

LGP Simulator Usage

30-Jan-2007

COPYRIGHT NOTICE

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

Original code published in 1993-2007, written by Robert M Supnik
Copyright (c) 1993-2007, 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	LGP Features	3
2.1	CPU	3
2.2	Typewriter Input (TTI)	5
2.3	Typewriter Output (TTO).....	6
2.4	High Speed Paper-Tape Reader (PTR).....	6
2.5	High Speed Paper-Tape Punch (PTP).....	7
3	Symbolic Display and Input.....	8
4	Character Set.....	9

This memorandum documents the LGP-30 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/lgp/      lgp_defs.h
              lgp_cpu.c
              lgp_stddev.c
              lgp_sys.c
```

2 LGP Features

The LGP is configured as follows:

device names	simulates
CPU	LGP-30 or LGP-21 CPU with 4096 words of memory
TTI	Typewriter input (keyboard and reader)
TTO	Typewriter output (printer and punch)
PTR	high-speed paper tape reader
PTP	high-speed paper tape punch

The LGP simulator implements the following unique stop conditions:

- LGP-30 only: arithmetic overflow
- LGP-21 only: reference to undefined I/O device.

The `LOAD` and `DUMP` commands are not implemented.

2.1 CPU

The CPU implements either the LGP-30 or the LGP-21:

```
SET CPU LGP30          set LGP-30
SET CPU LGP21          set LGP-21
```

The default is the LGP-30. Memory size is fixed at 4096 words.

The following commands implement various front panel functions:

D A value	equivalent to the MANUAL INPUT button
SET CPU FILL{=value}	equivalent to the FILL INSTRUCTION button; if no value is given, fills IR from A; else, fills IR from the specified value
SET CPU EXEC{=value}	equivalent to the EXECUTE button; if no value is given, executes the instruction in IR; else, executes the instruction specified by the value
SET CPU MANUAL	equivalent to setting the MANUAL INPUT switch on the Typewriter; Typewriter input is taken from the keyboard
SET CPU TAPE	equivalent to clearing the MANUAL INPUT switch on the Typewriter; Typewriter input is taken from the Typewriter paper tape reader

The following commands control the display of information:

SET CPU LGPHEX	numeric displays use LGP hexadecimal encoding
SET CPU STANDARD	numeric displays use standard hexadecimal
SET CPU TRACK	symbolic addresses are ttss, where tt = track (0-63) and ss = sector (0-63)
SET CPU NORMAL	symbolic addresses are normal linear addresses, from 0 to 4095.

The defaults are STANDARD hex and TRACK addresses.

The LGP-30 implements the following additional commands:

SET CPU 4B	sets the CPU to 4-bit input mode
SET CPU 6B	sets the CPU to 6-bit input mode
SET CPU INPUT=TTI	sets the CPU to read from the Typewriter
SET CPU INPUT=PTR	sets the CPU to read from the high-speed reader
SET CPU OUTPUT=TTO	sets the CPU to output to the Typewriter
SET CPU OUTPUT=PTP	sets the CPU to output to the high-speed punch

The defaults are TAPE input, 4B input mode, input and output assigned to the Typewriter.

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

name	size	comments
PC	12	counter
A	32	accumulator
IR	32	instruction register
OVF	1	overflow flag (LGP-21 only)
TSW	1	transfer switch
BP32	1	breakpoint 32 switch
BP16	1	breakpoint 16 switch

BP8	1	breakpoint 8 switch
BP4	1	breakpoint 4 switch
INPST	1	input pending flag
INPDN	1	input done flag
OUTST	1	output pending flag
OUTDN	1	output done flag
WRU	8	interrupt character

2.2 Typewriter Input (TTI)

The Typewriter input consists of two units: the keyboard (unit 0) and the paper-tape reader (unit 1). The keyboard is permanently associated with the console window. The paper-tape reader can be attached to a disk file. The RPOS register specifies the number of the next data item to be read. Thus, by changing RPOS, the user can backspace or advance the reader.

The Typewriter input has the following options:

SET TTI1 ASCII	default tape file format is ASCII-encoded Flex
SET TTI1 FLEX	default tape file format is transposed Flex
SET TTI1 CSTOP	reader recognizes conditional stop
SET TTI1 NOCSTOP	reader ignores conditional stop
SET TTI RSTART	start the reader; equivalent to the START READER lever
SET TTI RSTOP	stop the reader; equivalent to the STOP READER lever
SET TTI START	send START signal to the CPU; equivalent to the START COMPUTE lever

Transposed Flex has the tape channels in this order: 6-1-2-3-4-5.

The ATTACH command recognizes two switches:

ATT -A TTI1 <file>	file format is ASCII-encoded Flex
ATT -F TTI1 <file>	file format is transposed Flex

The Typewriter input implements these registers:

name	size	comments
BUF	6	data buffer
RDY	1	data ready flag
KPOS	32	count of keyboard characters
RPOS	32	position in the reader input file
TIME	24	time between keyboard polls or reader characters
STOP_IOE	1	stop on I/O error

Error handling for the Typewriter paper-tape reader 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

2.3 Typewriter Output (TTO)

The Typewriter output consists of two units: the printer (unit 0) and the paper-tape punch (unit 1). The printer is permanently associated with the console window. The paper-tape punch can be attached to a disk file. The PPOS register specifies the number of the next data item to be written. Thus, by changing PPOS, the user can backspace or advance the punch.

The Typewriter output has the following options:

SET TTO1 ASCII	default tape file format is ASCII-encoded Flex
SET TTO1 FLEX	default tape file format is transposed Flex
SET TTO1 FEED=n	punch 'n' feed (0) characters

Transposed Flex has the tape channels in this order: 6-1-2-3-4-5. The default is ASCII-encoded Flex.

The ATTACH command recognizes two switches:

ATT -A TTO1 <file>	file format is ASCII-encoded Flex
ATT -F TTO1 <file>	file format is transposed Flex

The Typewriter output implements these registers:

name	size	comments
BUF	6	data buffer
UC	1	upper case flag
TPOS	32	count of output characters
PPOS	32	position in the punch output file
TIME	24	time from I/O initiation to completion
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

2.4 High Speed 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 has the following options:

SET PTR ASCII	default tape file format is ASCII-encoded Flex
SET PTR FLEX	default tape file format is transposed Flex

Transposed Flex has the tape channels in this order: 6-1-2-3-4-5. The default is ASCII-encoded Flex.

The ATTACH command recognizes two switches:

