184

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

A FUNCTIONAL DESCRIPTION OF THE L-1 COMPUTER

Wesley A. Clark

51G-0012

March 23, 1960

The work reported in this document was performed by Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology with the joint support of the U.S. Army, Navy, and Air Force under Air Force Contract AF 19 (604)-5200.

LEXINGTON MASSACHUSETTS

A FUNCTIONAL DESCRIPTION OF THE L-1 COMPUTER

by

Wesley A. Clark

The L-1 is a very small, fast stored-program computer for use in the real-time analysis of data arising in certain RF receiver engineering problems. The computer is made of commercial transistor-circuit plug-in units and a ferrite-core memory stack, and (omitting its input-output teletype machine) occupies about 4 cubic feet of space, of which more than half is used for power supplies. It can do 80,000 10-bit operations per second and can store 256 10-bit words (numbers or instructions). The orders are very simple, like those of the TX-0 computer(1). Only two orders, add and store, have to do with stored words; the remaining orders deal only with the number held in an accumulator register and with input-output devices. The decision order, jump, is effective only when the number held in the accumulator register is positive; other orders shift this number to the right or to the left, complement it, replace it with the number zero, etc., as listed in detail in Table 1.

The provision of a single binary element called the link (designated by the letter L in the block diagram, Figure 1) is of particular value. The link simplifies operations on multiple-word numbers, and is itself a simpler and more useful version of a similar device of the Whirlwind computer(2).

In order to reduce equipment, the address of the instruction word is held in one of the memory registers rather than in a separate program-counter register.

(1) "A Functional Description of the TX-0 Computer," Peterson, H.P. and Gilmore, J.T., Lincoln Laboratory 6M-4789, Nov. 20, 1956.

(2) "Whirlwind I Operation Logic," Report R-221, Digital Computer Lab., Mann, M.F., Rathbone, R.R., Bennett, J.B., May 1, 1954.

