

HP2100 Simulator Usage

30 Sep-2008

COPYRIGHT NOTICE

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

Original code published in 1993-2008, written by Robert M Supnik
Copyright (c) 1993-2008, Robert M Supnik

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 ROBERT M SUPNIK 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 name of Robert M Supnik shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from Robert M Supnik.

1	Simulator Files	3
2	HP2100 Features	4
2.1	CPU	5
2.2	12581A/12892B Memory Protect	9
2.3	12607B/12578A/12895A DMA Controllers, 12897B DCPC Controller	10
2.4	Variable Device Assignments	10
2.4.1	Device State	10
2.5	Programmed I/O Devices	11
2.5.1	12597A-002 Duplex Register Interface (PTR) with 2748 Paper Tape Reader	11
2.5.2	12597A-005 Duplex Register Interface (PTP) with 2895 Paper Tape Punch	12
2.5.3	12531C Buffered Teleprinter Interface (TTY) with 2752 Teleprinter	13
2.5.4	12966A Buffered Asynchronous Communications Interface (BACI)	14
2.5.5	12653A Printer Controller (LPS) with 2767 Line Printer, 12566B Microcircuit Interface with Loopback Connector	16
2.5.6	12845B Printer Controller (LPT) with 2607 Line Printer	17
2.5.7	12539C Time Base Generator (CLK)	18
2.5.8	12792C 8-Channel Asynchronous Multiplexer (MPX)	18
2.5.9	12920A 16-Channel Terminal Multiplexer (MUX, MUXL, MUXM)	21
2.5.10	12875A Interprocessor Link (IPLI, IPLO)	24
2.5.11	12620A/12936A Privileged Interrupt Fence (PIF)	25
2.6	Disk Controllers	26
2.6.1	12557A Disk Controller (DPC, DPD) with Four 2781 Drives, 13210A Disk Controller (DPC, DPD) with Four 7900 Drives	26
2.6.2	12565A Disk Controller (DQC, DQD) with Two 2883 Drives	28
2.6.3	12606B Fixed Head Disk Controller (DRC, DRD) with 2770/2771 Disk, 12610B Drum Controller (DRC, DRD) with 2773/2774/2775 Drum	29
2.6.4	13037D Disk Controller (DS) with Eight 7905/7906/7920/7925 Drives	30
2.7	Magnetic Tape	32
2.7.1	12559C Magnetic Tape Controller (MTC, MTD) with One 3030 Drive	32
2.7.2	13181A Magnetic Tape Controller (MSC, MSD) with Four 7970B Drives, 13183A Magnetic Tape Controller (MSC, MSD) with Four 7970E Drives	33
3	Symbolic Display and Input	35

This memorandum documents the HP 2100 simulator.

1 Simulator Files

```
sim/      scp.h
          sim_console.h
          sim_defs.h
          sim_fio.h
          sim_rev.h
          sim_sock.h
          sim_tape.h
          sim_timer.h
          sim_tmxr.h
          scp.c
          sim_console.c
          sim_fio.c
          sim_sock.c
          sim_tape.c
          sim_timer.c
          sim_tmxr.c

sim/hp2100/ hp2100_cpu.h
            hp2100_cpu1.h
            hp2100_defs.h
            hp2100_fp.h
            hp2100_fp1.h
            hp2100_baci.c
            hp2100_cpu.c
            hp2100_cpu0.c
            hp2100_cpu1.c
            hp2100_cpu2.c
            hp2100_cpu3.c
            hp2100_cpu4.c
            hp2100_cpu5.c
            hp2100_cpu6.c
            hp2100_cpu7.c
            hp2100_fp.c
            hp2100_fp1.c
            hp2100_dp.c
            hp2100_dq.c
            hp2100_dr.c
            hp2100_ds.c
            hp2100_ipl.c
            hp2100_lps.c
            hp2100_lpt.c
            hp2100_mpx.c
            hp2100_mt.c
            hp2100_ms.c
            hp2100_mux.c
            hp2100_pif.c
            hp2100_stddev.c
            hp2100_sys.c
            hp2100_diag.txt
```

2 HP2100 Features

The HP2100 simulator is configured as follows:

device names	simulates
CPU	2114C CPU with up to 16KW of memory 2115A CPU with up to 8KW of memory 2116C CPU with up to 32KW of memory 2100A CPU with up to 32KW of memory 1000 (21MX) M/E/F-Series CPU with up to 1024KW of memory EAU, FP, IOP, DMS, FFP, DBI, VIS, SIGNAL, RTE-IV EMA, and/or RTE-6/VM OS and VMA microcode extensions
MP	12581A/12892B memory protect
DMA0, DMA1	12607B/12578A/12895A direct memory access controller
DCPC0, DCPC1	12897B dual-channel port controller
PTR	12597A duplex register interface with 2748 paper tape reader
PTP	12597A duplex register interface with 2895 paper tape punch
TTY	12531C buffered teleprinter interface with 2752 teleprinter
BACI	12966A buffered asynchronous communications interface
LPS	12653A printer controller with 2767 line printer 12566B microcircuit interface with loopback connector
LPT	12845B printer controller with 2607 line printer
CLK	12539C time base generator
MPX	12792C 8-channel asynchronous multiplexer
MUX, MUXL, MUXM	12920A 16-channel terminal multiplexer
DPD, DPC	12557A disk controller with four 2871 drives 13210A disk controller with four 7900 drives
DQD, DQC	12565A disk controller with two 2883 drives
DRD, DRC	12606B fixed head disk controller with 2770/2771 disk 12610B drum controller with 2773/2774/2775 drum
DS	13037D disk controller with eight 7905/7906/7920/7925 drives
MTD, MTC	12559C magnetic tape controller with one 3030 drive
MSD, MSC	13181A magnetic tape controller with four 7970B drives 13183A magnetic tape controller with four 7970E drives
IPLI, IPLO	12875A interprocessor link
PIF	12620A/12936A privileged interrupt fence

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

- 2000 E, F, and Access Time-Shared BASIC
- DOS-III
- RTE-II, III, IV, IVB, and 6/VM

In addition, the simulator generally passes the HP 24396 Diagnostics Suite; see *hp2100_diag.txt* for details.

The simulator implements several unique stop conditions:

- Attempted execution of an undefined instruction, and STOP_INST is set
- Reference to an undefined I/O device, and STOP_DEV is set
- More than INDMAX indirect references are detected during memory reference address decoding

The `LOAD` command supports standard absolute binary format. The `DUMP` command is not implemented.

Refer to the note in Section 2.5.1 regarding the use of the `LOAD` command with HP software.

2.1 2114C/2115A/2116C, 2100A, 1000-M/E/F CPUs

CPU options include choice of model, memory size, and instruction sets. Several microcode options are simulated:

EAU	Extended Arithmetic Unit
FP	Single-Precision Floating Point
IOP	2000/Access I/O Processor
DMS	Dynamic Mapping System
FFP	Fast FORTRAN Processor
DBI	Double Integer Instructions
VIS	Vector Instruction Set
SIGNAL	SIGNAL/1000 Instructions
EMA/VMA	Extended Memory/Virtual Memory/OS Instructions

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

EIG	Extended Instruction Group
FPP	Floating Point Processor
SIS	Scientific Instruction Set

The 2100 and 1000 CPUs supported user microprogramming. Under simulation, execution on these machines of all instructions in the octal ranges 101400-101777 and 105000-105777 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 undefined instruction stops. See the comments in the `hp2100_cpu0.c` source file for details.

