*SHOW CONNECTION-DATA-OF-ALL-NODES	Requests connect data from all other nodes on the CI and display various node-specific data, such as software type and versions, buffer sizes, and so on.
*SHOW COUNTERS	Causes the CI port counters to be read and displayed for this node on the CI.
*SHOW EVENT-QUEUE	Used only when writing your own tests and shows the status of the event queue for the connection made with the ACCEPT or CONNECT commands.
*SHOW ID-OF-CONNECTED-NODE	Used only for writing your own tests and shows the connect ID of the node connected with the CONNECT or LISTEN commands.
*SHOW LOCAL-NODE-NUMBER	Shows the CI node number assigned to the system the diagnostic is running on.

Table 4 DFCIB Commands (Cont)

*SHOW MAPPED-BUFFER-NAME	Used only for writing your own tests and is used to display the buffer name returned by the MAP-BUFFER command.
*SHOW MINIMUM-BUFFER-SIZES	Shows what monitor determines is the minimum buffer sizes for datagram and messages.
*SHOW PATH-SELECTION	Displays which paths were selected with the SET PATH-SELECTION command.
*SHOW PATH-STATUS-OF-NODE	Followed by a node number, shows what monitor determines is the current status of paths to that node.
*SHOW POLL-STATUS-OF-CONNECTED NODE	Used only for writing your own tests. It shows what monitor knows as the poll status of the currently connected node.
*SHOW RUN-TABLE	Displays which tests or qualifiers were run with the last RUN command. This list will be used if another RUN command is given with no tests/qualifiers specified. See RUN for more details.
*SHOW SELECTED-NODES	Displays which nodes on the CI were selected for testing. The SELECT NODE command selects nodes for testing and the DESELECT NODE command removes nodes from testing.
*SHOW STATUS-OF-CONNECTED-NODE	Used for writing your own tests. It displays the status of the connection made with the CONNECT or LISTEN commands.
*SHOW SWITCHES	Used to display the switch settings. The SET SWITCH command is used to set one or more of the switches. The CLEAR SWITCH command is used to clear switches. The switches are: BELL Ring the terminal bell on errors. HALT Stop testing when any error is detected. PALL Print all bytes in buffer (TYPE command). TRACE Print trace messages as each test is executed.
*SHOW TIME-OUT	Displays the time in seconds to wait for any event to complete. The default time out is five seconds. This command is used to change the current time out value.

Table 4 DFCIB Commands (Cont)

*TAKE	Used to cause a file of commands to be passed to the program. The default filename is DSK:DFCIB.CMD. For example, if the following commands were in a file called TEST15.CMD:
	SELECT NODE 15 SET LOGGING TEST15.LOG RUN/PASS:3 BUFFER-TESTS/ITER:10 ,COUNTER-TEST QUIT
	The command TAKE TEST15 causes node 15 to be selected, a log file called TEST15.LOG to be opened, three passes of the specified tests to be run, and the program to exit.
*TYPE	Used for writing your own tests. It is followed by one of the following to display the contents of program buffers.
	CTPPKT MBUF4 DBUF1 MBUF5 DBUF2 MBUF6 DBUF3 RETURNED-BUFFER DBUF4 SAVCTP DBUF5 SAVREQ DBUF6 SAVRSP MAPPED-BUFFER SAVSCS MBUF1 SCSCMD MBUF2 SCSEVT MBUF3 SCSRSP
*WAIT	Used only for writing your own tests. It is used to wait for one of the following.

CONNECTION-STATUS STATE, one of the following.

ACCEPT-REQUEST-SENT	CLOSED
CONNECT-REQUEST-RECEIVED	CONNECT-REQUEST-SENT
CONNECT-RESPONSE-RECEIVED	CONNECTION-IS-OPEN
DISCONNECT-REQUEST-RECEIVED	DISCONNECT-REQUEST-SENT
DISCONNECT-RESPONSE-RECEIVED	LISTENING
REJECT-REQUEST-SENT	WAITING-FOR-DISCONNECT-RESPONSE

CONNECTION-STATUS FLAG, of one of the following.

DATAGRAM-AVAILABLE-FLAG	DMA-TRANSFER-COMPLETE
EVENT-PENDING-FLAG	MESSAGE-AVAILABLE-FLAG

EVENT-STATUS, one of the following.

CONNECT-TO-LISTEN	CONNECTION-WAS-ACCEPTED
CONNECTION-WAS-REJECTED	CREDIT-IS-AVAILABLE
LITTLE-CREDIT-LEFT	MAINT-DATA-XFER-COMPLETE
MESSAGE-DATAGRAM-SEND-COMPLETE	NODE-CAME-ONLINE
NODE-WENT-OFFLINE	OK-TO-SEND-DATA
PORT-BROKE-CONNECTION	REMOTE-INITIATED-DISCONNECT
VC-BROKEN	

Run Command Qualifiers*

*DEFAULT	Runs the same tests as the ALL-TESTS qualifier, except it selects all nodes available for testing first. See ALL-TESTS for a list of the tests run by that qualifier.

---

* The remaining commands in this table are RUN command qualifiers. They are used with the RUN command in this format: RUN DEFAULT.

Table 4 DFCIB Commands (Cont)

*ALL-TESTS Runs the following tests.

TST01 Connect/Disconnect Test
TST02 Function Set Request/Response Test
TST10 Basic Message Test
TST11 Message Test Count 4, Four Data Types
TST12 Message Test Repeat 3, Four Data Types
TST13 Message Test Delay 4, Four Data Types
TST14 Message Test Size 1 to Maximum, Four Data Types
TST30 Basic Datagram Test
TST31 Datagram Test Size 4, Four Data Types
TST32 Datagram Test Repeat 3, Four Data Types
TST33 Repeat and Delay 4, Four Data Types
TST40 Datagram Test Under Message Service
TST50 Buffer Map and Buffer Unmap Test
TST51 Buffer Map, Controller Read, Buffer Unmap
TST52 Buffer Mpa, Buffer Move, Buffer Unmap
TST53 Buffer Move (Delay 4)
TST54 Buffer Move (Repeat 3)
TST55 Buffer Move (Variable Length)
TST56 Buffer Move (Size 1)
TST57 Buffer Move (Multiple 1)
TST58 Buffer Move (Exerciser-Responder Write) HSC50
TST59 Buffer Move (Exerciser-Responder Write) TOPS-10/20/VMS
TST80 Read Counter Test

*BASIC-TESTS Runs the following tests.

TST01 Connect/Disconnect Test
TST02 Function Set Request/Response Test

*BUFFER-TESTS Runs the following tests.

TST50 Buffer Map and Buffer Unmap Test
TST51 Buffer Map, Controller Read, Buffer Unmap
TST52 Buffer Map, Buffer Move, Buffer Unmap
TST53 Buffer Move (Delay 4)
TST54 Buffer Move (Repeat 3)
TST55 Buffer Move (Variable Length)
TST56 Buffer Move (Size 1)
TST57 Buffer Move (Multiple 1)
TST58 Buffer Move (Exerciser-Responder Write) HSC50
TST59 Buffer Move (Exerciser-Responder Write) TOPS-10/20/VMS

*COUNTER-TEST Runs the following test.

TST80 Read Counter Test

*DATA TESTS Runs the following tests.

TST30 Basic Datagram Test
TST31 Datagram Test Size 4, Four Data Types
TST32 Datagram Test Repeat 3, Four Data Types
TST33 Repeat and Delay 4, Four Data Types
TST40 Datagram Test Under Message Service

*MESSAGE-TESTS Runs the following tests.

TST10 Basic Message Test
TST11 Message Test Count 4, Four Data Types
TST12 Message Test Repeat 3, Four Data Types
TST13 Message Test Delay 4, Four Data Types
TST14 Message Test Size 1 to Maximum, Four Data Types

*TST01

TST01 Connect/Disconnect Test: Connects to the selected node and verifies proper status (Event Pending, Connection Accepted, and Connection Open). The test then disconnects from the selected node and verifies proper status (Connection Closed).

*TST02

TST02 Function Set Request/Response: Connects to the selected node and verifies proper status. A Function Set Request is sent and proper status verified. The Response message is read and verified. The test then disconnects from the selected node and verifies proper status.

Table 4 DFCIB Commands (Cont)

*TST10

TST10 Basic Message Test: Connects to the selected node and verifies proper status. A Generate Message Request with a word count of 0 is sent and proper status verified. The Response message is read and verified. The test then disconnects from the selected node and verifies proper status.

*TST11

TST11 Message Test Count 4, Four Data Types: For each data type, a Generate Message Request with a word count of 4 and buffer fill data of one of the data types is sent and proper status verified. The Response message is read and verified. The data field is verified.

*TST12

TST12 Message Test Repeat 3, Four Data Types: For each of the data types, a Generate Message request with a word count of 4 and buffer fill data of the data type and a repeat count of 3 (which generates four response messages) is sent and proper status verified. Each response message is read and verified. The data field is verified.

*TST13

TST13 Message Test Delay 4, Four Data Types: For each data type A Generate Message request with a word count of 4 and buffer fill data and a delay count of 4 (2 seconds) is sent and proper status verified. The response message is read and verified. The data field is verified as 4 bytes of 375.

*TST14

TST14 Message Test Size 1 to Maximum, Four Data Types: For each data type, a Generate Message Request with a variable byte count of 1 to maximum supported in the responder and buffer fill data is sent and proper status verified. The Response messages are read and verified. The data field is verified as the correct number of bytes of fill data.

*TST30

TST30 Basic Datagram Test: Connects to the selected node and verifies proper status. A Generate Datagram Request with a word count of 0 is sent and proper status verified. The Response Datagram is read and verified. The test then disconnects from the selected node and verifies proper status.

*TST31

TST31 Datagram Test Size 4, Four Data Types: For each data type, a Generate Datagram Request with a word count of 4 and buffer fill data is sent and proper status verified. The Response Datagram is read and verified. The data field is verified.

*TST32

TST32 Datagram Test Repeat 3, Four Data Types: For each data type, a Generate Datagram Request with a byte count of 4 and buffer fill data and a repeat count of 3 (which generates four response datagrams) is sent and proper status verified. Each Response Datagram is read and verified. The data field is verified.

*TST33

TST33 Datagram Test Repest and Delay 4, Four Data Types: For each data type, a Generate Datagram Request with a word count of 4 and buffer fill data and a delay count of 4 (2 seconds) is sent and proper status verified. The Response Datagram is read and verified. The data field is verified as 4 bytes of 375.

*TST40

TST40 Datagram Test Under Message Service: A Generate Datagram Request under Message Service with a word count of 26, buffer fill data of 201, a delay count of 4 (2 seconds), and a repeat count of 1 is sent and proper status verfied. The Reponse Datagram is read and verified. The data field is verified as 4 bytes of 201.

Table 4 DFCIB Commands (Cont)

*TST50

TST50 Buffer Map and Buffer Unmap Test: A Buffer Map Request for a real buffer with a buffer fill of 377 and a buffer length of 576 bytes is sent and proper status verified. The response message is read and verified. A Buffer Unmap Request for the same buffer is sent and proper status is verified. The Response Message is read and verified.

*TST51

TST51 Buffer Map, Controller Read, Buffer Unmap: For each data type, a Buffer Map Request for a real buffer with a buffer fill and a buffer length of four bytes is sent and proper status verfied. The Response Message is read and verified. An internal buffer is mapped (in the exerciser), and bytes of 125 are written into the buffer. Data is read, and the buffer is verified to contain bytes of fill data. The internal buffer is unmapped. A Buffer Unmap Request for the responder buffer is sent and proper status is verified. The Response Message is read and verified.

*TST52

TST52 Buffer Map, Buffer Move, Buffer Unmap: A Buffer Map Request for a real buffer with a buffer fill of 252 and a buffer length of 4 bytes is sent and proper status verified. The Response Message is read and verified. An internal buffer is mapped in the exerciser and bytes of 125 are written into the buffer. A Move Buffer Command is issued for a responder write of 4 four bytes with a delay, a repeat count, a packet multiplier and a packet size of 0 is sent and proper status verified. The Response Message is read and verified. The buffer is verified to contain bytes of 252. The internal buffer is unmapped. A Buffer Unmap Request for the same buffer is sent and proper status is verified, and the Response Message is read and verified.

*TST53

TST53 Buffer Move (Delay 4): This test is identical to TST53 except buffer fill data of 371 and delay count of 4 (2 seconds) is used.

*TST54

TST54 Buffer Move (Repeat 3): This test is identical to TST53 except buffer fill data of 370 and repeat count of 3 (generating 4 buffer moves) is used.

*TST55

TST55 Buffer Move (Variable Length): The following is performed starting with a buffer length of 4 and incrementing the buffer length by four until a buffer length of 1024 is reached. A buffer map request for a real buffer with a buffer fill of byte pair and a buffer length of 1024 bytes is sent and proper status verified. The Response Message is read and verified. Map an internal buffer 1024 bytes in length fill it with bytes of 125. Issue a Move Buffer Command for a responder write of variable length bytes with a delay, a repeat count, a packet multiplier and a packet size of 0. Proper status is verified. The Response Message is read and verified. Verify the buffer contains byte pair data. A Buffer Unmap request for the same buffer is sent and proper status is verified. The response message is read and verified. The internal buffer is unmapped.

*TST56

TST56 Buffer Move (Size 1): A buffer map request for a real buffer with buffer fill of 366, a buffer length of 1024 bytes and a packet size of 1 is sent and proper status verified. The response message is read and verified. An internal buffer is mapped and filled with bytes of 125. A Move Buffer command for a responder write of 1024 bytes with a delay, repeat count, a packet multiplier of 0 and a packet size of 1 is sent and proper status verified. The Response Message is read and verified. The buffer is verified to contain 366. A Buffer Unmap request for the same buffer is sent and proper status is verified. The response message is read and verified. The internal buffer is unmapped.

Table 4 DFCIB Commands (Cont)

*TST57

TST57 Buffer Move (Multiple 1): This test is identical to TST56 except a packet multiple of 1 is used.

*TST58

TST58 Buffer Move (Exerciser-Responder Write): Runs only against the HSC responder. A buffer map request for a real buffer with a buffer fill of 252 and a buffer length of 1024 bytes is sent and proper status verified. The Response Message is read and verified. Map first internal buffer in the exerciser and fill with bytes of 125. A Move Buffer Request for an exerciser write of 1024 bytes with a delay, a repeat count, a packet multiplier and a packet size of 0 is sent. Proper status is verified. The Response Message is read and verified. The first internal buffer is unmapped. Map a second internal buffer and fill with 0 bytes. A Move Buffer Request for an responder write of 1024 bytes with a delay, a repeat count, a packet multiplier and a packet size of 0 is issued. Proper status is verified. The Response Message is read and verified. The buffer is verified to contain bytes of 125. A Buffer Unmap Request for the same buffer is sent and proper status is verified. The response message is read and verified. Unmap the second internal buffer.

*TST59

TST59 Buffer Move (Exerciser-Responder Write): This test will not run against the HSC responder. Map a buffer in the responder, a Buffer Map Request for a real buffer with a buffer fill of 252 and a buffer length of 1024 bytes is sent and proper status verified. The Response Message is read and verified. Map an internal buffer of 1024 bytes and fill with bytes of 125. Transfer the data to the responder from the exerciser. Test for data transfer complete. Unmap the first internal buffer. Map a second internal buffer fill with 0 bytes. Issue a Move Buffer Request for a responder write of 1024 bytes with a delay, a repeat count, a packet multiplier and a packet size of 0 is sent and proper status verified. The response message is read and verified. Verify the buffer contains bytes of 125. A Buffer Unmap Request is sent and proper status is verified. The Response Message is read and verified.

*TST80

TST80 Read Counter Test: Connect to the selected node and verify proper status. A Read Counter Request is issued and proper status is verified. The Response Message is read and verified. The number of acks, nacks and no responses for path A and path B is printed out along with the number of disregarded datagrams.

*TST99

TST99 CTP Exerciser: Runs three iterations of the exerciser sequence while varying the repeat counts to 2, 32, and 64. This is done to increase the amount of traffic on the CI. The exerciser sequence uses the three basic methods of transfer: messages, datagrams, and buffer transfer, to create traffic on the CI.

The test begins by initializing the test tables and flags and goes on to get configuration data on all nodes on the CI. It uses this configuration data and the selected nodes to create selection list for the exerciser. Nodes that are selected but not accessible are dropped.

The exerciser sequence consists of the following for each node. If no /ITERATION switch is specified in the RUN command, then 100 iterations of the exerciser are performed. All operations are performed on each node before moving to the next item in the list.

- Connect to node.
- Issue Generate Message Commands of 64 bytes.
- Wait for and verify the Generate Message Responses.
- Issue Generate Datagram Requests of 20 bytes.
- Wait for and verify Generate Datagram Responses.
- Map internal buffer, Issue Map Buffer Request for 64 bytes, verify Map Buffer Response.
- Issue Move Buffer Request, wait for and verify Move Buffer Response.
- Verify move buffer data transfer.
- Issue Unmap Buffer Request, verify Unmap Buffer Response, Unmap internal buffer.
- Disconnect from node.

HSC-RAXX DIAG/MAINTENANCE PROCEDURES

Introduction
The following is a list of procedures for taking an HSC disk off-line to perform diagnostics and/or maintenance to that disk. Two procedures are listed. The first is for CRONIC V200, the current version. The second is for CRONIC V250.

To determine which version of CRONIC you are running, use the following procedure:

1. At the console of the HSC50 type Control-Y (^Y) or Control-C (^C) to get the command prompt HSC50>.

   NOTE
   Perform step 1 every time you type a command to the HSC50.

2. At this point, type the command Show System. This prints information about the HSC50, including which version of CRONIC you are running. The following is a sample output.

```
HSC50> sho sys
18-Sep-1985 12:28:13.54  Boot: 12-Sep-1985 12:48:40.14  Up: 143:39
Version:V250             System ID: %X00000000F00F      Name: HSC015
Front Panel Enabled      Sector size-576
Console Dump Enabled     TU58 Dump Disabled
Restart - Warm
Automatic Diagnostics  Enabled
Periodic Diagnostic Interval- 1  Enabled
DISK allocation class =   0    TAPE allocation class =   0
Start command file  Enabled


SETSHO-I Program Exit
```

3. To find the status of any disk connected to the HSC, type SHOW DISK at the command prompt HSC50>. If the drive is off-line to the host, you will see it listed under the heading "Drives Stored in saved NOHOST table." If the port was not reselected, the HSC50 does not see the drive and is not listed when you use the SHOW DISK command. See the following example.

   Drives 0 and 4 are ported to the HSC50. In this example Drive 4 has been set off-line by typing Set D4 NoHost_Access. To verify the status of this drive, type the following.

   Type ^C to get the HSC50 prompt.

```
HSC50> SHOW DISK

Unit #    R #    Port    TypeState/Version
------    ---    ----    ---------------------
   0       3      1      RA60online - host access / MC-  3HV-1

Drives stored in saved NOHOST table 4

SETSHO-I Program Exit
```

CRONIC V200
The following is the procedure to take a disk off-line for diagnostics and/or repair (current field image version).

1. Dismount the disk from each host.

2. Go to the disk drive and deselect both port buttons (pop them out).

3. Reboot the HSC50. Hit the INIT switch on the HSC50. Reboot takes about seven minutes.

4. If disk is powered off, power disk back on.

5. Go to the console of the HSC50 and type ^C or ^Y to get the command prompt HSC50>, and type the following.

   HSC50> **SET D# NOHOST_ACCESS**
   The "#" denotes the drive number.

6. Reselect the port button on the disk drive to the HSC50.

7. Run diagnostics and/or perform maintenance on the disk drive.

8. If diagnostics and/or repairs are completed and you wish to set the drive back online to the host, type the following.

   HSC50>**SET D# HOST_ACCESS** ; otherwise, go to 2.

CRONIC V250
The following is the procedure to take a disk off-line for diagnostics and/or repair.

1. Dismount the disks from each host.

2. Go to the disk and deselect the iractive port button (pop out the one that is not lit). This prevents the drive from being accessed through another path when you take it off-line.

3. Go to the console of the HSC50 and type ^C or ^Y to get the command prompt HSC50>, and type the following.

   HSC50>**SET D# NOHOST_ACCESS**
   The "#" denotes the drive number.

   If the disk drive is dual ported to another HSC50, you do not have to perform this procedure for the second HSC50, since you have already deselected the second port on the drive.

   At this point, you will see the following messages or a reasonable facsimile.

   SETSHO-I Unit currently online to host
   - De-port the drive to remove it immediately
   SETSHO-I Change to unit will take effect next time drive goes AVAILABLE

   SETSHO-I Program Exit

4. Deselect and then reselect the active port button on the disk drive. (Pop out the one that is lit and then pop it back in). The port light goes out and the HSC50 has have control of the drive.

5. Run diagnostics and/or perform maintenance on the disk drive.

6. At end of repair go to the console of the HSC50 and type ^C or ^Y to get the command prompt HSC50>, and type the following.

   HSC50>**SET D# HOST_ACCESS**".

7. Reselect the second port button.

8. Remount the disks at each host.

MASSBUS DISKS CLUSTER MODE DIAGNOSTIC PROCEDURES

Introduction

When a disk error occurs in a clustered environment, it is first necessary to isolate the "problem drive" to a single system. If the problem drive is clustered and owned by the system, it is so configured in that system's ver-CONFIG file with the ALLOW command. (Where ver-CONFIG = the current major software revision 6-1-CONFIG.CMD)

The Units utility matches drive serial-numbers to structure names and determine if the "problem structure" is seen through the CI (channel 7).

SPEAR may also be a useful tool in the isolation process. Even in a dual-ported situation between two KL10s, the problem MASSBUS (A or B) reports its errors to the physically connected drive over the path to which it is connected. At that point, device-specific diagnostics can run, as in a single-system environment.

Procedure for Checking the ver-CONFIG

The following example and procedure show how to check the ver-CONFIG file, showing the ALLOW command only. It is the simplest way to determine if a drive is clustered through the CI (channel 7) and which system owns the problem drive.

1. LOGIN on each CFS system and type the following.

```
$type   PS:<SYSTEM>6-1-CONFIG.CMD<RET>   !do this on both CFS
                                         systems;
                                         !This assumes TOPS
                                         20 version 6.1

;ALLOW CFS to talk to these guys...

ALLOW RP06 2191                          !CHANNEL 0 DRIVE 1
ALLOW RP06 2137                          !CHANNEL 0 DRIVE 3
ALLOW RP06 608                           !CHANNEL 3 DRIVE 3
ALLOW RP06 2167
ALLOW RP06 8192
```

The ALLOW command uses this format: ALLOW drive-type serial-number. In the above example, RP06s (serial-numbers) 2191, 2137, 608, 2167, and 8192 have been configured to "allow" CFS (Common File System) to access these drives. Therefore, the problem drive must be isolated to that single system.

2. At that point, run Units utility to match drive serial-numbers to structure names and determine if the "problem structure" is seen through the CI (channel 7). See the following example.

$UNITS

Status of Disk Units at 26-Jun-85 11:07:34

Mounted?	Type	Channel	Controller	Unit	Structure Name	Logical Unit	DSN
Yes	RP06	0	--	1	TANK:	0 (1 of 1)	2191
Yes	RP06	0	--	3	KEITH:	0 (1 of 1)	2137
No	RP06	3	--	2	Off-line		
Yes	RP06	3	--	3	MRCSSE:	1 (2 of 2)	608
Yes	RP06	0	--	0	MRCSSE:	0 (1 of 2)	2167
Yes	RP06	7	--	4	MRCSE:	0 (1 of 1)	8192

In this example the structure MRCSE: is not owned by the present system running the Units utility, but rather seen by the present system through the CI channel 7. Therefore, the drive owner system is not this system.

For a dual-ported "problem drive," both systems can be considered the owner system. Choose which system to designate as the owner system that is, the system from which you will run the diagnostics.

Dismounting and Removing the "Problem Structure"

After isolating the problem drive to the owner system, request that operations dismount and remove the problem structure. If operations is not available to dismount and remove the problem structure from BOTH systems, use the OPR DISMOUNT command (with REMOVAL on the owner system). First dismount the structure with "no-removal" on the foreign system. Then use the removal switch with dismount on the owner system.

```
OPR>dism str tank: (with) no-REMOVAL
OPR>
16:03:13          -- STRUCTURE DISMOUNTED --
               Structure TANK dismounted

16:03:13          -- DISMOUNT STRUCTURE --
Do not remove TANK: (alias TANK:)
User OPERATOR, Job 0, DETACHED

DISK DRIVE INFORMATION                   DISK PACK INFORMATION
     Chan-Cont   Disk   Mount   Mount                      Usage
Type   Drive    Status  Status  Count    Name              Options
---- ---------- ------  ------  -----    ----------        -----------
RP06 7,02,552   Avail   Free             TANK    (1/1)
Structure cannot be mounted unless MOUNTed via OPR
OPR>EXIT
```

2. Now the problem drive may be spun down and the user pack removed. Set the problem drive A/B port switch to the owner system port for a dual-ported drive. Go to step 3 if it is connected through the CI.

NOTE

Steps 3 and 4 are the same for a nonclustered disk drive.

3. Mount a scratch pack on the problem drive. Request that Operations mount a scratch pack already set up with HOME BLOCKS on the problem drive. If operations is not available, mount a scratch pack on the problem drive, spin the drive up and run CHECKD to set the scratch pack up with HOME BLOCKS, if necessary.

4. Request that Operations mount the KLAD pack on a free system drive. Device specific diagnostics can, now run as in a single-system environment. You can run the following diagnostics on these drives.

   RP04 DFRPH, DDRPI
   RP06 DFRPK, DDRPI
   RP07 DFRPM, DFRPN.

MULTIO.EXE INFORMATION
The following help document is designed to acquaint you with the program MULTIO.EXE. This program runs only under user-mode and can be very helpful for exercising disks (whether the disks are the MASSBUS CI-HSC-RA type). MULTIO.EXE can be found on the KLAD20 pack in the <UNSUPPORTED> area.

NOTE

Because this program's purpose is to test file I/O, running this program during prime hours can slow the system.

The following is an example of a batch control file that can be used to run MULTIO.EXE. The structure name and area being used is called PROJ:<TEST>

```
@mou str proj:
@RUN MULTIO.EXE
proj:<test>test.mem
proj:<test>test1.mem
proj:<test>test2.mem

150
300
;submitproj:multio.ctl/rest:no/output:nolog/time:1:00:00/after:20:00:00
@daytime
@i dis proj:<test>
@LOGO
```

To start this type of batch control file, do the following.

```
@CONN PROJ:<TEST>(CR)              ;CONNECT TO DESIRED STR:<AREA>
@SUBMIT MULTIO.CTL/OUTPUT:NOLOG/RESTARTABLE:NO/TIME:1:00:00 (CR)
```

NOTE

This example uses structure PROJ:, area is <TEST>, and assume the control file (MULTIO.CTL) and the program (MULTIO.EXE) exist in PROJ:<TEST>. Do not output the log, and set restart to No.

PAGES.EXE INFORMATION

The following help document is designed to acquaint you with the program PAGES.EXE. This program runs only under user-mode and can be very helpful for exercising disks (whether the disks are the MASSBUS or CI-HSC-RA type). PAGES.EXE can be found on the KLAD20 pack in the <UNSUPPORTED> area.

NOTE

Because this program's purpose is to test file I/O under TOPS20, running this program can slow the system.

The following is the PAGES help text.

```
@RUN PAGES.EXE <CR>
PAGES>help

This program provides a test for file I/O for TOPS-20  Large files
can be written and later checked for accuracy. The connected
structure and directory are used. The following are the Commands
used:

CONTENTS arg     Specify contents of files (INFORMATIVE or RANDOM).
DELETE           Delete and expunge the files.
EXIT             Leave the program.
FILES file       Specify the filename to begin files with.
HELP             Type this text.
LIMIT n          Specify the error typeout limit for each file.
MODE arg         Set the mode of the files (SEQUENTIAL or RANDOM).
NUMBER n         Set the number of files to use.
PUSH             Push to a new EXEC.
RANDOM-SEED n    Set the random number seed to n.
SIZE n           Set the size of each file.
SEEK             Do random seeks on the structure until stopped.
TAKE file        Read commands from the specified file.
VERIFY           Read and verify all files are correct.
WRITE            Write files with data.

Control/A        Type the current status of reading/writing/seeking.
Control/E        Abort reading/writing/seeking and close any open
                 files.

PAGES>exit <CR>
@
```

The following is an example of a batch control file that can be used to run PAGES.EXE.

NOTE

The structure name and area used are PROJ:<TEST>.

```
;This is the control file that will run the program pages
;on the RA81 disk drive. Its purpose is to create activity over
;the CI Bus and exercise the new hardware. If this pass of the
;program pages runs successfully, the control file will re-submit
;itself. It will continue to run until an operator or someone
;with that power cancels this batch job.
;
;First lets try and mount the structure "proj"
@mount structure proj:
@if (noerror) goto pass1::
;
@daytime
@goto end::
;
PASS1::
;We are here because the mount request was successful
;
```

```
@conn proj:<test>
@if (noerror) goto PASS2::
;
@daytime
@goto end::
;
PASS2::
; We are here because the connect command was successful
;
@run pages.exe
*files testy
*mode ran
*contents ran
*siz 100
*number 20
*write
*verify
*delete
*exit
;
;Now lets clean-up by expunging the files pages created and
deleted.
@exp proj:<test>
;
@if (noerror) goto pass3::
;
@daytime
@goto end::
;
;We are here due to some error in running PAGES.EXE
Pass3::
;
@submit
proj:<test>pages.ctl/rest:no/output:nolog/time:2:00:00/after:+00:1
0:00
@goto endl::
;
end::
;
;We are here due to an error in the ctl file
;
@daytime
@reset
@logo
;
endl::
;
;We are here because all went well and pages.ctl should have been
;resubmitted and start running in ten minutes.
@daytime
@reset
@logo
```

To start this type of batch control file up do the following.

```
@CONN PROJ:<TEST> (CR)             ;CONNECT TO DESIRED STR:<AREA>
@SUBMIT PAGES.CTL/OUTPUT:NOLOG/RESTARTABLE:NO/TIME:2:00:00 (CR)
```

NOTE

This example uses structure PROJ:, area is <TEST>, and assume the control file (PAGES.CTL) and the program (PAGES.EXE) exist in PROJ:<TEST>. We don't output the log and we set restart to no.

TOPS20 V6.1 NIA20 DISABLING
NIA20 USER MODE DIAG

TOPS20 V6.1 NIA20 DISABLING AT SYSTEM START_UP

Introduction
Use the following procedure to disable the NIA20 port at system startup.

Procedure
Follow normal boot procedure until the BOOT prompt. At the prompt, type the following.

1. BOOT> /E <cr> (or Structure:File.Nam/E if not using default structure or monitor).

2. System responds with EDDT, type the following.

3. NIINI/ <cr> (this is the starting location of NIA20 INIT).

4. RET<cr>

5. 147$G <cr> (This is the starting location of the monitor).

NOTE

Disabling the port at startup also disables any user-mode diagnostic capability on NIA20.

The system needs to be reloaded to return use of the NIA20 to the system.

NIA20 USER MODE DIAGNOSTIC PROCEDURES

Introduction
Use the following procedures to run NIA20 User-Mode Diagnostics. These procedures will help you run the diagnostics, but do not answer specific questions about a particular diagnostic. Refer to the diagnostic listing on microfiche or hardcopy for that information.

NIA20 User-Mode Diagnostics Requirements.

To run NIA20 User-Mode Diagnostics you need the following.

