MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

NOX

THE LINCOLN WRITER

J. T. Gilmore, Jr.

R. E. Savell

Approved

Group Report 51-8

October 6, 1959

The work reported in this document was performed by Lincoln Laboratory, a technical center operated by Massachusetts Institute of Technology with the joint support of the U.S. Army, Navy, and Air Force under Air Force Contract AF19(604)-5200

LEXINGTON MASSACHUSETTS

THE LINCOLN WRITER

by

J. T. Gilmore, Jr. and R. E. Savell

ABSTRACT

The Lincoln Writer is a new piece of terminal equipment for the TX-2 computer (see Figure 1). This report describes the unit, the reasons for building it, and how to use it.

INTRODUCTION

The Lincoln Writer is a data processing device which is used to communicate directly as well as indirectly with the computer. It consists of an electric typewriter, a separate electric coding keyboard, a paper tape punch, a paper tape reader, and a control console. The unit itself is not a new concept although it does have some original features. The main reason for its creation is the need to improve the language through which information is exchanged between man and machine. The typewriter in the unit has a specially designed type consisting of 88 symbols all of which can be typed in any of three script positions, (superscript, normal and subscript). The coding arrangement of the Lincoln Writer provides these symbols plus typewriter functions (and a few special signals) to be generated or detected as separate pieces of information. The original 88 symbols when used as separate characters, or when combined with others (by back spacing etc.) to form composite characters, provide the programmer with the ability to express arithmetic, mathematics, symbolic logic, Algorithms, and alpha-numeric language.

Programmers will use the Lincoln Writer in two ways:

1. They will prepare their original program on paper tape using one of the Lincoln Writers in the tape preparation room.

2. While they are at the console and their program is still in memory or on magnetic tape (after just having made an error or having completed a given task) they will be able to interrogate and command the computer's utility system regarding their program in its original programming language.

Man-Machine Communication Techniques

Our first experience with direct communication between man and machine occurred with the TX-0 computer. It was the first computer to possess a memory of 65,536 magnetic core registers, and also had a direct input-output typewriter. With these two features at our disposal, we wrote a large utility program which was capable of transmitting and receiving typewriter character codes and which would understand a limited English vocabulary as well as our standard programming language. We stored the program in the last 5000 registers of the memory and made that area "off limits" to other programs. In that way the utility program was always available for service to other programs. It provided the programmer at the console with the ability to:

1. Examine and/or modify his program and its parameters in programming language.

2. Search memory for particular constants, instruction words, or parts of same.

3. Introduce intermediate stopping or trapping points which switched control to the utility program thereby allowing the programmer to examine data and program variables and then continue his programs operation.

4. Request high speed (2000 characters per second) print-outs of storage indicating range, format and language.

5. Request punched binary copies of corrected programs, data etc.

The various routines in the utility program which provided these services were requested by typing their English names (e.g. "Prince" for print-out routine). They would in turn request, in English, the information needed using either lengthy explicit questions or single words depending on the programmer's preference. Whenever a programmer forgot the limitations of the utility program's vocabulary or misspelled a word he was informed of it. (See Figure 2.)

We learned that the time spent at the console in debugging a program or varying parameters for other runs was not always wasteful. In fact the extra computer time used for console communications was more often justified by the large decrease in real time to complete a given program - and most important of all by the high morale of the programmers who managed to accomplish something each time they used the computer - even if it was the discovery of why they should give up the rest of their scheduled time to study the logic of their programs.

The kind of research programs which were operated on TX-0 required that the computer be used more as an experimental tool than as a calculator for equations whose solution-methods were already known. The ability to stop and examine and modify data at the console became extremely useful and almost indispensable. We must concur, however, that excessive on-line communication time at a computation center's computer would be difficult to justify until either a multiple communication system or schedule sharing system were to be incorporated. A multiple communication system has several time-sharing on-line typewriters which communicate with one utility

system regarding information in memory or on magnetic tapes. A schedule-sharing system (which we prefer) is one which provides at least one tape unit for each on-line typewriter and has at least two or more of these in-out combinations. Only one of these combinations can be connected to the computer at a given time but to change to another only requires the push of a button. The schedule-sharing comes about in the following way. When programmer "A" who has been using the computer at his on-line typewriter realizes that he must spend the next several minutes thinking about one of his errors or writing a small programming change, then, instead of monopolizing the computer time he requests a memory "dump" on his own private magnetic tape. (On TX-2 the maximum length of time for this would be 30 seconds.) He then surrenders control of the computer to programmer "B" who has been waiting (and thinking) at his typewriter console. "B" then reads his program and data into memory from his private magnetic tape and assumes full control of the computer at his typewriter. This type of operation is of course primarily meant for debugging periods but could also be useful for certain kinds of programs already debugged.

Preliminary Study Using the TX-0 Computer

With the advent of the TX-2 computer and its efficient but complicated logic we knew that the typewriter in our present tape preparation unit and its list of characters would limit us in our ability to express TX-2 programming language. The decision to design and buy a new typewriter meant a complete change of console typewriters and tape preparation units. In addition the same characters chosen for the new typewriter also had to be incorporated in our high speed printer in order for memory "dumps" to be expressed in the same programming language. It was quite obvious that the

selection and style of the characters and the characteristics of the new typewriter were most important and worthy of careful study.

Rather than starting with a compromise in the search for a good typewriter and list of characters we decided to simulate a typewriter on TX-0 by using its point-by-point display oscilloscope and its photo diode light pen. In this way we were able to experiment with our ideas on characters and typewriter functions. The program was called "the Scope Writer" and simulated a 200 character typewriter. The typewriter's characters were capable of being drawn on the scope by the light pen using a point matrix two inches square. Then by light pen control the drawn character was reduced to the desired size, the 200 spot keyboard displayed, and the key designated to carry the new character was then touched by the light pen. (A plastic sheet was used to pencil sketch the characters over the corresponding key spots once they were assigned.) In addition to its 200 character keyboard the Scope Writer had the following functions keys (i.e. spots): (1) carriage return (2) tab (3) space (4) superscript selector (5) subscript selector (6) normal script selector (7) line feed-up (8) line feed-down (9) backspace.

