HP 2100 Simulator User's Guide
3-Nov-2020

COPYRIGHT NOTICE

The following copyright notice applies to the SIMH source, binary, and documentation:

Original code published 1993-2012, written by Robert M Supnik
Copyright © 1993-2012, Robert M Supnik
Copyright © 2012-2020, J. David Bryan

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the names of the authors shall not be used in advertising or otherwise to promote the sale, use, or other dealings in this Software without prior written authorization from the authors.

1 Introduction....4
1.1 The HP 21xx/1000 Computer Systems....4
1.2 Simulator Files....5

2 Simulator Features....8
2.1 Hardware-Equivalent Actions....9
2.2 Simulator-Specific Commands....10
2.2.1 Numeric Display and Entry....10
2.2.2 Symbolic Display and Entry....10
2.2.3 EXAMINE, DEPOSIT, IEXAMINE, and IDEPOSIT....12
2.2.4 BREAK and NOBREAK....12
2.2.5 RESET....13
2.2.6 RUN and GO....13
2.2.7 LOAD, DUMP, and BOOT....13
2.2.8 Device Configuration....14
2.2.9 Enabling and Disabling Devices....15
2.2.10 I/O Interface Select Code Assignments....15
2.2.11 SAVE and RESTORE....16
2.3 Realistic, Calibrated, and Optimized Timing....16
2.4 The Simulation Console and the System Console....18
2.5 The Keyboard Poll Device....18
2.6 Tracing Simulator Operations....18

3 Processor Device Simulations....20
3.1 Central Processing Unit....20
3.1.1 Bootstrap Loaders....24
3.1.2 Simulation Stops....26
3.1.3 Stop Messages....28
3.1.4 Tracing....29
3.1.5 Registers....32
3.2 12578A/12607B/12895A Direct Memory Access Controllers, 12897B Dual-Channel Port Controller....32
3.3 12581A/12892B Memory Protect....34
3.4 12731A Memory Expansion Module....35

4 General I/O Device Simulations....36
4.1 12531C Buffered Teleprinter Interface with One 2752/54 Teleprinter....36
4.2 12539C Time Base Generator....38
4.3 12566B Microcircuit Interface....40
4.4 12597A-002 Tape Reader Interface with One 2748 Paper Tape Reader....41
4.5 12597A-005 Tape Punch Interface with One 2895 Paper Tape Punch....44
4.6 12620A/12936A Privileged Interrupt Fence....45
4.7 12653A Line Printer Interface with One 2767 Line Printer....46
4.8 12792C 8-Channel Asynchronous Multiplexer....49
4.8.1 Device Options....50
4.8.2 Unit Options....51

4.8.3 Tracing and Registers....52
4.9 12845B Line Printer Interface with One 2607/13/17/18 Line Printer....54
4.9.1 Device Options....54
4.9.2 Unit Options....55
4.9.3 Vertical Format Unit....56
4.9.4 Tracing and Registers....59
4.10 12875A Processor Interconnect....61
4.11 12920A Asynchronous Multiplexer Interface....64
4.11.1 Lower and Upper Data PCAs....65
4.11.2 Control PCA....69
4.12 12966A Buffered Asynchronous Communications Interface....70

5 Disc I/O Device Simulations....74
5.1 12557A Cartridge Disc Interface with Four 2870 Drives, 13210A Disc Drive Interface with Four 7900 Drives....74
5.2 12565A Disc Interface with Two 2883 Drives....77
5.3 12606B Disc Memory Interface with One 2770/2771 Disc, 12610B Drum Memory Interface with One 2773/2774/2775 Drum....80
5.4 12821A Disc Interface with Four 7906H/7920H/7925H Drives....83
5.4.1 Device Options....84
5.4.2 Unit Options....85
5.4.3 LOAD and BOOT Commands....85
5.4.4 Tracing and Registers....86
5.5 13175D Disc Controller Interface with Eight 7905/7906/7920/7925 Drives....87
5.5.1 Device Options....88
5.5.2 Unit Options....88
5.5.3 LOAD and BOOT Commands....89
5.5.4 Tracing and Registers....90

6 Magnetic Tape I/O Device Simulations....92
6.1 12559C 9-Track Magnetic Tape Unit Interface with One 3030 Drive....92
6.2 13181B Digital Magnetic Tape Unit Interface with Four 7970B Drives, 13183B Digital Magnetic Tape Unit Interface with Four 7970E Drives....93
6.2.1 Device Options....94
6.2.2 Unit Options....95
6.2.3 LOAD and BOOT Commands....96
6.2.4 Tracing and Registers....97

1 Introduction

This manual documents the features and operation of the HP 2100 simulator. It is used in conjunction with the *SIMH Users' Guide* and *SIMH Users' Guide Supplement* manuals, which describe how to compile and run the simulator, as well as the general commands that may be entered at the Simulation Control Program (SCP) prompt.

This simulator is based on SIMH — the Computer History Simulator.

1.1 The HP 21xx/1000 Computer Systems

Hewlett-Packard sold the HP 21xx/1000 family of real-time computers from 1966 through 2000. There are three major divisions within this family: the 21xx core-memory machines, the 1000 (originally 21MX) M/E/F-Series semiconductor-memory machines, and the 1000 L/A-Series distributed-I/O machines. All machines are 16-bit accumulator-oriented CISC machines running the same base instruction set. A wide range of operating systems run on these machines, from a simple 4K word paper-tape-based monitor to a megaword multi-user, multiprogramming disc-based system and a multi-user time-shared BASIC system.

Within each division, several distinct models were produced:

- the HP 2116; introduced in 1966, expandable to 32 K words (originally 16 K words).
- the HP 2115 (1967); slower, more I/O-limited, expandable to 8 K words.
- the HP 2114 (1968); low-cost, still-more-limited expandability, expandable to 8 K words.
- the HP 2100 (1971); user-microprogrammable, optional floating-point instructions, expandable to 32 KW.
- the HP 21MX (1974, renamed 1000 M); memory-mapped, larger instruction set, expandable to 1024 KW.
- the HP 21MXE (1976, renamed 1000 E); faster CPU and memory, larger microprogram address space.
- the HP 1000 F-Series (1979); an E-Series CPU with a separate hardware floating-point processor.
- the HP 1000 L-Series (1979); low-cost, slower, distributed I/O system, limited to 32K words.
- the HP 1000 XL-Series (1980); an L-Series CPU with memory mapping, expandable to 256 KW.
- the HP 1000 A600 (1982); E-Series speed with L-Series distributed I/O, expandable to 2048 KW.
- the HP 1000 A700 (1982); an A600 with an optional hardware floating-point processor.
- the HP 1000 A900 (1983); faster pipelined processor, cache memory, expandable to 3072 KW.
- the HP 1000 A400 (1986); lower-cost, limited I/O expandability, expandable to 2048 KW.
- the HP 1000 A990 (1991); fastest and last 1000 made, expandable to 4096 KW.

HP claimed the relative CPU performance, normalized to a 1000 E-Series, as follows:

2114	***2115***	***2116***	***2100***	***1000 M***	***1000 E***	***1000 F***	***1000 L***	***A400***	***A600***	***A700***	***A900***	***A990***
0.22	0.37	0.42	0.55	0.45	1.0	1.4	0.3	1.1	1.0	1.67	3.3	6.7

All of the machines support a 15-bit logical address space, addressing a maximum of 32K words, divided into 1K-word pages. Memory-referencing instructions in the base set can directly address the 1024 words of the base page (page 0) or the 1024 words of the current page (the page containing the instruction). The instructions in the extended set directly address the 32768 words in the full logical address space. The A and B accumulators may be addressed as logical addresses 0 and 1, respectively.

Peripheral devices are connected to the CPU by interface cards installed in the I/O card cages present in the CPU and optional I/O extender chassis. Each slot in the card cage is assigned an address, called a *select code*, that may be referenced by I/O instructions in the base set. Select codes range from 0 to 77 octal, with the first eight select codes reserved for the system, providing connections for 56 possible interfaces.

The 211x machines use a hardwired processor providing 70 basic instructions and up to 32K of core memory. The base instruction set is divided into the Memory Reference Group, the Shift-Rotate Group, the Alter-Skip Group, and the I/O Group. SRG instruction words may contain from one to four suboperation codes that are executed from left-to-right, and ASG instruction words may contain from one to eight suboperations. An optional Extended Arithmetic Unit may be added to the 2115 and 2116 that provides hardware multiply and divide, double-load and -store, and double-word shift and rotate instructions.

The 2100 machine uses a microprogrammed processor that provides the 80 instructions of the base set and the EAU as standard equipment. Optional floating-point microcode adds six two-word single-precision instructions. User microprogramming is also supported. When used as part of an HP 2000 Access Time-Shared BASIC system, the CPU designated as the I/O processor is equipped with microcode implementing 18 additional OS accelerator instructions.

The 1000 M/E-Series machines also use microprogrammed processors and extend the 2100 instruction set with two new index registers, X and Y, and a new Extended Instruction Group consisting of 32 index-register instructions and 10 word-and-byte-manipulation instructions. The six 2100 floating-point instructions are also standard. The 1000 F-Series adds a hardware floating-point processor with 18 new triple- and quad-word instructions. A number of new optional microcode extensions are available with the M/E/F-Series.

The 21xx and 1000 machines provide Direct Memory Access and Dual-Channel Port Controller options, respectively, for high-speed I/O transfers. The 2114 supports a single DMA channel; all other machines support two DMA or DCPC channels.

1000 CPUs offer the optional Dynamic Mapping System, which provides memory mapping on a page-by-page basis. The 5-bit page number of a logical memory address selects one of 32 ten-bit map registers containing physical page numbers. The ten-bit page number combined with the ten-bit page offset yields a 20-bit physical address capable of accessing a location in a one-megaword memory. DMS provides separate maps for system and user programs, as well as for the two DCPC channels, and includes microcode that implements the 38 Dynamic Mapping Instructions used to manipulate the mapping system.

Optional memory protection is accomplished by dividing the logical address space into protected and unprotected parts. When protection is enabled, any attempt to write below the fence separating the two parts is inhibited, and an interrupt to the operating system occurs, which aborts the offending user program. If the DMS option is enabled as well, protection is enhanced by specifying read and write permissions on a page-by-page basis.

This implementation is a simulator for the HP 21xx and 1000 M/E/F-Series machines (hereafter referred collectively as the HP computer family unless a specific machine reference is required). A large variety of CPU options, device interface cards, and peripherals are provided.

This simulator does not model the 1000 L/A-Series machines. While these machines use the same base instruction set, the I/O structure is entirely different, and the interfaces are not interchangeable with those of the 21xx/1000 machines. Moreover, the peripherals provided by this simulator are, with few exceptions, not supported on the L/A-Series. An L/A-Series implementation would be almost a completely new simulator and therefore likely would be best constructed as a separate project.

1.2 Simulator Files

The simulator sources are divided into a set of files for the Simulator Control Program and its support libraries, and a set of files for the HP 21xx/1000 CPU and device simulations; the latter reside in a subdirectory of the directory that contains the SCP files. The former set is common to all SIMH simulators, whereas the latter set is specific to the virtual machine being simulated. The files that make up this simulator are:

Subdirectory	*File*	*Contains*
HP2100	hp2100_cpu.h	CPU architectural declarations
	hp2100_cpu_dmm.h	CPU-to-DMA/MEM/MP interface declarations
	hp2100_cpu_fp.h	Floating-point interface declarations
	hp2100_defs.h	System architectural declarations
	hp2100_di.h	12821A HP-IB Disc Interface simulator declarations
	hp2100_disclib.h	13037D MAC/ICD disc controller simulator library declarations
	hp2100_io.h	Device-to-CPU interface declarations
	hp2100_baci.c	Buffered Asynchronous Communications Interface simulator
	hp2100_cpu.c	Central Processing Unit simulator
	hp2100_cpu0.c	UIG, user microcode, and unimplemented instruction simulator
	hp2100_cpu1.c	EAU, FP, and IOP instruction set simulator
	hp2100_cpu2.c	DMS and EIG instruction set simulator
	hp2100_cpu3.c	FFP and DBI instruction set simulator
	hp2100_cpu4.c	FPP and SIS instruction set simulator
	hp2100_cpu5.c	RTE-IV EMA, VIS, and SIGNAL/1000 instruction set simulator
	hp2100_cpu6.c	RTE-6/VM OS instruction set simulator
	hp2100_cpu7.c	RTE-6/VM VMA instruction set simulator
	hp2100_cpu_fp.c	Firmware floating point instructions simulator
	hp2100_cpu_fpp.c	Floating-Point Processor simulator
	hp2100_di.c	12821A HP-IB Disc Interface simulator
	hp2100_di_da.c	HP-IB Disc Interface simulator for Amigo disc drives
	hp2100_disclib.c	13037D MAC/ICD disc controller simulator library
	hp2100_dma.c	DMA/DCPC simulator
	hp2100_dp.c	12557A/13210A Disc Interface simulator
	hp2100_dq.c	12565A Disc Interface simulator
	hp2100_dr.c	12606B/12610B Disc/Drum Memory Interface simulator
	hp2100_ds.c	13175D Disc Controller Interface simulator
	hp2100_ipl.c	12875A Processor Interconnect simulator
	hp2100_lps.c	12653A Line Printer Interface simulator
	hp2100_lpt.c	12845B Line Printer Interface simulator
	hp2100_mc.c	12566B Microcircuit Interface simulator
	hp2100_mem.c	Main memory/Memory Expansion/Memory Protect simulator
	hp2100_mpx.c	12792C 8-Channel Asynchronous Multiplexer simulator
	hp2100_ms.c	13181B/13183B Digital Magnetic Tape Unit Interface simulator
	hp2100_mt.c	12559C 9-Track Magnetic Tape Unit Interface simulator
	hp2100_mux.c	12920A Asynchronous Multiplexer Interface simulator
	hp2100_pif.c	12620A/12936A Privileged Interrupt Fence simulator
	hp2100_pt.c	12597A-002/005 Paper Tape Reader/Punch Interface simulator
	hp2100_sys.c	SCP interface
	hp2100_tbg.c	12539C Time Base Generator Interface simulator
	hp2100_tty.c	12531C Buffered Teleprinter Interface simulator
doc	hp2100_guide.pdf	HP 2100 Simulator User's Guide
	hp2100_bugfixes.txt	HP 2100 simulator bug fix reports (historical)
	hp2100_diag.txt	SIMH/HP 2100 Diagnostics Performance Report
	hp2100_release.txt	HP 2100 Simulator Release Notes

PDF files of the original HP 21xx and 1000-series hardware and software manuals are available from these repositories:

- Bitsavers — http://www.bitsavers.org/
- The HP Computer Museum — http://www.hpmuseum.net/

The reference manuals used to develop the various simulated devices are listed in the comments at the start of the respective simulator source modules.

See the *Available Software* section of the release notes file for information on the availability of operating system software that runs on the simulator.

The simulator supports the dual-CPU HP 2000 Time-Shared BASIC systems (2000C', 2000F, and 2000 Access). A paper describing the implications involved in synchronizing two independent simulator processes is available from the simulator distribution site:

- Running HP Time-Shared BASIC on SIMH

A paper describing the internal design of the I/O hardware simulation is also available:

- The Evolution of the HP21xx/1000 I/O Simulation

Additional software and documentation resources are listed in the *hp2100_release.txt* file.

2 Simulator Features

The HP 2100 simulator provides the following device simulations:

Device Name(s)	*Simulates*
CPU	2114C Computer with up to 16K words of memory 2115A Computer with up to 8K words of memory 2116C Computer with up to 32K words of memory 2100A Computer with up to 32K words of memory 1000 M/E/F-Series Computer with up to 1024K words of memory
DCPC1, DCPC2	12897B Dual-Channel Port Controller
DMA1, DMA2	12578A/12607B/12895A Direct Memory Access
MEM	12731A Memory Expansion Module
MP	12581A/12892B Memory Protect
BACI	12966A Buffered Asynchronous Communications Interface
DA	12821A Disc Interface with four 7906H/7920H/7925H drives
DPD, DPC	12557A Cartridge Disc Interface with four 2870 drives 13210A Disc Drive Interface with four 7900 drives
DQD, DQC	12565A Disc Interface with two 2883 drives
DRD, DRC	12606B Disc Memory Interface with one 2770/2771 disc 12610B Drum Memory Interface with one 2773/2774/2775 drum
DS	13175D Disc Controller Interface with eight 7905/7906/7920/7925 drives
IPLI, IPLO	12875A Processor Interconnect
LPS	12653A Line Printer Interface with one 2767 line printer
LPT	12845B Line Printer Interface with one 2607/13/17/18 line printer
MC1, MC2	12566B Microcircuit Interface
MPX	12792C 8-Channel Asynchronous Multiplexer
MSD, MSC	13181B Digital Magnetic Tape Unit Interface with four 7970B drives 13183B Digital Magnetic Tape Unit Interface with four 7970E drives
MTD, MTC	12559C 9-Track Magnetic Tape Unit Interface with one 3030 drive
MUX, MUXL, MUXC	12920A Asynchronous Multiplexer Interface
PIF	12620A/12936A Privileged Interrupt Fence
PTP	12597A-005 Tape Punch Interface with one 2895 paper tape punch
PTR	12597A-002 Tape Reader Interface with one 2748 paper tape reader
TBG	12539C Time Base Generator Interface
TTY	12531C Buffered Teleprinter Interface with one 2752/54 teleprinter

One instance of each listed device may be installed in the simulated computer chassis. Many devices support multiple connected units. As an example, the DS device simulates a single 13175D Disc Controller Interface that connects up to eight drives. However, installing a second 13175D to connect an additional eight drives is not supported.

The simulator has been tested with and supports the following operating systems:

- 2000E Time-Shared BASIC revision 1534.
- 2000F Time-Shared BASIC revision 1410.
- 2000 Access Time-Shared BASIC revision 1812.

- DOS Disc Operating System revision C.
- DOS-M Disc Operating System revision F.
- DOS-III Disc Operating System revision 1511.
- RTE-II Real-Time Executive revision 1926.
- RTE-III Real-Time Executive revision 1926.
- RTE-IV Real-Time Executive revision 2013.
- RTE-IVB Real-Time Executive revision 5010.
- RTE-6/VM Real-Time Executive revision 6210.

In addition, the simulator generally passes the HP 24396 Stand-Alone Diagnostics suite; see *hp2100_diag.txt* for details.

The simulator may be configured to stop for any of these conditions:

- Attempted execution of an unimplemented or undefined instruction.
- Attempted execution of an I/O instruction that addresses an unassigned select code.
- Attempted execution of a memory reference instruction with too many indirect references.
- Occurrence of an I/O error on a device that does not report error status to the CPU.

The simulator also provides extensive facilities for tracing CPU and I/O device operations.

2.1 Hardware-Equivalent Actions

The current implementation does not provide simulations of the CPU or peripheral device front panels. Instead, commands entered through the simulation console are used to perform hardware actions. The simulation commands that substitute for CPU front-panel actions are:

Hardware Front-Panel Action	***Equivalent Simulation Command***
Pressing the RUN button	GO
Pressing the HALT button	CTRL+E
Pressing the INSTR STEP button	STEP
Pressing the PRESET button	RESET
Pressing the PRESET and RUN buttons	RUN
Pressing the IBL button	LOAD CPU
Pressing the IBL, PRESET, and RUN buttons	BOOT CPU
Pressing the LOADER ENABLE button	SET CPU LOADERENABLE
Displaying memory	EXAMINE <address>
Displaying registers	EXAMINE <register>
Setting memory	DEPOSIT <address> <value>
Setting registers	DEPOSIT <register> <value>
Setting the switch register	DEPOSIT S <value>

Memory may also be displayed, but not set, by setting the M (memory address) register to the desired memory address and displaying the T (memory data) register.

Mounting media on a peripheral device is simulated by the ***ATTACH*** command. For example, entering the ***ATTACH PTR <tape-image-filename>*** command is equivalent to loading the paper tape into an HP 2748B Tape Reader. Inserting a disc pack into an HP 7900A disc drive set for unit 2 is simulated by the ***ATTACH DPC2 <disc-image-filename>*** command.

Each of these commands is explained in more detail below.

In hardware, loading programs into memory from a device is accomplished by running a bootstrap loader that is configured for the select code(s) used by the device interface. For the 2114, 2115, 2116, and 2100 CPUs, this involves enabling the protected area of core memory containing the loader, setting the P register to point at the loader's starting address, and pressing PRESET (or INTERNAL PRESET and EXTERNAL PRESET for the 2100) and RUN. For the 1000 M/E/F-Series CPUs, loading involves setting the S register to select the bootstrap loader ROM and interface select code, and pressing STORE, IBL, PRESET, and RUN.

In simulation, bootstrap loaders may be loaded and run explicitly or implicitly. Explicit operation is described in the *Bootstrap Loaders* section of the CPU device description below. As a convenience, ***BOOT <device>*** commands may be used to implicitly bootstrap their respective devices and are described in the individual device descriptions below.

2.2 Simulator-Specific Commands

In general, all of the commands documented in the *SIMH Users' Guide* manual are available for use with the HP 2100 simulator. Commands whose execution or parameters are implementation-defined are specified below.

2.2.1 Numeric Display and Entry

When examining or depositing into memory, the radix for addresses is octal, and the default radix for numeric data is octal. The data default may be changed for a specific device with the ***SET <device> <radix>*** command, or the radix may be overridden temporarily with a command line switch, as follows:

SET <radix> Option	*Command Switch*	*Data Interpretation*
BINARY	**–B**	Binary values
OCTAL	**–O**	Octal values
DECIMAL	**–D**	Decimal values
HEX	**–H**	Hexadecimal values

When examining or depositing into device registers, the default radix for the specified register is used unless overridden with one of the above command line switches. Defaults are listed in the register table associated with each device.

2.2.2 Symbolic Display and Entry

When examining or depositing into memory or certain registers, command line switches specifying the symbolic mode and format may be used to override the default numeric mode, as follows:

Switch	*Mode Interpretation*
–A	A single character in the right-hand byte
–C	A two-character packed string
–M	A CPU instruction mnemonic

In the absence of a mode switch or a specified symbolic default, entering values with a leading ' (apostrophe) implies ***–A***, a leading " (quotation mark) implies ***–C***, and a leading alphabetic or punctuation character implies ***–M***. The specific registers supporting symbolic mode are indicated in their respective device sections below.

If the ***–C*** switch is specified, the value is displayed as two characters separated by a comma. Alphanumeric, punctuation, and symbol characters are displayed within apostrophes, control characters are displayed as ASCII name abbreviations, and characters above 128 decimal are displayed in escaped numeric form with a leading backslash followed by an octal number. Depositing with ***–C*** accepts two displayable characters. If a single character is supplied, the low byte of the resulting value will be zero; follow the character with a space to pad the low byte with a blank.

If the ***–M*** switch is specified, the value is displayed or accepted as a CPU machine instruction mnemonic if it is defined and implemented in the currently enabled firmware set. If it is not, it is displayed as a numeric value in the CPU's data radix or is rejected if depositing. Any numeric operands present are displayed or accepted in a default radix unless overridden by the addition of one of these mutually exclusive format switches:

Switch	***Format Interpretation***
–A	A single character in the right-hand byte
–B	A binary value
–O	An octal value
–D	A decimal value
–H	A hexadecimal value

Operand values use a default radix suitable to the type of the value, as follows:

- Operand memory addresses and I/O instruction select codes use the CPU's address radix, which is octal.
- Extended Arithmetic Unit instruction shift and rotate counts and I/O Processor instruction index displacements use decimal unless overridden by a format switch on the command line.
- I/O Processor and Scientific Instruction Set instruction operand data values use the CPU's data radix, which defaults to octal but may be set to a different radix or overridden by a format switch on the command line.

If the CPU data radix is set to hexadecimal, caution must be exercised when entering hex values without a leading digit. A value that is the same as an instruction mnemonic will be interpreted as the latter. For example, ***CCE*** is an instruction mnemonic, but ***CCEE*** is a hex value. To avoid confusion, always enter hex values with the ***–H*** switch or with a leading zero (i.e., ***0CCE***).

Machine instruction entry uses the HP Assembler syntax, with the exceptions noted below. The set of recognized mnemonics depends on the current machine configuration. Instructions not implemented by the current CPU model and installed firmware extensions are displayed in octal form and rejected during symbolic entry.

Operands must be numeric and are interpreted by default in a radix suitable to the type of the value, rather than the HP Assembler convention of assuming a decimal value unless the letter B, indicating an octal value, follows the digits. For example, LDA 1000 loads from memory location 1000 octal, LIA 12 loads from select code 12 octal, and RRR 12 rotates the registers right by 12 decimal places.

Memory Reference Group instructions use this extended syntax:

```
<opcode> {C | Z} <address>{,I}
```

...where the optional ***C*** or ***Z*** specifies a current-page or a zero-page (base-page) address, and the optional ***I*** specifies indirect addressing. If ***C*** or ***Z*** is specified, the address is restricted to the octal range 0-1777; otherwise, the full 32K logical addressing range 0-77777 is allowed.

Memory Reference Group machine instructions address locations in either the 1024-word base page (page 0) or the 1024-word current page (the page containing the instruction). Normally, the base/current selection and page offset are implied by the address given: an address from 0-1777 is a base-page reference, an address within the page containing the deposited instruction is a current-page reference, and any other address is illegal. For example, entering ***DEPOSIT 11000 LDA 10000*** stores a current-page reference with an offset of 0. Entering ***DEPOSIT 11000 LDA 1000*** stores a base-page reference with an offset of 1000. Entering ***DEPOSIT 11000 LDA 12000*** is rejected as illegal, as the address is neither within the current page nor the base page.

C or ***Z*** must be specified when the instruction is being entered into memory above the 32K logical address space or into a device (e.g., disc) buffer. In these cases, the eventual containing address is unknown, making the implied current/base page determination impossible. Therefore, the current/zero indicator and the page offset must be entered explicitly.

2.2.3 EXAMINE, DEPOSIT, IEXAMINE, and IDEPOSIT

For these commands, the following address forms are valid:

- `<page>.<offset>`
- `<logical-address>`

...where ***page*** is a 10-bit physical page number and ***offset*** is a 10-bit offset within the page (e.g., ***1777.1777***), and ***logical-address*** is a 15-bit offset within the 32K logical address space (e.g., ***77777***). Command line switches modify the mapping of logical addresses to physical addresses as follows:

Switch	*Interpretation*
–N	Use the address directly with no mapping
–S	If memory expansion is enabled, use the system map
–U	If memory expansion is enabled, use the user map
–P	If memory expansion is enabled, use the port A map
–Q	If memory expansion is enabled, use the port B map

If no switch is specified, the address is interpreted using **–N** if memory expansion is currently disabled, otherwise it is interpreted using **–S** if the system map or **–U** if the user map is currently enabled. If mapping is requested, then memory expansion must be enabled, and the address must lie within the 32K logical address space.

Addresses within the first 32K are displayed in logical form. Addresses above 32K are displayed in physical form.

2.2.4 BREAK and NOBREAK

The ***BREAK*** and ***NOBREAK*** commands accept only logical addresses consisting of 15-bit offsets within the 32K logical address space (e.g., ***77777***). If the breakpoint address is not supplied, it defaults to the current P-register value. Breakpoint addresses are always displayed in logical form.

The CPU implements four different kinds of instruction breakpoints, as follows:

Switch	*Action*
–E	Break unconditionally
–N	Break if memory expansion is disabled
–S	Break if memory expansion and the system map are enabled
–U	Break if memory expansion and the user map are enabled

If no switch is specified, the default is to use **–N** if memory expansion is currently disabled, otherwise to use **–S** if the system map or **–U** if the user map is currently enabled. In a DMS environment, breakpoints on interrupt trap cells should specify the **–S** switch, as the system map will be enabled when the instruction is executed.

2.2.5 RESET

The ***RESET*** command is equivalent in hardware to pressing the PRESET front-panel button on the 211x and 1000 M/E/F-Series machines, and the INTERNAL PRESET and EXTERNAL PRESET buttons on the 2100. It presets all of the installed I/O interfaces. The ***RESET –P*** command is equivalent to cycling power on the CPU chassis and all installed peripheral devices. The ***RESET <device>*** and ***RESET –P <device>*** commands have no direct hardware analogs. They are functionally equivalent to presetting or power-cycling just the specified device and its associated interface card.

