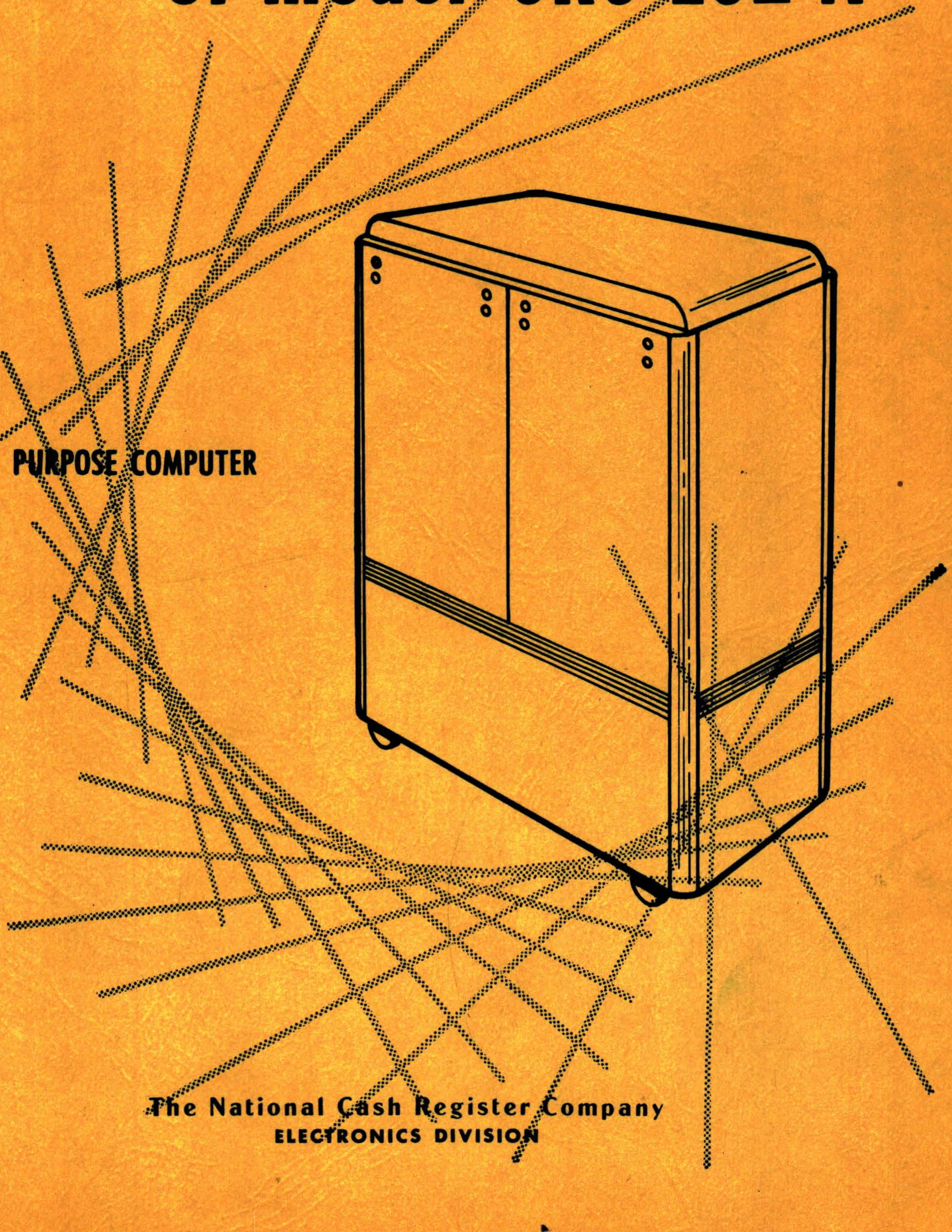
functional specifications of model CRC 102-A

GENERAL PURPOSE COMPUTER



The National Cash Register Company
ELECTRONICS DIVISION

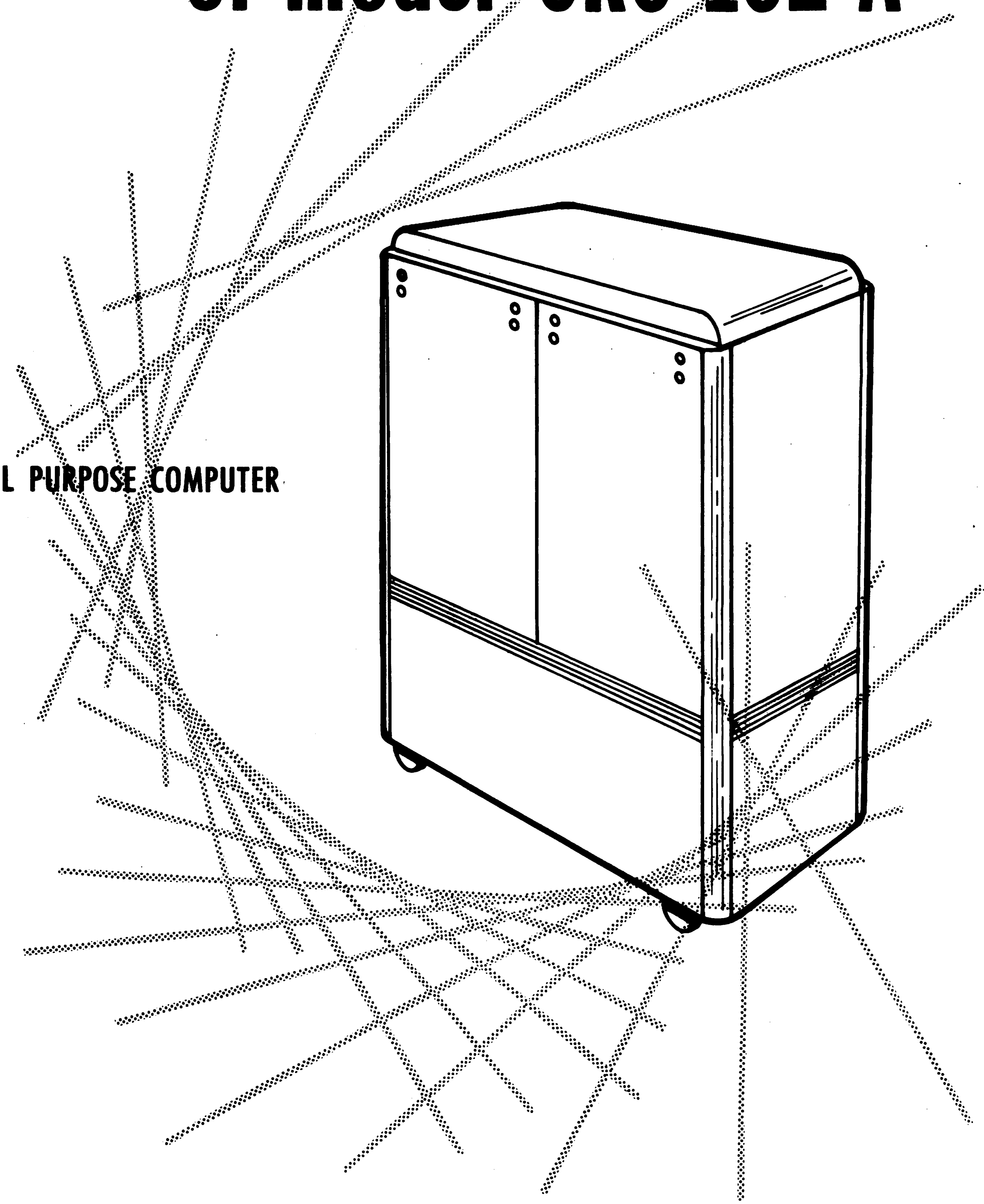
FUNCTIONAL SPECIFICATIONS

OF

MODEL CRC 102-A

functional specifications of model CRC 102-A

GENERAL PURPOSE COMPUTER



The National Cash Register Company

ELECTRONICS DIVISION

3348 WEST EL SEGUNDO BOULEVARD • HAWTHORNE, CALIFORNIA • OSborne 5-1171

THE NATIONAL CASH REGISTER COMPANY
ELECTRONICS DIVISION
FUNCTIONAL SPECIFICATIONS
OF MODEL CRC 102 - A

TABLE OF CONTENTS

Page No.

I
INTRODUCTION

Description of the CRC 102-A System

The CRC 102-A system consists of a computer proper and the associated equipment necessary to provide all of the functions of a general purpose computer. They are specifically: (the reader is referred to the illustration on the frontispiece).

The CRC 102-A Computer

This machine is the outgrowth of the prototype model 102 developed and built by CRC. It is a binary, magnetic drum, serial computer housed in a single cabinet, complete with power supply, all logical elements and air conditioning equipment.

The CRC 102-A Console

This is a desk of a conventional design into which has been built the operator's console. This console consists of six push buttons, six toggle switches, and seven indicator lights. A Flexowriter, mounted on the desk, is electrically connected to the computer so that signals from its keyboard may be used to fill and control the computer. The exact functions of all keys, switches, and lights will be explained in Section III.

The CRC 126 Magnetic Tape Unit

The computer may utilize up to ~~eight~~ SEVEN of these tape units which are connected to it through a common bus. Each tape unit contains logical circuitry which enables it to search for information stored on its tape independently of the computer. The arrangement and use of magnetic tapes will be explained in Section VII.

The IBM Machines

The computer is capable of accepting data from, and transmitting data to, IBM cards and requires two specially modified IBM machines to provide this feature. These modifications are provided by IBM upon request when ordering the IBM machines. Two machines are used separately, one to read, and one to punch the cards.

II

REPRESENTATION OF DATA IN THE CRC 102-A

The Definition of a "Word" in the CRC 102-A

The CRC 102-A is a binary computer, that is to say, the computer treats data as binary digits, and groups of binary digits, and is capable of doing the basic arithmetic operations in a binary fashion.