The printed page of the Scope Writer was the upper half of the display scope. A typing indicator was used to show the pen typist where his next character was going to appear. The carriage return moved everything on the scope page up one position and returned the typing indicator to the left margin. The tab moved the typing indicator to the right m character positions (m was variable). The superscript and subscript selectors moved the typing indicator up (super) or down (sub) by half the character height and all the characters which were selected while superscript or subscript was in effect were reduced in size by one half. The normal script selector

returned the writing indicator to the normal position and restored normal character size. The backspace moved the typing indicator to the left one full character position. The line feed-up and line feed-down moved the scope page down or up respectively without moving the typing indicator. In addition to these controls, there were also controls for erasing mistakes via the light pen.

With such a tool at our disposal it was easy for us to experiment with the characters nominated at our "character" meetings. Although the Scope Writer was interesting it proved to be unnecessary since it merely demonstrated the characters and typewriter capabilities which were agreed upon rather easily at our meetings. However, the program was later modified so that groups of characters printed on the scope page could be remembered and assigned to given keys. This feature plus a routine which allowed the light pen to "drag" characters or a group of characters about the scope page provided us with the ability to draw diagrams as well as to type text, (see Figure 3). So although the Scope Writer program played a small part in the development of the Lincoln Writer it did demonstrate the usefulness of the light pen in man-machine communications and convinced us that the utility program for TX-2 should use both the scope and light pen as well as the Lincoln Writer for on-line communication work.

The Lincoln Type

Figure 4 shows the characters chosen for the Lincoln Type.[1] Since there was no typewriter commercially available with more than 44 key levers we were forced to select just 88 characters. We discarded 14 of the small alphabetical letters and retained those 12 that are favorite subscripts and variables (h, i, j, k. n, t, p, q, w, x, y, z). This allowed us to include

1. See Fig. 12 for High Speed Printer's Character Matrix.

6 Greek letters and 8 symbolic logic characters. It also meant that our text typing would be all capital letters so we chose a clear style and made them smaller than standard capitals.

The typewriter which was chosen for the Lincoln Writer unit is a Soroban Computeriter which is an IBM Model B typewriter with a special decoding package. The maximum printable area of this typewriter's key slug was determined to be a rectangle _____" x _____". All of our characters were designed with respect to this rectangle. With the exception of the parentheses, braces, comma, asterisk, overbar, underbar, and prime, all other characters were centered in the rectangle. The underbar, overbar, box and circle were selected to be dead characters - i.e. no advance of the platen after they are typed. The box character just fits inside of the printing rectangle and can contain any other character. For contrast, the capital letters are larger than the numerals and small characters. The numerals are slanted for the same reason. With the exception of the capital letters and a few punctuation characters, all characters can be circled. It was our intention that by choosing the right 88 symbols a great many of the nominated characters which could not be included could be produced by some combination of the 88 in various script positions. See Figs. 5 & 6 for some Lincoln Writer typing examples.

Lincoln Writer Code

The paper tape photoelectric reader and high speed punch associated with the TX-2 computer handles a 6-bit code marked by a 7th hole information channel. Since the Lincoln Writer is primarily a program tape preparation unit for TX-2 its code is also 6 bits. See Fig. 7 for the code itself.

To encourage language compatability between TX-2 and TX-0 the Lincoln Type has been installed in a Flexowriter which will be lent to TX-0 users, see Figure 8 for Flexowriter code for Lincoln Type.

The Lincoln Typewriter and Keyboard

We originally planned for the typewriter to transmit as well as receive 6-bit information. This required an encoding package as well as a decoding package installed in the typewriter. The difficulty arose when we studied the typing problem. The lower case keyboard was almost standard (our capital letters were put on the lower case). But the upper case characters had no standard positions and required visual reference to 44 key buttons with two characters printed on each button. Furthermore the selection keys for super; sub; and normal-script as well as stop, delete, line feed-up and line feed-down were to be mounted on an auxiliary keyboard. When we tried to visualize the difficulty that a professional typist would have switching from one case to another and selecting script positions on a separate keyboard, we shuddered at what our "hunt and peck" programmers would do. As a result we decided to make the typewriter completely passive and do all the typing on one special keyboard with a key-button for each character and function.

The keyboard is actually two separate Soroban coding keyboards mounted on the same block. The lower keyboard contains the buttons for all the lower case characters and the typewriter functions. The upper board contains the buttons for upper case characters and a few special codes[1]. There was no need to have a 1:1 position relationship between upper case

1. Those codes not used to designate a character or function are used as special signals to a utility program for on-line operation, e.g. "Yes," "No," "Start" etc.

and lower case key buttons having the same code. Because of this we were able to arrange the upper case characters according to groups, i.e. Greek letters, punctuation symbols, logic, arithmetic, etc. To further aid the "hunt and peck" programmer typist, we color coded these groups. (See Fig. 9.)

Although the coding keyboard generates only 6 bit codes its internal circuitry utilizes a 7th bit, which is generated by each keyboard to indicate whether the key actuated is either a function key or a character key. In the case of a character key this bit is compared with the status of a relay which remembers the case of the previous character typed. If the cases are the same the new character's 6-bit code is immediately transmitted. If they differ the circuitry first transmits the proper case shift 6-bit code before transmitting the character's 6-bit code. For a function key there is no sensing of what case it is in and therefore no case shift code is ever transmitted - just the code for the function itself.

The keyboard is mechanically and electrically interlocked to prevent any character from being typed before the previous character has been transmitted. It also has an anti-repeat circuit to prevent a character from being transmitted twice if a key is depressed too long.

