SDS 940 Simulator Usage
30-Jan-2007

COPYRIGHT NOTICE

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

This memorandum documents the SDS 940 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/sds/	sds_defs.h
	sds_cpu.c
	sds_drm.c
	sds_dsk.c
	sds_io.c
	sds_lp.c
	sds_mt.c
	sds_mux.c
	sds_rad.c
	sds_stddev.c
	sds_sys.c

2 SDS 940 Features

The SDS-940 simulator is configured as follows:

```
device name(s)    simulates

CPU               SDS-940 CPU with 16KW to 64KW of memory
CHAN              I/O channels
PTR               paper tape reader
PTP               paper tape punch
TTI               console input
TTO               console output
LPT               line printer
RTC               real-time clock
MUX               terminal multiplexor
DRM               Project Genie drum
RAD               fixed head disk
DSK               9164/9165 rapid access (moving head) disk
MT                magnetic tape
```

Most devices can be disabled or enabled with the `SET <dev> DISABLED` and `SET <dev> ENABLED` commands, respectively.

The `LOAD` command is used to load a line printer carriage-control tape. The `DUMP` command is not implemented.

2.1 CPU

The CPU options set the size of main memory and the configuration of peripherals.

```
SET CPU 16K                 set memory size = 16KW
SET CPU 32K                 set memory size = 32KW
SET CPU 48K                 set memory size = 48KW
SET CPU 64K                 set memory size = 64KW
SET CPU GENIE               enable DRM, set terminal mux to GENIE mode
SET CPU SDS                 disable DRM, set terminal mux to SDS mode
```

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 64KW.

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

```
name            size        comments

P               14          program counter
A               24          accumulator A
B               24          accumulator B
X               24          index register
OV              1           overflow indicator
EM2             3           memory extension, quadrant 2
EM3             3           memory extension, quadrant 3
RL1             24          user relocation register 1
RL2             24          user relocation register 2
RL4             12          kernel relocation register
NML             1           normal mode flag
USR             1           user mode flag
MONUSR          1           monitor-to-user trap enable
ION             1           interrupt enable
INTDEF          1           interrupt defer
INTREQ          32          interrupt request flags
APIACT          5           highest active API level
APIREQ          5           highest requesting API level
XFRREQ          32          device transfer request flags
BPT             4           breakpoint switches
ALERT           6           outstanding alert number
STOP_INVINS     1           stop on invalid instruction
STOP_INVDEV     1           stop on invalid device number
STOP_INVIOP     1           stop on invalid I/O operation
INDLIM          8           maximum indirect nesting depth
EXULIM          8           maximum execute nesting depth
PCQ[0:63]       14          P prior to last branch or interrupt;
                            most recent P change first
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.

2.2 Channels (CHAN)

The SDS 940 has up to eight I/O channels, designated W, Y, C, D, E, F, G, and H. W, Y, C, and D are time-multiplexed communications channels (TMCC); E, F, G, and H are direct access communications channels (DACC). Unlike real SDS 940 channels, the simulated channels handle 6b, 12b, and 24b transfers simultaneously. The association between a device and a channel is displayed by the `SHOW <dev> CHAN` command:

```
SHOW LPT CHAN
channel=W
```

The user can change the association with the `SET <dev> CHAN=<chan>` command, where `<chan>` is a channel letter:

```
SET LPT CHAN=E
SHOW LPT CHAN
channel=E
```

Each channel has nine registers. The registers are arrays, with entry [0] for channel W, entry [1] for channel Y, etc.

name	size	comments
UAR[0:7]	6	unit address register
WCR[0:7]	15	word count register
MAR[0:7]	16	memory address register
DCR[0:7]	6	data chaining register
WAR[0:7]	24	word assembly register
CPW[0:7]	2	characters per word
CNT[0:7]	3	character count
MODE[0:7]	12	channel mode (from EOM instruction)
FLAG[0:7]	9	channel flags

The user can display all the registers in a channel with the command:

```
SHOW CHAN channel-letter
```

2.3 Console Input (TTI)

The console input (TTI) polls the console keyboard for input. It implements these registers:

name	size	comments
BUF	6	data buffer

```
XFR             1          transfer ready flag
POS             32         number of characters input
TIME            24         polling interval
```

By default, the console input is assigned to channel W.

2.4 Console Output (TTO)

The console output (TTO) writes to the simulator console window. It implements these registers:

```
name            size       comments

BUF             6          data buffer
XFR             1          transfer ready flag
POS             32         number of characters input
TIME            24         time from I/O initiation to interrupt
```

By default, the console output is assigned to channel W.

2.5 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 implements these registers:

```
name            size       comments

BUF             6          data buffer
XFR             1          transfer ready flag
SOR             1          start of record flag
CHAN            4          active channel
POS             32         number of characters input
TIME            24         time from I/O initiation to interrupt
STOP_IOE        1          stop on I/O error
```

The paper-tape reader supports the `BOOT` command. `BOOT PTR` simulates the standard console fill sequence.

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
```

By default, the paper tape reader is assigned to channel W.

2.6 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 implements these registers:

name	size	comments
BUF	6	data buffer
XFR	1	transfer ready flag
LDR	1	punch leader flag
CHAN	4	active channel
POS	32	number of characters input
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

By default, the paper tape punch is assigned to channel W.

2.7 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 implements these registers:

name	size	comments
BUF[0:131]	8	data buffer
BPTR	8	buffer pointer
XFR	1	transfer ready flag
ERR	1	error flag
CHAN	4	active channel
CCT[0:131]	8	carriage control tape
CCTP	8	pointer into carriage control tape
CCTL	8	length of carriage control tape
SPCINST	24	spacing instruction
POS	32	number of characters input
CTIME	24	intercharacter time
PTIME	24	print time
STIME	24	space time
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

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

By default, the line printer is assigned to channel W.