1. TOPS20 V6.1 KLAD or higher.

2. A privileged account.

3. A free RP06 disk drive.

4. Acknowledgment that the NIA20 will be taken off-line, leaving the system unable to communicate on the local area Network (Ethernet).

5. Knowledge of how to run D20MON.

Procedure

1. Load the KLAD pack on an available disk drive, LOGIN, enable, run OPR, and mount the KLAD onto the System. Type the following.

2. OPR>MOUNT STRUCTURE KLAD:/STRUCTURE ID:PS: <CR>

NOTE

In a cluster environment the Structure ID: of the KLAD can be any one of the following: PS:, PS1:, PS2:, PS3:. There are MIC files in your <F-S> account on the KLAD to make these changes. Refer to the KLAD Naming Convention section.

WARNING

You now need to set the NIA20 Port unavailable to the system. The Diagnostics may do, and in most cases corrupt any user data currently being used. Type the following.

3. OPR>SET PORT NI UNAVAILABLE <CR>
   OPR responds with a message that the port is disabled.

4. OPR>EXIT <CR>

5. $MOUNT STRUCTURE KLAD: <CR>
   This command mounts the KLAD to your user job.

6. $CONNECT KLAD:<2-DIAGNOSTICS> <CR>

7. RUN D20MON <CR>

   You may now select any one of the following three User-Mode Diagnostics that run on NIA20 hardware.

   DFPTA.A10 (Port Hardware)
   DFNIE.A10 (NIA Module)
   DFNIA.A10 (NIA20 Functional)

   There is a special microcode file that is loaded with DFNIA.A10 called NI20.ULD and is controlled by D20MON.

8. After running the NIA20 diagnostics, leave D20MON by typing CTRL C twice. Dismount the KLAD from your user job and the system. Set the NIA20 available. Type the following.

9. $DISMOUNT STRUCTURE KLAD: (CR)

10. OPR>DISMOUNT STRUCTURE KLAD: REMOVAL (CR)

11. OPR> SET PORT NI AVAILABLE (CR)

12. OPR> EXIT (CR)

List of Stopcodes -- BUG HLT, BUGCHK, and BUGINF
The following lists all stopcodes added to TOPS-20 since release 5.1. This list shows the name and type of each stopcode, the calling module, a phrase message (for which the name is a symbol), a brief description of the error that caused the stopcode, and any data item that can help analyze the cause. For more information, see the file BUGS.MAC in the PS:<SYSTEM> area, or consult the source code if available. This also can be found on the Distribution Tape in the Documentation area called BUG.MAN.

Table 5 BUGHLT, BUGINF, BUGCHK

Name	Type	Module	Message and Explanation
BADFEV	HLT	PHYSIO	CHKPDB - Wrong or bad front-end version. Front-end did not send a type 40 message to identify the front-end disk serial numbers.
BADIRB	HLT	PHYSIO	Bad IORB passed to GIVIRB. An IORB was passed to GIVIRB that does not have a legal address. This indicates a software problem in the monitor.
BREAKIN	INF	JSYSA	Password guess threshold exceeded. Someone has typed more than MXFLCT incorrect passwords. The system now refuses all subsequent passwords for some time. It is possible the person is trying to guess passwords. CTRLTT - Line number of the job USERNO - User number if the job is logged in STRNAM - Six-bit name of the target structure DIRNUM - Directory number of the target
CCBROT	INF	DIRECT	CPYBAK - Can't copy backup root-directory. Monitor detected a problem with the backup root-directory and is trying to copy the primary root-directory to the backup. The copy failed. LSTERR - Error returned from CPYBAK

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
CFACCF	HLT	CFSSRV	CFSSRV - SC.ACC failed. Call to SC.ACC failed. NODE - Node number CID - Connect ID ERR - Error returned by SC.ACC
CFANAE	HLT	CFSSRV	CFSSRV - No allocation entry. Caller wanted to update a directory allocation entry, and the entry could not be found.
CFBAFN	HLT	CFSSRV	CFSSRV - Bad function to CFSDAU. CFSDAU was called with an invalid function.
CFCCLZ	CHK	CFSSRV	CFSSRV - Can't close CFS connection. "Set CI offline" was requested and SCA refused to close a CFS connection. Call to SC.DIS to disconnect from the remote node failed. This may result in a CFRECN BUGHLT when the CI is put on-line. NODE - Node number of remote CID - Connect ID ERROR - Error code returned by SCA
CFCCML	INF	CFSSRV	CFSSRV - Cluster cease message lost. Another system sent a "cluster cease" that could not be queued because there was no available resident free space.
CFCONN	INF	CFSSRV	CFSSRV - CFS connection. CFS connection was received from another node on the CI20. NODE - Number of connecting node CID - Connect ID SERNUM - Serial number of remote node
CFDISC	INF	CFSSRV	CFSSRV - CFS disconnect. CFS disconnect request was received from a remote node on the CI20. NODE - Remote node number CID - Connect ID SERNUM - Serial number of remote node
CFGARD	HLT	CFSSRV	CFSSRV - Vote packet address is bad. Bad vote packet address was given to CFSWDN.
CFLISF	HLT	CFSSRV	CFSSRV - SC.LIS failed. Call to SC.LIS failed. ERR - Error code returned by SC.LIS
CFNLTK	HLT	CFSSRV	CFSSRV - Null disk address given to CFSAWT. Call was made to create an OFN access token but SPTH for the OFN is not set up.
CFRECN	HLT	CFSSRV	CFSSRV - Illegal reconnect. VC between this system and another continued illegally. NODE - Number that re-established a connection CID - Connect ID
CFSBNO	HLT	CFSSRV	Broadcast of unknown OFN. CFSBEF was called to broadcast the OF pointer for an OFN. This OFN does not have an entry in the CFSOFN table.
CFSBTP	HLT	CFSSRV	CFSSRV - Bad token packet. CFSAWT was called to acquire an access token for an OFN. OFN access token already exists on this system and the block address is in CFSOFN. But the OFN recorded in the block does not match the one passed into CFSAWT.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
CFSICN	HLT	CFSSRV	CFSSRV - Illegal configuration. This system detected an illegal configuration. There may be too many nodes in the network. The caller of this routine should be examined for more details.
CFSIGT	HLT	CFSSRV	CFSSRV - Illegal return from CFSGET. A call to CFSGET, CFSGTT, or CFSGTL returns +1, even though a wait-until-successful was requested.
CFSILJ	CHK	CFSSRV	CFSSRV - Illegal Local Job Number. LCL2GL was called to convert a local job number to a global index, but the local job number is invalid.
CFSKPD	HLT	CFSSRV	CFSSRV - The KLIPA failed. The KLIPA hardware or the CI failed and CFS cannot continue.
CFSMPB	HLT	CFSSRV	CFSSRV - CFSMAP returned in-use entry. CFSMAP returned a resource block that is already in use. This is a debugging check.
CFSNAF	HLT	CFSSRV	Allocation entry not found. An allocation entry was just created and CFSSRV can't find it in the hash table.
CFSNOT	HLT	CFSSRV	CFSSRV - OFN token table and hash table disagree. CFSSRV tried removing a file access token and has found the token in the hash table, but not in the OFN token table. This indicates that one of the data bases is incorrect.
CFSOFB	HLT	CFSSRV	CFSSRV - OFN owned at CFSOFC. CFS received a message from a remote system that an OFN changed and needs to be verified again. But CFS found that it owns the OFN. It should never have received the message.
CFSRNM	HLT	CFSSRV	CFRDSN - Could not rename DSN entry. A pack of a mounted structure was moved to a new unit and the new CFS mount resource already exists for the new drive. Or, a drive on which there is a pack of a mounted structure was given a new drive serial number and the new CFS mount resource already exists for the new drive. This indicates either the CFS data base is wrong, or PHYSIO's data base is wrong.
CFSSEZ	HLT	CFSSRV	CFSSRV - Section 0. HSHLOK was called from. section zero. HSHLOK must be called from a non-zero section.
CFSSUF	HLT	CFSSRV	CFSSUG - Could not find entry to upgrade. A request was made to change the mount type of a structure, and the CFS data base has no record of the structure being mounted.
CFSTND	HLT	CFSSRV	CFSSRV - Access token not deleted. CFSCON was called to verify that an access token was deleted before an OFN was released. This BUGHLT indicates that the token was not deleted.
CFSTUC	HLT	CFSSRV	CFSSRV - Unexpected error encountered during structure operation. A structure mount or dismount failed and generated an unexpected or illegal error code. If the error code is -1 then the failure was local to this system. This should not happen for structures. Also, there should never be an error code of zero. CODE - Bogus error code

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
CFSVFL	HLT	CFSSRV	CFSSRV - Structure verify failed. CFS could not verify an existing structure resource during the join operation. There is probably a structure-naming conflict.
CFSWMC	HLT	APRSRV	Wrong CODE for CFS. The KL microcode currently running does not support CFS.
CFZCNT	HLT	CFSSRV	CFSSRV - Zero HSHCNT before decrement. A routine wants to decrement the resource share count but the count is already zero.
CGROFN	INF	DIRECT	CHKBAK - Can't get root-directory OFN. An OFN cannot be assigned for the backup Root-Directory of a file. LSTERR - Error returned from ASGOFN
CHKRNR	HLT	SCHED	CHKR fork not run for too long. Monitor creates a fork in job zero that exists for the system's life. This fork runs periodically to perform essential functions. BUGHLT occurs when the scheduler detects that the CHKR fork has not run for too long a time. Possible causes for CHKR not running include: 1. A disk failure that prevents fork 0 from updating the disk 2. Removal of a mounted structure 3. Logic errors in the monitor 4. An HSC or MSCP server disk is hung
CINACF	CHK	CIDLL	Accept failed. CIDLL accepted an incoming connection, but the accept call to SCA failed. ERRCOD - Error code returned by SCAMPI
CINBCD	CHK	CIDLL	Bad CID. SCAMPI supplied a bad connect ID on a callback to CIDLL. CIDLL does not have any connection open with the particular connect ID. This bugcheck may occur if one system crashes while a DECnet/CI message is outstanding. CID - Bad connect ID DSPTCH - Function dispatch word
CINBSC	CHK	CIDLL	Unexpected SCA callback. SCAMPI issued a callback to CIDLL with an unexpected callback. SCAFUN - Bad function code SCACID - Connect ID that SCAMPI supplied on the call
CINFRB	CHK	CIDLL	Failed to recycle buffer. CIDLL received a datagram, and failed to return the buffer to the SCA receive queue. One less buffer is now in the port datagram receive queue. SCACID - Connect ID BUFADR - Address of buffer that could not be posted
CINLER	CHK	CIDLL	Local port # equal to remote. Activate routine noticed that we tried to connect to ourselves. CIDLL should have detected this before.
CINLIE	CHK	CIDLL	Listen failed. CIDLL asked SCA for a "promiscuous listen" but the call failed. As a consequence, the system does not accept future incoming DECnet/CI connections. ERRCOD - Error code returned from SCAMPI

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
CINNIC	CHK	CIDLL	**Received illegal packet format.** CIDLL received a DECnet datagram, but packet mode is not industry compatible. FLAGS - Flags and mode of packet
CINRRL	INF	CIDLL	**Remote rejected our protocol version.** Remote port does not understand the protocol version of DECnet/CI. The other system may be running another version of TOPS-20, or the remote system may be running VMS. DECnet/CI between TOPS-20 and VMS is not supported. LOCVER - TOPS-20 DECnet/CI protocol version
CINRWP	INF	CIDLL	**Remote supplied wrong protocol version.** The remote port is not running the same protocol version of DECnet/CI. This may happen if you are running DECnet/CI between two different versions of TOPS-20, or if you are running DECnet/CI between TOPS-20 and a VMS system. DECnet/CI between TOPS-20 and VMS is not supported. REMVER - Remote ends protocol version LOCVER - Local TOPS-20 systems protocol version
CINUCB	CHK	CIDLL	**Unexpected SCA callback.** SCA issued a callback to DECnet/CI that was not expected in the current state of the connection. STATE - DECnet/CI connection state SCACID - Connect ID for the connection in question ROUADR - Address of SCA callback processing routine
CINUDR	CHK	CIDLL	**Unexpected datagram receive.** CIDLL received a datagram with the connection state not being RUN. SCACID - Connect ID
CINUEC	CHK	CIDLL	**Unexpected connect response.** CIDLL received a callback from SCA stating that a connection response was available. CIDLL was not expecting any such callback for the port. SCACID - Connect ID
CINWNB	CHK	CIDLL	**Wrong number of buffers.** CIDLL asked SCAMPI to allocate 1 buffer, but received more than one. Extra buffers will now be lost. COUNT - Number of buffers SCAMPI allocated and returned
CIPDFQ	INF	PHYKLP	**PHYKLP - Datagram free queue empty.** Port found the datagram free queue empty.
CNTOUT	INF	DIAG	**Read of performance counter timed out.** KLIPA did not respond to a read of the performance counters in the allotted time.
COMBNN	CHK	D36COM	**Bad local node number.** Node number set with the NODE command in the CONFIG file is higher than the DECNET MAXIMUM-ADDRESS value set in the same file. DECnet cannot initialize.
COMCID	CHK	D36COM	**Couldn't initialize DECNET.** SCTINI found a reason to object about the DECnet environment. See SCTINI for the reasons it takes a non-skip return.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
COMDNP	CHK	D36COM	DNGPOS called with bad MS. The ac MS points to memory not used by message blocks. This was found during range checking.
COMDT1	HLT	DTESRV	MOVSLJ failed. DTESRV attempted to execute a MOVSLJ instruction, but failed.
COMDTE	HLT	DTESRV	MOVSLJ failed. DTESRV tried to execute a MOVSLJ instruction, but failed.
COMFWZ	CHK	D36COM	Tried to free words at zero. DNFWDS was called with a 0 pointer.
COMIEL	CHK	D36COM	Illegal end of list pointer. CHAVL, the available count, indicated there was at least one block on the free list, but the first pointer was zero.
COMMMS	HLT	D36COM	Bad pointer passed to memory manager. When DNGWDS gives out a block of memory, a check word is left just before the first word of memory given to the user. This word contains the length of the block in the right half, and a "check" quantity in the left to verify that this block is what is expected. This bug means that this word was either trashed, or the pointer we were passed is bad. BUFFER - Address of faulty buffer CALLER - Address of caller that provided the buffer
COMMTS	CHK	D36COM	New message block too short. A MOVSLJ instruction in D36COM failed.
COMMZP	CHK	D36COM	DNMINI was passed a zero pointer. Some caller probably meant to ask for zero bytes of user data in T2 and mistakenly put the count in T1, which is supposed to be the pointer to the message block to refresh.
COMODP	CHK	D36COM	DNGOPS called with bad MS. The ac MS points to memory not used by message blocks. This was found during range checking.
COMSTB	CHK	D36COM	Smear request too big. The caller requested that a very large block be smeared.
CPTMAP	CHK	PAGUTL	SETCPT - CPTPG already mapped. A routine was called to setup CPTPG while CPTPG was already setup. All callers should call RELCPT if CPTPG is mapped.
CSKBUG	HLT	SCHED	ECSKED when not CSKED. An ECSKED was done when the code was not really CSKED. This may cause sensitive code to be ruined because of races.
CTDCHB	HLT	CTHSRV	CTERM hibernate routine called. CTERM hibernate routine was called by a misguided DECnet. It should never be called.
CTDEPF	INF	CTHSRV	CTERM host enter passive failed. There was a free space allocation failure during an enter passive for a CTERM host.
CTDFRK	CHK	MEXEC	Cannot create CTERM fork. CTERM system fork could not be created and started at system startup.
CTDFSA	CHK	CTHSRV	Can't get free space for CTERM. During system startup CTERM couldn't get enough free space.
CTDILS	CHK	CTHSRV	CTERM link is in an unexpected state. CTERM link is in one of these states: Connect Sent, Connect Rejected, or some illegal state.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
CTDPRR	INF	CTHSRV	**CTERM protocol error.** A server sent TOPS-20 a message that it does not like. COUNT - Current byte count BEGIN - Pointer to beginning of message. CDB - CDB
CTYSTK	CHK	TTYSRV	**FE reload requested because CTY is stuck.** A JOB 0 fork tried to output to the console, but was unable. The job entered the J0TCOT scheduler test to wait for the CTY to clear, so that output could begin again. However, the CTY remained hung for a while and a FE reload was requested.
DDMINT	CHK	MEXEC	**Unexpected interrupt in DDMP process.** An unexpected error occurred in the process that handles page migration to disk. The error handler tries to reinitialize the context and resume processing. The stack may be examined for an indication of where the error occurred. ITFPC - PC when error occurred LSTERR - Last error code in fork
DDMPNR	HLT	SCHED	**DDMP fork not run for too long.** Monitor creates a fork in job zero that exists for the system's life. This fork runs periodically to perform essential functions. BUGHLT occurs when the scheduler detects that the DDMP fork has not run in too long a time. Possible causes for DDMP not running are. 1. A disk failure that prevents fork 0 from updating the disk 2. Mounted structure removal 3. Logic errors in the monitor
DDXFRK	HLT	MEXEC	**Cannot create CHKR fork.** CFORK% failed to create the old "Job 0" fork that runs CHKR, or the fork could not start in monitor mode with the MSFRK% JSYS. ERRCOD - Error code returned from JSYS
DDXIN	HLT	PAGUTL	**DDMP - Bad XB.** DDXBI was called to swap in a forced out index block but the index block is bad.
DIRRNA	CHK	JSYSA	**Remote node alias list inconsistency.** GTDRN1 was called to allocate space for the user's remote node alias block but the pointer to the monitor's remote node alias block provided by the caller does not contain the correct block type. DIRNUM - Directory number
DLLBPA	HLT	NISRV	**Illegal Portal supplied by PHYKNI.** PHYKNI returned an illegal portal block address when it called back NISRV. UN - UN block PR - Bad portal block address
DMPIOM	CHK	DISC	**DSKDM - I/O disk dump mode I/O called from monitor.** DSKDMI or DSKDMO called and the previous context indicates an exec mode DUMPI% or DUMPO% JSYS. There aren't any.
DNDCGE	CHK	DNADLL	**Couldn't get emergency buffer for DLL.** DNADLL requires that the memory manager save at least two buffers per link for DNADLL; one for the routing messages ROUTER keeps for each circuit, and one to guarantee some level of route-through ability. DNADLL was asked to open a data link, but the memory manager could not guarantee the buffers.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
DNDCGV	INF	DNADLL	Couldn't get memory for event arg block. DECnet used all its available memory and can not give us any.
DNDCIZ	CHK	DNADLL	Callback ID is zero. DTESRV has lost the callback ID for this line or never had one.
DNDEMF	CHK	DNADLL	Enable Ethernet multicast address failed. NISRV returned an error when trying to enable a multicast address. ERRCOD - Error code returned by NISRV
DNDNCE	CHK	DNADLL	Error from NISRV when closing portal. NISRV returned an error when asked to close our portal. ERRCOD - Error code returned by NISRV
DNDNNF	CHK	DNADLL	Network management failed. NISRV returned an error when asked to read network management parameters or counters. ERRCOD - Error code returned by NISRV
DNDNOF	CHK	DNADLL	Attempt to open an ethernet portal failed. NISRV returned an error when trying to open a portal. ERRCOD - Error code returned by NISRV
DNDRLF	CHK	DNADLL	Read channel list failed. NISRV returned an error when asked to return the channel list. ERRCOD - Error code returned by NISRV
DNDXMF	CHK	DNADLL	Transmit message to Ethernet failed. NISRV returned an error when trying to queue a message for transmit. ERRCOD - Error code returned by NISRV
DNSBPB	HLT	D36COM	DNSBP called with OWGBP. DNSBP was called with a one-word global byte pointer. DNSPB is only set up to handle local one-word and two-word byte pointers, without indexing or indirection.
DNSLJ	CHK	CIDLL	MOVSLJ failed. A MOVSLJ instruction did not skip.
DRXRNA	CHK	DIRECT	DIRRNA - Illegal formatted remote alias block in directory. Remote alias block is illegally formatted. DIRNUM - Directory number STRNAM - Six-bit structure name ADDR - Address in directory
DSKBRP	HLT	DSKALC	DSKDEA - Pages on multiple cylinders. DSKDEA was called to delete a number of pages, and the pages are on multiple cylinders. This is not allowed.
DTEBWS	INF	DTESRV	DTE MCB handshake incorrect. KL detected that the MCB's init bit was not correct during a QP2 protocol initialization handshake. A - DTE number B - PC of caller
DTECGB	CHK	DTESRV	DTE MCB initialization timed out. Couldn't allocate memory for section zero input or output buffers. DTE - DTE number PC - PC of caller

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
DTEKPA	INF	DTESRV	DTE keep alive fail. The DTE keep-alive counter is not updated by the PDP-11. DTENO - DTE number
DTESUI	INF	DTESRV	Front end requested reload or init. A PDP-11 requested a reload or init but the enabled protocol for this DTE is not DECnet. DTE - DTE number STATUS - Status word from PDP-11's comm region
DTETPR	INF	DTESRV	DTE protocol terminated. The protocol on the DTE stopped because of a BOOT% request. DTENO - DTE number
FILBAT	INF	DISC	DSKCLZ - File marked as possibly bad. A file is being closed and the OFN for the file contains a bit indicating a possible error. The file's FDB will be marked. DIRNUM - Directory number STR - Structure name in SIXBIT
FLKINT	INF	FORK	FLOCK - Called while NOINT. Routine FLOCK was called while the calling process was unable to be interrupted. The calling fork was not nesting the lock nor was it the top fork of the job. This indicated a logic error because if this fork was unable to acquire the lock it will DISMS while NOINT. This can cause a deadly embrace where the fork that owns the lock does not relinquish it until the fork that has been dismissed is interrupted, which will never happen because the fork is NOINT.
FPTMXX	HLT	PAGEM	FPTA - Process address in sched context. FPTA was called in scheduler context and given an address that is part of the process/job context area. ADR - Given address
FSICFS	HLT	DSKALC	Could not register PS with CFS. Some other CFS system has this structure mounted exclusively or as an alias and is preventing this system from mounting the structure. This is an administrative problem.
FSPANN	HLT	FREE	ASGFRE called OKINT. This is a free space problem. Calls to swappable free space routines should be made only while the calling process is NOINT. The calling routine is not protecting itself from losing free space. It is OKINT. Since it is OKINT it could be interrupted and never return, thus losing the free block assigned. POOLN - Pool number CALRPC - Caller of RELFSP
FSPARB	HLT	FREE	RELFSP - Bad block being released. The caller is trying to release a block that was already released. POOLN - Pool number CALRPC - PC of caller of RELFSP BLKADR - Address of user block

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
FSPBBS	HLT	FREE	**Bad blocksize.** This is a free space problem. The block size is either smaller than the minimum block size for this pool, or larger than the entire amount of space allocated to the pool. POOLN - Pool number BLKADR - Address of the block -- Zero indicates the pool descriptor itself contains the bad pointer.
FSPBLK	HLT	FREE	**Block damaged.** This is a free space problem. The block header does not match its trailer. POOLN - Pool number BLKADR - Address of the block -- zero indicates the pool descriptor itself contains the bad pointer.
FSPBND	HLT	FREE	**RELFSP - Block out of range.** This is a free space problem. The caller to the free space routines is trying to return a block that was not given out by the free space manager. The block is outside the range of free space management. POOLN - Pool number CALRPC - PC of caller to RELFSP BLKADR - Address of block being returned
FSPBPC	HLT	FREE	**RELFSP - Bad pool count.** This is a free space problem. The caller to the free space routines is trying to return a block so that when the pool count is increased by the blocksize, an invalid number results. The blocksize may be in error, or the pool count may already be in error. POOLN - Pool number CALRPC - PC of caller to RELFSP BLKADR - Address of block being returned
FSPDNN	HLT	FREE	**RELFSP called OKINT.** This is a free space problem. The calling routine is trying to release a swappable free space block while it is OKINT. This is dangerous since it could get interrupted and loose the block. All free space actions should occur while NOINT. POOLN - Pool number CALRPC - PC of caller of RELFSP
FSPOUT	INF	FREE	**Freespace pool exhausted.** This is a free space problem. There is no more space available in the free-space pool. POOLN - Free-space pool number
FSPPRE	HLT	FREE	**RELFSP - Bad block being released.** This is a free space problem. The block being returned does not fit into the free pool. The block would overlap the preceding block in the pool. POOLN - Pool number CALRPC - PC of caller of RELFSP BLKADR - Address of user block
FSPSCC	HLT	FREE	**RELFSP - Bad block being released.** This is a free space problem. The block being returned does not fit into the free pool. The block would overlap the succeeding block in the pool. POOLN - Pool number CALRPC - PC of caller of RELFSP BLKADR - Address of user block

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
FSPZER	HLT	FREE	**ASGFSP - Illegal to assign 0 FREE space.** An illegal request for free space was made. The calling routine is asking for zero words of free space. POOLN - Pool number CALRPC - PC of caller of ASGFSP
HOMGON	HLT	PHYSIO	**FRTHOB - Missing homeblock IORB.** Missing home block IORB when there should be one on the PWQ. CHN - Channel number KONT - Controller number UNIT - Unit number
ILDSTF	INF	DATIME	**Illegal Daylight Saving Time flag.** Location DSTFLG contains an illegal value. The most likely case of this bug is a new way of confusing DST that subroutine DSTCHK wasn't informed about. **DATA follows:** DSTFLG - Daylight savings time flag
ILESCD	HLT	PAGEM	**Monitor section pointer not shared.** A pointer for a monitor section was found that is not a share pointer. Only share pointers are expected. If other pointer types are used, this code must be enhanced. It is possible that the monitor section table has been clobbered. POINTER - Pointer SECTION - Monitor section for which it was found.
ILLMJS	CHK	APRSRV	**JSYS with E GTR 1000 executed in monitor.** A JSYS with E greater than 1000 was executed in the monitor. There should be no such cases. PC - PC of JSYS ILLTAB CHK LOGNAM TABLK2 - Table is not in proper format. A logical name table is not in alphabetical order. TABADD - Address of logical name table ILLUUO HLT APRSRV,KIBADU - Illegal UUO from monitor context. The monitor executed an instruction that the microcode treats as an MUUO. The op code is not 104 (for a JSYS) or one of the KA10 floating point instructions. FLAGS - Processor flags when MUUO is executed PC - PC in monitor address space where MUUO is executed EFFADR - Effective address of MUUO
INVDFN	CHK	DTESRV	**DTEDSP - Bad function specified.** Caller of DTEDSP supplied an illegal controller function.
ITNOJC	HLT	SCHED	**Instruction trap not in JSYS context.** Illegal instruction trap handler was entered, but the process is not in JSYS context. LSTERR - Last error code ERRPC - PC at which error was generated MUUOPC - Last MUUO PC

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
JB0CSH	CHK	MEXEC	**Job 0 crash.** An unexpected interrupt occurred in the job 0 fork that checks system status. The context is reinitialized, and the process restarted. The stack may be examined to determine the situation that caused the error. PC - PC at which error occurred LSTERR - Last error code for this fork
JB0INX	CHK	MEXEC	**Unexpected interrupt in job 0 during initialization.** An unexpected error occurred in Job 0 that results in control being transferred to the default error handler. This happened during job 0 initialization. The error handler tries to reset the context and continues at the specified error address, however, some system resources may be hung as a result of locks not being cleared. The stack can be examined to determine what was in progress when the error occurred. PPC - PC at which error occurred NEWPC - Address to which control is transferred after cleanup LSTERR - Last error code in this fork
JSTERR	HLT	FREE	**JSB stack error.** This is a problem with the JSB-stack logic; the count for the stack indicated that free cells were available, however, none could be found.
KLIPAF	CHK	MEXEC	**Failed to read in CI20 microcode.** At system startup we tried to read in the CI20 UCODE. Routine KLPUCD in module PHYKLP got a JSYS error while trying the read. ERRCOD - Error code returned
KLPBDS	HLT	PHYKLP	**PHYKLP - Bad dispatch from PHYSIO.** PHYKLP was called to perform a function of which it is not capable.
KLPBOP	CHK	PHYKLP	**PHYKLP - Bad op code on command queue.** A packet with an illegal op code was found while purging the command queue. BOC - Bad code
KLPBPK	HLT	PHYKLP	**PHYKLP - Bad packet.** The virtual address of the packet is invalid. KLPBRC INF PHYKLP, Bad Read-Counters. TOPS-20 removed a Read-Counters packet from response queue and the reason code field contains an illegal value.
KLPCBN	INF	PHYKLP	**PHYKLP - CBUS not available.** The port was not able to get the CBus. CSR - Result of last CONI LAR - CRAM's last address read EWORD3 - PCB error word 3 EWORD4 - PCB error word 4
KLPCBS	INF	PHYKLP	**PHYKLP - CBUS parity error.** The CI20 had a CBus parity error. CSR - Result of last CONI LAR - CRAM's last address read EWORD1 - PCB error word 1 EWORD2 - PCB error word 2
KLPCGN	CHK	PHYKLP	**PHYKLP - Can't get CI node number.** The CI20 driver did a Read-Register command to get the CI node number from the port; it timed out waiting for the reply. Most likely, the port is the problem.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
KLPCKE	CHK	PHYKLP	PHYKLP - SET-CIRCUIT command error. A Set-Circuit command failed. TOPS-20 doesn't retry such commands because the CI port always executes them properly. The port is probably in trouble. STATUS - Status field of packet FLAGS - Flags field of packet OPC - Op code field of packet
KLPCLB	INF	PHYKLP	Close buffer function failed. Look at the status word to find out. STATUS - Status word
KLPCRR	CHK	PHYKLP	PHYKLP - READ-REGISTER command failed. There is a problem with the CI20 port.
KLPCSR	INF	PHYKLP	PHYKLP - Grant CSR error. The port timed out waiting for Grant CSR. CSR - Result of last CONI LAR - CRAM's last address read CRAM1 - Contents of first CRAM word CRAM2 - Contents of next CRAM word
KLPCVC	INF	PHYKLP	PHYKLP - Closed virtual circuit. TOPS-20 has closed a virtual circuit to a remote node on the CI. NODE - CI node number
KLPDED	CHK	PHYKLP	PHYKLP - CI20 is dead, no longer trying to start it. TOPS-20 tried to restart the CI20 and the procedure failed twice in a row. The CI20 is left in its current state. ERROR - Error code for failure
KLPDPP	INF	PHYKLP	PHYKLP - Data path error. The port's MOVER/FMTR detected a parity error. CSR - Result of last CONI LAR - CRAM's last address read EWORD0 - PCB error word 0 EWORD1 - PCB error word 1
KLPDRQ	INF	PHYKLP	PHYKLP - CI ucode dump requested. TOPS-20 decided the CI20 microcode needs to be dumped.
KLPDUM	INF	PHYKLP	PHYKLP - CI20 ucode dump before in progress. DIAG% was executed before dumping the CI20 port's microcode. DFORK - Fork doing the dump
KLPEBP	INF	PHYKLP	PHYKLP - EBUS parity error. Port received a data word with bad parity from the KL. This did not happen while processing a queue. CSR - Result of last CONI LAR - CRAM's last address read EWORD0 - PCB error word 0
KLPEBQ	INF	PHYKLP	PHYKLP - EBUS parity error. Port received a data word with bad parity from the KL. This happened while processing a queue. CSR - Result of last CONI LAR - CRAM's last address read EWORD0 - PCB error word 0 EWORD1 - PCB error word 1
KLPELL	INF	PHYKLP	PHYKLP - Error Log Lost. Can't get free space to create ERROR.SYS entry for Error Log Message.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
KLPELT	INF	PHYKLP	PHYKLP - **Error Log Truncated.** An Error Log Message was truncated in its ERROR.SYS entry.
KLPEPB	CHK	PHYKLP	PHYKLP - **ERROR-LOGGING PACKET IS BAD.** TOPS-20 received an error-logging packet (PPD byte 5) that had an error. The packet is returned immediately to the free queue. The information it carried is lost. STATS - Status field of packet FLAGS - Flags field of packet OPC - Op code field of packet NODE - Node number
KLPERE	INF	PHYKLP	PHYKLP - **EBUS request error.** The port could not get the EBUS. CSR - Result of last CONI LAR - CRAM's last address read
KLPERQ	INF	PHYKLP	PHYKLP - **Empty response queue.** The monitor got an interrupt to remove a packet from the response queue. The queue was empty.
KLPERR	INF	PHYKLP	PHYKLP - **CI packet error.** The CI20 driver received a packet (message or named buffer) with an error. This causes the virtual circuit to be closed. STATS - Status field of packet FLAGS - Flags field of packet OPC - Op code field of packet NODE - Node number
KLPFST	INF	PHYKLP	PHYKLP - **Self test failed.** The port had a failure during its self-test. CSR - Result of last CONI VER - Microcode version LAR - CRAM's last address read
KLPHNG	INF	PHYKLP	PHYKLP - **CI20 is hung.** The response bit on a REQUEST-ID command was set and timed out waiting for it to appear on the response queue. There is a problem with the port.
KLPHOG	HLT	PHYKLP	PHYKLP - **Interlock value on queue is too large.** The KLIPA driver timed out the interlock, but the value isn't what is expected.
KLPIBN	INF	PHYKLP	PHYKLP - **Invalid buffer name.** The CI20 driver received a packet (message or named buffer) with an Invalid Buffer Name error. STATS - Status field of packet FLAGS - Flags field of packet OPC - Op code field of packet NODE - Node number
KLPILP	INF	PHYKLP	PHYKLP - **Software response bit off in locally-generated packet.** The response queue contains a packet whose op code indicates that the packet was queued by this host, but the software response bit is not set and there was no error. NODE - Node number STATUS - Status word
KLPINP	INF	PHYKLP	PHYKLP - **Internal port error.** Port found an inconsistency in an operation it was performing. CSR - Result of last CONI VER - Microcode code version LAR - CRAM's last address read.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
KLPIPA	CHK	PHYKLP	**PHYKLP - Invalid packet arrival.** CI20 driver received an application packet from a node with which it doesn't think it has ever communicated. STATS - Status field of packet FLAGS - Flags field of packet OPC - Op code field of packet NODE - Node number
KLPIRD	CHK	PHYKLP	**PHYKLP - Invalid remotely-generated data request.** CI20 driver received an error-free, remotely-generated packet with op code 10, 11, 12, or 20. This is illegal. STATS - Status field of packet FLAGS - Flags field of packet OPC - Op code field of packet NODE - Node number
KLPIRP	INF	PHYKLP	**PHYKLP - Software response bit on in remotely-generated packet.** Response queue contains a packet whose op code indicates that the packet was queued by a remote host but the software response bit is set. NODE - Node number STATUS - Status word
KLPLBF	CHK	PHYKLP	**PHYKLP - Loopback failed.** CI20 driver tried to send a loopback packet to the STAR coupler and it had a non-path error. STATS - Status field of packet FLAGS - Flags field of packet OPC - Op code field of packet CSR - Result of the last CONI
KLPLOA	INF	PHYKLP	**PHYKLP - CI20 ucode loaded.** PS:[SYSTEM]IPALOD.EXE was run or the monitor initiated the reload. EDIT - Edit number of microcode
KLPMBS	INF	PHYKLP	**PHYKLP - MBUS error.** Multiple MBus drivers are simultaneously accessing MBus. CSR - Result of last CONI LAR - CRAM's last address read CRAM1 - Contents of first CRAM word CRAM2 - Contents of next CRAM word
KLPMCE	INF	PHYKLP	**Received an MCNF or an MDATREC with an error.** Check the error code. NODE - Node number of the CI node STATUS - Status word of the packet
KLPMCR	CHK	PHYKLP	**Received an MCNF or an MDATREC from KLIPA when not expecting it.** Either the maintenance function timed out, or the KLIPA gave us a spurious one. NODE - Node number of CI node that sent MCNF or MDATREC
KLPMTY	HLT	PHYKLP	**PHYKLP - Queue is empty.** We want to trace the pointers on a queue but the queue is empty.
KLPNDE	CHK	PHYKLP	**PHYKLP - Packet with bad node number.** CI20 driver received a packet with an invalid node number. The packet was not returned to a free queue. STATS - Status field of packet FLAGS - Flags field of packet OPC - Op code field of packet NODE - Node number