The general command form is:

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

Options that may be specified are:

SET CPU 2114	2114C CPU
SET CPU 2115	2115A CPU
SET CPU 2116	2116C CPU
SET CPU 2100	2100A CPU
SET CPU 1000-M	1000 M-Series CPU
SET CPU 1000-E	1000 E-Series CPU
SET CPU 1000-F	1000 F-Series CPU
SET CPU 21MX-M	21MX M-series CPU (same as 1000-M)
SET CPU 21MX-E	21MX E-series CPU (same as 1000-E)
SET CPU LOADERENABLE	enable device loader
SET CPU LOADERDISABLE	disable device loader
SET CPU EAU	EAU instructions
SET CPU NOEAU	no EAU instructions
SET CPU FP	FP instructions
SET CPU NOFP	no FP instructions
SET CPU IOP	IOP instructions
SET CPU NOIOP	no IOP instructions

SET CPU FFP	FFP instructions
SET CPU NOFFP	no FFP instructions
SET CPU DMS	DMS instructions
SET CPU NODMS	no DMS instructions
SET CPU DBI	Double Integer instructions
SET CPU NODBI	no Double Integer instructions
SET CPU VIS	Vector Instruction Set
SET CPU NOVIS	no Vector Instruction Set
SET CPU SIGNAL	SIGNAL instructions
SET CPU NOSIGNAL	no SIGNAL instructions
SET CPU EMA	Extended Memory Area instructions
SET CPU NOEMA	no Extended Memory Area instructions
SET CPU VMA	Virtual Memory Area and OS instructions
SET CPU NOVMA	no Virtual Memory Area and OS instructions
SET CPU 4K	set memory size = 4K
SET CPU 8K	set memory size = 8K
SET CPU 12K	set memory size = 12K
SET CPU 16K	set memory size = 16K
SET CPU 24K	set memory size = 24K
SET CPU 32K	set memory size = 32K
SET CPU 64K	set memory size = 64K
SET CPU 128K	set memory size = 128K
SET CPU 256K	set memory size = 256K
SET CPU 512K	set memory size = 512K
SET CPU 1024K	set memory size = 1024K

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

<u>Feature</u>	<u>2114C</u>	<u>2115A</u>	<u>2116C</u>	<u>2100A</u>	<u>1000M</u>	<u>1000E</u>	<u>1000F</u>
Max mem	16K	8K	32K	32K	1024K	1024K	1024K
MP	no	no	opt	std	opt	opt	opt
DMA	opt	opt	opt	opt	opt	opt	opt
EAU	no	opt	opt	std	std	std	std
FP	no	no	no	opt	std	std	std
IOP	no	no	no	opt	opt	opt	opt
FFP	no	no	no	opt	opt	opt	std
EIG	no	no	no	no	std	std	std
DMS	no	no	no	no	opt	opt	opt
DBI	no	no	no	no	no	opt	std
EMA	no	no	no	no	no	opt	opt
VMA/OS	no	no	no	no	no	opt	opt
FPP	no	no	no	no	no	no	std
SIS	no	no	no	no	no	no	std
VIS	no	no	no	no	no	no	opt
SIGNAL	no	no	no	no	no	no	opt

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

On the 2100, the FP or FFP option and the IOP option are mutually exclusive. On the 1000-E/F, the RTE-IV EMA and RTE-6/VM VMA and OS options are mutually exclusive. On the 1000-F, the VIS and IOP options are mutually exclusive. To support CPU detection in the HP diagnostic configurator, the 1000-E/F TIMER instruction decodes as MPY on the 1000-M, and the RRR 16 instruction decodes as NOP on the 2114. The

1000-F FPP supports several undocumented instructions used by the FPP-SIS-FPP diagnostic. All other disabled or undefined instructions cause a simulator halt if STOP_INST is set and execute as NOP if not.

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 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. Enabling the loader makes this area available. Disabling the loader renders this area non-existent, so that reads from the area return zero, and writes to the area are ignored. The paper tape loader, configured for select code 10, initially resides in this area when the simulator is started. The LOAD command may be used with an appropriate absolute binary file to install a different loader.

The initial memory size is set to a maximum of 32K. If the memory size is being reduced, either by setting a smaller size or by changing to a CPU with a smaller memory capacity, 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 switch. Data in the truncated portion of memory is lost.

These switches are recognized when examining or depositing to CPU memory:

-n	interpret address without mapping (physical address)
-v	if DMS enabled, use current map, else no mapping
-s	if DMS enabled, use system map, else illegal
-u	if DMS enabled, use user map, else illegal
-p	if DMS enabled, use port A map, else illegal
-q	if DMS enabled, use port B map, else illegal

If no switch is specified, the default is to use -v, i.e., the address is virtual using the current map if DMS is enabled; otherwise the address is physical.

The simulator attempts to detect when the CPU is idle. When idle, the simulator does not use any host system processor time. Idle detection is controlled by these commands:

SET CPU IDLE	enable idle detection
SET CPU NOIDLE	disable idle detection

Idle detection is operating-system specific and is currently supported under all versions of the DOS and RTE operating systems. It is disabled by default.

The CPU implements four different kinds of instruction breakpoints:

-e	break unconditionally
-n	break if DMS is disabled
-s	break if DMS and system map are enabled
-u	break if DMS and user map are enabled

If no switch is specified, the default is to use -n if DMS 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 instruction execution is attempted.

When the simulator stops after reaching a breakpoint or completing a STEP command, the current program counter (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.

The CPU may be configured to send debugging information to the previously enabled debug output device using these commands:

```
SET CPU DEBUG           provide debug printouts
SET CPU NODEBUG        inhibit debug printouts
```

Six debug flags that control the reporting level are supported:

```
OS           Report RTE OS firmware instruction executions, except
              for time base generator interrupt instructions
OSTBG        Report RTE OS TBG firmware instruction executions
VMA          Report RTE VMA firmware instruction executions
EMA          Report RTE EMA firmware instruction executions
VIS          Report Vector Instruction Set firmware executions
SIG          Report SIGNAL firmware instruction executions
```

Because the time base generator interrupts 100 times per second and executes three firmware instructions per interrupt, a large number of debug statements can be produced very quickly. Therefore, tracing of these sequences must be enabled separately from the other operating system accelerator instructions.

CPU registers include the visible state of the processor and control registers for the interrupt system.

name	models	size	comments
P	all	15	program counter
A	all	16	A register
B	all	16	B register
M	all	15	M (memory address) register
T	all	16	T (memory data) register
X	1000	16	X index register
Y	1000	16	Y index register
S	all	16	switch/display register
E	all	1	extend flag
O	all	1	overflow flag
ION	all	1	interrupt enable flag
ION_DEFER	all	1	interrupt defer flag
CIR	all	6	central interrupt register
DMSENB	1000	1	DMS enabled flag
DMSCUR	1000	1	DMS current map (0/1 = system/user)
DMSSR	1000	16	DMS status register
DMSVR	1000	16	DMS violation register
DMSMAP[0:127]	1000	16	DMS map registers [0:31] system map [32:63] user map [64:95] port A map

			[96:127] port B map
IOPSP	2100	16	I/O processor stack pointer
STOP_INST	all	1	stop on undefined instruction
STOP_DEV	all	1	stop on undefined device
INDMAX	all	16	indirect address limit
PCQ[0:63]	all	15	origin of last programmed jump, most recent change first

BOOT CPU implements the 1000 IBL facility. IBL is controlled by the switch register S. S<15:14> selects the device to boot:

00	2748B paper-tape reader (12992K ROM)
01	7900A/2883 disk (12992A ROM)
10	7970B/E tape (12992D ROM)
11	13037D disk (12992B ROM)

For the 7900A/2883 only, S<13:12> specifies the type of disk:

00	7900A
10	2883

S<11:6> contains the device address. If the device has two addresses, S<11:6> specifies the lower address. S<5:3> are passed to the bootstrap program. S<2:0> specify options for the boot loader. IBL will not report an error if the device address in S<11:6> is incorrect.

2.2 12581A/12892B Memory Protect

Memory protect is standard equipment on the 2100 (although it may be disabled by removing a jumper) and optional on the 2116 and 1000. The following registers are implemented:

name	size	comments
CTL	1	memory protection enable
FLG	1	protection violation flag
FBF	1	protection violation flag buffer
FR	15	fence register
VR	16	violation register
EVR	1	enable violation register flag
MEV	1	memory expansion (DMS) violation flag

The 1000 memory protect card (12892B) has three feature options that are implemented by jumper settings. These are controlled by the following commands:

SET MP JSBIN	jumper W5 installed
SET MP JSBOUT	jumper W5 removed
SET MP INTIN	jumper W6 installed
SET MP INTOUT	jumper W6 removed
SET MP SEL1IN	jumper W7 installed
SET MP SEL1OUT	jumper W7 removed

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); note that I/O instructions referencing select code 1 are legal, and HLT instructions are illegal, regardless of the setting of W7.