2.8 Real-Time Clock (RTC)

The real-time clock (RTC) frequency can be adjusted as follows:

```
SET RTC 60HZ                  set frequency to 60Hz
SET RTC 50HZ                  set frequency to 50Hz
```

The default is 60Hz.

The clock implements these registers:

```
name              size        comments

PIE               1           interrupt enable
TIME              24          tick interval
```

The real-time clock autocalibrates; the clock interval is adjusted up or down so that the clock tracks actual elapsed time.

2.9 Terminal Multiplexer (MUX)

The terminal multiplexer provides 32 asynchronous interfaces. In Genie mode, the interfaces are hard-wired; in SDS mode, they implement modem control. The multiplexer has two controllers: MUX for the scanner, and MUXL 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 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) supports one option: UC, when set, causes lower case input characters to be automatically converted to upper case. In addition, each line supports output logging. The `SET MUXLn LOG` command enables logging on a line:

```
SET MUXLn filename            log output of line n to filename
```

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

Once MUX is attached and the simulator is running, the multiplexor 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 MUX DISCONNECT command, or a DETACH MUX command.

Other special multiplexer commands:

```
SHOW MUX CONNECTIONS          show current connections
SHOW MUX STATISTICS           show statistics for active connections
```

```
SET MUXLn DISCONNECT          disconnects the specified line.
```

The controller (MUX) implements these registers:

```
name              size        comments

STA[0:31]         6           status, lines 0 to 31
RBUF[0:31]        8           receive buffer, lines 0 to 31
XBUF[0:31]        8           transmit buffer, lines 0 to 31
FLAGS[0:127]      1           line flags, 0 to 3 for line 0,
                              4 to 7 for line 1, etc
SCAN              7           scanner current flag number
SLCK              1           scanner locked flag
TPS               8           character polls per second
```

The lines (MUXL) implements these registers:

```
name              size        comments

TIME[0:31]        24          transmit time, lines 0 to 31
```

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

2.10 Project Genie Drum (DRM)

The Project Genie drum (DRM) implements these registers:

```
name              size        comments

DA                19          drum address
CA                16          core address
WC                14          word count
PAR               12          cumulative sector parity
RW                1           read/write flag
ERR               1           error flag
STA               2           drum state
FTIME             24          channel program fetch time
XTIME             24          interword transfer time
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. Unlike conventional SDS 940 devices, the Project Genie drum does not use a channel.

2.11 Rapid Access (fixed head) Disk (RAD)

The rapid access disk (RAD) implements these registers:

```
name              size        comments

DA                15          disk address
SA                6           sector word address
BP                1           sector byte pointer
XFR               1           data transfer flag
NOBD              1           inhibit increment across track
ERR               1           error flag
CHAN              4           active channel
PROT              8           write protect switches
TIME              24          interval between halfword transfers
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
```

The rapid access disk is buffered in memory; end of file and OS I/O errors cannot occur. By default, the rapid access disk is assigned to channel E.

2.12 Moving Head Disk (DSK)

DSK options include the ability to make the drive write enabled or write locked:

```
SET RAD LOCKED                set write locked
SET RAD WRITEENABLED          set write enabled
```

The moving head disk implements these registers:

```
name              size        comments

BUF[0:63]         8           transfer buffer
BPTR              9           buffer pointer
BLNT              9           buffer length
DA                21          disk address
INST              24          disk instruction
XFR               1           data transfer flag
ERR               1           error flag
CHAN              4           active channel
WTIME             24          interval between character transfers
STIME             24          seek interval
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
```

By default, the moving head disk is assigned to channel F.

2.13 Magnetic Tape (MT)

MT options include the ability to make units write enabled or write locked.

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

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

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

Units can also be set `ENABLED` or `DISABLED`. The magnetic tape controller supports the `BOOT` command. `BOOT MTn` simulates the standard console fill sequence for unit n.

The magnetic tape implements these registers:

```
name              size        comments

BUF[0:131071]     8           transfer buffer
BPTR              18          buffer pointer
BLNT              18          buffer length
XFR               1           data transfer flag
CHAN              4           active channel
INST              24          magtape instruction
EOF               1           end-of-file flag
GAP               1           inter-record gap flag
SKIP              1           skip data flag
CTIME             24          interval between character transfers
GTIME             24          gap interval
POS[0:7]          32          position, drives 0:7
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       end of tape

OS I/O error      end of tape; if STOP_IOE, stop
```

By default, the magnetic tape is assigned to channel W.

3 Symbolic Display and Input

The SDS 940 simulator implements symbolic display and input. Display is controlled by command line switches:

```
-a              display as ASCII character
-c              display as four packed SDS 6b 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         four packed SDS 6b characters
alphabetic      instruction mnemonic
numeric         octal number
```

Instruction input uses (more or less) standard SDS 940 assembler syntax. There are eight instruction classes:

```
class           operands        examples        comments

no operand      none            EIR
POP (prog op)   op,addr{,tag}   POP 66,100
I/O             addr{,tag}      EOM 1266
mem reference   addr{,tag}      LDA 400,2
                                STA* 300        indirect addr
reg change      op op op...     CLA CLB         opcodes OR
shift           cnt{,tag}       LSH 10
chan command    chan            ALC W
chan test       chan            CAT Y
```

All numbers are octal. Channel designators can be alphabetic (W, Y, C, D, E, F, G, H) or numeric (0-7). Tags must be 0-7, with 2 indicating indexing.