2.2.6 RUN and GO

The ***RUN*** and ***GO*** commands accept only logical addresses consisting of 15-bit offsets within the 32K logical address space (e.g., ***77777***). If the starting address is supplied, it is stored in the P register before execution begins. Otherwise, execution begins at the current P-register location.

The ***RUN*** command does an implied ***RESET***, so it is equivalent in hardware to pressing the front-panel PRESET and RUN buttons on the CPU. As PRESET initializes all of the I/O interfaces, the ***RUN*** command is almost never the proper command to use after a program has been started. The ***GO*** command is equivalent to pressing the RUN button on the CPU and is the usual way of resuming execution after a programmed halt. If a PRESET is required before resuming, the ***RESET*** and ***GO*** commands are direct analogs of pressing the PRESET and RUN buttons, respectively.

2.2.7 LOAD, DUMP, and BOOT

The ***LOAD*** command is used to install a bootstrap loader into the last 64 words of available memory within the 32K logical address space. The command accepts two forms. The syntax of the first form is:

```
LOAD <device-name>
```

This command copies the loader associated with the specified bootable device into memory, configures the loader to the select code(s) currently used by the device interface, and sets the P register to the starting address of the loader program. For a given device, the loader selected depends on the current CPU model, as shown in the following table:

Device Name	*HP 21xx Bootstrap*	*HP 1000 Bootstrap*	*Boot Device*
DA	—	12992H	7906H/7920H/7925H/9895 ICD Disc
DPC	BMDL-7900	12992F	2870/7900 Disc
DQC	BMDL-2883	12992A	2883 Disc
DRC	BBDL	—	277x Fixed-Head Disc/Drum
DS	BMDL-7905	12992B	7905/7906/7920/7925 MAC Disc
IPL	BBL-IOP	12992K	12875A Processor Interconnect
MSC	BMTL	12992D	7970B/E Magnetic Tape
PTR	BBL	12992K	2748 Paper Tape Reader
CPU	—	(various)	Selected by S-register bits 15-14

The BMDL (Basic Moving-Head Disc Loader), BBDL (Basic Binary Disc Loader), BMTL (Basic Magnetic Tape Loader), and the BBL-IOP (Basic Binary Loader for the I/O Processor) are dual-purpose loaders. In addition to the specified device, they also permit booting from the paper tape reader by manually starting the loader from an alternate address. When one of these loaders is specified, the paper-tape portion is configured to use the select

code of the paper tape reader interface. The BBL (Basic Binary Loader) and all of the 12992 boot loader ROMs for the 1000 are single-purpose loaders.

For 1000 systems, the ***LOAD CPU*** command is equivalent to pressing the front-panel IBL (Initial Binary Loader) button. This copies the content of one of four installed boot loader ROMs into memory. The ROM is selected by setting bits 15-14 of the S register to the desired ROM socket number before issuing the command. The IBL operation is described in the *Bootstrap Loaders* section of the CPU device description below.

The DA and CPU devices are not supported on 21xx CPUs, and the DRC device is not supported on 1000 CPUs. Consequently, there are no loaders for these device/CPU combinations, and a corresponding ***LOAD*** command will be rejected with a **Command not allowed** error.

The second command form installs a loader from an absolute binary file. The syntax is:

```
LOAD <image-filename> {<select-code>}
```

The file is read into the last 64 words of memory. If the optional select code is supplied, the loader I/O instructions are configured to the specified device; otherwise, the loader is not modified.

The binary file must be targeted to addresses in the octal range x7700-x7777, where *x* may be 0-7. The loaded program will be relocated to the last 64 words of available memory, so the desired memory size must be set before issuing the ***LOAD*** command. If the optional configuration select code is supplied, all I/O instructions in the program that reference select codes >= 10 octal will be changed by adding the supplied value minus 10 to the instruction. The effect of this is that instructions that reference select code *10* + *n* will be changed to reference the supplied *select code* + *n*. This permits the configuration of loaders using two-card interfaces that reference select codes 10 and 11.

For 1000-series CPUs, the ***LOAD*** command serves to install boot loader ROM images other than the ones included with the device simulators. If the CPU is configured with more than 32K of memory, the loader is installed in the last 64 words of the 32K logical address space.

For either ***LOAD*** command form, if the CPU is configured as a 2114, 2115, 2116, or 2100, then the loader remains unprotected when the command completes to permit subsequent execution. The loader should be protected with a ***SET CPU LOADERDISABLE*** command if it will not be used immediately.

The ***DUMP*** command writes the bootstrap loader currently residing in memory to an absolute binary file. The command format is:

```
DUMP <image-filename>
```

For 21xx CPUs, the loader must be unprotected with a ***SET CPU LOADERENABLE*** command before entering the ***DUMP*** command; if the loader is protected, the output file will contain all zeros. When the command completes, the loader remains unprotected. The resulting file may be used in a subsequent ***LOAD*** command to reload the bootstrap program.

The ***BOOT*** command is equivalent to a sequence of ***LOAD***, ***RESET***, and ***RUN*** commands. The same list of bootable devices accepted by the ***LOAD*** command above may be used with the ***BOOT*** command. Entering a ***BOOT*** command for one of these devices will preset the S register to the appropriate value before loading and executing the applicable binary loader. The bootstrap loaders support booting only from unit 0 of the target device.

2.2.8 Device Configuration

Most devices support user configuration. The general forms of the configuration commands are:

```
SET {<switch> ...} <device> <option>{,<option> ...}
SET {<switch> ...} <unit> <option>{,<option> ...}
```

The options available and applicable switches are described in the individual device descriptions below.

2.2.9 Enabling and Disabling Devices

All devices other than the CPU may be disabled or enabled. Disabling a device simulates removing the associated accessory or interface from the system chassis. To disable or enable a device, use:

Command	*Action*
SET <device> DISABLED	Disable the device
SET <device> ENABLED	Enable the device

For devices that use two interface cards, enabling or disabling either card also enables or disables the other card. For example, entering ***SET DPC DISABLED*** to disable the DPC interface also disables the DPD interface.

Devices that consist of multiple addressable units connected to a controller allow the units to be individually disabled or enabled. Disabling simulates disconnecting the associated unit from the controller. The commands to disable or enable a unit are:

Command	*Action*
SET <unit> DISABLED	Disable the unit
SET <unit> ENABLED	Enable the unit

Each of the above command options is replicated in the option tables of the devices to which they apply.

2.2.10 I/O Interface Select Code Assignments

HP hardware I/O interfaces receive their select code assignments from the CPU backplane slots into which they are plugged. The first I/O slot corresponds to select code 10 octal and has the highest interrupt priority, with priority decreasing with increasing select code number. Empty slots may not exist between interface cards; if all interfaces are not in contiguous slots, HP 12777A Priority Jumper Cards must be installed in the unused slots.

A device's select code may be set with the ***SET <dev> SC=<n>*** command. The valid range of select code values is 10-77 octal. The current select code assignment for a device may be displayed with the ***SHOW <dev> SC*** command. For example:

```
SET MSC SC=26
SHOW PTR SC
```

The current select codes of all devices are displayed with the ***SHOW DEVICES*** command. The default settings are:

Device Name	*Select Code*	*Initially Disabled*	*Device Name*	*Select Code*	*Initially Disabled*
PTR	10		**MSD**	30	
TTY	11		**MSC**	31	
PTP	12		**IPLI**	32	●
TBG	13		**IPLO**	33	●
LPS	14	●	**DS**	34	
LPT	15		**BACI**	35	
MTD	20	●	**MPX**	36	
MTC	21	●	**PIF**	37	
DPD	22		**MUX**	40	
DPC	23		**MUXL**	41	
DQD	24		**MUXC**	42	
DQC	25		**DA**	43	
DRD	26		**MC1**	45	●
DRC	27		**MC2**	46	●

For devices with two interface cards (e.g., DP), specify the lower select code with either device name; the higher select code will be set automatically, and the cards will be assigned in the correct order. If a select code conflict occurs, the simulator will report the error when program execution is attempted. In simulation, empty slots behave as though they contain HP 12777A Priority Jumper Cards, so the device assignments need not be contiguous.

Because they are used infrequently, the LPS, MTD and MTC, IPLI and IPLO, and MC1 and MC2 devices are initially disabled.

The select codes of the CPU, MP, and DMA/DCPC devices are fixed and cannot be changed or displayed. The MEM device does not have a select code, as DMS instructions, and not I/O instructions, are used to control it.

2.2.11 SAVE and RESTORE

SAVE and ***RESTORE*** are supported only when the simulator executable used to restore the simulator state file is the same simulator executable used to save the file. Correctly restoring the state of the simulator depends on the layout of internal structure variables being identical to the layout of the structure variables that were saved. This is guaranteed only when using the same executable, as the layout used is implementation-defined.

2.3 Realistic, Calibrated, and Optimized Timing

Devices simulate their I/O operation delays (disc seeks, magnetic tape reads, printer paper movements, etc.) by counting specified numbers of "event ticks." For example, a disc seek completion might be scheduled to occur 500 event ticks after command initiation. Generally, one event tick is counted for each machine instruction executed, although some complex instructions may count an event tick for each internal step of the instruction (e.g., for each word moved within a block-move instruction). Device simulations provide commands that determine how the appropriate tick counts are selected for I/O operations timing:

Command	*Action*
SET <device> REALTIME	Configure the device to use realistic timing
SET <device> CALTIME	Configure the device to use calibrated timing
SET <device> FASTTIME	Configure the device to use optimized timing

A device configured to use *realistic timing* selects its tick counts to encompass the same number of machine instructions as would be executed in hardware. In real-time mode, an operation taking ten milliseconds in hardware will complete after 15772 machine instructions have executed, based on an average 1000 E-Series hardware instruction execution time of 0.634 microseconds. In this mode, a software program will execute approximately the same number of instructions during a device operation that it would do on a real machine. Host machine speed and execution of concurrent host programs will not affect the number of simulated instructions executed for a given operation.

A device configured to use *calibrated timing* selects its tick counts to align the simulated operation periods with the corresponding time periods on the host system. In calibrated-time mode, an operation taking ten milliseconds in hardware will complete after ten milliseconds has elapsed on the host system. Because the simulator is generally one or two orders of magnitude faster than the hardware, a software program will execute far more code during a device operation than it would do on a real machine. In this mode, the amount of code that executes will vary with the speed of the host machine and the load placed on that machine by other concurrent processes, as the simulator continually adjusts the tick counts up and down to maintain synchronization with the host time.

A device configured to use *optimized timing* selects its tick counts to minimize the operation delays. In fast-time mode, an operation taking ten milliseconds in hardware will complete after the minimum amount of time acceptable to the executing software has elapsed. In this mode, device operations complete far more quickly than they do in hardware. There are limits, however, to how fast I/O operations may occur without causing software malfunctions. In practice, system software often contains assumptions regarding the time certain operations take, so event timing may not be arbitrarily reduced. For instance, an I/O driver may "know" that a line printer takes 50 milliseconds to print a line, and therefore it can ignore interrupts safely for several milliseconds after initiating the print cycle. If the printing time is reduced below that threshold, the driver may fail to operate correctly. The default optimized-timing settings have been empirically determined to work with the supported operating systems listed above, and each device simulator allows the user to modify those settings via registers if needed.

To illustrate how the modes affect timing, consider a simulation of a Teletype terminal that operates at 10 characters per second. If the simulator runs 15 times faster than a real machine, then a user would observe that printing 100 characters takes:

- 10 seconds in CALTIME mode
  (100 characters × *n* event ticks per character adjusted to take exactly 100 mS each on the host system)

- 667 milliseconds in REALTIME mode
  (100 characters × 157,729 event ticks per character × 0.634 µS per tick ÷ 15 times hardware speed)

- 845 microseconds in FASTTIME mode
  (100 characters × 200 event ticks per character × 0.634 µS per tick ÷ 15 times hardware speed)

If the SCP ***SET THROTTLE*** command is used to reduce the speed of the simulator, CALTIME operations will not be affected, but REALTIME and FASTTIME operations will slow proportionally. Reducing simulator speed to that of the original hardware will cause REALTIME operation times to equal CALTIME times.

Devices offer only those modes that are generally useful. For example, the HP 12539C Time Base Generator may be configured to use REALTIME or CALTIME modes. The real-time mode will satisfy the expectations of the TBG diagnostic that checks the timing of operations via delay loops, whereas the calibrated mode will update the DOS, RTE, and TSB time-of-day clocks as expected by users of the simulated system. FASTTIME mode is not offered, as it makes no sense to ignore the programmed time period settings and use a fixed arbitrary period instead.

Devices performing input or output typically offer a REALTIME mode for use when running the diagnostics and a FASTTIME mode for use when running operating systems. In general, software running under simulation will run faster when devices are configured for optimized-time mode, and this is the default for all peripheral devices.

2.4 The Simulation Console and the System Console

When the simulator is started, the SCP command prompt appears at the *simulation console.* For windowed host operating systems, this is typically the same window from which the simulator was started. Once an execution command is entered at the simulation console, the standalone diagnostics and the operating systems begin their user interactions at the *system console*. The TTY device is typically used as the system console.

For convenience and by default, the system console is initially connected to the simulation console, so that SCP and operating system commands may be entered from the same window. Additional "user" terminals may be connected via Telnet or serial ports to the BACI, MPX, or MUX devices as described later.

The system console may be separated from the simulation console by using the ***SET CONSOLE TELNET=<port>*** or ***SET CONSOLE SERIAL=<port>*** command. This leaves the simulation console at the initiating window and moves the system console to a Telnet or serial port, allowing the use of an HP terminal or terminal emulator. Entering the ***SET CONSOLE NOTELNET*** or ***SET CONSOLE NOSERIAL*** command will rejoin the consoles.

2.5 The Keyboard Poll Device

The simulator obtains asynchronous external input, such as system console keystrokes, by polling. The poll interval is selected to provide good responsiveness while minimizing simulator overhead. Polling is established by the Keyboard Poll device, which is scheduled by a real-time clock with a ten-millisecond period. The device's event service routine is entered continuously while the simulator is running. Devices that accept asynchronous input, such as the TTY, BACI, MPX, and MUX devices, co-schedule their event service routines with the poll device. Entering the ***SHOW QUEUE*** command at the SCP prompt will display the poll device and any other devices currently polling for input.

The Keyboard Poll device provides no modifiers or registers, and it cannot be disabled.

2.6 Tracing Simulator Operations

The simulator provides options for extensive tracing of the internal operations of selected devices. This is useful as an aid to hardware and software debugging as well as to gain an understanding of the internal operations of the simulated devices. Devices offer multiple trace reporting levels, from command overviews to detailed backplane signal assertions. Tracing for each device and its separate reporting levels may be enabled independently.

To obtain a trace, two SCP commands must be given. First, a *debug log* must be established with the ***SET DEBUG <target>*** command. This command is described in detail in the "Controlling Debugging" section of the *SIMH Users' Guide* manual. Typically, the target is a text file, so that the trace may be reviewed after capture. Second, tracing must be enabled for the desired devices with ***SET <device> DEBUG=<option>*** commands. These are documented below in the sections that refer to the simulated devices. The formats of the trace output are specific to the devices being traced. Examples are provided in each device description section below.

The reporting level options table given for each device is arranged in order of increasing detail. The first option listed provides the broadest overview with the least specific detail and generates the smallest number of trace lines, thereby slowing program execution the least. Subsequent options provide increasing detail at the expense of larger debug log files. Enabling all trace options with a ***SET <device> DEBUG*** command provides the fullest picture of device operation but may generate very large log files.

Some options enable tracing of periodic events, e.g., a clock tick or a device poll. Use caution when specifying these options, as the trace log may fill rapidly. Options that trace periodic behavior are noted in the option descriptions.

Tracing does impose some overhead on the simulator, with more detailed tracing slowing the simulator more than higher-level tracing. No overhead is incurred when tracing is suspended with the ***SET NODEBUG*** command, even if individual device tracing options remain in effect.

3 Processor Device Simulations

An HP-family computer consists of the following subsystems:

- a 2114, 2115, 2116, 2100, or 1000 M/E/F-Series Central Processing Unit
- an optional 12578A/12607B/12895A DMA Controller or 12897B DCPC Controller
- an optional 12581A/12892B Memory Protect
- an optional 12731A Memory Expansion Module

3.1 Central Processing Unit

The HP computer Central Processing Unit contains the machine-instruction execution unit and main memory. CPU options specify the memory size, installed firmware, and simulation configuration. The CPU is configured with commands of the form:

```
SET {-F} CPU <option>
```

Device options that may be specified are:

Option	*Action*
2114	Use a 2114 computer
2115	Use a 2115 computer
2116	Use a 2116 computer; default
2100	Use a 2100 computer
1000-M	Use a 1000 M-Series computer
1000-E	Use a 1000 E-Series computer
1000-F	Use a 1000 F-Series computer
21MX-M	Use a 21MX M-Series computer (same as a 1000 M-Series)
21MX-E	Use a 21MX E-Series computer (same as a 1000 E-Series)
8K	Set the memory size to 8K words
12K	Set the memory size to 12K words
16K	Set the memory size to 16K words
24K	Set the memory size to 24K words
32K	Set the memory size to 32K words; default
64K	Set the memory size to 64K words
128K	Set the memory size to 128K words
256K	Set the memory size to 256K words
512K	Set the memory size to 512K words
1024K	Set the memory size to 1024K words
EAU	Enable the Extended Arithmetic Unit instructions
NOEAU	Disable the Extended Arithmetic Unit instructions; default
FP	Enable the Floating Point instructions
NOFP	Disable the Floating Point instructions; default
IOP	Enable the 2000/Access I/O Processor instructions
NOIOP	Disable the 2000/Access I/O Processor; default
DMS	Enable the Dynamic Mapping System instructions

Option	*Action*
NODMS	Disable the Dynamic Mapping System instructions; default
FFP	Enable the Fast FORTRAN Processor instructions
NOFFP	Disable Fast FORTRAN Processor instructions; default
DBI	Enable the Double Integer Instructions
NODBI	Disable the Double Integer Instructions; default
EMA	Enable the Extended Memory Array instructions
NOEMA	Disable the Extended Memory Array instructions; default
VMAOS	Enable the Virtual Memory and Operating System instructions
NOVMAOS	Disable the Virtual Memory and Operating System instructions; default
VIS	Enable the Vector Instruction Set
NOVIS	Disable the Vector Instruction Set; default
SIGNAL	Enable the SIGNAL/1000 instructions
NOSIGNAL	Disable the SIGNAL/1000 Instructions; default
LOADERENABLE	Enable access to the protected binary loader
LOADERDISABLE	Disable access to the protected binary loader; default
ROMS=<rom-list>	Change the set of installed boot loader ROMs
IDLE	Enable idle detection
NOIDLE	Disable idle detection; default
STOP=<option>[;<option]	Enable simulation stops
NOSTOP=<option>[;<option]	Disable simulation stops; default
INDIR=<limit>	Set the maximum indirect addressing chain length; default is 16
EXEC=<match>[;<mask>]	Enable execution tracing of matching instructions
NOEXEC	Disable execution tracing; default
DEBUG=<option>	Enable tracing
NODEBUG	Disable tracing; default

The abbreviated ***SET CPU 1000*** and ***SET CPU 21MX*** commands configure the CPU models as 1000 E-Series and 1000 M-Series machines, respectively.

If the memory size is being reduced, and the memory being truncated contains non-zero data, the simulator asks for confirmation before proceeding. The confirmation request may be suppressed by using the ***–F*** (force) switch. Data in the truncated portion of memory is lost.

The following standard microcode is automatically enabled when the applicable 1000-series CPU is selected:

Instruction Set	*Applicable to*
Extended Instruction Group	1000 M/E/F-Series
Floating-Point Processor	1000 F-Series
Scientific Instruction Set	1000 F-Series

The 1000 E/F-Series microcode self-test diagnostics are not simulated.

The ***EAU*** option simulates the installation of the 12579A Extended Arithmetic Unit providing hardware multiply, divide, double load, double store, and 32-bit rotate and shift instructions. EAU instructions are standard equipment on the 2100 and 1000 CPUs.

The ***FP*** option simulates the installation of the 12901A Floating Point firmware ROMs, providing single-precision (two-word) floating-point add, subtract, multiply, divide, fix, and float instructions. FP instructions are standard equipment on 1000 CPUs.

The ***IOP*** option simulates the installation of the 13206A (2100), 13207A (1000 M-Series), or 22702A (1000 E-Series) 2000/Access I/O Processor firmware ROMs. These instructions accelerate certain common operations of the I/O processor for the HP 2000/Access Time-Shared BASIC operating system.

The ***DMS*** option simulates the installation of the 12731A Memory Expansion Module and the 12976B (1000 M-Series) or 13307B (1000 E/F-Series) Dynamic Mapping System firmware ROMs, providing 38 instructions that control the MEM to expand the basic 32K-word logical address space to a 1024K-word physical space. The MEM is automatically enabled or disabled with the DMS instruction set.

The ***FFP*** option simulates the installation of the 12907A (2100), 12977B (1000 M-Series), or 13306B (1000 E-Series) Fast FORTRAN Processor firmware ROMs, providing several frequently used FORTRAN operations and extended-precision (three-word) floating-point instructions.

The ***DBI*** option simulates the installation of the 93585A Double Integer Instructions firmware ROMs. A product of the HP "specials group," the microcode added twelve 32-bit integer instructions to the 1000 E-Series repertoire. DBI instructions are standard equipment on the 1000 F-Series CPU.

The ***EMA*** option simulates the installation of the 92067A Extended Memory Area firmware ROMs. The RTE-IV operating system introduced the EMA instructions. EMA provided a mapped data area up to one megaword in size. These three instructions accelerated data accesses to variables stored in EMA partitions.

The ***VMAOS*** option simulates the installation of the 92084A Virtual Memory and Operating System firmware ROMs. The RTE-6/VM operating system introduced Virtual Memory Area (VMA) instructions — a superset of the RTE-IV EMA instructions — and a set of OS instructions that accelerate certain time-consuming internal operations.

The VMA/OS firmware cannot be installed on a 1000 M-Series; RTE-6/VM will automatically use software replacements on this machine. The firmware must be installed when running on 1000 E/F-Series hardware; if it is not, the firmware self-test instruction will fail, and RTE will halt with T = 102021. Under simulation, however, it is possible to configure the self-test instruction to use the software simulations on an E/F-Series. Entering the command ***SET CPU DEBUG=NOOS*** will return a firmware revision code of 0 to RTE, which will permit the use of the software replacements.

The ***VIS*** option simulates the installation of the 12824A (for RTE-IVB) or 12829A (for RTE-6/VM) Vector Instruction Set ROMs, providing vector arithmetic instructions using single- and double-precision values. Vectors may reside either in EMA/VMA or in regular memory.

The ***SIGNAL*** option simulates the installation of the 92835A SIGNAL/1000 firmware ROMs. These instructions provide fast Fourier transforms and complex arithmetic. They use the F-Series floating-point processor and the Vector Instruction Set.

The 2100 and 1000 CPUs support user microprogramming. Under simulation, execution on these machines of all instructions in the octal ranges 101400-101777 (2100 and 1000) and 105000-105777 (1000 only) that are not allocated to installed firmware options will be dispatched to a user-alterable module to aid in the implementation of user-written microcode simulations. In the absence of such simulations, execution will cause unimplemented instruction stops. See the comments in the *hp2100_cpu0.c* source file for details.

The 2114, 2115, 2116, and 2100 models support a protected area of memory containing an initial binary loader. The loader always resides in the highest 64 memory locations, regardless of capacity. Entering the ***SET CPU LOADERENABLE*** command makes this area available. Entering the ***SET CPU LOADERDISABLE*** command renders this area non-existent, so that reads from the area return zero, and writes to the area are ignored. The Basic Binary Loader (BBL), configured for the select code of the paper tape reader, initially resides in this area when the simulator is started. The ***LOAD*** command may be used with an appropriate device name or absolute binary file to install a different loader, such as the Basic Binary Disc Loader (BBDL) or Basic Moving-head Disc Loader (BMDL).

The ***SET CPU ROMS*** command is used to change the set of installed boot loader ROMs on the 1000 M/E/F-Series of machines. It is described in the *Bootstrap Loaders* section below.

When enabled by a ***SET CPU IDLE*** command, execution of the idle loop instructions within the DOS and RTE operating systems will idle the simulator. While idle, the simulator does not use any host system processor time. Idle detection is operating-system specific and is disabled by default. When disabled, the simulator will use 100% of the host-processor CPU time while executing simulated instructions.

Instruction execution trace behavior changes when idling is enabled. See the *Tracing* section below for details.

The ***SET CPU STOP*** command enables one or more simulation stop conditions, ***SET CPU NOSTOP*** disables the specified conditions, and ***SET CPU INDIR*** sets the maximum number of indirection levels permitted. These commands are described in the *Simulation Stops* section below.

The ***SET CPU EXEC*** command configures instruction execution tracing. This command is described in the *Tracing* section below.

The CPU models and feature applicability are shown in the following table:

Feature	***2114***	***2115***	***2116***	***2100***	***1000 M***	***1000 E***	***1000 F***
Maximum Memory	16K	8K	32K	32K	1024K	1024K	1024K
Memory Expansion	×	×	×	×	O	O	O
Memory Protect	×	×	O	✓	O	O	O
Direct Memory Access	O	O	O	O	O	O	O
EAU	×	O	O	✓	✓	✓	✓
FP	×	×	×	O	✓	✓	✓
IOP	×	×	×	O	O	O	×
FFP	×	×	×	O	O	O	✓
EIG	×	×	×	×	✓	✓	✓
DMS	×	×	×	×	O	O	O
DBI	×	×	×	×	×	O	✓
EMA	×	×	×	×	×	O	O
VMA/OS	×	×	×	×	×	O	O
FPP	×	×	×	×	×	×	✓
SIS	×	×	×	×	×	×	✓
VIS	×	×	×	×	×	×	O
SIGNAL	×	×	×	×	×	×	O

Key: × = unavailable O = optional ✓ = standard

If a feature is standard for a given CPU, it cannot be disabled; if a feature is unavailable, it cannot be enabled.

On the 2100, the FP or FFP option and the IOP option are mutually exclusive. On the 1000 E/F-Series, the RTE-IV EMA and RTE-6/VM VMA and OS options are mutually exclusive.

Setting the CPU model establishes a consistent set of standard features and common options. Additional ***SET CPU*** commands may follow to fine-tune the desired feature set.

The CPU configuration may be displayed with the following commands:

Command	*Action*
SHOW CPU	Display the device configuration
SHOW CPU ROMS	Display the set of installed boot loader ROMs
SHOW CPU IOCAGE	Display the set of installed I/O interfaces
SHOW CPU STOPS	Display the enabled simulation stops
SHOW CPU INDIR	Display the indirect chain length limit
SHOW CPU EXEC	Display the matching criteria for execution tracing
SHOW CPU SPEED	Display the current simulation speed

The current simulation speed, expressed as a multiple of the speed of a real HP 1000 E-Series, may be obtained with the ***SHOW CPU SPEED*** command. The speed reported will not be representative if the simulator was idling when it was stopped.

3.1.1 Bootstrap Loaders

When the CPU is configured as a 2114, 2115, 2116, or 2100, the Basic Binary Loader is initially installed in the highest 64 words of core memory. The BBL reads an absolute binary program from the paper tape reader into memory. Alternate loaders for other devices may be installed by entering ***LOAD*** commands as described above.

To execute the BBL, the following sequence of commands is used:

```
ATTACH PTR <tape-image-filename>
SET CPU LOADERENABLE
DEPOSIT S <value>
DEPOSIT P <address>
RESET
GO
```

...where *address* is the appropriate loader starting address — 07700 for 4K, 17700 for 8K, 27700 for 12K, 37700 for 16K, 57700 for 24K, or 77700 for 32K. The S-register value is set to 000000 to load the paper tape, 000001 to verify the paper tape, or 100000 to compare the paper tape to memory. Loader execution ends with one of the following halt instructions:

- HLT 00 — the tape does not compare with memory; A = the tape value that did not compare.
- HLT 11 — a checksum error occurred; A = the tape value, B = the calculated value.
- HLT 55 — the load address would overlay the BBL.
- HLT 77 — the end of tape was reached with a successful read.

The loader is automatically disabled when the CPU executes the halt instruction within the loader.

1000-series CPUs contain up to four Initial Binary Loaders stored in ROM. In hardware, a loader is selected by setting S register bits 15-14 to the ROM number, bits 13-12 and 5-0 to the ROM-specific options, and bits 11-6 to the device select code. Then the IBL, PRESET, and RUN front-panel buttons are pressed in sequence to copy the selected ROM into the highest 64 words of memory, initialize the I/O device interfaces, and begin execution.