The default configuration is JSB (W5) installed, INT (W6) installed, and SEL1 (W7) removed, providing compatibility with the 2116 and 2100 memory protect cards.

2.3 12607B/12578A/12895A DMA Controllers, 12897B DCPC Controller

The direct memory access/dual channel port controller is an option for all CPUs. The 12607B DMA controller provides one channel, DMA0, on the 2114. The other DMA/DCPC controllers provide two channels (DMA0 and DMA1 on the 2115, 2116, and 2100, or DCPC0 and DCPC1 on the 1000). Each channel has the following visible state:

name	models	size	comments
XFR	all	1	channel transfer enabled
CTL	all	1	interrupt enabled
FLG	all	1	channel ready
FBF	all	1	channel ready buffer
CTL2	all	1	command word 2/3 selector
CW1	all	16	command word 1
CW2	all	16	command word 2
CW3	all	16	command word 3
LATENCY	all	8	first-cycle delay
BYTE	12578A	1	even/odd byte-packing flag
PACKER	12578A	8	byte-packing register

2.4 Variable Device Assignments

On the HP2100, I/O devices take their device numbers (select codes) from the backplane slot they are plugged into. Thus, device number assignments vary considerably from system to system, and software package to software package. The HP2100 simulator supports dynamic device number assignment. To show the current device number, use the `SHOW <dev> DEVNO` command:

```
sim> SHOW PTR DEVNO
device=10
```

To change the device number, use the `SET <dev> DEVNO=<num>` command:

```
sim> SET PTR DEVNO=30
sim> SHOW PTR DEVNO
device=30
```

The new device number must be in the range 10-77 octal. For devices with two device numbers, `<num>` specifies the lower number; the higher is automatically set to `<num>+1`. If a device number conflict occurs, the simulator will report the error when program execution is attempted.

The hardware requires that no empty slots exist between interface cards. This is not required under simulation; empty slots behave as though they contain HP 12777A Priority Jumper Cards.

2.4.1 Device State

All devices other than the CPU and TTY may be disabled or enabled. Disabling a device simulates removing the associated interface from the CPU card cage. To disable or enable a device, use:

```
SET <dev> DISABLED          disable device
SET <dev> ENABLED           enable device
```

For devices with more than one device number, disabling or enabling any device in the set disables or enables all of the devices.

Devices consisting of multiple addressable units connected to a controller typically allow the units to be individually enabled or disabled. Disabling simulates disconnecting the associated unit from the controller. The commands to set units enabled and disabled are:

```
SET <unit> DISABLED        disable unit
SET <unit> ENABLED         enable unit
```

Some devices and units allow simulation of power-down conditions. Power settings are controlled by these commands:

```
SET <dev> POWEROFF         turn power off
SET <dev> POWERON          turn power on
```

Peripherals that provide operator-selectable disconnection, typically via an "offline" switch, provide these simulation equivalents:

```
SET <dev> OFFLINE          set peripheral offline
SET <dev> ONLINE           set peripheral online
```

2.5 Programmed I/O Devices

2.5.1 12597A-002 Duplex Register Interface (PTR) with 2748 Paper Tape Reader

The paper tape reader (PTR) reads data from a disk file. The POS register specifies the number of the next data item to be read. Thus, by changing POS, the user can backspace or advance the reader.

For diagnostic purposes, a tape loop may be simulated with the commands:

```
SET PTR DIAG               rewind tape at EOF
SET PTR READER             supply tape trailer at EOF
```

The paper tape reader supports the `BOOT` command. `BOOT PTR` copies the IBL into memory and starts it running. The switch register (S) is set automatically to the value expected by the IBL loader:

```
<15:12>    =    0000
<11:6>     =    device code
<5:3>      =    unchanged
<2:0>      =    000
```

Usage note: the `LOAD` command does not alter the A and B registers, whereas `BOOT PTR` leaves A and B set to zero after a successful load. `BOOT PTR` is preferred when loading HP software that may depend on the initial values of the A and B registers.

The paper tape reader implements these registers:

name	size	comments
BUF	8	last data item processed
CTL	1	device/interrupt enable
FLG	1	device ready
FBF	1	device ready buffer
TRLLIM	8	number of trailing nulls to append after end-of-file is detected
POS	32	position in the input file
TIME	24	time from I/O initiation to interrupt
STOP_IOE	1	stop on I/O error

The TRLLIM register specifies the number of nulls to supply as paper tape trailer when EOF is detected. If input is attempted when TRLLIM is set to zero or the count is exhausted, the reader will hang.

Error handling is as follows:

error	STOP_IOE	processed as
not attached	1	report error and stop
	0	out of tape
end of file	1	report error and stop
	0	out of tape
OS I/O error	x	report error and stop

2.5.2 12597A-005 Duplex Register Interface (PTP) with 2895 Paper Tape Punch

The paper tape punch (PTP) writes data to a disk file. The POS register specifies the number of the next data item to be written. Thus, by changing POS, the user can backspace or advance the punch.

The paper tape punch implements these registers:

name	size	comments
BUF	8	last data item processed
CTL	1	device/interrupt enable
FLG	1	device ready
FBF	1	device ready buffer
POS	32	position in the output file
TIME	24	time from I/O initiation to interrupt
STOP_IOE	1	stop on I/O error

Error handling is as follows:

error	STOP_IOE	processed as
not attached	1	report error and stop
	0	out of tape
OS I/O error	x	report error and stop

2.5.3 12531C Buffered Teleprinter Interface (TTY) with 2752 Teleprinter

The console teleprinter has three units: keyboard (unit 0), printer (unit 1), and punch (unit 2). The keyboard reads from the console keyboard; the printer writes to the simulator console window. The punch writes to a disk file. The keyboard and printer units (TTY0, TTY1) can be set to one of four modes: UC, 7P, 7B, or 8B:

mode	input characters	output characters
UC	lower case converted to upper case, high-order bit cleared	lower case converted to upper case, high-order bit cleared, non-printing characters suppressed
7P	high-order bit cleared	high-order bit cleared, non-printing characters suppressed
7B	high-order bit cleared	high-order bit cleared
8B	no changes	no changes

The default mode is UC.

Some HP software systems expect the console to transmit line-feed automatically following carriage-return. This feature is enabled with:

```
SET TTY AUTOLF
```

and disabled with:

```
SET TTY NOAUTOLF
```

The console teleprinter implements these registers:

name	size	comments
BUF	8	last data item processed
MODE	16	mode
CTL	1	device/interrupt enable
FLG	1	device ready
FBF	1	device ready buffer
KPOS	32	number of characters input
KTIME	24	keyboard polling interval
TPOS	32	number of characters printed
TTIME	24	time from I/O initiation to interrupt
PPOS	32	position in the punch output file
STOP_IOE	1	punch stop on I/O error

Error handling for the punch is as follows:

error	STOP_IOE	processed as
not attached	1	report error and stop
	0	out of tape
OS I/O error	x	report error and stop

2.5.4 12966A Buffered Asynchronous Communications Interface (BACI)

The BACI was designed specifically to interface HP 264x and HP 262x CRT terminals to the HP 1000 via hardwired and modem connections. It contains a 128-byte FIFO to support block-mode reads from the terminals. Under simulation, it connects with an HP terminal emulator via Telnet to a user-specified port. The `ATTACH` command specifies the port to be used, and `DETACH` disconnects the port:

```
ATTACH BACI <port>          set up listening port
DETACH BACI                  disconnect port
```

where `<port>` is a decimal number between 1 and 65535 that is not being used for other TCP/IP activities. Once the BACI is attached and the simulator is running, the specified port will be monitored for Telnet connections. The port may be forcibly disconnected from its associated Telnet session with:

```
SET BACI DISCONNECT          terminate Telnet session
```

Connections remain open until disconnected by the Telnet client, by a `DETACH BACI` command, or by a `SET BACI DISCONNECT` command.

A diagnostic mode is provided for use with the 12966-16001 Asynchronous Data Set diagnostic:

```
SET BACI DIAG                configure for diagnostic tests
SET BACI TERMINAL            configure for terminal connections
```

Enabling the diagnostic mode simulates the installation of the HP 12966-60003 diagnostic test (loopback) connector onto the 12966A card. Enabling terminal mode simulates the installation of the HP 12966-60008 (264x terminal) connector and RS-232 cable.