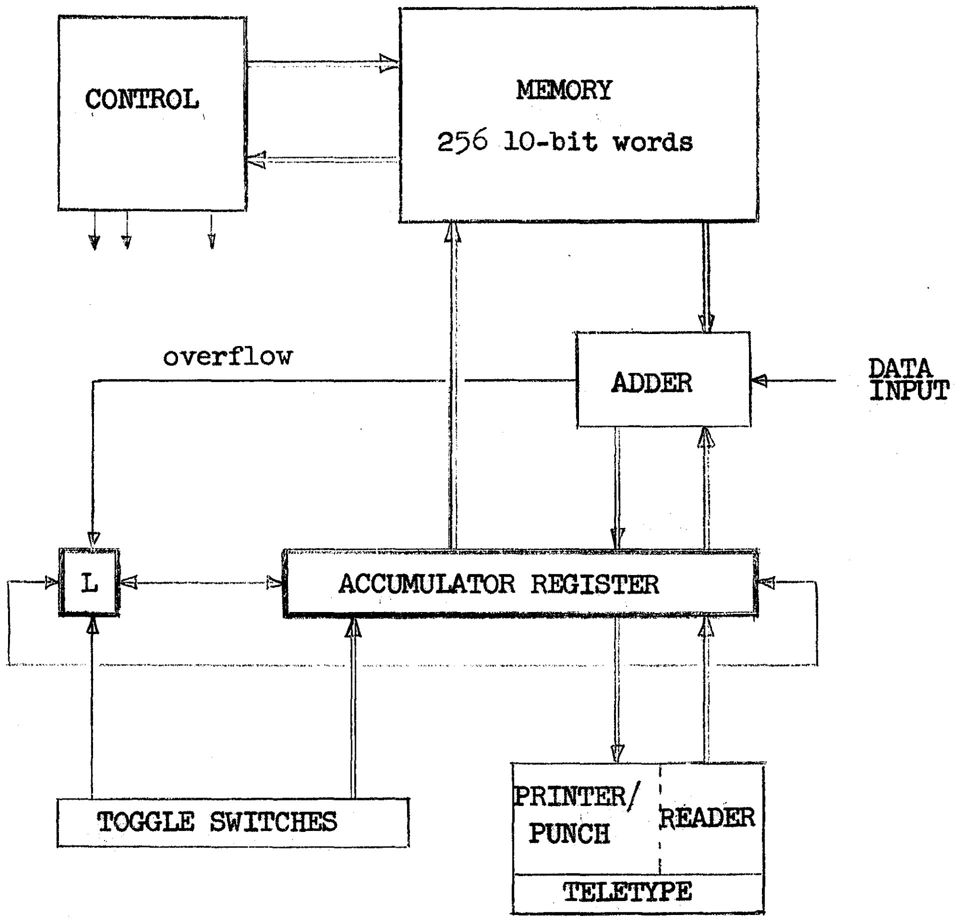


Figure 1

Block Diagram of the L-1-Computer

Information enters the computer through three different channels: teletype, toggle switches, and the data input channel. Teletype information is in the form of 5-hole code frames in punched tape. Two frames of tape are transferred in succession from the teletype tape reader to the accumulator register where they are assembled as a 10-bit word. A toggle switch is provided for each of the 10 accumulator register elements and for the link. Information is transferred from these switches to the accumulator and link by a computer instruction. Data in the form of a time sequence of 10-bit numbers are produced by an external device. A computer instruction adds the data number next in sequence to the number in the accumulator.

Computer output is in the form of 5-hole code frames punched in paper tape by the teletype printer/punch with corresponding symbols printed along the margin of the tape (Fig. 2). In addition, indicator lights and an audio signal derived from changes of state within the computer are provided.

Calculation is begun by starting the computer in its load mode, a mode of operation in which the first 32 frames of punched tape, which encode a read-in program, are transferred as 10-bit words to the first 16 registers of memory. The computer then changes automatically to its normal or calculate mode, and begins to execute the read-in program instructions now in its memory. This program, in turn, reads and stores the remainder of the tape, which is in the form of blocks of consecutively-addressed words, then jumps to the calculation program it has thus read in.

Other modes are provided in which the memory is tested by alternately storing and checking ONES and ZEROS in all registers (except the first), and in which timing of the teletype mechanism can be adjusted.

With the preceding introduction, a more detailed description can now be given:

The elements of the computer are treated as variables which take on values as instructions are executed, rather than as registers which "hold" information. The variables of the computer are the 256 10-bit memory words M_0, M_1, ..., M_{255}; a 10-bit accumulator word, A; a 1-bit link element, L; the 11-bit word, T, held in toggle switches; the 5-bit number, R_i, encoded in the i^{th} frame of the tape in the reader (the i^{th} frame is positioned at the reading station); the 5-bit number, P_j, which will appear in the j^{th} frame of the tape in the printer/punch (the j^{th} frame is positioned at the printing/punching station); the 10-bit datum, Y_k, (the k^{th}, number is the next one to be generated at the data input).

Instructions are assumed to be executed at the times t_i, i=1,2,3,..., and only the values of the variables at these times are of interest.

The convention $U \Rightarrow V$ ("U replaces V")

is used to express the equation

$$U\,(t_i) = V\,(t_{i+1})$$

where V is a computer variable and U is a function of computer variables. If U and V are binary variables, then

$$U \wedge V = \begin{cases} 1 \text{ if } U=V=1 \\ 0 \text{ otherwise} \end{cases}$$

$$U \vee V = \begin{cases} 1 \text{ if } U=1 \text{ or } V=1 \\ 0 \text{ otherwise} \end{cases}$$

$$U \circledvee V = \begin{cases} 1 \text{ if } U \neq V \\ 0 \text{ otherwise} \end{cases}$$

$$\overline{U} = \begin{cases} 1 \text{ if } U = 0 \\ 0 \text{ otherwise} \end{cases}$$

If U is an n-bit binary number and V is an m-bit binary number, then UV is the (n+m)-bit binary number formed by adjoining the binary components of U and V. The arithmetic product of U and V is written $U \cdot V$; the arithmetic sum, U+V.