In simulation, the ***LOAD CPU*** command implements the IBL facility. The following set of ROMs is initially installed in the CPU:

ROM Number	*Product Number*	*Boot Device Description*	*Boot Device*
00	12992K	2748B paper tape reader	PTR
01	12992A	7900 or 2883 disc	DPC or DQC
10	12992D	7970B or 7970E magnetic tape	MSC
11	12992B	7905, 7906, 7920, or 7925 disc	DS

To copy and execute a loader, the following sequence of commands is used:

```
ATTACH <dev> <image-filename>
DEPOSIT S <value>
LOAD CPU
RESET
GO
```

A successful copy will set the O (overflow) register to 0 and the P register to the starting address of the loader. If bits 15-14 of the S register specify an empty ROM socket, the command will be rejected with a **Non-existent device** error. If the select code specified in the S register is invalid (less than 10 octal), the O register will be set to 1, and the command will be rejected with an **Invalid argument** error. The P register and memory will not be altered in either case. No error is reported if the select code does not reference the correct device. The ***BOOT CPU*** command may be used in place of the ***LOAD CPU*-*RESET*-*GO*** sequence. The specific S-register configurations required, supported image file types, and the associated loader halt codes are listed under their respective devices below.

The set of boot loaders installed in a 1000-series CPU may be changed with the following command:

```
SET CPU ROMS=<rom-list>
```

The ROM list may be omitted or contain from one to four device names, separated by semicolons, that specify the corresponding loaders to install in the ROM sockets. All four ROM sockets are altered for each command. If no devices are specified, then all sockets are emptied. Otherwise, specifying a valid device name installs the device loader ROM into the socket corresponding to the position of the device name in the list. Sockets may be left empty by omitting the corresponding device name or by supplying fewer than four device names. For example:

Command	*Action*
SET CPU ROMS=	Remove ROMs from sockets 0-3
SET CPU ROMS=PTR	Install the PTR ROM in socket 0; leave sockets 1-3 empty
SET CPU ROMS=DS;MS	Install the DS ROM in 0 and the MS ROM in 1; leave 2 and 3 empty
SET CPU ROMS=;;DPC	Install the DPC ROM in 2; leave 0, 1, and 3 empty
SET CPU ROMS=DQC;;;DA	Install the DQC ROM in 0 and the DA ROM in 3; leave 1 and 2 empty

The current CPU model must be a 1000-series machine, and each device must be bootable and have a loader ROM assigned, or the command will be rejected. A rejected command does not alter any of the ROM assignments.

The ***SHOW CPU ROMS*** command displays the set of loader ROMs currently installed. For example:

```
Socket  Device  ROM Description
------  ------  ----------------------------------------
  0      PTR    12992K Paper Tape Loader
  1      DQC    12992A 7900/7901/2883 Disc Loader
  2      MSC    12992D 7970 Magnetic Tape Loader
  3      DS     12992B 7905/7906/7920/7925 Disc Loader
```

The ***SHOW CPU IOCAGE*** command displays the set of interfaces currently installed in the I/O card cage. For example:

```
SC  Device  Interface Description
--  ------  ------------------------------------------------------------
10   PTR    12597A-002 Tape Reader Interface
11   TBG    12539C Time Base Generator Interface
12   DS     13175D Disc Controller Interface
13   MSD    13181B/13183B Digital Magnetic Tape Unit Interface Data Channel
14   MSC    13181B/13183B Digital Magnetic Tape Unit Interface Command Channel
15  (none)  12777A Priority Jumper Card
16   TTY    12531C Buffered Teleprinter Interface
17   PTP    12597A-005 Tape Punch Interface
20   LPT    12845B Line Printer Interface
```

The devices appear in select code order, beginning with select code 10 and ending with the last occupied select code. Empty card cage slots between devices behave as though they contain HP 12777A Priority Jumper Cards as shown above. Multiple devices assigned to the same select code are reported as a conflict. For instance:

```
sim> SET PTP SC=12
sim> SHOW CPU IOCAGE

Select code 12 conflict (PTP and DS)

SC  Device  Interface Description
--  ------  ------------------------------------------------------------
10   PTR    12597A-002 Tape Reader Interface
11   TBG    12539C Time Base Generator Interface
12   ---    (multiple assignments)
13   MSD    13181B/13183B Digital Magnetic Tape Unit Interface Data Channel
14   MSC    13181B/13183B Digital Magnetic Tape Unit Interface Command Channel
15  (none)  12777A Priority Jumper Card
16   TTY    12531C Buffered Teleprinter Interface
17  (none)  12777A Priority Jumper Card
20   LPT    12845B Line Printer Interface
```

The simulator will not begin program execution until the conflict is resolved.

3.1.2 Simulation Stops

The simulator can be configured to detect certain machine instruction conditions and stop execution when one of them occurs. Stops may be enabled or disabled individually with these commands:

```
SET CPU STOP=<option>[;<option]
SET CPU NOSTOP=<option>[;<option]
```

The ***SET CPU STOP*** and ***SET CPU NOSTOP*** commands enable or disable all of the configurable stops.

These options control the simulation stop conditions:

Option	*Stop Condition*
UNIMPL	Stop when an unimplemented instruction is executed
UNDEF	Stop when an undefined instruction is executed
UNSC	Stop when an unused select code is referenced
IOERR	Stop when an unreported I/O error occurs

Setting the ***UNIMPL*** option stops the simulator if execution of an unimplemented instruction is attempted. An instruction is unimplemented if it is provided by a firmware option that is not currently installed (e.g., a DAD instruction when the double-integer firmware is not installed) or is an opcode provided by an installed option but not assigned to an instruction (e.g., opcode 105335 from the double-integer firmware set). If the stop is not set, unimplemented instructions execute as NOP (no-operation), as in hardware.

Setting the ***UNDEF*** option stops the simulator if execution is attempted of an instruction containing a decoded reserved bit pattern other than the one defined in the Operating and Reference Manual for the CPU. For example, opcodes 101700 and 105700 are not defined as DMS instructions, but the microcode executes them as XMM instructions, rather than as NOPs. The EAU opcode 100060 is defined as the 1000 E/F-Series TIMER instruction; it is undefined on other machines, but it executes as MPY on the M-Series and NOP on the 2100. The intent of this stop is to catch instructions containing reserved fields with values that change the meaning of those instructions. If the stop is not set, undefined instructions execute in the same manner as on the selected CPU hardware.

Certain programs, such as the HP diagnostic configurator, intentionally execute unimplemented instructions to determine the CPU model and configuration. For example, the configurator executes an RRR 16 instruction to detect the presence of the EAU. If the EAU is installed, the A- and B-register contents are exchanged; otherwise, the instruction executes as NOP, and the contents are unaltered. Also, the 1000 M-Series is differentiated from the E/F-Series by executing the TIMER instruction; on the M-Series, it decodes as MPY and so returns to P+2, while TIMER returns to P+1 on the E/F-Series. Enabling the ***UNIMPL*** and ***UNDEF*** options prevent this feature detection code from working as expected. Disabling or bypassing these stops is required to permit correct operation.

Setting the ***UNSC*** option stops the simulator if an I/O instruction references the select code of an I/O backplane slot that does not contain an interface card. If this option is not set, referencing an empty slot results in a no-operation, as in hardware.

Setting the ***IOERR*** option stops the simulator if an I/O error occurs on a device that does not return error status to the CPU. For example, the paper tape punch returns low- or out-of-tape status, the line printer returns out-of-paper and offline status, and the disc drive returns not-ready and drive-fault status. By contrast, the paper tape reader gives no indication that a tape is loaded. If the reader is commanded to read when no tape is mounted, the interface hangs while waiting for the handshake with the device to complete. In hardware, the CPU can detect this condition only by timing the operation and concluding that the tape is missing if the timeout is exceeded. However, if a ***SET CPU STOP=IOERR*** command has been issued, then the simulator will stop with an error message to permit the condition to be fixed. For instance, attempting to read from the paper tape reader with no paper tape image file attached will print **No tape loaded in the PTR device** and will stop the simulator. After issuing an ***ATTACH PTR*** command to load the tape into the reader, resuming simulation will retry the read. Devices that respond to the ***IOERR*** stop are indicated in their respective sections below.

After a simulation stop, execution may be resumed in one of two ways. If the cause of the stop has not been remedied and the stop option has not been disabled, resuming execution with ***CONTINUE***, ***STEP***, ***GO***, or ***RUN*** will cause the stop to occur again. Alternately, specifying the ***–B*** switch with any of the preceding commands will resume execution while bypassing the stop for the current instruction. Bypassing a stop has the following effect:

Option	*Bypass Action*
UNIMPL	Resume execution with the instruction ignored
UNDEF	Resume execution with the instruction decoded as in hardware
UNSC	Resume execution with the I/O instruction ignored
IOERR	Resume execution with the action as in hardware

In addition to these configurable stops, an unconditional simulation stop will occur if an indirect addressing chain exceeds the maximum length specified by a ***SET CPU INDIR=<limit>*** command. HP computer family memory addresses may be indirect to indicate that the values point to the target addresses rather than contain the target addresses. The target of an indirect address may itself be indirect, and the CPU follows this chain of addresses until it finds a direct address. Indirect addressing is typically only one or two levels deep, but if the chain loops back on itself (e.g., if an indirect address points to itself), then instruction execution will be stopped. For example, these commands:

```
DEPOSIT 0 LDA 1,I
DEPOSIT 1 100001
RUN 0
```

...will stop the simulator with an **Indirect address loop** message.

The limit may be set to any number of levels up to 32,768. This is the absolute maximum number of levels that can be created without an infinite loop — each location in memory points to the next one except for the last, which contains the target value. In practice, anything over a few levels likely represents a programming error. The default setting is 16 levels.

3.1.3 Stop Messages

When the CPU executes a HLT instruction, simulated execution terminates, and control returns to the SCP prompt. The contents of the T and P registers, and the halt instruction and the next instruction to execute are displayed. For example:

```
Programmed halt, T: 102023 (HLT 23), P: 00101 (CLA,INA)
sim>
```

When the simulator stops for a user request or after reaching a breakpoint or completing a ***STEP*** command, the P-register value and the next instruction to execute are displayed. For example:

```
Simulation stopped, P: 03306 (JMP 3311)
sim>
```

If an interrupt is pending, however, the instruction contained in the trap cell corresponding to the highest-priority interrupting device will be displayed instead:

```
Simulation stopped, P: 03306 (IAK 11: JSB 1644,I)
sim>
```

This indicates that when simulation resumes, the device with select code 11 will receive the interrupt acknowledgement, and JSB 1644,I will be executed. Note that P does not change when an interrupt occurs and a trap cell instruction is executed.

3.1.4 Tracing

When a debug log has been established, tracing may be configured by specifying one or more of the reporting level options:

Option	*Reporting Level*
INSTR	Machine instructions executed
DATA	Memory data accesses
FETCH	Memory instruction fetches
REG	Register values
OPND	Memory operand values
EXEC	Matching instruction execution states

A section of an example trace is:

```
>>CPU instr: S 0002 05756  102705  STC 5
>>CPU fetch: S 0002 05757  105736    instruction fetch
>>CPU   reg: P **** 01011  042200    A 177777, B 177777, X 177777, Y 000000, E O I
>>CPU instr: S 0002 05757  105736  UJP 2111
>>CPU fetch: S 0002 05760  002111    instruction fetch
>>CPU fetch: U 0001 02111  026111    instruction fetch
>>CPU   reg: P **** 01011  042200    A 177777, B 177777, X 177777, Y 000000, E O I
>>CPU instr: U 0001 02111  026111  JMP 2111
>>CPU instr: U 0001 02111  000011  interrupt
>>CPU fetch: S 0000 00011  115013    instruction fetch
>>CPU   reg: - **** 01011  042200    A 177777, B 177777, X 177777, Y 000000, E O I
>>CPU   reg: - **** *****  ******    MPF 000000, MPV 002111, MES 163011, MEV 030000
>>CPU instr: S 0000 00011  115013  JSB 1013,I
>>CPU  data: S 0000 01013  005557    data read
>>CPU  data: S 0002 05557  002111    data write
>>CPU fetch: S 0002 05560  103100    instruction fetch
>>CPU   reg: - **** 01011  042200    A 177777, B 177777, X 177777, Y 000000, E O I
```

The ***INSTR*** option traces instruction executions. Each instruction is printed before it is executed.

The ***DATA*** option traces reads from and writes to memory. Each is classified by the type of access — data or unprotected. Normal data accesses are translated by the current map if DMS is enabled and are subject to memory and DMS page protections. Unprotected accesses are translated by the specified map if DMS is enabled and have unrestricted access to memory. Certain firmware extension instructions make unprotected accesses through the system or user maps.

The ***FETCH*** option traces instruction fetches from memory. Interrupt trap cell instructions and the additional words of multiword instructions are included in this classification.

The ***REG*** option traces register values. Two sets of registers may be printed. After executing each instruction, the accumulators, index registers, extend and overflow registers, and the state of the interrupt system are printed. After executing an instruction that may alter the memory-protect or memory-expansion registers, the fence, violation, and status registers are printed.

The ***OPND*** option traces instruction operand values. Some instructions take memory and register operands that are difficult to decode from ***DATA*** or ***REG*** traces. This option presents these operands in a higher-level format. The memory bank and address values are always those of the operands. The operand data and values printed are specific to the instruction.

The ***EXEC*** option traces the execution of instructions that match user-specified criteria. When a match occurs, all CPU trace options are turned on for the duration of the execution of the matched instruction. The prior trace settings are restored when a match fails. This option allows detailed tracing of specified instructions while minimizing the log file size compared to a full instruction trace.

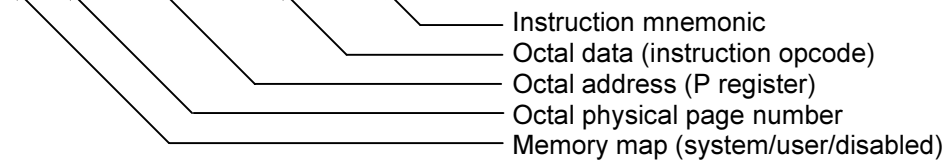
The ***SET CPU EXEC*** command configures the match and mask values used to qualify instructions for execution tracing. Qualification is performed by ANDing the current instruction with the specified *mask* value and then comparing the result with the specified *match* value. If the *mask* value is omitted, the *match* value must match the instruction exactly. The values are entered in the CPU's data radix, which defaults to octal but may be set to a different radix or overridden by a switch on the command line.

Setting the *mask* value allows matching a range of instructions or an instruction with a range of operand values. For example, ***SET CPU EXEC=105240;177760*** will trace execution of all EMA or VMA firmware instructions, ***SET CPU EXEC=102700;177700*** will trace execution of STC instructions, and ***SET CPU EXEC=105040*** will trace execution of FMP instructions.

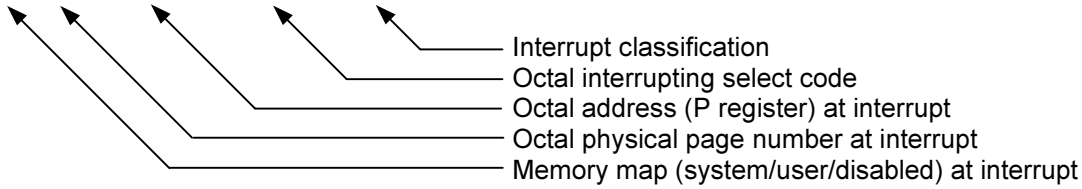
If idling has been enabled with the ***SET CPU IDLE*** command, tracing will be suspended when the simulator detects an entry into the operating system idle loop. This is done to avoid printing the hundreds or thousands of loop instructions that are executed while awaiting an interrupt. Tracing is automatically resumed when the idle loop is exited either by an interrupt or a simulation stop.

The trace formats are interpreted as follows:

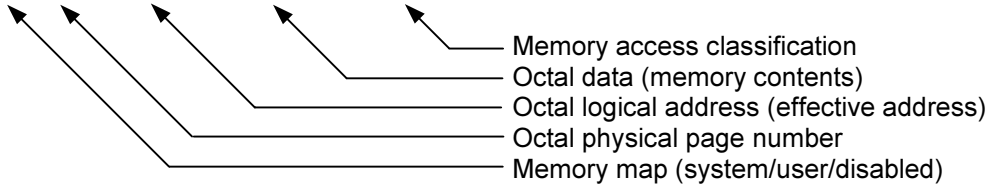
```
>>CPU instr: S 0002 05757  105736  UJP 2111
```



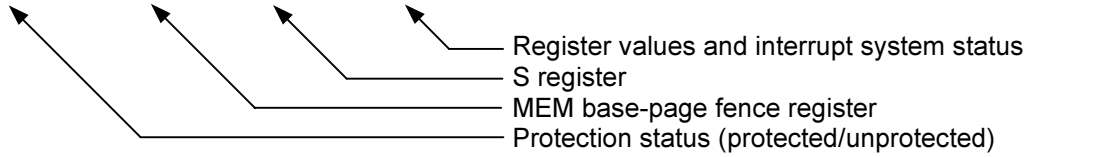

```
>>CPU instr: U 0001 02111  000011  interrupt
```



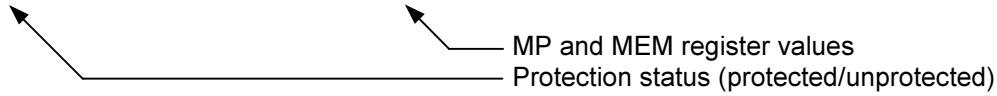

```
>>CPU  data: - 0000 01013  005557    data read
>>CPU  data: U 0000 01776  000104    unprotected write
>>CPU fetch: S 0002 05757  105736    instruction fetch
```



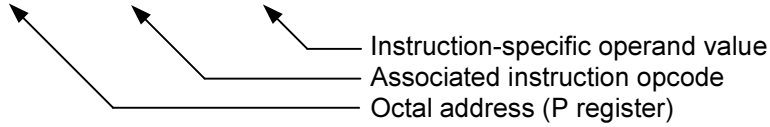

```
>>CPU   reg: P **** 01011  042200    A 177777, B 177777, X 177777, Y 000000, e O I
```



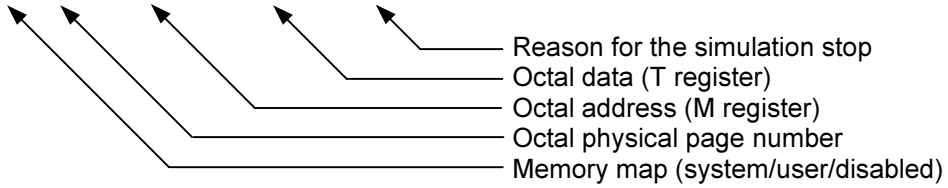

```
>>CPU   reg: - **** *****  ******    MPF 000000, MPV 002111, MES 163011, MEV 030000
```




```
>>CPU  opnd: * **** 20056  105251    virtual address 1400006
>>CPU  opnd: * **** 20056  105251    dimension 1 element count 1024
>>CPU  opnd: * **** 20056  105251    element size 2 offset 00000000000
```




```
>>CPU instr: - 0001 02200  102077  simulation stop: Programmed halt
```



Enabling CPU tracing can produce a very large number of lines very quickly, so care should be used to enable tracing only around the area of interest. Breakpoint actions may be used to implement this; for example:

```
BREAK 100; SET CPU DEBUG; GO
BREAK 200; SET CPU NODEBUG; GO
```

These commands will enable tracing when the program counter reaches location 100 and disable tracing when it reaches location 200, thereby producing a trace of instructions executed between locations 100 and 200. Alternately, if the execution of specific instructions is of interest, the ***EXEC*** trace option may be used to reduce the debug log file size.

3.1.5 Registers

The CPU state contains the registers visible to the programmer and the interrupt and dynamic mapping system control registers:

Name	*Size*	*Radix*	*Symbolic*	*Read-Only*	*Description*
P	15	8			Program Counter
A	16	8	✓		Accumulator A
B	16	8	✓		Accumulator B
M	15	8			Memory Address Register
T	16	8	✓	✓	Memory Data Register
X	16	8	✓		Index Register X
Y	16	8	✓		Index Register Y
S	16	8	✓		Display Register
E	1	2			Extend Register
O	1	2			Overflow Register
CIR	6	8			Central Interrupt Register
INTSYS	1	2			Interrupt System Enabled
INTEN	1	2			Interrupt Recognition Enabled
IOPSP	16	8			I/O Processor Stack Pointer
PCQ [0:63]	15	8		✓	Program Counter Queue

The A, B, T, X, Y, and S registers may be examined or deposited using any of the modes described in the *Symbolic Display and Entry* section above.

3.2 12578A/12607B/12895A Direct Memory Access Controllers, 12897B Dual-Channel Port Controller

Direct memory access controllers permit the computer to transfer data directly between an I/O device and memory on a cycle-stealing basis. Depending on the CPU, the device interface, and main memory speed, DMA is capable of transferring data blocks from 1 to 32,768 words in length at rates between 500,000 and 1,000,000 words per second.

The HP 12578A DMA controller provides two channels for the 2115 and 2116 computers. The channels support configurable byte packing and unpacking between memory and the I/O interface; blocks are limited to 16,384 words in length. The HP 12607B DMA controller provides a single channel for the 2114 computer. Block length is limited to 8,192 words, and byte packing and unpacking are not supported. The 12895A DMA controller provides two channels for the 2100 computer, and the HP 12897B Dual-Channel Port Controller provides two channels for the 1000 M/E/F-Series computers. Neither provides byte packing or unpacking, and blocks up to 32,768 words may be

transferred. The channels are permanently assigned select codes 2 and 6 for channel 1, and select codes 3 and 7 for channel 2.

The device names are ***DMA1*** for the 12607B, ***DMA1*** and ***DMA2*** for the 12578A and 12895A, and ***DCPC1*** and ***DCPC2*** for the 12897B. Device options that may be specified are:

Option	*Action*
DEBUG=<option>	Enable tracing
NODEBUG	Disable tracing; default
ENABLED	Enable the device; default
DISABLED	Disable the device

When a debug log has been established, tracing may be configured by specifying one or more of the reporting level options:

Option	*Reporting Level*
CMD	Channel commands
CSRW	Control, status, read, and write actions
SR	Service requests received
DATA	Direct memory data accesses
IOBUS	I/O bus signals and data words received and returned

The ***CMD*** option traces the initiation and completion of channel transfers. The ***CSRW*** option traces control, status, read, and write commands sent to the channel. The ***SR*** option traces service requests received from the device. The ***DATA*** option traces memory accesses performed by the port controller. The ***IOBUS*** option traces the I/O backplane signals and data received and returned via the controller.

The trace formats are interpreted as follows:

```
>>DCPC1   cmd: Channel transfer completed
>>DCPC1  csrw: Control word 1 is STC | CLC | select code 23
>>DCPC1    sr: Select code 23 asserted SRQ
>>DCPC1 iobus: Received data 000000 with signals STC | CLF | SIR | PRH
```

Operational message

```
>>DCPC1 data: B 0002 05757  105736    dma write
```

Memory access classification
Octal data (memory contents)
Octal logical address (effective address)
Octal physical page number
Memory map (port A/port B/disabled)

Each DMA/DCPC channel state contains these registers:

Name	*Size*	*Radix*	*Symbolic*	*Description*
XFR	6	8		Channel is active if > 0
CTL	1	2		Channel interrupt is enabled
FLG	1	2		Channel is ready
FBF	1	2		Channel is ready (buffer)
CTL2	1	2		Command word 2/3 selector
CW1	16	8		Command word 1
CW2	16	8		Command word 2
CW3	16	8		Command word 3
BYTE	1	2		Byte-packing register is occupied
PACKER	8	8	✓	Byte-packing register

The BYTE and PACKER registers are applicable only to the 12578A controller. The PACKER register defaults to single-character format display and entry but may be overridden with a numeric-format switch, if desired.

3.3 12581A/12892B Memory Protect

Memory protection is standard equipment on the 2100 (although it may be disabled by removing a jumper) and optional on the 2116 and 1000. It provides a Fence Register whose setting divides the logical address space of the CPU into protected and unprotected parts. When enabled, memory locations below the fence cannot be altered, nor can execution jump to addresses below the fence. Violations cause the offending instruction to be aborted, the address of the instruction to be recorded in the Violation Register, and an interrupt to select code 5. The 12892B Memory Protect option for the 1000 has three feature options that are implemented by jumper settings. The default configuration provides compatibility with 12581A Memory Protect option for the 2116 and the standard memory protection for the 2100.

Device options that may be specified are:

Option	*Action*
JSBIN	Install jumper W5; default
JSBOUT	Remove jumper W5
INTIN	Install jumper W6; default
INTOUT	Remove jumper W6
SEL1IN	Install jumper W7
SEL1OUT	Remove jumper W7; default
ENABLED	Enable the device; default
DISABLED	Disable the device

W5 determines whether JSB instructions referencing memory locations 0 and 1 are legal (installed) or illegal (removed). W6 controls whether the first three levels of indirect addressing hold off (installed) or permit (removed) pending interrupts. W7 determines whether I/O instructions referencing select codes other than 1 are legal (installed) or illegal (removed). I/O instructions referencing select code 1 are legal, and HLT instructions are illegal, regardless of the setting of W7.

Memory Protect works in conjunction with the Memory Expansion Module to abort programs that cause MEM violations. If Memory Protect is not installed (i.e., is disabled), then MEM violations are recorded in the Violation Register, but the offending program is allowed to continue.

When a debug log has been established, tracing may be configured by specifying the reporting level option:

Option	*Reporting Level*
IOBUS	I/O bus signals and data words received and returned

The ***IOBUS*** option traces the I/O backplane signals and data received and returned by the device.

Examples of the trace formats follow:

```
>>MP iobus: Received data 000000 with signals ENF | SIR | IEN | PRH
>>MP iobus: Returned data 000000 with signals FLG | IRQ
>>MP iobus: Received data 000000 with signals IAK | SIR | IEN | PRH
>>MP iobus: Returned data 000000 with signals PRL
>>MP iobus: Received data 000000 with signals IOI
>>MP iobus: Returned data 024601 with signals (none)
```

The Memory Protect state contains these registers:

Name	*Size*	*Radix*	*Description*
CTL	1	2	Control flip-flop
FLG	1	2	Flag flip-flop
FBF	1	2	Flag buffer flip-flop
FR	15	8	Fence register
VR	16	8	Violation register
EVR	1	2	Enable violation register flip-flop
MEV	1	2	Memory expansion violation flip-flop

3.4 12731A Memory Expansion Module

The HP 12731A Memory Expansion Module is an option for the HP 1000 M/E/F-Series computers. The MEM provides mapping of the 32 1024-word logical memory pages into a one-megaword physical memory. Four separate maps are provided: system, user, DCPC port A, and DCPC port B. The MEM is controlled by the associated Dynamic Mapping System instructions and contains Status and Violation Registers. If the MEM is disabled, logical memory addresses access the first 32K of physical memory. If the MEM is enabled, logical memory addresses are translated to physical memory addresses via the system map or user map, depending on which one is currently enabled. DCPC memory addresses are translated via the port A map for channel 1 accesses and the port B map for channel 2 accesses.

The MEM simulator is enabled and disabled by the ***SET CPU DMS*** and ***SET CPU NODMS*** commands, rather than the ***SET MEM ENABLED*** and ***DISABLED*** commands. The MEM state contains these registers:

Name	*Size*	*Radix*	*Description*
ENABLED	1	2	Memory expansion is enabled
CURMAP	1	2	The user map is currently selected
STATUS	16	8	Status register
VIOL	16	8	Violation register
SMAP [0:31]	16	8	System map registers
UMAP [0:31]	16	8	User map registers
PAMAP [0:31]	16	8	DCPC port A registers
PBMAP [0:31]	16	8	DCPC port B registers

4 General I/O Device Simulations

The simulator provides these general I/O device interfaces and their associated peripherals:

- 12531C Buffered Teleprinter Interface
- 12539C Time Base Generator
- 12566B Microcircuit Interface
- 12597A-002 Tape Reader Interface
- 12597A-005 Tape Punch Interface
- 12620A/12936A Privileged Interrupt Fence
- 12653A Line Printer Interface
- 12792C 8-Channel Asynchronous Multiplexer
- 12845B Line Printer Interface
- 12875A Processor Interconnect
- 12920A Asynchronous Multiplexer Interface
- 12966A Buffered Asynchronous Communications Interface

4.1 12531C Buffered Teleprinter Interface with One 2752/54 Teleprinter

The 12531C Buffered Teleprinter Interface connects current-loop devices, such as the HP 2752A (ASR33) and 2754A (ASR35) teleprinters, as well as EIA RS-232 devices, such as the HP 2749A (ASR33) teleprinter and HP 2600 CRT terminal, to the HP computer family.