The typewriter is equipped mechanically to decode 6-bit characters and print them at about 10 characters per second. It will not, however, decode the color change, superscript, normal, subscript, line feed-up or line feed-down functions. These are decoded separately. The platen rotator which enables the above functions (except color change) to be performed was designed at Lincoln Laboratory. It consists of two rotary solenoids and appropriate gearing to rotate the platen $\frac{1}{2}$ line space up or down. Except while actually moving the platen, the mechanism is

disconnected from it so that the platen may be rotated by hand. The regular platen detent on the typewriter is used to hold the platen in position once it has been rotated.

When the typewriter receives the code for a carriage return it will also set itself to lower case and normal script without having to receive those corresponding codes. This fact is known and included in the logic of the keyboard circuitry as well as in all programs which deal with the Lincoln Writer code. It is worthy of note that a carriage return is all that is necessary to synchronize case and script between an input-output program and the Lincoln Writer. It should also be emphasized that the keyboard and the typewriter are synchronized with respect to case and can be considered synchronized with respect to script since any redundant script selections from the keyboard will be ignored by the typewriter. For example if a programmer were communicating with the computer via the Lincoln Writer and his last remark left the typewriter in lower case and normal script and the computer responded and left the typewriter in upper case and subscript, then the keyboard relay would be reset to upper case and the subscript light indicator adjacent to the keyboard would be turned on.

The line feed selectors do not affect the last remembered script position of the typewriter. Therefore after one or more line feed selections, care must be taken to return the typewriter platen to its original position by the selection of the opposite line feed. For example, if the typewriter platen were in normal script position and a line feed-up were selected - followed by printable characters, the printed result would be the same as if a superscript had been selected. However, if this were to be followed by a normal script selection there would be no change since

the typewriter would act as if it was already in normal script, (since no other script had been selected). Consequently the line feed selections should be used for situations like superscripting a superscript or subscripting a subscript, etc. On these occasions the opposite line feed should be used to return the typewriter to its original script position.

The Paper Tape Reader

The reader is a Soroban product and can operate at either 9.5 or 19 characters per second. The lower speed is used whenever the typewriter is connected since it can only print at 10 characters per second. The reader's circuitry is designed to allow flexible operation as will be seen in the description of the control panel.

Tapes may be rewound on the reader. Accidental rewinds which would damage the tape are prevented by interlocking the rewind circuit so that the reader must be off and the "no tape" contact in the "no tape" position before rewind will take place. The reader is also interlocked with the keyboard so that they will not transmit simultaneously.

The Paper Tape Punch

The punch is a modified Commercial Controls product (Model 2) which can punch at speeds up to twenty characters per second. It is mounted on a sliding shelf which has two extended positions for ease of operation and maintenance.

The Control Panel

Figure 10 is a block diagram showing the relationship of the component parts of the Lincoln Writer. It should be noted that there are two separate paths from the input units (those which generate characters) to

the output units (those which receive characters). The keyboard or reader may drive the punch, typewriter, or computer, either separately or simultaneously via one path, and the computer may drive the typewriter and/or punch via the other path. Two input units may not simultaneously drive the same output unit, however. Connections between the units are made by push button switches located on the control panel. (See Figure 11.) Illuminating pushbuttons are used to clearly show which connections and modes are active. There are also lights to indicate script position. Lights will also be added to indicate case because of the following reason: During normal operation of the Lincoln Writer there is no need for separate selection of case since it is done automatically. However, in the event that a tape is being read and is to be supplemented by occasional insertions from the keyboard, there is a possibility that the reader and typewriter will not be synchronized with respect to script and/or case after one of these insertions. Before making an insertion via the keyboard one must first note the script and case of the typewriter and return it to same after the insertion. This could require the request for a case change without generating another character which will be accomplished in the future by pushing the desired case's light indicator.

To avoid this, any insertions or deletions should be full line modifications which would take advantage of the synchronizing of the carriage return.

One mode of Lincoln Writer operation would have all units active. The programmer would use the keyboard to communicate with the computer and at the same time transmit information to the typewriter and to the paper tape punch - thereby printing and punching a record of what he was transmitting to the computer. At the same time when it became the computer's

turn to communicate to the programmer it would transmit information both to the typewriter and to the puncn thereby recording its side of the conversation as well.

To communicate with the computer, two cables must be used to connect the Lincoln Writer with the computer's in-out control (buffer registers, necessary control circuitry for timing signals, etc.). At present the Lincoln Writer must be in the computer room for this operation but in the future it is hoped that the cables may be extended a great deal further - perhaps to the tape preparation room or to a programmer's office. The Lincoln Writer was primarily designed as a tape preparation system and because of this it was necessary to make the system as portable as possible. All circuits and power supply necessary for interconnection of the parts in the system are contained within the console in order that they be used in any location simply by plugging into a 110 volt outlet.

The control panel was designed to be self explanatory but the punch and reader controls should be discussed briefly. The end mark button will cause a 6-bit code (72) without a 7th hole to be punched on paper tape. This character is used by the TX-2 photoelectric paper tape reader to sense the end of a tape. The tape feed button will cause blank tape to be punched as long as it is pressed. The spare button will eventually be used to cause a character to be punched, the code of which will be determined by seven small toggle switches - (yet to be installed).

The reader control buttons are bi-colored red and green and engraved with the possible modes of operation. Colored lights behind the engraved names of the various modes are illuminated to indicate which modes have been selected.