The BACI supported fifteen programmable baud rates from 50 to 9600 baud, as well as an external rate determined by the connected terminal, which was usually set for 9600 baud as well. The simulator can be set to model the actual transmission times (in machine instructions elapsed per character), or it can be set to operate substantially faster with these commands:

```
SET BACI REALTIME            use realistic timing
SET BACI FASTTIME            use optimized timing
```

Setting the BACI to fast timing 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.

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:

```
SET BACI CAPSLOCK            upshift lower case input to upper case
SET BACI NOCAPSLOCK          characters are processed as received
```

Logging of BACI output to a file may be turned on and off with the following commands:

```
SET BACI LOG=<filename>      log terminal output to a file
SET BACI NOLOG                terminate output logging
```

Information about BACI operation is provided by these commands:

SHOW BACI CONNECTION	displays active connection address and time
SHOW BACI STATISTICS	displays active connection I/O statistics

The BACI may be configured to send debugging information to the previously enabled debug output device by using these commands:

SET BACI DEBUG{=f1;f2;...}	provide debug printouts
SET BACI NODEBUG{=f1;f2;...}	inhibit debug printouts
SHOW BACI DEBUG	display current debug settings

Four debug flags that control various levels of reporting are supported:

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 received and transmitted via Telnet

If individual flags are not specified in the SET BACI DEBUG and SET BACI NODEBUG commands, then all reporting levels are implied.

The BACI implements these registers:

name	size	comments
IBUF	16	input buffer (from card to CPU)
OBUF	16	output buffer (from CPU to card)
STATUS	16	card status word
EDSIW	16	enable device status interrupt word
DSRW	16	device status reference word
CFCW	16	character frame control word
ICW	16	interface control word
ISRW	16	interrupt status reset word
FIFOPUT	8	FIFO load index
FIFOGET	8	FIFO unload index
FIFOCNTR	8	FIFO character counter
BRKCNTR	16	BREAK clock counter
FIFO[0:127]	8	FIFO buffer
SPCHAR[0:255]	1	special character RAM
UARTTHR	16	UART transmitter holding register
UARTTR	16	UART transmitter register
UARTHR	16	UART receiver holding register
UARTRR	16	UART receiver register
UARTCLK	16	UART transmitter/receiver clock counter
CTIME	19	inter-character poll time
LKO	1	interface interrupt lockout
CTL	1	interface control
FLG	1	interface flag
FBF	1	interface flag buffer
SRQ	1	interface DMA service request

While the BACI can connect to generic Telnet clients, RTE drivers for the BACI required the use of HP terminals. *QCTerm*, a free HP 700/92 terminal emulator for Windows, is available from AICS Research at <http://www.aics-research.com/qcterm/>.

2.5.5 12653A Printer Controller (LPS) with 2767 Line Printer, 12566B Microcircuit Interface with Loopback Connector

The 2767 line printer uses the 12653A line printer interface as its controller. As a line printer, LPS writes data to a disk file. The POS register specifies the number of the next data item to be written. Thus, by changing POS, the user can backspace or advance the printer.

The line printer responds to `SET LPS POWEROFF` as if the power were removed or the printer cable were disconnected, and to `DETACH LPS` as if the paper were out. It also provides these additional state commands:

<code>SET LPS OFFLINE</code>	simulate ONLINE button up
<code>SET LPS ONLINE</code>	simulate ONLINE button down
<code>SET LPS REALTIME</code>	use realistic timing for print operations
<code>SET LPS FASTTIME</code>	use optimized timing for print operations

As a 12566B microcircuit interface, LPS provides the test device for running several of the HP diagnostics. Printer mode versus diagnostic mode is controlled by the commands:

<code>SET LPS PRINTER</code>	configure as line printer
<code>SET LPS DIAG</code>	configure for diagnostic tests

In diagnostic mode, LPS simulates the installation of the HP 1251-0332 diagnostic test (loopback) connector onto the 12566B card and the setting of the appropriate card jumpers for the diagnostic.

LPS may be configured to send debugging information to the previously enabled debug output device using these commands:

<code>SET LPS DEBUG</code>	provide debug printouts
<code>SET LPS NODEBUG</code>	inhibit debug printouts

Diagnostic information includes characters supplied to and status received from the interface, as well as data transfer initiations and completions.

The 12653A is disabled by default.

The 12653A implements these registers:

name	size	comments
BUF	16	output buffer
STA	16	input buffer or status
POWER	2	printer power state (read-only)
CTL	1	device/interrupt enable
FLG	1	device ready
FBF	1	device ready buffer
CCNT	7	current character count
LCNT	7	current line count
POS	32	position in the output file
CTIME	24	character transfer time

PTIME	24	per-zone print operation time
STIME	24	per-line paper slew time
RTIME	24	power-on ready delay time
STOP_IOE	1	stop on I/O error

In printer mode, error handling is as follows:

error	STOP_IOE	processed as
not attached	1	report error and stop
	0	out of paper
SET POWEROFF	1	report error and stop
	0	powered off
SET OFFLINE	1	report error and stop
	0	offline
OS I/O error	x	report error and stop

With STOP_IOE set to 0, output performed when the device is powered off or offline will initiate but then hang, waiting for the device to be returned online. When it is, the output operation will complete.

In diagnostic mode, there are no errors; data sent to the output buffer is looped back to the status register immediately.

2.5.6 12845B Printer Controller (LPT) with 2607 Line Printer

The line printer (LPT) writes data to a disk file. The POS register specifies the number of the next data item to be written. Thus, by changing POS, the user can backspace or advance the printer.

The line printer responds to SET LPT POWEROFF as if the power were removed or the printer cable were disconnected and DETACH LPT as if the paper were out. It also provides these additional state commands:

SET LPT OFFLINE	simulate PAPER button up
SET LPT ONLINE	simulate PAPER button down

The line printer implements these registers:

name	size	comments
BUF	7	last data item processed
CTL	1	device/interrupt enable
FLG	1	device ready
FBF	1	device ready buffer
LCNT	7	line count within page
POS	32	position in the output file
CTIME	24	time between characters
PTIME	24	time for a print operation
STOP_IOE	1	stop on I/O error

Error handling is as follows:

error	STOP_IOE	processed as
not attached	1	report error and stop

	0	out of paper
SET POWEROFF	1	report error and stop
	0	powered off
SET OFFLINE	1	report error and stop
	0	offline
OS I/O error	x	report error and stop

With STOP_IOE set to 0, output performed when the device is powered off or offline will initiate but then hang, waiting for the device to be returned online. When it is, the output operation will complete.

2.5.7 12539C Time Base Generator (CLK)

The time base generator (CLK) may be set for diagnostic mode:

SET CLK DIAG	configure for diagnostic mode
SET CLK CALIBRATED	configure for calibrated timing mode

Diagnostic mode corresponds to setting jumper W2 to position "B." This turns off autocalibration and divides the longest time intervals down by 1000.

The time base generator implements these registers:

name	size	comments
SEL	3	time base select
CTR	14	repeat counter for < 1Hz operation
CTL	1	device/interrupt enable
FLG	1	device ready
FBF	1	device ready buffer
ERR	1	error flag
TIME[0:7]	31	clock intervals, select = 0-7
IPTICK	24	calculated CPU instructions per clock tick

The time base generator autocalibrates; the clock interval is adjusted up or down so that the clock tracks actual elapsed time. Operation at the fastest rates (100 microseconds, 1 millisecond) is not recommended.

2.5.8 12792C 8-Channel Asynchronous Multiplexer (MPX)

The 12792C was an eight-line asynchronous serial multiplexer that connected terminals, modems, serial printers, and other devices that used the RS-232 standard to the CPU. It used an on-board Z80 microprocessor and provided input and output buffering to support block-mode reads from HP 264x and 262x terminals. The card handled character editing, echoing, and read-terminator detection, reducing the load on the CPU. Under simulation, it connects with HP terminal emulators via Telnet to a user-specified port.