The fundamental Unit of Information in the CRC 102-A is defined as the word, and consists of 42 binary digits. In the CRC 102-A a word may be a binary number, a decimal number, or a binary command.

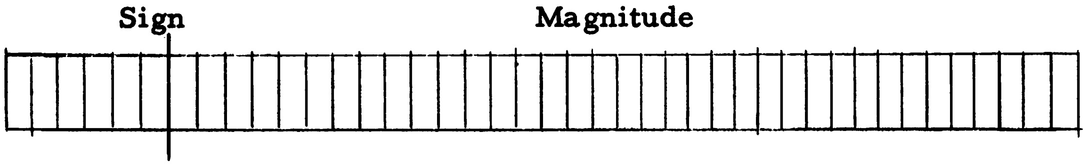
Although the primary number system in the computer is the binary system, the 42 binary digits of the word are grouped into 14 triads (groups of three each) each of which may be represented as an octal digit. The following table shows the equivalence in the octal system of the eight possible binary groups.

TABLE I

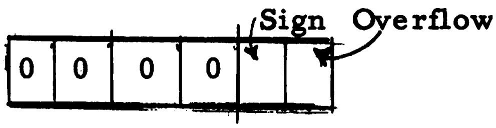
Binary Group	Octal Digit
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Binary Numbers in the CRC 102-A

A binary number in the 102-A is split into two sections. The first six binary digits make up the sign section, and contain the sign and the overflow digits. The remaining 36 digits make up the magnitude section and contain the absolute value of the number.



The first four binary digits in the sign are always zero. The fifth digit of the sign indicates the algebraic sign of the magnitude. When it is zero, the number is positive or absolute; when it is one, the number is negative. The sixth binary digit of the sign is an overflow marker. When it is zero, it indicates no overflow; when it is one, it indicates overflow. It may be filled initially, with the magnitude, as a special marker



to be identified later by the Test for Overflow Marker Command (See Section VI, page 46). Also when the computer performs an arithmetic operation which exceeds the capacity of the arithmetic element, a one is automatically inserted into the overflow position of the result. Thus the following code may be adopted for the sign digits of a binary number.

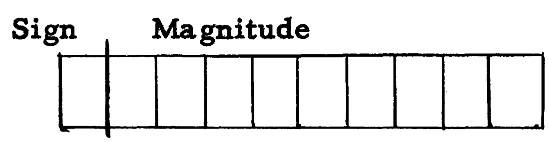
TABLE II

Binary	Octal Equivalent	Interpretation
000 000	00	Positive, No Overflow
000 001	01	Positive, Overflow
000 010	02	Negative, No Overflow
000 011	03	Negative, Overflow

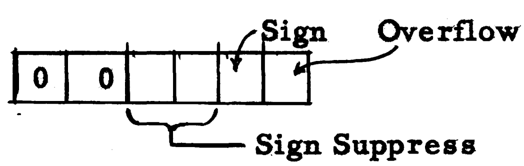
The magnitude consists of 36 binary digits or 12 octal digits. The position of the binary point in the magnitude is essentially arbitrary according to the user's desires. The computer treats the number as if the point were at the split between the sign and the magnitude, i.e., as if the number were scaled to be less than one. If numbers greater than or equal to one are to be used in a computation, scale factors must be introduced and care must be exercised to keep track of these throughout the program.

Decimal Numbers in the CRC 102-A

Numerical data is usually represented in the decimal number system. Decimal numbers may be represented directly in the 102-A and may be printed out directly. They may not be used for normal computer operation except when converting. They may be converted to binary numbers and binary numbers may be converted to decimal numbers automatically within the computer with a conversion program. Each decimal number in the 102-A consists of a sign and nine decimal digits.



The decimal sign consists of six binary digits. The four least significant of these are interpreted as the sign proper.



The sign digits may be interpreted binary digit by binary digit the same as the binary sign digits (See Section II, page 4). The decimal code is represented outside the machine according to the following table:

TABLE III

Decimal Filling	Sign Digit Printing	Binary Equivalent	Interpretation
+, 0	+	00 0000	Positive, no overflow
-, 2	-	00 0010	Negative, no overflow
1	p	00 0001	Positive, overflow
3	n	00 0011	Negative, overflow
Space Bar	Space	00 1100	Suppress sign printing (Print absolute)

see p 21

Note

It is possible to suppress printing the sign when desired.

The nine decimal digits consist of 36 binary digits. Each decimal digit is represented by a four binary digit code which is automatically entered when filling in the decimal mode. Two special characters are included to facilitate editing output data. The following table indicates the binary code for decimal digits and the special characters.

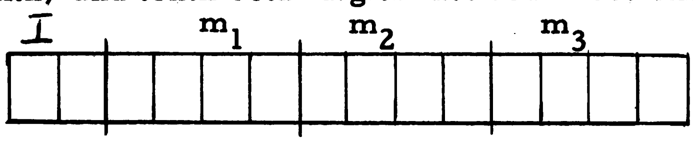
TABLE IV

Decimal Digit	Special Character	Binary Equivalent
0		0000
1		0001
2		0010
3		0011
4		0100
5		0101
6		0110
7		0111
8		1000
9		1001
	Space	1100
	Period	1111

see p 21

Commands in the CRC 102-A

Commands are used in the CRC 102-A in the binary notation, although they are written outside of the computer in the octal notation. Each command consists of 14 octal digits which are usually split after the second, sixth, and tenth octal digits into four sections.



The two instruction digits identify the operation and the remaining three groups of four octal digits each usually refer to cells in the memory. The commands for use with magnetic tape are not of this form and will be discussed in Section VII.

The instruction digits are interpreted according to the code shown in Table V. For exact descriptions of their functions see Sections V through IX.

The 102-A is called a three-address-computer because most commands refer to three separate cells (addresses) during their execution. Each cell is referred to by an address consisting of four octal digits. The three groups of four octal digits are called respectively: m_1 (first reference to the memory), m_2 (second reference to the memory) and m_3 (third reference to the memory). For further details on the memory and the significance of the separate octal and binary digits of each address see Section IV.

Special Codes.

The CRC 102-A is capable of unscrambling data entered in a variety of patterns. All input and output data must be in the same general form as described above, but it is possible for a programmer to establish a convention whereby different digits in the same word would be interpreted separately by the program and the user.

TABLE V

Octal Code	Abbreviation	Name
00-03		Numbers - cause alarm*
04	bo	Buffer Out
05	bl	Buffer Load
06	rc	Read Card
07		Extra - causes alarm*
10		Extra - causes alarm*
11	fl	Fill (from paper tape)
12	pd	Punch Decimal
13	po	Punch Octal
14	bs	Block Search
15	wt	Write (magnetic) Tape
16	rt	Read (magnetic) Tape
17	ts	Test Switch: Test Search
20	pl**	Extra - causes alarm*
21	pr	Print (on Flexowriter)
22	ht	Halt (Computation)
23	dr	Divide and Round-Off
24	dd	Divide and Save Remainder
25	mr	Multiply and Round-Off
26	md	Multiply Double Length
27	sl	Shift Logically
30	sm	Shift Magnitude
31	sf	Scale Factor
32	ex	Extract
33	ta	Test Algebraically
34	tm	Test Magnitude
35	ad	Add
36	su	Subtract
37	to	Test for Overflow Marker

* When used as an instruction.
** Plot command is provided when an automatic digital point plotter is used with the 102-A Computer.

For example: Because conversions are usually mandatory when dealing with decimal numbers, it becomes feasible to put numerical information in the sign digits for convenience of input. The conversion program can then separate out this digit. The same holds true for octal numbers. BUT NO NUMBER OVER 3 CAN GO IN O_{13} ($P_2 O_{13}$ always zero in the computer)

When a complicated problem requires that the scale factor be carried with a number, it is customary to assign the last seven or eight binary digits of a word to the scale factor. The two sections are separated and manipulated separately by the program.

Certain varieties of logical information are best coded by the existence or lack of a binary one in a certain position of a word. This type of information must be written in binary and then consolidated to octal before entering it into the computer. Of course, it could be assembled by a program if desired.

The binary number system has been retained in the CRC 102-A computer because it allows the programmer the utmost flexibility in representation of varied data, finer gradations in scaling, powerful logical operations, and greatest convenience in packing.

III

OPERATOR'S CONSOLE; FLEXOWRITER; FILLING AND OPERATING COMPUTER.

Console

The console contains all the control necessary to use the computer. It consists of a desk, on which is mounted a small control panel, and a modified Flexowriter.

The control panel is arranged as in Figure I. There are six pushbuttons, seven toggle switches, and seven indicator lights. They perform the following functions:

Power Control

1) No. 0, a toggle switch, performs two functions. When it is in the "ON" position, the computer cannot be turned on or off. The "ON" and "OFF" pushbuttons are made inoperable as far as the computer is concerned. The magnetic tape units however are not affected by this switch and may be turned on or off as described in Section VII, page 51. When this toggle switch is in the "OFF" position, it automatically activates the "CLEAR" pushbutton and allows the computer to be turned on or off without impairment of the information on the magnetic drum.

2) No. 1, a pushbutton marked "ON", turns on all power in the computer and in all magnetic tape units connected to the computer which are switched to "AUTO" (See Section VII, page 51). If the No. 0 toggle switch is in the "ON" position, this pushbutton is inoperable except for the power to the tape units. Before turning on the computer, the No. 0 toggle switch must be set to the "OFF" position.