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
KLPNDG	INF	PHYKLP	**PHYKLP - No datagram buffer.** TOPS-20 tried to remove a buffer from the datagram free queue but the queue was empty.
KLPNDM	HLT	PHYKLP	**PHYKLP - CI20 ucode needs dumping.** Port microcode needs to be dumped but there is a timeout waiting for it to get started.
KLPNEN	CHK	PHYKLP	**PHYKLP - CI20 not enabled.** TOPS-20 believes the CI20 should be enabled but found otherwise.
KLPNMG	INF	PHYKLP	**PHYKLP - No message buffer.** TOPS-20 tried to remove a buffer from the message-free queue but the queue was empty.
KLPNOA	CHK	PHYKLP	**PHYKLP - Remote port is not answering.** Remote node is asking REQUEST-IDs but not sending IDRECs. Remote system needs to be checked. PORT - Remote port number.
KLPNOD	HLT	PHYKLP	**PHYKLP - Can't stock datagram free queue.** CALL SC.ALD failed. SCA can't handle the request.
KLPNOM	HLT	PHYKLP	**PHYKLP - Physical address doesn't match.** Physical address of a packet is stored in the packet. The physical address of this packet doesn't match what is in the packet.
KLPNRL	HLT	PHYKLP	**PHYKLP - CI20 ucode needs reloading.** Port microcode needs to be reloaded but there is a time-out waiting for it to get started. JOB 0 probably has a problem.
KLPNSB	HLT	PHYKLP	**PHYKLP - No system block at OPENVC.** OPENVC was called with a system block address of 0.
KLPONC	HLT	PHYKLP	**PHYKLP - Trying to open a VC which isn't closed.** OPENVC was called when the VC was not closed.
KLPOPC	CHK	PHYKLP	**PHYKLP - Packet with bad op-code.** CI20 driver received a packet with an invalid op code. The packet was not returned to a free queue. STATS - Status field of packet FLAGS - Flags field of packet OPC - Op code field of packet NODE - Node number
KLPOVC	INF	PHYKLP	**PHYKLP - Opened virtual circuit.** TOPS-20 opened a virtual circuit to a remote node on the CI. NODE - CI node number
KLPPCB	HLT	PHYKLP	**PHYKLP - PCB is corrupted.** PCB has invalid data.
KLPPIA	INF	PHYKLP	**PHYKLP - CI20 has lost its PIA.** The CI20 no longer knows its interrupt assignment. CSR - Result of the last CONI
KLPPPD	CHK	PHYKLP	**PHYKLP - Packet with bad PPD byte.** The CI20 driver received a packet with an invalid PPD byte. The packet was not returned to a free queue. STATS - Status field of packet OPC - Op code field of packet NODE - Node number PPD - PPD byte

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
KLPPPE	INF	PHYKLP	**PHYKLP - PLI parity error.** The port detected bad parity on a PLI BUS read. CSR - Result of last CONI LAR - CRAM's last address read
KLPPRI	HLT	PHYKLP	**PHYKLP - Invalid priority.** KLPSND was called with an invalid priority.
KLPRAE	INF	PHYKLP	**PHYKLP - Spurious receive attention error.** Port found Attention up but the packet was not totally stored in the receive buffers. CSR - Result of last CONI VER - Microcode version LAR - CRAM's last address read
KLPRCE	CHK	PHYKLP	**PHYKLP - READ-COUNTERS command failed.** There is a problem with the CI20 port.
KLPRRQ	INF	PHYKLP	**PHYKLP - CI20 ucode reload requested.** TOPS-20 has decided the CI20 microcode needs to be reloaded.
KLPRSF	INF	PHYKLP	**PHYKLP - CI restart failed.** TOPS-20 tried to restart the CI20 and the procedure failed. ERROR - Error code for failure
KLPRSH	INF	PHYKLP	**PHYKLP - Received shutdown message.** A CI node notified our node that it is closing our VC. NODE - Node number
KLPSCE	INF	PHYKLP	**PHYKLP - Spurious channel error.** Channel Error was asserted but no channel error information was in the channel logout word. CSR - Result of last CONI VER - Microcode version LAR - CRAM's last address read LWORD1 - CHANNEL LOGOUT WORD 1
KLPSCR	CHK	PHYKLP	**PHYKLP - SET-CIRCUIT command received.** TOPS-20 found an error-free SET-CIRCUIT command on the response queue. CI port done did something wrong because the response bit is never set, so this packet should not be seen. STATUS - Status field of packet FLAGS - Flags field of packet OPC - Op code field of packet
KLPSDM	CHK	PHYKLP	**PHYKLP - CI20 ucode still dumping.** Port microcode is dumped and there is a time-out waiting for it to complete.
KLPSRL	CHK	PHYKLP	**PHYKLP - CI20 ucode still reloading.** The port microcode is reloaded and there is a time-out waiting for it to complete.
KLPSRM	INF	PHYKLP	**PHYKLP - Cannot start remote node.** This node wanted to start a remote HSC node, but it is not the node that did the last Reset Remote on the remote HSC. HOST NODE - Node number of this system RESET NODE - Node number that last reset the remote node REMOTE NODE - Remote's node number
KLPSTP	INF	PHYKLP	**PHYKLP - CI20 stopped.** TOPS-20 stopped the CI20.
KLPSTR	INF	PHYKLP	**PHYKLP - CI20 started.** TOPS-20 restarted the CI20.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
KLPSWC	INF	PHYKLP	**PHYKLP - Short word count.** Port detected a short word count CBUS channel error. CSR - Result of last CONI LAR - CRAM's last address read EWORD1 - PCB error word 1 EWORD2 - PCB error word 2
KLPSWO	INF	PHYKLP	**PHYKLP - Received a START when VC was open.** TOPS-20 closed a virtual circuit because it received a Start packet while the circuit was open. NODE - Node number
KLPTAE	INF	PHYKLP	**PHYKLP - Spurious transmit attention error.** Port found Attention up before the Transmit Packet function completed. CSR - Result of last CONI VER - Microcode version LAR - CRAM's last address read REG - Transmit status register
KLPTIM	CHK	PHYKLP	**PHYKLP - Timed out waiting for queue interlock.** KLIPA driver timed out trying to get the interlock for a queue. KLIPA microcode should never have the lock this long. QUEUE - Address of the queue's interlock word
KLPTMO	INF	PHYKLP	**PHYKLP - Transmitter timeout.** Someone is hogging the CI. The Link module could not transmit over the CI due to carrier detect being continuously asserted. CSR - Result of last CONI REG - Transmit status register VER - Microcode version
KLPTPE	INF	PHYKLP	**PHYKLP - Transmit buffer parity error.** A bit was dropped or picked up in the Transmit Buffer or the Transmit Data Bus. CSR - Result of last CONI REG - Transmit status register VER - Microcode version
KLPUCP	INF	PHYKLP	**PHYKLP - Unplanned CRAM parity error.** Port had an unplanned CRAM parity error. CSR - Result of last CONI LAR - CRAM's last address read CRAM1 - Contents of first CRAM word CRAM2 - Contents of next CRAM word
KLPUMV	INF	PHYKLP	**Unexpected CI20 microcode version.** The monitor has an assembled-in value of the CI20 microcode which it is expecting to load. The microcode just loaded is a different version. AVER - Actual version loaded EVER - Expected version
KLPUPC	INF	PHYKLP	**PHYKLP - Undefined planned CRAM parity error.** Port had a planned CRAM parity error but it is not defined. CSR - Result of last CONI LAR - CRAM's last address read CRAM1 - Contents of first CRAM word CRAM2 - Contents of next CRAM word

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
KLPVIR	CHK	PHYKLP	**PHYKLP - Virtual address in packet is wrong.** The virtual address of a packet is incorrect. This indicates some sort of inconsistency in one of the queues. QUEUE - Address of the queue's interlock word VMA - Contents of the software word in the packet PMA - Physical address of word pointed to FLINK - FLINK word from PCB
KLPWAB	INF	PHYKLP	**PHYKLP - CI wire A has gone from good to bad.** A loopback packet that already succeeded, failed on wire A. STATS - Status field of packet FLAGS - Flags field of packet OPC - Op code field of packet CSR - Result of the last CONI
KLPWAG	INF	PHYKLP	**PHYKLP - CI wire A has gone from bad to good.** A loopback packet that already failed, successfully returned on wire A.
KLPWBB	INF	PHYKLP	**PHYKLP - CI wire B has gone from good to bad.** A loopback packet that already succeeded, failed on wire B. STATS - Status field of packet FLAGS - Flags field of packet OPC - Op code field of packet CSR - Result of the last CONI
KLPWBG	INF	PHYKLP	**PHYKLP - CI wire B has gone from bad to good.** A loopback packet that already failed has successfully returned on wire B.
KNIADE	HLT	PHYKNI	**PHYKNI - Multicast address disable error.** NIDPT got an error from NIDRA when trying to disable a multicast address that was supposedly enabled.
KNIADR	CHK	PHYKNI	**Monitor address does not match KLNI address.** PHYKNI just read the Ethernet address from the KLNI and found it different from the shadow copy stored in the monitor. KLNHIO & KLNLO - KLNI's copy of Ethernet address MONHIO & MONLO - Monitor's copy of the Ethernet address
KNIBFC	HLT	PHYKNI	**PHYKNI - Illegal NI function code.** NISRV called PHYKNI with a bad function code. The code is in T1. FUNC - Illegal function code
KNIBLV	HLT	PHYKNI	**PHYKNI - Buffer length violation.** BSD chain contained inconsistent length information for the transmit or receive command that caused it.
KNIBTB	HLT	PHYKNI	**PHYKNI - Bad BYTAB entry.** BYTAB has been corrupted. ENTRY - Corrupted entry BYTPTR - Byte pointer used to fetch this entry

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
KNICAE	INF	PHYKNI	**PHYKNI - CBUS available timeout.** NIA20 was unable to acquire control of the CBus within 50 microseconds from the start of a CBus request. KNILDR dumps and restarts the NIA20. CSR - CONI KNI ADDR - Address of parity error LOGOU1 - Channel logout word 1 LOGOU2 - Channel logout word 2
KNICCF	INF	PHYKNI	**PHYKNI - Carrier check failed.** NIA module did not detect it's own carrier while it was transmitting. TDR - TDR value
KNICDF	CHK	PHYKNI	**PHYKNI - Collision detect check failed.** H4000 did not assert the collision detect signal shortly after completion of a transmission. (This signal is also known as the "Heartbeat" of the H4000).
KNICFF	CHK	PHYKNI	**PHYKNI - Cannot reload the KLNI.** Monitor Was unable to find SYSTEM:KNILDR.EXE when it tried to reload or dump the port. ERROR - Error code from RUNDII (probably a JSYS error)
KNICFP	HLT	NISRV	**Cannot find portal block during close.** NISRV was unable to find a portal block on the portal block list during a close portal callback. PR - Portal block address
KNICPE	INF	PHYKNI	**PHYKNI - CBUS parity error.** NIA20 detected bad parity for data that was read over the CBus. KNILDR dumps and restarts the NIA20. CSR - CONI KNI ADDR - Address of parity error LOGOU1 - Channel logout word 1 LOGOU2 - Channel logout word 2
KNIDM1	INF	NISRV	**KNIDMD continued.** Additional data for KNIDMD. PROTO - Protocol type
KNIDMD	INF	NISRV	**Portal not enabled for this multicast.** A portal received a multicast frame on an address it wasn't enabled for. The frame is discarded, and the buffer is re-used. HIDST - High order destination address LODST - Low order destination address HISRC - High order source address LOSRC - Low order source address
KNIDOV	CHK	PHYKNI	**PHYKNI - NIA buffer overrun.** NIA module did not have enough free space to store an incoming datagram.
KNIDPE	INF	PHYKNI	**PHYKNI - NIA20 data path error.** Threshold (5) for data mover parity errors was exceeded. KNILDR dumps and restarts the NIA20. CSR - CONI KNI ADDR - Address of parity error LOGOU1 - Channel logout word 1 LOGOU2 - Channel logout word 2

Table 5   BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
KNIEPE	INF	PHYKNI	**PHYKNI - EBUS parity error.** NIA20 received a word with bad parity from the EBus. KNILDR dumps and restarts the NIA20. CSR - CONI KNI ADDR - Address of parity error LOGOU1 - Channel logout word 1 LOGOU2 - Channel logout word 2
KNIERE	INF	PHYKNI	**PHYKNI - EBUS request timeout.** NIA20 was unable to get control of the EBus within 20 milliseconds after making a PI request. KNILDR dumps and restarts the NIA20. CSR - CONI KNI ADDR - Address of parity error LOGOU1 - Channel logout word 1 LOGOU2 - Channel logout word 2
KNIERP	HLT	NISRV	**Illegal error return from PHYKNI.** NISRV got an error return from PHYKNI while processing a state change callback. The error code (one of the UNxyz% errors) is in T1. ERROR - Error returned from PHYKNI.
KNIFBE	INF	PHYKNI	**PHYKNI - NIA20 free buffer list parity error.** NIA receive status indicated that there was a free buffer list parity error. KNILDR dumps and restarts the NIA20. CSR - CONI KNI ADDR - Address of parity error LOGOU1 - Channel logout word 1 LOGOU2 - Channel logout word 2
KNIFQE	INF	PHYKNI	**PHYKNI - Free Queue Error.** KLNI received a packet for a protocol, and there were no free packets available for that protocol type.
KNIFST	INF	PHYKNI	**PHYKNI - Failed self test.** When the NIA20 is idle it runs a self-test to check out various pieces of logic (such as the ALU, the microsequencer, and the data mover/formatter). It also performs a self-test when it is first started. In one of those cases, the self-test failed. KNILDR dumps and restarts the NIA20. CSR - CONI KNI ADDR - Address of parity error LOGOU1 - Channel logout word 1 LOGOU2 - Channel logout word 2
KNIFTL	CHK	PHYKNI	**PHYKNI - Frame too long.** NIA module detected that it sent a frame longer than 1536 bytes.
KNIFTS	HLT	PHYKNI	**PHYKNI - Frame too short.** Port was told to send a frame with less than 46 bytes of user data and the pad flag (CMPAD) was not set. NISND should have detected this.
KNIGCE	INF	PHYKNI	**PHYKNI - Grant CSR timeout.** NIA20 was unable to control the CSR (CONI word) within 10 milliseconds after requesting it. KNILDR dumps and restarts the NIA20. CSR - CONI KNI ADDR - Address of parity error LOGOU1 - Channel logout word 1 LOGOU2 - Channel logout word 2

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
KNIHED	CHK	PHYKNI	**PHYKNI - Hard error detected.** MBUS ERROR, or EBUS PARITY ERROR was detected. This is a KLNI hardware problem. The address (ADDR) and it's contents (LOCMSB and LOCLSB) are printed out. CONI - CONI KNI PC - PC (Microcode PC at time of problem)
KNIIAM	CHK	PHYKNI	**PHYKNI - Illegal addressing mode.** An illegal addressing mode was specified. ADR - Mode specified
KNIICA	HLT	PHYKNI	**PHYKNI - Illegal channel block address.** Channel block address for this portal is invalid. PS - Bad channel block address PR - Bad portal block address
KNIICF	HLT	PHYKNI	**PHYKNI - Illegal read counters function.** Read counters callback routine detected an illegal function code in the field C1FNC of the command block.
KNIIEC	HLT	PHYKNI	**PHYKNI - Illegal port error code.** Port generated a response that contained an unknown error code and an inappropriate error code for the command. CODE - Error code CMD - Command
KNIIFD	CHK	PHYKNI	**PHYKNI - Illegal function from DLL.** NIDLL called the driver with a function it does not yet handle. BLKADR - Function block address FNC - Function code
KNIINF	CHK	PHYKNI	**PHYKNI - KLNI initialization timed out.** KLNI timed out during Initialization. Either "disable complete" or "enable complete" didn't set (the CONI indicates which). This is probably a hardware problem, because the microcode version number was valid, and there was no specific error indication in the CONI. CONI - CONI KNI
KNIIPE	INF	PHYKNI	**PHYKNI - Internal port error.** NIA20 detected an inconsistency in an operation it was performing. The inconsistency can be caused by many things, but the result is that the function did not occur correctly or was not logical. KNILDR dumps and restarts the NIA20. CSR - CONI KNI VERSION - NIA20 microcode version number ADDR - Address of parity error
KNIIPF	HLT	PHYKNI	**PHYKNI - Illegal channel dispatch.** KLNI driver was called to perform a PHYSIO function it is not able to do.
KNIIPT	HLT	PHYKNI	**PHYKNI - Illegal protocol type on close.** A protocol type was specified on the close that was NOT enabled. PTYPE - Specified protocol type
KNIIRC	HLT	PHYKNI	**Illegal status on close.** Status field contained an unexpected value upon return from the close function. STATUS - Status

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
KNINBS	HLT	PHYKNI	**PHYKNI - Non-BSD datagram sent.** A NON-BSD style datagram was sent. The driver does not send this style. BUFFER - Buffer address
KNINIB	HLT	PHYKNI	**PHYKNI - No control buffer at interrupt level.** Port Storage (PS) block was not set up with the address of a UN block to be used at interrupt level.
KNIPER	CHK	PHYKNI	**PHYKNI - CRAM parity error.** KLNI detected a parity error in it's Control RAM. This is a hardware problem. CONI - CONI ADDR - Parity error address LOCMSB & LOCLSB - Memory location contents
KNIPIE	INF	PHYKNI	**PHYKNI - PLI parity error.** More than five parity errors occurred when reading data over the PLI interface. KNILDR dumps and restarts the NIA20.
KNIQUE	CHK	PHYKNI	**PHYKNI - Queue empty on entry.** A queue was empty when the routine REMQUE was called. QUE - Queue header address
KNIRFD	INF	PHYKNI	**PHYKNI - Remote failure to defer.** A collision was detected after the NIA "acquired" control of the Ethernet cable. This is also known as a "late collision." A collision may only occur during the transmission of a frame preamble. This problem occurs when the collision is detected after the preamble was sent. TDR - TDR value
KNIRIT	CHK	PHYKNI	**PHYKNI - Response queue interlock timed out.** PHYKNI did not get the response queue interlock after 5000 tries.
KNIRLF	CHK	PHYKNI	**PHYKNI - KLNI Reload Failed.** KNILDR ran, but failed to reload the KLNI. STATE - State of KLNI
KNISCE	INF	PHYKNI	**PHYKNI - Spurious channel error.** A spurious channel error occurs when the channel raises the error signal, but no error bits are in the channel logout area. This error occurs after the threshold (5) of spurious channel errors is exceeded. KNILDR dumps and restarts the NIA20. CSR - CONI KNI ADDR - Address of parity error LOGOU1 - Channel logout word 1 LOGOU2 - Channel logout word 2
KNISTA	INF	PHYKNI	**PHYKNI - NIA20 spurious transmit attention.** NIA module set the PLI transmit attention bit, but the transmit status was zero. KNILDR dumps and restarts the NIA20. CSR - CONI KNI ADDR - Address of parity error LOGOU1 - Channel logout word 1 LOGOU2 - Channel logout word 2
KNISTP	CHK	PHYKNI	**PHYKNI - KLNI STOPPED.** No response from KLNI after five seconds. CONI - CONI KNI LAR - Latched Address Register

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
KNISWC	INF	PHYKNI	**PHYKNI - Channel short word count.** When the NIA20 completes a CBus transfer, the channel has a short word count error. KNILDR dumps and restarts the NIA20. CSR - CONI KNI ADDR - Parity error address LOGOU1 - Channel logout word 1 LOGOU2 - Channel logout word 2
KNIUBE	INF	PHYKNI	**PHYKNI - NIA20 used buffer list parity error.** Port received a PLI parity error while reading the NIA module's user buffer list. This error is only reported after a threshold (5) for this type of error is exceeded. KNILDR dumps and restarts the NIA20. CSR - CONI KNI ADDR - Address of parity error LOGOU1 - Channel logout word 1 LOGOU2 - Channel logout word 2
KNIUOP	HLT	PHYKNI	**PHYKNI - Unknown response.** The port gave us a response we don't know about. RESP - Response
KNIUPE	INF	PHYKNI	**PHYKNI - NIA20 unknown planned CRAM parity error.** NIA20 got a CRAM parity error in the range of 7750 to 7777. This particular error falls into this range, but is not known to TOPS-20. KNILDR dumps and restarts the NIA20. CSR - CONI KNI ADDR - Address of parity error LOGOU1 - Channel logout word 1 LOGOU2 - Channel logout word 2
KNIVAR	CHK	PHYKNI	**Monitor variables do not match KLNI variables.** PHYKNI just read some status variables from the KLNI and found them different from the shadow copies stored in the monitor. KLNI - KLNI's version of the variables MON - Monitor's version of the variables
KNIVER	CHK	PHYKNI	**Bad KLNI microcode version.** PHYKNI read the microcode version number from the KLNI, and determined that it is below the minimum revision level required for proper port/driver operation. The port does not start in this case. BADMAJ - Major version number read from KLNI BADMIN - Minor version number read from KLNI GODMAJ - Major version number we require GODMIN - Minor version number we require
KNIXPE	INF	PHYKNI	**PHYKNI - NIA20 transmit buffer parity error.** NIA transmit status indicated a transmit buffer parity error. This error is not reported until a threshold (5) of this type of error is exceeded. KNILDR dumps and restarts the NIA20. CSR - CONI KNI ADDR - Address of parity error LOGOU1 - Channel logout word 1 LOGOU2 - Channel logout word 2
LAPRBF	CHK	LATSRV	**Specify Receive Buffer Failure.** LATSRV received an error from NISRV while trying to post a receive buffer. DLLERC - Error code returned by NISRV

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
LATICB	CHK	LATSRV	**LATCBR called from NISRV with illegal callback function code.** NISRV called the LATSRV callback routine with an invalid function code. CODE - Function code
LATIMT	CHK	LATSRV	**LAT Illegal Message Type.** LAT virtual circuit message was received with a message type out of range. MSGID - Message type
LATINE	CHK	LATSRV	**LATINI failed to initialize.** Could not obtain enough memory for the LAT host databases. HN.LST words for the host node database, CBMAXI words for CBVECT,and NTTLAH words for SBVECT.
LATIST	INF	LATSRV	**LAT Illegal Slot Type.** LAT Slot received with Slot type out of range. SLTID - Slot ID
LATNSC	INF	LATSRV	**LAT Host node stopped circuit.** LAT Host node stopped the circuit. CODE - Reason code PC - PC
LGFAIL	INF	MEXEC	**LGOUT or LOGIN JSYS failed.** An attempt to log in/out a job fails when it should have succeeded. The probable cause is that terminals have been TTYSTPed, then a LGOUT or LOGIN JSYS was tried. An attempt to detach the terminal, then logout the job is made. If either of these fail, the job is put in a permanent wait state. JOBPT - Terminal number LSTERR - Reason of the failure
LLIBWK	CHK	LLINKS	**SCTNSF call from sched w/o lock.** DECnet entry point NSP was called from scheduler level when the NSP interlock was locked. This should never happen. CALLER - Address of routine that requested the interlock
LLIDIR	CHK	LLINKS	**Duplicate Interrupt Message Received.** There is a duplicate interrupt message on the unacked interrupt receive queue. This should not happen because the NSP interlock should not release with anything on the receive queue. ELPTR - Pointer to EL block ESPTR - Pointer to ES block MBPTR - Pointer to message block
LLIFNS	CHK	LLINKS	**SCTL passed bad NSPpid.** Session control gave LLINKS a bad ID. This is a coding error in SCLINK, or a memory manager problem. ELPTR - Pointer to the bad ELB
LLIFZM	CHK	LLINKS	**Tried to free zero msg.** FREMSG was requested to free a message. However, the pointer to the message block was zero. This is a coding error in LLINKS.
LLIHTG	HLT	LLINKS	**INIHSH cant get a hash table.** Routine that initializes the LLINKS link hash table failed to get memory for the hash table. If the value for the hash table size is reasonable, this should never fail.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
LLIHTS	HLT	LLINKS	**NSPHTS not set up.** Monitor has a bad value for the hash table size.
LLIIFC	CHK	LLINKS	**Illegal flow control type.** An illegal flow control type was requested on transmit. A higher layer should have checked this. Inspect the stack to find the path that caused the bad value. Submit an SPR and include a dump and the additional data. ELPTR - Pointer to EL block ESPTR - Pointer to ES block MBPTR - Pointer to message block
LLIORC	CHK	LLINKS	**ORC should never be negative.** LLINKS has requested that a message be returned from ROUTER after transmission. ROUTER just returned such a message to LLINKS, but the count of outstanding messages was zero.
LLIPIM	CHK	LLINKS	**PROCXQ found illegal message type.** A message being resent had a bad message type. The message was overwritten while it was waiting on the resend queue. The message type was good when the message was sent the first time. MBPTR - Pointer to message block describing the bad message
LLIQIN	CHK	LLINKS	**Queued interrupt message illegal.** LLINKS was asked to transmit two interrupt messages simultaneously. A maximum of one is allowed. This is a software problem. Submit an SPR if it happens more than once, and include a dump of the system and the additional data. ELPTR - Address of EL block ESPTR - Address of ES block MBPTR - Address of message block
LLIS2S	CHK	LLINKS	**Illegal flow control at PRCRQS.** An illegal flow control type was found at PRCRQS when the receive queue was processed. If a remote node had sent us a bad flow control type, it should have been found by the message parsing routines. Therefore, this should never happen. ELPTR - Address of EL block ESPTR - Address of ES block MBPTR - Message block address
LLITNE	CHK	LLINKS	**Unknown event at NSPEVT.** The caller of the NSPEVT routine supplied a bad event class and type. NSPEVT may be called by SCLINK as well as by LLINKS. The caller's address is on the stack. EVC - Event class EVT - Event type
LLMCIF	CHK	LLMOP	**LLMOP Read Channel Info Failed.** A LLMOP attempt to read the Ethernet channel status failed when the Data Link Layer was called. DLLERC - The error code returned from the DLL.LLMIL1,INF LLMOP,LLMOP Received Invalid Loopback Message. LLMOP received a loopback message that was too short or was improperly formatted. This is a MOP protocol violation by a remote node. MSGLEN - Received message length HIORD - Ethernet address (high order bits) LOORD - Ethernet address (low order bits)