2.5.8.1 MPX Device Commands

The ATTACH command specifies the Telnet port to be used to establish connections, and DETACH disconnects both the multiplexer control and all line sessions:

ATTACH MPX <port>	set up listening port
DETACH MPX	disconnect multiplexer and all lines

...where <port> is a decimal number between 1 and 65535 that is not being used for other TCP/IP activities. Once the multiplexer is attached and the simulator is running, the specified port will be monitored for Telnet connections.

The order in which connections are assigned to multiplexer lines may be specified and displayed with these commands:

```
SET MPX LINEORDER=l1{;l2;...} set line connection order
SHOW MPX LINEORDER          show line connection order
```

...where <l_n> may be a single line number or a range of line numbers of the form m-n. Telnet connections to the listening port will be assigned to multiplexer lines in the sequence specified. In the absence of a SET MPX LINEORDER command, connections will be assigned by default in ascending line order. The default order may be reestablished by specifying this command:

```
SET MPX LINEORDER=ALL      restore default ascending connection order
```

A multiplexer line may be forcibly disconnected from its associated Telnet session with:

```
SET MPX DISCONNECT=<n>    terminate Telnet session on line <n>
```

Connections remain open until disconnected by the Telnet client, by a DETACH MPX command, or by a SET MPX DISCONNECT command.

Information about multiplexer operation is provided by these commands:

```
SHOW MPX CONNECTIONS      displays connection addresses and times
SHOW MPX STATISTICS        displays connection I/O statistics
```

The multiplexer may be configured to send debugging information to the previously enabled debug output device by using these commands:

```
SET MPX DEBUG{=f1;f2;...} provide debug printouts
SET MPX NODEBUG{=f1;f2;...} inhibit debug printouts
SHOW MPX DEBUG             display current debug settings
```

Four debug flags that control various levels of reporting are supported:

```
CMDS      Command initiations and completions
CPU       Words received from and sent to the CPU
BUF       Data read from and written to line buffers
XFER      Data received and transmitted via Telnet
```

If individual flags are not specified in the SET MPX DEBUG and SET MPX NODEBUG commands, then all reporting levels are implied.

2.5.8.2 MPX Line Commands

The 12792C supported twelve programmable baud rates from 50 to 19200 baud. The simulator can be set to model the actual transmission times (in machine instructions elapsed per character), or it can be set to operate substantially faster with these commands:

```
SET MPX<n> REALTIME      use realistic timing
SET MPX<n> FASTTIME      use optimized timing
```

...where <n> is the desired line number. Setting the multiplexer to fast timing 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, 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:

```
SET MPX<n> CAPSLOCK          upshift lower case input to upper case
SET MPX<n> NOCAPSLOCK       characters are processed as received
```

Logging of multiplexer output to a file may be turned on and off with the following commands:

```
SET MPX<n> LOG=<filename>    log line <n> output to a file
SET MPX<n> NOLOG             terminate output logging
```

2.5.8.3 MPX Implementation

The multiplexer is simulated with ten units:

unit	function
0	I/O line 0
1	I/O line 1
2	I/O line 2
3	I/O line 3
4	I/O line 4
5	I/O line 5
6	I/O line 6
7	I/O line 7
8	Card controller
9	Telnet line poll

Units 8 and 9 are disabled so that they do not appear in the normal unit list, but they will appear in the simulator event queue.

There were four major revisions to the multiplexer firmware, designated revisions A-D. Revisions A-C had an upward-compatible command set and were supported under RTE-M, RTE-IVB, and RTE-6/VM. Revision D implemented completely different commands, required different RTE drivers, and was 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 (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`, the multiplexer would enter "echoplex" mode. In this mode, each line would default to 9600 baud, 8 bits per character, 1 stop bit, no parity, and would echo characters as received. Echoplex is implemented under simulation; until a line is configured by the appropriate RTE driver control call, characters entered in a Telnet session will be echoed.

The 12792C had two baud-rate generators that were assigned to lines by the wiring configuration in the I/O cable connector hood. Two counter/timer circuits were used to implement the BRGs for all eight lines. For lines connected to the same BRG, only subsets of the configurable rates were allowed, and assigning

mutually incompatible rates caused corruption of the rates on lines assigned earlier. Under simulation, any baud rate may be assigned to any line without interaction.

While the multiplexer can connect to generic Telnet clients, RTE programs generally assumed the use of HP terminals. *QCTerm*, a free HP 700/92 terminal emulator for Windows, is available from AICS Research at <http://www.aics-research.com/qcterm/>.

The multiplexer does not support save and restore. All open connections are lost when the simulator shuts down.

2.5.8.4 MPX Registers

The multiplexer implements these registers:

name	size	comments
STATE	3	controller state (idle, cmd, param, exec)
IBUF	16	input buffer (from multiplexer to CPU)
OBUF	16	output buffer (from CPU to multiplexer)
CMD	8	command code
PARAM	16	command parameter
PORT	8	port number for last command
PORTKEY	8	port key for last command
IOLEN	16	I/O transfer length for last command
UIEN	1	unsolicited interrupts enabled
UIPORT	3	unsolicited interrupt port
UICODE	3	unsolicited interrupt reason code
KEYS[0:7]	8	port keys, ports 0-7
PCONFIG[0:7]	16	port configuration, ports 0-7
RCVTYPE[0:7]	16	receive type, ports 0-7
CHARCNT[0:7]	16	character count, ports 0-7
FLOWCNTL[0:7]	16	flow control, ports 0-7
ENQCNTN[0:7]	7	ENQ character counter, ports 0-7
ACKWAIT[0:7]	10	ACK wait counter, ports 0-7
PFLAGS[0:7]	12	port flags, ports 0-7
RBUF[0:7][0:513]	8	read buffers, ports 0-7
WBUF[0:7][0:513]	8	write buffers, ports 0-7
GET[0:7][0:1]	16	read/write get indexes, ports 0-7
SEP[0:7][0:1]	16	read/write separator indexes, ports 0-7
PUT[0:7][0:1]	16	read/write put indexes, ports 0-7
CTL	1	interface control
FLG	1	interface flag
FBF	1	interface flag buffer

2.5.9 12920A 16-Channel Terminal Multiplexer (MUX, MUXL, MUXM)

The 12920A is a 16-line terminal multiplexer, with five additional receive-only diagnostic lines. It consists of three devices:

MUX	scanning logic (corresponds to the upper data card)
MUXL	individual lines (corresponds to the lower data card)
MUXM	modem control and status logic (corresponds to the control card)

The MUX performs input and output through Telnet sessions connected to a user-specified port. The `ATTACH` command to the scanning logic specifies the port to be used:

```
ATTACH MUX <port>          set up listening port
```

where <port> is a decimal number between 1 and 65535 that is not being used for other TCP/IP activities.

Each line (each unit of MUXL) can be set to one of four modes: UC, 7P, 7B, or 8B:

mode	input characters	output characters
UC	lower case converted to upper case, high-order bit cleared	lower case converted to upper case, high-order bit cleared, non-printing characters suppressed
7P	high-order bit cleared	high-order bit cleared, non-printing characters suppressed
7B	high-order bit cleared	high-order bit cleared
8B	no changes	no changes

The default mode is UC. If the MUX 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 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.

In addition, each line supports the DATASET option:

```
SET MUXLn DATASET          enable modem control on line n
SET MUXLn NODATASET       disable modem control on line n
```

When modem control is enabled, a connected line that has DTR (command bit C1) high will set CD (status bit S2) in the line status, and a programmed drop of the DTR line will disconnect the Telnet session.

Finally, each line supports output logging:

```
SET MUXLn LOG=filename     log output of line n to filename
SET MUXLn NOLOG           terminate output logging of line n
```

Terminating logging also closes the log file.

The modem controls model a simplified Bell 103A dataset with just four lines: data terminal ready and request to send from the computer to the data set, and carrier detect and data set ready from the data set to the computer. There is no ring detection. A new Telnet connection sets data set ready and is established whether data terminal ready is set or not.

Once MUX is attached and the simulator is running, the multiplexer listens for connections on the specified port. It assumes that the incoming connections are Telnet connections. The connections remain open until disconnected either by the Telnet client, a SET MUXL DISCONNECT command, or a DETACH MUX command.