Binary components of the computer variables are numbered from left to right: $A_0A_1 \ldots A_9$, $M_{x.0}M_{x.1} \ldots M_{x.9}$, etc.

The number $N = N_0N_1 \ldots N_9$ can be expressed as the decimal fraction

$$+(N_1 \cdot 2^{-1} + N_2 \cdot 2^{-2} + \ldots + N_9 \cdot 2^{-9}) \qquad \text{if } N_0=0$$

$$-(\bar{N}_1 \cdot 2^{-1} + \bar{N}_2 \cdot 2^{-2} + \ldots + \bar{N}_9 \cdot 2^{-9}) - 2^{-9} \qquad \text{if } N_0=1$$

It can also be expressed as the positive integer

$$N_0 \cdot 2^9 + N_1 \cdot 2^8 + \ldots + N_9 \cdot 2^0 \quad .$$

Other representations are possible and are used where required.

The integer $p = M_{0.2}M_{0.3} \ldots M_{0.9}$, $(0 \leq p \leq 255)$, picks the instruction to be executed ($M_{0.0}$ and $M_{0.1}$ are unused) by identifying the instruction word, M_p. In turn, M_p has two parts: an address part $x = M_{p.2}M_{p.3} \ldots M_{p.9}$, $(0 \leq x \leq 255)$, and an instruction code part $n = M_{p.0}M_{p.1}$, $(n=0,1,2,3)$. If n=3 (the order operate), x is not used as an address but specifies operations not involving the memory. Table 1 lists the actions which take place when the instruction is executed.

TABLE 1

The Order Code Of The L-1 Computer

M_p = instruction				Action	
n	x	name	abbrev.		
0	x	jump	jp x	$(A_0=0)$	$x \Rightarrow p$
				$(A_0=1)$	
1	x	add	ad x	$\text{Sum}(A,M_x) \Rightarrow A$ * $\text{Overflow} \vee L \Rightarrow L$	
2	x	store	st x	$A \Rightarrow M_x \quad 0 \Rightarrow A$	
3	128	clear	cl	$0 \Rightarrow A \quad 0 \Rightarrow L$	If $p \neq 255$
3	64	complement	cp	$\bar{A}_i \Rightarrow A_i \quad i = 0,1,\ldots,9$	$p+1 \Rightarrow p$
3	32	shift right	sr	$A_i \Rightarrow A_{i+1} \quad i = 0,1,\ldots,8$ $A_9 \Rightarrow L$ $L \Rightarrow A_0$	If $p = 255$ $1 \Rightarrow p$
3	16	shift left	sl	$A_{i+1} \Rightarrow A_i \quad i = 0,1\ldots,8$ $L \Rightarrow A_9$ $A_0 \Rightarrow L$	
3	8	read switches	rs	$T_i \vee A_i \Rightarrow A_i \quad i = 0,1\ldots,9$ $T_{10} \vee L \Rightarrow L$	
3	4	sample	sa	$\text{Sum}(A,Y_k) \Rightarrow A$ * $\text{Overflow} \vee L \Rightarrow L$	
3	2	print	pr	$A_5 \Rightarrow P_{j.0}$ $A_6 \Rightarrow P_{j.1}$ $\vdots$ $A_9 \Rightarrow P_{j.4}$ $1 \Rightarrow A_i \quad i = 0,1,\ldots,9$ $1 \Rightarrow L$	
3	1	read tape	rt	$R_{i.0} \Rightarrow A_0 \quad R_{i+1.0} \Rightarrow A_5$ $R_{i.1} \Rightarrow A_1 \quad R_{i+1.1} \Rightarrow A_6$ $\vdots \qquad \vdots$ $R_{i.4} \Rightarrow A_4 \quad R_{i+1.4} \Rightarrow A_9$ $1 \Rightarrow L$	