This simulation provides three units. Unit 0 reads from the teleprinter keyboard. Unit 1 writes to the teleprinter printer. Unit 2 writes to the teleprinter paper tape punch. Attaching a paper tape image file to unit 2 simulates loading blank paper tape into the punch:

```
ATTACH {-N} TTY2 <image-filename>
```

Adding the ***–N*** (new file) switch clears the contents of the image file if present. Without the ***–N*** switch, punch output will be appended to any preexisting image file content. Detaching the image simulates removing the punched tape.

The interface may be programmed to enable printing and punching separately. Only the HP 2754A teleprinter responds to the individual enables. All other supported devices print if either the printer or the punch is enabled. The TTY device simulates an HP 2754A only if the punch unit is attached. When the unit is detached, an HP 2752A is simulated, which will print if either printing or punching is enabled.

If the host operating system returns an error when writing to the paper tape image file, the simulator will report the error to the simulation console, e.g.:

```
HP 2100 simulator teleprinter punch I/O error: Permission denied
```

...and the simulator stops. Simulation may be resumed and the operation retried if the problem has been corrected or with the punch operation incomplete if the problem is uncorrectable.

Device options that may be specified are:

Option	*Action*
FASTTIME	Use optimized timing; default
REALTIME	Use realistic timing
SC=<n>	Set the interface card select code; default is 11
ENABLED	Enable the device; default
DISABLED	Disable the device

The interface supports realistic and optimized timing modes. Realistic timing accurately models the 10 character per second print and punch rates (in machine instructions). Optimized timing reduces operation delays to the minimums necessary to operate correctly; this is much faster than a real teleprinter would operate.

The delay used by the simulator in ***FASTTIME*** mode may be set via the register interface. The value may be adjusted as necessary to work around any HP software problems that are triggered by the unusually rapid print and punch operations. Resetting the device with the ***RESET –P*** (power-on reset) command restores the original optimized time.

If the TTY device disabled while the simulator is running, the simulation console will respond only to the SCP interrupt character, which defaults to CTRL+E.

Device configuration may be displayed with the following commands:

Command	*Action*
SHOW TTY	Display the device and unit configuration
SHOW TTY MODE	Display the timing mode

Unit options that may be specified for the keyboard and printer are:

Option	*Action*
UC	Upshift lowercase characters to uppercase; default
7B	Transfer characters with the high-order bit cleared
7P	Transfer characters with the high-order bit cleared, non-printing suppressed
8B	Transfer characters without changing
AUTOLF	Enable automatic output of line-feed characters
NOAUTOLF	Disable automatic output of line-feed characters; default

The printer may be set to mode ***UC***, ***7B***, ***7P***, or ***8B***. The set of non-printing characters allowed by mode ***7P*** is configured independently by the ***SET CONSOLE PCHAR*** command as described in the *SIMH User's Guide* and defaults to CR, LF, HT, BS, and BEL. The keyboard may be set to mode ***UC***, ***7B***, or ***8B***; mode ***7P*** is not supported for input and will be changed to mode ***7B*** if specified. The default mode is ***UC*** for both units.

Some HP software systems expect the console terminal to transmit line feed characters automatically following each carriage return. As an aid to avoid typing LF characters after pressing ENTER, the ***AUTOLF*** option may be specified for the keyboard unit. This simulates pressing the AUTO LF latching key on an HP 264x terminal.

When a debug log has been established, tracing may be configured by specifying one or more of the reporting level options:

Option	*Reporting Level*
CSRW	Interface control, status, read, and write actions
SERV	Printer and punch unit service events
PSERV	Keyboard unit service events (periodic)
XFER	Data receptions and transmissions
IOBUS	I/O bus signals and data words received and returned

The ***CSRW*** option traces control, status, read, and write commands sent to the interface. The ***SERV*** option traces print and punch output service scheduling and entries. The ***PSERV*** option traces keyboard poll unit service scheduling and entries. The poll service is entered once every 10 milliseconds to poll for input characters to be received by the interface. The ***XFER*** option traces the characters entered at the keyboard and output to the printer and punch. The ***IOBUS*** option traces the I/O backplane signals and data received and returned by the interface.

Examples of the trace formats follow:

```
>>TTY  csrw: Control is output | printer enabled
>>TTY  csrw: Output data is CR
>>TTY  csrw: Status is idle | 'T'
>>TTY  serv: Unit delay 200 service scheduled
>>TTY  serv: Printer and punch service entered
>>TTY pserv: Poll delay 221566 service entered
>>TTY  xfer: Character 'T' entered at keyboard
>>TTY  xfer: Character CR sent to printer
>>TTY iobus: Received data 000000 with signals IOI
>>TTY iobus: Returned data 000115 with signals (none)
>>TTY iobus: Received data 000000 with signals ENF | SIR | IEN | PRH
>>TTY iobus: Returned data 000000 with signals FLG | IRQ | SRQ
```

The teleprinter interface state contains these registers:

Name	*Size*	*Radix*	*Description*
BUF	8	8	Output data buffer
MODE	3	2	Control word bits 14-12
CTL	1	2	Control flip-flop
FLG	1	2	Flag flip-flop
FBF	1	2	Flag buffer flip-flop
KPOS	32	10	Number of characters input
TPOS	32	10	Number of characters printed
PPOS	32	10	Position in the punch output file
TTIME	24	10	Fast print/punch time

4.2 12539C Time Base Generator

The HP 12539C Time Base Generator provides a programmable interval clock employed as the DOS, RTE, and TSB time-of-day and process-switching clocks. The TBG provides programmable periods of 100 microseconds to 1000 seconds in decade increments. Each “tick” of the clock may request an interrupt, and a status indication is provided if the TBG ticks a second time without acknowledgement.

Device options that may be specified are:

Option	*Action*
CALTIME	Use calibrated timing; default
REALTIME	Use realistic timing
W1A	Install jumper W1 in the A position; default
W1B	Install jumper W1 in the B position
W2A	Install jumper W2 in the A position; default
W2B	Install jumper W2 in the B position
SC=<n>	Set the interface card select code; default is 13
ENABLED	Enable the device; default
DISABLED	Disable the device

Calibrated timing aligns the simulated clock periods with the clock on the host system. When calibrated, each of the programmable periods will elapse after the corresponding amount of host-system time.

When realistic timing is enabled, the simulator models the programmable periods in terms of machine instructions executed. Calibrated timing is required to enable CPU idling and is generally preferred when running operating systems that provide time-of-day clocks. Realistic timing is necessary to pass the hardware diagnostic.

Hardware jumpers W1 and W2 may be installed in either their A or B positions. With W1 in the A position, lost ticks are reported in bit 4 of the status word. In the B position, lost ticks are reported in both bits 5 and 4. With W2 in the A position, the four longest intervals are 1, 10, 100, and 1000 seconds per tick. In the B position, these intervals are divided by 1000 and tick after 1, 10, 100, and 1000 milliseconds to facilitate testing with the diagnostic.

When a debug log has been established, tracing may be configured by specifying one or more of the reporting level options:

Option	*Reporting Level*
CSRW	Interface control, status, read, and write actions
PSERV	Clock unit service events (periodic)
IOBUS	I/O bus signals and data words received and returned

The ***CSRW*** option traces control, status, read, and write commands sent to the interface. The ***PSERV*** option traces event service scheduling and entries, which occur at a periodic rate dependent on the programmable configuration. The ***IOBUS*** option traces the I/O backplane signals and data received and returned via the interface.

Examples of the trace formats follow:

```
>>TBG  csrw: Control is 10 millisecond rate
>>TBG  csrw: Status is lost tick
>>TBG pserv: Rate 100 millisecond delay 157729 service scheduled
>>TBG iobus: Received data 000000 with signals STC | CLF | SIR | PRH
>>TBG iobus: Returned data 000020 with signals PRL
```

The Time Base Generator state contains these registers:

Name	*Size*	*Radix*	*Description*
SEL	3	8	Time base selector
CTR	14	10	Prescale counter for < 1 Hz operation
CTL	1	2	Control flip-flop
FLG	1	2	Flag flip-flop
FBF	1	2	Flag buffer flip-flop
ERR	1	2	Error flip-flop

4.3 12566B Microcircuit Interface

The HP 12566B Microcircuit Interface provides a general-purpose 16-bit bidirectional data path at TTL logic levels for interfacing to a variety of peripheral devices. The card contains a 16-bit Data Output register and a 16-bit Data Input register for transfers between the CPU and the device. A Device Command signal from the card enables the device to transfer data, while a Device Flag signal from the device indicates transfer completion to the CPU. The card supports skip-on-flag and interrupt I/O modes, as well as DMA transfers up to half bandwidth (approximately 300K-400K words per second).

In simulation, this interface is not connected to an associated peripheral. Instead, it provides a configurable target for several diagnostics that test input, output, and interrupt responses. Two identical cards are provided with device names MC1 and MC2. They are disabled by default, as they are only useful when running diagnostics.

Device options that may be specified are:

Option	*Action*
DEVICE	Remove the diagnostic loopback connector; default
DIAGNOSTIC	Install the diagnostic loopback connector
SC=<n>	Set the interface card select code; defaults are 44 and 45
ENABLED	Enable the device
DISABLED	Disable the device; default

The ***DIAGNOSTIC*** option simulates the installation of the HP 1251-0332 diagnostic test (loopback) connector in place of the device cable. It also configures hardware jumpers W1-W3 on the interface as required by the following diagnostics:

W1	*W2*	*W3*	*DSN*	*Diagnostic*
C	**B**	**B**	**143300**	General Purpose Register
C	**B**	**B**	**141203**	I/O Instruction Group
C	**B**	**B**	**102103**	Memory Expansion Unit
C	**B**	**B**	**101220**	DMA/DCPC for 2100/1000
B	**A**	**A**	**—**	DMA for 2100 (HP 24195)
B	**A**	**A**	**101105**	DMA for 2114/2115/2116
B	**C**	**B**	**—**	DMA for 2115/2116 (HP 24185)

Setting the ***DEVICE*** option removes the loopback connector and leaves the device interface disconnected.

When a debug log has been established, tracing may be configured by specifying one or more of the reporting level options:

Option	*Reporting Level*
XFER	Data transmissions or receptions
IOBUS	I/O bus signals and data words received and returned

The ***XFER*** option traces the words sent or received by the interface during diagnostic operation. The ***IOBUS*** option traces the I/O backplane signals and data received and returned via the interface.

Examples of the trace formats follow:

```
>>MC1  xfer: Output data word 062140 looped back to input
>>MC1 iobus: Received data 000000 with signals ENF | SIR | IEN | PRH
>>MC1 iobus: Returned data 000000 with signals FLG | IRQ | SRQ
>>MC2 iobus: Received data 000000 with signals IOI
>>MC2 iobus: Returned data 000145 with signals (none)
```

The interface state contains these registers:

Name	*Size*	*Radix*	*Description*
IN	16	8	Input data register
OUT	16	8	Output data register
CTL	1	2	Control flip-flop
FLG	1	2	Flag flip-flop
FBF	1	2	Flag buffer flip-flop
CMD	1	2	Command flip-flop

4.4 12597A-002 Tape Reader Interface with One 2748 Paper Tape Reader

The 12597A-002 Tape Reader Interface is an 8-bit duplex register card that connects the HP 2748A/B Tape Reader to the HP computer family. The 2748 is a photoreader capable of reading eight-level punched paper tapes at a rate of 500 bytes per second.

The 12597A interface contains nine configuration jumpers. These are preset for proper tape reader operation.

The simulation provides one reader unit. Attaching a paper tape image file to the unit:

```
ATTACH PTR <image-filename>
```

... simulates pressing the LOAD button on the reader's front panel, loading the paper tape into the reader, and pressing the READ button.

The interface does not provide status to the CPU to indicate that a tape is not loaded or that the end of the tape has passed through the reader. Attempting to read under these conditions will cause the reader to hang, as it will in hardware. Specifying the ***SET CPU STOP=IOERR*** command will cause a simulation stop with a **No tape loaded** error if either of these conditions occur. If the error is corrected, either by attaching the paper tape image file or by rewinding the tape, then resuming simulation will retry the operation.

If the host operating system returns an error when reading from the paper tape image file, the simulator will report the error to the simulation console, e.g.:

```
HP 2100 simulator paper tape reader I/O error: Permission denied
```

...and the simulator stops. Simulation may be resumed and the operation retried if the problem has been corrected or with the reader operation incomplete if the problem is uncorrectable.

Device options that may be specified are:

Option	*Action*
READER	Use a normal paper tape; default
DIAGNOSTIC	Use a continuous paper tape loop or loopback connector
FASTTIME	Use optimized timing; default
REALTIME	Use realistic timing
REWIND	Reposition the paper tape to the beginning of the image
SC=<n>	Set the interface card select code; default is 10
ENABLED	Enable the device; default
DISABLED	Disable the device

The simulator provides two diagnostic modes. If a paper tape image file is not attached, then the ***DIAGNOSTIC*** option simulates the installation of the HP 1251-0332 diagnostic test (loopback) connector in place of the reader cable. This is needed to run the General Purpose Register Diagnostic (DSN 143300) as well as to serve as the standard I/O card for several other diagnostics that test interrupts. If a file is attached, then the ***DIAGNOSTIC*** option converts the paper tape image into a continuous loop by logically joining the ends of the tape. This is used by the High-Speed Tape Reader/Punch Diagnostic (DSN 146200). Setting the ***READER*** option returns the tape image to its normal linear configuration.

The reader supports realistic and optimized timing modes. Realistic timing attempts to model the paper tape movement delay inherent in the physical hardware. Optimized timing reduces the operation delay to the minimum necessary to operate correctly; this is much faster than a real reader would operate.

The delay used by the simulation in ***FASTTIME*** mode may be set via the register interface. The value may be adjusted as necessary to work around any HP software problems that are triggered by the unusually rapid reading operations. Resetting the device with the ***RESET –P*** (power-on reset) command restores the original optimized time.

Setting the ***REWIND*** option repositions the tape to the beginning of the paper tape image file. This is equivalent to detaching and then reattaching the same file and may be used to re-read the same tape.

The interface supports the ***LOAD PTR*** and ***BOOT PTR*** commands to load an absolute binary paper tape image into memory. Depending on the current CPU model, the ***LOAD*** command copies either the Basic Binary Loader (BBL) or the 12992K boot loader ROM into memory, configures it for the interface's select code, and sets the P register to the loader's starting address. The ***BOOT*** command performs these same actions and then presets the system and begins execution.

For a 2100/14/15/16 CPU containing a previously configured BBL in memory, the commands are equivalent in hardware to setting the P register to the starting address, pressing the LOADER ENABLE button on the front panel, and, for the ***BOOT*** command, pressing the INTERNAL PRESET, EXTERNAL PRESET, and RUN buttons to begin execution. Before execution, the S register must be set as follows:

15	***14***	***13***	***12***	***11***	***10***	***9***	***8***	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
●	0	0	0	0	0	0	0	0	0	0	0	0	0	0	●

If bit 15 is set to 1, the loader will compare the absolute program on tape to the contents of memory. If bit 0 is set to 1, the loader will verify the checksums of the absolute binary records on tape without altering memory. If neither bit is set, the loader will read the absolute program on the paper tape into memory. Loader execution ends with one of the following halt instructions:

- HLT 00 — a comparison error occurred; A = the tape value.

- HLT 11 — a checksum error occurred; A = the tape value, B = the calculated value.
- HLT 55 — the program load address would overlay the loader.
- HLT 77 — the end of tape was reached with a successful read.

For a 1000 M/E/F-Series CPU, the commands are equivalent in hardware to setting the S register to select and configure the HP 12992K boot loader ROM, pressing the IBL button to copy the ROM into memory, and, for the ***BOOT*** command, pressing the PRESET and RUN buttons to begin execution. The S register is preset as follows:

15	***14***	***13***	***12***	***11***	***10***	***9***	***8***	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
0	0	0	0	PTR select code						0	0	0	0	0	0

Loader execution ends with one of the following halt instructions:

- HLT 11 — a checksum error occurred; A = the calculated value, B = the tape value.
- HLT 55 — the load address would overlay the ROM loader.
- HLT 77 — the end of tape was reached with a successful read.

When a debug log has been established, tracing may be configured by specifying one or more of the reporting level options:

Option	***Reporting Level***
SERV	Reader unit service events
XFER	Data bytes read from the tape
IOBUS	I/O bus signals and data words received and returned

The ***SERV*** option traces reader event service scheduling and entries. The ***XFER*** option traces the bytes received from the reader. The ***IOBUS*** option traces the I/O backplane signals and data received and returned via the interface.

Examples of the trace formats follow:

```
>>PTR  serv: Unit delay 3155 service scheduled
>>PTR  serv: Reader service entered
>>PTR  xfer: Data 145 character 'e' received from reader
>>PTR iobus: Received data 000000 with signals ENF | SIR | IEN | PRH
>>PTR iobus: Returned data 000000 with signals FLG | IRQ | SRQ
>>PTR iobus: Received data 000000 with signals IOI
>>PTR iobus: Returned data 000145 with signals (none)
```

The paper tape reader state contains these registers:

Name	***Size***	***Radix***	***Description***
BUF	8	8	Input data buffer
CTL	1	2	Control flip-flop
FLG	1	2	Flag flip-flop
FBF	1	2	Flag buffer flip-flop
TRLLIM	8	10	Trailing feed frame limit
POS	32	10	Paper tape image file position
TIME	24	10	Fast read time

The TRLLIM register specifies the number of feed frames (zero bytes) to supply as paper tape trailer when the end of the tape image file is reached; the default is 40 frames. If input is attempted when TRLLIM is set to zero or the count is exhausted, the reader will hang.

4.5 12597A-005 Tape Punch Interface with One 2895 Paper Tape Punch

The 12597A-005 Tape Punch Interface is an 8-bit duplex register card that connects the HP 2895A/B Tape Punch to the HP computer family. The 2895 is an eight-level paper tape punch capable of punching at a rate of 75 bytes per second.

The 12597A interface contains nine configuration jumpers. These are preset for proper tape punch operation.

The simulation provides one punch unit. Attaching a paper tape image file to the unit simulates loading blank paper tape into the punch:

```
ATTACH {-N} PTP <image-filename>
```

Adding the ***–N*** (new file) switch clears the contents of the image file if present. Without the ***–N*** switch, punch output will be appended to any preexisting image file content. Detaching the image simulates removing the punched tape.

If the host operating system returns an error when writing to the paper tape image file, the simulator will report the error to the simulation console, e.g.:

```
HP 2100 simulator paper tape punch I/O error: No space left on device
```

...and the simulator stops. Simulation may be resumed and the operation retried if the problem has been corrected or with the punch operation incomplete if the problem is uncorrectable.

Device options that may be specified are:

Option	*Action*
PUNCH	Connect to the paper tape punch; default
DIAGNOSTIC	Use a loopback connector
FASTTIME	Use optimized timing; default
REALTIME	Use realistic timing
SC=<n>	Set the interface card select code; default is 12
ENABLED	Enable the device; default
DISABLED	Disable the device

Setting the ***DIAGNOSTIC*** option simulates the installation of the HP 1251-0332 diagnostic test (loopback) connector in place of the punch cable. This is needed to run the General Purpose Register Diagnostic (DSN 143300) as well as to serve as the standard I/O card for several other diagnostics that test interrupts. Setting the ***PUNCH*** option reinstalls the standard punch cable.

The punch supports realistic and optimized timing modes. Realistic timing attempts to model the punch and tape advance delays inherent in the physical hardware. Optimized timing reduces the operation delays to the minimums necessary to operate correctly; this is much faster than a real punch would operate.

The delay used by the simulation in ***FASTTIME*** mode may be set via the register interface. The value may be adjusted as necessary to work around any HP software problems that are triggered by the unusually rapid punching operations. Resetting the device with the ***RESET –P*** (power-on reset) command restores the original optimized time.

When a debug log has been established, tracing may be configured by specifying one or more of the reporting level options:

Option	*Reporting Level*
CSRW	Interface control, status, read, and write actions
SERV	Punch unit service events
XFER	Data bytes written to the tape
IOBUS	I/O bus signals and data words received and returned

The ***CSRW*** option traces control, status, read, and write commands sent to the interface. The ***SERV*** option traces punch event service scheduling and entries. The ***XFER*** option traces the bytes sent to the punch. The ***IOBUS*** option traces the I/O backplane signals and data received and returned via the interface.

Examples of the trace formats follow:

```
>>PTP  csrw: Status is tape low
>>PTP  serv: Unit delay 20978 service scheduled
>>PTP  serv: Punch service entered
>>PTP  xfer: Data 045 character '%' sent to punch
>>PTP iobus: Received data 000045 with signals IOO
>>PTP iobus: Returned data 000000 with signals (none)
>>PTP iobus: Received data 000000 with signals STC | CLF | SIR
>>PTP iobus: Returned data 000000 with signals (none)
```

The paper tape punch state contains these registers:

Name	*Size*	*Radix*	*Description*
BUF	8	8	Output data buffer
CTL	1	2	Control flip-flop
FLG	1	2	Flag flip-flop
FBF	1	2	Flag buffer flip-flop
POS	32	10	Paper tape image file position
TIME	24	10	Fast punch time

4.6 12620A/12936A Privileged Interrupt Fence

The Privileged Interrupt Fence (PIF) is used in DOS and RTE systems to provide privileged interrupt capability. The PIF is installed in the I/O backplane to separate privileged from unprivileged devices by controlling the interrupt priority chain signal (PRL) to lower-priority devices. The privileged cards located below the fence are allowed to interrupt the operating system and the service routines of the unprivileged cards located above the fence. Privileged devices employ specially written device drivers that bypass the operating system during interrupts. This provides very fast interrupt service time.

HP produced two PIF cards: the 12620A Breadboard Interface for the RTE operating systems, and the 12936A Privileged Interrupt Fence Accessory for DOS. They behave quite differently and are not interchangeable.

Device options that may be specified are:

Option	*Action*
12620A	Use a 12620A RTE fence; default
12936A	Use a 12936A DOS fence
SC=<n>	Set the interface card select code; default is 37
DEBUG=<option>	Enable tracing
NODEBUG	Disable tracing; default
ENABLED	Enable the device; default
DISABLED	Disable the device

Device configuration may be displayed with the following command:

Command	*Action*
SHOW PIF TYPE	Display the current fence type

When a debug log has been established, tracing may be configured by specifying one or more of the reporting level options:

Option	*Reporting Level*
CMD	Interface commands
IOBUS	I/O bus signals and data words received and returned

The ***CMD*** option traces commands executed by the interface. The ***IOBUS*** option traces the I/O backplane signals and data received and returned via the interface.

Examples of the trace formats follow:

```
>>PIF   cmd: Control reset
>>PIF   cmd: Fence allows lower-priority interrupts
>>PIF   cmd: Fence inhibits lower-priority interrupts
>>PIF iobus: Received data 000000 with signals SFS | PRH
>>PIF iobus: Returned data 000000 with signals SKF | PRL
```

Because the PIF is accessed twice for each unprivileged interrupt, systems employing the 12539C Time Base Generator will generate a large number of debug statements very quickly. In such cases, tracing should be enabled for the minimum time necessary.

The Privileged Interrupt Fence state contains these registers:

Name	*Size*	*Radix*	*Description*
CTL	1	2	Control flip-flop
FLG	1	2	Flag flip-flop
FBF	1	2	Flag Buffer flip-flop

4.7 12653A Line Printer Interface with One 2767 Line Printer

The HP 12653A Line Printer Interface connects the HP 2767A line printer to the HP computer family. The printer provides 80-column output at rates from 356 to 1110 lines per minute, depending on the number of columns printed. It uses a 64-character repertoire (ASCII codes 32 through 95), prints a fixed 60 lines per page, and

automatically skips page perforations. It does not have a vertical format unit (VFU) but will slew to the top-of-form upon receipt of a form-feed character.

The simulation provides one printer unit. Attaching a text file to the unit simulates loading paper into the printer:

```
ATTACH {-N} LPS <image-filename>
```

Adding the ***–N*** (new file) switch clears the contents of the image file if present. Without the ***–N*** switch, printer output will be appended to any preexisting image file content.

Printer output written to the image file is typically buffered by the host operating system's underlying stream I/O routines. While it is running, the simulator flushes the file after each printer top-of-form request to permit convenient inspection of the image file. Stopping the simulator also flushes the file.

If the host operating system returns an error when writing to the printer image file, the simulator will report the error to the simulation console, e.g.:

```
HP 2100 simulator printer I/O error: No space left on device
```

The printer goes offline, and the simulator stops. Simulation may then be resumed, either with the printer set back online if the problem is fixed, or with the printer remaining offline if the problem is uncorrectable.

Device and unit options include configuring the printer type and timing, turning the printer power off and on, and the ability to set the printer offline or online. The command forms are:

```
SET LPS <device-option>
SET LPS <unit-option>
```

Device options that may be specified are:

Option	***Action***
FASTTIME	Use optimized timing; default
REALTIME	Use realistic timing
PRINTER	Install the standard printer cable; default
DIAGNOSTIC	Install the diagnostic loopback connector
SC=<n>	Set the interface card select code; default is 14
DEBUG=<option>	Enable tracing
NODEBUG	Disable tracing; default
ENABLED	Enable the device
DISABLED	Disable the device; default

The printer supports realistic and optimized timing modes. Realistic timing attempts to model the print buffer load and print-and-space operation delays inherent in the physical hardware. For example, output of lines with more characters takes longer than output of lines with fewer characters, and spacing six lines takes approximately six times longer than spacing one line.

Optimized timing reduces operation delays to the minimums necessary to operate correctly; this is much faster than a real line printer would operate.

The delays used by the simulation in ***FASTTIME*** mode may be set via the register interface. The values may be adjusted as necessary to work around any HP software problems that are triggered by the unusually rapid print operations. Resetting the device with the ***RESET –P*** (power-on reset) command restores the original optimized times.

The ***DIAGNOSTIC*** option simulates the installation of the HP 1251-0332 diagnostic test (loopback) connector in place of the printer cable. This is needed to run the General Purpose Register Diagnostic (DSN 143300) as well as to serve as the standard I/O card for several other diagnostics that test interrupts.

Setting the ***PRINTER*** option reinstalls the standard printer cable.

Device configuration may be displayed with the following commands:

Command	*Action*
SHOW LPS	Display the device and unit configuration
SHOW LPS TIMING	Display the timing mode

Unit options that may be specified are:

Option	*Action*
OFFLINE	Set the printer offline; default when detached
ONLINE	Set the printer online; default when attached
POWEROFF	Turn the printer power off
POWERON	Turn the printer power on; default

The ***OFFLINE*** and ***ONLINE*** options place the printer offline and online, respectively. The former provides a convenient method of setting the printer "down" without detaching the associated output file.

Entering the ***SET LPS POWEROFF*** command simulates removing power from the line printer. A ***SET LPS POWERON*** command restores power to the printer.

When a debug log has been established, tracing may be configured by specifying one or more of the reporting level options:

Option	*Reporting Level*
CMDS	Command initiations and completions
CPU	Words received from and sent to the CPU
XFER	Data receptions and transmissions
STATE	Printer state transitions
IOBUS	I/O bus signals and data words received and returned

The ***CMDS*** option traces the commands executed by the printer. The ***CPU*** option traces words received from (control and write data) and sent to (status and read data) the CPU. The ***XFER*** option traces the characters sent to the printer. The ***STATE*** option traces printer state changes. The ***IOBUS*** option traces the I/O backplane signals and data received and returned via the interface.