The order in which connections are assigned to multiplexer lines may be specified and displayed with these commands:

```
SET MUX LINEORDER=11{;12;...} set line connection order
SHOW MUX LINEORDER          show line connection order
```

...where <ln> may be a single line number or a range of line numbers of the form m-n. Telnet connections to the listening port will be assigned to multiplexer lines in the sequence specified. In the absence of a SET

MUX LINEORDER command, connections will be assigned by default in ascending line order. The default order may be reestablished by specifying the command:

```
SET MUX LINEORDER=ALL           restore default ascending connection order
```

Other special multiplexer commands:

```
SET MUX DIAG                     configure for diagnostic tests
SET MUX TERM                     configure for terminal connections
SHOW MUX CONNECTIONS            show current connections
SHOW MUX STATISTICS             show statistics for active connections
SET MUXLn DISCONNECT            disconnects the specified line
```

Enabling the diagnostic mode simulates the installation of eight HP 30062-60003 diagnostic test (loopback) cables between lines 0-1, 2-3, etc., as required by the multiplexer diagnostic. In addition, all Telnet sessions are disconnected, and the multiplexer is detached from the 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 multiplexer may be configured to send debugging information to the previously enabled debug output device using these commands:

```
SET MUX DEBUG                   provide debug printouts
SET MUX NODEBUG                inhibit debug printouts
```

Three debug flags that control the reporting level are supported:

```
CMDS                           Command initiations and completions
CPU                             Words received from and sent to the CPU
XFER                            Data transferred via Telnet or loopback
```

The scanner (MUX) implements these registers:

name	size	comments
IBUF	16	input buffer, holds line status
OBUF	16	output buffer, holds channel select

The lines (MUXL) implements these registers:

name	size	comments
CTL	1	device/interrupt enable
FLG	1	device ready
FBF	1	device ready buffer
STA[0:20]	16	line status, lines 0-20
RPAR[0:20]	16	receive parameters, lines 0-20
XPAR[0:15]	16	transmit parameters, lines 0-15
RBUF[0:20]	16	receive buffer, lines 0-20
XBUF[0:15]	16	transmit buffer, lines 0-15
RCHP[0:20]	1	receive character present, lines 0-20
XDON[0:15]	1	transmit done, lines 0-15
BDFR[0:15]	1	break deferred, lines 0-15
TIME[0:15]	24	transmit time, lines 0-15

The modem control (MUXM) implements these registers:

name	size	comments
CTL	1	device/interrupt enable
FLG	1	device ready
FBF	1	device ready buffer
SCAN	1	scan enabled
CHAN	4	current line
DSO[0:15]	6	C2,C1,ES2,ES1,SS2,SS1, lines 0-15
DSI[0:15]	2	S2,S1, lines 0-15

The terminal multiplexer does not support save and restore. All open connections are lost when the simulator shuts down or MUX is detached.

2.5.10 12875A Interprocessor Link (IPLI, IPLO)

The interprocessor link is a pair of 12566A parallel interfaces that are cross-coupled to provide interprocessor communications to a second copy of the HP2100 simulator. The IPL is intended to support simulation of a two-system HP 2000 Time-Shared BASIC configuration and occupies two adjacent I/O slots. The input-line interface is installed in the higher-priority (lower-numbered) slot, and the output-line interface is installed in the lower-priority (higher-numbered) slot. Each interface is actually a bi-directional, half-duplex line that is used in the indicated direction for commands and in the reverse direction for status. The IPL is disabled by default.

To operate, the IPL devices must be enabled and then connected to the IPL devices in another copy of the simulator. The IPLI device in the first simulator is connected to the IPLO device in the second, and vice versa. Connections are established with the `ATTACH` command. One copy of the simulator listens for connections on the specified ports (`ATTACH -L`), and the other establishes connections to those ports (`ATTACH -C`). Either copy may perform either operation, but the operations must be done in matched pairs:

```

simulator #1                      simulator #2

sim> SET IPLI ENABLED              sim> SET IPLI ENABLED
(also enables IPLO)              (also enables IPLO)

sim> ATTACH -LW IPLI 4000          sim> ATTACH -C IPLO 4000
Listening on port 4000            Connecting to IP address 127.0.0.1, port 4000
Waiting for connection

Connection established

sim> ATTACH -LW IPLO 4001          sim> ATTACH -C IPLI 4001
Listening on port 4001            Connecting to IP address 127.0.0.1, port 4001
Waiting for connection

Connection established

```

Both forms of `ATTACH` take a modifier `-w` (wait); if specified, the command will wait up to 30 seconds for the connection process to complete. `ATTACH -C` can specify both an IP address and a port, in the form `aa.bb.cc.dd:port`; if the IP address is omitted, it defaults to 127.0.0.1 (local system). The connections may be forcibly broken with the `SET IPLI DISCONNECT` and `SET IPLO DISCONNECT` commands.

Both IPLI and IPLO implement the `BOOT` command. `BOOT` loads the HP 2000/Access Basic Block Loader for the IOP into the top 64 words of memory and starts it running.

The interprocessor link supports the HP processor interconnect cable diagnostic with these commands:

```
SET IPLI DIAG          configure for diagnostic tests
SET IPLI LINK          configure for interprocessor connections
```

Enabling the diagnostic mode simulates the connection of the interprocessor cable from one card to the other within the same machine. Configuring either IPLI or IPLO for diagnostic mode automatically configures the other card as well.

Each link device may be configured separately to send debugging information to the previously enabled debug output device using these commands:

```
SET IPLI DEBUG        provide input-side debug printouts
SET IPLI NODEBUG      inhibit input-side debug printouts
SET IPLO DEBUG        provide output-side debug printouts
SET IPLO NODEBUG      inhibit output-side debug printouts
```

Three debug flags that control the reporting level are supported:

```
CMDS          Command initiations and completions
CPU           Words received from and sent to the CPU
XFER          Data transferred via the link
```

Both IPLI and IPLO implement these registers:

name	size	comments
IBUF	16	input buffer
OBUF	16	output buffer
CTL	1	device/interrupt enable
FLG	1	device ready
FBF	1	device ready buffer
HOLD	8	holding buffer
TIME	24	polling interval for input
STOP_IOE	1	stop on I/O error

2.5.11 12620A/12936A Privileged Interrupt Fence (PIF)

The Privileged Interrupt Fence (PIF) was used in DOS and RTE systems to provide privileged interrupt capability. The PIF was 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, were allowed to interrupt the operating system and service routines of the unprivileged cards that were located above the fence. Privileged devices employed specially written device drivers that bypassed the operating system during interrupts. This provided 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 behaved quite differently and were not interchangeable. Under simulation, the type of the PIF card may be selected and displayed with these commands:

```
SET PIF 12620A        simulate 12620A RTE fence
SET PIF 12936A        simulate 12936A DOS fence
SHOW PIF TYPE         display current fence type
```

The PIF may be configured to send debugging information to the previously enabled debug output device by using these commands:

```

SET PIF DEBUG           provide debug printouts
SET PIF NODEBUG        inhibit debug printouts
SHOW PIF DEBUG         display current debug state

```

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

The PIF implements these registers:

name	size	comments
CTL	1	interface control
FLG	1	interface flag
FBF	1	interface flag buffer

2.6 Disk Controllers

2.6.1 12557A Disk Controller (DPC, DPD) with Four 2781 Drives, 13210A Disk Controller (DPC, DPD) with Four 7900 Drives

The 12557A/13210A disk controller has two separate devices, a data channel (DPD) and a command channel (DPC). The data channel includes a 128-word (one sector) buffer for reads and writes. The command channel includes the four disk drives. The 12557A/13210A controller can be configured as either a 12557A, supporting 2.5MB drives, or a 13210A, supporting 5MB drives, with the commands:

```

SET DPC 12557A         provide 2781 (2.5MB) drives
SET DPC 13210A        provide 7900 (5.0MB) drives
SHOW DPC TYPE         display current controller type

```

Drive types cannot be intermixed; the controller is configured for one type or the other. The 13210A is selected by default.

Individual drives may be protected against writing. These commands simulate the Upper/Lower Disc Protect switches on the drives:

```

SET DPCn LOCKED       set unit n write locked
SET DPCn WRITEENABLED set unit n write enabled

```

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.