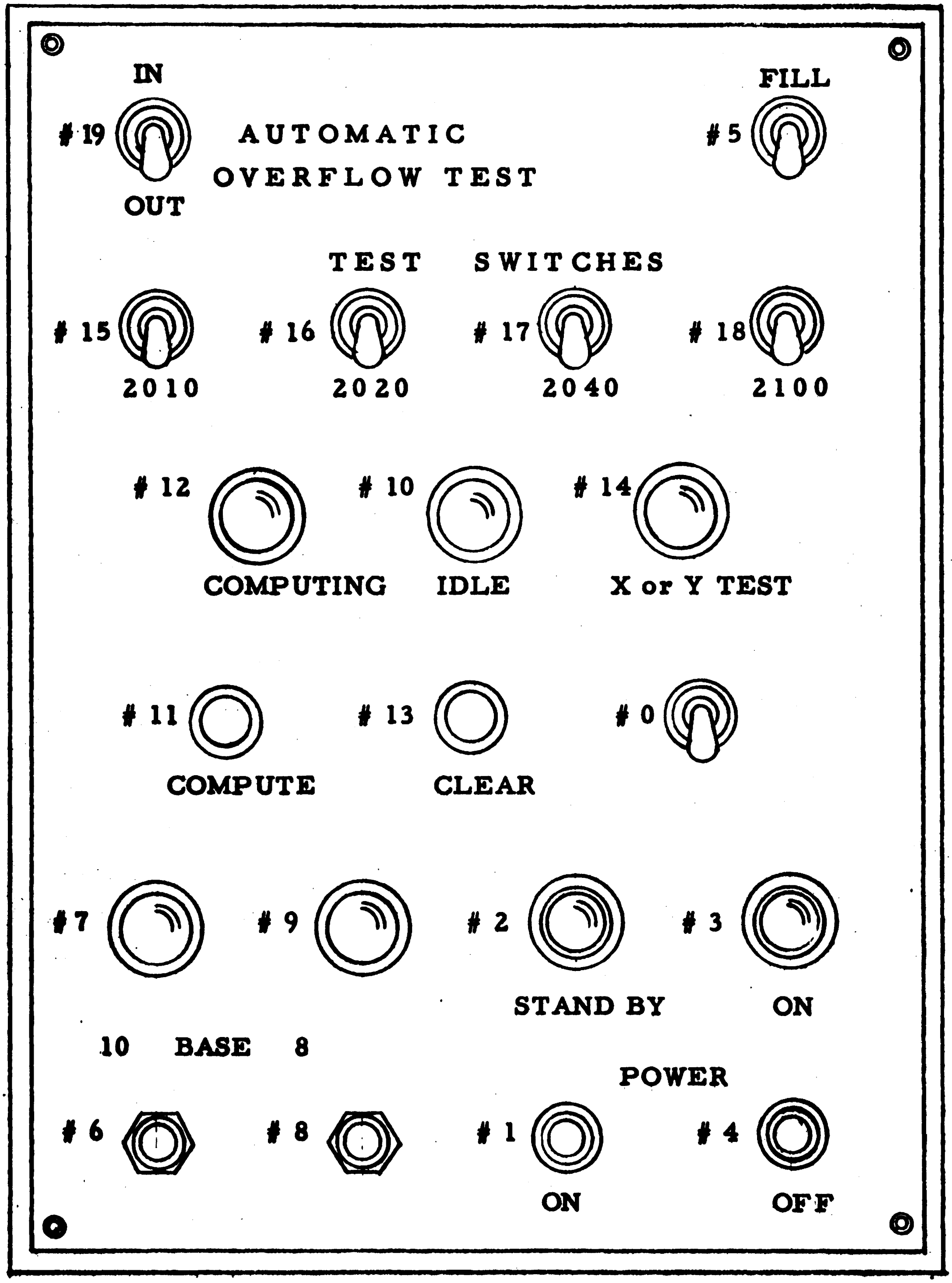


Figure I. Computer Control Panel

Flexowriter and Computer Control Panel

3) No. 2, an indicator light marked "STAND BY," indicates that the computer is warming up.

4) No. 3, an indicator light marked "ON," indicates that the computer is completely on and ready for operation.

5) No. 4, a pushbutton marked "OFF," turns off all power in the computer and all magnetic tape units connected to the computer which are switched to "AUTO". If the No. 0 toggle switch is in the "ON" position, the pushbutton is inoperable except for the power to the tape units. Before turning off the computer, the No. 0 toggle switch must be set to the "OFF" position.

Filling Control

1) No. 5, a toggle switch, connects the Flexowriter to the computer electrically when in the "FILL" position. When not in the "FILL" position, the information typed on the Flexowriter will not enter the computer, but the computer may still type output results on the Flexowriter.

2) No. 6, a pushbutton marked "10," prepares the computer to accept decimal numbers. The "d" key on the Flexowriter also accomplishes this. The "d" key and pushbutton No. 6 differ in their operation in that the "d" key also clears the Input Register and pushbutton No. 6 does not.

3) No. 7, a indicator light, is associated with the No. 6 pushbutton and the "d" key on the Flexowriter. It indicates that the computer is prepared to accept decimal numbers.

4) No. 8, a pushbutton marked "8," prepared the computer to accept octal numbers or commands. The "o" key on the Flexowriter also accomplishes this. The "o" key and pushbutton No. 8 differ in their operation in that the "o" key also clears the Input Register

and pushbutton No. 8 does not.

5) No. 9, an indicator light, is associated with the No. 8 pushbutton and the "o" key on the Flexowriter. It indicates that the computer is prepared to accept octal numbers or commands.

Computer Control

1) No. 10, an indicator light, marked "IDLE," indicates that the computer is ready for operation but is not computing.

2) No. 11, a pushbutton marked "COMPUTE," starts the computation. The "s" key on the Flexowriter also accomplishes this.

3) No. 12, an indicator light marked "COMPUTING," indicates that computation is in progress.

4) No. 13, a pushbutton marked "CLEAR", stops the computation and prepares the computer to accept octal information. This button may not be used as an operator halt; computation may not be restarted where it was stopped. The function of this button is to synchronize the control section of the machine with the memory when the machine is first turned on. When the No. 0 toggle switch is in the "OFF" position, this is done automatically. If pressed during the middle of a computation, it will immediately halt computation, possibly during the execution of a command, and place the computer in the IDLE condition ready to accept octal input. It does not clear data out of any portion of the Main Memory, the Buffer Register, or Control Register.

5) No. 14, an indicator light marked "X" or "Y TEST," indicates that some switch on the engineering test control panel is thrown and must be reset before the computer will operate properly.

Test Switches

1) Switches No. 15 through No. 18, are used in conjunction with the Test Switch command (See Section IV, page 32 ; Section VI, page 48 .)

Suppression of Overflow Checking Switch

1) Switch No. 19, marked "AUTOMATIC OVERFLOW TEST," controls the overflow checking feature. (See Section V, pages 38 , 39 ;Section VI, page 46 .) When this switch is in the "OUT" position, the overflow checking feature is not in operation.

Flexowriter

The Flexowriter is the primary input-output device for the computer. The Flexowriter Keyboard appears as in Figure II. The following keys are significant to the computer when toggle switch No. 5 on the control panel is in the "FILL" position, and the computer is in the "IDLE" condition as indicated by light No. 10.

Control Keys

1) The letter "d" prepares the computer to accept decimal numbers, and clears the Input Register.

2) The letter "o" prepares the computer to accept octal numbers and commands and clears the Input Register.

3) The letter "s" starts computation.

4) "Tab" key transfers a word from the Input Register to a pre-selected memory cell.

5) (-), the hyphen key (on the lowest row of keyboard), transfers a word from the Input Register to the Control Register.

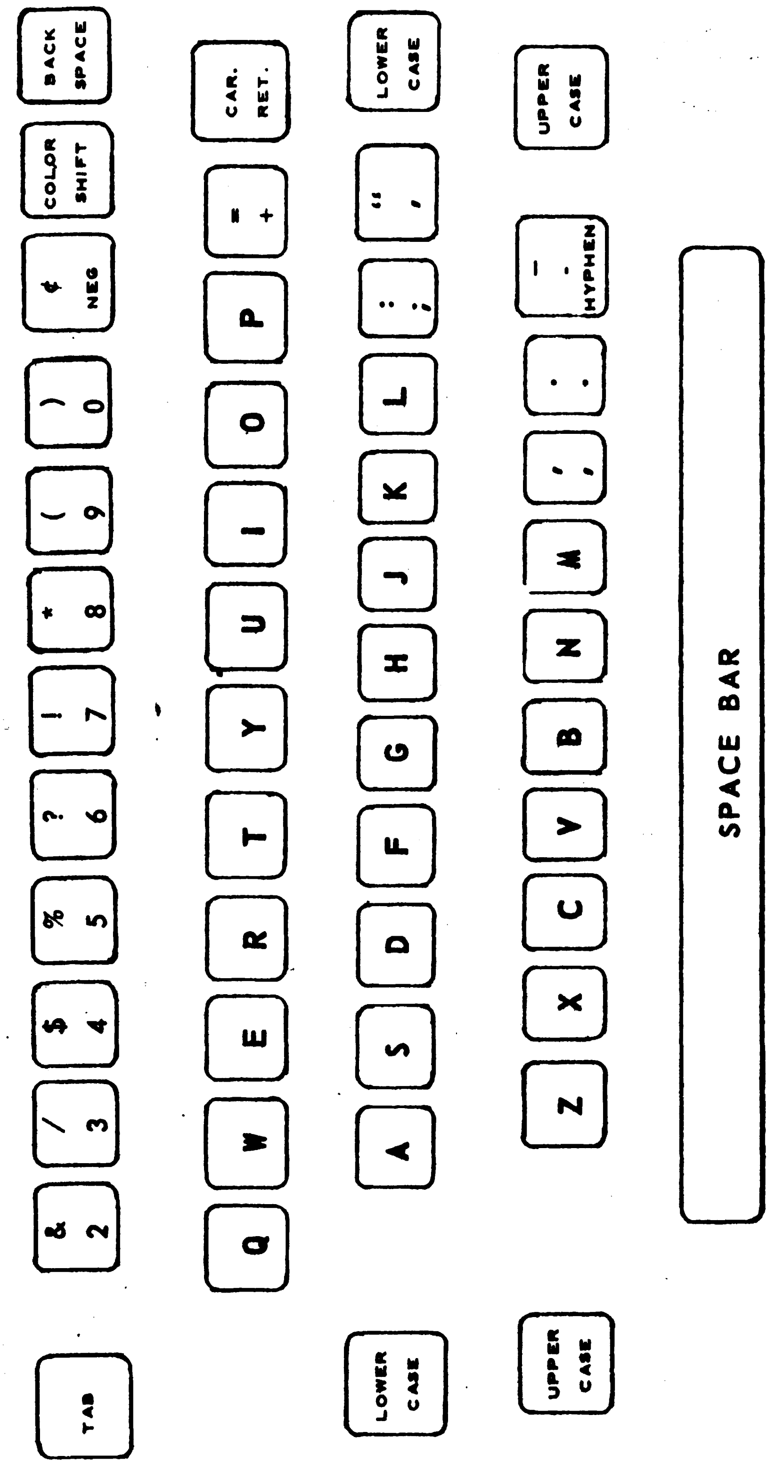


Figure 1 Flexowriter Keyboard

Input Keys for octal numbers

These keys are effective for octal numbers when the computer is prepared to accept octal numbers or commands as indicated by the No. 9 indicator light. Each numerical key fills three binary digits according to the Table VI.