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
LLMILF	INF	LLMOP	**LLMOP Invalid Loopback Function Code.** LLMOP received a loopback message that was neither a loopback reply message or a forward data message. This is a MOP protocol violation by a remote node. FUNCOD - Function code HIORD - Ethernet address of the transmitting node (high order) LOORD - Ethernet address of the transmitting node (low order)
LLMIR1	INF	LLMOP	**LLMOP Received Invalid Remote Console Message.** LLMOP received a remote console message that was too short, too long, or was improperly formatted. This is a MOP protocol violation by a remote node. MSGLEN - Received message length
LLMLXF	INF	LLMOP	**LLMOP Loopback Transmit Failed.** LLMOP was unable to transmit a forward data message. DLLERC - Error code returned from DLL STATUS - Channel status returned from the DLL CHANNEL - Channel on which the failure occurred
LLMMCF	CHK	LLMOP	**LLMOP Declare Multicast Address Failed.** A LLMOP attempt to declare the Assistant Multi-Cast Address failed when the Data Link Layer was called. DLLERC - Error code returned from the DLL
LLMOPF	CHK	LLMOP	**LLMOP Open Portal Failed.** LLMOP failed to open an NI portal with the Data Link Layer. DLLERC - Error code returned from the DLL
LLMRQC	CHK	LLMOP	**LLMOP RB Queue Corrupted.** LLMOP tried to remove an RB queue entry from an empty queue. It is also possible that the RB was not on the queue. RBADDRESS - Address of RB queue entry
LLMRRF	INF	LLMOP	**LLMOP Response Transmit Failed.** LLMOP was unable to send a MOP request message. DLLERC - Error code returned from DLL CHANNEL - Channel on which the failure occurred
LLMRXF	CHK	LLMOP	**LLMOP Resource Failure.** LLMOP was not able to obtain resources from the memory manager.
LLMSB2	CHK	LLMOP	**LLMOP Specify Receive Buffer Failure.** LLMOP could not post a receive buffer to the Data Link Layer. DLLERC - Error code returned from DLL
LLMSCA	INF	LLMOP	**LLMOP Ethernet Channel Address Change - CHAN,ADDR1,ADDR2.** LLMOP was called by NIDDL on change of state. CHANNEL - Channel number
LLMSTC	INF	LLMOP	**LLMOP data link state change.** LLMOP was called by NIDDL on change of state. This message is for information only. No corrective action is required. CHANNEL - Channel number PTRLID - Portal ID STATUS - Status bits

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
LNGLNG	HLT	DISC	**NEWLFP - File going long is already long.** A file is becoming long for the first time. This BUG indicates that the file is already long.
LPRIXC	HLT	LLMOP	**LLMOP Invalid Xmit Complete.** NIDLL called back to LLMOP with a transmit complete event for an RB which is not in Transmit Initiated state. This is software bug. RBSTT - Current RB state UNSTA - Status in the UN block
LPRLXF	INF	LLMOP	**LLMOP Loop Request Transmit Failed.** LLMOP was unable to transmit a forward data message. DLLERC - Error code returned from DLL CHANNEL - Channel on which the failure occurred
LPSIFC	CHK	LLMOP	**LLMOP LPSCBR called with invalid function code.** LLMOP Loopback Protocol Server Call Back Routine was called by the Data Link Layer with an invalid callback function code. This is a software bug. FUNCODE - Function code
MACBTO	INF	DIAG	**DIAG - Close buffer timed out.** DIAG close buffer operation timed out before completion.
MARK1	HLT	PAGUTL	**BADCPG - Not an OFN.** An OFN is in error but the SPT index is not pointing to an OFN. SPTIDX - SPT index COREPG - Core page number
MONBKB	CHK	MEXEC	**Cannot set monitor error interrupt.** Monitor tried to enable interrupts on the monitor error channels. This BUG indicates that the AIC failed. LSTERR - Last process error
MONNEJ	CHK	SCHED	**Nested JSYS without ERJMP.** An illegal instruction trap occurred and the previous context is the monitor but no ERJMP is present following the nested JSYS call. This violates required coding practice because the previous context may have locks that need to be released. FLAGS - Processor flags PC - PC at which faulty nested JSYS was done
MOPIFC	INF	LLMOP	**LLMOP Received an invalid MOP message.** The LLMOP Remote Console Protocol Server received a MOP message with an invalid function code. This is a MOP protocol violation by a remote node. FUNCODE - Function code
MSCAOL	CHK	PHYMSC	**PHYMSC - Online node event while node already online.** SCAMPI told us that this node was coming back on-line but we think that it is already on-line. NODE - Node number CID - Connect ID SBI - System block index

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
MSCAVA	INF	PHYMSC	**PHYMSC - Available message received.** An Available message was received. NODE - Node number CID - Connect ID UNIT - Unit number
MSCBAD	HLT	PHYMSC	**PHYMSC - Bad dispatch from PHYSIO.** PHYMSC was called to perform a function that cannot be performed.
MSCBCN	CHK	PHYMSC	**PHYMSC - Command reference number bad.** The command reference number is invalid. NODE - Node number CID - Connect ID ENDCODE - Packet end code FUNCTION - Command request
MSCBHE	HLT	PHYMSC	**PHYMSC - BHD error bit set.** BHD error bit was set. This implies that the BSD had the wrong length. Something is inconsistent in the state or too much data was sent.
MSCBID	HLT	PHYMSC	**PHYMSC - Bad connect ID from SCAMPI.** A connect response available occurred and a negative or zero connect Id was returned from SCA. CID - Connect ID
MSCBPK	CHK	PHYMSC	**PHYMSC - QOR bad packet.** HSC sent a packet whose command reference number can't be found. NODE - Node number CID - Connect ID ENDCODE - Packet end code CRN - Command reference number
MSCCDF	INF	PHYMSC	**PHYMSC - Connect to disk failure.** A connect failure occurred after an indication that an HSC was present. NODE - Node number ERRCOD - Error code
MSCCRN	INF	PHYMSC	**PHYMSC - Connect did not complete in reasonable timeout.** There was a connect request and no response. NODE - Node number CID - Connect ID INDEX - MSCCID table index
MSCCTF	INF	PHYMSC	**PHYMSC - Connect to tape failure.** There was a failure to connect to the tape controller. NODE - Node number ERRCOD - Error code
MSCCTO	INF	PHYMSC	**PHYMSC - Request HSC disconnect - command timeout.** HSC has not correctly responded to Get Command Status request. NODE - Node number CID - Connect ID
MSCCWM	INF	PHYMSC	**PHYMSC - Controller not in 576 MODE.** HSC controller is not in 576 bytes per sector mode. NODE - Node number CID - Connect ID UNIT - Unit number

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
MSCDIS	INF	PHYMSC	**PHYMSC - Request HSC disconnect.** The messages from the HSC indicate a problem. NODE - Node number CID - Connect ID INDEX - MSCCID table index
MSCDSR	INF	PHYMSC	**PHYMSC - Disconnect request by remote.** Remote node has disconnected. NODE - Node number CID - Connect ID REASON - Reason for disconnect
MSCDWM	INF	PHYMSC	**PHYMSC - Disk not in 576 MODE.** HSC disk is not a 576 bytes per sector disk. NODE - Node number CID - Connect ID UNIT - Unit number
MSCGON	INF	PHYMSC	**PHYMSC - IORB/QOR gone.** PHYMSC had a data structure that pointed to an IORB. It cannot find the IORB on the unit transfer queue. CID - Connect ID IORB - IORB address STATUS - Status of IORB
MSCIDG	CHK	PHYMSC	**PHYMSC - Connect ID gone.** Connect ID is now gone. NODE - Destination node number CID - Source connect ID
MSCILD	HLT	PHYMSC	**PHYMSC - Illegal dispatch from SCAMPI.** A dispatch value from Scampi is illegal. CODE - Dispatch value
MSCILF	CHK	PHYMSC	**PHYMSC - Illegal function at start IO.** Illegal function at call to MSCRIO. FCN - Function
MSCIVC	CHK	PHYMSC	**PHYMSC - Illegal command.** Remote node claimed an illegal command. CHAN - Channel number KONT - Controller number UNIT - Unit number STS - IORB status word
MSCMID	CHK	PHYMSC	**PHYMSC - Missing connect ID.** There is a missing or zero connect ID on call to FNDNDX.
MSCN2S	INF	PHYMSC	**PHYMSC - More drives than table space, excess ignored.** Number of tape drives available exceeds the constant value MTAN. Only MTAN drives can be configured. KDB - KDB address CHN - Channel number
MSCNIR	HLT	PHYMSC	**PHYMSC - IORB ZERO.** PHYMSC found the IORB register zero in a place it did not expect. CID - Connect ID
MSCNRA	CHK	PHYMSC	**PHYMSC - Node response available when not requested.** A connect response available occurred on a node that isn't expected to have an available happen. NODE - Node number CID - Connect ID

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
MSCNUF	INF	PHYMSC	**PHYMSC - Get next unit failed.** SC.SMG failed. NODE - Node number CID - Connect ID ERRCOD - Error code
MSCNXF	INF	PHYMSC	**PHYMSC - Get next unit failed.** Get a next unit failed. All the units on this HSC50 may not be found. NODE - Node number CID - Connect ID ERRCOD - Error code INDEX - MSCCID table index
MSCOLE	CHK	PHYMSC	**PHYMSC - Online failed.** An on-line request failed. This happens when duplicate unit numbers are found. STATUS - Unit status
MSCOLF	INF	PHYMSC	**PHYMSC - Available online failed.** An attempt to put an available unit on-line failed because of a send failure. NODE - Node number CID - Connect ID ERRCOD - Error code
MSCORO	INF	PHYMSC	**PHYMSC - Offline return to online when we were told avail.** A node that indicated an on-line is not available when on-line is tried. NODE - Node number CID - Connect ID CODE - End packet status code
MSCPEI	CHK	PHYMSC	**PHYMSC - Packet end code incorrect.** HSC sent a packet that had a bad packet end code. NODE - Node number CID - Connect ID ENDCODE - Packet end code CRN - Command reference number
MSCPTG	INF	PHYMSC	**PHYMSC - Port went away.** Port has dropped the connection. NODE - Node number CID - Connect ID
MSCQRC	CHK	PHYMSC	**PHYMSC - QOR list clobbered.** QOR list has been clobbered and has a 0. NODE - Node number KONT - Controller number CRN - Command reference number
MSCREJ	INF	PHYMSC	**PHYMSC - Node connection reject.** A connection response available was rejected. The node cannot be reached. NODE - Node number CID - Connect ID
MSCRLD	INF	PHYMSC	**PHYMSC - HSC control reload initiated.** The HSC is not responding correctly. NODE - Node number CID - Connect ID
MSCRLF	CHK	PHYMSC	**PHYMSC - Start or reset failed.** PHYMSC called SCAMPI and was unable to start or reset the remote. NODE - Node number CID - Connect ID ERRCOD - Error code

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
MSCSCF	INF	PHYMSC	**PHYMSC - SETCCH failed to set characteristics.** SETCHH failed to set characteristics. NODE - Node number CID - Connect ID ERRCOD - Error code INDEX - MSCCID table index
MSCSCW	CHK	PHYMSC	**PHYMSC - Send found wrong connect state.** The State of the connection is incorrect for the connect state. Previous states should have caught this unless the state changed during the send. The send should have been done with the channel off. NODE - Node number CID - Connect ID ERRCOD - Error code
MSCSDF	CHK	PHYMSC	**PHYMSC - Send failure.** A message sent to SCAMPI failed and an error will be returned. NODE - Node number CID - Connect ID ERRCOD - Error code
MSCSIF	CHK	PHYMSC	**PHYMSC - Start IO failed.** A call to MSCRIO failed when it was not expected to. UDB - UDB address KDB - KDB address CHAN - Channel number
MSCSOA	HLT	PHYMSC	**PHYMSC - SC.SOA failed.** Interrupts were requested and failed. ERRCOD - Error code
MSCSUF	INF	PHYMSC	**PHYMSC - Set density failed.** Set unit characteristics command failed. NODE - Node number CID - Connect ID CODE - Status code
MSCTMU	INF	PHYMSC	**PHYMSC - Too many units for KDB.** There are more units than space for UDB entries. KDB - KDB address CHN - Channel
MSCUDB	HLT	PHYMSC	**PHYMSC - UDB missing.** We have just set up a unit during initalization and now we can't find it.
MSCUKD	INF	PHYMSC	**PHYMSC - Unknown disk type.** Device type is unknown to the system. NODE - Node number CID - Connect ID
MSSBCM	INF	PHYMVR	BADCMD - **MSCP server bad command.** MSCP server received a command with an illegal or unsupported operation specified. NODE - Node number CID - Connect ID OPCODE - Operation code ERRBIT - Error bits and status of command
MSSCAC	CHK	PHYMVR	**MSCP server can't accept connection.** MSCP server cannot accept a connection. NODE - Node number CID - Connect ID REASON - Reason for failure

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
MSSCFS	HLT	PHYMVR	Server detected no CFS when writing. MSCP server was requested to perform a write operation when CFS was not running or negotiating connections. This may corrupt the file system. NODE - Node number CID - Connect ID
MSSCGL	INF	PHYMVR	MSCP server can't get listener. MSCP server cannot get a listener for connection requests. The server continues to try to get a listener. ERROR - Error code returned by SC.LIS
MSSCID	HLT	PHYMVR	Illegal connect ID index. MSCP server cannot find a SCDB for the given connect ID. NODE - Node number CID - Connect ID
MSSCTO	INF	PHYMVR	PHYMVR - Command timeout. Unknown. A command did not complete in the timeout interval. NODE - Node number CID - Connect ID STATE - Command state
MSSDNQ	HLT	PHYMVR	DMADON - DMA done queue entry not found. A DMA complete interrupt occurred and no commands were found that had a matching buffer name. This indicates a software inconsistency. CID - Connect ID BUFF - 32-bit buffer name
MSSER0	CHK	PHYMVR	IORB done error and error bits 0. An IORB completed with bit IS.ERR set indicating an error. The MSCP server could not find any relevant error. IRBERR - IORB status word
MSSLNM	HLT	PHYMVR	MSCP server listener does not match. The listener index does not match the known index of the listener.
MSSNWO	HLT	PHYMVR	OK2SND - OK to send when not waiting. The MSCP server received notification of okay to send from a node. The node in question was not flagged as waiting for an okay to send. CID - Connect ID
MSSREJ	INF	PHYMVR	MSCP server rejecting connection. The MSCP server is rejecting a connection because the connector cannot be identified due to an SCA failure or because the connector is not on a KL10 processor. NODE - Node number CID - Connect ID ERROR - SCA error code
MSSSBD	INF	PHYMVR	Send failed. A message send failed for an unexpected reason. Connection is shut down. NODE - Node number CID - Connect ID ERROR - SCA error code
MSSSCA	HLT	PHYMVR	MSCP SERVER - Server detected SCA error. MSCP server detected an illegal response from SCA. SCAFNC - SCA function code ARG1 - ARG3 - SCA function arguments

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
MSSSHT	INF	PHYMVR	**MSCP server shutdown node.** MSCP server was forced to shut down a node. NODE - Node number CID - Connect ID STATUS - Connection status ERROR - Last SCA error
MSSSTA	HLT	PHYMVR	**MSCP SERVER - Illegal state.** MSCP server detected an illegal command or connection state.
MSSTML	HLT	PHYMVR	**LISTEN - MSCP server too many listeners.** MSCP server tried to obtain a listener when one already existed. This indicates an inconsistency in the software.
MSSUMP	CHK	PHYMVR	**Unmap buffer failed.** A routine was called to unmap a buffer and failed when it should have succeeded. REASON - Error code
NIDUNF	HLT	NISRV	**Unknown Callback code from Port Driver.** Port driver has called back with either a code in T1 that is not understood or that is not expected to be called back on.
NIJECL	HLT	NIUSR	**Error closing portal.** NISRV returned an error when we tried to close a portal. Error code was not UNRES% (Resource error), which is the only one that may occur. ERROR - Returned error code
NIJIPB	HLT	NIUSR	**Illegal Portal Block.** NIUSR did not find a proper portal block pointer in the job's portal list. JOBPPR - Job's portal list address
NIJPMU	HLT	NIUSR	**Portal List messed up.** NIUSR tried to create a portal and install it in the portal list. According to PLNUM, there were some free spots in the portal list. A search of the list was not able to find a free slot. This is an inconsistency. PRLIST - Portal list address
NISEC6	CHK	D36COM	**Not in section 6.** Code that should be running in section 6 is not. CALADR - Address of routine not in section 6.
NMXTBG	CHK	JNTMAN	**NMXTIM table obsolete.** Table used by NMXTIM is obsolete.
NOCHKR	CHK	SCHED	**CHKR fork blocked.** CHKR fork has not run in a while. Monitor is getting nervous. If the CHKR fork continues to not run for a long time the a CHKRNR BUGHLT will result. CHKDUE - Count of times CHKR was overdue
NODDMP	CHK	SCHED	**DDMP fork blocked.** DDMP fork has not run in a while. Monitor is getting nervous. If the DDMP fork continues to not run for a long time the a DDMPNR BUGHLT will result. DDPDUE - Count of times DDMP was overdue
NODMPF	CHK	MEXEC	**Could not find CI-20 microcode dump program.** KLIPA (IPA20) RAM needs to be dumped. File PS:<SYSTEM>IPADMP.EXE is supposed to be run to do this. However, the file does not exist.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
NOFSEC	HLT	PAGUTL	ASGVAS failure at startup. ASGVAS was called to get a free section for SCA at startup but failed.
NOLODF	CHK	MEXEC	Could not find CI-20 microcode load program. KLIPA (IPA20) RAM needs to be reloaded. File PS:<SYSTEM>IPALOD.EXE is supposed to be run to do this. However, the file does not exist. TOPS-20 is ignoring the CI20.
NOOFN	INF	PAGUTL	ASOF4 - Attempt to create new OFN failed - no more OFNs available. As a result of an OPENF, an attempt was made to create a new OFN. It fails because the system has no more OFNs available for use. The user receives an OPNX10 error. This BUGINF is issued at most once every 30 minutes regardless of how many OPENF attempts are made during the time the OFN space is exhausted.
NTBSUP	CHK	D36COM	Buffer supplied. The routine NTPARM was called to handle a network management parameter. The routine can only handle returns of a single value, but NTMAN had supplied a multi-word buffer.
NTBTSM	CHK	D36COM	Buffer too small. NTMAN requested a show counter operation, but did not supply a buffer large enough to store all the counters.
NTMBCF	CHK	NTMAN	Bad coded field on output. Output for a Show is being formatted, and there was a request to generate a CODED field of more than one byte. This can't be done.
NTMBCL	CHK	NTMAN	Bad counter byte length. While generating output for a numeric field, there was a request to generate an illegal number of bytes.
NTMBDL	CHK	NTMAN	Bad multiple byte length. While generating output for a numeric field, there was a request to generate an illegal number of bytes.
NTMBFP	CHK	NTMAN	Bad format type encountered. While reading a value from the user string, descriptor tables returned an invalid format for this item. The AC "NT" points to the descriptor for this item, and field NTSEQ tells which item is being referred to.
NTMCBL	CHK	NTMAN	Bad Counter Block length. A DECnet Layer returned an invalid length for a Counter Block.
NTMCNO	HLT	NTMAN	Circuit name overrun. More than 16 bytes of data returned into a 16 byte field. Data beyond the buffer was trashed.
NTMDVI	CHK	NTMAN	NMXDSP value illegal. There is a call to a "layer" to obtain or set a value for an item. The routine value in the descriptor block pointed to by NT is illegal. A "layer" is any routine described at NMXDSP.
NTMEFO	CHK	NTMAN	Event function out of range. Event function supplied by a DECnet layer to NMXEVT was out of range.
NTMEOR	CHK	NTMAN	Entity type out of range. While double checking the entity ID before dispatching on it, the value was found to be illegal. Since the value the user supplies is checked at GETBLK, this means that field NXENT was been trashed.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
NTMFOR	CHK	NTMAN	**Format out of range.** While formatting output for a show, the format block for this item was found to have an illegal format type.
NTMFUR	CHK	NTMAN	**Function code out of range.** While dispatching by function code, the code is out of range. Since the function code the user supplies is checked in GETBLK, this means that field NXFNC was trashed in the meantime.
NTMICF	CHK	NTMAN	**Non-counter function in PRSCOU.** There is an illegal function in the PRSCOU routine. NXFNC is wrong.
NTMILN	CHK	NTMAN	**Illegal number size.** When reading a numeric value from the user's string, the format descriptor block for this item has specified an illegal number of bytes to read.
NTMINT	CHK	NTMAN	**Invalid numeric type.** When generating output for a numeric field, something other than decimal, hexadecimal or octal was requested.
NTMKOR	CHK	NTMAN	**Controller out of range in Circuit-id.** The controller field in a line-iD is out of range. The value LD.MAX defines the number of controllers known by D36PAR, and thus by NTMAN. The most likely cause of this bug is a trashed AC. **Note:** A controller is any device driver to which a routes interfaces. It is currently used to define the name of a Circuit/Line, assuming that each Controller controls a single line type.
NTMLTR	CHK	NTMAN	**Line type is out of range.** To determine entries to return (for function .NTSHO), it is necessary to know the line type (CI,NIDTE,...). Other entities (Nodes, Modules) should have this field zero. This field is set by ENTCVT.
NTMNEC	CHK	NTMAN	**No error code with error return.** A routine returned non-skip, but has not given an error code by calling NTExxx. A return to the top level found field NXERR zero.
NTMNTR	CHK	NTMAN	**Node type is out of range.** It is necessary to know the node type (executor, remote, or loop) to select entries to return (for function .NTSHO). Other entities (circuit, lines) should have this field zero. This field is set by ENTCVT.
NTMORE	CHK	NTMAN	**Unrecognized entity type.** An event was received from a DECnet layer, and the entity type is not legal.
NTMSOR	CHK	NTMAN	**Selection criteria is out of range.** Criteria is out of range for selecting items to return (for .NTSHO) dependent on the selection criteria.
NTMSQF	CHK	NTMAN	**Signal queue full.** Signal queue was full when a new signal was logged. This can be caused by a malfunctioning NMLT20 that does not read the signals from the signal queue, or by a DECnet device driver going bad. A signal tells NMLT20 that a device needs attention/reload.
NTNBFS	CHK	D36COM	**No buffer supplied.** Routine NTPARM was called to handle a network management parameter. The caller of NTPARM said that it expects the call from NTMAN to supply a buffer for the parameters to be read from or stored into. None was supplied.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
NTNBUF	CHK	D36COM	**No buffer supplied.** NTMAN requested a show counter operation, but did not supply a buffer to store the counters in.
NVTILS	CHK	TTPHDV	**NRT link in unexpected state.** DECnet called NRT's host service for a link in an unexpected state.
NVTINP	CHK	TTPHDV	**NRT Input to DECnet failed.** An input call to DECnet's SCTNSF entry point failed unexpectedly.
NVTNHB	CHK	TTPHDV	**NRTHBR should never be called.** DECnet has called NRT's host service at its "hiber" address. This should never happen, since NRT always uses asynchronous calls to DECnet.
NVTOUT	CHK	TTPHDV	**NRT output to DECnet failed.** An output call to DECnet's SCTNSF entry point failed unexpectedly.
NVTPCL	CHK	TTPHDV	**Partial Configuration Msg Loss.** NRT's host service failed to send the configuration message in a single DECnet message segment.
NVTSAB	HLT	TTPHDV	**No memory for NRT's SAB.** NRT's initialization code was unable to get resident free space to build its control blocks.
NVTSJB	HLT	TTPHDV	**No memory for NRT's SJB.** NRT's initialization code was unable to get resident free space to build its control blocks.
NVTWWC	CHK	TTPHDV	**Wrong Channel on Connect Wait Wake.** NRT's host service was waked for a circuit that is not the logical link in connect wait state and has no TTY line number associated with it.
NVTWWN	CHK	TTPHDV	**No NRTCWN Connect Wait Wake.** NRT's host service has been waked for a circuit that has no TTY line number associated with it, yet there is no NRB for a logical link in connect wait state.
OFJFBD	HLT	DISC	**OFNJFN - OFNJFN found bad data.** An OFN was found whose bits indicated that it was or was not a secondary index block. SPT04 was found to disagree.
OFNBDB	HLT	PAGUTL	**OFN bad data base.** There are many causes of this BUGHLT. They all indicate some error in the monitors internal OFN data. The cause of the BUGHLT can be found by examining the dump.
PCIN0	CHK	PAGEM	**PAGEM - PC has gone into section 0.** A reference was made to RSCOD or NRCOD in section 0. This should not happen because section 0 code cannot reference data in extended sections. As an expedient, the page being referenced is mapped to section 1 with an indirect pointer. PC - PC PFW - Page fail word
PDBSTA	CHK	PHYSIO	**PHYSIO - Inconsistent state of UDB status bits.** UDBST1 is inconsistent. STATUS - UDB status bits

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
PGRIXM	HLT	PAGUTL	**PGRINI - Boot overlaps resident tables.** Values in BUTPHY indicate that BOOT was left in pages that are expected to be available for resident code or storage areas. This can be the result of a bad monitor build or an attempt to run the monitor with insufficient memory.
PH2IUA	HLT	PHYH2	**Wrong and inactive unit interrupted.** Routine RH2INT was called to handle an interrupt, and it determined that I/O had finished for a controller but that active controller can't be found.
PH2PIX	CHK	PHYH2	**PHYH2 - RH20 returned from the twilight zone.** Routine RH2CHK was called for a periodic check on the channel status, and found that the PI assignment for the channel was not as expected. A second check of the channel status found the correct PI assignment. CHAN - Channel number CONI - Results of final CONI on RH OLD - Results of first CONI on the RH PIA - PI assignment we expected to see.
PHYCPI	INF	PHYSIO	**CI path ignored for Massbus disk.** TOPS-20 is able to access a disk over the CI (through another system's MSCP server) but it already had access to the disk via the MASSBUS. The system ignores the CI path. OCHN - Old channel NCHN - New channel
PHYDCD	INF	PHYSIO	**PHYSIO - Don't-care disk on do-care drive.** A don't-care disk was found on a standard drive. CHAN - Channel number CONT - Controller number or -1 UNIT - Unit number
PHYDCR	INF	PHYSIO	**PHYSIO - Disk being treated as DON'T-CARE.** A don't-care disk was found on a don't-care drive. CHAN - Channel number CONT - Controller number or -1 UNIT - Unit number
PHYDCU	INF	PHYSIO	**PHYSIO - Do-care disk on don't-care drive.** A standard disk was detected on a drive that was declared DON'T-CARE. CHAN - Channel number CONT - Controller number or -1 UNIT - Unit number
PHYNOS	CHK	PHYSIO	**PHYSIO - No serial number for disk drive.** Serial number of the drive is missing. CHAN - Channel number CONT- Controller number UNIT - Unit number
PHYNUN	HLT	PHYSIO	**PHYSIO no unit number.** A unit number was not found for a given CDB, KDB, UDB. This indicates a software problem. CHAN - Channel number KONT - Controller number UDB - UDB address
PIOERR	HLT	APRSRV	**Unvectored interrupt on channel 0.** Monitor received an unvectored hardware interrupt on PI channel 0. This is not supposed to happen. This could be incorrect PI requests generated by faulty hardware.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
PINIC1	HLT	APRSRV	MAPIPG - Page table not in core. A routine was called to map a page into a special address slot. The requested page is not in memory.
PIRACE	HLT	APRSRV	MAPIPG - Called with PI on. This routine uses a MMAP entry that may be used at PI level. To avoid races the PI should be off when it is called. This particular caller did not turn off the PI.
PLKMOD	HLT	PAGEM	Page lock overly decremented. Monitor decremented the lock count of a page past zero. This indicates a software problem.
PPGOFN	HLT	PAGEM	SPHYPT - Destination is OFN. SPHYPG or SPHYPG was given a destination argument which is an OFN. This type of mapping may only be done in non-file page tables. OFN - The OFN
PTPTE1	CHK	APRSRV	Page table parity error. A page table entry has bad parity. The monitor clears the entry and tries again. If it fails repeatedly, PTMPE results. PFW - Page fail word
RCS3XF	INF	LLMOP	LLMOP Transmit Failed. LLMOP was unable to send a forward data message. DLLERC - Error code returned from DLL CHANNEL - Channel on which the failure occurred
RCSIFC	CHK	LLMOP	LLMOP RCSCBR called with invalid function code. Data Link Layer called the LLMOP Remote Console Protocol Server Call Back Routine with an invalid callback function code. This is a software bug. FUNCODE - Function code
RCSPIS	INF	LLMOP	LLMOP Ethernet Periodic Identify-Self. This is a temporary debugging BUGINF. It indicates that the periodic Identify-Self transmission is being performed.
RELINC	HLT	FREE	RELFSP - Bad block being released. This is a free space problem. The block being returned does not fit into the free space. When blocks are returned to the free space pool, a consistency check is performed. The block is merged into existing blocks that follow it in free space. This block overlaps into existing free blocks. It cannot be merged. POOLN - Pool number CALRPC - PC of caller of RELFSP BLKADR - Address of user block
REVLEV	INF	PHYM78	TM78 Microcode is outdated. The TM78 does not have a microcode version that the monitor needs. ACTUAL LEVELS - Actual revision levels in the TM78 MINIMUM LEVELS - Levels required by this monitor
ROUATL	CHK	ROUTER	A routing message contains a start ID greater than we can handle. An adjacent node sent a routing message with the start ID that causes indexing into the per adjacency vector past the end of the vector.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
ROUAWS	CHK	ROUTER	Adjacency block in queue when state is unused. An adjacency block was left in the queue of active adjacencies, but its state is unused.
ROUBCD	CHK	ROUTER	Bad Checksum detected when building routing msg. The internal reachability vector was damaged since the last rebuilding.
ROUBMB	CHK	ROUTER	Bad message block pointer. DNADLL called RTRDLE with a function requiring a message block, and the pointer supplied (in T3) is either 0 or out of range.
ROUBMT	CHK	ROUTER	Bad message type received from the DLL. DLL received a bad message from another node or incorrectly copied a message into the message block.
ROUBSN	CHK	ROUTER	Bad source node in message from NSP. We received a message from NSP to send. However, the source node address is not that of the local Router.
ROUBSZ	CHK	ROUTER	Router circuit block size was zero on a running circuit. The blocksize for a circuit default to RTRBSZ and is updated with information from nodes on the circuit to determine a new minimum blocksize for the circuit. Somehow this ended up as zero.
ROUBTF	INF	ROUTER	Bad Test message format. Received a hello message from a P3 node or a P4 endnode that contained too many bytes of test data.
ROUBTM	INF	ROUTER	Bad Hello or Test message. Received bad test data in a hello message.
ROUCGV	INF	ROUTER	Couldn't get memory for event arg block. DECnet exhausted its free space.
ROUEHB	CHK	ROUTER	No Message Block for Event data. We are trying to read data from an MB to report in an event but the caller failed to supply a message address.
ROUEHM	CHK	ROUTER	No Message Block for Event data. We are trying to read data from an MB to report in an event but the caller failed to supply a message address.
ROUIFS	CHK	ROUTER	Router got through the forward routine without picking a route. RTRFWD got through its Forward process and either did not pick up a route or failed to flag a message for the local node or an unreachable message.
ROUILS	CHK	ROUTER	Illegal Circuit Specified in NSP msg. There is a request to send a message on a particular circuit, however the routing layer never initialized the circuit.
ROUNAV	CHK	ROUTER	An adjacency has no routing vector. A routing vector is built for each routing adjacency when the adjacency block is created. Either we didn't build one or we cleared the pointer to it.
ROUNLN	CHK	ROUTER	Trying to return msg to non-local NSP. We returned a message to the local NSP but the local NSP was not the originator.
ROUNSO	CHK	ROUTER	NSP sent out of range packet. There is a request to forward a packet to a node whose address is outside the range of our routing vector. Either our NSP has given a packet we cannot forward or we have received one from the wire.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
ROURCE	INF	ROUTER	**Bad NI Router list message format.** We have received a router hello message with more than 256 known two-way adjacencies.
ROURFN	INF	ROUTER	**Routing message received from non-routing node.** We received a routing message from a node we believe to be an endnode so we have no vector to store it in count.
ROURML	CHK	ROUTER	**Stored routing message format error in RTRBAV.** We have received a P3 routing message with a negative count of nodes in it or no checksum or a P4 routing message with a negative segment count. Count - Count or checksum
ROUUER	CHK	ROUTER	**Unexpected end of routing message.** Number of bytes in the routing message did not correspond to the length expected. This may be caused by reading too many bytes out of the message without decrementing the byte count read or caused by an improper routing message.
ROUUET	CHK	ROUTER	**Unknown event type in RTNEVT.** We supplied a bad event code.
ROUUOC	CHK	ROUTER	**Unable to obtain count of nodes in Phase IV message.** We received what we think is a routing message that DNLENG said has more bytes than we have read. When we try to read another two DNG2BY it says the count is exhausted.
ROUXNZ	CHK	ROUTER	**R2NCAL called with MB=0.** Somehow MB was trashed in the forward process. It is unlikely to get this far if RTRFWD received a bad MB.
ROUZXT	CHK	ROUTER	**Tried to free msg with MB=0.** FREMSG called to free an MB but was given a zero pointer.
SBXSE0	HLT	SYSERR	**SYSERR called from SEC 0 with ext blk.** SEBCPY/QUESEB/OR SEBCPY with unextended function call address was performed when the SYSERR block was in extended free space. PC - Callers PC to the SYSERR routine
SCABAL	CHK	SCAMPI	**SCA - Connection block already linked.** SCA is linking a connection block onto a system block. However, the connection block's pointers indicate that it is already linked to some other block. NODE - Node number CID - Connect ID FLINK - Address of next connection block BLINK - Address of previous connection block
SCABMT	CHK	SCAMPI	**SCA - Bad message type from remote node.** A bad message type was found on range checking. This shouldn't happen if the port and port driver are working correctly. The message is thrown away. NODE - Node number CID - Connect ID OPCODE - SCS op code received
SCABSF	CHK	SCAMPI	**SCA - Buffer section full.** SCA went to create more buffers and discovered that the section is full. This indicates that buffers are not being returned.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
SCACCD	CHK	SCAMPI	**SCA - Can't cancel datagram buffer.** A sysap performed the "cancel receive datagram" function of SCA, and the port's queue did not contain as many buffers as the system believes it should contain. NODE - Node number CID - Connect ID COUNT - Number of buffers we couldn't get
SCACCI	HLT	SCAMPI	**SCA - Cannot complete initialization.** During the init SCA detected an error it could not recover from. SCADIE is called by the location and the stack points out the faulty phase of init.
SCACFO	HLT	SCAMPI	**SCA - SC.CON received failure from SC.OUT.** SC.CON created a new connection block and then called SC.OUT to check its state. The call should never fail.
SCACGD	CHK	SCAMPI	**SCA - Can't get datagram buffer when reaping.** When reaping a connection block, a buffer count indicates that datagram buffers are queued to the port. However, the port's queue emptied while these buffers were removed. NODE - Node number CID - Connect ID COUNT - Number of buffers remaining to be dequeued.
SCACGM	CHK	SCAMPI	**SCA - Can't get message buffer when reaping.** While reaping a connection block, a receive credit indicates that message buffers are queued to the port. However, the port's queue emptied while these buffers were removed. NODE - Node number CID - Connect ID COUNT - Number of buffers remaining to be dequeued
SCACLB	HLT	SCAMPI	**SCA - Incoming connect_request on closed v.c.** SCAMPI received a connect_request and matched it to a listener. But when SCAMPI tried to queue the connection block to the system block, it found that the vc closed. Since SCAMPI had checked for that state earlier, and this is happening at interrupt level, something unexpected has happened. NODE - Node number
SCACRB	CHK	SCAMPI	**SCA - Can't reclaim buffers.** Based on the return_credit field for this connection, SCAMPI is trying to reclaim buffers from the port's queue. The queue is empty. This reflects confusion about credit, since these buffers should have been queued at some time in the past. NODE - Node number CID - Connect ID at this node COUNT - Number of buffers we couldn't get
SCACSC	CHK	SCAMPI	**SCA - Can't send credit request.** SCA wants to send a credit request, but the connection block already has some other message pending. This reflects some sort of inconsistency, since the state was "open", and the interlock word for credit requests was 0. NODE - Node number CID - Connect ID STATE - BLock state