Drives may also have their heads unloaded and loaded:

```

SET DPCn UNLOADED     unload heads on unit n
SET DPCn LOADED       load heads on unit n

```

This provides a convenient method of setting a drive "down" without detaching the associated disk image file. Drives can also be set `DISABLED` or `ENABLED`.

The 12557A/13210A supports the `BOOT` command. `BOOT DPC` copies the IBL for 7900 class disks into memory and starts it running. `BOOT -R DPC` boots from the removable platter (head 0). The switch register (S) is set automatically to the value expected by the IBL loader:

```

<15:14>    =    01
<13:12>    =    00
<11:6>     =    data channel device code
<5:3>      =    unchanged
<2:1>      =    00
<0>        =    1 if booting from the removable platter

```

The data channel (DPD) implements these registers:

name	size	comments
IBUF	16	input buffer
OBUF	16	output buffer
DBUF[0:127]	16	sector buffer
BPTR	7	sector buffer pointer
CMD	1	channel enable
CTL	1	interrupt enable
FLG	1	channel ready
FBF	1	channel ready buffer
XFER	1	transfer in progress flag
WVAL	1	write data valid flag

The command channel (DPC) implements these registers:

name	size	comments
OBUF	16	output buffer
BUSY	4	busy (unit #, + 1, of active unit)
CNT	5	check record count
CMD	1	controller enable
CTL	1	interrupt enable
FLG	1	controller ready
FBF	1	controller ready buffer
EOC	1	end of cylinder pending
POLL	1	attention polling enabled
RARC	8	record address register (cylinder)
RARH	2	record address register (head)
RARS	4	record address register (sector)
CYL[0:3]	8	current cylinder, drives 0-3
STA[0:3]	16	drive status, drives 0-3
CTIME	24	data transfer command delay time
DTIME	24	data channel command delay time
STIME	24	seek delay time, per cylinder
XTIME	24	inter-word transfer time

Error handling is as follows:

error	processed as
not attached	disk not ready (heads unloaded)
end of file	assume rest of disk is zero

OS I/O error report error and stop

2.6.2 12565A Disk Controller (DQC, DQD) with Two 2883 Drives

The 12565A disk controller has two separate devices, a data channel (DQD) and a command channel (DQC). The data channel includes a 128-word (one sector) buffer for reads and writes. The command channel includes the two disk drives.

Individual drives may be protected against writing:

```
SET DQcN LOCKED                    set unit n write locked
SET DQcN WRITEENABLED            set unit n write enabled
```

Drives may have their heads unloaded and loaded:

```
SET DQcN UNLOADED                unload heads on unit n
SET DQcN LOADED                 load heads on unit n
```

This provides a convenient method of setting a drive "down" without detaching the associated disk image file. Disk drives can also be set `DISABLED` or `ENABLED`.

The 12565A supports the `BOOT` command. `BOOT DQC` copies the IBL for 2883 class disks into memory and starts it running. The switch register (S) is set automatically to the value expected by the IBL loader:

```
<15:12>        =        0110
<11:6>         =        data channel device code
<5:3>          =        unchanged
<2:0>          =        000
```

The data channel (DQD) implements these registers:

name	size	comments
IBUF	16	input buffer
OBUF	16	output buffer
DBUF[0:127]	16	sector buffer
BPTR	7	sector buffer pointer
CMD	1	channel enable
CTL	1	interrupt enable
FLG	1	channel ready
FBF	1	channel ready buffer
XFER	1	transfer in progress flag
WVAL	1	write data valid flag

The command channel (DQC) implements these registers:

name	size	comments
OBUF	16	output buffer
BUSY	2	busy (unit # + 1 of active unit)
CNT	9	check record count
CMD	1	controller enable
CTL	1	interrupt enable
FLG	1	controller ready
FBF	1	controller ready buffer

RARC	8	record address register (cylinder)
RARH	5	record address register (head)
RARS	5	record address register (sector)
CYL[0:1]	8	current cylinder, drives 0-1
HED[0:1]	5	current head, drives 0-1
STA[0:1]	16	drive status, drives 0-1
CTIME	24	data transfer command delay time
DTIME	24	data channel command delay time
STIME	24	seek delay time, per cylinder
XTIME	24	inter-word transfer time

Error handling is as follows:

error	processed as
not attached	disk not ready (heads unloaded)
end of file	assume rest of disk is zero
OS I/O error	report error and stop

2.6.3 12606B Fixed Head Disk Controller (DRC, DRD) with 2770/2771 Disk, 12610B Drum Controller (DRC, DRD) with 2773/2774/2775 Drum

The 12606B/12610B fixed head disk/drum controller has two separate devices, a data channel (DRD) and a command channel (DRC). The command channel includes the actual drive. Ten different models are supported:

command	interface and size	model
SET DRC 180K	12606B, 180K words	2770A
SET DRC 360K	12606B, 360K words	2771A
SET DRC 720K	12606B, 720K words	2771A-001
SET DRC 384K	12610B, 384K words	2773A
SET DRC 512K	12610B, 512K words	2773A-001
SET DRC 640K	12610B, 640K words	2773A-002
SET DRC 768K	12610B, 768K words	2774A
SET DRC 896K	12610B, 896K words	2774A-001
SET DRC 1024K	12610B, 1024K words	2774A-002
SET DRC 1536K	12610B, 1536K words	2775A

The command channel supports write-protected tracks. Track protection is enabled with this command:

```
SET DRC PROTECTED
```

In addition, the number of protected tracks is specified by the command:

```
SET DRC TRACKPROT=count
```

The track protect count must be a power of two from 1 to 128 on the 12606 interface and from 1 to 512, or 768, on the 12610 interface. If the drive has fewer tracks than the track protect count, then all tracks on the drive are eligible for protection.

Track protection is disabled with this command:

```
SET DRC UNPROTECTED
```

The 12606B/12610B support the `BOOT` command. `BOOT DRD` loads the first sector from the disk or drum into locations 0-77 and then jumps to 77. This is very different from the IBL loader protocol used by the 12565A and the 12557A/13210A.

The data channel (DRD) implements these registers:

name	size	comments
IBUF	16	input buffer
OBUF	16	output buffer
CTL	1	interrupt enable
FLG	1	channel ready
BPTR	6	sector buffer pointer

The command channel implements these registers:

name	size	comments
CW	16	command word
STA	16	status
RUN	1	run flip-flop
TIME	24	inter-word transfer time
STOP_IOE	1	stop on I/O error

Error handling is as follows:

error	processed as
not attached	drum or disk not ready

12606B/12610B data files are buffered in memory; therefore, end of file and OS I/O errors cannot occur.

2.6.4 13037D Disk Controller (DS) with Eight 7905/7906/7920/7925 Drives

The 13037D disk controller supports 7905 (15MB), 7906 (20MB), 7920 (50MB), or 7925 (120MB) disk drives, as well as autosizing, based on the size of the disk image file:

```

SET DSn 7905           drive n is a 15MB drive
SET DSn 7906           drive n is a 20MB drive
SET DSn 7920           drive n is a 50MB drive
SET DSn 7925           drive n is a 120MB drive
SET DSn AUTOSIZE       drive n type based on file size at ATTACH

```

Drive types can be intermixed. The 7905 is selected by default.

Individual drives may be protected against writing. These commands simulate the Disc Protect/Read Only switches on the drives:

```

SET DSn LOCKED         set unit n write locked
SET DSn WRITEENABLED   set unit n write enabled

```

Separate protection for the upper and lower platters of the 7905 and 7906 drives is not supported. Protecting a 7905 or 7906 drive behaves as though both of the Disc Protect switches were on.

Drives may also have their heads unloaded and loaded:

```

SET DSn UNLOADED          unload heads on unit n
SET DSn LOADED            load heads on unit n

```

This provides a convenient method of setting a drive "down" without detaching the associated disk image file.

The setting of the drive Format switch may be changed with:

```

SET DSn FORMAT            set format enabled
SET DSn NOFORMAT         set format disabled

```

Drives can also be set DISABLED or ENABLED.

