PDP-1 Simulator Usage
13-Jul-2016

COPYRIGHT NOTICE

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

This memorandum documents the PDP-1 simulator.

1 Simulator Files

sim/	scp.h
	sim_console.h
	sim_defs.h
	sim_fio.h
	sim_rev.h
	sim_sock.h
	sim_timer.h
	sim_tmxr.h
	scp.c
	sim_console.c
	sim_fio.c
	sim_sock.c
	sim_timer.c
	sim_tmxr.c
sim/pdp1/	pdp1_defs.h
	pdp1_clk.c
	pdp1_cpu.c
	pdp1_dcs.c
	pdp1_drm.c
	pdp1_dt.c
	pdp1_lp.c
	pdp1_stddev.c
	pdp1_sys.c

2 PDP-1 Features

The PDP-1 is configured as follows:

```
device name(s)    simulates

CPU               PDP-1 CPU with up to 64KW of memory
                  optional automatic multiply/divide
                  optional 16-channel sequence break system
                  optional PDP-1D extended features
CLK               1Khz time-sharing clock (PDP-1D)
PTR,PTP           integral paper tape reader/punch
TTI,TTO           console typewriter
LPT               Type 62 line printer
DRM               Type 24 serial drum
DRP               Type 23 parallel drum
DT                Type 550 Microtape (DECtape)
DCS,DCSL          Type 630 Data Communications Subsystem
```

The PDP-1 simulator implements the following unique stop conditions:

- An unimplemented instruction is decoded, and register STOP_INST is set

- More than IND_MAX indirect addresses are detected during memory reference address decoding
- More than XCT_MAX nested executes are detected during instruction execution
- I/O wait, and no I/O operations outstanding (i.e, no I/O completion will ever occur)
- A simulated DECtape runs off the end of its reel

The `LOAD` command supports RIM format tapes and BLK format tapes. If the file to be loaded has an extension of .BIN, or switch -B is specified, the file is assumed to be BLK format; otherwise, it defaults to RIM format. `LOAD` takes an optional argument that specifies the starting address of the field to be loaded:

```
LOAD lisp.rim                load RIM format file lisp.rim
LOAD ddt.rim 70000           load RIM format file ddt.rim into
                             the field starting at 70000
LOAD -B macro.blk            load BLK format file macro.blk
```

The `DUMP` command is not implemented.

2.1 CPU

The only CPU options are the presence of hardware multiply/divide and the size of main memory.

```
SET CPU MDV                  enable multiply/divide
SET CPU NOMDV                disable multiply/divide
SET CPU SBS16                enable 16-channel sequence break system
SET CPU NOSBS16              disable 16-channel sequence break system
SET CPU PDP1C                set CPU to standard PDP-1C
SET CPU PDP1DS45             set CPU to PDP-1D, serial# 45 (BBN)
SET CPU PDP1DS48             set CPU to PDP-1D, serial# 48 (Stanford)
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 20K                  set memory size = 20K
SET CPU 24K                  set memory size = 24K
SET CPU 28K                  set memory size = 28K
SET CPU 32K                  set memory size = 32K
SET CPU 48K                  set memory size = 48K
SET CPU 64K                  set memory size = 64K
```

If memory size is being reduced, and the memory being truncated contains non-zero data, the simulator asks for confirmation. Data in the truncated portion of memory is lost. Initial memory size is 64K. Setting the CPU to PDP-1D also enables multiply/divide and the 16-channel sequence break system.

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

```
name             size        comments

PC               16          program counter
AC               18          accumulator
IO               18          IO register
OV               1           overflow flag
PF               6           program flags<1:6>
SS               6           sense switches<1:6>
TA               16          address switches
```

```
TW              18          test word (front panel switches)
EXTM            1           extend mode
RNGM            1           ring mode (PDP-1D only)
L               1           link (PDP-1D #45 only)
RM              1           restrict mode (PDP-1D)
RMASK           1           restrict memory mask (PDP-1D)
RTB             18          restrict trap buffer (PDP-1D #45 only)
RNAME[0:3]      2           rename map (PDP-1D #45 only)
IOSTA           18          IO status register
SBON            1           sequence break enable
SBRQ            1           sequence break request
SBIP            1           sequence break in progress
SBSREQ          16          pending sequence break requests
SBSENB          16          enabled sequence break levels
SBSACT          16          active sequence break levels
IOH             1           I/O halt in progress
IOS             1           I/O synchronizer (completion)
PCQ[0:63]       16          PC prior to last jump or interrupt;
                            most recent PC change first
STOP_INST       1           stop on undefined instruction
SBS_INIT        1           initial state of sequence break enable
EXTM_INIT       1           initial state of extend mode
XCT_MAX         8           maximum XCT chain
IND_MAX         8           maximum nested indirect addresses
WRU             8           interrupt character
```

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.