Examples of the trace formats follow:

```
>>LPS  cmds: Character 'M' scheduled for line 6, column 20, time = 13825
>>LPS  cmds: Character CR scheduled for line 6, column 81, time = 0
>>LPS   cpu: Control 000015 (CR) output
>>LPS   cpu: Status 100001 returned
>>LPS  xfer: Character 'D' printed
>>LPS  xfer: Character ETX erased
>>LPS state: Power state is OFF
>>LPS state: Power state is TURNING ON, scheduled time = 39500
>>LPS iobus: Received data 000000 with signals CLC | CLF | SIR | PRH
>>LPS iobus: Returned data 000000 with signals PRL
```

The Line Printer Controller state contains these registers:

Name	*Size*	*Radix*	*Symbolic*	*Read-Only*	*Action*
BUF	16	8	✓		Output buffer
STA	16	8			Printer status
POWER	2	8		✓	Printer power state
CTL	1	2			Control flip-flop
FLG	1	2			Flag flip-flop
FBF	1	2			Flag buffer flip-flop
CCNT	7	10			Current character count
LCNT	7	10			Current line count
POS	32	10			Printer file position
CTIME	24	10			Character transfer time
PTIME	24	10			Per-zone print operation time
STIME	24	10			Per-line paper slew time
RTIME	24	10			Power-on ready delay time

The BUF register may be displayed in character format, if desired. Symbolic entry is also supported.

4.8 12792C 8-Channel Asynchronous Multiplexer

The 12792C is an eight-channel asynchronous serial multiplexer that connects terminals, modems, serial line printers, and "black box" devices that use the RS-232 standard to the CPU. It uses an on-board Z80 microprocessor and provides input and output buffering to support block-mode reads from HP 264x and 262x terminals at speeds up to 19.2K baud. The interface handles character editing, echoing, ENQ/ACK handshaking, and read terminator detection, substantially reducing the load on the CPU over the earlier 12920A multiplexer.

There are four major revisions to the multiplexer firmware, designated revisions A-D. Revisions A-C have an upward-compatible command set and are supported under RTE-M, RTE-IVB, and RTE-6/VM. Revision D implements completely different commands, requires different RTE drivers, and is supported only under RTE-6/VM. This simulation implements revision C.

Revisions B and C added support for the 37214A Systems Modem subsystem and the RTE-A Virtual Control Panel (VCP), as the same firmware was used in the A-Series version of the multiplexer, the HP 12040B/C. Under simulation, the modem commands return status codes indicating that no modems are present, and the VCP commands are not implemented.

After pressing the CPU front-panel PRESET button or executing a CLC 0 instruction, the multiplexer enters “echoplex” mode. In this mode, each channel defaults to 9600 baud, 8 bits per character, 1 stop bit, no parity, and echoes characters as they are received. Echoplex is implemented under simulation — until a channel is configured by the appropriate RTE driver control call, characters entered in a Telnet session will be echoed.

The 12792C has two baud-rate generators that are assigned to channels by the wiring configuration in the I/O cable connector hood. Two counter/timer circuits on the interface are used to implement the BRGs for all eight channels. For channels connected to the same BRG, only subsets of the configurable rates are allowed, and assigning mutually incompatible rates causes corruption of the rates on channels assigned earlier. Under simulation, any baud rate may be assigned to any channel without interaction.

While the multiplexer can connect to generic Telnet clients, RTE programs generally assumed the use of HP terminals.

Under simulation, the multiplexer performs input and output through Telnet sessions connected to a user-specified listening port or through individually specified host serial ports. It supports concurrent Telnet and serial connections. The ***ATTACH*** command specifies the local port to be used for Telnet connections:

```
ATTACH MPX <port>
```

...where ***port*** is a decimal number between 1 and 65535 that is not being used for other TCP/IP activities. When the multiplexer is attached and the simulator is running, the multiplexer listens for connections on the specified port and assigns them to channels in ascending numeric order, unless an alternate connection order has been specified.

The ***ATTACH*** command is also used to specify the host serial port for an individual multiplexer channel:

```
ATTACH MPX <chan> <port-name>{;<rate>-<size><parity><stopbits>}
```

...where ***chan*** is the multiplexer channel number from 0-7, and ***port-name*** is the host name of the serial port to use (e.g., /dev/*ttyS0* or *COM1*).

An optional serial port configuration string may be supplied after the host name. The required values are:

- ***rate*** is the baud rate in bits per second.
- ***size*** is the character size in bits including the parity bit, if designated.
- ***parity*** designates the parity to use: *N* (no), *E* (even), *O* (odd), *M* (mark), or *S* (space).
- ***stopbits*** is the number of stop bits (*1*, *1.5*, or *2*).

If the port configuration string is omitted, the default configuration specified by the host system for that port is used. Configuration options are available for the device and for the individual units. The command forms are:

```
SET MPX <device-option>
SET MPXn <unit-option>
```

4.8.1 Device Options

Device options that may be specified are:

Option	***Action***
LINEORDER=<c1>[;<c2>...]	Set the channel connection order
SC=<n>	Set the interface card select code; default is 36
DEBUG=<option>	Enable tracing
NODEBUG	Disable tracing; default
ENABLED	Enable the device; default
DISABLED	Disable the device

The ***LINEORDER*** option specifies the order in which new connections are assigned to multiplexer channels. The arguments may be single channel numbers or ranges of channel numbers of the form ***m-n***, with multiple arguments separated by semicolons. Telnet connections to the listening port will be assigned to multiplexer channels in the sequence specified. Omitted channels will not receive connections, unless the ***ALL*** keyword is supplied as the last argument. In the absence of a ***SET MPX LINEORDER*** command, connections will be assigned by default in ascending channel order. The default order may be reestablished by specifying the command ***SET MPX LINEORDER=ALL***.

4.8.2 Unit Options

Unit options that may be specified are:

Option	*Action*
FASTTIME	Use optimized timing; default
REALTIME	Use realistic timing
CAPSLOCK	Upshift lowercase input characters to uppercase
NOCAPSLOCK	Input characters are unchanged; default
LOG=<filename>	Enable I/O logging
NOLOG	Disable I/O logging; default
DISCONNECT	Disconnect the channel

The 12792C supports twelve programmable baud rates from 50 to 19200 baud. When realistic timing is enabled, the simulation accurately models the transmission and reception baud rates (in machine instructions). For example, setting the interface to 1200 baud will take twice as long to output a given listing as setting the interface to 2400 baud. Optimized timing reduces the timing to the minimum necessary to operate correctly; this is much faster than a real multiplexer would operate.

Setting the multiplexer to fast timing also enables three optimizations. First, buffered characters are transferred via Telnet in blocks, rather than one character at a time. Second, ENQ/ACK handshaking is done locally, without involving the Telnet client. Third, when editing and echo is enabled, entering BS echoes a backspace, a space, and a backspace, and entering DEL echoes a backslash, a carriage return, and a line feed, providing better compatibility with prior RTE terminal drivers.

Many HP operating systems require command input in upper case, although mixed-case output is supported. As an aid to avoid toggling the host keyboard in and out of CAPS LOCK mode, the multiplexer provides this function locally. The default mode is ***NOCAPSLOCK***.

Each multiplexer channel supports independent I/O logging to a file. Adding the ***–N*** (new file) switch clears the contents of the log file if it is present. Without the ***–N*** switch, channel output will be appended to any preexisting log file content. Disabling logging also closes the log file.

A channel may be manually disconnected from its associated Telnet session with the ***SET MPXn DISCONNECT*** command. Otherwise, the connection will remain open until disconnected either by the Telnet client or a ***DETACH MPX*** command. For a serial connection, the ***SET MPXn DISCONNECT*** command will drop and then raise the Data Terminal Ready line; to disconnect the serial port, issue a ***DETACH MPXn*** command.

The device configuration may be displayed with the following commands:

Command	*Action*
SHOW MPX	Display the device configuration
SHOW MPX LINEORDER	Display the channel connection order
SHOW MPX CONNECTIONS	Display the channel connections
SHOW MPX STATISTICS	Display the channel I/O statistics for the active connections

The unit configuration may be displayed with the following commands:

Command	*Action*
SHOW MPX<n>	Display the selected channel's configuration
SHOW MPX<n> LOG	Display the selected channel's logging status

4.8.3 Tracing and Registers

When a debug log has been established, tracing may be configured by specifying one or more of the reporting level options:

Option	*Reporting Level*
CMDS	Command initiations and completions
CPU	Words received from and sent to the CPU
BUF	Data read from and written to the channel buffers
XFER	Data receptions and transmissions
IOBUS	I/O bus signals and data words received and returned

The ***CMDS*** option traces the commands executed by the multiplexer. The ***CPU*** option traces words received from (control and write data) and sent to (status and read data) the CPU. The ***BUF*** option traces reads and writes from the internal channel buffers. The ***XFER*** option traces the characters sent and received from the multiplexer channels. The ***IOBUS*** option traces the I/O backplane signals and data received and returned via the interface.

Examples of the trace formats follow:

```
>>MPX  cmds: [OTx] Command 037 parameter 100000 scheduled, time = 1250
>>MPX   cpu: [Lix,C] Status = 102000
>>MPX   buf: Port 2 character 'A' get from read buffer [73]
>>MPX  xfer: Port 0 character LF transmitted
>>MPX iobus: Received data 000000 with signals STF | PRH
```

The multiplexer state contains these registers:

Name	*Size*	*Radix*	*Symbolic*	*Action*
STATE	3	10		Controller state
IBUF	16	8	✓	Input buffer (from multiplexer to CPU)
OBUF	16	8	✓	Output buffer (from CPU to multiplexer)
CMD	8	8		Command code
PARAM	16	8		Command parameter
PORT	8	10		Port number for last command
PORTKEY	8	10		Port key for last command
IOLEN	16	10		I/O transfer length for last command
UIEN	1	2		Unsolicited interrupts enabled
UIPORT	3	10		Unsolicited interrupt port
UICODE	3	10		Unsolicited interrupt reason code
KEYS [0:7]	8	10		Port keys, ports 0-7
PCONFIG [0:7]	16	8		Port configuration, ports 0-7
RCVTYPE [0:7]	16	2		Receive type, ports 0-7
CHARCNT [0:7]	16	8		Received character count, ports 0-7
TERMCNT [0:7]	16	8		Termination character count, ports 0-7
FLOWCNTL [0:7]	16	8		Flow control, ports 0-7
ENQCNTR [0:7]	7	10		ENQ character counter, ports 0-7
ACKWAIT [0:7]	10	10		ACK wait counter, ports 0-7
PFLAGS [0:7]	12	2		Port flags, ports 0-7
RBUF [0:7] [0:513]	8	8	✓	Read buffers, ports 0-7
WBUF [0:7] [0:513]	8	8	✓	Write buffers, ports 0-7
GET [0:7] [0:1]	10	10		Read/write get indexes, ports 0-7
SEP [0:7] [0:1]	10	10		Read/write separator indexes, ports 0-7
PUT [0:7] [0:1]	10	10		Read/write put indexes, ports 0-7
CTL	1	2		Control flip-flop
FLG	1	2		Flag flip-flop
FBF	1	2		Flag buffer flip-flop

The RBUF and WBUF registers logically represent eight character buffers of 514 bytes each, and the GET, SEP, and PUT registers logically represent eight sets of two-word indexes. However, SCP supports only one-dimensional register arrays, so they must be examined and deposited as linear arrays of 4112 bytes and 16 words, respectively. So, for example, RBUF [0] is byte 0 of buffer 0, RBUF [513] is byte 513 of buffer 0, RBUF [514] is byte 0 of buffer 1, etc.

The IBUF and OBUF registers default to octal display but may be displayed in single- or two-character formats by specifying the ***–A*** or ***–C*** switch, respectively. Symbolic entry for these registers is also supported. The RBUF and WBUF registers default to single-character format display and entry but may be overridden with numeric-format switches, if desired.

4.9 12845B Line Printer Interface with One 2607/13/17/18 Line Printer

The HP 12845B Line Printer Interface connects the HP 2607A (200 lines per minute), HP 2613 (300 lpm), HP 2617 (600 lpm), and HP 2618 (1250 lpm) line printers to the HP computer family. The simulation provides one printer unit; the 2607 is selected by default. Attaching a text file to the unit simulates loading paper into the printer:

```
ATTACH {-N} LPT <image-filename>
```

Adding the ***–N*** (new file) switch clears the contents of the image file if present. Without the ***–N*** switch, printer output will be appended to any preexisting image file content.

Printer output written to the image file is typically buffered by the host operating system's underlying stream I/O routines. While it is running, the simulator flushes the file after each printer top-of-form request to permit convenient inspection of the image file. Stopping the simulator also flushes the file.

If the host operating system returns an error when writing to the printer image file, the simulator will report the error to the simulation console, e.g.:

```
HP 2100 simulator printer I/O error: No space left on device
```

The printer goes offline with an alarm condition, and the simulator stops. Simulation may then be resumed, either with the printer set back online if the problem is fixed, or with the printer remaining offline if the problem is uncorrectable.

Detaching the text file from the unit with the ***DETACH LPT*** command simulates running out of paper. If the command is entered while there are characters in the print buffer or, for the 2607 only, the print location is not at the top of the form, **Command not completed** is displayed on the simulation console, and the file remains attached until the required conditions are true. Once simulation is resumed and the print operations complete, the printer is set offline and detached automatically.

Detaching may also be forced with the ***DETACH –F LPT*** command. This simulates physically removing the paper and takes effect immediately, regardless of any printing operations in progress.

Device and unit options include configuring the printer type and timing, output format, vertical format unit (VFU), and the ability to set the printer offline or online. The command forms are:

```
SET LPT <device-option>
SET LPT <unit-option>
```

4.9.1 Device Options

Device options that may be specified are:

Option	*Action*
FASTTIME	Use optimized timing; default
REALTIME	Use realistic timing
PRINTER	Install the standard printer VFU tape; default
DIAGNOSTIC	Install the diagnostic VFU tape
SC=<n>	Set the interface card select code; default is 15
DEBUG=<option>	Enable tracing
NODEBUG	Disable tracing; default
ENABLED	Enable the device; default
DISABLED	Disable the device

The printer supports realistic and optimized timing modes. Realistic timing attempts to model the print buffer load and print-and-space operation delays inherent in the physical hardware. For example, output of lines with more characters takes longer than output of lines with fewer characters, and spacing six lines takes approximately six times longer than spacing one line.

Optimized timing reduces operation delays to the minimums necessary to operate correctly; this is much faster than a real line printer would operate.

The delays used by the simulator in ***FASTTIME*** mode may be set via the register interface. The values may be adjusted as necessary to work around any HP software problems that are triggered by the unusually rapid print operations. Resetting the device with the ***RESET –P*** (power-on reset) command restores the original optimized times.

The ***DIAGNOSTIC*** option simulates the installation of the HP 02613-80002 or 02618-80002 diagnostic VFU tape. This is needed to run the 2613/17/18 diagnostic (DSN 145103). Setting the ***PRINTER*** option reinstalls the standard VFU tape (02607-80024 for the 2607 and 02613-80001 for the 2613, 2617, and 2618).

Device configuration may be displayed with the following commands:

Command	*Action*
SHOW LPT	Display the device and unit configuration
SHOW LPT MODES	Display the timing and VFU tape modes

4.9.2 Unit Options

Unit options that may be specified are:

Option	*Action*
2607	Set the printer model to 2607; default
2613	Set the printer model to 2613
2617	Set the printer model to 2617
2618	Set the printer model to 2618
VFU	Install the HP standard 66-line VFU tape; default
VFU=<filename>	Install a custom VFU tape image
OFFLINE	Set the printer offline; default when detached
ONLINE	Set the printer online; default when attached
EXPAND	Write expanded output to the printer image file; default
COMPACT	Write compact output to the printer image file
POWEROFF	Turn the printer power off
POWERON	Turn the printer power on; default

The ***2607***, ***2613***, ***2617***, and ***2618*** options select the printer model. Each printer is configured with Option 001, which provides a 128-character set for the HP 2607 and 96-character sets for the HP 2613, 2617, and 2618. The 2607 and 2618 support 132-character print line lengths, while the 2613 and 2617 support 136-character lines. Exceeding the line length causes an automatic print-and-space operation on the 2607 and discards the extra characters on the 2613, 2617, and 2618.

The ***VFU*** option configures the printer's vertical format unit, as described in the *Vertical Format Unit* section below.

The ***OFFLINE*** and ***ONLINE*** options place the printer offline and online, respectively. The former provides a convenient method of setting the printer "down" without detaching the associated output file.

The printer will not go offline if there are characters in the print buffer. Instead, the offline condition is held off until the line is printed and paper movement is complete. The ***SET LPT OFFLINE*** command checks for data in the print buffer or a print operation in progress. If either condition is true, the action is deferred, and **Command not completed** is displayed on the simulation console. A ***SHOW LPT*** command will show that the device is still online. Once simulation is resumed and the print operation completes, the printer is set offline. No console message reports this, although a subsequent ***SHOW LPT*** command will indicate the new status.

Entering a ***SET LPT ONLINE*** command while an offline or detach action is deferred will cancel the action without triggering any programmed offline-to-online status transition interrupt. A ***RESET LPT*** command also cancels any deferred offline or detach action. Additionally, it clears the print buffer and terminates any print action in progress, so a ***SET LPT OFFLINE*** or ***DETACH LPT*** will succeed if issued afterward.

The ***EXPAND*** and ***COMPACT*** options control the format of lines written to the printer image file. In compact mode, a carriage-return/line-feed character pair terminates a printed line, but subsequent line spacing is performed by line-feed characters alone. A top-of-form request will emit a form-feed character instead of the number of line-feeds required to reach the top of the next form. This mode is suitable for sending the printer output to a physical printer connected to the host.

In expanded mode, paper advance is handled by emitting the correct number of carriage-return/line-feed pairs. This mode is suitable for retaining printer output as a text file.

The HP 2613, 2617, and 2618 printers are capable of overprinting. If the printer is sent a format code to suppress spacing, the characters in the buffer are printed and the buffer is emptied, but the paper is not advanced. The next print operation will print its characters over those already present on the paper. In simulation, overprinting is performed in one of two ways, depending on the print mode.

In compact mode, overprinting is simulated by emitting a carriage-return character at the end of the line. In expanded mode, overprinting is simulated by merging characters in the buffer before writing them as a single line in the printer image file. As the second and subsequent lines are output, each new character is compared with its corresponding character in the buffer. If the character in the buffer is a space, the new character replaces it. If the new character is a space or is the same as the character in the buffer, the character in the buffer is retained. Otherwise, the character in the buffer is replaced with a special character representing an overprinted combination. This character defaults to DEL (octal 177) but may be changed by altering the OVPCHR-register value.

The HP 2607 printer cannot overprint. A request to suppress spacing will result in a single-line paper advance after printing.

Entering the ***SET LPT POWEROFF*** command simulates removing power from the line printer. A ***SET LPT POWERON*** command restores power to the printer.

4.9.3 Vertical Format Unit

The HP 2607 supports an 8-channel vertical format unit (VFU), and the HP 2613, 2617, and 2618 printers support 12-channel VFUs. A continuous punched paper tape that controls paper spacing is installed in the VFU reader. The printer may be commanded to advance the paper until a punched hole is detected in a specified VFU channel. The length of the tape establishes the length of the forms loaded into the printer.

Initially, the standard VFU tape (part number 02607-80024 for the HP 2607 or 02613-80001 for the HP 2613, 2617, and 2618) is installed. This tape associates commands for channels 1-8 with the following printer actions:

Channel	*Printer Advances to the*
1	Top of form
2	Bottom of form
3	Next single space
4	Next double space
5	Next triple space
6	Next half-page
7	Next quarter-page
8	Next sixth-page

For the 02613-80001 tape, channel 9 is punched the same as channel 2, and channels 10-12 are uncommitted.

If the device is in diagnostic mode, the appropriate diagnostic VFU tape (02613-80002 for the 2613 and 2617or 02618-80002 for the 2618) is installed. These tapes associate commands for channels 1-8 as above and for channels 9-12 with the following actions:

Channel	*Printer Advances to the*
9	Bottom of form
10	Line before the bottom of form
11	Line before the top of form
12	Top of form

Setting the device back to printer mode restores the standard VFU tape.

The simulator supports the use of custom VFU tape images. A custom tape may be installed in place of the standard tape by issuing the ***SET LPT VFU=<filename>*** command. Custom tapes must reserve channel 1 for the top-of-form location, but the other channels may be defined as desired.

A custom tape image is a plain-text file that starts with a VFU definition line and then contains one channel-definition line for each line of the form. The number of channel-definition lines establishes the form length.

A semicolon appearing anywhere on a line begins a comment, and the semicolon and all following characters are ignored. Zero-length lines, including lines beginning with a semicolon, are ignored. Note that a line containing one or more blanks is not a zero-length line, so, for example, the line " *; a comment starting in column 2*" is not ignored.

The first (non-ignored) line in the file must be a VFU definition line of this exact form:

```
VFU=<punch characters>,<no-punch character>{,<title>}
```

...where:

- ***punch characters*** is a set of one or more characters used interchangeably to represent a punched location.
- ***no-punch character*** is a single character representing a non-punched location.
- ***title*** is an optional description that is printed by the ***SHOW LPT VFU*** command; the description "Custom VFU" is used if the title is omitted.

If the ***VFU*** line is missing or not of the correct form, then **Format error** is displayed on the simulation console, and the VFU tape is not changed.

The remaining lines define the channels punched for each line of the printed form. The line format consists of a sequence of punch, no-punch, and “other” characters, in channel order. Each punch or no-punch character defines a channel state, starting with channel 1 and proceeding left-to-right until all channels for the VFU are defined; any the extra channel states on the line are ignored. If the line terminates before all channels are defined, the remaining channels are set to the no-punch state. Any "other" characters, i.e., neither punch characters nor no-punch characters, are ignored and may be used freely to delineate the tape channels.

For a standard 66-line form, the first printable line (form line 1) is paper line 4, and the last printable line (form line 60) is paper line 63. The first channel-definition line in the file specifies form line 1, i.e., the top of the form, and must have a punch in channel 1. Channel line 60 defines form line 60, i.e., the bottom of the form. Six more channel lines follow: three for the bottom margin, and three for the top margin of the next form.

An example custom VFU file using a 66-line tape definition for an 8-channel VFU is:

```
; the VFU definition
;
; the set of punch characters is "1" and "X"
; the no-punch character is "0"
; all other characters are ignored

VFU=1X,0,A binary tape image

; the channel definitions

1 0 1 1 1 1 1 1   ; top of form (line 1; must have a punch in channel 1)
0-0-X-0-0-0-0-0   ; single space (line 2)
0011              ; channels 5-8 default to no-punch (line 3)
[...]
0 1 1 0 0 0 0 0   ; bottom of form (line 60)
0 0 0 0 0 0 0 0   ; bottom form margin (line 61)
0 0 0 0 0 0 0 0   ; bottom form margin (line 62)
0 0 0 0 0 0 0 0   ; bottom form margin (line 63)
0 0 0 0 0 0 0 0   ; top of form margin (line 64)
0 0 0 0 0 0 0 0   ; top of form margin (line 65)
0 0 0 0 0 0 0 0   ; top of form margin (line 66)
```

If a custom tape has been used, the standard tape may be reinstalled by issuing the ***SET LPT VFU*** command.

Attempting to command an advance to a channel that is not punched will cause a tape fault, and the printer will go offline. Setting the printer back online will clear the fault.

The current VFU definition may be displayed with the following command:

Command	***Action***
SHOW LPT VFU	Display the currently loaded VFU tape definition

This command displays the current VFU tape title and then the channel definitions for each form line. By default, a punched channel is indicated by an “O” character, and an unpunched channel is indicated by a period (“.”). These characters may be changed by depositing new values into the PUNCHR and UNPCHR registers, respectively.

4.9.4 Tracing and Registers

When a debug log has been established, tracing may be configured by specifying one or more of the reporting level options:

Option	*Reporting Level*
CMD	Printer commands executed
CSRW	Interface control, status, read, and write actions
SERV	Printer unit service events
XFER	Data transmissions
IOBUS	I/O bus signals and data words received and returned

The ***CMD*** option traces the commands executed by the printer. The ***CSRW*** option traces control, status, read, and write commands sent to the interface. The ***SERV*** option traces printer event service entries. The ***XFER*** option traces the data words and format commands written to the printer. The ***IOBUS*** option traces the I/O backplane signals and data received and returned via the interface.

Examples of the trace formats follow:

```
>>LPT   cmd: Printed 136 characters on line 1
>>LPT   cmd: Printer commanded to slew to VFU channel 3 from line 1
>>LPT   cmd: Printer advanced 1 line to line 2
>>LPT  csrw: Control is character | '#'
>>LPT  csrw: Status is online | ready | VFU 12 | idle
>>LPT  serv: Unit delay 1577 service scheduled
>>LPT  serv: Printer service entered
>>LPT  xfer: Character 'G' sent to printer
>>LPT  xfer: Format code 102 sent to printer
>>LPT iobus: Received data 000000 with signals STC | CLF | SIR | PRH
>>LPT iobus: Returned data 000000 with signals PRL
```

The Line Printer Controller state contains these registers:

Name	*Size*	*Radix*	*Symbolic*	*Read-Only*	*Description*
DEVCTL	1	2			Device control flip-flop
DEVFLG	1	2			Device flag flip-flop
DEVFBF	1	2			Device flag buffer flip-flop
STROBE	1	2			Strobe signal to the printer
DEMAND	1	2			Demand signal from the printer
OUTPUT	16	8	✓		Output data word to the printer
STATUS	16	8			Device status from the printer
PFAULT	1	2			Paper fault exists
TFAULT	1	2			Tape fault exists
OLPEND	1	2			Offline transition is pending
PRLINE	8	10			Current print line number
BUFIDX	8	10			Current print buffer index
PRTBUF [0:423]	8	8	✓		Print buffer
OVPCHR	8	8	✓		Overprint character
FORMLN	8	10		✓	Form length in lines
VFU [0:144]	12	2		✓	Vertical format unit channels
PUNCHR	8	8	✓		Punched channel character
UNPCHR	8	8	✓		Unpunched channel character
BTIME	24	10			Fast printer buffer load delay time
PTIME	24	10			Fast printing delay time
STIME	24	10			Fast paper slew per-line delay time
POS	32	10			Printer file current position

The PRTBUF, OVPCHR, PUNCHR, and UNPCHR registers default to single-character format display and entry but may be overridden with numeric-format switches, if desired. The OUTPUT register defaults to octal display but may be displayed in single-character format by specifying the ***–A*** switch. Symbolic entry for this register is also supported.

The PFAULT and TFAULT registers indicate a paper fault and a tape fault, respectively, when they contain the value 1. Attaching the printer will clear a paper fault. Setting the printer online will clear a tape fault.

The character used to represent an overprinted position may be changed by depositing a new value into the OVPCHR register. For example, the ***DEPOSIT LPT OVPCHR @*** command will change the character from the default DEL (octal 177) to the commercial-at sign (octal 100). Similarly, the characters used to represent punched and unpunched VFU channels in the ***SHOW LPT VFU*** display may be changed by depositing new values in the PUNCHR and UNPCHR registers, respectively.

The FORMLN register holds the current VFU form length, and the VFU register array holds the content of the current VFU tape. The register defaults to binary display, with channel 1 in the most-significant bit and channel 8 or 12 in the least-significant bit, depending on the VFU width. Elements 1-n correspond to VFU form lines 1-n, where a 1 value indicates a punch, and a 0 value indicates a no-punch. Element 0 is the logical OR of elements 1-n, so a 1 value indicates that the channel is punched somewhere on the tape, and a 0 value indicates that the channel is not punched.

4.10 12875A Processor Interconnect

The HP 12875A Processor Interconnect kit is used to communicate between the System Processor and the I/O Processor of a two-CPU HP 2000 Time-Shared BASIC system. The kit consists of four identical 12566A Microcircuit Interfaces and two interconnecting cables. One pair of interfaces is installed in adjacent I/O slots in each CPU, and the cables are used to connect the higher-priority (lower select code) interface in each computer to the lower-priority interface in the other computer. This interconnection provides a full-duplex 16-bit parallel communication channel between the processors. Each interface is actually a bi-directional, half-duplex line that is used in the primary direction for commands and in the reverse direction for status.

Two instances of the HP2100 simulator are run to simulate the SP and IOP. Each simulator contains an Inbound Data interface assigned to the lower-numbered select code, and an Outbound Data interface assigned to the higher-numbered select code. The IPLI and IPLO devices, respectively, simulate these interfaces, while the IPL device represents the combination. A shared memory area simulates the interconnecting cables.

During TSB startup, the SP transfers each word of the IOP program across the interconnection. When the IOP determines that the transfer is complete, it halts and stops responding to the SP. This lack of response is interpreted by the SP as meaning the transfer has finished successfully. In hardware, both processors run unimpeded, so this interpretation is correct. In simulation, however, the host operating system may block or preempt the IOP instance in response to system load, resulting in an incomplete transfer and subsequent TSB startup failure. The IPL device simulation provides two synchronization event mechanisms to ensure that the transfer is completed, regardless of host system load.

