PDP-10 Simulator Usage
01-Dec-2008

COPYRIGHT NOTICE

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

This memorandum documents the DEC PDP-10 simulator.

1 Simulator Files

To compile the PDP-10, you must define VM_PDP10 and USE_INT64 as part of the compilation command line.

sim/	scp.h
	sim_console.h
	sim_defs.h
	sim_ether.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_ether.c
	sim_fio.c
	sim_sock.c
	sim_tape.c
	sim_timer.c
	sim_tmxr.c
sim/pdp10/	pdp10_defs.h
	pdp10_cpu.c
	pdp10_fe.c
	pdp10_ksio.c
	pdp10_lp20.c
	pdp10_mdfp.c
	pdp10_pag.c
	pdp10_rp.c
	pdp10_sys.c
	pdp10_tu.c
	pdp10_xtnd.c
sim/pdp11/	pdp11_cr_dat.h
	pdp11_xu.h
	pdp11_cr.c
	pdp11_dz.c
	pdp11_pt.c
	pdp11_ry.c
	pdp11_xu.c

2 PDP-10 Features

The PDP-10 simulator is configured as follows:

```
device name(s)    simulates
```

CPU	KS10 CPU with 1MW of memory
PAG	paging unit (translation maps)
UBA	Unibus adapters (translation maps)
FE	console
TIM	timer
PTR,PTP	PC11 paper tape reader/punch
RY	RX211/RX02 floppy disk and two drives
DZ	DZ11 8-line terminal multiplexer (up to 4)
LP20	LP20 line printer
CR	CD20 (CD11) card reader
RP	RH11 controller with eight RP04/RP05/RP06/RP07, RM03/RM05/RM80 drives
TU	RH11/TM02 controller with eight TU45 drives
XU	DEUNA/DELUA Ethernet controller

The PTR, PTP, RX211, and CR are initially set DISABLED. The DZ11 and LP20 can also be set DISABLED. Some devices support the `SET <device> ADDRESS` command, which allows the I/O page address of the device to be changed, and the `SET <device> VECTOR` command, which allows the vector of the device to be changed. All devices support the `SHOW <device> ADDRESS` and `SHOW <device> VECTOR` commands, which display the device address and vector, respectively.

The PDP-10 simulator implements several unique stop condition:

- Illegal instruction (000) in kernel mode
- Indirect addressing nesting exceeds limit
- Execute chaining exceeds limit
- Page fail or other error in interrupt sequence
- Illegal instruction in interrupt sequence
- Invalid vector pointer in interrupt sequence
- Invalid Unibus adapter number
- Non-existent exec or user page table address

The `LOAD` command supports RIM10B format paper tapes, SAV binary files, and EXE binary files. `LOAD` switches -r, -s, -e specify RIM10, SAV, EXE format, respectively. If no switch is specified, the `LOAD` command checks the file extension; .RIM, .SAV, .EXE specify RIM10, SAV, EXE format, respectively. If no switch is specified, and no extension matches, the `LOAD` command checks the file format to try to determine the file type.

2.1 CPU

The CPU options allow the user to specify the operating system to be run. This in turn controls the microcode feature set, how system idling is detected, and how the system timer runs.

```
SET CPU TOPS10              TOPS-10
SET CPU TOPS20              TOPS-20
SET CPU ITS                 ITS
SET CPU KLAD                diagnostic environment (no idle detection)
```

The CPU implements a SHOW command to display the I/O space address map:

```
SHOW CPU IOSPACE            show I/O space address map
```