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
SCADCF	CHK	SCAMPI	**SCA - Datagram buffer creation failure.** SCA detected that the level of buffers maintained was below minimum. The attempt to create more datagram buffers failed. The error code is in T1. Output is given as additional data. ERROR - Error code COUNT - Number of datagram buffers in SCA's pool
SCAEBD	HLT	SCAMPI	**SCA - Error handling buffer deferral request.** SCA was unable to create buffers when running in job 0. This should never happen since job 0 can create pages as needed. NODE - Node number CID - Connect ID ERROR - Error code
SCAFN2	HLT	SCAMPI	**SCA - Can't complete deferred call to SC.DIS.** A sysap called SCAMPI at SC.DIS when the connection block was locked. The connection block is being unlocked, and the request is being processed. SC.OUT has returned failure, indicating that this function can't be performed for the current state. There is no way to return that failure to the sysap, which believes that the disconnect has proceeded normally. The system will crash to determine the cause.
SCAFN3	HLT	SCAMPI	**SCA - Can't complete deferred call to SC.DRQ.** PHYKLP called SCAMPI at SC.DRQ when the connection block was locked. At the time, the incoming packet was legal for the current state of the connection. Now it is not legal. This shouldn't happen, and it is uncertain how to proceed. It is possible to close the vc and continue, but there is a halt to analyze the protocol confusion and fix the bug.
SCALFO	HLT	SCAMPI	**SCA - SC.LIS received failure from SC.OUT.** SC.LIS created a new connection block and then called SC.OUT to check its state. The call should never fail.
SCAMCF	CHK	SCAMPI	**SCA - Message buffer creation failure.** SCA detected that the level of buffers maintained was below minimum. The attempt to create more message buffers failed. ERROR - Error code COUNT - Number of message buffers in SCA's pool
SCAMCR	CHK	SCAMPI	**SCA - Message buffer count was incorrect.** There are no message buffers when the count indicated there are enough. COUNT - Count of buffers we believed we had TOPQ - Pointer to top of message free queue BOTQ - Pointer to bottom of message free queue BUFNUM - Number of buffers requested
SCANBL	HLT	SCAMPI	**SCA - No buffer for online list.** SC.ABF was called to get a buffer for the address list to be used to call SYSAPs when a node comes on-line. Without this list no one can be told when a node comes on-line and hence we cannot run.
SCANLF	CHK	SCAMPI	**SCA - Notice table full.** So many SYSAPs requested notification of nodes that come on and go off-line that the table of notification addresses overflowed.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
SCANMB	CHK	SCAMPI	SCA - **Can't return SCS control message buffer.** A node went off-line, the local node tried to retrieve two message buffers from the port's queue but found the queue empty. NODE - Node number
SCANOC	CHK	SCAMPI	SCA - **Received packet and connection block doesn't exist.** An incoming packet's destination CID doesn't match any connection block. This may reflect disagreement with another node about the state of a previously-existing connection. NODE - Node number CID - Connect ID OPCODE - Op code
SCANPT	HLT	SCAMPI	SCA - **No page for CID table.** SCA called PGRSKD for a page to put its data tables in. The call failed. Nothing can be done without these tables.
SCANSB	HLT	SCAMPI	SCA - **System block has gone away.** SC.DEF found a system block marked as stuck for buffers, but the address of the system block is 0. NODE - Node number
SCANSC	CHK	SCAMPI	SCA - **Negative system count.** SCA was notified of a system going off-line and decremented the count of systems currently on-line. In doing so, the count went negative.
SCAOBI	CHK	SCAMPI	SCA - **Online before initialization done.** A node came on-line before the initialization of SCA was completed.
SCAODI	HLT	SCAMPI	SCA - **Overly decremented CI interlock.** A CION was done when no previous CIOFF had occurred. This leads to an overly-decremented lock.
SCAOF2	CHK	SCAMPI	SCA - **Offline twice for a node.** SC.ERR was called when a system block was already flagged as off-line. While this won't cause an immediate problem, it indicates internal confusion and should be checked. NODE - Node number
SCAPER	CHK	SCAMPI	SCA - **Protocol error.** An incoming message violated the SCS protocol. This message is illegal. Closing the vc eliminates confusion. NODE - Node number CID - Connect ID at this node OPCODE - Op code of incoming packet STATE - State of connection.
SCARTO	CHK	SCAMPI	SCA - **Reap timed out.** A block that is reputable cannot be reaped because either the count of outstanding packets is non-zero or a debugging check failed. After several postponements, these were not corrected. The block is now deleted. NODE - Node number CID - Connect ID at this node STATE - Block state COUNT - Contents of CBNPO (number of queued messages or datagrams)

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
SCASBN	INF	SCAMPI	**SCA - Block state already non-zero.** While trying to set a connection's block state, it is found to be already non-zero. This can happen legitimately under some conditions. NODE - Node number CID - Connect ID OLDSTA - Existing block state NEWSTA - State we're trying to set
SCASCQ	HLT	SCAMPI	**SCA - Can't get connection management buffers.** SCA was notified of a new system coming on-line. It tried to allocate two buffers for connection management, and failed. This indicates that a large number of buffers were allocated at interrupt level, and the process that creates more hasn't run recently. ERROR - Error code from allocation routine
SCATMO	INF	SCAMPI	**SCA - SCA timed out remote node.** SCA sent a message to another node, and did not receive a response within a timeout period. NODE - Node number TIME - Time since we sent timed message
SCAUXR	CHK	SCAMPI	**SCA - Unexpected response.** A connection management response arrived for a particular connection, but the op code is not the expected one. NODE - Node number CID - Connect ID OPCODE - Op code of incoming packet EXPECT - Expected op code for this connection
SCLCBN	INF	SCLINK	**Phase-II buffering not implemented.** Conservative buffering is not yet implemented. We should never have a logical link open to a phase II node.
SCLNZE	CHK	SCLINK	**Passing zero error code to SCMUUO.** Routine that is supposed to store an error code for the user is zero. This is an illegal value.
SCLRIB	CHK	SCLINK	**Bad SCTRIB call from LLINKS.** LLINKS has called SCTRIB for permission to send a message to SCLINK and has passed an invalid SLB address in T1. The data structures for this logical link are inconsistent. ADDR - Bad SLB pointer
SCLSLB	CHK	SCLINK	**SLB bad at FRESLB.** There is no Session Control Job Block (SJB) for this Session Control Link Block (SLB). This error can happen any time during the life of the link after it was actively transferring data. SLBPTR/ pointer to the SLB that lacked a SJB pointer.
SCLSPF	CHK	SCLINK	**SLB self pointers messed up in FNDSLB.** DECnet data structures for this link are inconsistent. If this happens more than once, submit an SPR. CHAN - DECnet channel number SJBPTR - Pointer to the SJB

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
SCLTFJ	CHK	SCLINK	**Freeing SJB with SLB entries existing.** FRESJB was called to free up a SJB. However, there are still active links in use for this SJB. This should never happen, and there is an internal inconsistency in the DECnet data structures. Submit an SPR if this happens more than once. SJBPTR - Pointer to the SJB
SCLTFS	CHK	SCLINK	**Tried to free wrong SLB.** Channel table entry didn't point to the correct SLB. There is an internal inconsistency in the DECnet data structures for this link. SLBPTR - Pointer to the bad SLB
SCLVAS	CHK	SCLINK	**Couldn't get memory.** SCLINK called ASGVAS to assign virtual address space for the node name/address database. Since the requested memory is non-resident, this should always succeed. However, ASGVAS gave a fail return.
SCSA2M	HLT	SCSJSY	**SCSJSY - Attempt to map second PSB.** Some routine mapped a PSB but did not release it, or did not use the correct interlock. The result is that we are trying to map another PSB while we still have the first one mapped. OWNFRK - Number of the fork that did the first map CURFRK - Fork doing the second lock
SCSABF	CHK	SCSJSY	**SCSJSY - Connection abort failure on fork delete.** During the deletion process for a fork we tried to abort the connections it had open. The attempt failed. ERRCOD - Error code returned by SC.DIS
SCSACF	INF	SCSJSY	**SCSJSY - A JSYS call to ASGRES failed.** A call to ASGRES (by the JSYS) failed. With the error code and caller's PC given by the BUGINF, figuring out why it failed is easy. ERRCOD - Error code CALLPC - PC of caller
SCSBDE	CHK	SCSJSY	**SCSJSY - Bad entry type found.** An illegal type of message buffer tried to return. It is now lost. TYPE - Message buffer type BLKADR - Free space block address
SCSCDC	CHK	SCSJSY	**SCSJSY - Cannot delete connect block from fork queue.** We tried to remove a connect block from the owning fork's list of connect blocks. The most likely failure is a +1 return from SCSMPS. This fails only when we map a PSB but do not unmap it. ERRCOD - Error code
SCSFR1	CHK	SCSJSY	**SCSJSY - SCS% fork removing entries that do not belong to it.** It is assumed that only the owning fork can manipulate SCS% in a CB or in its own PSB. FRKNUM - Fork number to be checked CURFRK - Current fork ADDRESS - Address of calling routine

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
SCSNOI	CHK	SCSJSY	**SCSJSY - SCS% cannot receive node online/offline interrupts.** SCA has told the JSYS SYSAP that there are too many SYSAPs and the JSYS is not allowed to see on-line/off-line interrupts. The system can run but many diagnostics will get upset as will anything that uses the JSYS.
SCSPBF	CHK	SCSJSY	**SCSJSY - PSI block build failure.** Routine to build an event block failed. It is very likely that ASGRES did not have the space available. ERRCOD - Error code returned by ONTBLD
SCSUBL	INF	SCSJSY	**SCSJSY - User buffer lost during error recovery.** Bad access to user memory or a failing routine caused SCS to try to place the currently owned user buffer back on the buffer list. The attempt failed and the buffer address was lost. Note that there is no memory loss, the monitor has just forgotten one user buffer address. ERRCOD - Error code CURFRK - Current fork BUFADR - Buffer address
SCTBWK	CHK	SCLINK	**SCTNSF call from sched without lock.** DECnet entry point SCTNSF was called from scheduler level when the Session Control interlock was locked.
SEBINT	CHK	MEXEC	**Unexpected interrupt in SYSERR process.** An unexpected error occurred in the process that handles error logging. The error handler tries to reinitialize the context and resume processing. The stack may be examined for an indication of where the error occurred. ITFPC - PC when error occurred LSTERR - Last error code in fork
SKDFKS	HLT	SCHED	**Illegal scheduler action while fork context setup.** Scheduler was about to perform an action that requires that no fork context is setup. The monitor found that a fork was not negative, which indicates that fork context was setup.
SMGFUL	CHK	PAGEM	**Can't swap multiple pages (drum is full).** The monitor is trying to swap a group of core pages to the drum. There is no space available. The general handling of drum assignments should insure that there are always a few pages available for "critical" assignments such as this case. It is possible that some user program could overtax the normal reserves and cause this failure.
SPGNLK	HLT	PAGEM	**SPHYPG - Page not locked.** SPHYPG or SPHYPT requires a locked physical page to map. The argument given is either not a physical core page or is not locked. PAGE - Offending argument
SPRZR1	INF	SYSERR	**SEBCHK - SPRCNT went to zero.** SYSERR fork keeps a running count of the number of entries made on the error file in SPRCNT. This count is continuous over system reloads and crashes. This BUG indicates that the count has overflowed its one word value.

Table 5 BUGHLT, BUGINF, BUGCHK (Cont)

Name	Type	Module	Message and Explanation
SPRZRO	INF	JSYSA	**SETSPR - SPRCNT was set to zero.** SMON% function .SFSPR (set count of SPEAR entries output) was called with a value of 0. This indicates that the monitor could not get the running count of the number of SPEAR entries output from either the dump file or ERROR.SYS.SPRCNT is a cell which should contain the running number of SPEAR entries made in the ERROR.SYS file over the life of the system.
SPSCHF	HLT	PAGEM	**SPSCH - Destination is file.** A file page identifier has been passed to SPSCH as the destination page. ID - OFN.PN of offending identifier
STROFF	CHK	MSTR	**OFN on mounted structure but STRTAB entry is zero.** SPTH table has the N+1 through NOFN number of files on a particular structure marked as being on a mounted structure, but the STRTAB entry for this structure is zero.
SWPDIR	CHK	PAGEM	**Swap error in directory page.** The monitor detected an error while swapping in a page with the same OFN as the currently mapped directory. The directory is marked. STRX - Structure number
SYENCD	INF	SYSERR	**SYSERR - Missing code for error type.** User forgot to supply a code type for the error entry. JAWBONE - Job number, internal index JOBPNM - Job program name
TM8REW	INF	PHYM78	**PHYM78 - Spurious rewind interrupt.** TM78 gave a spurious rewind interrupt. CHANNEL - Channel number CONTROLLER - Controller number
XBLTAL	HLT	APRSRV	**XBLTA asked to copy too much.** XBLTA was called with a 'length to BLT' of more than one section. It is unlikely that the caller really intended to copy this much. LENGTH - Number of words XBLTA was asked to copy
XTRAPT	HLT	DISC	**NEWLFT - EXTRA PAGE TABLE IN LONG FILE.** The monitor is trying to create a new file section in a long file. This bughlt indicates that the page table slot in the super PT already contains a pointer to a second level PT. This indicates a race of some kind when a a new page table is created.

COMPUTER INTERCONNECT

Table of Contents

COMPUTER INTERCONNECT

DOCUMENTATION
Refer to the CI20 Reference Manual (EK-OCT20-TM) for functional/logic/installation descriptions.

The Print Set can be ordered as follows.

Order Number	Print Set
MP01903	CI20-A (KL10-E)
MP01906-01	CI20-B (KL10-D)
MP01909-01	CI20-C (KL10-R)

CI CARD CAGE

REAR PANEL CONNECTORS

CARD CAGE·

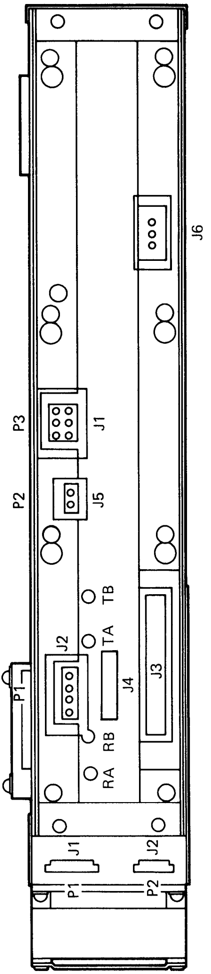
J1	DC POWER +5 0 VOLTS, GND
J2	DC POWER −5.2 VOLTS, GND
J3	PLI
J4	FOR NIA USE ONLY
J5	VOLTAGE MONITOR FOR +5.0 VOLTS
J6	VANE SWITCH
TB	TRANSMIT PATH B
BA	TRANSMIT PATH A
RB	RECEIVE PATH B
RA	RECEIVE PATH B

FAN SUBASSEMBLY.

J1	VANE SWITCH
J2	FAN AC

REAR VIEW

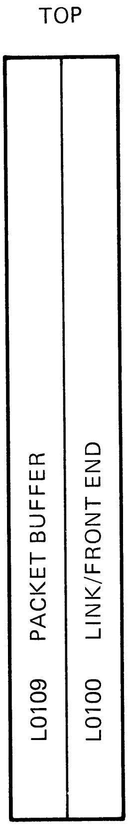
TOP



BOTTOM

MR-14240

INTERNAL SLOT ASSIGNMENTS



CI CARD CAGE
(OPENED FRONT DOOR VIEW)

MUL DECAL
(LOCATED ON SIDE DOOR)

MODULE	LOCATION
L0109	LEFT
L0100	RIGHT

MR-14264

MODULE LOCATIONS
RH/DTE/CI/NIA

The following CI modules are located as follows.

Slot	Module
13	M3001 EBus Interface/Port ALU
14	M3002 Port Microprocessor Control
15	M3003 CBus/PLI Interface
16}	
17}	Blank Module Assembly
18}	

Refer to Volume I Diagrams MULs for all other slot assignments.

CPU

Slot	Module
31	M8532-YA PI BOARD PIC

POWER SUPPLY SPECIFICATIONS

The following voltage measurements are to be made from the backplane of the option.

POWER SUPPLY TYPE	OUTPUT	TOLERANCE MAXIMUM	TOLERANCE MINIMUM	MAXIMUM RIPPLE IN MILLIVOLTS
744/7440	+5	+5.05	+4.95	150

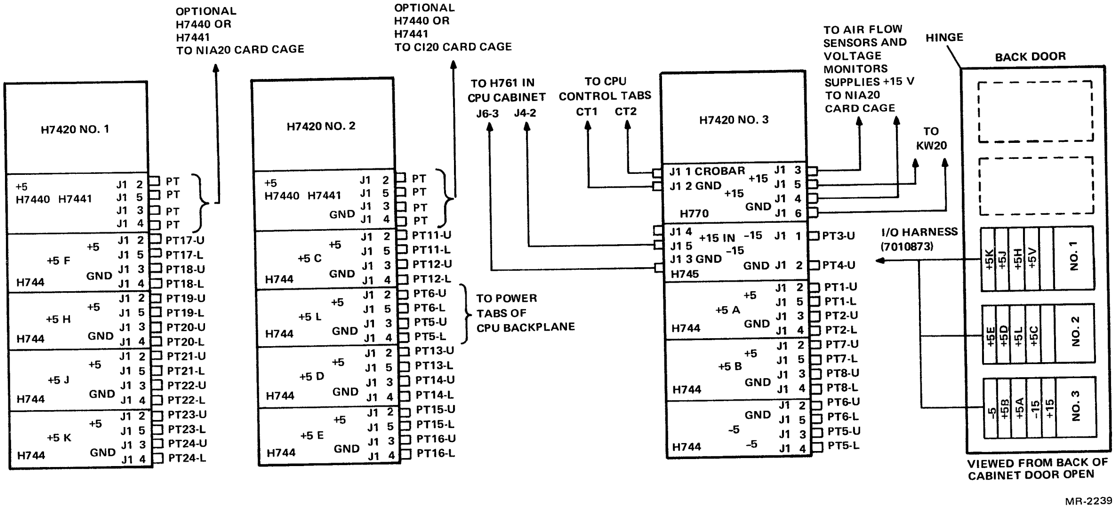
SWITCH SETTINGS
L0100 LINK/FRONT END MODULE

0=OFF (CLOSED)
1=ON (OPEN)

Node	Switch Setting S1	S2	S3	S4	S5	S6	S7	S8
0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0
3	1	1	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0
5	1	0	1	0	0	0	0	0
6	0	1	1	0	0	0	0	0
7	1	1	1	0	0	0	0	0
8	0	0	0	1	0	0	0	0
9	1	0	0	1	0	0	0	0
10	0	1	0	1	0	0	0	0
11	1	1	0	1	0	0	0	0
12	0	0	1	1	0	0	0	0
13	1	0	1	1	0	0	0	0
14	0	1	1	1	0	0	0	0
15	1	1	1	1	0	0	0	0

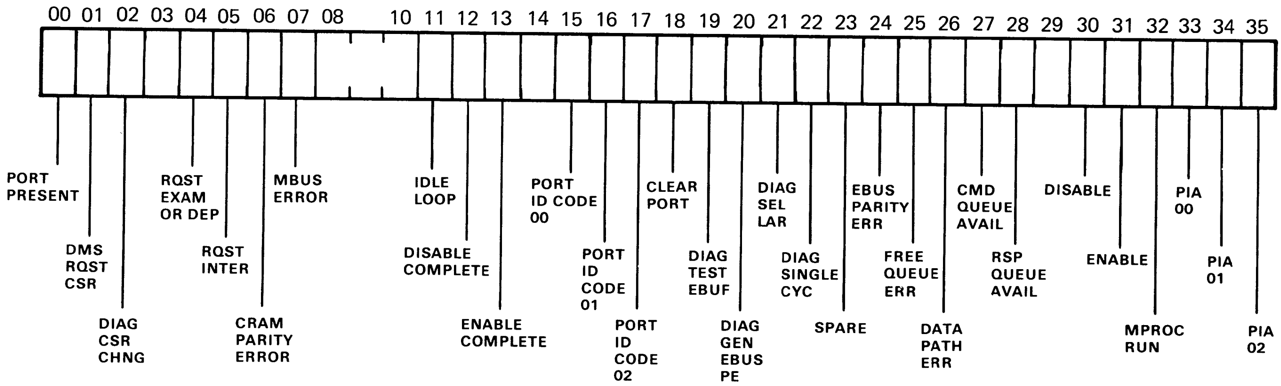
DC VOLTAGE MONITOR BOARD 5414506-01

Switch 1 should be on.
All other switches should be off.



COMPUTER INTERCONNECT

Control and Status Register (CSR) Bit Map



MR-14237

CSR BIT DEFINITIONS

BIT	NAME	DEFINITION
00	PORT PRESENT	Indicates to the KL10 that the port is present (installed and powered-up).
01	DIAG RQST CSR	When set, this diagnostic bit indicates that the port has requested access to the CSR.
02	DIAG CSR CHNG	This diagnostic bit indicates that the contents of the CSR have changed since it was last read by the port microprocessor.
03	UNUSED	Not used by either the port microprocessor or the KL10.
04	RQST EXAM OR DEP	Used by the port microprocessor to request an EBus interrupt on PI level 00 (Examine or Deposit function). The setting of this bit immediately generates the interrupt request.
05	RQST INTERRUPT	Used by the port microprocessor to request an EBus interrupt on PI levels 01 through 07. The setting of this bit immediately generates the interrupt request.
06	CRAM PAR ERR	Indicates that a control RAM (CRAM) parity error is detected. If this bit is set, the port microprocessor is immediately halted and RQST INTERRUPT (CSR bit 05) is set. A hardware nonvectored (40 + 2n) interrupt is forced. A CRAM PAR ERR may be forced to halt the port microprocessor at a specific location (break point). The port microprocessor cannot be restarted (CSR bit 32 set) until this bit is cleared.
07	MBUS ERR	Indicates that more than one MBus driver has turned on at the same time. That is, more than one set of port logic is trying to drive the MBus at the same time. If this bit is set, the port microprocessor is immediately halted and RQST INTERRUPT (CSR bit 05) is set. A hardware nonvectored (40 + 2n) interrupt is forced. The port microprocessor cannot be restarted (CSR bit 32 set) until this bit is cleared.
08	UNUSED	Neither the port microprocessor or the KL10 use this bit.
09	UNUSED	Neither the port microprocessor or the KL10 use this bit.
10	UNUSED	Neither the port microprocessor or the KL10 use this bit.
11	IDLE LOOP	Indicates the port microprocessor is in the Idle Loop, and is not "hung" in some other microcode routine.
12	DISABLE COMPLETE	Informs the KL10 that the port microprocessor placed itself in the DISABLED state.
13	ENABLE COMPLETE	Informs the KL10 that the port microprocessor placed itself in the ENABLED state.

CSR BIT DEFINITIONS (Cont)

BIT	NAME	DEFINITION
14	UNUSED	Neither the port microprocessor or the KL10 use this bit.
15 16 17	PORT ID CODE 00 PORT ID CODE 01 PORT ID CODE 02 Hardwired so that:	Three-bit PORT IDENT CODE field. Informs software that this is a CI20 port and not an RH20 controller. 00 = 0 01 = 1 02 = 1
18	CLEAR PORT	When set by the KL10, this bit resets the port. The microprocessor is halted and all pertinent registers and control logic are placed in a reset state. The bit clears itself after the reset function is completed.
19	DIAG TEST EBUF	This diagnostic bit enables the KL10 to do an EBus interface loopback function by loading and reading the EBus buffer (EBUF). If the port is not running (CSR bit 32 is reset) and this bit is set, then a KL10: DATAO loads EBus data into the EBUF. DATAI places EBUF data on the EBus.
20	DIAG GEN EBUS PE	This diagnostic bit enables the KL10 to test the EBus parity checker by forcing it to decode an EBus parity error. When this bit is set, EBUS PAR ERR (CSR bit 24) is also set on the same CONO, assuming there was no real EBus parity error.
21	DIAG SEL LAR	This diagnostic bit enables a KL10 DATAI to read the CRAM address, in the Latch Address Register (LAR). If this bit is set and bits 19 and 32 are reset, then the DATAI causes the LAR contents to be asserted on EBus D01-D12.
22	DIAG SINGLE CYC	This diagnostic bit enables the port microprocessor to be single cycled. If this bit is set and the KL10 sets MPROC RUN (CSR bit 32), the port microprocessor runs one microcycle and halts. MPROC RUN is cleared when the microprocessor halts. The current address to be executed is fetched from the RAM Address Register (RAR). The next address to be executed is stored in the LAR at the completion of the microcycle. The KL10 must read the address from the Latch Address Register (LAR) and load it into the RAR before executing the next single cycle.
23	SPARE	Reserved for future software use.
24	EBUS PARITY ERR	When read by the KL10, this bit indicated that an EBus parity error was detected. When written as a "1" by the KL10, this bit clears itself and CRAM PARITY ERR (CSR bit 06).
25	FREE QUEUE ERR	Used by the port to inform the Port Driver that there are no free queue entries available on either the Datagram Free Queue or the Message Free Queue.

CSR BIT DEFINITIONS (Cont)

BIT	NAME	DEFINITION
26	DATA PATH ERR	Informs the Port Driver that the port microprocessor detected an error in the DMA data path.
27	CMD QUEUE AVAIL	Used by the Port Driver to inform the port that it placed a command queue entry on a previously empty command queue.
28	RESP QUEUE AVAIL	Used by the port to inform the Port Driver that it placed an entry on the previously empty Response Queue.
29	UNUSED	Not used by either the port microprocessor or the KL10.
30	DISABLE	Used by the Port Driver to tell the port to place itself in the DISABLED state (set CSR bit 12).
31	ENABLE	Used by the Port Driver to tell the port to place itself in the ENABLED state (set CSR bit 13).
32	MPROC RUN	When set by the KL10, this bit causes the CRAM Control Register to reset and enables the port microprocessor clocks. The port starts cycling at the address in the RAM Address Register (RAR). The next and subsequent addresses are fetched from the Am2910 sequencer.
33 34	PIA00 PIA01 PIA02	Three-bit KL10 EBus Physical Interrupt Assignment (PIA) field (PI level 01 35 through 07).

Port Control Block Content (See Figure)

Word	Description
0	Buffer descriptor table (BDT) starting address: The KL10 physical memory address of the first word of the buffer descriptors. Buffer descriptors contain the information needed to tell the port where and how to access a data buffer in KL10 memory.
1	Message free queue entry length: The maximum number of words allowed in an entry on a message free queue. This is a software restriction.
2	Datagram free queue entry length: The maximum number of words allowed in an entry on a datagram free queue.
3	Reserved
4-24	Queue Interlock words, FLINKs, and BLINKs.
25-28	Reserved
29-33	Port error words 0, 1, 2, 3, and 4. The port writes the error words when it encounters a fatal error, writing as much information as possible directly into KL10 memory.
34	PCB base address: The KL10 physical memory address of PCB word 0. The CI20 has no other way to find the PCB.
35	The priority interrupt (PI) level assigned to the CI20.
36	Channel logout word 1 address.
37	Channel command word (CCW): The port writes a CCW-style word in this location to transfer data over the KL10 CBus. The port driver software writes a channel jump word in the EPT location corresponding to the RH20 position that the CI20 occupies.