If the 16-channel sequence break system is enabled, devices can be assigned to any break level between 0 (the default) and 15, with the following command:

```
SET <dev> SBSLVL=n          assign device to sequence break level n
```

Because each PDP-1 configuration was unique, there are no default assignments for the 16-channel sequence break system.

2.2 Programmed I/O Devices

2.2.1 Paper Tape Reader (PTR)

The paper tape reader (PTR) reads data from or 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 supports the `BOOT` command. `BOOT PTR` copies the RIM loader into memory and starts it running. `BOOT PTR` loads into the field selected by TA<0:3> (the high order four bits of the address switches).

The paper tape reader recognizes one switch at ATTACH time:

```
ATT -A PTR <file>             convert input characters from ASCII
```

By default, the paper tape reader does no conversions on input characters.

The paper tape reader implements these registers:

```
name              size        comments

BUF               8           last data item processed
DONE              1           device done flag
RPLS              1           return restart pulse flag
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
```

2.2.2 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 recognizes one switch at ATTACH time:

```
ATT -A PTP <file>             output characters as ASCII text
```

By default, the paper tape punch punches files with no conversions.

The paper tape punch implements these registers:

```
name              size        comments

BUF               8           last data item processed
DONE              1           device done flag
RPLS              1           return restart pulse flag
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
```

2.2.3 Console Typewriter Input (TTI), Output (TTO)

The Typewriter is a half-duplex electric typewriter (originally a Friden Flexowriter, later a Sorobon-modified IBM B). It has only a single buffer and a single carriage state but distinct input and output done and interrupt flags. The typewriter input (TTI) polls the console keyboard for input. The typewriter output (TTO) writes to the simulator console window.

The Typewriter recognizes one option:

```
SET TTO ET                  Expensive Typewriter mode
SET TTO NOET                normal mode
```

In Expensive Typewriter mode, ribbon changes are output as strings ([red], [black]) to indicate which mode the program is in (red for command, black for text). In normal mode, ribbon changes are ignored.

The typewriter input implements these registers:

```
name              size        comments

BUF               6           typewriter buffer (shared)
UC                1           upper case/lower case state (shared)
DONE              1           input ready flag
POS               32          number of characters input
TIME              24          keyboard polling interval
```

The typewriter output implements these registers:

```
name              size        comments

BUF               6           typewriter buffer (shared)
UC                1           upper case/lower case state (shared)
RPLS              1           return restart pulse flag
DONE              1           output done flag
POS               32          number of characters output
TIME              24          time from I/O initiation to interrupt
```

2.2.4 Type 62 Line Printer (LPT)

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 can be disabled and enabled with the `SET LPT DISABLED` and `SET LPT ENABLED` commands, respectively.

The line printer implements these registers:

```
name              size        comments

BUF               8           last data item processed
```

```
PNT             1          printing done flag
SPC             1          spacing done flag
RPLS            1          return restart pulse flag
BPTR            6          print buffer pointer
POS             32         position in the output file
TIME            24         time from I/O initiation to interrupt
STOP_IOE        1          stop on I/O error
LBUF[0:119]     8          line buffer
```

Error handling is as follows:

```
error         STOP_IOE      processed as

not attached      1         report error and stop
                  0         out of tape or paper

OS I/O error      x         report error and stop
```

2.2.5 Type 550/555 Microtape (DECtape) (DT)

The PDP-1 uses the Type 550 Microtape (later renamed DECtape), a programmed I/O controller. PDP-1 DECtape format has 4 18b words in its block headers and trailers.

DECtapes drives are numbered 1-8; in the simulator, drive 8 is unit 0. DECtape options include the ability to make units write enabled or write locked.

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

Units can also be set `ENABLED` or `DISABLED`.

The DECtape controller can be disabled and enabled with the `SET DT DISABLED` and `SET DT ENABLED` commands, respectively.