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

name	size	comments
PC	18	program counter
FLAGS	18	processor flags (<13:17> unused)
AC0..AC17	36	active register set
IR	36	instruction register
EBR	18	executive base register
PGON	1	paging enabled flag
T20P	1	TOPS-20 paging
UBR	18	user base register
CURAC	3	current AC block
PRVAC	3	previous AC block
SPT	36	shared pointer table
CST	36	core status table
PUR	36	process update register
CSTM	36	CST mask
HSB	18	halt status block address
DBR1	18	descriptor base register 1 (ITS)
DBR2	18	descriptor base register 2 (ITS)
DBR3	18	descriptor base register 3 (ITS)
DBR4	18	descriptor base register 4 (ITS)
PIENB	7	PI levels enabled
PIACT	7	PI levels active
PIPRQ	7	PI levels with program requests
PIIOQ	7	PI levels with IO requests
PIAPR	7	PI levels with APR requests
APRENB	8	APR flags enabled
APRFLG	8	APR flags active
APRLVL	3	PI level for APR interrupt
IND_MAX	8	indirect address nesting limit
XCT_MAX	8	execute chaining limit
PCQ[0:63]	18	PC prior to last jump or interrupt; most recent PC change first
WRU	8	interrupt character
REG[0:127]	36	register sets

The CPU attempts to detect when the simulator is idle. When idle, the simulator does not use any resources on the host system. Idle detection is controlled by the `SET IDLE` and `SET NOIDLE` commands:

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

Idle detection is disabled by default and is operating system dependent:

TOPS-10	SOJG 6,1 in AC1 in user mode
TOPS-20	SOJG 2,3 in AC3 in monitor mode
ITS	AOJA 0,17 in AC17 in user mode

There is no idle detection in diagnostic mode.

The CPU can maintain a history of the most recently executed instructions. This is controlled by the `SET CPU HISTORY` and `SHOW CPU HISTORY` commands:

SET CPU HISTORY	clear history buffer
SET CPU HISTORY=0	disable history

```
SET CPU HISTORY=n              enable history, length = n
SHOW CPU HISTORY               print CPU history
SHOW CPU HISTORY=n             print first n entries of CPU history
```

The maximum length for the history is 65536 entries.

2.2 Pager

The pager contains the page maps for executive and user mode. The executive page map is the memory space for unit 0, the user page map the memory space for unit 1. A page map entry is 32 bits wide and has the following format:

bit	content
31	page is writeable
30	entry is valid
29:19	mbz
18:9	physical page base address
8:0	mbz

The pager has no registers.

2.3 Unibus Adapters

The Unibus adapters link the system I/O devices to the CPU. Unibus adapter 1 (UBA1) is unit 0, and Unibus adapter 3 is unit 1. The adapter's Unibus map is the memory space of the corresponding unit.

The Unibus adapter has the following registers:

name	size	comments
INTREQ	32	interrupt requests
UB1CS	16	Unibus adapter 1 control/status
UB3CS	16	Unibus adapter 3 control/status

2.4 Front End (FE)

The front end is the system console. The keyboard input is unit 0, the console output is unit 1. It supports one option:

```
SET FE STOP                    halt the PDP-10 operating system
```

The front end has the following registers:

name	size	comments
IBUF	8	input buffer
ICOUNT	32	count of input characters
ITIME	24	input polling interval (if 0, the keyboard is polled synchronously with the clock)
OBUF	8	output buffer
OCOUNT	32	count of output characters
OTIME	24	console output response time

2.5 Timer (TIM)

The timer (TIM) implements the system timer, the interval timer, and the time of day clock used to get the date and time at system startup. Because most PDP-10 software is not Y2K compliant, the timer implements one option:

SET TIM NOY2K	software not Y2K compliant, limit time of day clock to 1999 (default)
SET TIM Y2K	software is Y2K compliant

The timer has the following registers:

name	size	comments
TIMBASE	59	time base (double precision)
TTG	36	time to go (remaining time) for interval
PERIOD	36	reset value for interval
QUANT	36	quantum timer (ITS only)
TIME	24	tick delay

Unless the CPU is set to diagnostic mode, the timer autocalibrates; the tick delay is adjusted up or down so that the time base tracks actual elapsed time. This may cause time-dependent diagnostics to report errors.

2.6 PC11 Paper Tape Reader (PTR)

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.

The paper tape reader requires an unsupported driver under TOPS-10 and is not supported under TOPS-20 or ITS.