TABLE VI

KEY	BINARY CONFIGURATION
0 (Digit 0 in top row)	000
1 (Letter L, lower case)	001
2	010
3	011
4	100
5	101
6	110
7	111
f (Special character - fills four octal zeros, twelve binary zeros).	

Input Keys for decimal numbers

These keys are effective for decimal numbers when the computer is prepared to accept decimal numbers as indicated by the No. 7 indicator light. Each numerical key fills four binary digits according to the Table VII.

FILL
DECIMAL MODE

TABLE VII

KEY	BINARY CONFIGURATION
0 (Digit 0 in top row)	0000
1 (Letter L, lower case)	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
Space bar	1100
Period	1111
+	0000
- (Minus distinguished from hyphen)	0010
f (Special character - fills four decimal zeros.)	

see p vi

Care must be exercised when using the "f" key. It will fill four octal or decimal zeros only if the three most significant octal or decimal digits of the word are zero. Basically, this is because striking the "f" key inserts a zero as the zero key would, and then circularly shifts the register four octal or decimal places to the left. If a filling error is made, the Input Register cannot generally be cleared by repeatedly striking the "f" key, but it can be cleared by striking the "d" or "o" key, depending on the filling mode in use. ~~For a more detailed explanation of the operation of the "f" key, see the Appendix II.~~

On the side of the Flexowriter is mounted a paper tape reader which may be used to read data automatically into the computer. Tapes may be prepared for this purpose using all of the keys listed previously.

For convenience in input and output the Flexowriter is equipped with a special tab stop on the rear of the carriage. This stop can be adjusted manually to cause the carriage to return at any desired point across the line.

Flexowriter Code in the CRC 102-A

The CRC 102-A can activate any key on the Flexowriter when printing out in the alphabetic mode (See Section IX, page 81). Each key is referred to by two octal digits. Data coded for this type of printing must be entered as octal numbers; the operator must convert the letter to its code according to the following table and enter the octal digits. The CRC 102-A will not accept alphabetic data directly. Each word of alphabetic data for typing consists of six pairs of octal digits with the two leftmost digits not used.

0	0										

ALPHABETIC MODE

TABLE VIII

Character on Key Upper	Lower	Octal Code	Character on Key Upper	Lower	Octal Code
A	a	41	Z	z	55
B	b	57	&	2	00
C	c	62	/	3	01
D	d	47	$	4	04
E	e	45	%	5	07
F	f	46	?	6	06
G	g	73	!	7	03
H	h	74	*	8	05
I	i	60	(	9	53
J	j	43	)	0	52
K	k	42	¢	Neg.	22
L	l	71	=	+	23
M	m	76	"	'	20
N	n	66	,	,	26
O	o	77	.	.	27
P	p	70	_	- (hyphen)	21
Q	q	50	:	;	24
R	r	63	shift down		32
S	s	44	shift up		36
T	t	75	carr. return		34
U	u	40	back space		15
V	v	72	space bar		64
W	w	51	color shift		67
X	x	56	tab		31
Y	y	54	code delete		12

Filling the Computer

The following instructions for filling the computer apply to both paper tape and manual filling. A paper tape must be provided with all of the symbols which a human operator would use to enter the data. There are two filling modes, octal and decimal, and it is very important that the proper mode be selected at the proper time.

There are two registers which are used in the filling operation. Both of them are one word delay lines.

MODEL CRC 102-A FUNCTIONAL SPECIFICATIONS

DECIMAL MODE

Config	Key for Fill	Print Sign	Print Nmbr
0000	0,+	+	0
0001	1	p	1
0010	2,neg	-	2
0011	3	n	3
0100	4	4	4
0101	5	8	5
0110	6	6	6
0111	7	5	7
1000	8	---	8
1001	9	---	9
1010	---	---	--- Code Delete?
1011	---	---	---
1100	space	space	space
1101	---	back	back
1110	----	----	----
1111	period	---	period

OCTAL MODE

Config	Key for Fill	Print
000	0,+,8	0
001	1,9	1
010	2,neg	2
011	3	3
100	4,space	4
101	5,[illegible]	5
110	6	6
111	7,period	7

PRINT CONTROL

No Address	Address	
00,~~04~~ 20,~~24~~	02,~~06~~ 22,~~26~~	Octal
01,~~05~~ 21,~~25~~	03,~~07~~ 23,~~27~~	Decimal
10,11 ~~14,15~~ 30,31 ~~34,35~~	12,13 ~~16,17~~ 32,33 ~~36,37~~	Alphabetic

NUMBERS CROSSED OUT
PRINT ONE WORD ONLY, THEN HALT
NECESSARY TO CLEAR BEFORE PROCEEDING

Consider filling the octal number - 314200000000

a. First input register is clear -

0	0	0	0	0	0	0	0	0	0	0	0	0	0

b. Strike (2) key (for negative, no overflow)

0	0	0	0	0	0	0	0	0	0	0	0	0	2

c. Strike (3) key -

0	0	0	0	0	0	0	0	0	0	0	0	2	3

d. Strike (1) key -

0	0	0	0	0	0	0	0	0	0	0	2	3	1

e. Strike (4) key -

0	0	0	0	0	0	0	0	0	0	2	3	1	4

f. Strike (2) key -

0	0	0	0	0	0	0	0	0	2	3	1	4	2

g. Strike (f) key (enter four zeros)

0	0	0	0	0	2	3	1	4	2	0	0	0	0

h. Strike (f) key (enter four zeros)

0	2	3	1	4	2	0	0	0	0	0	0	0	0

The control register controls the dispatching of the assembled word to its proper destination cell and contains the address of the next command in order. It has the same structure as an octal command. The m_2 position is the address of the next command in order and the m_3 position is the destination of the next word to be dispatched. For example, if the control register held

0	0	0	0	0	0	1	6	4	2	0	7	3	5

the next piece of data dispatched would go to cell 735, or if computation were started the computer would take its first command from cell 1642.

The contents of the input register are transferred to the control register by hitting the hyphen (-) key on the Flexowriter.

The filling process proceeds as follows:

1. Select the octal number system in order to fill control register with address of first cell to be filled. This may be verified by observing indicator light No. 9.

   a. "CLEAR" button sets computer to fill in octal.

   b. If computer has been computing and has been stopped either by a "HALT" or "FILL" command it is set to fill in octal.

   c. If decimal data has just been filled, then it is necessary to press the "o" key or push button No. 8 on the console which will always set the computer to fill octal.

2. Select the address of first cell to be filled.

   a. If ~~computer~~ CONTROL REGISTER has been cleared, input will automatically start at cell 0000 if control register is not modified.

   b. If the computer has been computing and stops for a "HALT" or "FILL" command, the m_3 position of the "HALT" or "FILL" command will be the cell where input will start if the control register is not modified.

c. To specify a particular first address to the control register, the desired octal address, followed by a hyphen, is typed on the Flexowriter.

3. Select the number system necessary for the data to be entered. Once a number system is selected, it is preserved until changed. The indicator lights, No. 7 and No. 9, on the console indicate the selected mode.

a. Commands and octal numbers must be entered in the octal mode. If the octal system is not selected, hit the "o" key or push button No. 8 on the console.

b. Decimal data can be entered when decimal mode is selected. This is done by hitting the "d" key or push-button No. 6 on the console.

4. Enter data.

a. Commands - octal system.

1) Enter the two digits identifying the command.

2) Enter the three addresses; where one of the addresses is not significant to the command, it may be filled with any desired value. Generally four zeros will be most convenient, which may be entered with the "f" key. For minimum access coding enter 2100 or 3000.

b. Octal numbers.

1) Enter octal sign.

If the sign is positive without overflow, it need not be entered. However, if any of the other signs are intended, the right-hand sign digit must be entered.

TABLE IX

Desired Sign	Filling Operation
positive, no overflow	- enter nothing or a "0"
negative, no overflow	- enter a "2"
positive, overflowed	- enter a "1"
negative, overflowed	- enter a "3"

2) Enter octal magnitude.

If the sign was positive without overflow, and was not entered, zeros on the left which would have been needed to fill out the 12 octal digits of the word may be ignored, however, if any sign is entered, all 12 digits of magnitude must be entered.

c. Decimal numbers.

1) Enter decimal sign.

If the sign is positive without overflow, it need not be entered. If any of the other signs are intended, the sign digit must be entered.

TABLE X

Desired Sign	Filling Operation
positive, no overflow	- enter nothing, "+" or "0" (zero)
negative, no overflow	- enter (minus) "-" or "2"
positive overflowed	- enter "1"
negative overflowed	- enter "3"

2) Enter decimal magnitude.

If the sign was positive without overflow and was not entered, zeros on the left which would have been needed to fill out the 9 decimal digits of magnitude may be ignored. However, if any sign was entered, all 9 digits of magnitude must be entered.