The IPLI and IPLO devices are initially disabled. A ***SET IPL ENABLED*** command will enable both devices. Attaching the IPL devices of one machine to the IPL devices of the other simulates connecting the cables. The System Processor instance establishes its connections with this command:

```
ATTACH –S {-E} IPL <code>
```

...where ***code*** is a user-selected decimal number between 1 and 65535 that uniquely identifies the instance pair to interconnect. The I/O Processor instance establishes its connections with this command:

```
ATTACH –I {-E} IPL <code>
```

...where ***code*** is the same decimal number specified in the System Processor's ***ATTACH*** command. If the host platform does not support shared memory, **Command not allowed** is printed. If the shared memory area is supported but cannot be created, **File open error** is printed. If the shared memory creation succeeds but the synchronization event creation fails, a **Command not completed** error occurs; in this case, adding the ***–E*** switch prints the message but substitutes timed pauses for synchronization events in ***WAIT*** commands.

The interface does not provide status to the CPU to indicate that the interconnecting cables are disconnected. Attempting to send or receive data with the cables not connected will cause the operation to hang, as it will in hardware. Specifying the ***SET CPU STOP=IOERR*** command will cause a simulation stop with a **Cable not connected** error if this condition occurs. If the error is corrected by reattaching the device, then resuming simulation will retry the operation.

Processor interconnect options that may be specified for the IPL device are:

Option	*Action*
INTERLOCK=<n>	Rendezvous with the other instance every ***n*** machine instructions
WAIT	Suspend execution until a signal is received from the other instance
SIGNAL	Signal the other instance to resume execution
SC=<n>	Set the interface card select codes; defaults are 32/33
ENABLED	Enable the IPLI and IPLO devices
DISABLED	Disable the IPLI and IPLO devices; default

The ***INTERLOCK*** option is used to synchronize the two simulator instances connected by ***ATTACH*** commands specifying the same code number. Entering a ***SET IPL INTERLOCK=<n>*** command permits each instance to execute the specified number of machine instructions before executing a rendezvous with the other instance. Whichever instance arrives at the rendezvous point first waits for the other instance to arrive. Once the rendezvous occurs, both instances resume, and the process repeats.

Executing an ***INTERLOCK*** command in one instance will synchronize both instances. However, there may be a short delay before the other instance begins synchronous execution unless it executes an ***ATTACH IPL***, ***RESET***, ***BOOT***, or ***RUN*** command after receiving the ***INTERLOCK*** command. Specifying a value of 0 cancels the rendezvous, and both instances are freed to execute asynchronously.

The ***WAIT*** and ***SIGNAL*** options are used to synchronize command files currently executing on two simulator instances. Executing a ***SET IPL WAIT*** command suspends the current simulator instance until the other instance executes a ***SET IPL SIGNAL*** command. If the signal is set before the wait is requested, the current simulator instance will not suspend but instead will continue immediately with the next command in the file.

An instance waiting for a rendezvous or executing a ***SET IPL WAIT*** command will wait forever for the corresponding rendezvous or signal. To abort the wait manually and return to the SCP prompt, enter the SCP interrupt character, which defaults to CTRL+E.

If a synchronization command is attempted before the IPL device has been attached, a **Unit not attached** error is printed. If the ***ATTACH*** command did not complete, a **Command not allowed** error is printed. If the ***–E*** switch was used, the ***WAIT*** command suspends for a fixed time rather than until the other instance signals, and the ***SIGNAL*** command has no effect.

When entering the ***SC*** option, specify the lower select code; the higher select code will be set automatically, and the cards will be assigned in the correct order.

Processor interconnect options that may be specified independently for either the IPLI or IPLO device are:

Option	***Action***
DIAGNOSTIC	Connect using the diagnostic test connector or cable
LINK	Connect using the processor interconnect cables
DEBUG=<option>	Enable tracing
NODEBUG	Disable tracing; default

Enabling diagnostic mode for either device individually simulates the installation of the HP 1251-0332 diagnostic test (loopback) connector in place of the processor interconnect cable. This is needed to run the General Purpose Register Diagnostic (DSN 143300) on that device. Enabling diagnostic mode for both devices concurrently simulates the connection of a processor interconnect cable between the two interfaces within the same machine. With this connection, the HP 2100 Processor Interconnect Cable Test (product number 24197) may be run. Entering the ***DIAGNOSTIC*** option automatically detaches any existing cable connections. Attempting to attach a device in diagnostic mode will be rejected with a **Command not allowed** error. Enabling link mode removes the diagnostic connector or cable and permits the processor interconnect cables to be installed with an ***ATTACH*** command.

Device configuration may be displayed with the following command:

Command	***Action***
SHOW IPL INTERLOCK	Display the interlock setting

When a debug log has been established, tracing of the Inbound and Outbound Data interfaces may be configured independently by specifying one or more of the reporting level options:

Option	*Reporting Level*
CMD	Command initiations and completions
CSRW	Control, status, read, and write actions
PSERV	Interface service scheduling calls and entries (periodic)
XFER	Data receptions and transmissions
IOBUS	I/O bus signals and data words received and returned

The ***CMD*** option traces the SP and IOP commands passed between the processors during Time-Shared BASIC operation. The ***CSRW*** option traces words received from and sent to the CPU. The ***PSERV*** option for the Inbound Data interface traces data event service scheduling and entries, which occur at a periodic rate dependent on incoming data availability. The ***PSERV*** option for the Outbound Data interface traces rendezvous service scheduling and entries, which occur at a periodic rate dependent on the ***INTERLOCK*** setting. The ***XFER*** option traces the words received and transmitted over the interconnecting cables. The ***IOBUS*** option traces the I/O backplane signals and data received and returned via the interface.

The Outbound Data interface supports one additional tracing option:

Option	*Reporting Level*
STATE	Process synchronizer state changes

This option traces calls to the host platform synchronization routines, as well as state changes in the interlock synchronizer as it regulates execution of the two instances.

Examples of the trace formats follow:

```
>>IPLO   cmd: Initialize IOP command
>>IPLO   cmd: Returned value is 1
>>IPLO   cmd: Process output string command port 0 count 8
>>IPLO   cmd: Returned status is Buffer not ready
>>IPLI   cmd: Line printer select code command device 14
>>IPLI   cmd: Delaying DMA completion interrupt for 1 msec
>>IPLI  csrw: Command input word is 040000
>>IPLI  csrw: Status output word is 000103
>>IPLO state: Created event 00000784 with identifier "/HP 2100-EVT-4020-1"
>>IPLO state: Event 00000780 wait timed out
>>IPLO state: Synchronizer transitioning after 1202 iterations with gate Unlocking
>>IPLO state: Synchronizer signaling the release event with gate Unlocking
>>IPLO pserv: Synchronizer delay 20 service entered with gate Unlocked
>>IPLO pserv: Synchronizer locked
>>IPLO pserv: Synchronizer unlocking with gate Locked and waiting
>>IPLI pserv: Poll delay 32 service scheduled
>>IPLI pserv: Poll delay 50 service entered
>>IPLO  xfer: Word 000015 sent to link
>>IPLO  xfer: Word 000040 delta 63 received from link
>>IPLO iobus: Received data 000000 with signals IOO | STC | CLC | CLF | EDT | SIR
>>IPLO iobus: Returned data 000000 with signals (none)
```

The interface supports the ***LOAD IPL*** and ***BOOT IPL*** commands to load an absolute binary program into memory from the other processor. Depending on the current CPU model, the ***LOAD*** command copies either the Basic Binary Loader for the I/O Processor (BBL-IOP) or the 12992K boot loader ROM into memory, configures it for the interface's select code, and sets the P register to the loader's starting address. The ***BOOT*** command performs these same actions and then presets the system and begins execution.

For a 2100/14/15/16 CPU containing a previously configured BBL-IOP in memory, the commands are equivalent in hardware to setting the P register to the starting address, pressing the LOADER ENABLE button on the front panel, and, for the ***BOOT*** command, pressing the INTERNAL PRESET, EXTERNAL PRESET, and RUN buttons to begin execution. The S register value is not relevant for this loader.

The BBL-IOP is a dual-purpose loader that also permits loading an absolute binary program from the paper tape reader. When the loader is copied into memory, the processor interconnect portion is configured for the select code of the Inbound Data interface, and the paper tape portion is configured for the select code of the paper tape reader interface. The P register is automatically set to the starting address of the processor interconnect portion, which starts at x7750 (i.e., offset 1750 within the highest page in memory). To execute the paper tape portion, the P register may be reset to x7700 (offset 1700).

For a 1000 M/E/F-Series CPU, the commands are equivalent in hardware to setting the S register to select and configure the HP 12992K boot loader ROM, pressing the IBL button to copy the ROM into memory, and, for the ***BOOT*** command, pressing the PRESET and RUN buttons to begin execution. The S register is preset as follows:

15	***14***	***13***	***12***	***11***	***10***	***9***	***8***	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
0	0	0	0	IPLI select code						0	0	0	0	0	0

For either loader, execution ends with one of the following halt instructions:

- HLT 11 — a checksum error occurred; A = the calculated value, B = the tape value.
- HLT 55 — the program load address would overlay the loader.
- HLT 77 — the end of the program was reached with a successful read.

The IPLI and IPLO state contains these registers:

Name	***Size***	***Radix***	***Action***
IBUF	16	8	Input buffer
OBUF	16	8	Output buffer
CTL	1	2	Control flip-flop
FLG	1	2	Flag flip-flop
FBF	1	2	Flag buffer flip-flop

The IPLI state contains two additional registers:

Name	***Size***	***Radix***	***Action***
TIME	24	10	Maximum polling interval for input
WAIT	24	10	Pause time in seconds

Input polling is adaptive — it speeds up to improve performance when data transfers are occurring regularly and slows down to reduce simulator overhead when transfers stop. The ***TIME*** value determines the maximum time between checks for input transfer resumption. The ***WAIT*** value specifies the number of seconds to pause when using timed suspensions instead of synchronization events.

4.11 12920A Asynchronous Multiplexer Interface

The HP 12920A Asynchronous Multiplexer Interface is a 16-channel terminal multiplexer commonly used with the HP 2000 series of Time-Shared BASIC systems. It connects from 1 to 16 serial terminals or modems at programmable baud rates from 75 to 2400 bits per second. Character sizes are also programmable from 5 to 12

bits in length, including the start and stop bits. Each channel can be independently configured for separate send and receive rates. The multiplexer is not buffered, so the CPU has to retrieve each character from a given channel before the next character arrives. To avoid saturating the CPU with interrupt requests, the multiplexer maintains an internal "mini-interrupt" system that queues requests and holds additional interrupts off until the CPU acknowledges the current request.

The 12920A consists of two Data Printed Circuit Assemblies — a Lower Data PCA and an Upper Data PCA — and an optional Control PCA. In simulation, the multiplexer is implemented as a three-device set, as follows:

- MUXL — the Lower Data PCA
- MUX — the Upper Data PCA
- MUXC — the Control PCA

The Data PCAs provide the serial data line connections for terminals and data sets. Five additional receive-only auxiliary channels may be connected as a group under software control to one of the sixteen main channels to detect the incoming baud rate. The Control PCA provides serial control and status lines for Bell 103 data sets.

The Lower and Upper Data PCAs must be in adjacent I/O slots. The Control PCA may be placed in any slot, although in practice it was placed in the slot above the Upper Data PCA, so that all three PCAs were physically together.

The 12920A supported one or two Control PCAs (two were used with 801-type automatic dialers). Under simulation, only one Control PCA is supported.

4.11.1 Lower and Upper Data PCAs

The Data PCAs provide the Transmitted Data (BA) and Received Data (BB) lines for up to sixteen terminals. They perform input and output through Telnet sessions connected to a user-specified listening port or through individually specified host serial ports. The Data PCAs support concurrent Telnet and serial connections. The ***ATTACH*** command specifies the local port to be used for Telnet connections:

```
ATTACH MUX <port>
```

...where ***port*** is a decimal number between 1 and 65535 that is not being used for other TCP/IP activities. When the Upper Data PCA is attached and the simulator is running, the multiplexer listens for connections on the specified port and assigns them to channels in ascending numeric order, unless an alternate connection order has been specified.

The ***ATTACH*** command is also used to specify the host serial port for an individual multiplexer channel:

```
ATTACH MUXL<chan> <port-name>{;<rate>-<size><parity><stopbits>}
```

...where ***chan*** is the multiplexer channel number from 0-15, and ***port-name*** is the host name of the serial port to use (e.g., /dev/*ttyS0* or *COM1*).

An optional serial port configuration string may be supplied after the host name. The required values are:

- ***rate*** is the baud rate in bits per second.
- ***size*** is the character size in bits including the parity bit, if designated.
- ***parity*** designates the parity to use: *N* (no), *E* (even), *O* (odd), *M* (mark), or *S* (space).
- ***stopbits*** is the number of stop bits (*1*, *1.5*, or *2*).

If the port configuration string is omitted, the default configuration specified by the host system for that port is used.

Upper and Lower Data PCA configuration options are available for the device and for the individual units. The command forms are:

```
SET MUX <device-option>
SET MUXL <device-option>
SET MUXLn <unit-option>
```

Device options that may be specified for the Upper Data PCA are:

Option	*Action*
TERMINAL	Connect using Telnet or serial ports; default
DIAGNOSTIC	Connect using diagnostic test cables
LINEORDER=<c1>[;<c2>...]	Set the channel connection order
SC=<n>	Set the interface card select codes; default is 40/41
DEBUG=<option>	Enable tracing
NODEBUG	Disable tracing; default
ENABLED	Enable the device; default
DISABLED	Disable the device

Enabling the ***DIAGNOSTIC*** mode simulates the installation of eight HP 30062-60003 diagnostic test (loopback) cables between channels 0-1, 2-3, etc., as required by the multiplexer diagnostic. Each cable connects the Transmitted Data, Data Terminal Ready, and Request to Send lines, respectively, of one channel to the Received Data, Data Set Ready, and Carrier Detect lines of the other channel, and vice versa. In addition, all sessions are disconnected, and the multiplexer is detached from the Telnet listening port. While in diagnostic mode, the ***ATTACH MUX*** command is not allowed. Enabling ***TERMINAL*** mode allows the multiplexer to be attached to accept incoming connections again.

The ***LINEORDER*** option specifies the order in which connections are assigned to multiplexer channels. The arguments may be single channel numbers or ranges of channel numbers of the form ***m-n***, with multiple arguments separated by semicolons. Telnet connections to the listening port will be assigned to multiplexer channels in the sequence specified. Omitted channels will not receive connections, unless the ***ALL*** keyword is supplied as the last argument. In the absence of a ***SET MUX LINEORDER*** command, connections will be assigned by default in ascending channel order. The default order may be reestablished by specifying the command ***SET MUX LINEORDER=ALL***.

When entering the ***SC*** option, specify the lower select code with either device name; the higher select code will be set automatically, and the cards will be assigned in the correct order.

Device options that may be specified for the Lower Data PCA are:

Option	*Action*
SC=<n>	Set the interface card select codes; default is 40/41
DEBUG=<option>	Enable tracing
NODEBUG	Disable tracing; default
ENABLED	Enable the device; default
DISABLED	Disable the device

Unit options that may be specified for individual Lower Data PCA channels are:

Option	*Action*
UC	Upshift lowercase output characters to uppercase; default
7B	Output with high-order bit cleared
7P	Output with high-order bit cleared, non-printing suppressed
8B	Output characters without changing
DATASET	Enable modem control via the Control PCA
NODATASET	Disable modem control; default
LOG=<filename>	Enable I/O logging
NOLOG	Disable I/O logging; default
DISCONNECT	Disconnect the channel

Each channel may be set to one of four output modes (***UC***, ***7B***, ***7P***, or ***8B***). The default mode is ***UC*** for all channels. If the multiplexer is used with RTE driver DVS00 and logical driver LDV05, or DOS driver DVR73 and logical drivers PMT01 or PMT02, the mode must be changed to ***7B*** or ***8B*** to allow ESC, DC1, and ENQ characters to pass through to the terminal. Alternatively, the ***SET CONSOLE PCHAR*** SCP command may be used to redefine the set of printable characters. Modes ***UC*** and ***7P*** are compatible with DVS00 and DVR73 using the default logical driver.

The ***DATASET*** option enables modem control via the corresponding channel on the Control PCA. When modem control is enabled, a connected channel that has Data Terminal Ready (command bit C1) high will set Carrier Detect (status bit S2) in the line status, and a programmed drop of the DTR line will disconnect the Telnet session. Setting the ***NODATASET*** option causes DTR transitions to be ignored.

Each channel supports independent I/O logging to a file. Adding the ***–N*** (new file) switch clears the contents of the log file if it is present. Without the ***–N*** switch, channel output will be appended to any preexisting log file content. Disabling logging also closes the log file.

A channel may be manually disconnected from its associated Telnet session with the ***SET MUXLn DISCONNECT*** command. Otherwise, the connection will remain open until disconnected either by the Telnet client or a ***DETACH MUX*** command, unless the channel is controlled by the Control PCA. For a serial connection, the ***SET MUXLn DISCONNECT*** command will drop and then raise the Data Terminal Ready line; to disconnect the serial port, issue a ***DETACH MUXLn*** command.

Upper Data PCA device configuration may be displayed with the following commands:

Command	*Action*
SHOW MUX	Display the device configuration
SHOW MUX LINEORDER	Display the channel connection order
SHOW MUX CONNECTIONS	Display the channel connections
SHOW MUX STATISTICS	Display the channel I/O statistics for the active connections

Lower Data PCA unit configuration may be displayed with the following commands:

Command	*Action*
SHOW MUXL<n>	Display the selected channel's configuration
SHOW MUXL<n> LOG	Display the selected channel's logging status

In addition to the current configuration settings, a channel controlled by the Control PCA will be marked **data set** in the listing; a channel not controlled will be marked **direct**.

When a debug log has been established, tracing of the Upper Data PCA device may be configured by specifying one or more of these reporting level options:

Option	*Reporting Level*
CSRW	Control, status, read, and write actions
PSERV	Poll unit service events (periodic)
IOBUS	I/O bus signals and data words received and returned

Tracing of the Lower Data PCA device may be configured by specifying these options:

Option	*Reporting Level*
CSRW	Control, status, read, and write actions
SERV	Channel unit service events
XFER	Data receptions and transmissions
IOBUS	I/O bus signals and data words received and returned

The ***CSRW*** option traces control, status, read, and write commands sent to the interface. The ***PSERV*** option traces poll unit service scheduling and entries. The poll service is entered once every 10 milliseconds to poll for new Telnet or serial connections and for input characters to be received by the active channels. The ***SERV*** option traces channel output service scheduling and entries. The ***XFER*** option traces the words received and transmitted via the multiplexer channels. The ***IOBUS*** option traces the I/O backplane signals and data received and returned via the interface.

Examples of the trace formats follow:

```
>>MUX   csrw: Channel 0 is selected
>>MUX   csrw: Status is channel 0 | receive
>>MUXL  csrw: Parameter is send | enable parity/echo | 11 bits | 110 baud
>>MUXL  csrw: Channel 0 receive parameter 131202 stored
>>MUXL  csrw: Channel 0 send interrupt requested
>>MUX  pserv: Poll delay 15773 service entered
>>MUXL  serv: Channel 0 delay 500 service scheduled
>>MUXL  serv: Channel 0 service entered
>>MUXL  xfer: Channel 0 character 'R' sent
>>MUXL  xfer: Channel 0 character LF sent
>>MUXL  xfer: Channel 0 character 'H' received
>>MUXL  xfer: Channel 2 break detected
>>MUXL iobus: Received data 114015 with signals SFS | PRH
>>MUXL iobus: Returned data 114015 with signals SKF | PRL
>>MUX  iobus: Received data 000000 with signals IOI
>>MUX  iobus: Returned data 000001 with signals (none)
```

The Upper Data PCA state contains these registers:

Name	*Size*	*Radix*	*Action*
IBUF	16	8	Input buffer; holds channel status
OBUF	16	8	Output buffer; hold channel select

The Lower Data PCA state contains these registers:

Name	*Size*	*Radix*	*Symbolic*	*Action*
CTL	1	2		Control flip-flop
FLG	1	2		Flag flip-flop
FBF	1	2		Flag buffer flip-flop
STA [0:20]	16	8		Receive channel status, channels 0-20
RPAR [0:20]	16	8		Receive channel parameters, channels 0-20
XPAR [0:15]	16	8		Transmit channel parameters, channels 0-15
RBUF [0:20]	16	8	✓	Receive channel buffers, channels 0-20
XBUF [0:15]	16	8	✓	Transmit channel buffers, channels 0-15
RCHP [0:20]	1	2		Receive channel character present, channels 0-20
XDON [0:15]	1	2		Transmit channel done, channels 0-15
BDFR [0:15]	1	2		Receive channel break deferred, channels 0-15
TIME [0:15]	24	10		Transmit channel time, channels 0-15

The RBUF and XBUF registers default to single-character format display and entry but may be overridden with numeric-format switches, if desired.

4.11.2 Control PCA

The Control PCA provides the Request to Send (CA) and Data Terminal Ready (CD) control lines, and the Data Set Ready (CC) and Carrier Detect (CF) status lines for each of sixteen terminals or data sets. The modem controls model the Bell 103A data set without ring detection.

A new Telnet connection sets Data Set Ready and is established whether Data Terminal Ready is set or not.

Control PCA device options that may be specified are:

Option	*Action*
SC=<n>	Set the interface card select code; default is 42

The Control PCA is usually installed in the slot above the Upper Data PCA, but this is not required.

There are no Control PCA units or unit commands.

Control PCA device configuration may be displayed with the following commands:

Command	*Action*
SHOW MUXC	Display the device configuration

When a debug log has been established, tracing may be configured by specifying one or more of the reporting level options:

Option	*Reporting Level*
CSRW	Control, status, read, and write actions
XFER	Control and status line changes
IOBUS	I/O bus signals and data words received and returned

The CSRW option traces control, status, read, and write commands sent to the interface. The XFER option traces the changes in control and status lines by the active channels. The ***IOBUS*** option traces the I/O backplane signals and data received and returned via the interface.

Examples of the trace formats follow:

```
>>MUXC  csrw: Status is channel 0 | I2 | I1 | ES2 | ES1 | ~S2 | ~S1
>>MUXC  csrw: Control is channel 0 | update | ~C2 | ~C1 | ES2 | ES1 | S2 | S1
>>MUXC  csrw: Channel 1 disconnected by DTR drop
>>MUXC  csrw: Channel 1 interrupt requested
>>MUXC  xfer: Channel 0 connected
>>MUXC  xfer: Channel 0 line status is RTS | DTR
>>MUXC  xfer: Channel 0 disconnect dropped DCD and DSR
>>MUXC iobus: Received data 031374 with signals IOO
>>MUXC iobus: Returned data 141414 with signals (none)
```

The Control PCA state contains these registers:

Name	*Size*	*Radix*	*Description*
CTL	1	2	Control flip-flop
FLG	1	2	Flag flip-flop
FBF	1	2	Flag buffer flip-flop
SCAN	1	2	Scan flip-flop
CHAN	4	8	Current channel number
DSO [0:15]	6	2	Control lines C2, C1, ES2, ES1, SS2, and SS1, channels 0-15
DSI [0:15]	2	2	Status lines S2 and S1, channels 0-15

4.12 12966A Buffered Asynchronous Communications Interface

The 12966A Buffered Asynchronous Communications Interface card supplanted the 12531C Teletype and 12880A CRT interfaces as the primary terminal connection for HP computer systems. The main advantage of this card over the others was its 128-character FIFO memory. While this allowed more efficient I/O than its interrupt-per-character predecessors, the most significant improvement was that block input from the 264x-series of CRT terminals was supported. The 264x were the first HP-supported terminals to provide local editing and character storage, as well as mass storage via dual DC-100 minicartridge drives. This support meant that input from the terminal could come in bursts at the full baud rate, which would overrun the older cards that needed a small intercharacter handling time. The older cards also placed a substantial load on the CPU in high-baud-rate output applications.

The BACI simulator performs input and output through a Telnet session connected to a user-specified listening port or through a specified host serial port. The ***ATTACH*** command specifies the local port to be used for Telnet connections:

```
ATTACH BACI <port>
```

...where ***port*** is a decimal number between 1 and 65535 that is not being used for other TCP/IP activities. When the BACI is attached and the simulator is running, the BACI listens for connections on the specified port. Only one connection at a time is permitted.

The ***ATTACH*** command is also used to specify a host port for serial connections:

```
ATTACH BACI <port-name>{;<rate>-<size><parity><stopbits>}
```

...where ***port-name*** is the host name of the serial port to use (e.g., /dev/*ttyS0* or *COM1*).

An optional serial port configuration string may be supplied after the host name. The required values are:

- ***rate*** is the baud rate in bits per second.
- ***size*** is the character size in bits including the parity bit, if designated.
- ***parity*** designates the parity to use: *N* (no), *E* (even), *O* (odd), *M* (mark), or *S* (space).
- ***stopbits*** is the number of stop bits (*1*, *1.5*, or *2*).

If the port configuration string is omitted, the default configuration specified by the host system for that port is used.

While the BACI can connect to generic clients, RTE drivers for the BACI require the use of HP terminals or terminal emulators.

Device and unit configuration options specify timing and output modes. The command forms are:

```
SET BACI <device-option>
SET BACI <unit-option>
```

Device options that may be specified are:

Option	*Action*
DIAGNOSTIC	Connect using a diagnostic test connector
TERMINAL	Connect using Telnet or a serial port; default
FASTTIME	Use optimized timing; default
REALTIME	Use realistic timing
SC=<n>	Set the interface card select code; default is 35
DEBUG=<option>	Enable tracing
NODEBUG	Disable tracing; default
ENABLED	Enable the device; default
DISABLED	Disable the device

Enabling ***DIAGNOSTIC*** mode simulates the installation of the HP 12966-60003 diagnostic test (loopback) connector onto the 12966A card. In addition, any current session is disconnected, and the BACI is detached from the Telnet listening port. While in diagnostic mode, the ***ATTACH BACI*** command is not allowed. Enabling ***TERMINAL*** mode simulates the installation of the HP 12966-60008 (264x terminal) connector and RS-232 cable and allows the BACI to be attached to accept an incoming connection again.

The BACI supports fifteen programmable baud rates from 50 to 9600 baud, as well as an external rate determined by the connected terminal. With the ***REALTIME*** option, the simulation accurately models the transmission and reception baud rates (in machine instructions). For example, setting the interface to 1200 baud will take twice as long to output a given listing as setting the interface to 2400 baud. Enabling optimized timing with the ***FASTTIME*** option reduces the timing to the minimum necessary to operate correctly; this is much faster than the real BACI would operate.

Setting the BACI to fast timing also enables three optimizations. First, buffered characters are transferred via Telnet in blocks, rather than a character at a time. Second, ENQ/ACK handshaking is done locally, without involving the Telnet client. Third, reception of characters entered during output operations is delayed until after the first ENQ/ACK handshake at the end of the line; this significantly improves break-mode response under RTE.

Unit options that may be specified are:

Option	*Action*
CAPSLOCK	Upshift lowercase input characters to uppercase
NOCAPSLOCK	Input characters are unchanged; default
LOG=<filename>	Enable I/O logging
NOLOG	Disable I/O logging; default
DISCONNECT	Disconnect the line

Many HP operating systems require command input in upper case, although mixed-case output is supported. As an aid to avoid toggling the host keyboard in and out of CAPS LOCK mode, the BACI provides this function locally. The default mode is ***NOCAPSLOCK***.

The BACI supports I/O logging to a file. Adding the ***–N*** (new file) switch clears the contents of the log file if it is present. Without the ***–N*** switch, BACI output will be appended to any preexisting log file content. Disabling logging also closes the log file.

The BACI may be manually disconnected from its associated Telnet session with the ***SET BACI DISCONNECT*** command. Otherwise, the connection will remain open until disconnected either by the Telnet client or a ***DETACH BACI*** command. For a serial connection, the ***SET BACI DISCONNECT*** command will drop and then raise the Data Terminal Ready line; to disconnect the serial port, issue a ***DETACH BACI*** command.