The following are the capabilities and selectable modes of the Lincoln Writer reader:

Kind of Character	Character's Code in Octal	Mode Color	Mode Function
"Stop" character	76	Green Red	"Stop" Reader "Read" and transmit code to punch and/or computer
"Delete" character	77	Green Red	"Ignore" "Read" and transmit code to punch and/or computer
"No 7th hole" character	--	Green Red	"Stop" Reader "Read" and transmit to punch, i.e. "Reproduce"
Blank tape	--	Green Red	"Ignore" "Reproduce"

Those modes which are color indicated in green are usually selected when Lincoln Writer-produced tapes are to be read. Those modes indicated in red are usually selected when binary computer produced tapes are to be read (and reproduced).

The start, stop and rewind buttons are obvious. The ability to read one line at a time is provided when the operator holds the stop button in the down position and pushes the start button once per line. A separate "line at a time" button will be added later if programmers feel the need for it.

Summary

The Lincoln Writer provides the means for improving the language between man and machine. With the right kind of utility program it can be a computer console as well as a tape preparation unit. It will make its most significant contribution if it is used by programmers who will look ahead and see that saving man's time is much more important and valuable than saving computer time.

John T. Gilmore, Jr.

Robert E. Savell

LIST OF FIGURE CAPTIONS

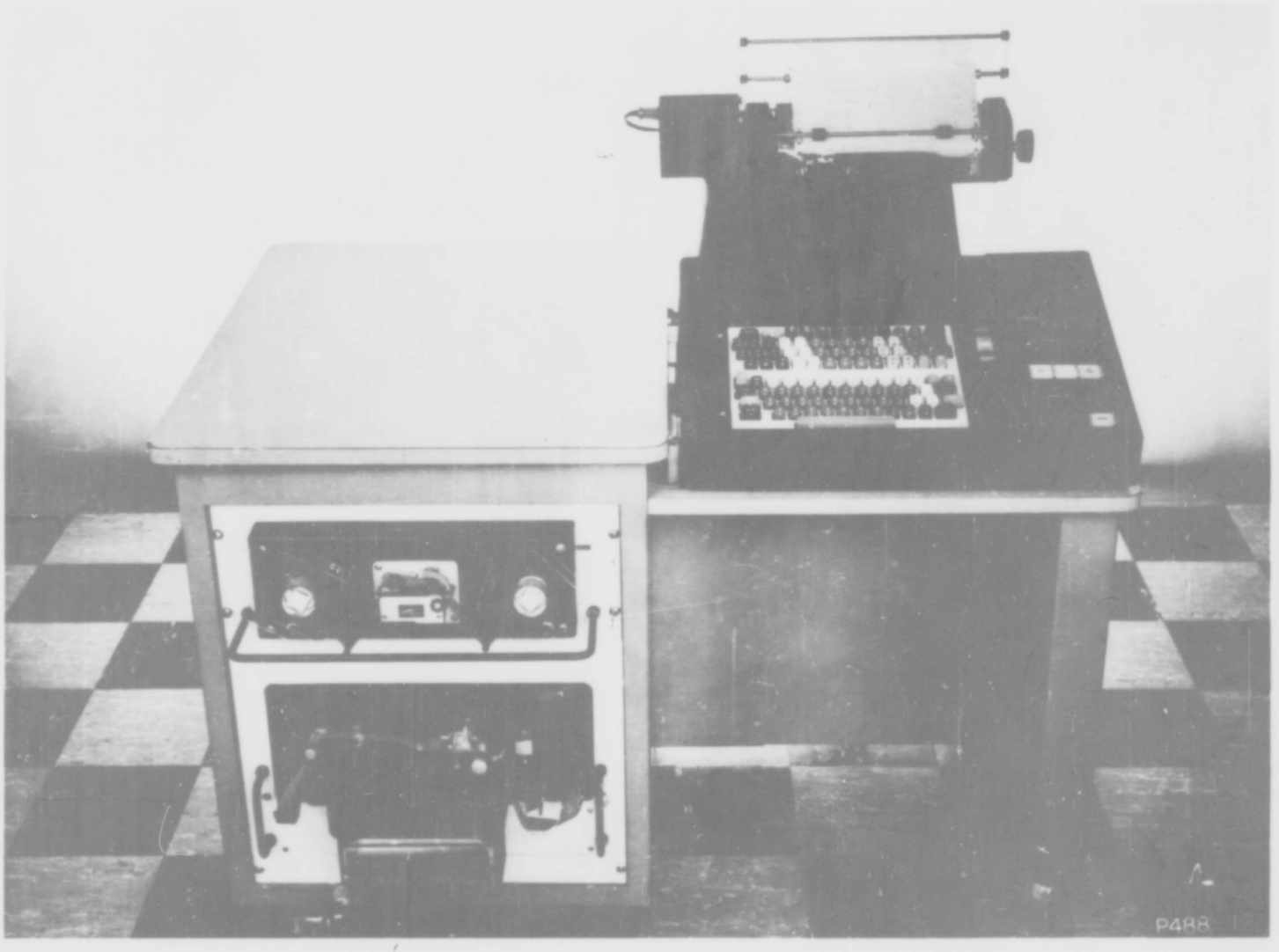


Figure 1 Lincoln Writer Full Picture

EXAMPLE OF TX-0 TYPEWRITER CONVERSATION

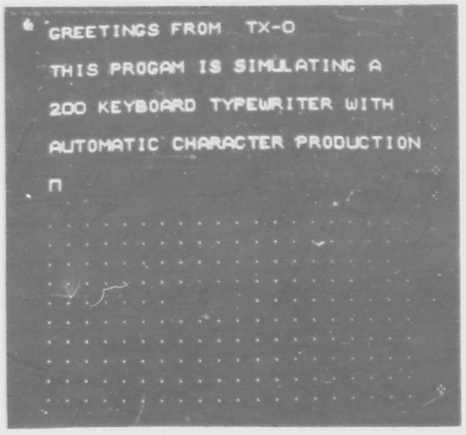
The Programmer's Typing Is Underlined Since This Report Is In One Color