The Type 550 supports PDP-8 format, PDP-11 format, and 18b format DECtape images. `ATTACH` assumes the image is in 18b format; the user can other choices with switches:

```
-t                            PDP-8 format
-s                            PDP-11 format
-a                            autoselect based on file size
```

The DECtape controller is a data-only simulator; the timing and mark track, and block header and trailer, are not stored. Thus, the WRITE TIMING AND MARK TRACK function is not supported; the READ ALL function always returns the hardware standard block header and trailer; and the WRITE ALL function dumps non-data words into the bit bucket.

The DECtape controller implements these registers:

```
name            size       comments

DTSA            12         status register A
DTSB            12         status register B
DTDB            18         data buffer
DTF             1          DECtape flag
BEF             1          block end flag
```

```
ERF             1           error flag
LTIME           31          time between lines
DCTIME          31          time to decelerate to a full stop
SUBSTATE        2           read/write command substate
POS[0:7]        32          position, in lines, units 0-7
STATT[0:7]      18          unit state, units 0-7
STOP_OFFR       1           stop on off-reel error
```

It is critically important to maintain certain timing relationships among the DECtape parameters, or the DECtape simulator will fail to operate correctly.

- LTIME must be at least 6
- DCTIME needs to be at least 100 times LTIME

Acceleration time is set to 75% of deceleration time.

2.2.6 PDP-1D Timesharing Clock (CLK)

The PDP-1D implements a timesharing clock, which operates at 1Khz. The clock has a readable counter and generates interrupts at 32 ms and 1 minute intervals. There is no other visible state. The clock is disabled by default.

The clock implements these registers:

```
name            size        comments

CNTR            16          clock counter, range 0-59999(10)
```

The clock requires the 16-channel sequence break system and is assigned to two different SBS levels:

```
SET CLK SBS32MS=n           assign 32 msec interrupt to SBS level n
SET CLK SBS1MIN=n           assign 1 minute interrupt to SBS level n
```

2.2.7 Type 630 Data Communications Subsystem (DCS, DCSL)

The Type 630 Data Communications Subsystem provides up to 32 asynchronous interfaces. The Type 630 consists of two independent devices: DCS for the scanner, and DCSL for the individual lines. The terminal multiplexer performs input and output through Telnet sessions connected to a user-specified port. The `ATTACH` command specifies the port to be used:

```
ATTACH DCS <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 number of lines can be changed with `SET DCL LINES` command:

```
SET DCS LINES=n         set number of lines to n, where n is 1-32
```

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

```
mode  input characters        output characters

UC    lower case converted    lower case converted to upper case,
       to upper case,         high-order bit cleared,
      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. Finally, each line supports output logging. `The SET DCSLn LOG` command enables logging on a line:

```
SET DCSLn LOG=filename        log output of line n to filename
```

The `SET DCSLn NOLOG` command disables logging and closes the open log file, if any.

Once DCS 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 DCS DISCONNECT` command, or a `DETACH DCS` command.

Other special commands:

```
SHOW DCS CONNECTIONS          show current connections
SHOW DCS STATISTICS           show statistics for active connections
SET DCSLn DISCONNECT          disconnects the specified line.
```

The multiplexer scanner (DCS) implements these registers:

```
name              size        comments

BUF[0:31]         8           input buffer, lines 0 to 31
FLG[0:31]         1           line ready flag, lines 0 to 31
SCNF              1           scanner ready flag
SCAN              5           scanner line number
SEND              5           output line number
```

The individual lines (DCSL) implement these registers:

```
name              size        comments

TIME[0:31]        24          time from I/O initiation to interrupt,
                              lines 0 to 31
```

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

2.3 Drums

The PDP-1 supports two drums: the Type 23 parallel drum (DRP) and the Type 24 serial drum (DRM). Both use device addresses 061-064; accordingly, only one can be enabled at a time. By default, the Type 24 serial drum is enabled, and the Type 23 parallel drum is disabled. The PDP-1D requires the Type 23 parallel drum.