When the word has been typed in the input register, it can be transferred to the previously selected destination cell by hitting the "tab" key. This moves the typewriter carriage, dispatches the word to its destination, indexes the control register to prepare to dispatch the next word to the next consecutively numbered memory cell, and clears the input register.

5. In the case of an error.
   a. An error before the "Tab" button is hit can be corrected by entering the entire correct command or number, always including both sign digits and all leftmost zeros, and then tabbing.

   b. An error detected after tabbing can be corrected by selecting the octal mode (if not selected), setting up the control register (as in Section III, page 23) to the address of the cell where the incorrect data was entered, and re-entering the corrected data.

6. In the case of a gap in consecutiveness of the data, the first register of the next group of data is selected by setting up the control register (as in Section III, page 23).

Operating Computer

1. Turn No. 0 toggle switch to the "OFF" position.
2. Turn computer on with push button No. 1 and wait until indicator light No. 3 comes on.
3. Turn No. 0 toggle switch to the "ON" position.
4. Clear computer with push button No. 13.
5. Enter commands and data according to instructions above.

6. Select address of first instruction.

   a. Prepare computer to accept octal numbers as in Section III, page 23).

   b. If computer has been cleared, computation will start with cell 0000 if the control register is not modified.

   c. If the computer has been computing and stops from a "HALT" or "FILL" command, and the control register is not modified, computation will restart at command in cell following the "HALT" or "FILL" command.

   d. To specify a particular starting address to the control register, the desired octal address followed by an "f", followed by a hyphen (-), is typed on the Flexowriter.

7. Start computation by hitting "s" key on typewriter or with push button No. 11 marked "COMPUTE".

8. Program control will bring computer to a stop after due process of computation.

9. Turn No. 0 toggle switch to "OFF" position.

10. Turn off computer with push button No. 4.

Further operating instructions for the entire system are included in sections on Test Switch Command, Magnetic Tape Unit Description, IBM Description, and Flexowriter Operations.

IV

MEMORY

The 102-A has a magnetic drum memory. The drum rotates on a vertical axis at approximately 40 rps and yields a basic pulse rate of approximately 100 kc. The surface on which data is recorded is referenced in two dimensions: on the vertical the drum is divided into channels; on the periphery it is divided into sixty-four sectors. These sectors are identified by a special channel called the "Word Channel." This channel is numbered with the octal numbers 00 through 77 in one of several possible patterns and serves to select the time when any desired sector comes under a reading and writing head. Since the word channel controls the time selection of data and commands which is very important in minimum access coding, provision is made to renumber it in special patterns which facilitate minimum access programming.

Main Memory

The Main Memory consists of sixteen complete channels. The cells of the main memory are referred to by addresses 0000 through 1777 (octal numbers). During computer operation, the main memory stores the program and the greater part of the data being worked on. There is no restriction on the respective positions of commands and data. The main memory can be filled directly either manually or from paper tape.

Buffer Register

The Buffer Register is an eight word delay line synchronized with the least significant digits of the word channel. It is customarily referred to by the addresses 2000 through 2007, but other addresses may also refer to it; see the table at the end of this section for their structure.

The buffer register is so called because it is a buffer between the magnetic tape or the IBM cards and the main memory. It is possible to transfer data in eight word blocks back and forth between the main memory and the buffer register. Data or programs on magnetic tape or IBM cards are read into the buffer and then transferred to the main memory. To record on magnetic tape or to punch cards, the buffer register is first loaded from the main memory and then the contents of the buffer register are recorded on the external unit. The buffer register may also be filled manually or from paper tape.

The buffer register may also be used as an extension of the main memory. Because of the nature of the buffer as a delay line, the access time to any cell in it is a maximum of one-eighth of a drum revolution as against one drum revolution for the main memory.

Cell 2100

This is a special electronic feature which provides the programmer with a positive zero. It is not a particular position on the drum but rather it refers to a switching configuration which puts nothing into an arithmetic register. For this reason, cell 2100 is always perfectly minimum access coded. It may not, for the same reason, be used as an address when recording into the memory. Other addresses may also refer to cell 2100; see the Table XI at the end of this section for their structure.

Cell 3000

This cell is provided as a convenience for use with relative programming techniques. It may be referred to in the normal course of computation by all commands except the PRINT command. The contents of cell 3000 has the same structure as an octal command and depends directly upon the command preceding the command which refers to it.

1. If the command preceding the command which refers to cell 3000 is (see Section II, page 7 , Table V.) bo, bl, rc, pd, po, bs, dr, dd, [redacted], [redacted], sl, sm, sf, ex, or a test command to, ta, tm, or ts, <u>when the test does not work</u> then cell 3000 will contain:

   a. In the two octal instruction digits, the octal code 00.
   except if cmd was mr or md – then I contains 37
   b. In m_1, the m_1 portion of the previously obeyed command.
   c. In m_2, the address of the command which follows the command which is referring to cell 3000.
   d. In m_3, the m_3 portion of the previously obeyed command.

2. If the command which refers to cell 3000 was not counted into from the previous consecutive command, but was skipped into from a test command which worked, either to, ta, tm, or ts, then cell 3000 will contain:

   a. In the two octal instruction digits, the two least significant octal digits of the m_1 portion of the previous command.
   b. In m_1, the address of the command following the test command which skipped to the command which refers to cell 3000.
   c. In m_2, the address of the command which follows the command which is referring to cell 3000.
   d. In m_3, 000y   y is MSD of m_1 of the Test Command which worked

3. If the command which refers to cell 3000 is the starting command immediately after filling, and assuming that the computer has halted with a "FILL" or "HALT" command and that during the filling operation the hyphen (-) key was not struck, or read from the tape, then cell 3000 will contain:

   a. In the two octal instruction digits, the octal code 00.
   b. In m_1, the m_1 portion of the FILL or HALT command.

c. In m_2, the address of the command which follows the command which is referring to cell 3000.

d. In m_3, the address of the cell following the last cell which was filled.

4. If the command which refers to cell 3000 is the starting command immediately after filling and the hyphen (-) key was struck or the hyphen (-) was read from tape during the filling operation, then cell 3000 will contain:

   a. In the two octal instruction digits, the two digits entered into the sign digits just before the last hyphen was struck on the keyboard or read from the tape.

   b. In m_1, the m_1 portion of the word entered just before the last hyphen was struck or read from the tape.

   c. In m_2, the address of the command which follows the command which is referring to cell 3000.

   d. In m_3, the address of the cell following the last cell which was filled unless the hyphen key was struck after the last cell was filled. In this case m_3 will contain the m_3 portion of the word entered just before the last hyphen was struck or read from the tape.

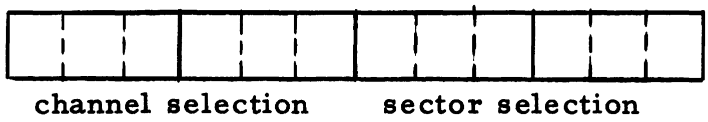
5. When any of the three commands rt, wt, or pr, precede a reference to cell 3000, the contents of cell 3000 is determinable in a fashion distinct from any other command. The information in it is derived directly from the command and does not provide the programmer with useful data. Therefore its forms are not included in this paper.

Test Switches

Test switches are used solely with the Test Switch command. They are numbered on the console as 2010, 2020, 2040, and 2100. They have no connection with any other command and these addresses have different meanings when referred to by any other command. Associated with the Test Switch command is another address 2200. This refers to the magnetic tape units, see the Test Search command for the significance of this.

Interpretation of Address

Reference to the memory is distinct in the two operations of reading and writing. Cells 2100 and 3000 may be read from but not written into, and the switching operation must make this differentiation. All addresses consist of four octal digits, or as the machine interprets it, twelve binary digits.



The two left octal digits select the channel and the two right octal digits select the sector.

Channel selection is accomplished by inspection of the six binary digits and their combinations. In this inspection the difference between reading and writing is significant. The following two tables define the selection process for writing and reading respectively.

Sector selection is accomplished by comparing the sector digits with the word channel. When the sector digits are exactly equal to the word channel, the next cell under the proper read-write head is the cell with the desired address. This comparison takes place across all six binary digits for reference to the main memory. For reference to the buffer register the comparison only takes place across the three right-hand binary digits. These three binary digits are the least significant octal digits of the address.

TABLE XI

Writing into Memory. Legend ≡ no significance; X significant for specific channel; 0 must be 0; 1 must be 1.

Binary Digits (L_5 L_4 L_3 L_2 L_1)	Logical Description	Octal Digits for Channel Selection	Function
≡ 0 X X X X	L_5 is 0	00 - 17 or 40 - 57	Select Main Memory Channel
≡ 1 ≡ ≡ ≡ ≡	L_5 is 1	20 - 37 or 60 - 77	Select buffer register

TABLE XII

Reading from Memory.

Binary Digits (L_5 L_4 L_3 L_2 L_1)	Logical Description	Octal Digits for Channel Selection	Function
≡ 0 X X X X	L_5 is 0	00 - 17 or 40 - 57	Select Main Memory Channel
≡ 1 0 ≡ ≡ 0	L_5 is 1 L_4 is 0 L_1 is 0	20, 22, 24 26, or 60, 62, 64, 66	Select Buffer Register
≡ 1 0 ≡ ≡ 1	L_5 is 1 L_4 is 0 L_1 is 1	21, 23, 25 27, or 61, 63, 65, 67	Select Cell 2100
≡ 1 1 ≡ ≡ ≡	L_5 is 1 L_4 is 1	30 - 37 or 70 - 77	Select Cell 3000

The selection process and the length of the buffer register imply certain restrictions in the numbering of the word channel. The least significant octal digits must be arranged so that they repeat every eight sectors. The arrangement of the most significant octal digits is arbitrary. Certain arrangements of the word channel have been considered for their facility in minimum access coding and are discussed in the ~~appendix~~. SECTION VIII.