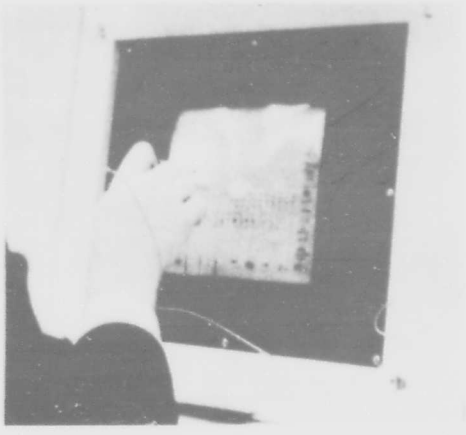
princr
error princr
prince
do you want vertical column layout be brief
yes sir
yes
instructions no
first 7150
last 7177

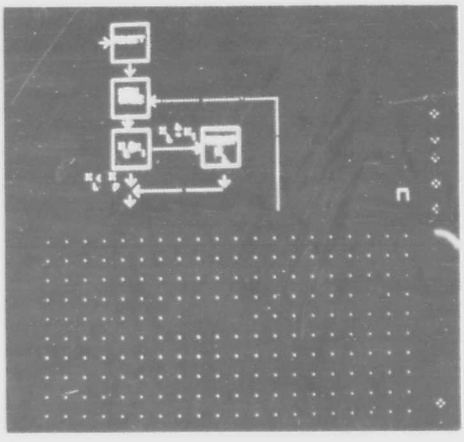
```
7150|   120
  51| 13355
  52|  6101
  53| 10311
  54|   664
  55|  5521
  56| 74240
  57|   155
  60|   125
  61|  3265
  62|  1432
  63|   404
  64|   431
  65|    64
  66|   441
  67|   144
  70|    55
  71|   231
  72|     5
  73|    32
  74|     2
  75|    35
  76|   231
  77|    42
```

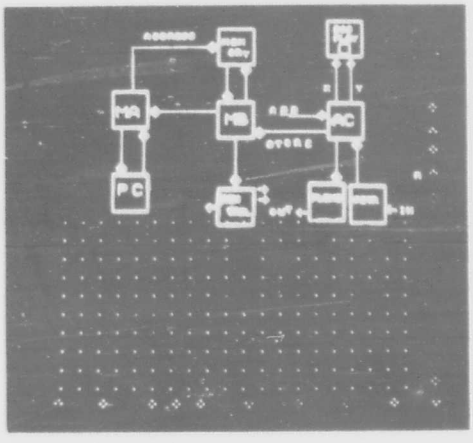
more no
finished
hark → "hark" was the name of the direct input-out conversion program which allowed the programmer to examine and/or modify individual words in his program.
7137| =sto
161041| =sto
1617xxx → nullify signal
160741| =sto 62623.
160742| =sto 7
7003| =add 33214 add 55051
6776| =add 55050 add 76706
7003| =add 55051
6776| =add76706
161161|| → the double bar indicated that computer control was to be transferred to register 161161.

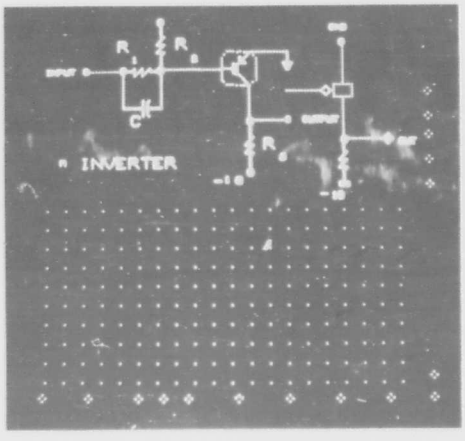
Figure 2











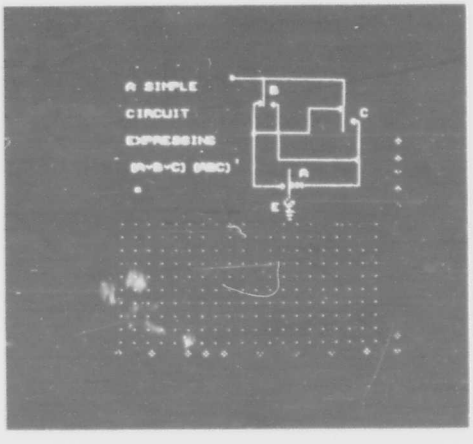


Figure 3 Scope Writer Examples

```
26      A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
12                    h i j k     n   p q     t       w x y z
 6      α β γ Δ ϵ             _ λ
 9      ■          → ~              ∨  ∧  .  ∪ ⊂  ∩ ⊃
10      0 1 2 3 4 5 6 7 8 9
 9      = + − × · / < > ∑
11      ‖ | ( ) { } # % ' ? ,
 4      ¯ _ □ ○
 1      ☞
---
88
```