The 13037D supports the `BOOT` command. `BOOT DS` copies the IBL loader for the 13037D controller into memory and starts it running. The switch register (S) is set automatically to the value expected by the IBL loader:

```

<15:14>    =    11
<13:12>    =    01
<11:6>     =    data channel device code
<5:3>      =    unchanged
<2>        =    0
<1:0>      =    unchanged (head number)

```

The DS controller implements these registers:

name	size	comments
CMD	16	command register
FIFO[0:15]	16	data FIFO
SR1	16	status register 1
VCTR	16	verify counter
FMASK	8	file mask
CYL	16	cylinder address register
HS	16	head/sector address register
STATE	2	controller state
LASTA	3	last unit polled for attention flag
FIP	4	FIFO insertion pointer
FRP	4	FIFO removal pointer
FCNT	5	FIFO counter
CTL	1	interrupt enable
FLG	1	ready flag
FBF	1	ready flag buffer
SRQ	1	DMA service request
BUSY	1	visible busy status
CMDF	1	command follows flag
CMDP	1	command pending flag
EOC	1	end of cylinder flag
EOD	1	end of data flag
DBUF[0:127]	16	sector buffer
DPTR	8	sector buffer pointer
CTIME	24	command response time
DTIME	24	data transfer response time
STIME	24	seek time (per cylinder)
RTIME	24	rotation time
TIMEOUT	31	controller timeout

Error handling is as follows:

error	processed as
not attached	disk not ready (heads unloaded)
end of file	assume rest of disk is zero
OS I/O error	report error and stop

2.7 Magnetic Tape

2.7.1 12559C Magnetic Tape Controller (MTC, MTD) with One 3030 Drive

The 12559C magnetic tape drive controller has two separate devices, a data channel (MTD) and a command channel (MTC). The data channel includes a maximum record sized buffer for reads and writes. The command channel includes the tape unit and supports setting the tape image file format. Magnetic tape options include the ability to make the unit write enabled or write locked:

SET MTC LOCKED	set unit write locked
SET MTC WRITEENABLED	set unit write enabled

The `BOOT` command is not supported. The 12559C is disabled by default.

The data channel (MTD) implements these registers:

name	size	comments
FLG	1	channel ready
FBF	1	channel ready buffer
DBUF[0:65535]	8	transfer buffer
BPTR	16	buffer pointer (reads and writes)
BMAX	16	buffer size (writes)

The command channel (MTC) implements these registers:

name	size	comments
FNC	8	current function
STA	9	tape status
BUF	8	buffer
CTL	1	interrupt enabled
FLG	1	controller ready
FBF	1	controller ready buffer
DTF	1	data transfer flop
FSVC	1	first service flop
POS	32	tape position
CTIME	24	command delay time
GTIME	24	inter-record gap traversal time
XTIME	24	inter-word transfer delay time
STOP_IOE	1	stop on I/O error

Error handling is as follows:

error	processed as
not attached	tape not ready; if STOP_IOE, stop
end of file	parity error
OS I/O error	parity error; if STOP_IOE, stop

2.7.2 13181A Magnetic Tape Controller (MSC, MSD) with Four 7970B Drives, 13183A Magnetic Tape Controller (MSC, MSD) with Four 7970E Drives

The 13181A/13183A magnetic tape drive controller has two separate devices, a data channel (MTD) and a command channel (MTC). The data channel includes a maximum record sized buffer for reads and writes. The command channel includes the tape units and supports setting the tape image file format. Magnetic tape options include the ability to select the controller type and timing, drive tape reel size, and the ability to set drives offline, online, write enabled, or write locked:

SET MSC 13181A	provide 7970B (800 BPI) drives
SET MSC 13183A	provide 7970E (1600 BPI) drives
SHOW MSC TYPE	display current controller type
SET MSC REALTIME	set controller to actual timing
SET MSC FASTTIME	set controller to optimized timing (default)
SET MSCn OFFLINE	set unit n offline
SET MSCn ONLINE	set unit n online
SET MSCn LOCKED	set unit n write locked
SET MSCn WRITEENABLED	set unit n write enabled
SET MSCn REEL=length	set unit tape reel size 0 = unlimited (default) 600 = 600 feet 1200 = 1200 feet 2400 = 2400 feet
SET MSCn CAPAC=m	set unit n capacity to m MB (0 = unlimited)
SHOW MSCn REEL	display current reel size or capacity
SHOW MSCn CAPAC	display current reel size or capacity

Setting the controller to optimized timing with SET MSC FASTTIME also enables two other optimizations: the initial gap normally written after BOT is omitted, and the Gap and File Mark (GFM) command is executed as a Write File Mark (WFM) command instead. Note that this does not affect the Write Gap (GAP) command.

MSC may be configured to send debugging information to the previously enabled debug output device using these commands:

SET MSC DEBUG	provide debug printouts
SET MSC NODEBUG	inhibit debug printouts

Three debug flags that control the reporting level are supported:

CMDS	Command initiations and completions
CPU	Words received from and sent to the CPU
RWS	Tape reads, writes, and status returns

The 13181A/13183A supports the `BOOT` command. `BOOT MSC` loads the IBL for 7970B/E magnetic tape drives into memory and starts it running. `BOOT -S MSC` causes the loader to position to the file number specified in the A register before starting to load data. The switch register (S) is set automatically to the value expected by the IBL loader:

```

<15:12>    =    1000
<11:6>     =    data channel device code
<5:3>      =    unchanged
<2:0>      =    00
<0>        =    1 if position tape before loading

```

The data channel (MSD) implements these registers:

name	size	comments
BUF	16	data buffer
CTL	1	interrupt enabled
FLG	1	channel ready
FBF	1	channel ready buffer
DBUF[0:65535]	8	transfer buffer
BPTR	17	buffer pointer (reads and writes)
BMAX	17	buffer size (writes)

The command channel (MSC) implements these registers:

name	size	comments
STA	12	tape status
BUF	16	buffer
USEL	2	currently selected unit
FSVC	1	first service flip-flop
CTL	1	interrupt enabled
FLG	1	controller ready
FBF	1	controller ready buffer
POS[0:3]	32	tape position
BTIME	24	BOT start delay time
CTIME	24	command delay time
GTIME	24	gap traversal time
ITIME	24	IRG traversal time
RTIME	24	rewind initiation time
XTIME	24	inter-word transfer delay time
STOP_IOE	1	stop on I/O error

Error handling is as follows:

error	processed as
not attached	tape not ready; if <code>STOP_IOE</code> , stop
end of file	parity error
OS I/O error	parity error; if <code>STOP_IOE</code> , stop

3 Symbolic Display and Input

The HP2100 simulator implements symbolic display and input. These command line switches control the display:

-a	display as ASCII character
-c	display as two packed ASCII characters
-m	display instruction mnemonics

Input parsing is controlled by the first character typed in or by command line switches:

' or -a	ASCII character
" or -c	two packed ASCII characters
alphabetic	instruction mnemonic
numeric	octal number

Instruction input uses standard HP2100 assembler syntax. There are seven instruction classes: memory reference, I/O group, shift/rotate, alter/skip, extended shift, extended memory reference, extended two-address reference.

Memory reference instructions have the format:

```
memref {C/Z} address{,I}
```

where *I* signifies indirect addressing, *C* a current-page reference, and *Z* a zero-page reference. The address is an octal number in the range 0 - 77777; if *C* or *Z* is specified, the address is a page offset in the range 0 - 1777.

C/Z is not needed when entering instructions into CPU memory; the simulator figures out from the target address what mode to use. However, when entering instructions into device address spaces (e.g., disks), the eventual target address is unknown, and *C* must be used to specify current-page addressing (omitting *C/Z* implies zero-page addressing).

I/O instructions have the format:

```
io device{,C}
```

where *C* signifies that the device flag is to be cleared. The device is an octal number in the range 0 - 77.

Shift/rotate and alter/skip instructions have the format:

```
sub-op, sub-op, sub-op...
```

The simulator checks that the combination of sub-opcodes is legal.

Extended shift instructions have the format:

```
extshift count
```

where *count* is a decimal number in the range 1 - 16.

Extended memory reference instructions have the format:

```
extmemref address{,I}
```

where I signifies indirect addressing and `address` is an octal number in the range 0 - 77777.

Extended two-address instructions have the format:

```
ext2addr addr1{,I},addr2{,I}
```

where I signifies indirect addressing. Both `addr1` and `addr2` are octal numbers in the range 0 - 77777.