PORT CONTROL BLOCK

0	BUFFER DESCRIPTOR TABLE STARTING ADDRESS
1	MESSAGE FREE QUEUE ENTRY LENGTH
2	DATAGRAM FREE QUEUE ENTRY LENGTH
3	RESERVED
4	COMMAND QUEUE 3 INTERLOCK
5	COMMAND QUEUE 3 FLINK
6	COMMAND QUEUE 3 BLINK
7	COMMAND QUEUE 2 INTERLOCK
8	COMMAND QUEUE 2 FLINK
9	COMMAND QUEUE 2 BLINK
10	COMMAND QUEUE 1 INTERLOCK
11	COMMAND QUEUE 1 FLINK
12	COMMAND QUEUE 1 BLINK
13	COMMAND QUEUE 0 INTERLOCK
14	COMMAND QUEUE 0 FLINK
15	COMMAND QUEUE 0 BLINK
16	RESPONSE QUEUE INTERLOCK
17	RESPONSE QUEUE FLINK
18	RESPONSE QUEUE BLINK
19	MESSAGE FREE QUEUE INTERLOCK
20	MESSAGE FREE QUEUE FLINK
21	MESSAGE FREE QUEUE BLINK
22	DATAGRAM FREE QUEUE INTERLOCK
23	DATAGRAM FREE QUEUE FLINK
24	DATAGRAM FREE QUEUE BLINK
25	RESERVED
26	RESERVED
27	RESERVED
28	RESERVED
29	PORT ERROR WORD 0
30	PORT ERROR WORD 1
31	PORT ERROR WORD 2
32	PORT ERROR WORD 3
33	PORT ERROR WORD 4
34	PCB BASE ADDRESS
35	PI LEVEL
36	CHANNEL LOGOUT WORD 1 ADDRESS
37	CHANNEL COMMAND WORD
38	RESERVED TO PORT

MR-14247

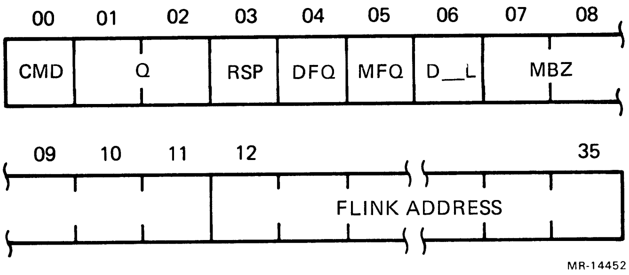
Port Control Block Content (Cont)

Word	Description
38	Reserved to the port microcode.

Error Words 0,1 (words 29,30) are written by the port when it encounters fatal errors associated with queue manipulation. This error reporting strategy requires the port to write as much information as possible directly into the host memory. This approach requires the smallest subset of port hardware and microcode to be working to report these errors.

The information in these words provides enough data for the port driver to determine the type of error and where the error occurred. When the error is detected, the port writes the contents of the error words in the PCB, enters the disabled state, and generates a host interrupt.

The format of error word 0 is shown in the following figure and described in the following table.

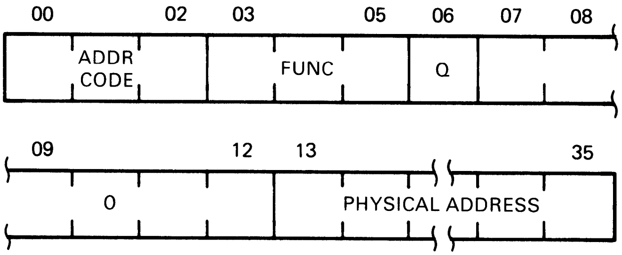


Bit Map, Error Word 0

Error Word 0 Bit Descriptions

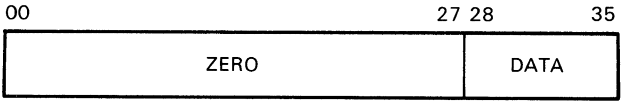
BITS	NAME	DESCRIPTION
0	CMD	Error occurred while touching a command queue entry. The queue with the error is in QUEUE.
1-2	QUEUE	The command queue that had the error. These bits are valid only if the CMD bit is on. 00 = CMD QUEUE 0 01 = CMD QUEUE 1 10 = CMD QUEUE 2 11 = CMD QUEUE 3
3	RSP	Bit is on if error occurred while port was trying to build a response queue entry.
4	DFQ	Bit is on if error occurred while port was touching a command on the datagram free queue.
5	MFQ	Bit is on if error occurred while port was touching a command on the message free queue.
6	D_L	Bit is on if error occurred while port was linking a command to a queue. Bit is off if error occurred while port was delinking a command from a queue. Bit is valid only with bits 0,4, and 5.
7-11	MBZ	Bits are zero.
12-35	FLINK ADR	Address of the FLINK word of the queue entry in question.

Error word 1 (word 30) contains the API function word that the port processor used to access memory when the memory error occurred. This word is written here in the same format as it appeared on the EBUS. The format of this word is given in the following figure.



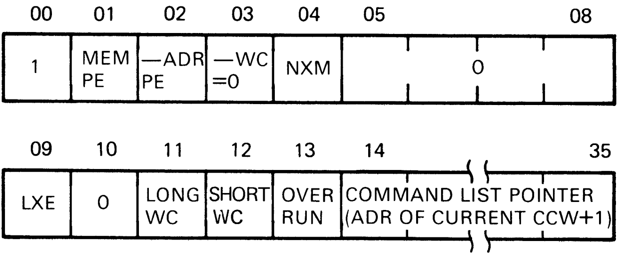
MR-14257

Error word 2 (word 31) contains the register data on transmitter or receiver spurious attention. The format of error word 2 is given in the following figure.



MR-14252

Error word 3 (word 32) contains the channel logout word 1 written by the port on any kind of channel error detected during or immediatly after a DMA transfer. The format of error word 3 is given in the following figure and described in the following table.

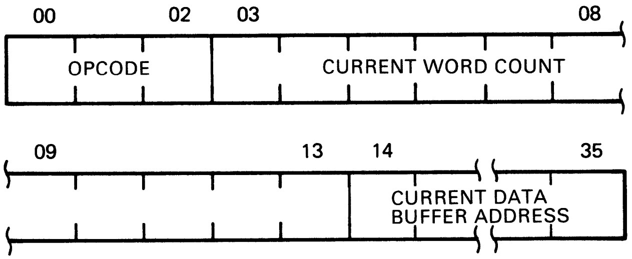


MR-14253

Error Word 3 Bit Descriptions

Bits	Name	Description
01	MEM PE	Memory parity error.
02	-ADR PE	Not address parity error.
03	-WC=0	Channel word count did not = 0 when channel did a store to EPT.
04	NXM	Channel reference did not exist in memory.
09	LXE	Error detected after port term transfer, channel aborts next transfer.
11	LONG WC	Port completed transfer, but word count in CCW not reached.
12	SHORT WC	Channel transferred data specified by CCW, but port still has data.
13	OVER RUN	If device read, port sent data but channel buffers were full. If device write, port req data but channel buffers were empty

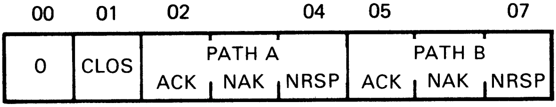
Error word 4 (word 33) contains channel logout word 2 written by the port on any kind of channel error detected during or immediately after DMA transfer. The format of error word 4 is given in the following figure.



MR-14254

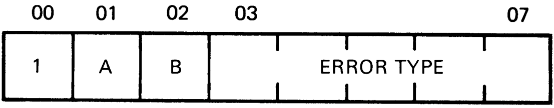
STATUS FIELD

The STATUS field is updated by the port when it builds a response queue entry. The various valid values of the STATUS field are defined below. Note that bit 0 of the STATUS field defines the definition of the remaining bits.



MR-14259

BIT	NAME	DESCRIPTION
1	CLOS	A packet had a retry failure on a path but was transmitted successfully on the other path. The path that failed and the type of failure is indicated in the Path bits. The indicated path is also marked as being bad in the VCDT (Virtual Circuit Descriptor Table).
2	PATH A ACK	The packet was ACKed on this path.
3	PATH A NAK	The packet was NAKed at least once on this path.
4	PATH A NRSP	The packet received No ReSPonse at least once on this path.
5	PATH B ACK	The packet was ACKed on this path.
6	PATH B NAK	The packet was NAKed at least once on this path.
7	PATH B NRSP	The packet received No ReSPonse at least once on this path.



MR-14258

BITS	NAME	DESCRIPTION
1	PTH_A	The error is associated with path A.
2	PTH_B	The error is associated with path B.
3-7	ERROR TYPE	

NO PATH ERRORS

(402)	ERROR = 1	=> Access Control violation.
(404)	ERROR = 2	=> Invalid Buffer Name.
(406)	ERROR = 3	=> Buffer Length violation.
(410)	ERROR = 4	=> Packet size violation.
(414)	ERROR = 6	=> Local unrecognized command.
(416)	ERROR = 7	=> Internal port hardware error.
(420)	ERROR = 10	=> Invalid Remote port.
(422)	ERROR = 11	=> CRC error reported on received packet.
(424)	ERROR = 12	=> No legal path.
(426)	ERROR = 13	=> Command not legal in disabled state.
(430)	ERROR = 14	=> PLI data PE in SRC byte.
(432)	ERROR = 15	=> PLI data PE in OPC byte.
(434)	ERROR = 16	=> PLI data PE in body.
(436)	ERROR = 17	=> Port disabled during processing.

Path B Errors

(502)	ERROR = 41	=> Remote unrecognized command
(504)	ERROR = 42	=> Virtual Cicuit closed
(506)	ERROR = 43	=> Retries Exhausted (NAK)
(510)	ERROR = 44	=> Retries Exhausted (NRSP)
(512)	ERROR = 45	=> Transmitter Timeout

PATH A Errors
=============

(602) ERROR = 101 => Remote unrecognized command
(604) ERROR = 102 => Virtual Cicuit closed
(606) ERROR = 103 => Retries Exhausted (NAK)
(610) ERROR = 104 => Retries Exhausted (NRSP)
(612) ERROR = 105 => Transmitter Timeout

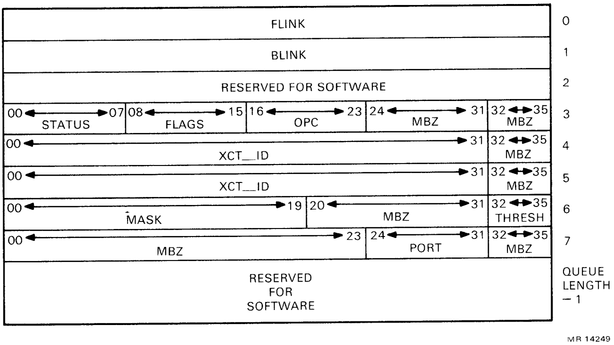
PATHS A,B Errors
================

(704) ERROR = 142 => Virtual Cicuit closed
(706) ERROR = 143 => Retries Exhausted (NAK)
(710) ERROR = 144 => Retries Exhausted (NRSP)
(712) ERROR = 145 => Transmitter Timeout

PORT PERFORMANCE MONITORING

The port microcode implements several counters that are under the control of the port driver. The command queue entry Set Counters (SETCNT) allows the port driver to point and/or clear the counters. It also allows the port driver to enable or disable the event counting. There is a mask used to control the loading and enabling of the various event counters. For each counter, there are 2 bits in the mask; the first bit enables the counting of the event, and the second bit controls the clearing of the event counter. The port driver may instruct the port to count events for a specified port or a cumulative count for all ports.

The format of the SETCNT command is:



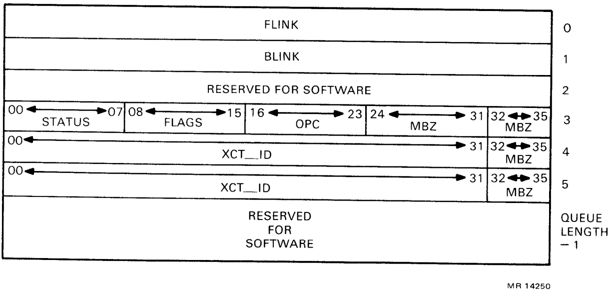
WORD:BITS	NAME	DESCRIPTION
3:16-23	OPCODE	OPCODE = 201 octal (SETCNT).
6:0-19	MASK	This is the 18 bit mask used to control the enabling and loading of the counters.
6:0	PTH_A ACK	If on, count ACKs received on Path A.
6:1	PTH_A ACKC	If on, clear the counter.
6:2	PTH_A NAK	If on, count NAKs received on Path A.
6:3	PTH_A NAKC	If on, clear the counter.
6:4	PTH_A NRSP	If on, count NO_RSPs received on Path A.
6:5	PTH_A NRSPC	If on, clear the counter.
6:6	PTH_B ACK	If on, count ACKs received on Path B.
6:7	PTH_B ACKC	If on, clear the counter.

6:8	PTH_B NAK	If on, count NAKs received on Path B.
6:9	PTH_B NAKC	If on, clear the counter.
6:10	PTH_B NRSP	If on, count NO_RSPs received on Path B.
6:11	PTH_B NRSPC	If on, clear the counter.
6:12	DG DISCARDED	The count of discarded datagrams because of no DGFree Queue entries.
6:13	DG DISC CLR	If on, clear the counter.
6:14	XMT CNT	Count the packets transmitted to the designated port.
6:15	XMT CLR	If on, clear the counter.
6:16	RCV CNT	Count the packets received from the designated port.
6:17	RCV CLR	If on, clear the counter.
6:18	ERR_CNTR_CLR	If on, clear all error counters (see CNTRD response).
6:19	SET_THRESH	If on, load Port Recoverable Error Threshold value.
6:32-35	THRESH_VAL	Value to load for Port Recoverable Error Threshold.
7:24-31	PORT	This is the designated port for which the above counters are tracked. If the port value is set to 255, then the counting is done for all ports.

If the R (response) bit is set in the Set Counters command (SETCNT) it is placed on the Response Queue instead of the DGFree Queue as a counters Set (CNTSET) command. The format for a Counters Set (CNTSET) command is the same as the Set Counters (SETCNT) command.

Every time the port enters the Enabled state, it clears all the counters and sets the PORT field to the "all ports" value. The port driver reads these counters, with a Read Counters (RDCNT) command. This command returns the information in the various counters.

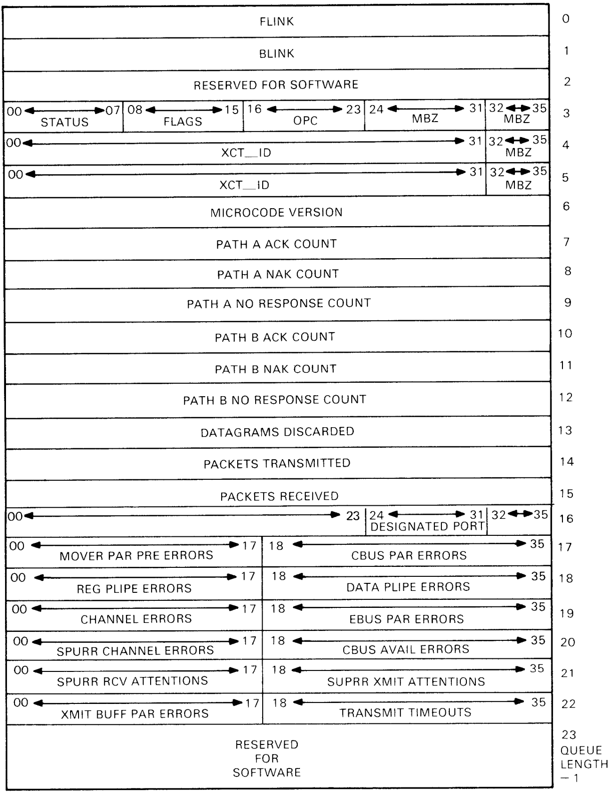
The format of a Read Counters (RDCNT) command is:



WORD:BITS	NAME	DESCRIPTION
3:16-23	OPCODE	OPCODE = 202 octal (RDCNT).

The port will always generate a Counters Read (CNTRD) response to the Read Counters (RDCNT) command.

The format of the Counters Read (CNTRD) response is:



MR 14251

WORD:BITS	NAME	DESCRIPTION
3:15	ERROR	This bit is set if the CNTRD was generated as a result of a Planned CRAM Parity Error (see KLCI Error spec).
3:16-23	OPCODE	OPCODE = 202 octal (CNTRD).

Words 17-22 are called the Port Recoverable Error Counters. The errors have a threshold initially set to 5 by the port during initialization. The threshold can be changed by the port driver with the SETCNT command. The threshold has a value range of 0-17.

DIAGNOSTICS
The following abstracts describe the CI20 diagnostic programs:

- DFPTA Port Basic Device Diagnostic
- DFCIA CIA Functional Diagnostic
- DFSXA KL10 Channel/DTE20-Interaction Test
- DFCIB CI20 Exerciser
- DFCIC PDP10 Computer Intercconnect (CI) Responder

DFPTA - PROGRAM ABSTRACT
DFPTA is the basic device diagnostic for the CI20/NI20 controller on the KL10. It detects and isolates hard faults or faults in the device. Manufacturing uses DFPTA to repair CI20/NI20 modules. Field Service uses DFPTA to verify port operation or isolate a fault to a replaceable module.

DFPTA tests one or two CI20/NI20 controllers. Each controller consists of three port modules in RH20 slot 5 or 7.

DFPTA isolates faults to a network of failing chips. Each network is a set of chips functionally related. Typically, several networks are printed, with the first network being the most probable.

DFPTA consists of two major program sections.

- Test section - Contains all hardware tests. This section is used to debug a module or verify a port.

- Debug section - Allows detailed manipulation or inspection of the port.

DFCIA - PROGRAM ABSTRACT
DFCIA is a functional diagnostic used to check the functionality of a CI20 port consisting of a three port modules, Packet Buffer module, Link module, CI cables, and Star Coupler. It isolates faults to one of three items -

1. 3 Port modules
2. Packet Buffer module
3. Link module, cables, and Star Coupler.

DFCIA tests the CI port consisting of three port modules in RH20 slot 7 and a Packet Buffer module and a Link module in a separate card cage.

DFCIA consists of two major program sections.

1. Test section - Contains all hardware tests. This section functionally checks the port or isolates an actual problem.

2. Debug section - Allows detailed manipulation or inspection of the port.

DFSXA - PROGRAM ABSTRACT
"MD-10-DFSXA" is the KL10 Channel/DTE20 interaction test for the KL10 computer system. It is a software tool that allows the engineer to interactively test all data channels into and out of the KL10's internal memory. The program tests up to eight KL10 I/O Channels (RH20 MASSBUS Controllers or NIA20 or CI20 Ports) along with up to four front-end PDP-11 Channels (DTE20 KL10 to PDP-11 interfaces).

The RH20 Channels can be tested in one of two operating modes:

1. In internal loopback mode (deviceless), or

2.. By writing or reading data to or from either an RP04 disk pack or a TU16 magtape drive.

The NIA20 or CI20 Ports are tested in much the same way as an RH20 Channel in internal loopback mode. Data transfers occur from KL10 memory over the CBUS back to memory over the EBUS interface (a "write") or in the opposite direction (a "read").

The DTE20 Channels can be tested in one of two possible operating modes:

1. With only that minimum PDP-11 resident software required to transfer data between "11" and "10" core, or

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 allow the engineer to evaluate system performance and help detect and isolate interactive system problems to the faulty subsystem. This is the only program in the set of DECSYSTEM-20 diagnostics that provides simultaneous testing of both the PDP-11 front-ends and the KL10 I/O channels.

DFCIB - PROGRAM ABSTRACT

The Computer Interconnect Exerciser (DFCIB) runs under the Diagnostic Monitor (KCSUB) in user mode only. It exercises the entire CI cluster or a desired subset of it. It guarantees the integrity of the CI and isolates faults to a failing node. The DFCIB is the driver (controller) in a two-process system, the driven process (responder) is implemented by each node in the CI cluster. The DFCIB resident in the KLIPA (IPA20) communicates with responders using the Cluster Test Protocol (CTP) and services provided under the Systems Communications Service (SCS). It ensures compliance with the CI Spec for ID, Message, Datagram, and Data Transfer functions. It also provides System Performance Data to users.

DFCIC - PROGRAM ABSTRACT

The CI Responder is the slave portion of a master/slave pair of cooperating programs. Its function is to receive and act according to instructions from the master process referred to as a Controller Process. In the CI environment, the Controller Process may be the CI Node Tester (CINT) or a copy of the CI Exerciser Program (CIE) running in some CI Node.

The CI Responder is a user-mode-only Program that runs under the Tops-20 Operating System (Release 6.0 or newer). The Responder runs during normal timesharing as an Operator Job and does not require exclusive use of any System Resources.

Command and Response Packets are passed between Controller and Responder Processes using the Cluster Test Protocol (CTP). The System Communication Service (SCS) of the Operating System sends CTP messages over the CI.

DESKEWING/ADJUSTMENT PROCEDURE

1. Deskew the port modules by performing the following steps. A Tektronix 475 or equivalent (100 Mhz min) oscilloscope is needed.

2. Connect channel 1 of the oscilloscope to MTR MBOX CLK H, 4D33P1, on the CPU backplane. Use a ground clip.

3. Set the time base to 20 ns.

4. Set channel 1 vertical gain to 0.5 V/division. Set the ground reference to 1.3 volts above horizontal center level of oscilloscope. (MTR MBOX CLK H is an ECL signal.)

5. Set the oscilloscope sync to positive external.

6. Connect external sync input to CHT0 H, 4B09K1 on the CPU backplane. Use a ground clip.

7. Connect channel 2 to CDS1, EBUS CLK L, 2A15F1, on the I/O backplane. Set the channel 2 vertical gain to 0.5 V/division. Use ground clip. To measure TTL voltages, set the ground reference to 1.5 volts below horizontal center line of oscilloscope.

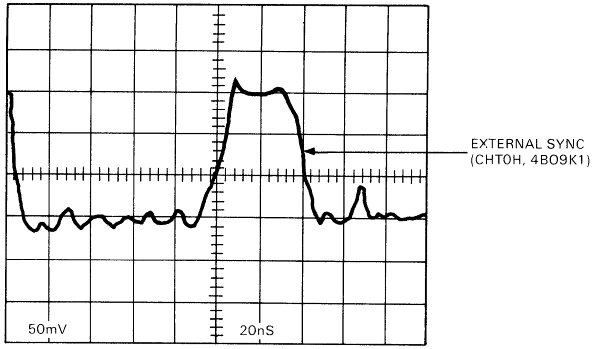
8. Push the Trigger View Switch of the oscilloscope and display the external sync. Adjust the display, so that the rising edge of the external sync aligns with the vertical center line of the oscilloscope.

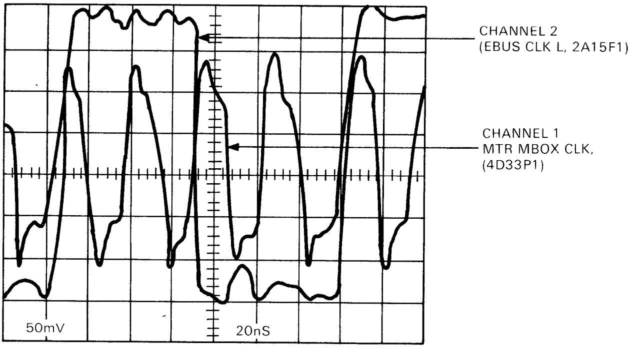
9. Display MBOX CLK H, channel 1. Identify the rising edge of MBOX CLK H that occurs before the vertical center line of the oscilloscope. Display channel 1 and channel 2.

10. Put the KL10 in the override fault state. Remove the I/O rear door to access the I/O backplane.

11. In slot 12 of the I/O backplane, find the bottom potentiometer on the clock distribution module (M8559). Using this potentiometer, adjust the Falling edge of channel 2, EBUS CLK L so that it crosses the Rising edge of MBOX CLK H. This crossing occurs on the horizontal center line of the oscilloscope.

12. Disconnect all probes.





MR-9846

HARDWARE ERROR CHART

This chart reflects all known CI20 hardware errors. It summarizes the error interface between the Operating System and the Port Hardware. You should be familiar with the entire contents of this error specification before using this chart.

Event	Monitor Sees	BUGINF	BUGINF Data	Reload Microcode	Start CI20
single nonplanned CRAM PARITY error 0-7747	CSR bit 6	KLPUCP	CSR, UCODE Adr, UCODE Data	Yes	Yes
EBUS PARITY error	CSR bit 6,24 LAR 7752	KLPEBP	CSR, UCODE ADR, PCB ERROR WORD 0 PAGE FAIL WORD	No	Yes
EBUS PARITY error	CSR bit 6,24 LAR 7753	KLPEBQ	CSR, UCODE ADR, PCB ERROR WORD 0 PCB ERROR WORD 1	No	Yes
PLI Parity error	CSR bit 6 LAR 7754	KLPPPE	" "	No	Yes
DATA PATH error	CSR bit 6,26 LAR 7756	KLPDPP	" "	No	Yes
CBUS Parity error	CSR bit 6 LAR 7755	KLPCBS	" "	No	Yes
CHANNEL ERROR	CSR bit 6 LAR 7763	KLPSCE	" "	No	Yes
SHORT WORD COUNT	CSR bit 6 LAR 7762	KLPSWC	" "	No	Yes
ADDR PAR ERROR	APR INT PCB CHAN WORDS 1+2	**STANDARD**	**STANDARD**	No	No
NXM	APR INT PCB CHAN WORDS 1+2	**STANDARD**	**STANDARD**	No	No
MEM PAR ERROR	APR INT PCB CHAN WORDS 1+2	**STANDARD**	**STANDARD**	No	No
OVERUN	PCB CHAN WORDS 1+2	**STANDARD**	**STANDARD**	No	No
CBUS AVAIL error	CSR bit 6 LAR 7757	KLPCBN	CSR, UCODE ADR PCB CHAN WORDS 1 + 2	No	Yes
EBUS REQUEST err	CSR bit 6 LAR 7760	KLPERE	CSR, UCODE ADR,	No	Yes
MBUS error	CSR bit 7	KLPMBS	CSR, UCODE ADR, UCODE DATA (TWO WORDS)	No	Yes
GRANT CSR	CSR bit 6 LAR 7761	KLPCSR	" "	No	Yes
TTO	STATUS FIELD	KLPTMO	CSR, UCODE VER, TRANSMIT STATUS REG	No	No
TRANS BUF PARITY ERROR	STATUS FIELD	KLPTPE	CSR, UCODE VER, TRANSMIT STATUS REG	No	No
INTERNAL PORT ERR	CSR bit 6 LAR 7750	KLPINP	CSR, UCODE VER, UCODE ADR	No	Yes
FAILED SELF TEST	CSR bit 6 LAR 7751	KLPFST	" "	No	Yes
SPURIOUS TRANS ATTENTION ERR	CSR bit 6 LAR 7764	KLPTAE	CSR, UCODE VER, UCODE ADR, TRANSMIT STATUS REGISTER	No	Yes
SPURIOUS REC ATTENTION ERR	CSR bit 6 LAR 7765	KLPRAE	CSR, UCODE VER, UCODE ADR, RECEIVE STATUS REGISTER	No	Yes
SPURIOUS CHAN ATTENTION ERR	CSR bit 6 LAR 7763	KLPSCE	CSR, UCODE VER, UCODE ADR, CHAN LOGO WORD 1	No	Yes

STANDARD implies that this error reporting was implemented in previous releases of TOPS-20 and has not changed.

Excluding CI20 at start up, perform a switch register load and display the KLI> prompt. Respond to the questions displayed as follows.

```
RSX-20F vb15-20 16:10 24-Aug-84

[SY0: redirected to DB0:]
[DB0: mounted]
KLI -- VERSION VB15-12 RUNNING
KLI -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
KLI>YES
KLI -- KL10 S/N:3543., MODEL B, 60 HERTZ
KLI -- KL10 HARDWARE ENVIRONMENT:
        MCA25 CACHE PAGER
        MOS MASTER OSCILLATOR
        EXTENDED ADDRESSING
        INTERNAL CHANNELS
        CABLE

KLI -- SELECT PAGE TABLE [FILE,BOTH,0,1]?
KLI>BOTH
KLI -- PAGE TABLE SELECTED: BOTH
KLI -- RELOAD MICROCODE [YES,VERIFY,FIX,NO]?
KLI>YES
KLI -- MICROCODE VERSION 352 LOADED
KLI -- RECONFIGURE CACHE [FILE,ALL,YES,NO]?
KLI>ALL
KLI -- ALL CACHES ENABLED
KLI -- CONFIGURE KL MEMORY [FILE,ALL,REVERSE,FORCE,YES,NO]?
KLI>ALL

LOGICAL MEMORY CONFIGURATION
  ADDRESS  SIZE  INT  TYPE CONTROLLER
 00000000 1024K   4   MG20  10
 04000000 1024K   4   MG20  11

KLI -- LOAD KL BOOTSTRAP [FILE,YES,NO,FILENAME]?
KLI>YES
KLI -- WRITE CONFIGRATION FILE [YES,NO]?
KLI>NO
KLI -- BOOTSTRAP LOADED AND STARTED

BOOT V10.0(201)

BOOT>
```

ENTER THE PUBLIC STRUCTURE AND START EDDT AS FOLLOWS:

```
BOOT> PS :/E
[BOOT: [LOADING] [OK]
EDDT
```

ENTER RET COMMAND AT LOCATION KLPINI+1 FOLLOWED BY STARTING LOCATION OF THE MONITOR.

```
KLPINI+1/    CONI 574,T1    RET

147$G
```

CI20 Labels

LABEL FOR
CI CARD
CAGE MUL

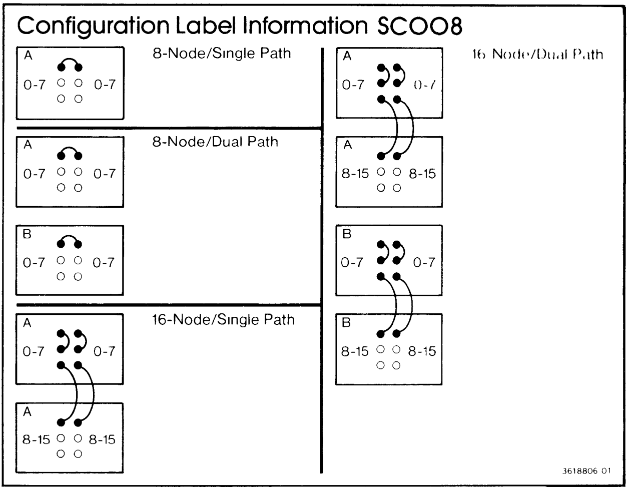
MODULE	LOCATION
L0109	LEFT
L0100	RIGHT

MR-14245

AIR FLOW FAULT LABEL IS PLACED OVER THE EXISTING CPU AIR FAULT MESSAGE DECAL ON THE 863 FAULT SWITCH.

AIR FLOW CPU/CI/NI

MR-14246



TK-9219

Star Coupler SC008 Configuration Path Label

The procedures for READ COUNTERS and SETTING THRESHOLDS will be supplied at a later date.

NETWORK INTERCONNECT ADAPTER

Table of Contents

NETWORK INTERCONNECT ADAPTER

DOCUMENTATION
Refer to the NIA20 Reference Manual (EK-NIA20-RM) for detailed functional/logic/installation descriptions. The Print Sets can be ordered as follows.