KEY SLUG POSITIONS OF THE SYMBOLS

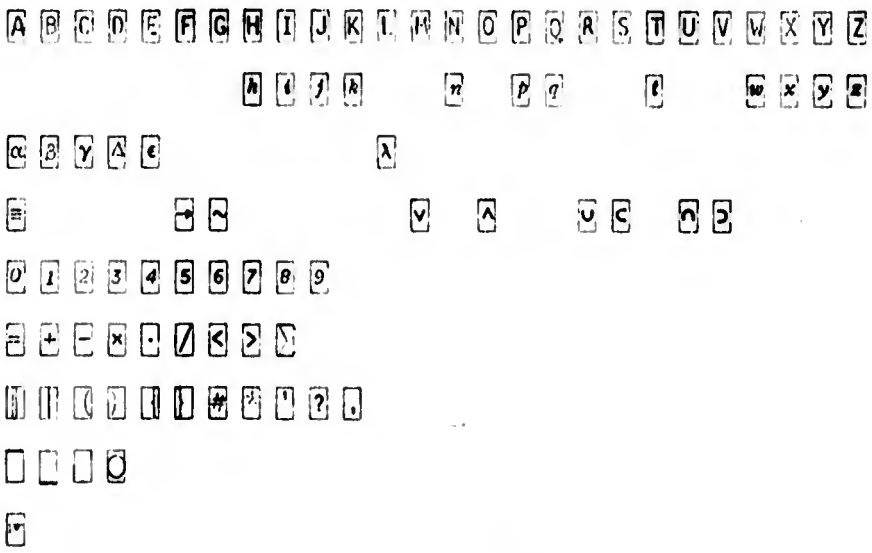


Figure 4

TYPING EXAMPLES FROM THE LINCOLN WRITER

$$\{(x,y)(x-\alpha)^2+(y-\beta)^2<\epsilon\}$$

$$\alpha\cup(\beta\cap\gamma)=(\alpha\cup\beta)\cap(\alpha\cup\gamma)$$

$$F(X)=\alpha_n X^n+\alpha_{n-1}X^{n-1}+\cdots+\alpha_1 X+\alpha_0$$

$$F(\alpha)=\sum_{i=1}^{M}\ (\sum_{i=1}^{N}\alpha_{ij}\alpha_i)\,\alpha_j'$$

$$\begin{vmatrix}\alpha_{11}&\alpha_{12}&\cdots&\alpha_{1N}\\ \alpha_{21}&\alpha_{22}&\cdots&\alpha_{2N}\\ \cdot&\cdot&\cdot&\cdot\\ \alpha_{M1}&\alpha_{M2}&\cdots&\alpha_{MN}\end{vmatrix}$$

$$\overline{y^2}=\overline{x^2}+\overline{x^2}=S+N$$

$$Y(w)=\epsilon^{(\quad w)}+iB(w)$$

Figure 5

```
END OF POS FORM→‖  LDE          PART'S LINE POS                    <43·21>
                   STE          F TABLE + X_FORM + ΔLINE
          F1→      hJPX_FORM    F2          <-1>
                   SKZ MKN      2ND TRY BIT
                   ALARM        ∑NO RULE FOR THIS STRING
                   RSX_LINE     F TABLE +1 + ΔLINE                 <43·21>
                   JUMP         TRY AGAIN
          F2→      DPX_FORM     # ELEM                             <43·21>
                   RSX_INDEX    (+1}
                   RSX_ELEM     # ELEM
          F2·1→    RSX_FORM     # ELEM
          F3→      hJPX_FORM    F4          <+0>
                   JUMP         FORM FOUND
          F4→      LDE          F TABLE + X_FORM                   <43·21>
                   SED          {F SRCH* + X_ELEM} * + X_FORM
                   hJPX_FORM    F3          <-1>
                   SED          {-0}                               <43·21>
                   hJPX_INDEX   F2·1        <+1>
                   RSX_FORM     #ELEM                              <43·21>
                   JUMP         F1

     3·4Y_t→       SIM SRCH [6] ,, DEF^t[4] , w^ε
F TABLE→           567** , +0

2ND TRY BIT ≙ (2·1) TEST BITS

E·T·C·
```

Figure 6

LINCOLN WRITER CODE

00	0	⌐
01	1	∑
02	2	|
03	3	‖
04	4	/
05	5	×
06	6	#
07	7	→
10	8	<
11	9	>
12		¯ _
13	○	□
14	Read In	
15	Begin	
16	No	
17	Yes	
20	A	∩
21	B	⊂
22	C	∨
23	D	∘
24	E	γ
25	F	ι
26	G	ω
27	H	χ
30	I	ι
31	J	ψ
32	K	ζ
33	L	?
34	M	∪
35	N	∩
36	O	∫
37	P	∂

40	Q	α
41	R	Δ
42	S	ρ
43	T	ε
44	U	η
45	V	⊃
46	W	β
47	X	∧
50	Y	λ
51	Z	~
52	(	{
53	)	}
54	+	≡
55	-	≈
56	,	'
57	.	☆
60	CAR RETURN	
61	TAB	
62	BACK SPACE	
63	COLOR (BLACK)	
64	SUPER	
65	NORMAL	
66	SUB	
67	COLOR (RED)	
70	SPACE	
71	Word Exam	
72	Line Feed Down	
73	Line Feed Up	
74	(LOWER CASE)	
75	(UPPER CASE)	
76	STOP	
77	NULLIFY	

Figure 7

THE LINCOLN TYPE ON THE FLEXOWRITER

26	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
12	*h i j k n p q t w x y z*
6	α β γ Δ ϵ λ
10	0 1 2 3 4 5 6 7 8 9
9	≡ ~ → ∧ ∨ ⊃ ⊂ ∩ ∪
9	= + − × / · ∑ < >
11	| ‖ # * { } ' ( ) , ?
1	←

84	

Note that the box, circle, underbar, and overbar are missing.

00		
01		
02	E	γ
03	6	#
04		
05	,	'
06	A	n
07	1	∑
10	SPACE	
11	+	≡
12	S	p
13	2	|
14	I	i
15	−	=
16	U	h
17	0	←
20	COLOR	
21	)	'
22	D	q
23	3	‖
24	R	Δ
25	.	*
26	J	y
27	5	×
30	N	∩
31	(	{
32	F	t
33	4	/
34	C	∨
35	9	>
36	K	z
37		

40	T	ϵ
41		
42	Z	~
43	BACKSPACE	
44	L	?
45	TAB	
46	W	β
47		
50	H	x
51	CAR RET.	
52	Y	λ
53		
54	P	k
55		
56	Q	α
57		
60	O	j
61		
62	B	c
63		
64	G	w
65		
66	7	→
67		
70	M	∪
71	UPPER CASE	
72	X	∧
73		
74	V	⊃
75	LOWER CASE	
76	8	<
77	DELETE	