V

ARITHMETIC COMMANDS

The arithmetic commands are used to perform the bulk of any computation, to modify commands, and for all subtle operations which the programmer might assign to them. In general, most functions of the computer involve the combined use of many commands, therefore, it is recommended that the reader acquaint himself with all of the commands before working through the examples.

In certain functions commands are operated on as if they were numbers. In such cases the two octal digits which identify the command are treated as sign digits, except in the ADD and SUBTRACT commands which are discussed below. The following information is given for each command.

1. Name and octal code digits.
2. Abbreviation and address structure.
3. Equivalent algebraic and overflow sign when command is used as a number in all but the ADD and SUBTRACT commands.
4. Minimum execution time. The final execution time would depend on the location of the commands on the surface of the drum which in turn depends on the word channel. The quoted figure assumes optimum placement of the operands.
5. Description of the function of the command.
6. Examples where required.

A.

1. ADD 35

2. ad m_1 m_2 m_3

3. Positive, overflow.

4. 7.4 or 8.2 milliseconds, depending on the signs of the operands.

5. (a) If (m_1) is an octal number, add (m_1) to (m_2) with due regard to the algebraic sign of each, and record the sum in cell m_3. If the magnitude of the sum requires more than thirty-six binary digits to be represented in the machine, record an overflow digit in the overflow position of the sign (see example). When an overflow digit is so generated, the computer will cause an overflow alarm (See Section VI, page 46) unless the next command is TEST FOR OVERFLOW MARKER or SHIFT LOGICALLY. (This feature may be suppressed by placing the console switch No. 9 marked AUTOMATIC OVERFLOW TEST in the "OUT" position.) If an overflow marker existed in the (m_1) before the add, an overflow marker will record in m_3, but this will not cause the overflow alarm unless the sum of the (m_1) and (m_2) exceeds thirty-six binary digits. An overflow marker in (m_2) has no effect on the sum.

   (b) If (m_1) is a command, add the absolute value of (m_1) to the algebraic value of (m_2) and record the sum in m_3. The two octal digits identifying the command in (m_1) are retained in m_3. Any overflow generated in taking the sum will not cause the overflow alarm, nor affect the overflow position of the sum.

6. Examples: (Decimal Numbers used in numerical examples.)

(m_1) +500000000	(m_1) p314159265	(m_1) +271828183
(m_2) +700000000	(m_2) +271828183	(m_2) p314159265
(m_3) p200000000	(m_3) p585987448	(m_3) +585987448
Overflow generated	No overflow generated	

(Octal Commands)

(m_1) 36036402771630	(m_1) 00000100100054	(m_1) 36036402771630
(m_2) 00000100100054	(m_2) 36026402771630	(m_2) 00741375006160
(m_3) 36036503071704	(m_3) 02026302671554	(m_3) 36000000000010
		No overflow generated

B.

1. Subtract 36
2. su m_1 m_2 m_3
3. Negative, no overflow
4. 7.4 or 8.2 milliseconds, depending on the signs of the operands.
5. The SUBTRACT operation in the 102-A is exactly like the ADD operation with the exception that the algebraic sign of (m_2) is inverted before the addition takes place. See the ADD command above.

C.

1. Multiply and Round-Off 25
2. mr m_1 m_2 m_3
3. Positive, overflow.
4. 25.0 milliseconds.
5. Multiply the algebraic values of (m_1) and (m_2) as if the binary points were at the left end of the magnitude, i.e., as if (m_1) and (m_2) were both binary fractions, and record the most significant

half of the product rounded off to thirty-six binary digits (twelve octal digits) with the proper algebraic sign in cell m_3. A product is always a positive or negative number without overflow. Any overflow marker in (m_1) or (m_2) has no effect on the product.

D.

1. Multiply Double Length 26
2. md m_1 m_2 m_3
3. Negative, no overflow.
4. 25.0 milliseconds.
5. Multiply the algebraic values of (m_1) and (m_2) and record the least significant thirty-six binary digits (twelve octal digits) of the product with proper algebraic sign in cell m_3 and record the most significant thirty-six binary digits (twelve octal digits) of the product with proper algebraic sign in the next higher cell adjacent to m_3 on the surface of the drum. A product is always a positive or negative number without overflow. Any overflow marker in (m_1) or (m_2) has no effect on either half of the product.

E.

1. Divide and Round-Off 23
2. dr m_1 m_2 m_3
3. Negative, overflow.
4. 25.8 milliseconds.
5. (a) Divide the algebraic value of (m_1) by the algebraic value of (m_2), treating (m_2) as if the binary point were at the left end of the magnitude, i.e., as if (m_2) were a fraction, and record the quotient rounded off to thirty-six binary digits with the proper algebraic sign in m_3. Note that the quotient will always be greater

than the dividend.

(b) The divide command will generate an overflow digit if the absolute dividend is greater than or equal to the absolute divisor. This function has no regard for any arbitrarily assigned scale factors; for this purpose, it may be assumed that the machine considers both divisor and dividend as if the binary point were at the left end of the magnitude. If, in such a case, the absolute quotient is less than two, the quotient recorded in m_3 is correct with the unit appearing in the overflow digit position. If, however, the absolute quotient is equal to or greater than two, the result recorded in m_3 is meaningless. In any case, when an overflow digit is generated, the computer will cause an overflow alarm, (see Section VI, page 46) unless the next command is TEST FOR OVERFLOW or SHIFT LOGICALLY. This feature may be suppressed by placing the console switch No. 9 (see Section III, page 11) marked AUTOMATIC OVERFLOW TEST in the "OUT" position.

F.

1. Divide and Save Remainder 24
2. dd m_1 m_2 m_3
3. Positive, no overflow.
4. 25.8 milliseconds.
5. (a). Divide the algebraic value of (m_1) by the algebraic value of (m_2), treating (m_2) as if the binary point were at the left most end of the magnitude, i.e., as if (m_2) were a fraction, and record the remainder with the same sign as (m_1) in m_3 and the first thirty-six binary digits of the quotient with the proper algebraic sign in the next higher cell adjacent to m_3 on the surface of the drum. (See examples.)

(b) The overflow feature of DIVIDE AND SAVE REMAINDER is the same as for DIVIDE AND ROUND-OFF command.

6. Examples: (Decimal numbers used in example)

(m_1)	-100000000
(m_2)	+300000000
Remainder in m_3	-100000000
Quotient in next higher cell	-333333333
(m_1)	+000000375
(m_2)	+000000200
Remainder in m_3	+000000000
Quotient in next higher cell	p875000000

G.

1. Shift Magnitude 30

2. sm m_1 m_2 m_3

3. Positive, no overflow.

4. 8.6 milliseconds, basic time, plus 0.78 milliseconds for each shift.

5. (a) Shift the magnitude section of (m_1) according to (m_2) and record the shifted value in m_3. Digits shifted off to the left or right end of the magnitude are lost and zeros are shifted in on the opposite end.

   (b) (m_2) is interpreted as follows: If (m_2) is positive, shift left. If (m_2) is negative, shift right. The number of binary shifts is the entire magnitude of (m_2) treated as an octal integer.

6. Examples: Effect of sm m_1 m_2 m_3 on the following numbers.

m_1 (octal command)	26014311632777
m_2 (shift control)	+000000000030
m_3 (result)	26277700000000

H.

1. Shift Logically 27
2. sl m_1 m_2 m_3
3. Negative, overflow.
4. 9.0 milliseconds, basic time plus 0.78 milliseconds for each shift.
5. (a) Shift (m_1) including the sign digits according to (m_2) and record the shifted value in m_3. Digits shifted off the left or right end of the word are lost, and zeros are shifted in on the opposite end. In this command it must be noted that the leftmost binary digit of the word is always a zero and therefore the leftmost octal digit may not exceed three. See example.

   (b) (m_2) is interpreted in SHIFT LOGICALLY exactly as in SHIFT MAGNITUDE.
6. Example: Effect of sl m_1 m_2 m_3 on the following numbers.

   m_1 (octal command) 26014311632777
   m_2 (shift control) +000000000030
   m_3 (result) 23277700000000

   Note that the octal number six has been recorded as octal number two because of the restriction of the binary digit at the left end of the word.

I.

1. Scale Factor 31
2. sf m_1 m_2 m_3
3. Positive, overflow.
4. 10.6 milliseconds basic time plus 0.78 milliseconds for each shift.

5. (a) If (m_2) has anything in the magnitude section, shift only the magnitude left until a binary "one" appears in the most significant binary digit of the magnitude. If (m_2) is zero, do not shift.

   (b) Subtract the number of required shifts from (m_1), treated as an octal integer. If (m_2) is zero, subtract sixty-four (octal 100) from (m_1), treated as an octal integer.

   (c) Record the modified (m_1) in cell m_3 and the shifted results in the next higher cell adjacent to m_3 on the surface of the drum.

6. Example: Effect of sf m_1 m_2 m_3 on the following numbers.

m_1 (octal scale factor	+000000000030
m_2 (octal command)	26014311632777
m_3 (resulting octal scale factor)	+000000000023
Next adjacent cell (resulting command)	26614471537740

VI

LOGICAL AND TRANSFER COMMANDS

A.

1. Extract 32
2. ex m_1 m_2 m_3
3. Negative, no overflow
4. 32.4 milliseconds if m_3 is in main memory; 10.6 milliseconds if m_3 is in buffer register.
5. (a) The extract command copies selected binary digits from one word into the same positions of another word. From (m_1), those binary digits which are in the same positions as the binary "ones" of (m_2) are copied into the corresponding positions of m_3. Where there are binary "zeros" in (m_2) the corresponding positions of (m_3) remain unchanged.

   (b) In executing the EXTRACT command the computer reads (m_3) into an arithmetic register, carries out the extract operation in this register, and then places the results back in cell m_3. Address m_3 is interpreted first as a "look up" address for reading and second as a "put away" address for writing. Section IV, page 33 explains the difference in this interpretation.