Order Number	Print Set
MP-01984	NIA20-A (KL10-E)
MP-01907	NIA20-B (KL10-D)
MP-01908	NIA20-C (KL10-R)

NIA20 CARD CAGE

REAR PANEL CONNECTORS

CARD CAGE

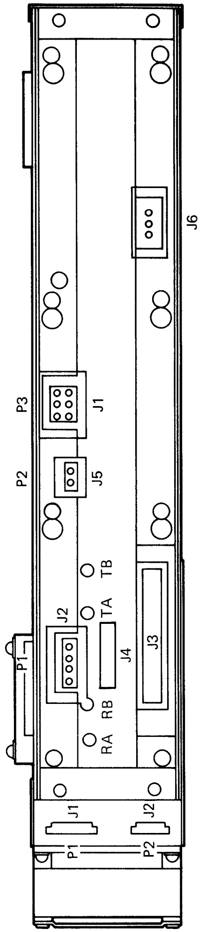
J1 DC POWER +5 0 VOLTS, GND
J2 DC POWER −5 2 VOLTS, GND
J3 PLI
J4 NIA20 INTERNAL TRANSCEIVER CABLE
J5 VOLTAGE MONITOR FOR +5.0 VOLTS
J6 VANE SWITCH
TB, TA, RB, RA: FOR CI20 USE ONLY

FAN SUBASSEMBLY

J1 VAN SWITCH
J2 FAN AC

REAR VIEW

TOP

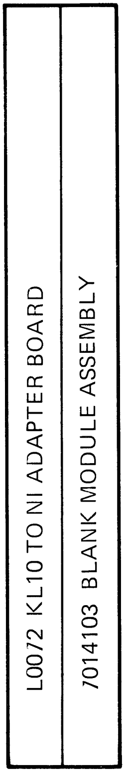


BOTTOM

MR-14241

INTERNAL SLOT ASSIGNMENTS

TOP



MUL DECAL
(LOCATED ON SIDE DOOR)

MODULE	LOCATION
7014103	RIGHT
L0072	LEFT

NIA20 CARD CAGE
(OPENED FRONT DOOR VIEW)

MR-14263

MODULE LOCATIONS
RH/DTE/CI/NIA

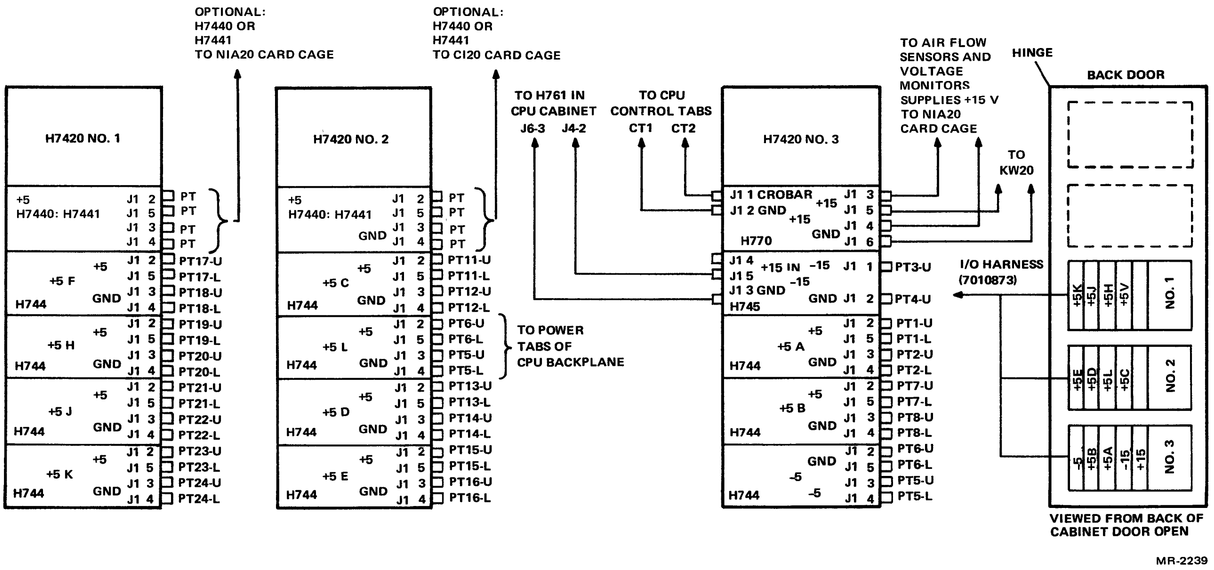
The following NIA modules are located as indicated.

Slot	Module
19	M3001 EBUS INTERFACE/PORT ALU
20	M3002 PORT MICROPROCESSOR CONTROL
21	M3003 CBUS/PLI INTERFACE
22}	
23}	BLANK MODULE ASSEMBLY
24}	

Refer to Volume I DIAGRAMS MULS for all other slot assignments.

CPU

Slot	Module
31	M8532-YA PI BOARD PIC


OPTIONAL:
H7440 OR
H7441
TO NIA20 CARD CAGE
OPTIONAL:
H7440 OR
H7441
TO CI20 CARD CAGE
TO H761 IN
CPU CABINET
J6-3 J4-2
TO CPU
CONTROL TABS
CT1 CT2
TO AIR FLOW
SENSORS AND
VOLTAGE
MONITORS
SUPPLIES +15 V
TO NIA20
CARD CAGE
HINGE
BACK DOOR
TO
KW20
H7420 NO. 1
+5
H7440: H7441
J1 2 PT
J1 5 PT
J1 3 PT
J1 4 PT
+5 F
+5
H744
GND
J1 2 PT17-U
J1 5 PT17-L
J1 3 PT18-U
J1 4 PT18-L
+5 H
+5
H744
GND
J1 2 PT19-U
J1 5 PT19-L
J1 3 PT20-U
J1 4 PT20-L
+5 J
+5
H744
GND
J1 2 PT21-U
J1 5 PT21-L
J1 3 PT22-U
J1 4 PT22-L
+5 K
+5
H744
GND
J1 2 PT23-U
J1 5 PT23-L
J1 3 PT24-U
J1 4 PT24-L
H7420 NO. 2
+5
H7440: H7441
GND
J1 2 PT
J1 5 PT
J1 3 PT
J1 4 PT
+5 C
+5
H744
GND
J1 2 PT11-U
J1 5 PT11-L
J1 3 PT12-U
J1 4 PT12-L
+5 L
+5
H744
GND
J1 2 PT6-U
J1 5 PT6-L
J1 3 PT5-U
J1 4 PT5-L
TO POWER
TABS OF
CPU BACKPLANE
+5 D
+5
H744
GND
J1 2 PT13-U
J1 5 PT13-L
J1 3 PT14-U
J1 4 PT14-L
+5 E
+5
H744
GND
J1 2 PT15-U
J1 5 PT15-L
J1 3 PT16-U
J1 4 PT16-L
H7420 NO. 3
J1 1 CROBAR
J1 2 GND
+15
+15
GND
H770
J1 3
J1 5
J1 4
J1 6
J1 4
J1 5
J1 3 GND
H745
+15 IN
-15
-15
GND
J1 1 PT3-U
J1 2 PT4-U
I/O HARNESS
(7010873)
+5 A
+5
H744
GND
J1 2 PT1-U
J1 5 PT1-L
J1 3 PT2-U
J1 4 PT2-L
+5 B
+5
H744
GND
J1 2 PT7-U
J1 5 PT7-L
J1 3 PT8-U
J1 4 PT8-L
GND
-5
-5
H744
J1 2 PT6-U
J1 5 PT6-L
J1 3 PT5-U
J1 4 PT5-L
+5K +5J +5H +5V
NO. 1
+5E +5D +5L +5C
NO. 2
-5 +5B +5A -15 +15
NO. 3
VIEWED FROM BACK OF
CABINET DOOR OPEN
MR-2239

GENERAL POWER SUPPLY SPECIFICATIONS

The following voltage measurements are to be made from the backplane of the option.

POWER SUPPLY TYPE	OUTPUT	TOLERANCE MAXIMUM	TOLERANCE MINIMUM	MAXIMUM RIPPLE IN MILLIVOLTS
744/7440	+5	+5.05	+4.95	150

SWITCH SETTINGS
DC Voltage Monitor Board (DEC P/N 5414506-01)

Switch 1 should be set to ON. All other switches should be OFF.

CONTROL AND STATUS REGISTER (CSR)

BIT NO	BIT DEFINITION	RD/WR KL10	RD/WR PORT
00	PORT PRESENT	R	H
01	DIAG RQST CSR	R	H
02	DIAG CSR CHNG	R/H	H
03		*	*
04	RQST EXAM OR DEP	R/H	R/S
05	RQST INTERRUPT	R/H	R/S
06	CARD PARITY ERR	R/C	H
07	MBUS ERROR	R	H
08		*	*
09		*	*
10		*	*
11	IDLE	R	R/W
12	DISABLE COMPLETE	R	R/W
13	ENABLE COMPLETE	R	R/W
14		*	*
15	PORT ID CODE 00	R	H
16	PORT ID CODE 01	R	H
17	PORT ID CODE 02	R	H

BIT NO	BIT DEFINITION	RD/WR KL10	RD/WR PORT
18	CLEAR PORT	W	*
19	DIAG TEST EBUF	R/W	*
20	DIAG GEN EBUS PE	R/W	*
21	DIAG SEL LAR	R/W	*
22	DIAG SINGLE CYC	R/W	*
23	SPARE	R/W	*
24	EBUS PARITY ERR	H R/C	R/H
25	FREE QUEUE ERR	R/C	R/S
26	DATA PATH ERR	R/C	R/S
27	CMD QUEUE AVAIL	R/S	R/C
28	RSP QUEUE AVAIL	R/C	R/S
29		*	*
30	DISABLE	R/S	R/C
31	ENABLE	R/S	R/C
32	MPROC RUN	R/W	R/H
33	PIA 00	R/W	R
34	PIA 01	R/W	R
35	PIA 02	R/W	R

* = NOT DEFINED
R = READABLE
W = WRITABLE (SET OR CLEAR)
C = CLEARABLE ONLY
S = SETTABLE ONLY
H = HARDWARE CONTROLLED

MR 14248

CSR BIT DEFINITIONS

BIT	NAME	DEFINITION
00	PORT PRESENT	Indicates to the KL10 that the port is present (installed and powered-up).
01	DIAG RQST CSR	When set, this diagnostic bit indicates that the port requested access to the CSR.
02	DIAG CSR CHNG	This diagnostic bit indicates that the contents of the CSR changed since it was last read by the port microprocessor.
03	UNUSED	Not used by either the port microprocessor or the KL10.

CSR BIT DEFINITIONS (Cont)

BIT	NAME	DEFINITION
04	RQST EXAM OR DEP	Used by the port microprocessor to request an EBus interrupt on PI level 00 (Examine or Deposit function). The setting of this bit immediately generates the interrupt request.
05	RQST INTERRUPT	Used by the port microprocessor to request an EBus interrupt on PI levels 01 through 07. The setting of this bit immediately generates the interrupt request.
06	CRAM PAR ERR	Indicates that a control RAM (CRAM) parity error was detected. If this bit is set, the port microprocessor is immediately halted and RQST INTERRUPT (CSR bit 05) is set. A hardware nonvectored (40 + 2n) interrupt is forced. A CRAM PAR ERR may be forced to halt the port microprocessor at a specific location (break point). The port microprocessor cannot be restarted (CSR bit 32 set) until this bit is cleared.
07	MBUS ERR	Indicates that more than one MBus driver has turned on at the same time. That is, more than one set of port logic is trying to drive the MBus at the same time. If this bit is set, the port microprocessor is immediately halted and RQST INTERRUPT (CSR bit 05) is set. A hardware nonvectored (40 + 2n) interrupt is forced. The port microprocessor cannot be restarted (CSR bit 32 set) until this bit is cleared.
08	UNUSED	Neither the port microprocessor or the KL10 uses this bit.
09	UNUSED	Neither the port microprocessor or the KL10 uses this bit.
10	UNUSED	Neither the port microprocessor or the KL10 uses this bit.
11	IDLE LOOP	Indicates the port microprocessor is in the Idle Loop, and is not "hung" in some other microcode routine.
12	DISABLE COMPLETE	Informs the KL10 that the port microprocessor placed itself in the DISABLED state.
13	ENABLE COMPLETE	Informs the KL10 that the port microprocessor placed itself in the ENABLED state.
14	UNUSED	Neither the port microprocessor or the KL10 uses this bit.
15 16 17	PORT ID CODE 00 PORT ID CODE 01 PORT ID CODE 02 Hardwired so that:	Three-bit PORT IDENT CODE field. Informs software that this is a CI20 port and not an RH20 controller. 00 = 0 01 = 1 02 = 1

CSR BIT DEFINITIONS (Cont)

BIT	NAME	DEFINITION
18	CLEAR PORT	When set by the KL10, this bit resets the port. The microprocessor is halted and all pertinent registers and control logic are placed in a reset state. The bit clears itself after the reset function is completed.
19	DIAG TEST EBUF	This diagnostic bit enables the KL10 to do an EBus interface loopback function by loading and reading the EBus buffer (EBUF). If the port is not running (CSR bit 32 is reset) and this bit is set, then a KL10: DATAO loads EBus data into the EBUF. DATAI places EBUF data on the EBus.
20	DIAG GEN EBUS PE	This diagnostic bit enables the KL10 to test the EBus parity checker by forcing it to decode an EBus parity error. When this bit is set, EBUS PAR ERR (CSR bit 24) is also set on the same CONO, assuming there was no real EBus parity error.
21	DIAG SEL LAR	This diagnostic bit enables a KL10 DATAI to read the CRAM address, in the Latch Address Register (LAR). If this bit is set and bits 19 and 32 are reset, then the DATAI causes the LAR contents to be asserted on EBus D01-D12.
22	DIAG SINGLE CYC	This diagnostic bit enables the port microprocessor to be single cycled. If this bit is set and the KL10 sets MPROC RUN (CSR bit 32), the port microprocessor runs one microcycle and halts. MPROC RUN is cleared when the microprocessor halts. The current address to be executed is fetched from the RAM Address Register (RAR). The next address to be executed is stored in the LAR at the completion of the microcycle. The KL10 must read the address from the Latch Address Register (LAR) and load it into the RAR before executing the next single cycle.
23	SPARE	Reserved for future software use.
24	EBUS PARITY ERR	When read by the KL10, this bit indicated that an EBus parity error was detected. When written as a "1" by the KL10, this bit clears itself and CRAM PARITY ERR (CSR bit 06).
25	FREE QUEUE ERR	Used by the port to inform the Port Driver that there are no free queue entries available on either the Datagram Free Queue or the Message Free Queue.
26	DATA PATH ERR	Informs the Port Driver that the port microprocessor detected an error in the DMA data path.
27	CMD QUEUE AVAIL	Used by the Port Driver to inform the port that it placed a command queue entry on a previously empty command queue.

CSR BIT DEFINITIONS (Cont)

BIT	NAME	DEFINITION
28	RESP QUEUE AVAIL	Used by the port to inform the Port Driver that it placed an entry on the previously empty Response Queue.
29	UNUSED	Neither the port microprocessor or the KL10 uses this bit.
30	DISABLE	Used by the Port Driver to tell the port to place itself in the DISABLED state (set CSR bit 12).
31	ENABLE	Used by the Port Driver to tell the port to place itself in the ENABLED state (set CSR bit 13).
32	MPROC RUN	When set by the KL10, this bit causes the CRAM Control Register to reset and enables the port microprocessor clocks. The port starts cycling at the address in the RAM Address Register (RAR). The next and subsequent addresses are fetched from the Am2910 sequencer.
33 34 35	PIA00 PIA01 PIA02	Three-bit KL10 EBus Physical Interrupt Assignment (PIA) field (PI level 01 through 07).

PORT CONTROL BLOCK CONTENTS (See Figure)

WORD	DESCRIPTION
0-2	Command Queue Interlock Word, Flink, and Blink.
3	Reserved
4-6	Response Queue Interlock Word, Flink, and Blink.
7	Reserved
10-13	Unknown Protocol Type Free Queue Interlock Word, Flink, Blink, and Queue Entry Length.
14	Reserved
15	Protocol Type Table Starting Address is used to find the beginning of the Protocol Type Table. This table has free queue pointers for each enabled protocol type.
16	Multi-cast Address Table Starting Address is used to find the beginning of the Multi-cast Address Table. This table has a list of Multi-cast Addresses to which the port is to respond.
17	Reserved
20-21	Error Logout Words 0,1 are written by the port when it encounters fatal errors associated with Queue Manipulation. This error reporting stategy requires the port to write as much information as possible directly into the host memory. This approach requires the smallest subset of port hardware and microcode to be working to report these errors.
22	Contains the address of the EPT Channel Logout Word 1, which is written by the port driver at initialization. PCB words 22 and 23 are used by the port during Channel Error Recovery.
23	Holds the contents of EPT Channel Logout Word 1, which is written by the port on any kind of channel error detected during or immediately after a DMA transfer.
24	PCB Base Address contains the physical memory address of location 0 of the PCB. The NIA20 has no other way of finding the PCB.

KL10 MEMORY PORT CONTROL BLOCK (PCB)

PCB FORMAT	OCT
COMMAND QUEUE INTERLOCK	0
COMMAND QUEUE FLINK	1
COMMAND QUEUE BLINK	2
RESERVED FOR SOFTWARE	3
RESPONSE QUEUE INTERLOCK	4
RESPONSE QUEUE FLINK	5
RESPONSE QUEUE BLINK	6
RESERVED FOR SOFTWARE	7
UNKNOWN PROTOCOL TYPE FREE QUEUE INTERLOCK	10
UNKNOWN PROTOCOL TYPE FREE QUEUE FLINK	11
UNKNOWN PROTOCOL TYPE FREE QUEUE BLINK	12
UNKNOWN PROTOCOL QUEUE ENTRY LENGTH	13
RESERVED FOR SOFTWARE	14
PROTOCOL TYPE TABLE STARTING ADDRESS	15
MULTI-CAST ADDRESS TABLE STARTING ADDRESS	16
RESERVED FOR SOFTWARE	17
ERROR LOGOUT 0	20
ERROR LOGOUT 1	21
EPT CHANNEL LOGOUT WORD 1 ADDRESS	22
EPT CHANNEL LOGOUT WORD 1 CONTENTS	23
PCB BASE ADDRESS	24
PIA ASSIGNMENT	25
RESERVED TO PORT	26
CHANNEL COMMAND WORD	27
READ COUNTERS DATA BUFFER STARTING ADDRESS	30

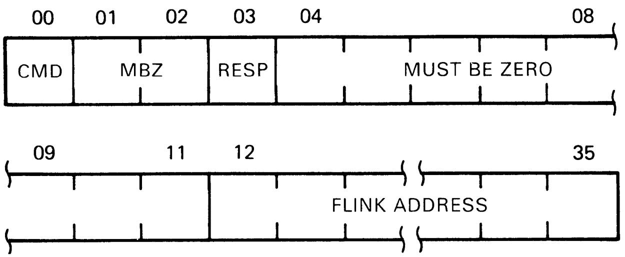
MR-14261

WORD	DESCRIPTION
25	PIA Assignment contains the Priority Interrupt Level assigned to the NIA20.
26	Reserved
27	Channel Command Word contains a CCW-Style word written by the port to transfer data over the KL10 CBUS. The port driver is responsible for writing a Channel Jump Word into the appropriate EPT location corresponding to the RH20 backplane slot that the NIA20 occupies.
30	Read Counters Data Buffer Starting Address points to the beginning of the Read Counters Data Buffer.

The Error Words (Words 20, 21) are written by the port when it encounters fatal errors associated with Queue manipulation. This error reporting strategy requires the port to write as much information as possible directly into the host memory. This approach requires the smallest subset of port hardware and microcode to be working to report these errors.

The information in these words provides enough data for the port driver to determine the type of error and where the error occurred. When the error is detected, the port writes the contents of the Error Words in the PCB, enters the Disabled State, and generates a host interrupt.

The format of Error Word 0 is:

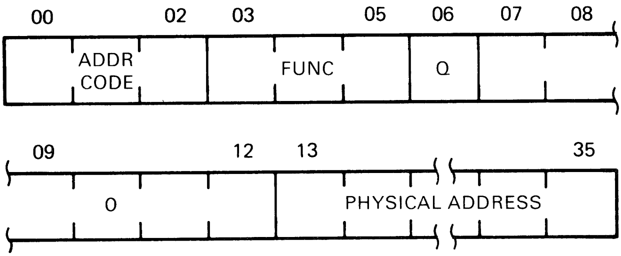


MR-14256

BITS	NAME	DESCRIPTION
0	CMD	Error occurred while reading a command queue entry.
1-2	MBZ	These bits must be zero.
3	RESPONSE	This bit is on if the error occurred while the port was trying to build a response queue entry.
4-11	MBZ	These bits must be zero.
12-35	FLINK ADR	This is the address of the FLINK word of the queue in question.

Error Word 1 contains the API function word that the port processor used to access memory when the memory error occurred. This word is written here in the same format as it should have appeared on the EBUS.

The format of this word is:

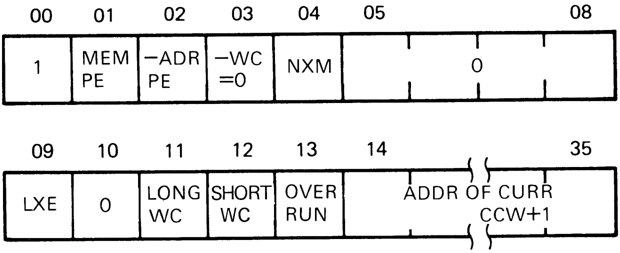


MR-14257

Word 22 of the PCB is written by the port during initialization time with the address of the EPT Channel Logout Word 1 which the port gets from the port driver software. Words 22 and 23 of the PCB are used by the port during Channel Error recovery.

Word 23 contains the Channel Logout Word 1 written by port on any kind of channel error detected during or immediately after, a DMA transfer.

The format of Error Word 3 is:



MR-14255

BITS	NAME	DESCRIPTION
01	MEM PE	Memory Parity Error.
02	-ADR PE	Not Address Parity Error.
03	-WC=0	Chan Word Count did not = 0 when chan did a store to EPT.
04	NXM	Chan ref non exist mem.
09	LXE	Error detected after port term transfer. Chan aborts next transfer.
11	LONG WC	Port comp Xfer, but word count in CCW not reached.
12	SHORT WC	Chan Xferred data spec by CCW, but port still has data.
13	OVER RUN	If device read, Port sent data but chan buff was full. If device write, Port req data but chan buff was empty.

Word 24 of the PCB is the address of the first word of the PCB; the NI20 has no other way of finding the PCB.

Word 27 is reserved for the Channel Command Word. The port writes a CCW-style word here when it wishes to transfer data over the KL10 CBus. The port driver is responsible for writing a Channel Jump Word into the appropriate EPT location corresponding to the RH20 backplane slot that the NI20 is installed in.

Word 25 is always reserved for port microcode use; the port driver should never write this location nor depend upon its value.

Word 30 of the PCB is a pointer to the beginning of the Read Counter Data Buffer. This address is supplied by the port driver software at initialization.

When the NIA20 is initialized, the port driver must set up the channel to transfer the contents of the PCB into the port. This is done by setting up a CCW to transfer three words starting with word 24 of the PCB from KL10 memory to the channel. The port starts the channel and reads the contents of these locations. This provides the port with the base of the PCB, and its PI assignment.

It is important to realize that since the port uses the channel to transfer large blocks of data, the channel writes logout information into the EPT. An error that the channel discovers is reported in the usual manner through the EPT.

Status Field From Queue

The status field in the following format is used by the port to report the status of all completed commands. This field appears in the response word of the queue entry. When valid, this field indicates the logging of an exception event.

When the CRAM PE field is set to 1, the forthcoming read counter response is due to execution of a planned CRAM parity error.

When the send/receive bit is 0, an error occurred on receive; receive failed. When this bit is 1, an error occurred on transmit; transmission failed.

0	1	2	3	4	5	6	7
CRAM PE	SEND/ RECEIVE	ERROR TYPE					ERROR

MR-13794

When the error bit is 0, the status field has no meaning and must be zero (MBZ). When this bit is 1, the status field is reporting an error event. The definition of the error type fields, and of the direction field, comes into effect.

The error type field indicates the error event being logged. The field is set according to the following table.

Error Log and Type

Bit Value (Octal)	Event Type
00	Excessive collisions
01	Carrier check failed (carrier lost)
02	Collision detect check failed
03	Short circuit
04	Open circuit
05	Frame too long
06	Remote failure to defer (late collision)
07	Block check error (CRC error)
10	Framing error
11	Data overrun (NIA buffer space exhausted)
12	Unrecognized protocol type
13	Frame too short
30	Channel error WC not equal zero
31	Queue length violation
32	Illegal PLI function
33	Unrecognized command
34	Buffer length violation
35	Reserved
36	Transmit buffer parity error
37	Internal error

Read and Clear Performance Counters RCCNT Command.
This command reads the performance counters, returning their value in the read counters block pointed to by PCB +30, and clears the performance event counters as specified by a bit (14) in the flags field.

The format of the command queue entry needed to do this is as follows.

QUEUE FLINK					
QUEUE BLINK					
RESERVED FOR SOFTWARE					
<0-7> STATUS	<8-13> FLAGS	<14> CLRCTR	<15>	<16-23> OPCODE	MBZ

MR 13671

The operation code for this command packet is 4. A response is built if the response bit in the flags word is set.

Counters Read or Cleared (CNTRC) Response.
If the response bit of the flags field of the original command packet is on, then a response to the above command is built. The response has the following format.

<table>
<tr><td colspan="4">QUEUE FLINK</td></tr>
<tr><td colspan="4">QUEUE BLINK</td></tr>
<tr><td colspan="4">RESERVED FOR SOFTWARE</td></tr>
<tr><td><0-7>
STATUS</td><td><8-15>
FLAGS</td><td><16-23>
OPCODE</td><td>MBZ</td></tr>
</table>

(PCB+30) →

<table>
<tr><td colspan="2">BR</td></tr>
<tr><td colspan="2">BX</td></tr>
<tr><td colspan="2">FR</td></tr>
<tr><td colspan="2">FX</td></tr>
<tr><td colspan="2">MCBR</td></tr>
<tr><td colspan="2">MCFR</td></tr>
<tr><td colspan="2">FXID</td></tr>
<tr><td colspan="2">FXSC</td></tr>
<tr><td colspan="2">FXMC</td></tr>
<tr><td colspan="2">XF</td></tr>
<tr><td colspan="2">XFBM</td></tr>
<tr><td colspan="2">CDCF</td></tr>
<tr><td colspan="2">RF</td></tr>
<tr><td colspan="2">RFBM</td></tr>
<tr><td colspan="2">DDUPT</td></tr>
<tr><td colspan="2">DDPT1</td></tr>
<tr><td colspan="2">DDPT2</td></tr>
<tr><td colspan="2">DDPT3</td></tr>
<tr><td colspan="2">DDPT4</td></tr>
<tr><td colspan="2">DDPT5</td></tr>
<tr><td colspan="2">DDPT6</td></tr>
<tr><td colspan="2">DDPT7</td></tr>
<tr><td colspan="2">DDPT8</td></tr>
<tr><td colspan="2">DDPT9</td></tr>
<tr><td colspan="2">DDPT10</td></tr>
<tr><td colspan="2">DDPT11</td></tr>
<tr><td colspan="2">DDPT12</td></tr>
<tr><td colspan="2">DDPT13</td></tr>
<tr><td colspan="2">DDPT14</td></tr>
<tr><td colspan="2">DDPT15</td></tr>
<tr><td colspan="2">DDPT16</td></tr>
<tr><td colspan="2">URFD</td></tr>
<tr><td colspan="2">DOVR</td></tr>
<tr><td colspan="2">SBUA</td></tr>
<tr><td colspan="2">UBUA</td></tr>
<tr><td>PLI REG RD PAR ERROR</td><td>PLU PARITY ERROR</td></tr>
<tr><td>MOVER PARITY ERROR</td><td>CBUS PARITY ERROR</td></tr>
<tr><td>EBUS PARITY ERROR</td><td>EBUS QUEUE PARITY ERROR</td></tr>
<tr><td>CHANNEL ERROR</td><td>SPUR CHANNEL ERROR</td></tr>
<tr><td>SPUR XMIT ATTN ERROR</td><td>CBUS REQ TIMEOUT ERROR</td></tr>
<tr><td>EBUS REQ TIMEOUT ERR</td><td>CSR GRNT TIMEOUT ERROR</td></tr>
<tr><td>USED BUFF PARITY ERR</td><td>XMIT BUF PARITY ERROR</td></tr>
<tr><td>RSVD FOR UCODE</td><td>RSVD FOR UCODE</td></tr>
<tr><td>RSVD FOR UCODE</td><td>RSVD FOR UCODE</td></tr>
<tr><td>RSVD FOR UCODE</td><td>RSVD FOR UCODE</td></tr>
</table>

0 17 18 35

MR 1677

The following describes the bits for the Counters Read or Cleared (CNTRC) Response.

BR -- **Bytes received.** The number of 8-bit data text characters received as datagrams over the NI. This includes maintenance operation protocol (MOP) packets.

BX -- **Bytes transmitted.** The number of 8 bit data text characters transmitted successfully as datagrams over the NI.

FR -- **Frames received.** The number of frames (packets or datagrams) received over the NI wire.

FX -- **Frames transmitted.** The number of frames successfully transmitted over the NI wire.

MCBR -- **Multicast bytes received.** The number of 8-bit bytes received in packets with the multicast bit set in the destination field. This includes broadcast.

MCFR -- **Multicast frames received.** The number of frames received with the multicast bit in the destination field set. This includes broadcast.

FXID -- **Frames transmitted, initially deferred.** The number of frames transmitted that deferred to other traffic on the NI wire before transmission.

FXSC -- **Frames transmitted, single collision.** The number of frames successfully transmitted, and which collided with another transmission exactly once.

FXMC -- **Frames transmitted, multiple collisions.** The number of frames successfully transmitted, and which collided with another transmission more than once.

XF -- **Transmit failures.** The number of frames that were not successfully transmitted. This counter is incremented for excessive collisions, parity errors, and so on. This counter is associated with the XFBM, which notes occurrence of error classes.

XFBM -- **Transmit failure bit mask.** Gives the accumulated reasons for transmission failures. The bit meanings are given in the following table.

Transmission Failure Bit Mask Assignments

Bit Number	Reason for Failure
0-23	Unassigned
24	Loss of carrier
25	Transmit buffer parity error
26	Remote failure to defer
27	Frame too long
28	Open circuit
29	Short circuit
30	Carrier check failed (collision detect check failed)
31	Excessive collisions

CDCF -- **Collision detect check failed.** The number of times the collision detect check failed after a transmit. The number of times that heartbeat failed to assert after a transmit ended. This counter has meaning only if the H4000 mode bit is set.

RF -- **Receive failures.** The number of received frames whose reception ultimately failed. This counter is associated with the RFBM counter, which marks occurrence of the various error types.

RFBM -- **Receive failure bit mask.** Gives the accumulated reasons for receive failures. The bit definitions are given in the following table.

Reception Failure Bit Mask Assignments

Bit Number	Reason for Failure
0-26	Unassigned
27	Free list parity error
28	Data overrun (no free buffers)
29	Frame too long
30	Framing error
31	Block check error

DDUPT -- Datagram discarded for unknown protocol type. Keeps track of the number of datagrams discarded for the unknown protocol type free queue. Any time a datagram is discarded with an unrecognized protocol type, this counter is incremented.

DDPT1 to DDPT16 -- Datagram discarded for protocol type N. Keep track of the number of datagrams discarded for each of the protocol type free queues. When a datagram is discarded because of no vailable free space, one of these counters is incremented, if the protocol type was enabled. There are as many of these counters as needed to support the number of protocol types allowed in the NI configuration registers.

URFD -- Unrecognized frame destination. Has no meaning for the NIA20 and is always reported as zero.

DOVR -- Data overrun. Represents the number of packets incorrectly received because buffer space in the NIA is exhausted. Such packets are discarded.

SBUA -- System buffer unavailable. Has no meaning for the NIA and is always reported as zero.