Figure 8

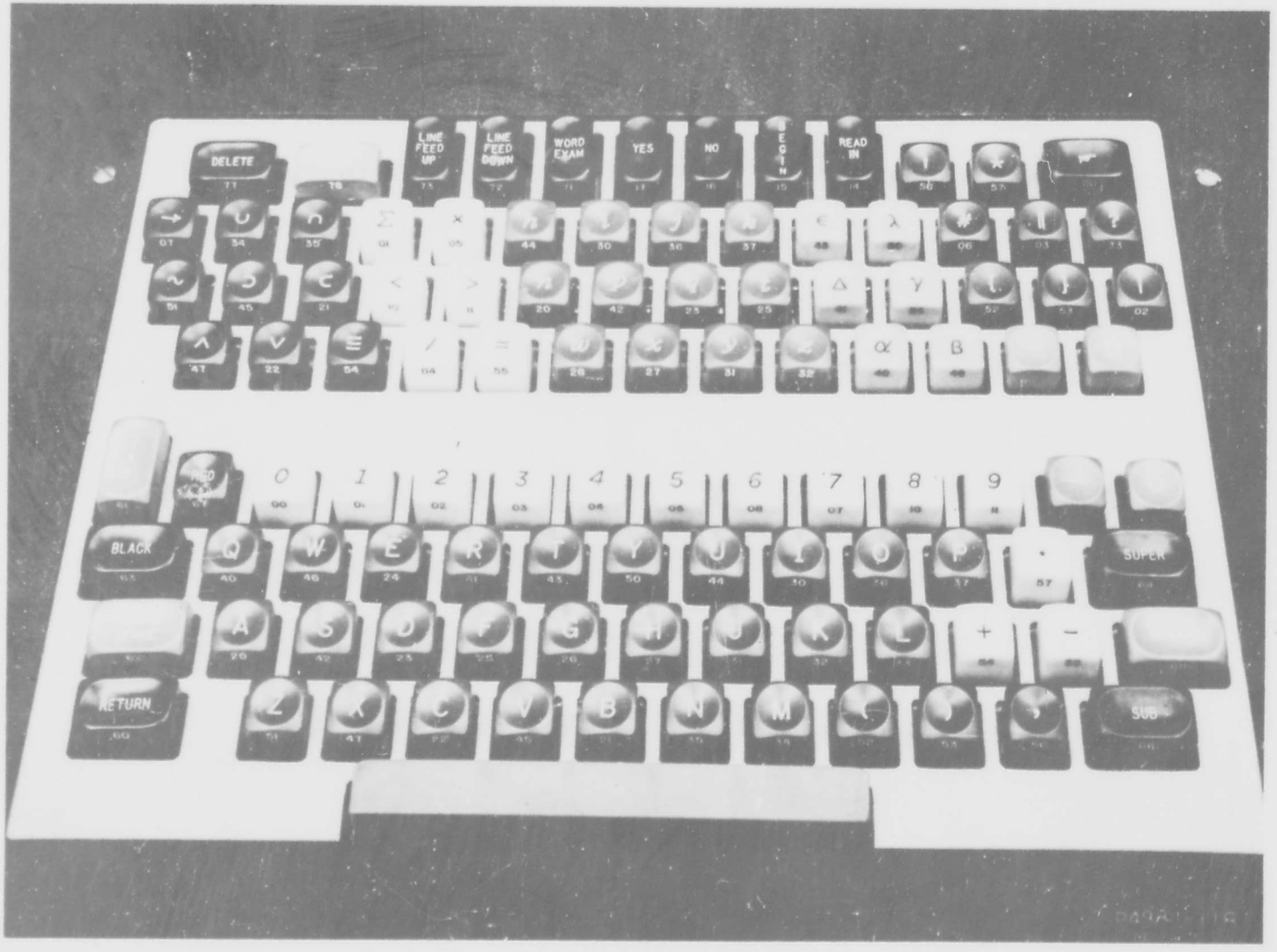


Figure 9 Lincoln Keyboard (Close Up)