Device configuration may be displayed with the following commands:

Command	*Action*
SHOW BACI	Display the device and unit configuration
SHOW BACI CONNECTION	Display the active connection status
SHOW BACI STATISTICS	Display the I/O statistics for the active connection

When a debug log has been established, tracing may be configured by specifying one or more of the reporting level options:

Option	*Reporting Level*
CMDS	Command initiations and completions
CPU	Words received from and sent to the CPU
BUF	Data read from and written to the FIFO
XFER	Data receptions and transmissions
IOBUS	I/O bus signals and data words received and returned

The ***CMDS*** option traces the commands executed by the interface. The ***CPU*** option traces words received from (control and write data) and sent to (status and read data) the CPU. The ***BUF*** option traces gets from and puts to the internal FIFO buffers. The ***XFER*** option traces the words received and transmitted via the Telnet or serial connection. The ***IOBUS*** option traces the I/O backplane signals and data received and returned via the interface.

Examples of the trace formats follow:

```
>>BACI  cmds: [STF] Flag, SRQ, and lockout set
>>BACI   cpu: [Lix,C] Received data = 102042
>>BACI   buf: Character LF put to FIFO [29], character counter = 53
>>BACI  xfer: Character 'R' transmitted from the UART
>>BACI iobus: Received data 000000 with signals STF | SIR
>>BACI iobus: Returned data 000000 with signals (none)
```

The BACI state contains these registers:

Name	*Size*	*Radix*	*Symbolic*	*Action*
IBUF	16	8	✓	Input buffer; holds line status
OBUF	16	8	✓	Output buffer; hold channel select
STATUS	16	2		Card status word
EDSIW	16	8		Enable device status interrupt word
DSRW	16	8		Device status reference word
CFCW	16	8		Character frame control word
ICW	16	8		Interface control word
ISRW	16	8		Interrupt status reset word
FIFOPUT	8	10		FIFO load index
FIFOGET	8	10		FIFO unload index
FIFOCNTR	8	10		FIFO character counter
BRKCNTR	16	10		BREAK clock counter
FIFO [0:127]	8	8	✓	FIFO buffer
SPCHAR [0:255]	1	2		Special character RAM
UARTTHR	16	8	✓	UART transmitter holding register
UARTTR	16	8	✓	UART transmitter register
UARTRHR	16	8	✓	UART receiver holding register
UARTRR	16	8	✓	UART receiver register
UARTCLK	16	10		UART transmitter/receiver clock counter
CTIME	19	10		Inter-character poll time
LKO	1	2		Interface interrupt lockout flip-flop
CTL	1	2		Interface control flip-flop
FLG	1	2		Interface flag flip-flop
FBF	1	2		Interface flag buffer flip-flop
SRQ	1	2		Interface DMA service request flip-flop

The FIFO register defaults to single-character format display and entry but may be overridden with a numeric-format switch, if desired. The other symbolic registers default to octal display but may be displayed in single-character format by specifying the ***–A*** switch. Symbolic entry for these registers is also supported

5 Disc I/O Device Simulations

The simulator provides these disc interfaces and their associated peripherals:

- 12557A Cartridge Disc Interface
- 13210A Disc Drive Interface
- 12565A Disc Interface
- 12606B Disc Memory Interface
- 12610B Drum Memory Interface
- 12821A Disc Interface
- 13175D Disc Controller Interface

5.1 12557A Cartridge Disc Interface with Four 2870 Drives, 13210A Disc Drive Interface with Four 7900 Drives

The 12557A Cartridge Disc Interface connects from one to four HP 2870A (2.5 MB) disc drives to the HP computer family. The 13210A Disc Drive Interface connects from one to four HP 7900A (5.0 MB) disc drives. Both interfaces consist of two Printed Circuit Assemblies installed in adjacent I/O slots. The Data PCA is assigned the higher-priority (lower numbered) select code, and the Control PCA is assigned the lower-priority select code. The DPD and DPC devices simulate these PCAs.

Attaching a disc image file to a unit simulates inserting a disc pack into a drive:

```
ATTACH {-R | -N} DPCn <image-filename>
```

Adding the ***–R*** (read-only) switch is equivalent to setting the drive's Upper and Lower Data Protect switches to the *On* position. Adding the ***–N*** (new file) switch creates a full-size image file, equivalent to formatting the new disc before use.

If the host operating system returns an error when reading or writing a disc image file, the simulator will report the error to the simulation console, e.g.:

```
HP 2100 simulator DP disc I/O error: No space left on device
```

...and the simulation will stop with an **I/O Error** message. No error recovery is attempted.

A drive's unit number is not set explicitly. Instead, the drive unit number is derived from the simulation unit number. For example, unit DPC0 responds to drive select number 0. Changing the drive select jumpers on a mounted drive is simulated by detaching and reattaching the disc image file to the corresponding simulation unit.

Device and unit options include configuring the interface and drive type, protection and format status, and the ability to set drives ready or not-ready. The command forms are:

```
SET DPC <device-option>
SET DPCn <unit-option>
```

Device options that may be specified are:

Option	*Action*
12557A	Use the 12557A interface and 2870A drives
13210A	Use the 13210A interface and 7900A drives; default
SC=<n>	Set the interface card select codes; default is 22/23
ENABLED	Enable the device; default
DISABLED	Disable the device

The ***12557A*** and ***13210A*** options set the interface type and, by implication, the disc drive type.

When entering the ***SC*** option, specify the lower select code with either device name; the higher select code will be set automatically, and the cards will be assigned in the correct order.

Device configuration may be displayed with the following commands:

Command	*Action*
SHOW DPC	Display the device and unit configuration
SHOW DPC TYPE	Display the interface type

Unit options that may be specified for the individual disc drives are:

Option	*Action*
UNLOAD	Set the drive's Run/Stop switch to *Stop*; default when detached
LOAD	Set the drive's Run/Stop switch to *Run*; default when attached
PROTECT	Set the drive's Data Protect switches to *On*
UNPROTECT	Set the drive's Data Protect switches to *Off*; default
ENABLED	Enable the unit; default
DISABLED	Disable the unit

The ***UNLOAD*** and ***LOAD*** options unload and load the drive's heads from the disc pack, setting the drive not-ready and ready, respectively. ***UNLOAD*** provides a convenient method of setting a drive "down" without detaching the associated disc image file.

The ***PROTECT*** option prevents writing to the disc. It is the hardware equivalent of setting the Upper and Lower Data Protect switches to the *On* position. The ***UNPROTECT*** option permits writing and is equivalent to setting the switches to *Off*. Separate protection for the upper and lower platters of the 7900 drive is not supported. Also, the drive Protect/Override switch is not supported; drive protection is permanently overridden.

The interface supports the ***LOAD {–R} DPC*** and ***BOOT{–R} DPC*** commands to load a disc boot extension into memory. Depending on the current CPU model, the ***LOAD*** command copies either the Basic Moving-Head Disc Loader (BMDL-7900) or the 12992F boot loader ROM into memory, configures it for the Data interface's select code, and sets the P register to the loader's starting address. The ***BOOT*** command performs these same actions and then presets the system and begins execution. For either command, if the ***–R*** switch is specified, the boot extension will be read from the removable platter. If ***–R*** is not specified, then the boot extension will be read from the fixed platter.

For a 2100/14/15/16 CPU containing a previously configured BMDL in memory, the commands are equivalent in hardware to setting the P register to the starting address, pressing the LOADER ENABLE button on the front panel, and, for the ***BOOT*** command, pressing the INTERNAL PRESET, EXTERNAL PRESET, and RUN buttons to begin execution. The S register value is not relevant for this loader.

The BMDL is a dual-purpose loader that also permits loading an absolute binary program from the paper tape reader. When the loader is copied into memory, the disc portion is configured for the select code of the Data interface, and the paper tape portion is configured for the select code of the paper tape reader interface. The P register is automatically set to the starting address of the disc portion, which starts at x7750 (i.e., offset 1750 within the highest page in memory). The loader reads 6144 words (for the 7900A) or 3027 words (for the 2870A) from cylinder 0 sector 0 of the specified subchannel into memory starting at location 2011 octal. Loader execution ends with the following instruction:

- JSB 2055,I — the disc read completed.

Normal completion status is End of Cylinder and Any Error (000041 octal).

To execute the paper tape portion, the P register may be reset to x7700 (offset 1700). Loader execution ends with one of the following halt instructions:

- HLT 11 — a checksum error occurred; A = the tape value, B = the calculated value.
- HLT 55 — the program load address would overlay the loader.
- HLT 77 — the end of tape was reached with a successful read.

For a 1000 M/E/F-Series CPU, the commands are equivalent in hardware to setting the S register to select and configure the HP 12992F boot loader ROM, pressing the IBL button to copy the ROM into memory, and, for the ***BOOT*** command, pressing the PRESET and RUN buttons to begin execution. The S register is preset as follows:

15	***14***	***13***	***12***	***11***	***10***	***9***	***8***	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
0	0	0	0	DPD select code						unchanged			0	0	●

If the ***–R*** switch is specified, bit 0 is set to 1 to boot from the removable platter. If ***–R*** is not specified, then bit 0 is set to 0 to boot from the fixed platter. Bits 5-3 of the existing S-register value are retained when the loader is executed and are available to the boot extension program.

The loader reads 6144 words from cylinder 0 sector 0 of the specified subchannel into memory starting at location 2011 octal. Loader execution ends with one of the following instructions:

- HLT 30 — a drive fault occurred; enter the ***GO*** command to retry.
- JSB 2055,I — the disc read succeeded.

When a debug log has been established, tracing may be configured for each card separately by specifying the reporting level option:

Option	***Reporting Level***
IOBUS	I/O bus signals and data words received and returned

The ***IOBUS*** option traces the I/O backplane signals and data received and returned by the interface.

The Data PCA state contains these registers:

Name	*Size*	*Radix*	*Action*
IBUF	16	8	Input buffer
OBUF	16	8	Output buffer
DBUF [0:127]	16	8	Sector buffer
BPTR	7	10	Sector buffer pointer
CMD	1	2	Command flip-flop
CTL	1	2	Control flip-flop
FLG	1	2	Flag flip-flop
FBF	1	2	Flag buffer flip-flop
XFER	1	2	Transfer in progress flag
WVAL	1	2	Write data valid flag

The Control PCA state contains these registers:

Name	*Size*	*Radix*	*Read-Only*	*Action*
OBUF	16	8		Output buffer
BUSY	4	8	✓	Busy unit number + 1
CNT	5	8		Check record counter
CMD	1	2		Command flip-flop
CTL	1	2		Control flip-flop
FLG	1	2		Flag flip-flop
FBF	1	2		Flag buffer flip-flop
EOC	1	2		End of cylinder pending
POLL	1	2		Attention polling enabled
RARC	8	10		Record address register (cylinder)
RARH	2	10		Record address register (head)
RARS	4	10		Record address register (sector)
CYL [0:3]	8	10		Current cylinder, drives 0-3
STA [0:3]	16	8		Drive status, drives 0-3
CTIME	24	10		Data transfer command delay time
DTIME	24	10		Data channel command delay time
STIME	24	10		Seek delay time, per cylinder
XTIME	24	10		Inter-word transfer time

5.2 12565A Disc Interface with Two 2883 Drives

The 12565A Disc Interface connects one or two HP 2883A (23 MB) disc drives to the HP computer family. The interface consists of two Printed Circuit Assemblies installed in adjacent I/O slots. The Data PCA is assigned the higher-priority (lower numbered) select code, and the Control PCA is assigned the lower-priority select code. The DQD and DQC devices simulate these PCAs.

Attaching a disc image file to a unit simulates inserting a disc pack into a drive:

```
ATTACH {-R | -N} DQCn <image-filename>
```

Adding the ***–R*** (read-only) switch prevents writing to the disc image. Adding the ***–N*** (new file) switch creates a full-size image file, equivalent to formatting the new disc before use.

If the host operating system returns an error when reading or writing a disc image file, the simulator will report the error to the simulation console, e.g.:

```
HP 2100 simulator DQ disc I/O error: No space left on device
```

...and the simulation will stop with an **I/O Error** message. No error recovery is attempted.

A drive's unit number is not set explicitly. Instead, the drive unit number is derived from the simulation unit number. For example, unit DQC0 responds to drive select number 0. Changing the drive select jumpers on a mounted drive is simulated by detaching and reattaching the disc image file to the corresponding simulation unit.

Device and unit options include configuring the protection and drive-ready status. The command forms are:

```
SET DQC <device-option>
SET DQCn <unit-option>
```

Device options that may be specified are:

Option	*Action*
SC=<n>	Set the interface card select codes; default is 24/25
ENABLED	Enable the device; default
DISABLED	Disable the device

When entering the ***SC*** option, specify the lower select code with either device name; the higher select code will be set automatically, and the cards will be assigned in the correct order.

Device configuration may be displayed with the following command:

Command	*Action*
SHOW DQC	Display the device and unit configuration

Unit options that may be specified for individual disc drives are:

Option	*Action*
UNLOAD	Set the drive's Run/Stop switch to *Stop*; default when detached
LOAD	Set the drive's Run/Stop switch to *Run*; default when attached
PROTECT	Prohibit writing to the drive
UNPROTECT	Permit writing to the drive; default
ENABLED	Enable the unit; default
DISABLED	Disable the unit

The ***UNLOAD*** and ***LOAD*** options unload and load the drive's heads from the disc pack, setting the drive not-ready and ready, respectively. ***UNLOAD*** provides a convenient method of setting a drive "down" without detaching the associated disc image file.

The ***PROTECT*** option prevents writing to the disc, and the ***UNPROTECT*** option permits writing. These options have no hardware analog; writing is always permitted on the 2883A. Also, the drive Format switch is not supported; drive address protection is permanently overridden.

The interface supports the ***LOAD DQC*** and ***BOOT DQC*** commands to load a disc boot extension into memory. Depending on the current CPU model, the ***LOAD*** command copies either the Basic Moving-Head Disc Loader (BMDL-2883) or the 12992A boot loader ROM into memory, configures it for the Data interface's select code, and

sets the P register to the loader's starting address. The ***BOOT*** command performs these same actions and then presets the system and begins execution.

For a 2100/14/15/16 CPU containing a previously configured BMDL in memory, the commands are equivalent in hardware to setting the P register to the starting address, pressing the LOADER ENABLE button on the front panel, and, for the ***BOOT*** command, pressing the INTERNAL PRESET, EXTERNAL PRESET, and RUN buttons to begin execution. The S register value is not relevant for this loader.

The BMDL is a dual-purpose loader that also permits loading an absolute binary program from the paper tape reader. When the loader is copied into memory, the disc portion is configured for the select code of the Data interface, and the paper tape portion is configured for the select code of the paper tape reader interface. The P register is automatically set to the starting address of the disc portion, which starts at x7750 (i.e., offset 1750 within the highest page in memory). The loader reads 128 words from cylinder 0 head 0 sector 0 into memory starting at location 2011 octal. Loader execution ends with the following instruction:

- JMP 2055,I — the disc read completed.

To execute the paper tape portion, the P register may be reset to x7700 (offset 1700). Loader execution ends with one of the following halt instructions:

- HLT 11 — a checksum error occurred; A = the tape value, B = the calculated value.
- HLT 55 — the program load address would overlay the loader.
- HLT 77 — the end of tape was reached with a successful read.

For a 1000 M/E/F-Series CPU, the commands are equivalent in hardware to setting the S register to select and configure the HP 12992A boot loader ROM, pressing the IBL button to copy the ROM into memory, and, for the ***BOOT*** command, pressing the PRESET and RUN buttons to begin execution. The S register is preset as follows:

15	***14***	***13***	***12***	***11***	***10***	***9***	***8***	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
0	0	1	0	DQD select code						unchanged			0	0	0

Bits 5-3 of the existing S-register value are retained when the loader is executed and are available to the boot extension program.

The loader reads 128 words from cylinder 0 head 0 sector 0 into memory starting at location 2011 octal. Loader execution ends with the following instruction:

- JSB 2055,I — the disc read completed.

The success or failure of the transfer is not checked.

When a debug log has been established, tracing may be configured for each card separately by specifying the reporting level option:

Option	***Reporting Level***
IOBUS	I/O bus signals and data words received and returned

The ***IOBUS*** option traces the I/O backplane signals and data received and returned by the interface.

The Data PCA state contains these registers:

Name	*Size*	*Radix*	*Action*
IBUF	16	8	Input buffer
OBUF	16	8	Output buffer
DBUF [0:127]	16	8	Sector buffer
BPTR	7	10	Sector buffer pointer
CMD	1	2	Command flip-flop
CTL	1	2	Control flip-flop
FLG	1	2	Flag flip-flop
FBF	1	2	Flag buffer flip-flop
XFER	1	2	Transfer in progress flag
WVAL	1	2	Write data valid flag

The Control PCA state contains these registers:

Name	*Size*	*Radix*	*Read-Only*	*Action*
OBUF	16	8		Output buffer
BUSY	2	8	✓	Busy unit number + 1
CNT	9	8		Check record counter
CMD	1	2		Command flip-flop
CTL	1	2		Control flip-flop
FLG	1	2		Flag flip-flop
FBF	1	2		Flag buffer flip-flop
RARC	8	10		Record address register (cylinder)
RARH	5	10		Record address register (head)
RARS	5	10		Record address register (sector)
CYL [0:1]	8	10		Current cylinder, drives 0-1
HED [0:1]	5	10		Current head, drives 0-1
STA [0:1]	16	8		Drive status, drives 0-1
CTIME	24	10		Data transfer command delay time
DTIME	24	10		Data channel command delay time
STIME	24	10		Seek delay time, per cylinder
XTIME	24	10		Inter-word transfer time

5.3 12606B Disc Memory Interface with One 2770/2771 Disc, 12610B Drum Memory Interface with One 2773/2774/2775 Drum

The 12606B Disc Memory Interface connects one HP 2770A or 2771A fixed-head disc drive to the HP computer family. The 12610B Drum Memory Interface connects one HP 2773A 2774A, or 2775A fixed-head drum drive. Both interfaces consist of two Printed Circuit Assemblies installed in adjacent I/O slots. The Data PCA is assigned the higher-priority (lower numbered) select code, and the Control PCA is assigned the lower-priority select code. The DRD and DRC devices simulate these PCAs.

The storage capacity of the HP 277x drives is expandable by adding sets of heads to the base model. This simulation supports the following models and options:

Model-Option	*Tracks*	*Sectors*	*Words*	*Capacity*
2770A	32	90	64	180 KW
2771A	64	90	64	360 KW
2771A-001	128	90	64	720 KW
2773A	192	32	64	384 KW
2773A-001	256	32	64	512 KW
2773A-002	320	32	64	640 KW
2774A	384	32	64	768 KW
2774A-001	448	32	64	896 KW
2774A-002	512	32	64	1024 KW
2775A	768	32	64	1536 KW

Attaching a disc or drum image file to a unit simulates applying power to the drive:

```
ATTACH {-R | -N} DRC <image-filename>
```

Adding the ***–R*** (read-only) switch is equivalent to removing all of the track-protection diodes and setting the Data PCA Track Protect switch to the *On* position. Adding the ***–N*** (new file) switch creates a full-size image file, equivalent to formatting the new disc or drum before use.

This device buffers the image file in memory, and the file is not written until the unit is detached. If the host operating system returns an error when writing the file, the simulator will report the error to the simulation console, e.g.:

```
HP 2100: I/O error - No space left on device
```

No error recovery is attempted.

Device and unit options include configuring the drive type and track protection. The command forms are:

```
SET DRC <device-option>
SET DRC <unit-option>
```

Device options that may be specified are:

Option	*Action*
SC=<n>	Set the interface card select codes; default is 26/27
ENABLED	Enable the device; default
DISABLED	Disable the device

When entering the ***SC*** option, specify the lower select code with either device name; the higher select code will be set automatically, and the cards will be assigned in the correct order.

Device configuration may be displayed with the following command:

Command	*Action*
SHOW DRC	Display the device and unit configuration

Unit options that may be specified are:

Option	*Action*
180K	Use the 12606B interface and 2770A drive
360K	Use the 12606B interface and 2771A drive
720K	Use the 12606B interface and 2771A-001 drive
384K	Use the 12610B interface and 2773A drive
512K	Use the 12610B interface and 2773A-001 drive
640K	Use the 12610B interface and 2773A-002 drive
768K	Use the 12610B interface and 2774A drive
896K	Use the 12610B interface and 2774A-001 drive
1024K	Use the 12610B interface and 2774A-002 drive
1536K	Use the 12610B interface and 2775A drive
PROTECT	Prohibit writing to the protected tracks
UNPROTECT	Permit writing to all tracks; default
TRACKPROT=<n>	Set the number of protected tracks; default is 1

The ***180K***, ***360K***, and ***720K*** options set the interface type and drive model for the fixed-head disc. The ***384K***, ***512K***, ***640K***, ***768K***, ***896K***, ***1024K***, and ***1536K*** set the interface type and drive model for the fixed-head drum.

The ***PROTECT*** option prevents writing to the tracks that have been configured as protected, and the ***UNPROTECT*** option permits writing to all tracks.

The ***TRACKPROT*** option configures the number of protected tracks. In hardware, this corresponds to installing or removing the track-protection diodes on the Data PCA. The track count must be a power of two from 1 to 128 on the 12606B interface and from 1 to 512, or 768, on the 12610B interface. If the drive has fewer tracks than the track-protection count, then all tracks on the drive are eligible for protection.

The interface supports the ***LOAD DRC*** and ***BOOT DRC*** commands to load a disc boot extension into memory. The ***LOAD*** command copies the Basic Binary Disc Loader (BBDL) into memory, configures it for the Data interface's select code, and sets the P register to the loader's starting address. The ***BOOT*** command performs these same actions and then presets the system and begins execution.

For a 2100/14/15/16 CPU containing a previously configured BBDL in memory, the commands are equivalent in hardware to setting the P register to the starting address, pressing the LOADER ENABLE button on the front panel, and, for the ***BOOT*** command, pressing the INTERNAL PRESET, EXTERNAL PRESET, and RUN buttons to begin execution. The S register value is not relevant for this loader.

The BBDL is a dual-purpose loader that also permits loading an absolute binary program from the paper tape reader. When the loader is copied into memory, the disc/drum portion is configured for the select code of the Data interface, and the paper tape portion is configured for the select code of the paper tape reader interface. The P register is automatically set to the starting address of the disc/drum portion, which starts at x7760 (i.e., offset 1760 within the highest page in memory). The loader stores a JMP * instruction into location 77 octal and then reads 64 words from track 0 sector 0 into memory starting at location 0. Loader execution ends when the JMP * is overlaid with a JMP to the address of the start of the bootstrap extension just read. The success or failure of the transfer is not checked.

To execute the paper tape portion, the P register may be reset to x7700 (offset 1700). Loader execution ends with one of the following halt instructions:

- HLT 11 — a checksum error occurred; A = the tape value, B = the calculated value.
- HLT 55 — the program load address would overlay the loader.
- HLT 77 — the end of tape was reached with a successful read.

The 1000 M/E/F-Series does not support this device. Attempting to execute a ***LOAD DRC*** or ***BOOT DRC*** command will be rejected with a **Command not allowed** error.

When a debug log has been established, tracing may be configured for each card separately by specifying the reporting level option:

Option	*Reporting Level*
IOBUS	I/O bus signals and data words received and returned

The ***IOBUS*** option traces the I/O backplane signals and data received and returned by the interface.

The Data PCA state contains these registers:

Name	*Size*	*Radix*	*Action*
IBUF	16	8	Input buffer
OBUF	16	8	Output buffer
CTL	1	2	Control flip-flop
FLG	1	2	Flag flip-flop
BPTR	6	8	Sector buffer pointer

The Control PCA state contains these registers:

Name	*Size*	*Radix*	*Action*
CW	16	8	Command word
STA	16	8	Status word
RUN	1	2	Run flip-flop
TIME	24	10	Inter-word transfer time

5.4 12821A Disc Interface with Four 7906H/7920H/7925H Drives

The HP 12821A Disc Interface connects from one to four HP 7906H (20 MB), 7920H (50 MB), or 7925H (120 MB) Integrated Controller Disc (ICD) drives to the HP 1000. These drives are identical to the 7906M, 7920M, and 7925M Multiple Access Controller (MAC) units but incorporate internal two-card controllers in each drive and connect to the CPU interface via the Hewlett-Packard Interface Bus (HP-IB), HP's implementation of IEEE-488. Each controller is dedicated to a single drive and operates similarly to the 12745 Disc Controller to HP-IB Adapter option for the 13037 Disc Controller chassis. The 7906H was introduced in 1980 (there was no 7905H version, as the 7905 was obsolete by that time). Up to four ICD drives can be connected to a single 12821A card; bus loading and the 937.5 KB/second data transfer rate impose the limitation.

Attaching a disc image file to a unit simulates inserting a disc pack into a drive:

```
ATTACH {-R | -N} DAn <image-filename>
```

Adding the ***–R*** (read-only) switch is equivalent to setting the drive's Disc Protect or Read Only switch to the *On* position. Adding the ***–N*** (new file) switch creates a full-size image file, equivalent to formatting the new disc before use.

If the host operating system returns an error when seeking, reading, or writing a disc image file, the simulator will report the error to the simulation console, e.g.:

```
HP 2100 simulator disc library I/O error: No space left on device
```

A simulated disc seek will fail with Status 2 Error (Drive Fault) status, and the drive's heads will unload. Reloading the heads will clear the drive fault. A simulated disc read or write will fail with Uncorrectable Data Error status. The target operating system will then react to this error as though the drive had encountered a bad disc sector.

A drive's unit number is not set explicitly. Instead, the drive unit number is derived from the simulation unit number. For example, unit DA0 responds to disc unit select number 0. Changing the unit select switch on a mounted drive is simulated by detaching and reattaching the disc image file to the corresponding simulation unit.

Device and unit options include configuring the timing, drive type, protection and format status, and the ability to set drives ready or not-ready. The command forms are:

```
SET DA <device-option>
SET DAn <unit-option>
```

5.4.1 Device Options

Device options that may be specified are:

Option	*Action*
ADDRESS=<n>	Set the HP-IB address; default is 0
DIAGNOSTIC	Connect the HP-IB cable to a second interface
HPIB	Connect the HP-IB cable to the drives
SC=<n>	Set the interface card select code; default is 43
DEBUG=<option>	Enable tracing
NODEBUG	Disable tracing; default
ENABLED	Enable the device; default
DISABLED	Disable the device

The ***ADDRESS*** option configures the interface to use the specified HP-IB address from 0-7. The interface address is independent of the disc drive unit address and is only used for diagnostics.

Setting the ***DIAGNOSTIC*** option enables a second 12821A device (named DC) and simulates the installation of an HP-IB cable between the two 12821A Disc Interface cards as required by the HP 12821A ICD Disc Interface Diagnostic. Setting the ***HPIB*** option disables the second device and simulates the installation of an HP-IB cable between the interface and the drives.

Device configuration may be displayed with the following commands:

Command	*Action*
SHOW DA	Display the device and unit configuration
SHOW DA CABLE	Display the HP-IB cable configuration
SHOW DA BUS	Display the bus address

5.4.2 Unit Options

Unit options that may be specified for individual disc drives are:

Option	*Action*
7906H	Set the drive type to 7906H; default
7920H	Set the drive type to 7920H
7925H	Set the drive type to 7925H
BUS=<n>	Set the drive's HP-IB address; default is the unit number
UNLOAD	Set the drive's Run/Stop switch to *Stop*; default when detached
LOAD	Set the drive's Run/Stop switch to *Run*; default when attached
PROTECT	Set the drive's Disc Protect or Read Only switch to *On*
UNPROTECT	Set the drive's Disc Protect or Read Only switch to *Off*
FORMAT	Set the drive's Format switch to *Enabled*
NOFORMAT	Set the drive's Format switch to *Disabled*; default
ENABLED	Enable the unit; default
DISABLED	Disable the unit

The ***7906H***, ***7920H***, and ***7925H*** options select the disc drive model.

The ***BUS*** option sets the HP-IB address of the drive. Bus addresses range from 0-7, and each unit's address must be unique. Each unit's bus address is initially set to its unit number.

The ***UNLOAD*** and ***LOAD*** options unload and load the drive's heads from the disc pack, setting the drive not-ready and ready, respectively. The former provides a convenient method of setting a drive "down" without detaching the associated disc image file.

The ***PROTECT*** option prevents writing to the disc. It is the hardware equivalent of setting the 7906H Upper and Lower Disc Protect switches or the 7920H or 7925H Read Only switch to the *On* position. The ***UNPROTECT*** option enables writing and is equivalent to setting the switches to *Off*. Separate protection for the upper and lower platters of the 7906H drive is not supported.