UBUA -- User buffer unavailable. Represents the total number of packets discarded because a free queue was exhausted. This number is the total of the datagram discarded for protocol type N counters; and the datagram discarded for unknown protocol type counter.

Words 47-55 are the NIA20 port recoverable errors. Have an initial threshold of 5, set during port initialization. This threshold count is variable and can be set through a write station information command.

RSVD -- Reserved for microcode. Reserved for the port microcode and for the present are returned as zeros.

DIAGNOSTICS

The following abstracts describe the NIA20 diagnostic programs:

- DFPTA Port Basic Device Diagnostic
- DFNIE NIA Module (L0072) Diagnostic
- DFNIA NIA20 Functional Diagnostic
- DFNIB Network Interconnect Exerciser
- DFSXA KL10 Channel/DTE20-Interconnection Test

DFPTA - PROGRAM ABSTRACT

DFPTA is the basic device diagnostic for the CI20/NIA20 controller on the KL10. It detects and isolates hard faults stuck in the device. Manufacturing uses DFPTA to repair CI20/NI20 modules. Field Service uses DFPTA to verify port operation or isolate a fault to a replaceable module.

DFPTA tests one or two CI20/NIA20 controllers. Each controller consists of three port modules in RH20 slot 5 or 7.

DFPTA isolates faults to a network of failing chips. Each network is a set of chips functionally related. Typically, several networks are printed, with the first network being the most probable.

DFPTA consists of two major program sections.

- Test section - Contains all hardware tests. This section is used to debug a module or verify a port.
- Debug section - Allows detailed manipulation or inspection of the port.

DFNIE - PROGRAM ABSTRACT
DFNIE is the hardware diagnostic for the Network Interface Adapter (NIA) on the KL10. It detects and isolates hard faults or faults stuck in the device. Manufacturing uses DFPTA to repair NIA modules. Field Service uses DFPTA to verify NIA operation and isolate a fault to the replaceable module (NIA).

DFNIE tests one NIA controller. Each controller consists of an NIA20 port in RH20 slot 5.

DFNIE isolates faults to a network of failing chips. Each network is a set of chips functionally related. Typically, several networks are printed, with the first network being the most probable.

DFNIE consists of two major program sections.

1. Test section - Contains all hardware tests. This section is used to debug a module or verify proper NIA hardware operation.
2. Debug section - Allows detailed manipulation or inspection of the PLI.

DFNIA - PROGRAM ABSTRACT
DFNIA is a functional diagnostic used to verify the functionality of an NIA20 port consisting of three Port Modules, an NIA module, H4000 Transceiver, and an NI cable. It isolates faults to either of three Port Modules, and NIA module, H4000 Transceiver, and cables.

DFNIA tests the NI port consisting of three port modules in RH20 slot 5 and an NIA module in a separate card cage.

DFNIA consists of two major program sections.

1. Test section - Contains all hardware tests. This section functionally verifies the port or isolates an actual problem.
2. Debug section - Allows detailed port manipulation or inspection.

DFNIB - PROGRAM ABSTRACT
The Network Interconnect Exerciser exercises from a KL10 system the ability of all nodes on an NI network to communicate with each other. Testing is done with the Low Level Maintenance Operations (LLMOP) of the NI. DFNIB does not interfere with normal NI network traffic and runs concurrently with normal NI-DECNET network traffic. DFNIB is a USER mode-only diagnostic and requires TOPS-20 Version 6.0 or later with NI-DECNET support. DFNIB is a self-contained program and does not require any diagnostic support programs.

USERS AND USES
DFNIB is part of the standard KL10 Diagnostic package and is available to those users who are eligible to receive KL10 diagnostics and related updates.

The program is designed for engineers and technicians who are qualified to test and maintain NI networks. To run the diagnostic, the user must have maintenance, wheel, or operator privileges.

Some typical DFNIB uses are:

- Installation testing of a new node.
- Installation acceptance of a new node.
- Isolation of faulty nodes.
- Verification of a repaired node.
- Exercising an NI network.

DFSXA - PROGRAM ABSTRACT
MD-10-DFSXA is the KL10 Channel/DTE20 interaction test for the KL10 computer system. It is a software tool that allows the engineer to interactively test all data channels into and out of the KL10's internal memory. The program tests up to eight KL10 I/O Channels (RH20 MASSBUS Controllers or NIA20 or CI20 Ports) along with up to four front-end PDP-11 Channels (DTE20 KL10 to PDP-11 interfaces).

The RH20 Channels can be tested in one of two operating modes.

1. In internal loopback mode (deviceless), or

2. By writing or reading data to or from either an RP04 disk pack or a TU16 magtape drive.

The NIA20 or CI20 Ports are tested in much the same way as an RH20 Channel in internal loopback mode. Data transfers occur from KL10 memory over the CBUS back to memory over the EBUS interface (a "write") or in the opposite direction (a "read").

The DTE20 Channels can be tested in one of two possible operating modes:

1. With only the minimum PDP-11 resident software required to transfer data between "11" and "10" core, or

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 allow the engineer to evaluate system performance and help detect and isolate interactive system problems to the faulty subsystem. This is the only program in the set of DECSYSTEM-20 diagnostics that provides simultaneous testing of both the PDP-11 front-ends and the KL10 I/O channels.

PREREQUISITE SOFTWARE
DFNIB requires TOPS-20 with NI-DECNET running. TOPS-20 assumes that no solid faults exist in the KL10 cpu/memory/node hardware. Run the following diagnostics before running TOPS-20.

- CPU and MEMORY diagnostics (all)
- NI NODE diagnostics (all)

DESKEWING/ADJUSTMENT PROCEDURE

1. Deskew the port modules using a Tektronix 475 (or equivalent 100 MHZ minimum) oscilloscope by performing the following steps (see Figure).

2. Connect channel 1 of the oscilloscope to MTR MBOX CLK H, 4D33P1, on the CPU backplane. Use a ground clip.

3. Set the time base to 20 ns.

4. Set channel 1 vertical gain to 0.5 V/division. Set the ground reference to 1.3 volts above the horizontal center level of the oscilloscope. (MTR MBOX CLK H is an ECL signal.)

5. Set the oscilloscope sync to positive external.

6. Connect external sync input to CHTO H, 4B09K1 on the CPU backplane. Use a ground clip.

7. Connect channel 2 to CDS1, EBUS CLK L, 2A21F1 on the I/O backplane. Set the channel 2 vertical gain to 0.5 V/division. Use a ground clip. To measure TTL voltages, set the ground reference to 1.5 volts below the horizontal center line of the oscilloscope.

8. Press the Trigger View Switch of the oscilloscope and display the external sync. Adjust the display, so that the rising edge of the external sync aligns with the vertical center line of the oscilloscope.

9. Display MBOX CLK H, channel 1. Identify the rising edge of MBOX CLK H that occurs before the vertical center line of the oscilloscope. Display channel 1 and channel 2.

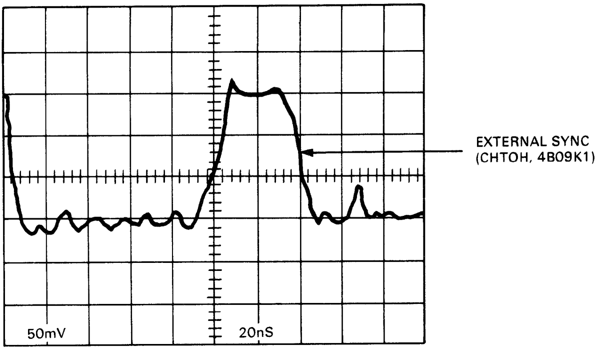
10. Put the KL10-E in the override fault state. Remove the I/O rear door to access the I/O backplane.

11. Find the potentiometer (DL6) that is third from the bottom on the clock module (M8559) in slot 12 of the I/O backplane. Using this potentiometer, adjust the Falling edge of channel 2, EBUS CLK L so that it crosses the Rising edge of MBOX CLK H. This crossing occurs on the horizontal center line of the oscilloscope.

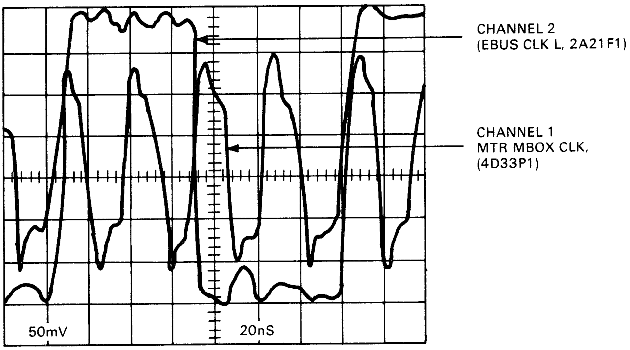
12. Disconnect all probes.

13. Mount the KLAD back on the front end RP06.

14. Load and run diagnostic DFPTA to verify proper functioning of the port modules. If the modules fail, troubleshoot as the diagnostic directs. If the modules are functioning properly, continue with the installation.



EXTERNAL SYNC (CHTO H)



EBUS CKL L AND MTR MBOX CLK

MR 13732

NIA20 Deskewing Timing

NIA20 LABELS

CARD CAGE MUL LABEL

MODULE	LOCATION
7014103	RIGHT
L0072	LEFT

MR-14243

AIR FLOW FAULT LABEL

(PLACED OVER THE EXISTING CPU AIR FAULT MESSAGE DECAL ON THE 863 FAULT SWITCH)

AIR FLOW CPU/CI/NIA

MR-14244

Table of Contents

(Refer to tab Computer Interconnect section for information.)

RA81 ERROR CODES

SYSTEM FAULT CODES
The RA81 drive errors and fault display codes are listed below.

NOTE
Loopback plugs left in the drive after running Diagnostic Test C cause multiple false error codes to display.

Drive Error Display Codes

Error Code	Name
01	Spindle motor speed transducer timeout.
02	Spindle speed of 100 r/min not reached in six seconds during spin-up.
03	Spindle not accelerating during spinup.
04	Spindle speed of 3420 r/min not reached in 40 seconds during spin-up.
05	Start-up inhibited by power sequencing grant not present or start in progress (SIP) present (this could be legitimate status).
06	Microcode fault.
07	Level two message frame sequencing error.
08	Level two message checksum error.
09	SDI message framing error.
0A	Invalid operation code parity for level 1 or 2 message.
0B	Invalid operation code for level 1 or 2 message.
0C	Invalid operation length for level 2 message.
0D	Status error byte is nonzero while trying to execute a command.
0E	Group select code is nonzero while trying to execute a command.
0F	Write protect switch is in the PROTECT position while trying to write enable the drive.
10	Invalid error code sent by slave.
13	Fine track not reached during detent mode.
15	Too much time taken to execute a seek or recal command.
16	Guard band detected while performing a seek command.
17	Seek logic difference counter decremented past 0 before expected cylinder was reached.
1A	Seek command contains an invalid cylinder address.
1B	Velocity calibration detected too many bad seeks.
1C	Unsuccessful recalibrate command detected during a velocity calibration.
1D	Drive seeks are more than 10 percent too fast.
1E	Drive seeks are more than 15 percent too slow.
1F	A sector pulse is detected during the execution of a read or write of a sector.
20	A parity is error detected on the controller real-time state line.
21	Two or more pulses of the same polarity are detected on the controller real-time state line (control pulse error).

Drive Error Display Codes (Cont)

Error Code	Name
22	Two or more pulses of the same polarity are detected on the controller write command data line (data pulse error).
23	Spindle motor interlock broken (belt tension lever is released).
25	Servo error (SVOCHK) detected (off track during detent mode).
26	Spindle speed is detected as being less than 3420 r/min (3600-5%).
27	HDA temperature is too high.
28	Servo module temperature is too high.
29	Invalid error recovery level is specified.
2A	Invalid subunit is specified.
2B	Invalid region is specified in a diagnose command.
2C	Seek or recal command tried while the spindle is not spinning.
2D	Invalid command timeout value is given.
2E	Controller flags are detected prohibiting drive spinup.
2F	RUN/STOP switch is in the stop position while trying a run command.
30	Write current is turned on without write gate being asserted.
31	A read command is tried with write gate asserted.
32	A read or write command is tried while the drive is faulted.
33	A burst error is detected while writing.
34	Read data separator/encoder error.
35	Write unsafe error detected while trying a write command.
36	Short circuit is detected in head winding.
37	No write current is detected with write gate asserted.
38	A read command is tried with multiple heads selected.
39	A write command is tried while the positioner is off track (not detented).
3A	A write command is tried while the drive is write protected.
3B	Servo/HDA interlock is broken.
3C	Servo interlock is broken.
3D	Read/Write interlock is broken.
3E	Control panel interlock is broken.
3F	Personality module interlock is broken.
40	Invalid R/W region is specified.
41	SDI controller response timed out.
42	Drive is not in an on-line state while trying a seek command.

Drive Error Display Codes (Cont)

Error Code	Name
43	R/W READY is not set while trying a real-time command.
44	Format enable is not set while trying a format command.
45	Real-time command has an invalid head (track) address.
46	R/W safety interrupt occurred with no cause bits set.
47	Disconnect command has an incorrect TT bit.
48	Invalid write memory is offset or byte count.
49	Invalid command found while in topology mode.
4A	Drive is disabled by DD bit.
4B	Error in Index pulse.
4D	Bad embedded servo data found during a write command.
50	Slave diagnostic timeout. NOTE This error could be the result of the positioner lock lever in the lock position.
51	The sector/byte failed to count properly.
52	Group 0 cannot be selected.
53	Group 1 cannot be selected.
54	R/W head select error while running the head select multiplexer test.
57	Master RAM timer is faulty.
60	R/W head select error while running read/write test.
61	R/W data setup error (diagnostic write data register is not equal to diagnostic read data reg).
62	The data from three or more heads is bad while running the read-only test.
63	The data from one or two heads is bad while running the read-only test.
65	Read/Write test sector could not be found within two disk revolutions.
66	Read failure is caused by servo being off-track.
67	Write test cannot be run because Test F was not run or failed.
68	Read-only cylinder cannot be reformatted without jumper.
69	Read/Write diagnostic comparison circuitry never detects an error.
6F	Diagnostic write tried while write protected.
70	Command available timeout error during execution of PCB wrap test.
71	SDI INIT is always asserted.
72	No control clock or framing error occurred during execution of PCB wrap test.
73	Microprocessor and PCB board data bus communication error.
74	Initial personality board status incorrect for off-line condition.

Drive Error Display Codes (Cont)

Error Code	Name
75	Failure is in control clock error detect circuit.
76	Failure is in data clock error detect circuit.
77	Port A select latch failed.
78	Error is in Port A data transmitter/receiver.
79	Error is in Port A control line transmitter/receiver.
7A	Port B select latch failed.
7B	Spindle motor spinning while trying PCB test.
7C	Received error in frame code during PCB wrap test.
7D	Received error in data byte during PCB wrap test.
80	Consistency fault detected in master ROM.
81	Command available did not reset after reading data.
82	Frame code did not reset after reading data.
83	Cannot receive SDI INIT from Port B.
84	INIT bit not clear following INIT.
85	Master RAM 0 failed RAM test.
86	Master RAM 1 failed RAM test.
87	Checksum error detected in master ROM 0.
8F	Checksum error detected in master ROM 1.
90	Error in Port B data transmitter/receiver.
91	Error in Port B control line transmitter/receiver.
92	Error in Port A wraparound.
93	Error in Response serializer.
94	Looparound frame was not received properly.
95	Looparound frame was not decoded properly.
96	Looparound frame data byte was not received properly.
97	Checksum error detected in master ROM 2.
9F	Checksum error detected in master ROM 3.
A0	Unable to clear faults at R/W safety register.
A1	Unable to force head short error while writing.
A2	Unable to force multiple head select error while reading.
A3	Unable to force read gate and write gate active at the same time.
A4	Unable to force write current without write gate.
A5	Unable to force write gate without write current.
A6	Unable to force a miscompare of the data from the write compensation logic and the read encoder logic.
A7	Checksum error was detected in master ROM 4.
A9	Servo fault caused a R/W forced fault.
AA	Servo fault caused a R/W forced fault.
AA	Diagnostic firmware okay prompt code.

Drive Error Display Codes (Cont)

Error Code	Name
AF	Checksum error detected in master ROM 5.
B0	Test will not run due to a drive fault.
B3	Bus test error in microprocessor module.
B4	Bus test error in personality module.
B6	Error in UART transmitter/receiver.
B8	Stream test write clock is not active.
B9	Stream read clock is not active.
BA	Stream buffered servo clock is not active.
BB	Stream ECL write data is not present.
BC	Stream encoded data is not present.
BD	Stread decoded read data is not present.
BE	Stream ECL read data is not present.
BF	Unable to reset stream bit.
C0	Error in fine track timer.
D9	Load from inner guard band failed.
DA	Spindle is not spinning.
DB	Diagnostic seek failed.
DC	Incorrect parameters entered during execution of drive-resident diagnostics.
DD	Recalibrate error.
DE	Low velocity seek error.
DF	Spinup failed.
E0	Random seek error.
E1	Integrator error.
E2	Slave ROM consistency error.
E3	Checksum error on slave ROM 0.
E4	Checksum error on slave ROM 1.
E5	Checksum error on slave ROM 2.
E6	Checksum error on slave ROM 3.
E7*	Test execution code.
E8	Error in slave RAM 0.
E9	Error in slave RAM 1.
EE	Incorrect test parameter entered.
F0	Slave uncommanded spin-down.
F1	Slave load timeout.
F2	Slave sent an unexpected message.
F3	Slave received a bad command packet.
F4	Slave operation code parity error.
F5	Slave received an invalid operation code.
F6	Master received a bad status packet from slave.

Drive Error Display Codes (Cont)

Error Code	Name
F7	Slave recalibrate timeout. NOTE This error could be the result of the positioner lock lever in the lock position.
F8	Slave seek timeout.
F9	Slave offset timeout.
FA	Slave spin-up timeout.
FB	Slave spin-down timeout.
FC	Slave send status timeout.
FD	Slave initialization timeout.
FE	Slave speed check timeout.

FRONT PANEL FAULT IDENTIFICATION
The following table shows the status of the front panel lights for each type of error reported.

Drive Front-Panel Fault Identification Codes

Fault Condition	RUN/ STOP	FAULT	RDY	WRITE PROT	A	B	Status Byte 15 Hex Code
Index error	on	on					11
Master/slave error		on		on			12
Servo diagnostic test error	on	on		on			13
Microprocessor fault		on			on		14
Drive disabled by DD bit	on	on			on		15
Servo coarse positoning error		on		on	on		16
Diagnostic idle loop test error	on	on		on	on		17
Spin error		on				on	18
Write and bad embedded data	on	on				on	19
Servo fine positoning error		on		on		on	1A
R/W diagnostic test error	on	on		on		on	1B
SDI error		on			on	on	1C
Write enable and write protect asserted error	on	on			on	on	1D
Spindle motor interlock error		on		on	on	on	1E
Servo or HDA overtemp error		on	on				30
Servo/microprocessor interlock error		on	on	on			32

Drive Front-Panel Fault Identification Codes (Cont)

Fault Condition	RUN/ STOP	FAULT	RDY	WRITE PROT	A	B	Status Byte 15 Hex Code
R/W command error		on	on		on		34
Control panel/ microprocessor interlock error		on	on	on	on		36
R/W unsafe error		on	on			on	38
Read/write microprocessor interlock error		on	on	on		on	3A
Servo/HDA interlock error		on	on		on	on	3C
Personality/ microprocessor interlock error		on	on	on	on	on	3E
Microprocessor hardcore test error	on	on	on	on	on	on	3F
DC low	on		on	on	on	on	*

NOTE
The DC Low condition locks up the drive logic and, therefore, cannot have a hexadecimal code represented by the host system diagnostics.

RA60 FRONT PANEL CODES
The following table describes the fault, indicates the front panel light configuration, and identifies the hex code of the lights.

Description of Error	RUN/ STOP	FAULT	RDY	WRITE PROT	A	B	Hex Code
Microcode error (not sent to front panel)	*	off	off	off	off	off	00
Heads home switch fault	*	on	off	off	off	on	01
Front panel fault	*	on	off	off	on	off	02
Long spin-up time fault	*	on	off	off	on	on	03
Bad servo samples fault	*	on	off	on	off	on	05
Bad velocity fault	*	on	off	on	on	off	06
Linear mode fault	*	on	off	on	on	on	07
Retry on seek fault	*	on	on	off	off	off	08
Lost servo samples fault	*	on	on	off	off	on	09
Heads home switch would not open	*	on	on	off	on	off	0A
Master processor fault	*	on	on	off	off	off	10
SDI fault	*	on	on	off	off	on	11
Invalid SDI level 1 command	*	on	on	off	on	on	13
Master processor fault	on	on	on	on	on	on	3F

*The run/stop switch may be on or off depending on the state of the drive.

RA60 FRONT PANEL FRU CALLOUT
The fault code, a description of the fault, and the most likely failing FRUs are shown below.

RA60 Front Panel Fault Codes

Front Panel Codes	Description	Most Likely FRU Failure
00	Microcode error (not sent to front panel	1. None
01	Heads home switch fault	1. Push heads home 2. Check P803 3. Heads home switch assembly
02	Front panel fault	1. Front panel module 2. Drive logic module 3. Front panel ribbon cable
03	Long spin-up time fault	1. Check spindle motor connections 2. Heat sink module 3. Spindle motor 4. Drive logic module
05	Bad servo samples fault	1. Pack 2. Post amp/data separator module 3. R/W preamplifier module 4. Heads
06	Bad velocity fault	1. Pack 2. Post amp/data separator module 3. R/W preamplifier module 4. Drive logic module 5. Positioner assembly
07	Linear mode fault	1. Pack 2. Post amp/data separator module 3. Drive logic module
08	Retry on seek fault	1. Pack 2. Post amp/data separator module 3. R/W preamplifier module 4. Drive logic module
09	Lost servo samples fault	1. Pack 2. Heads 3. Post amp/data separator module
0A	Heads home switch would not open	1. Shipping pin not removed 2. Check P803 3. Heads home switch assembly 4. Heatsink module
10	Master processor fault	1. Drive logic module 2. SDI module 3. Post amp/data separator module 4. Backplane module
11	SDI Faults	1. SDI module 2. Drive logic module 3. SDI cable 4. Backplane module
13	Invalid SDI level 1 command	1. SDI module 2. Post amp/data separator module 3. R/W preamplifier module 4. Backplane module

RA60 Front Panel Fault Codes (Cont)

Front Panel Codes	Description	Most Likely FRU Failure
3F	Master processor fault	1. Drive logic module 2. SDI module 3. Post amp/data separator module 4. Backplane module

RA60 ERROR MESSAGES
The following table provides the error code, a description of the error, and indicates the most likely failing Field Replaceable Units (FRUs).

RA60 Error Messages

Error Code	Description	Most Likely FRU Failure
01	Command cannot be executed with the drive in its current state	1. None
02	Nonexistent head or cylinder requested with seek	
03	Opcode was not one of six valid opcodes	
04	Cover was not closed when run was issued	
05	Lid was not locked when run command was issued	
06	There was a program error during head load	
08	Heads were not home when run command was issued	1. Push heads home 2. Check P803
09	Heads home switch opened during spin-up	3. Heads home switch assembly
0A	Heads home switch did not close during unload	
0B	Cover or lid lock opened during head load	
0C	Spin-up L was not low during spin-up	1. Front panel module
10	Lid lock opened during spin-up	2. Drive logic module
11	Cover opened during spin-up	3. Front panel ribbon cable
12	Cover or lid lock opened during head load	
13	Lid was not locked while run command was issued	
18	Spin-up took too long	1. Check spindle motor connections
19	Motor sample did not change during spin-up	2. Heat sink module
20	Spin-up required to short a time	3. Spindle motor
24	Too long to acquire spindle speed control on head load	4. Drive logic module
26	More than 32 sectors to settle on track	1. Pack

RA60 Error Messages (Cont)

Error Code	Description	Most Likely FRU Failure
27	More than 32 sectors to settle on track retry failure	2. Post amp/data separator module
28	Too many bad servo samples during RTZ	3. R/W preamplifier
29	Too long to get good samples on head load	4. Heads
2A	Head load seek did not return to track 0	
2B	Bad servo samples when bumping guard band on head load	
2C	Bad servo samples during RTZ on head load	
2D	Bad samples during linear mode	
2E	Off speed when in linear mode	
2F	Off track in linear state	1. Pack
30	Velocity was too high during RTZ on head load	2. Post amp/data separator module
31	Velocity was too low during RTZ on head load	3. R/W preamplifier module
32	Bad velocity on seek	4. Drive logic module
		5. Positioner assembly
38	Too long to get on track in linear mode	1. Pack
		2. Post amp/data separator module
		3. Drive logic module
40	Retry on seek due to bad servo samples	1. Pack
		2. Post amp/data separator module
41	Retry on seek due to bad servo samples failed	
		3. R/W preamplifier module
42	Retry on seek due to bad guard band flags	
		4. Drive logic module
43	Retry on seek due to bad guard band flags failed	
44	Retry on see due to seek timeout	
45	Retry on see due to seek timeout failed	
46	Retry on seek due wrong grey code	1. Pack
		2. Heads
47	Retry on seek failed due to wrong grey code	3. Post amp/data separator module
48	Lost servo samples	
50	Too long for heads home switch to open on head load	1. Shipping pin not removed
		2. Check P803
		3. Heads home switch assembly
		4. Heatsink module

RA60 Error Messages (Cont)

Error Code	Description	Most Likely FRU Failure
81	Inits and tests control panel	1. Drive logic module
82	Checks processor registers	2. SDI module
83	Tests RAM 0	3. Post amp/data separator module
84	Tests ROM 0	4. Backplane module
85	Tests ROM 1	
86	Tests ROM 2	
87	Verify ROM version numbers	
88	Tests the SDI clear interface bit	
89	Tests the SDI control/status register	1. Drive logic module
8A	Tests front panel and serial number ROM	2. SDI module
8B	Inits the UART	3. Post amp/data separator module
8C	Tests the error registers	4. Backplane module
8D	Tests the 8155 timer	
8E	Tests the slave control port	
8F	Slave Diagnostics	
90	Command available or slave done already set when issuing a new slave command	
91	Slave command receiver timeout on opcode	
92	Slave attention timeout	1. Drive logic module
93	Slave done timeout to slave stop (TOSTP)	2. SDI module
94	Slave done timeout to status (1 ms)	3. Post amp/data separator module
95	Slave done timeout	4. Backplane module
96	Slave attention timeout	
97	Solenoid release timeout	1. Drive logic module
98	Watchdog timer detected master insane	2. Post amp/data separator module
99	Watchdog timer detected slave insane	3. SDI module
9A	Run switch or cover invalid at spin-up	
9B	Spindle not ready during recalibrate command	
9C	SDI transfer error from DC703	1. SDI module
9D	Write or format when write protected	2. Post amp/data separator module
9F	Transfer command when drive error	4. Backplane module
A0	Sector read/write overrun error	

RA60 Error Messages (Cont)

Error Code	Description	Most Likely FRU Failure
A1	Transfer command when read/write error	
A2	SDI command checksum error	1. SDI module
A3	SDI frame error	2. Drive logic module
A4	SDI command parity error	3. Post amp/data separator module
A5	SDI command out of range	
A6	SDI command length error	4. Backplane module
A7	SDI error status byte was nonzero	
A8	Slave done timeout to seek command	1. Drive logic module
A9	Slave done timeout to RTZ command	2. SDI module 3. Post amp/data separator module
AA	Slave response error to RTZ command	4. Backplane module
AB	Slave done timeout during first pass	
AC	Command receiver timeout during diagnostic stop or ASCII port command time = todgn 2	
AD	Slave done timeout during diagnostic stop or ASCII port time = 1 MS	
AE	Drive is not on-line	1. Drive logic module
AF	Command bytes 4 and 5 are not zero	2. Post amp/data separator module
B0	Invalid group number	3. SDI module
B1	Invalid head select number	4. Backplane module
B2	Invalid cylinder range	
B3	Spindle not ready	
B4	Controller timeout = 0	1. Drive logic module
B5	Spindle not ready	2. SDI module
B6	Illegal memory region	3. Post amp/data separator module
B7	DD bit set	
B8	Tried to clear a hard fault	4. Backplane module
B9	Slave done timeout (1 MS) to get status	
B1	Nonzero level	1. Drive logic module
BB	Wrong subunit error	2. SDI module
BC	Diagnostic command, memory region high not zero	3. Post amp/data separator module
BD	Diagnostic command, invalid parameters	4. Backplane module
BE	Write enable a write protected drive	
BF	Setting S7 = 1	

REGULATOR MODULE LEDs
The following table lists possible causes for Regulator Module LED indications. Use the table to isolate power supply problems.

Regulator LEDs Fault Codes

Green D19 REF	Red D18 Thermal Fault	Red D17 +27.5 -27.5	Green D16 +5	Green D15 -5.2	Green D14 -15	Green D13 +15	Possible Causes: Order of Most Probable
1	0	0	1	1	1	1	Normal State
0	0	0	0	0	0	0	Suspect ref on Reg 1. Check AC cabling and switch plate assembly 2. Transformer 3. Regulator 4. Cap/rec assembly
0	0	1	1	1	1	1	27.5 volt failure 1. Heat sink module 2. Cap/rec assembly 3. Transformer 4. Regulator
0	0	0	at least one off				Suspect a short circuit 1. Logic module shorts 2. Heat sink module
0	0	0	0	0	1	1	Suspect short or over-voltage to +5 and -5.2 1. Logic module shorts 2. Check Circuit breaker 3. Heat sink module 4. Capacitor assembly 5. Regulator module
0	0	0	1	1	0	0	Suspect shorts to 15 V 1. Logic module shorts 2. Check circuit breaker 3. Heat sink module 4. Capacitor assembly 5. Regulator module
0	1	0	1	1	1	1	Thermal Shutdown 1. Check fans 2. Regulator module
0	1	0	at least on off				Thermal shutdown with faulted Supply 1. Check fans 2. Heat sink module 3. Regulator module

HSC 50 ERROR CODES
The following table lists the operator control panel fault codes for the HSC50.

Operator Control Panel Fault Codes

	HEXA-DECIMAL	OCTAL	INIT	FAULT	ON LINE		
PORT PROCESSOR MODULE FAILURE	01	01	OFF	OFF	OFF	OFF	ON
DISK DATA CHANNEL FAILURE	02	02	OFF	OFF	OFF	ON	OFF
TAPE DATA CHANNEL FAILURE	03	03	OFF	OFF	OFF	ON	ON
I/O CONTROL PROCESSOR MODULE FAILURE	11	21	ON	OFF	OFF	OFF	ON
MEMORY MODULE FAILURE	12	22	ON	OFF	OFF	ON	OFF
TU58 FAILURE	13	23	ON	OFF	OFF	ON	ON
PORT BUFFER MODULE FAILURE	14	24	ON	OFF	ON	OFF	OFF
PORT LINK MODULE FAILURE	15	25	ON	OFF	ON	OFF	ON
MISSING REQUIRED FILES	16	26	ON	OFF	ON	ON	OFF
ERROR LOG ATTENTION REQUIRED	17	27	ON	OFF	ON	ON	ON
NOT ENOUGH WORKING REQUESTORS IN SUBSYSTEM	18	30	ON	ON	OFF	OFF	OFF
REBOOT BEFORE PREVIOUS BOOT COMPLETE	19	31	ON	ON	OFF	OFF	ON
SOFTWARE DETECTED INCONSISTENCY	1A	32	ON	ON	OFF	ON	OFF

MR 15101

NOTE

If the OCP Power indicator is not on, ensure that the ac power cord is plugged in and that the computer room circuit breakers are switched on. If the Power indicator still does not come on, call your field service office.