```
ATT -A PTR <file>          file format is ASCII-encoded Flex
ATT -F PTR <file>          file format is transposed Flex
```

The paper tape reader implements these registers:

name	size	comments
BUF	6	last data item processed
RDY	1	data ready flag
POS	32	position in the input file
TIME	24	time from I/O initiation to completion
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

2.5 High Speed 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 has the following options:

```
SET PTP ASCII              default tape file format is ASCII-encoded
                             Flex
SET PTP FLEX                default tape file format is transposed Flex
SET PTP FEED=n             punch 'n' feed (0) characters
```

Transposed Flex has the tape channels in this order: 6-1-2-3-4-5. The default is ASCII-encoded Flex.

The ATTACH command recognizes two switches:

```
ATT -A PTP <file>          file format is ASCII-encoded Flex
ATT -F PTP <file>          file format is transposed Flex
```

The paper tape punch implements these registers:

name	size	comments
BUF	6	last data item processed
POS	32	position in the output file
TIME	24	time from I/O initiation to completion

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

3 Symbolic Display and Input

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

-a	display as character (tape files only)
-h	display as standard hexadecimal
-l	display as LGP hexadecimal
-m	display instruction mnemonics
-n	display addresses in normal format (overrides SET CPU TRACK)
-t	display addresses as track/sector (overrides SET CPU NORMAL)

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

' or -a	Flex character
- or opcode	instruction mnemonic
numeric	hexadecimal number

LGP hexadecimal differs from standard hexadecimal in the characters used for digits 10-15

digit	standard hex	LPG hex
10	A	F
11	B	G
12	C	J
13	D	K
14	E	Q
15	F	W

There is only instruction format:

```
{-}op address
```

'op' is always a single letter. A track/sector address (specified by SET CPU TRACK or switch -t) is two decimal numbers between 0 and 63, representing the track and sector. A linear address (specified by SET CPU NORMAL or switch -n) is one decimal number between 0 and 4095. For example:

```
sim> d -n 64 10640
sim> ex -mn 64
64: B 400
sim> ex -mt 100
0100: B 0616
```

4 Character Set

The LGP Typewriter was a Friden Flexowriter. Input was always upper case; output could be either upper case or lower case. The following table provides equivalences between LPG Typewriter coding and ASCII.

Typewriter code (hex)	Input	LC output	UC output
00	illegal	illegal	illegal
01	z or Z	z	Z
02	0 or)	0)
03	space	space	space
04	illegal	lower case	lower case
05	b or B	b	B
06	l or L	l	L
07	- or _	0	_
10	illegal	upper case	upper case
11	y or Y	y	Y
12	2 or *	2	*
13	+ or =	+	=
14	illegal	color shift	color shift
15	r or R	r	R
16	3 or "	3	"
17	; or :	;	:
20	newline	newline	newline
21	i or I	i	I
22	4 or ^	4	^
23	/ or ?	/	?
24	illegal	backspace	backspace
25	d or D	d	D
26	5 or %	5	%
27	. or]	.]
30	tab	tab	tab
31	n or N	n	N
32	6 or \$	6	\$
33	, or [,	[
34	illegal	illegal	illegal
35	m or M	m	M
36	7 or ~	7	~
37	v or V	v	V
40	' (cond stop)	'	'
41	p or P	p	P
42	8 or #	8	#
43	o or O	o	O
44	illegal	illegal	illegal
45	e or E	e	E
46	9 or (9	(
47	x or X	x	X
50	illegal	illegal	illegal
51	u or U	u	U
52	f or F	f	F
53	illegal	illegal	illegal
54	illegal	illegal	illegal
55	t or T	t	T

56	g or G	g	G
57	illegal	illegal	illegal
60	illegal	illegal	illegal
61	h or H	h	H
62	j or J	j	J
63	illegal	illegal	illegal
64	illegal	illegal	illegal
65	c or C	c	C
66	k or K	k	K
67	illegal	illegal	illegal
70	illegal	illegal	illegal
71	a or A	a	A
72	q or Q	q	Q
73	illegal	illegal	illegal
74	illegal	illegal	illegal
75	s or S	s	S
76	w or W	w	W
77	illegal	illegal	illegal

Certain characters on the Flexowriter keyboard don't exist in ASCII. The following table provides ASCII substitution characters for the unique Flexowriter characters (this is compatible with the coding in the LGP30 paper tape archive):

Typewriter Code (hex)	Flex	ASCII
UC 12	delta	^
UC 1E	pi	~
UC 22	sigma	#

Certain Flexowriter codes have no character equivalent of any kind. For paper-tape reader and punch files, these are encoded as #dd, where dd is a decimal number between 0 and 63.