   (c) The extract command may be used with cell 2100 in a very convenient fashion. The operation ex A B 210X yields the logical product of A and B in Cell 200X. Refer to Section IV, page 33 on the difference between channel selection for reading and writing.

6. Examples:

Command:	ex	m_1	m_2	m_3
(m_1) =	ad	0107	1523	2007
(m_2) =	00	0000	7777	0000
(m_3) =	mr	2007	0674	2000

Resulting (m_3) = mr 2007 1523 2000

Command: ex m_1 m_2 2103

(m_1) = 10723635214263

(m_2) = 30236241253637

(m_3) = ZERO [redacted]

Result in 2003 = 10222201210223

B.

1. Test for Overflow Marker 37
2. to m_1 3000 m_3
3. Negative, overflow.
4. 6.6 milliseconds when test works.
5. Examine (m_1) as if it were an octal number. If the sign contains a binary "one" in the overflow digit, take the next command from cell m_3; otherwise, proceed normally. A binary "one" may appear in the overflow digit for any one of the following reasons:
   a. It may be filled initially as a word with an overflow marker.
   b. It may be a command noted as having a "one" in the overflow digit.
   c. It may be the result of an ADD, SUBTRACT; DIVIDE, or DIVIDE AND SAVE REMAINDER command which generated an overflow.

d. It may have been placed there by the SHIFT LOGICALLY command.

C.

1. Test Magnitude 34
2. tm m_1 m_2 m_3
3. Positive, no overflow.
4. 7.8 milliseconds when test works.
5. (a) Compare the absolute magnitudes of (m_1) and (m_2). If $|(m_1)| > |(m_2)|$ take the next command from cell m_3; otherwise proceed normally.

   (b) TEST MAGNITUDE can be used for a zero test on cell X with: tm X 2100 Y. If (X) $\neq$ zero, the machine will skip to Y for the next command.

   (c) TEST MAGNITUDE can be used for an unconditional transfer without minimum access coding effort with: tm 3000 2100 Y. Cell 3000 will always contain some value and 2100 is always zero.

D.

1. Test Algebraically 33
2. ta m_1 m_2 m_3
3. Negative, overflow.
4. 8.2 milliseconds when test works.
5. Compare (m_1) and (m_2) algebraically. If $(m_1) > (m_2)$ take the next command from cell m_3; otherwise, proceed normally. The TEST ALGEBRAICALLY command considers positive zero greater than negative zero.

E.

SIGNIFICANT BITS

1. Test Switch or Test Search (See Section VII, page 55) 17

2010	xxx	0x0	001	xxx
2020	xxx	0x0	010	xxx
2040	xxx	0x0	100	xxx
2100	xxx	0x1	000	xxx
2400	xxx	1x0	000	xxx

2. ts m_1 3000 m_3

3. Negative, overflow.

4. 6.3 milliseconds when test works.

5. (a) If m_1 is 2010, 2020, 2040, or 2100 and the respectively numbered switch on the console is in the up position, take the next command from cell m_3; if the switch is down, proceed normally.

   2400

   (b) If m_1 is [redacted] and any Magnetic Tape Unit attached to the computer is block searching (see Section VII, page 55) take the next command from cell m_3, if no Magnetic Tape Unit is searching, proceed normally.

F.

1. Halt 22

2. ht 3000 3000 m_3

3. Negative, no overflow.

4. 9.3 milliseconds (meaningless).

5. a. Stop computation and return machine to idle.
   b. Prepare computer to accept octal.
   c. Leave control register prepared to place the first word entered from the Flexowriter in the normal filling operation into cell m_3.

G.

1. Buffer Out 04

2. bo 3000 3000 m_3

3. Positive, no overflow

4. 10.2 milliseconds

5. Transfer the entire eight words of the buffer register into the eight physically adjacent main memory cells starting with m_3. The word which will be entered into m_3 is the one whose buffer cell has the same least significant digit in its address as the address m_3. The BUFFER OUT command does not clear the Buffer Register; it merely copies its contents into the main memory. The buffer cells are treated in a cyclic fashion, with cell 2000 considered as following cell 2007.

H.

1. Buffer Load 05

2. bl 3000 3000 m_3

3. Positive, Overflow.

4. 9.8 milliseconds

5. Transfer the eight words in the physically adjacent main memory cells starting with m_3 into the buffer register. The word in cell m_3 will be entered into the cell of the buffer register whose address has the same least significant digit as the address m_3. The BUFFER LOAD command replaces the previous contents of the Buffer Register with the contents of eight adjacent main memory cells as described; the previous contents of the Buffer Register are lost. The Buffer Register may be cleared by the command: bl 3000 3000 2100.

VII

MAGNETIC TAPE OPERATION

Magnetic Tape Unit Model CRC 126

The CRC 102-A drum provides sufficient internal storage for the execution of most problems of moderate complexity. However, the drum is limited, and for those applications which require exceptionally long programs, or many programs, or large quantities of tabular data some automatic data storage medium external to the computer is necessary. The CRC 126 Magnetic Tape Unit is adaptable to the CRC 102-A for this purpose. The CRC 102-A is capable of having SEVEN ~~eight~~ such tape handling units connected to it at any one time. Each individual unit handles one tape at a time but tapes may be changed manually with ease.

When more than one tape unit is used at a time, the first unit is plugged into the computer, the second unit is plugged into the first, and so forth. The line on the plugs are common to all of the tape units. The computer does not have separate lines to each; instead three lines from the computer common to all of the tape units are energized in one of eight combinations. (Each line may be either at a high or low voltage potential). Each tape unit recognizes only one configuration by allowing that configuration to energize a relay.

The CRC Model 126 Magnetic Tape Unit has a small control panel with three push buttons, three indicator lights, four toggle switches, two indicator switches, and a meter. See Figure III.

The various controls accomplish the following:

a. Indicator switch No. 1 disconnects the tape unit from the computer for manual service or connects it for automatic computer controlled operation. When the switch is in the position marked

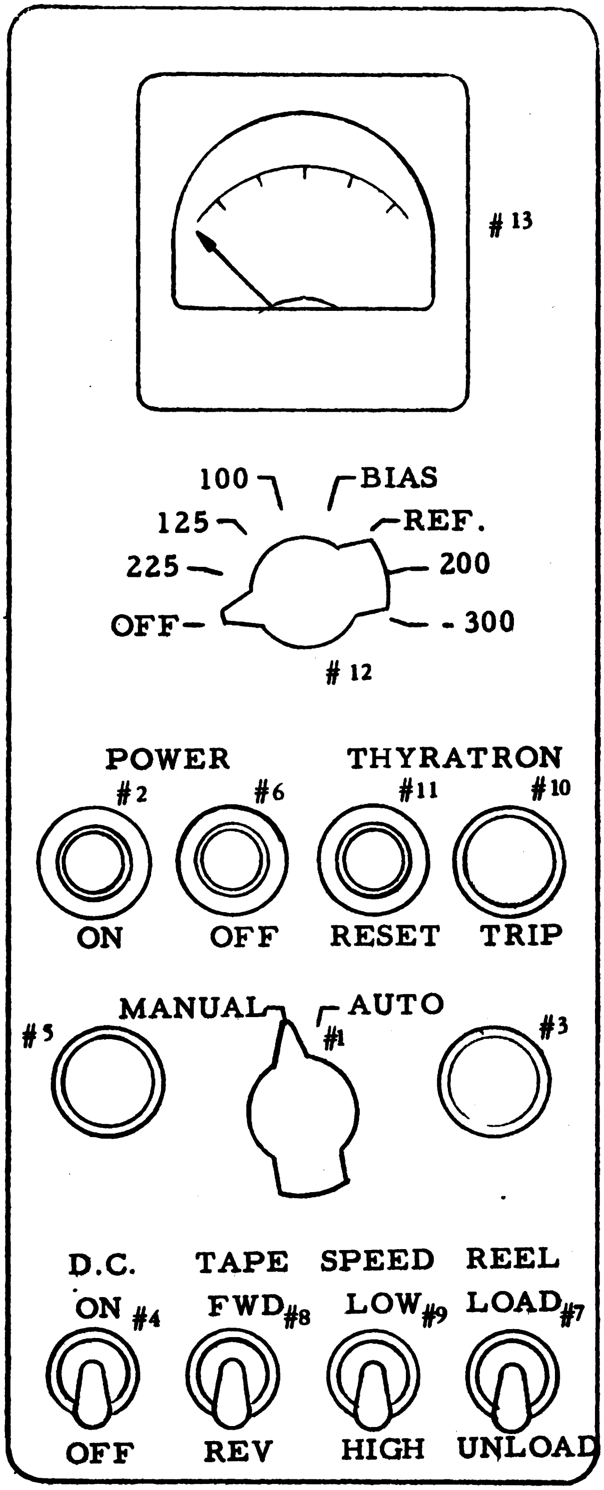


Figure [illegible]. Tape Unit Control Panel

III

"MANUAL", the computer has no effect on the tape unit. When the switch is in the position marked "AUTO" the tape unit is under the control of the computer, but may be turned on and off manually.

b. Power control in both AUTO and MANUAL states.

1) Push button No. 2 marked "ON" turns on the AC power.

2) Indicator light No. 3 indicates AC power is on.

3) Toggle switch No. 4 marked "DC" turns DC power on or off when in positions marked "ON" or "OFF" respectively.

4) Indicator light No. 5 indicates DC power is on.

5) Push button No. 6 turns all power off.

c. Tape Control for manual operation.

1) Toggle switch No. 7 marked "REEL" is a two position switch which controls the rotation of the tape reel. It can be used only when switch No. 1 is on "MANUAL". When switch is in position marked "LOAD" the reel is driven counterclockwise and winds up the tape out of the rear basket. When switch is in position marked "UNLOAD" the reel freewheels. When tape is in normal use, switch should be in "UNLOAD" position.