The paper tape reader implements these registers:

name	size	comments
BUF	8	last data item processed
CSR	16	control/status register
INT	1	interrupt pending flag
ERR	1	error flag (CSR<15>)
BUSY	1	busy flag (CSR<11>)
DONE	1	device done flag (CSR<7>)
IE	1	interrupt enable flag (CSR<6>)
POS	32	position in the input 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
end of file	1	report error and stop
	0	out of tape

OS I/O error	x	report error and stop

2.7 PC11 Paper Tape Punch (PTP)

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 requires an unsupported driver under TOPS-10 and is not supported under TOPS-20 or ITS.

The paper tape punch implements these registers:

name	size	comments
BUF	8	last data item processed
CSR	16	control/status register
INT	1	interrupt pending flag
ERR	1	error flag (CSR<15>)
DONE	1	device done flag (CSR<7>)
IE	1	interrupt enable flag (CSR<6>)
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

2.8 DZ11 Terminal Multiplexer (DZ)

The DZ11 is an 8-line terminal multiplexer. Up to 4 DZ11's (32 lines) are supported. The number of lines can be changed with the command

```
SET DZ LINES=n                set line count to n
```

The line count must be a multiple of 8, with a maximum of 32.

The DZ11 supports three character processing modes, 7P, 7B, and 8B:

mode	input characters	output characters
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 is 7B, for compatibility with TOPS-20.

The DZ11 supports logging on a per-line basis. The command

```
SET DZ LOG=line=filename
```

enables logging for the specified line to the indicated file. The command

```
SET DZ NOLOG=line
```

disables logging for the specified line and closes any open log file. Finally, the command

```
SHOW DZ LOG
```

displays logging information for all DZ lines.

The terminal lines perform input and output through Telnet sessions connected to a user-specified port. The `ATTACH` command specifies the port to be used:

```
ATTACH {-am} DZ <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. The optional switch -m turns on the DZ11's modem controls; the optional switch -a turns on active disconnects (disconnect session if computer clears Data Terminal Ready). Without modem control, the DZ behaves as though terminals were directly connected; disconnecting the Telnet session does not cause any operating system-visible change in line status.

Once the DZ is attached and the simulator is running, the DZ will listen for connections on the specified port. It assumes that the incoming connections are Telnet connections. The connection remains open until disconnected by the simulated program, the Telnet client, a `SET DZ DISCONNECT` command, or a `DETACH DZ` command.

Other special DZ commands:

```
SHOW DZ CONNECTIONS            show current connections
SHOW DZ STATISTICS             show statistics for active connections
SET DZ DISCONNECT=linenumber   disconnects the specified line.
```

The DZ11 implements these registers:

name	size	comments
CSR[0:3]	16	control/status register, boards 0 to 3
RBUF[0:3]	16	receive buffer, boards 0 to 3
LPR[0:3]	16	line parameter register, boards 0 to 3
TCR[0:3]	16	transmission control register, boards 0 to 3
MSR[0:3]	16	modem status register, boards 0 to 3
TDR[0:3]	16	transmit data register, boards 0 to 3
SAENB[0:3]	1	silo alarm enabled, boards 0 to 3
RXINT	4	receive interrupts, boards 3 to 0
TXINT	4	transmit interrupts, boards 3 to 0
MDMTCL	1	modem control enabled
AUTODS	1	autodisconnect enabled

The DZ11 does not support save and restore. All open connections are lost when the simulator shuts down or the DZ is detached.

2.9 RH11 Adapter, RP04/05/06/07, RM02/03/05/80 drives (RP)

The RP controller implements the Massbus 18b (RH11) direct interface for large disk drives. RP options include the ability to set units write enabled or write locked, to set the drive type to one of six disk types, or autosize:

`SET RPn LOCKED`	set unit n write locked
`SET RPn WRITEENABLED`	set unit n write enabled
`SET RPn RM03`	set type to RM03 (same as RM02)
`SET RPn RM05`	set type to RM05
`SET RPn RM80`	set type to RM80
`SET RPn RP04`	set type to RP04 (same as RP05)
`SET RPn RP06`	set type to RP06
`SET RPn RP07`	set type to RP07
`SET RPn AUTOSIZE`	set type based on file size at attach
`SET RPn BADBLOCK`	write bad block table on last track

The type options can be used only when a unit is not attached to a file. Note that TOPS-10 V7.03 supported only the RP06 and RM03; V7.04 added support for the RP07. TOPS-20 V4.1 also supported only the RP06 and RM03. Units can be set `ENABLED` or `DISABLED`. The RP controller supports the `BOOT` command.

The RP controller implements these registers:

name	size	comments
RPCS1	16	control/status 1
RPWC	16	word count
RPBA	16	bus address
RPCS2	16	control/status 2
RPDB	16	data buffer
RPDA[0:7]	16	desired surface, sector
RPDS[0:7]	16	drive status, drives 0 to 7
RPER1[0:7]	16	drive errors, drives 0 to 7
RPHR[0:7]	16	holding register, drives 0 to 7
RPOF[0:7]	16	offset, drives 0 to 7
RPDC[0:7]	8	desired cylinder, drives 0 to 7
RPER2[0:7]	16	error status 2, drives 0 to 7
RPER3[0:7]	16	error status 3, drives 0 to 7
RPEC1[0:7]	16	ECC syndrome 1, drives 0 to 7
RPEC2[0:7]	16	ECC syndrome 2, drives 0 to 7
RPMR[0:7]	16	maintenance register, drives 0 to 7
RPMR2[0:7]	16	maintenance register 2, drives 0 to 7
IFF	1	transfer complete interrupt request flop
INT	1	interrupt pending flag
SC	1	special condition (CSR1<15>)
DONE	1	device done flag (CSR1<7>)
IE	1	interrupt enable flag (CSR1<6>)
STIME	24	seek time, per cylinder
RTIME	24	rotational delay
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	disk not ready
end of file	x	assume rest of disk is zero
OS I/O error	x	report error and stop

2.10 RH11 Adapter, TM02 Formatter, TU45 Magnetic Tape (TU)

The magnetic tape simulator simulates an RH11 Massbus adapter with one TM02 formatter and up to eight TU45 drives. Magnetic tape options include the ability to make units write enabled or locked.

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

Magnetic tape units can be set to a specific reel capacity in MB, or to unlimited capacity:

```
SET TUn CAPAC=m               set unit n capacity to m MB (0 = unlimited)
SHOW TUn CAPAC                show unit n capacity in MB
```

Units can also be set `ENABLED` or `DISABLED`. The TU controller supports the `BOOT` command.

The magnetic tape controller implements these registers:

name	size	comments
MTCS1	16	control/status 1
MTBA	16	memory address
MTWC	16	word count
MTFC	16	frame count
MTCS2	16	control/status 2
MTFS	16	formatter status
MTER	16	error status
MTCC	16	check character
MTDB	16	data buffer
MTMR	16	maintenance register
MTTC	16	tape control register
INT	1	interrupt pending flag
DONE	1	device done flag
IE	1	interrupt enable flag
STOP_IOE	1	stop on I/O error
TIME	24	delay
UST[0:7]	16	unit status, units 0 to 7
POS[0:7]	32	position, units 0 to 7

Error handling is as follows:

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

2.11 LP20 DMA Line Printer (LP20)

The LP20 is a DMA-based line printer controller. There is one line printer option to clear the vertical forms unit (VFU):

```
SET LP20 VFUCLEAR              clear the vertical forms unit
```

The LP20 implements these registers:

name	size	comments
LPCSA	16	control/status register A
LPCSB	16	control/status register B
LPBA	16	bus address register
LPBC	12	byte count register
LPPAGC	12	page count register
LPRDAT	12	RAM data register
LPCBUF	8	character buffer register
LPCOLC	8	column counter register
LPPDAT	8	printer data register
LPCSUM	8	checksum register
DVPTR	7	vertical forms unit pointer
DVLNT	7	vertical forms unit length
INT	1	interrupt request
ERR	1	error flag
DONE	1	done flag
IE	1	interrupt enable flag
POS	32	position in output file
TIME	24	response time
STOP_IOE	1	stop on I/O error
TXRAM[0:255]	12	translation RAM
DAVFU[0:142]	12	vertical forms unit array

Error handling is as follows:

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

2.12 RX211/RX02 Floppy Disk (RY)

RX211 options include the ability to set units write enabled or write locked, single or double density, or autosized:

```
SET RYn LOCKED                 set unit n write locked
SET RYn WRITEENABLED           set unit n write enabled
SET RYn SINGLE                 set unit n single density
SET RYn DOUBLE                 set unit n double density (default)
SET RYn AUTOSIZE               set unit n autosized
```

The floppy disk requires an unsupported driver under TOPS-10 and is not supported under TOPS-20 or ITS.

The RX211 implements these registers:

name	size	comments
RYCS	16	status
RYBA	16	buffer address
RYWC	8	word count
RYDB	16	data buffer
RYES	12	error status
RYERR	8	error code
RYTA	8	current track
RYSA	8	current sector
STAPTR	4	controller state
INT	1	interrupt pending flag
ERR	1	error flag (CSR<15>)
TR	1	transfer ready flag (CSR<7>)
IE	1	interrupt enable flag (CSR<6>)
DONE	1	device done flag (CSR<5>)
CTIME	24	command completion time
STIME	24	seek time, per track
XTIME	24	transfer ready delay
STOP_IOE	1	stop on I/O error
SBUF[0:255]	8	sector buffer array

Error handling is as follows:

error	STOP_IOE	processed as
not attached	1	report error and stop
	0	disk not ready

RX02 data files are buffered in memory; therefore, end of file and OS I/O errors cannot occur.

2.13 CD20 Card Reader (CR)

The card reader (CR) implements a single controller (CD20, aka CD11) and card reader (e.g., Documation M200, GDI Model 100) by reading a file and presenting lines or cards to the simulator. Card decks may be represented by plain text ASCII files, card image files, or column binary files.

Card image files are a file format designed by Douglas W. Jones at the University of Iowa to support the interchange of card deck data. These files have a much richer information carrying capacity than plain ASCII files. Card Image files can contain such interchange information as card-stock color, corner cuts, special artwork, as well as the binary punch data representing all 12 columns. Complete details on the format, as well as sample code, are available at Prof. Jones's site: http://www.cs.uiowa.edu/~jones/cards/.

The card reader supports ASCII, card image, and column binary format card “decks.” When reading plain ASCII files, lines longer than 80 characters are silently truncated. Card image support is included for 80 column Hollerith, 82 column Hollerith (silently ignoring columns 0 and 81), and 40 column Hollerith (mark-sense) cards. Column binary supports 80 column card images only. All files are attached read-only (as if the -R switch were given).

```
ATTACH –A CR <file>                 file is ASCII text
ATTACH –B CR <file>                 file is column binary
ATTACH –I CR <file>                 file is card image format
```

If no flags are given, the file extension is evaluated. If the filename ends in .TXT, the file is treated as ASCII text. If the filename ends in .CBN, the file is treated as column binary. Otherwise, the CR driver looks for a card image header. If a correct header is found the file is treated as card image format, otherwise it is treated as ASCII text.

The correct character translation MUST be set if a plain text file is to be used for card deck input. The correct translation SHOULD be set to allow correct ASCII debugging of a card image or column binary input deck. Depending upon the operating system in use, how it was generated, and how the card data will be read and used, the translation must be set correctly so that the proper character set is used by the driver. Use the following command to explicitly set the correct translation:

```
SET TRANSLATION={DEFAULT|026|026FTN|029|EBCDIC}
```

This command should be given after a deck is attached to the simulator. The mappings above are completely described at http://www.cs.uiowa.edu/~jones/cards/codes.html. Note that DEC typically used 029 or 026FTN mappings.

DEC operating systems used a variety of methods to determine the end of a deck (recognizing that 'hopper empty' does not necessarily mean the end of a deck. Below is a summary of the various operating system conventions for signaling end of deck:

```
RT-11:       12-11-0-1-6-7-8-9 punch in column 1

RSTS/E:      12-11-0-1 or 12-11-0-1-6-7-8-9 punch in column 1

RSX:         12-11-0-1-6-7-8-9 punch

VMS:         12-11-0-1-6-7-8-9 punch in first 8 columns

TOPS:        12-11-0-1 or 12-11-0-1-6-7-8-9 punch in column 1
```

Using the AUTOEOF setting, the card reader can be set to automatically generate an EOF card consisting of the 12-11-0-1-6-7-8-9 punch in columns 1-8. When set to CD11 mode, this switch also enables automatic setting of the EOF bit in the controller after the EOF card has been processed. [The CR11 does not have a similar capability.] By default AUTOEOF is enabled.

```
SET CR AUTOEOF
SET CR NOAUTOEOF
```

The default card reader rate for the CD11 is 1000 cpm. The reader rate can be set to its default value or to anywhere in the range 200..1200 cpm. This rate may be changed while the unit is attached.

```
SET CR RATE={DEFAULT|200..1200}
```

It is standard operating procedure for operators to load a card deck and press the momentary action RESET button to clear any error conditions and alert the processor that a deck is available to read. Use the following command to simulate pressing the card reader RESET button,

```
SET CR RESET
```

Another common control of physical card readers is the STOP button. An operator could use this button to finish the read operation for the current card and terminate reading a deck early. Use the following command to simulate pressing the card reader STOP button.

```
SET CR STOP
```

The simulator does not support the `BOOT` command. The simulator does not stop on file I/O errors. Instead the controller signals a reader check to the CPU.

The CR controller implements these registers:

name	size	comments
BUF	8	ASCII value of last column processed
CRS	16	CR11 status register
CRB1	16	CR11 12-bit Hollerith character
CRB2	16	CR11 8-bit compressed character
CRM	16	CR11 maintenance register
CDST	16	CD11 control/status register
CDCC	16	CD11 column count
CDBA	16	CD11 current bus address
CDDB	16	CD11 data buffer, 2nd status
BLOWER	2	blower state value
INT	1	interrupt pending flag
ERR	1	error flag (CRS<15>)
IE	1	interrupt enable flag (CRS<6>)
POS	32	file position - do not alter
TIME	24	delay time between columns

The CD11 simulation includes the Rev. J modification to make the CDDB act as a second status register during non-data transfer periods.

2.14 DEUNA/DELUA Ethernet Controller (XU)

XU simulates the DEUNA/DELUA Ethernet controller. The current implementation is experimental, and its operation is not supported under any known operating system.

3 Symbolic Display and Input

The PDP-10 simulator implements symbolic display and input. Display is controlled by command line switches:

-a	display as ASCII character
-c	display as six sixbit packed characters
-p	display as five packed ASCII (7b) characters
-m	display instruction mnemonics
-v	interpret address as virtual
-e	force executive mode
-u	force user mode

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

' or -a	ASCII character
" or -c	six sixbit packed characters
# or -p	five packed ASCII (7b) characters
alphabetic	instruction mnemonic
numeric	octal number

Instruction input uses standard PDP-10 assembler syntax. There are three instruction classes: memory reference, memory reference with AC, and I/O.

Memory reference instructions have the format

```
memref {@}address{(index)}
```

memory reference with AC instructions have the format

```
memac ac,{@}address{(index)}
```

and I/O instructions have the format

```
io device,{@}address{(index)}
```

where @ signifies indirect. The address is a signed octal number in the range 0 - 0777777. The ac and index are unsigned octal numbers in the range 0-17. The device is either a recognized device mnemonic (APR, PI, TIM) or an octal number in the range 0 - 0177.

The simulator recognizes the standard MACRO alternate mnemonics (CLEAR for SETZ, OR for IORI), the individual definitions for JRST and JFCL variants, and the extended instruction mnemonics.