2.3.1 Type 24 Serial Drum (DRM)

The serial drum (DRM) implements these registers:

```
name              size        comments

DA                9           drum address (sector number)
```

```
MA              16          current memory address
DONE            1           device done flag
ERR             1           error flag
WLK             32          write lock switches
TIME            24          rotational latency, per word
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         drum not ready
```

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

2.3.2 Type 23 Parallel Drum (DRP)

The parallel drum (DRP) implements these registers:

```
name            size        comments

TA              12          track address
RDF             5           read field
RDE             1           read enable flag
WRF             5           write field
WRF             1           write enable flag
MA              16          current memory address
WC              12          word count
BUSY            1           device busy flag
ERR             1           error flag
TIME            24          rotational latency, per word
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         drum not ready
```

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

3 Symbolic Display and Input

The PDP-1 simulator implements symbolic display and input. Display is controlled by command line switches:

```
-a              display as ASCII character
-c              display as three packed FIODEC 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          three packed FIODEC characters
alphabetic       instruction mnemonic
numeric          octal number
```

Instruction input uses modified PDP-1 assembler syntax. There are six instruction classes: memory reference, shift, skip, operate, IOT, and LAW.

Memory reference instructions have the format

```
memref {I} address
```

where I signifies indirect reference. The address is an octal number in the range 0 - 0177777.

Shift instructions have the format

```
shift shift_count
```

The shift count is an octal number in the range 0-9.

Skip instructions consist of single mnemonics, eg, SZA, SZS4. Skip instructions may be or'd together

```
skip skip skip...
```

The sense of a skip can be inverted by including the mnemonic I.

Operate instructions consist of single mnemonics, eg, CLA, CLI. Operate instructions may be or'd together

```
opr opr opr...
```

IOT instructions consist of single mnemonics, eg, TYI, TYO. IOT instructions may include an octal numeric modifier or the modifier I:

```
iot modifier
```

The simulator does not check the legality of skip, operate, or IOT combinations.

Finally, the LAW instruction has the format

```
LAW {I} immediate
```

where immediate is in the range 0 to 07777.

4 Character Sets

The PDP-1's first console was a Frieden Flexowriter; its character encoding was known as FIODEC. The PDP-1's line printer used a modified Hollerith character set. The following table provides equivalences between ASCII characters and the PDP-1's I/O devices. In the console table, UC stands for upper case. The console table also applies to ASCII mode for the paper tape reader and punch.

ASCII	PDP-1 console	PDP-1 line printer
000 - 007	none	none

bs	075	none
tab	036	none
012 - 013	none	none
ff	013	none
cr	077	none
016 - 037	none	none
space	000	000
!	{OR} UC+005	none
"	UC+001	none
#	{IMPLIES} UC+004	none
$	none	none
%	none	none
&	{AND} UC+006	none
'	UC+002	none
(	057	057
)	055	055
*	{TIMES} UC+073	072
+	UC+054	074
,	033	033
-	054	054
.	073	073
/	021	021
0	020	020
1	001	001
2	002	002
3	003	003
4	004	004
5	005	005
6	006	006
7	007	007
8	010	010
9	011	011
:	none	none
;	none	none
<	UC+007	034
=	UC+033	053
>	UC+010	034
?	UC+021	037
@	{MID DOT} 040	{MID DOT} 040
A	UC+061	061
B	UC+062	062
C	UC+063	063
D	UC+064	064
E	UC+065	065
F	UC+066	066
G	UC+067	067
H	UC+070	070
I	UC+071	071
J	UC+041	041
K	UC+042	042
L	UC+043	043
M	UC+044	044
N	UC+045	045
O	UC+046	046
P	UC+047	047
Q	UC+050	050
R	UC+051	051

S	UC+022	022
T	UC+023	023
U	UC+024	024
V	UC+025	025
W	UC+026	026
X	UC+027	027
Y	UC+030	030
Z	UC+031	031
[	UC+057	none
\	{OVERLINE} 056	{OVERLINE} 056
]	UC+055	none
^	{UP ARROW} UC+011	{UP ARROW} 035
_	UC+040	UC+040
`	{RT ARROW} UC+020	036
a	061	none
b	062	none
c	063	none
d	064	none
e	065	none
f	066	none
g	067	none
h	070	none
i	071	none
j	041	none
k	042	none
l	043	none
m	044	none
n	045	none
o	046	none
p	047	none
q	050	none
r	051	none
s	022	none
t	023	none
u	024	none
v	025	none
w	026	none
x	027	none
y	030	none
z	031	none
{	none	none
|	UC+056	076
}	none	none
~	UC+003	013
del	075	none