The ***FORMAT*** option enables certain controller commands, such as Initialize, that alter the sector address fields. Typically, this option is needed when installing an operating system. The ***NOFORMAT*** option inhibits these commands and permits only the standard Write command, subject to the appropriate drive protection status.

Drive configuration may be displayed with the following command:

Command	*Action*
SHOW DA<n>	Display the selected drive's configuration

5.4.3 LOAD and BOOT Commands

The interface supports the ***LOAD DA*** and ***BOOT DA*** commands to load a disc boot extension into memory. The ***LOAD*** command copies the 12992H boot loader ROM into memory, configures it for the interface's select code, and sets the P register to the loader's starting address. The ***BOOT*** command performs these same actions and then presets the system and begins execution.

For a 1000 M/E/F-Series CPU, the commands are equivalent in hardware to setting the S register to select and configure the HP 12992H boot loader ROM, pressing the IBL button to copy the ROM into memory, and, for the ***BOOT*** command, pressing the PRESET and RUN buttons to begin execution. The S register is preset as follows:

15	*14*	*13*	*12*	*11*	*10*	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
0	0	0	1	DA select code						unchanged			0	head	

The head number is obtained from bits 1-0 of the existing S-register value when the loader is executed. Bits 5-3 of the existing S-register value are also retained and are available to the boot extension program.

The loader reads 620 words from cylinder 0 sector 0 of the specified head into memory starting at location 2011 octal. Loader execution ends with one of the following instructions:

- HLT 11 — the drive aborted the transfer due to an unrecoverable error.
- JSB 2055,I — the disc read succeeded.

The 2100/14/15/16 CPU does not support this device. Attempting to execute a ***LOAD DA*** or ***BOOT DA*** command will be rejected with a **Command not allowed** error.

5.4.4 Tracing and Registers

When a debug log has been established, tracing may be configured by specifying one or more of the reporting level options:

Option	*Reporting Level*
RWSC	Controller read, write, status, and control commands
CMDS	Interface commands received from the CPU
CPU	Words received from and sent to the CPU
BUF	Data read from and written to the interface FIFO
XFER	Data received and transmitted via HP-IB
SERV	Disc unit service events
IOBUS	I/O bus signals and data words received and returned

The ***RWSC*** option traces the read, write, status, and control commands executed by the disc controller. The ***CMDS*** option traces the commands executed by the interface. The ***CPU*** option traces words received from (control and write data) and sent to (status and read data) the CPU. The ***BUF*** option traces loads into and unloads from the interface FIFO. The ***XFER*** option traces the data words read from or written to the disc. The ***IOBUS*** option traces the I/O backplane signals and data received and returned via the interface.

Examples of the trace formats follow:

```
>>DA  rwsc: Unit 1 position 1069056 seek disc command initiated
>>DA  rwsc: Unit 1 seek disc command completed
>>DA  cmds: [STC] Control set (data mode)
>>DA  cmds: SRQ cleared
>>DA   cpu: [OTx] Control = 004003
>>DA   cpu: [LIx,C] Data = 063613
>>DA   buf: Data 277 tag ATN | EDT unloaded from FIFO (0)
>>DA   buf: FIFO cleared
>>DA  xfer: Data 100677 tag ATN | EDT loaded into FIFO (1)
>>DA  xfer: HP-IB DIO 277 signals ATN available
>>DA  xfer: HP-IB address 0 accepted unlisten
>>DA  xfer: HP-IB address 1 accepted secondary 08H
>>DA  serv: Unit 1 state read transfer delay 3 service scheduled
>>DA  serv: Unit 1 state command execution delay 100 service scheduled
>>DA iobus: Received data 000000 with signals CLC | CLF | SIR | PRH
>>DA iobus: Returned data 000000 with signals PRL
```

The Disc Interface state contains these registers:

Name	*Size*	*Radix*	*Description*
CWR	16	8	Control word register
SWR	16	8	Status word register
IDR	16	8	Input data register
FCNT	5	10	FIFO fill count
FIFO [0:15]	20	8	Data FIFO
ACPT	4	2	Bus acceptor units bitmap
LSTN	4	2	Bus listener units bitmap
TALK	4	2	Bus talker unit bitmap
PPR	8	2	Parallel poll response bitmap
BUSCTL	8	2	HP-IB control lines
CTL	1	2	Interface control
FLG	1	2	Interface flag
FBF	1	2	Interface flag buffer
SRQ	1	2	Interface DCPC service request
EDT	1	2	End of data transfer flag
EOR	1	2	End of record flag
BUFFER [0:137]	16	8	Sector buffer
DSJ [0:3]	2	10	Controller Device-Specified Jump, drives 0-3
ISTATE [0:3]	4	10	Interface state, drives 0-3
ICMD [0:3]	4	10	Interface command, drives 0-3

5.5 13175D Disc Controller Interface with Eight 7905/7906/7920/7925 Drives

The HP 13175D Disc Controller Interface connects 7905A (15 MB), 7906A (20 MB), 7920A (50 MB), and 7925A (120 MB) disc drives to the HP 1000. The subsystem consists of the interface, a 13037D Multiple Access Controller (MAC), and from one to eight MAC drives. The subsystem uses DCPC to achieve a 937.5 KB/second transfer rate to the CPU.

The disc controller connects from one to eight disc drives to interfaces installed in from one to eight CPUs. The drives use a common command set and present data to the controller synchronously at a 468.75 kiloword per second (2.133 microseconds per word) data rate.

The disc interface is used to connect the HP 1000 CPU to the 13037's device controller. While the controller supports multiple-CPU systems in hardware, the simulator does not provide this capability.

Attaching a disc image file to a unit simulates inserting a disc pack into a drive:

```
ATTACH {-R | -N} DSn <image-filename>
```

Adding the ***–R*** (read-only) switch is equivalent to setting the drive's Disc Protect or Read Only switch to the *On* position. Adding the ***–N*** (new file) switch creates a full-size image file, equivalent to formatting the new disc before use.

If the host operating system returns an error when seeking, reading, or writing a disc image file, the simulator will report the error to the simulation console, e.g.:

```
HP 2100 simulator disc library I/O error: No space left on device
```

A simulated disc seek will fail with Status 2 Error (Drive Fault) status, and the drive's heads will unload. Reloading the heads will clear the drive fault. A simulated disc read or write will fail with Uncorrectable Data Error status. The target operating system will then react to this error as though the drive had encountered a bad disc sector.

A drive's unit number is not set explicitly. Instead, the drive unit number is derived from the simulation unit number. For example, unit DS0 responds to disc unit select number 0. Changing the unit select switch on a mounted drive is simulated by detaching and reattaching the disc image file to the corresponding simulation unit.

Device and unit options include configuring the timing, drive type, protection and format status, and the ability to set drives ready or not-ready. The command forms are:

```
SET DS <device-option>
SET DSn <unit-option>
```

5.5.1 Device Options

Device options that may be specified are:

Option	*Action*
SC=<n>	Set the interface card select code; default is 34
DEBUG=<option>	Enable tracing
NODEBUG	Disable tracing; default
ENABLED	Enable the device; default
DISABLED	Disable the device

Device configuration may be displayed with the following commands:

Command	*Action*
SHOW DS	Display the device and unit configuration

5.5.2 Unit Options

Unit options that may be specified for individual disc drives are:

Option	*Action*
7905	Set the drive type to 7905; default
7906	Set the drive type to 7906
7920	Set the drive type to 7920
7925	Set the drive type to 7925
AUTOSIZE	Set the drive type based on the attached file size
UNLOAD	Set the drive's Run/Stop switch to *Stop*; default when detached
LOAD	Set the drive's Run/Stop switch to *Run*; default when attached
PROTECT	Set the drive's Disc Protect or Read Only switch to *On*
UNPROTECT	Set the drive's Disc Protect or Read Only switch to *Off*
FORMAT	Set the drive's Format switch to *Enabled*
NOFORMAT	Set the drive's Format switch to *Disabled*; default
ENABLED	Enable the unit; default
DISABLED	Disable the unit

The ***7905***, ***7906***, ***7920***, and ***7925*** options select the disc drive model. The ***AUTOSIZE*** option selects the disc drive model by the size of the attached disc image file. The algorithm selects the smallest-capacity drive capable of supporting the image. This option is useful when several previously written images of various sizes are attached sequentially.

The ***UNLOAD*** and ***LOAD*** options unload and load the drive's heads from the disc pack, setting the drive not-ready and ready, respectively. The former option provides a convenient method of setting a drive "down" without detaching the associated disc image file.

The ***PROTECT*** option prevents writing to the disc. It is the hardware equivalent of setting the 7905 or 7906 Upper and Lower Disc Protect switches or the 7920 or 7925 Read Only switch to the *On* position. The ***UNPROTECT*** option enables writing and is equivalent to setting the switches to *Off*. Separate protection for the upper and lower platters of the 7905 and 7906 drives is not supported.

The ***FORMAT*** option enables certain controller commands, such as Initialize, that alter the sector address fields. Typically, this option is needed when installing the operating system. The ***NOFORMAT*** option inhibits these commands and permits only the standard Write command, subject to the appropriate drive protection status.

Drive configuration may be displayed with the following command:

Command	***Action***
SHOW DS<n>	Display the selected drive's configuration

5.5.3 LOAD and BOOT Commands

The interface supports the ***LOAD DS*** and ***BOOT DS*** commands to load a disc boot extension into memory. Depending on the current CPU model, the ***LOAD*** command copies either the Basic Moving-Head Disc Loader (BMDL-7905) or the 12992B boot loader ROM into memory, configures it for the interface's select code, and sets the P register to the loader's starting address. The ***BOOT*** command performs these same actions and then presets the system and begins execution.

For a 2100/14/15/16 CPU containing a previously configured BMDL in memory, the commands are equivalent in hardware to setting the P register to the starting address, pressing the LOADER ENABLE button on the front panel, and, for the ***BOOT*** command, pressing the INTERNAL PRESET, EXTERNAL PRESET, and RUN buttons to begin execution.

The BMDL is a dual-purpose loader that also permits loading an absolute binary program from the paper tape reader. When the loader is copied into memory, the disc portion is configured for the select code of the disc interface, and the paper tape portion is configured for the select code of the paper tape reader interface. The P register is automatically set to the starting address of the disc portion, which starts at x7750 (i.e., offset 1750 within the highest page in memory). Before execution, the S register must be set as follows:

15	***14***	***13***	***12***	***11***	***10***	***9***	***8***	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
0	0	0	0	0	0	0	0	0	0	0	0	0	0	head	

The loader stores the value 002055 into location 2055 octal, stores a HLT 11 into location 2056 octal, and then reads 2047 words from cylinder 0 sector 0 of the specified head into memory starting at location 2011 octal. Loader execution ends with one of the following instructions:

- HLT 11 — the disc read failed.
- JSB 2055,I — the disc read succeeded.

To execute the paper tape portion, the P register may be reset to x7700 (offset 1700). Loader execution ends with one of the following halt instructions:

- HLT 11 — a checksum error occurred; A = the tape value, B = the calculated value.
- HLT 55 — the program load address would overlay the loader.
- HLT 77 — the end of tape was reached with a successful read.

For a 1000 M/E/F-Series CPU, the commands are equivalent in hardware to setting the S register to select and configure the HP 12992B boot loader ROM, pressing the IBL button to copy the ROM into memory, and, for the ***BOOT*** command, pressing the PRESET and RUN buttons to begin execution. The S register is preset as follows:

15	***14***	***13***	***12***	***11***	***10***	***9***	***8***	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
0	0	0	1	DS select code						unchanged			0	head	

The head number is obtained from bits 1-0 of the existing S-register value when the loader is executed. Bits 5-3 of the existing S-register value are also retained and are available to the boot extension program.

Loader execution ends with one of the following instructions:

- HLT 30 — the drive is not ready.
- JSB 2055,I — the disc read succeeded.

5.5.4 Tracing and Registers

When a debug log has been established, tracing may be configured by specifying one or more of the reporting level options:

Option	***Reporting Level***
RWSC	Controller read, write, status, and control commands
CMDS	Interface commands received from the CPU
CPU	Words received from and sent to the CPU
BUF	Data read from and written to the interface FIFO
SERV	Disc unit service events
IOBUS	I/O bus signals and data words received and returned

The ***RWSC*** option traces the read, write, status, and control commands executed by the disc controller. The ***CMDS*** option traces the commands executed by the interface. The ***CPU*** option traces words received from (control and write data) and sent to (status and read data) the CPU. The ***BUF*** option traces loads into and unloads from the interface FIFO. The ***IOBUS*** option traces the I/O backplane signals and data received and returned via the interface.

Examples of the trace formats follow:

```
>>DS  rwsc: Unit 0 position 0 cold load read command initiated
>>DS  rwsc: Controller address record command completed
>>DS  cmds: [STF] Flag set
>>DS  cmds: [LIx] SRQ cleared
>>DS   cpu: [OTx,C] Command = 001200
>>DS   cpu: [EDT] DCPC transfer ended
>>DS   buf: Data 020334 loaded into FIFO (5)
>>DS   buf: Data 170001 unloaded from FIFO (0)
>>DS  serv: Unit 0 delay 2400 service scheduled
```

```
>>DS  serv: Controller delay 3 service scheduled
>>DS iobus: Received data 021057 with signals IOO | CLF | EDT | SIR
>>DS iobus: Returned data 174163 with signals (none)
```

The Disc Interface state contains these registers:

Name	*Size*	*Radix*	*Description*
CMFOL	1	2	Command follows flip-flop
CMRDY	1	2	Command ready flip-flop
FCNT	5	10	FIFO fill count
FIFO [0:15]	16	8	Data FIFO
STATE	2	8	Controller state
OPCODE	6	8	Current operation code
STATUS	6	8	Last operation status
EOC	1	2	End of cylinder flip-flop
EOD	1	2	End of data flip-flop
SPDU	16	8	S/P/D flags and last unit
FLMASK	4	8	File mask
CYL	16	10	Controller cylinder
HEAD	6	10	Controller head
SECTOR	8	10	Controller sector
VFYCNT	16	8	Verify counter
LASPOL	3	8	Last unit polled
BUFFER [0:137]	16	8	Sector buffer
INDEX	8	10	Buffer index
LENGTH	8	10	Buffer length
STIME	24	10	Per-cylinder seek delay time
ITIME	24	10	Intersector delay time
CTIME	24	10	Command response time
DTIME	24	10	Data transfer response time
WTIME	31	10	Command wait time
CTL	1	2	Interface control flip-flop
FLG	1	2	Interface flag flip-flop
FBF	1	2	Interface flag buffer flip-flop
SRQ	1	2	Interface DCPC service request flip-flop
EDT	1	2	End of data transfer flag
UCYL [0:9]	10	10	Current cylinder, drives 0-7, controller, and timer
UOP [0:9]	6	8	Current operation code, drives 0-7, controller, and timer
USTAT [0:9]	8	2	Unit status, drives 0-7, controller, and timer
UPHASE [0:9]	3	8	Current command state, drives 0-7, controller, and timer
UPOS [0:9]	32	8	Current byte position, drives 0-7, controller, and timer
UWAIT [0:9]	32	8	Scheduled wait delay, drives 0-7, controller, and timer

6 Magnetic Tape I/O Device Simulations

The simulator provides these magnetic tape device interfaces and their associated peripherals:

- 12559C 9-Track Magnetic Tape Unit Interface
- 13181B Digital Magnetic Tape Unit Interface
- 13183B Digital Magnetic Tape Unit Interface

6.1 12559C 9-Track Magnetic Tape Unit Interface with One 3030 Drive

The 12559C 9-Track Magnetic Tape Unit Interface connects one HP 3030 800-bpi NRZI ½-inch magnetic tape drive to the HP computer family. The interface consists of two Printed Circuit Assemblies installed in adjacent I/O slots. The Data PCA is assigned the higher-priority (lower numbered) select code, and the Control PCA is assigned the lower-priority select code. The MTD and MTC devices simulate these PCAs

Attaching a tape image file simulates mounting a tape reel on a drive:

```
ATTACH {-R} {-F} MTC {<format>} <image-filename>
```

Adding the ***–R*** (read-only) switch is equivalent to mounting the tape without a write ring in place. Adding the ***–F*** switch and a format identifier declares the tape image format to be used. Supported formats are ***SIMH***, ***E11***, ***TPC***, and ***P7B***. If the ***–F*** switch and a format identifier are not supplied, then SIMH tape image format is used. Note that erase gaps embedded in the tape image file are supported only in SIMH image format mode.

If the host operating system returns an error when reading or writing a tape image file, the simulator will report the error to the simulation console, e.g.:

```
HP 2100: Magtape library I/O error: No space left on device
```

If this or another tape library error occurs, e.g., due to an illegal or damaged tape format, the simulator stops with an appropriate error message. Resuming the simulation will fail the tape operation with Tape Error status. The target operating system will then react to this error as though the drive had encountered a bad tape block (CRCC or LRCC error).

Device options that may be specified with the ***SET MTC*** command are:

Option	*Action*
FORMAT=<fmt>	Set the tape image format; default is SIMH format
SC=<n>	Set the interface card select codes; defaults are 20/11
ENABLED	Enable the device;
DISABLED	Disable the device; default

The tape image format for future ***ATTACH*** commands may be set to one of the format identifiers listed above. The unit must be detached when the format is set.

When entering the ***SC*** option, specify the lower select code with either device name; the higher select code will be set automatically, and the cards will be assigned in the correct order.

Drive configuration may be displayed with the following commands:

Command	*Action*
SHOW MTC	Display the drive's configuration
SHOW MTC FORMAT	Display the drive's tape image format

The ***BOOT*** command is not supported. The 12559C is disabled by default.

When a debug log has been established, tracing may be configured for each card separately by specifying the reporting level option:

Option	*Reporting Level*
IOBUS	I/O bus signals and data words received and returned

The ***IOBUS*** option traces the I/O backplane signals and data received and returned by the interface.

The Data PCA state contains these registers:

Name	*Size*	*Radix*	*Action*
FLG	1	2	Flag flip-flop
FBF	1	2	Flag buffer flip-flop
DBUF [0:65535]	8	8	Tape record buffer
BPTR	17	10	Buffer pointer
BMAX	17	10	Buffer size for writes

The Control PCA state contains these registers:

Name	*Size*	*Radix*	*Action*
FNC	8	8	Current function
STA	9	8	Tape status
BUF	8	8	Command buffer
CTL	1	2	Control flip-flop
FLG	1	2	Flag flip-flop
FBF	1	2	Flag buffer flip-flop
DTF	1	2	Data transfer flip-flop
FSVC	1	2	First service flip-flop
POS	32	10	Tape position
CTIME	24	10	Command delay time
GTIME	24	10	Inter-record gap traversal time
XTIME	24	10	Inter-word transfer time

6.2 13181B Digital Magnetic Tape Unit Interface with Four 7970B Drives, 13183B Digital Magnetic Tape Unit Interface with Four 7970E Drives

The HP 13181B and 13183B Digital Magnetic Tape Unit Interfaces connect from one to four HP 7970B 800-bpi NRZI and HP 7970E 1600-bpi PE ½-inch magnetic tape drives, respectively, to the HP computer family. Both interfaces consist of two Printed Circuit Assemblies installed in adjacent I/O slots. The Data PCA is assigned the higher-priority (lower numbered) select code, and the Control PCA is assigned the lower-priority select code. The MSD and MSC devices simulate these PCAs.

Attaching a tape image file to a unit simulates mounting a tape reel on a drive:

```
ATTACH {-R} {-F} MSCn {<format>} <image-filename>
```

Adding the ***–R*** (read-only) switch is equivalent to mounting the tape without a write ring in place. Adding the ***–F*** switch and a format identifier declares the tape image format to be used. Supported formats are ***SIMH***, ***E11***, ***TPC***, and ***P7B***. If the ***–F*** switch and a format identifier are not supplied, then SIMH tape image format is used. Note that erase gaps embedded in the tape image file are supported only in SIMH image format mode.

If the host operating system returns an error when reading or writing a tape image file, the simulator will report the error to the simulation console, e.g.:

```
HP 2100: Magtape library I/O error: No space left on device
```

If this or another tape library error occurs, e.g., due to an illegal or damaged tape format, the simulator stops with an appropriate error message. Resuming the simulation will fail the tape operation with Tape Error status. The target operating system will then react to this error as though the drive had encountered a bad tape block (CRCC/LRCC or STE/MTE error).

A drive's unit number is not set explicitly. Instead, the drive unit number is derived from the simulation unit number. For example, unit MSC0 responds to tape unit select number 0. Pressing a different unit select button on a mounted drive is equivalent to detaching and reattaching the tape image file to the corresponding simulation unit. Pressing the *OFF* button is equivalent to setting the drive offline.

Device and unit options include configuring the drive type and timing, tape reel size, tape image format, and the ability to set drives offline or online. The command forms are:

```
SET MSC <device-option>
SET MSCn <unit-option>
```

6.2.1 Device Options

The device options that may be specified are:

Option	***Action***
13181	Use the 13181B interface and 7970B drives; default
13183	Use the 13183B interface and 7970E drives
FASTTIME	Use optimized timing; default
REALTIME	Use realistic timing
SC=<n>	Set the interface card select codes; defaults are 30/31
DEBUG=<option>	Enable tracing
NODEBUG	Disable tracing; default
ENABLED	Enable the device; default
DISABLED	Disable the device

The ***13181*** and ***13183*** options set the interface type and, by implication, the tape drive type.

When realistic timing is enabled, the simulation accurately models the tape movement times (in machine instructions). For example, rewinding takes longer if the tape is positioned farther from the load point. Realistic timing is necessary to pass the magnetic tape diagnostic timing tests.

Optimized timing reduces the timing to the minimum necessary to operate correctly; this is much faster than the real tape drive would operate. In addition, enabling optimized timing also omits the initial erase gap normally written after the BOT marker and the erase gap written by the Gap and File Mark (GFM), which is executed as a Write File Mark (WFM) command. This does not affect gaps written with the Write Gap command.

The delay times used by the simulator in ***FASTTIME*** mode may be set via the register interface. The values may be adjusted as necessary to work around any HP software problems that are triggered by the unusually rapid tape operation. Resetting the device with the ***RESET –P*** (power-on reset) command restores the original optimized times.

When entering the ***SC*** option, specify the lower select code with either device name; the higher select code will be set automatically, and the cards will be assigned in the correct order.

Device configuration may be displayed with the following commands:

Command	*Action*
SHOW MSC	Display the device and unit configuration
SHOW MSC TIMING	Display the timing mode
SHOW MSC TYPE	Display the interface type

6.2.2 Unit Options

Unit options that may be specified for individual tape drives are:

Option	*Action*
OFFLINE	Set the unit offline; default when detached
ONLINE	Set the unit online; default when attached
REEL=<n>	Set the reel size in feet; default is unlimited
CAPACITY=<n>	Set the reel capacity in megabytes; default is unlimited
FORMAT=<fmt>	Set the tape image format; default is SIMH format
ENABLED	Enable the unit; default
DISABLED	Disable the unit

The ***OFFLINE*** and ***ONLINE*** options place a drive offline and online, respectively. The former provides a convenient method of setting a drive "down" without detaching the associated tape image file.

The reel size may be set to ***600***-, ***1200***-, or ***2400***-foot capacity. Setting the capacity or reel size to ***0*** specifies unlimited capacity; in this configuration, the interface never returns an end-of-tape indication.

The tape image format for future ***ATTACH*** commands may be set to one of the format identifiers listed above. The unit must be detached when the format is set.

Drive configuration may be displayed with the following commands:

Command	*Action*
SHOW MSC<n>	Display the selected drive’s configuration
SHOW MSC<n> REEL	Display the selected drive’s reel size or capacity
SHOW MSC<n> CAPACITY	Display the selected drive’s reel size or capacity
SHOW MSC<n> FORMAT	Display the selected drive’s tape image format

6.2.3 LOAD and BOOT Commands

The interface supports the ***LOAD {–S} MSC*** and ***BOOT{–S} MSC*** commands to load an absolute binary program into memory. Depending on the current CPU model, the ***LOAD*** command copies either the Basic Magnetic Tape Loader (BMTL) or the 12992D boot loader ROM into memory, configures it for the Data interface's select code, and sets the P register to the loader's starting address. The ***BOOT*** command performs these same actions and then presets the system and begins execution.

For a 2100/14/15/16 CPU containing a previously configured BMTL in memory, the commands are equivalent in hardware to setting the P register to the starting address, pressing the LOADER ENABLE button on the front panel, and, for the ***BOOT*** command, pressing the INTERNAL PRESET, EXTERNAL PRESET, and RUN buttons to begin execution. Before execution, the S register must be set as follows:

15	***14***	***13***	***12***	***11***	***10***	***9***	***8***	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
0	0	0	0	0	0	0	0	0	0	file number					

If S-register bits 5-0 are zero, the file located at the current tape position is read. If the bits are non-zero, the tape is rewound, and the file number (1 - n) specified by the bits is read. The tape format must be absolute binary, and a tape mark must end each file. Loader execution ends with one of the following instructions:

- HLT 00 — a tape read (parity) error occurred.
- HLT 11 — a checksum error occurred; A = the calculated value, B = the tape value.
- HLT 77 — the end of the file was reached with a successful read.

For a 1000 M/E/F-Series CPU, the commands are equivalent in hardware to setting the S register to select and configure the HP 12992D boot loader ROM, pressing the IBL button to copy the ROM into memory, and, for the ***BOOT*** command, pressing the PRESET and RUN buttons to begin execution. The S register is preset as follows:

15	***14***	***13***	***12***	***11***	***10***	***9***	***8***	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
1	0	0	0	MSD select code						unchanged			0	0	●

If the ***–S*** switch is specified, bit 0 is set to 1 to rewind the tape and then load the file number (1 - n) specified in the A register. If ***–S*** is not specified, then bit 0 is set to 0 to load the file located at the current tape position. Bits 5-3 of the existing S-register value are retained when the loader is executed and are available to the loaded program.

Loader execution ends with one of the following halt instructions:

- HLT 00 — a tape read (parity) error occurred.
- HLT 11 — a checksum error occurred; A = the calculated value, B = the tape value.
- HLT 77 — the end of the file was reached with a successful read.

6.2.4 Tracing and Registers

When a debug log has been established, tracing may be configured by specifying one or more of the reporting level options:

Option	*Reporting Level*
CMDS	Controller command initiations and completions
RWS	Tape reads, writes, and status returns
CPU	Words received from and sent to the CPU
IOBUS	I/O bus signals and data words received and returned

The ***CMDS*** option traces the commands executed by the tape controller. The ***RWS*** option traces the read, write, and status commands executed by the controller. The ***CPU*** option traces words received from (control and write data) and sent to (status and read data) the CPU. The ***IOBUS*** option traces the I/O backplane signals and data received and returned via the interface.

Examples of the trace formats follow:

```
>>MSC  cmds: Unit 0 command 023 (RRF) scheduled, pos = 0, time = 1001
>>MSC  cmds: Unit 0 command 023 (RRF) complete
>>MSC   rws: Unit 0 read 33 word record
>>MSC   rws: Unit 0 tape library status = 1
>>MSC   cpu: Command = 000101
>>MSC   cpu: Status = 000004
>>MSC iobus: Received data 000000 with signals STC | CLF | SIR | PRH
>>MSC iobus: Returned data 000020 with signals PRL
```

The Data PCA state contains these registers:

Name	*Size*	*Radix*	*Action*
BUF	16	8	Data buffer
CTL	1	2	Control flip-flop
FLG	1	2	Flag flip-flop
FBF	1	2	Flag buffer flip-flop
DBUF [0:65535]	8	8	Tape record buffer
BPTR	17	10	Buffer pointer
BMAX	17	10	Buffer size for writes

The Control PCA state contains these registers:

Name	*Size*	*Radix*	*Action*
STA	12	8	Tape status
BUF	16	8	Command buffer
USEL	2	8	Currently selected unit
FSVC	1	2	First service flip-flop
CTL	1	2	Control flip-flop
FLG	1	2	Flag flip-flop
FBF	1	2	Flag buffer flip-flop
POS [0:3]	32	10	Tape position, drives 0-3
BTIME	24	10	BOT start delay time
CTIME	24	10	Command delay time
GTIME	24	10	Gap traversal time
ITIME	24	10	IRG traversal time
RTIME	24	10	Rewind initiation time
XTIME	24	10	Inter-word transfer time