* Sum and overflow functions are defined as follows:

$$\left.\begin{array}{l} \text{Sum}_i(A,B) = A_i \odot B_i \odot C_{i+1} \\ C_i = A_i \wedge B_i \vee (A_i \vee B_i) \wedge C_{i+1} \end{array}\right\} \begin{array}{l} i = 0,1,\ldots,9 \\ C_{10} = 0 \end{array}$$

$\text{Overflow} = C_0$

As an example of the use of the link in multiple-word arithmetic, the following program segment illustrates addition of the 20-bit numbers M_1M_2 and M_3M_4:

```
 .
 .      SUM (M1M2, M3M4) => M5M6
 .      ________________________
cl
ad 2
ad 4    (overflow => L)
st 6    (note 0 => A)
sl      (recovers previous overflow)
ad 1    (adds in previous overflow)
ad 3
st 5
 .
 .
 l
```

A second example illustrates the use of the link in shifting operations. The following program segment shifts the 10·n bit number one place to the right:

```
 .       ------->
 .      M1M2...Mn
 .      _________
cl
ad 1
sr      (M1.9 => L)
st 1    (0 => A)
ad 2
sr      (L => A0, M2.9 => L)
st 2    (0 => A)
 .
 .
 .
ad n
sr
st n
 .
 .
```

A final example illustrates the multiplication of two 20-bit numbers yielding a 40-bit product:

$M_1M_2 \cdot M_3M_4 \Rightarrow M_5M_6M_7M_8$

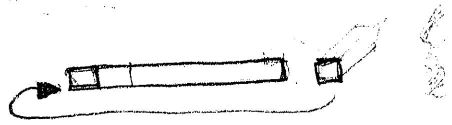
Assume: $M_{1.0} = M_{3.0} = 0$ (positive numbers)

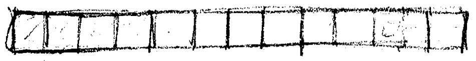
```
100)      cl
          ad  135
          st  136
          st    5 }  puts 0 in 5 and 6
          st    6 }     Set up
          ad    3
          st    7
          ad    4
          st    8
          jp  121
          ________

110)      ad    6
          ad    2
          st    6       Add
          sl         for over flow
          ad    1
115)      ad    5

          sr
          st    5
          ad    6
          sr
          st    6       Shift right
121)      ad    7 }     one place
          sr      }
          st    7 }
          ad    8 }
          sr      }
          st    8 }
          ad  136
          ad  137
          jp  200       (finished)
          ---------
          st  136
          sr
          jp  115    if A0=1
          ---------
          cl
          jp  110
          ________

135)      -20
136)      (counter)
137)      +1
```





$$\overset{\frown}{-1} + 1 = 0$$

$$-9 \quad \begin{array}{r} 9 \\ +1 \\ \hline 10 \end{array}$$

conditional

when finished A₀=0
[illegible] could be any other no.
when [illegible] counting

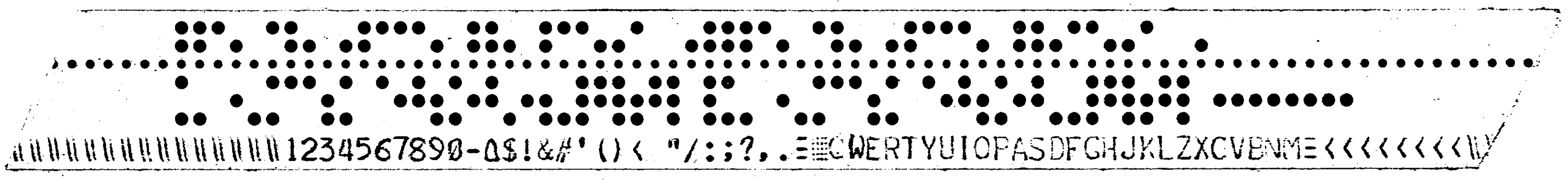


Figure 2. Teletype Tape used in the L-1 Computer