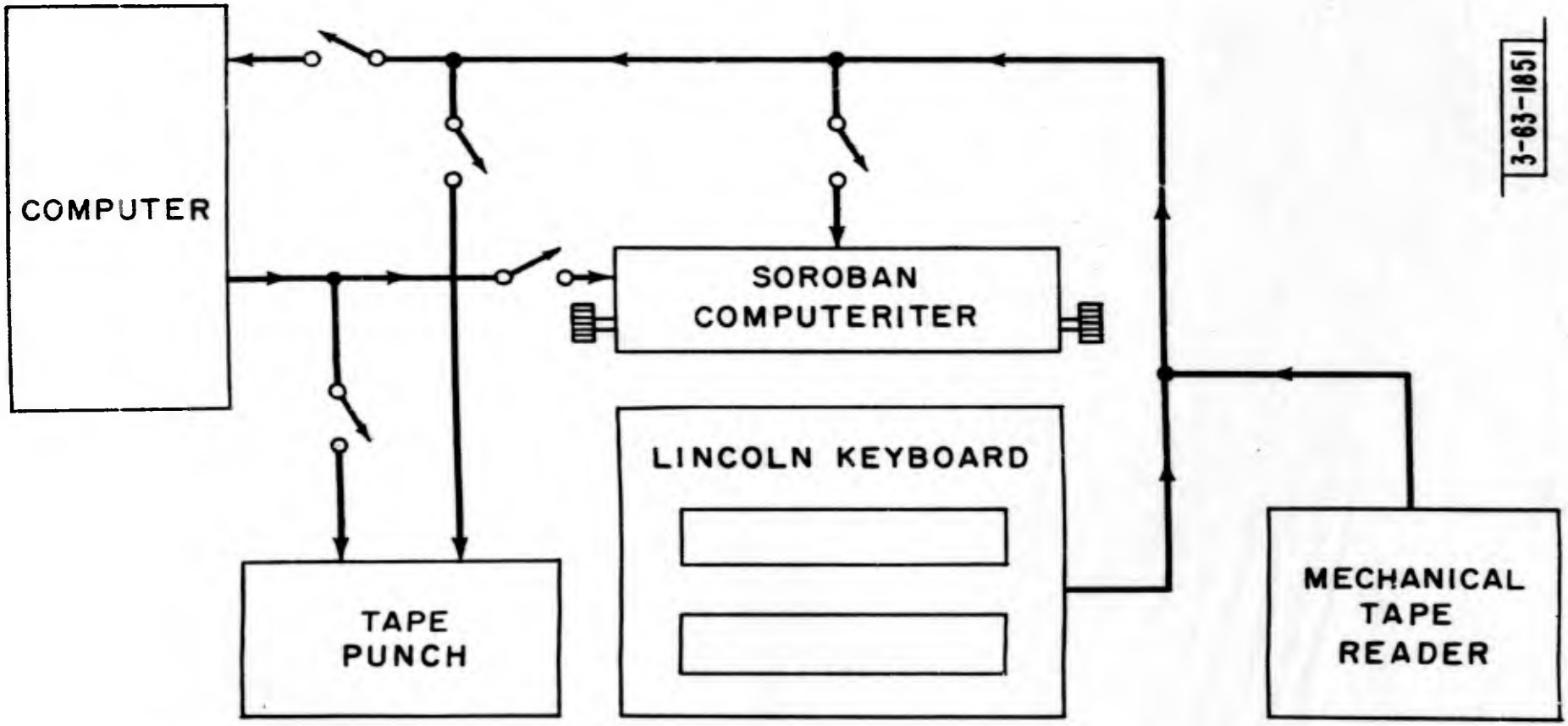


Figure 10 Lincoln Writer Block Diagram

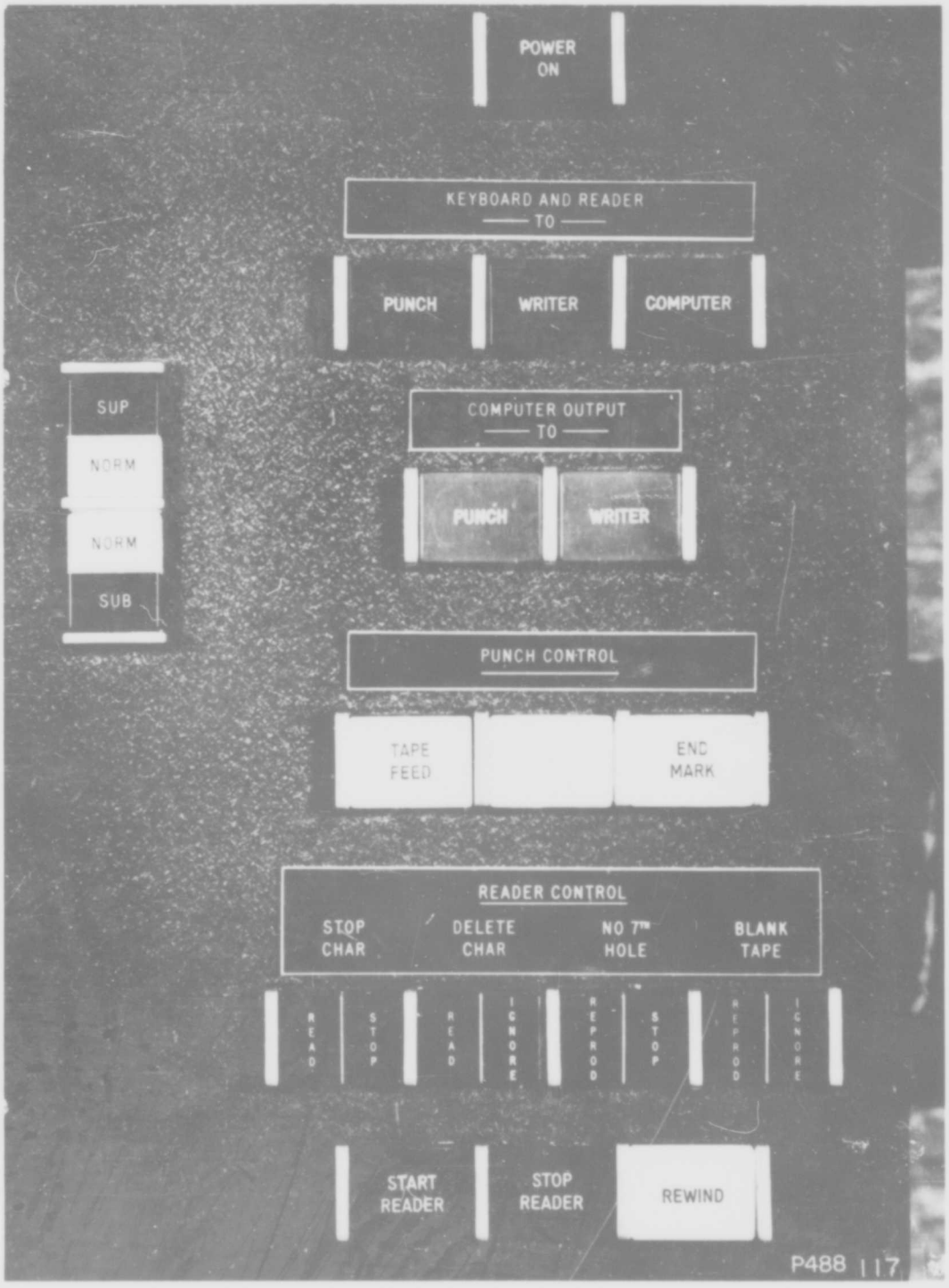


Figure 11 Lincoln Writer Control Console (Close Up)