2) Toggle Switch No. 8 marked "TAPE" is a three-position switch which controls the direction of tape moving through the capstans. It can be used only when switch No. 1 is on "MANUAL". When switch is in the "OFF" position the tape will not move. When switch is in the "FWD" position, the tape will move from left to right. When switch is in the "REV" position, the tape will move from right to left. ?

3) Toggle Switch No. 9 marked "SPEED" is a two position switch which controls the moving speed of the tape. It can be used only when switch No. 1 is on "MANUAL." When switch is in the "LOW" position, the tape moves at approximately fifteen inches per second. When switch is in the "HIGH" position, the tape moves at approximately 90 inches per second.

d. Testing Controls - For use by Maintenance Men.

1) Indicator light No. 10 indicates a critical variation in a DC voltage has automatically turned off the DC voltage even though toggle switch No. 4 is in the "ON" position.

2) Push button No. 11 will reset DC voltage alarm and permit the DC voltage to come back on.

3) Indicator switch No. 12 selects voltage to be measured on meter No. 13.

Arrangement of Data on Magnetic Tapes used with the CRC 102-A

The magnetic tapes used with the CRC 126 Magnetic Tape Unit are manufactured by the Minnesota Mining Corporation. Each tape consists of a thin transparent cellulose acetate strip, one inch wide, approximately 1200 feet long, impregnated on one side with aluminum powder to discharge static electricity, coated on the other side with a magnetic sensitive oxide, and either wound on a reel for storage, or piled in a plastic basket when in use in the tape unit. Magnetic tape should be handled with care, kept in a magnetically shielded can, and should not be exposed to strong AC fields or extreme heat (greater than 300° F).

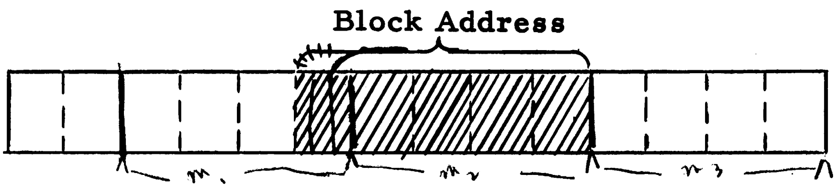
Data is arranged on the magnetic tape in blocks of words. Each is divided into three groups: 1) the Block Marker word, 2) eight words referred to as section No. 1, and eight words referred to as Section No. 2. The following illustration shows how the words appear on the tape:

Block Marker word.

Section No. 1 words.

Section No. 2 words.

a. The block Marker Word has the same structure as an octal number. The fourth through eighth octal digits of the Block Marker Word are the block address.



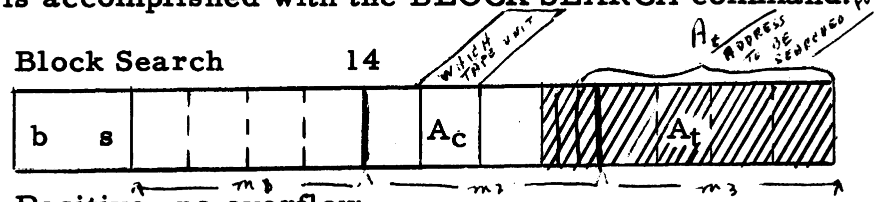
Blocks may be addressed from 00000 through 17, 777. The restrictions on addressing the tape are discussed in the description of the BLOCK SEARCH command (see Section VII, page 55)

b. Data is recorded in the blocks in two sections of eight words each. Reference to the separate sections is discussed in the READ and WRITE commands. (see Section VII, pages 58 and 59).

Block Search on the CRC Model 126

The 126 Magnetic Tape Unit will search for a specific block on the tape independently of the computer, and controlled by the program.

This is accomplished with the BLOCK SEARCH command.

1. Block Search 14
2. 

3. Positive, no overflow
4. Less than 20milliseconds computer time; indeterminate tape unit time.
5. (a) A_c is a single octal digit which specifies the tape unit in which the search is to take place. In systems with only one tape unit A_c is always zero and may be omitted.

   (b) A_t is ~~five octal digits~~ 13 binary bits which specify the address on the tape which is to be searched for. — not over 17777

   (c) It is impossible to tell a tape unit to Block Search when a read or write is taking place; however, it is possible to tell a tape unit to Block Search while a previous search is still in progress on the same unit. In such a case the new BLOCK SEARCH command supersedes the previous one and the tape unit starts to search for the new address.

   (d) The searching process operates as follows:

   Immediately upon receiving the Block Address from the computer, the tape unit will stop any previous incomplete block search which might be in progress, and start the tape moving in the forward direction, that is: each successive block which passes under the read-record head has an address larger than or equal to the previous block. The block searching mechanism compares each block address as it passes under the read-record head with the desired block address. As soon as a block is passed whose address is greater than or equal to the desired block address, the tape stops and reverses its direction. (Note that the tape has gone past the desired address if the address just compared is equal to the desired address because the tape must pass under the head in order to be read.)

The Block Search mechanism now compares each Block Address that passes under the read-record head with the desired address as the tape moves in the reverse direction. As soon as a block is passed whose address is exactly equal to the desired address, the tape stops.

Thus, when block searching for a certain address, the block searching mechanism ignores all addresses smaller than the desired address and will not go past an address greater than the desired address.

Given a tape numbered from A to Z, where A< Z, and assuming that the read-record head is between two values, B and Y. The tape unit is told to search for address X. The following table indicates the behavior of the search.

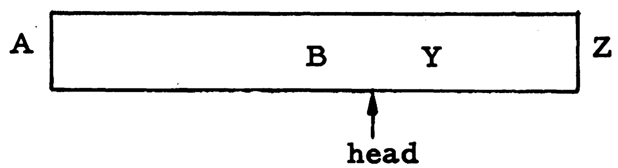


TABLE XIII

CONDITION	TAPE WILL:
$X > Z$	Run off of the capstan at the high end.
$X < A$	Run off of the capstan at the low end.
$A <$ $X < Z$ but X is not equal to any block address on the entire tape.	Run off of the capstan at the low end.
$B = X$, $Y = X$ and not equal to any other address between B and Y.	Stop at Y.
$B \leq X \leq Y$ and X lies between B and Y.	Stop at X.

D.

1. Read Tape 16

for min access

which tape unit

mode: 0, 1, 2

LSD of address in buffer of 1st word to be read [usually zero]

2.

r	t	~~3~~ 0	0	0	0	~~3~~ 0	A_c		K				N

m_1 m_2 m_3

3. Negative, no overflow

4. Approximately 170 milliseconds

5. (a) A_c is a single octal digit which specifies which tape unit will be read from. In systems with only one tape unit A_c will be zero and may be omitted. If the tape unit specified by A_c is block searching when the READ TAPE command is given, reading will not commence until the block search process is complete. Only one block is read with one READ TAPE command. In order to be read, the tape must move; therefore, after every READ TAPE

command on a specific tape, the next READ TAPE command before a BLOCK SEARCH command on that specific tape will read the next adjacent block in the forward direction.

(b) If K is equal to zero, read the BLOCK ADDRESS word into the buffer register cell whose address has the least significant octal digit of N.

(c) If K is equal to one, read the No. 1 section of the block into the buffer register. The first word will read into the buffer register whose least significant digit is N. The remaining eight words will read into the buffer register in cell 200 N + 1 and so forth. When a word has been read to 2007, the next word, if there is one, will read to 2000 and so on. This is because of the cyclic nature of the buffer register.

(d) If K is equal to two, read the No. 2 section of the block specified by A_t into the Buffer Register in the same fashion as described for the number 1 section.

E.

1. Write Tape. 15 for min access

which tape unit — mode: 0, 1, 2 — must be zero

2.

W	t	0	0	0	0	0	A_c		K				O

m_1 — m_2 — m_3

3. Positive, overflow

4. Approximately 170 milliseconds.

5. (a) A_c is a single octal digit which specifies which tape unit will be written on. In systems with only one tape unit, A_c will be zero and may be omitted. If the tape unit specified by A_c is block searching when the WRITE TAPE command is given, writing will not commence until the Block Search process is complete. Only one block is written with any one WRITE TAPE command. In order

to be written on, the tape must move; therefore, each time the WRITE TAPE command is executed, one entire block passes by. (See illustration, Section VII, page 55).

The next WRITE TAPE command before a BLOCK SEARCH command on that specific tape will record on the next adjacent block in the forward direction. To write into both the No. 1 section and the No. 2 section of the same block, the recording on the first section must be followed by a BLOCK SEARCH command to back the tape up one block before recording on the second section. Note that the last octal digit of the WRITE TAPE command <u>must</u> be a zero. If it is different from zero, the tape will not record from the buffer correctly.

(b) If K is equal to zero, write the word in cell 2000 into the Block Marker word of the block.

(c) If K is equal to one, write the contents of the Buffer Register into the No. 1 section of the block in the following manner: Cell 2000 to first cell on No. 1 section, Cell 2001 on second cell of section No. 1 and so on until cell 2007 has been written on the eighth cell of section No. 1.

(d) If K is equal to two, write the contents of the Buffer Register into the No. 2 section of the tape block in the same fashion as it was written in the No. 1 section.

There must be a delay of 4 1/2 drum revolutions between successive "Read" or "Write" Instructions

In ~~[illegible]~~ writing addresses on blank tape - it is necessary to block search "zero" before writing, or to give a dummy read or write command one block search will wipe out ~~[illegible]~~ an immediately previous one which has not yet been completed to be written on, the tape must move; therefore, each time the WRITE TAPE command is executed, one entire block passes by. (See illustration, Section VII, page 55).

The next WRITE TAPE command before a BLOCK SEARCH command on that specific tape will record on the next adjacent block in the forward direction. To write into both the No. 1 section and the No. 2 section of the same block, the recording on the first section must be followed by a BLOCK SEARCH command to back the tape up one block before recording on the second section. Note that the last octal digit of the WRITE TAPE command must be a zero. If it is different from zero, the tape will not record from the buffer correctly.

(b) If K is equal to zero, write the word in cell 2000 into the Block Marker word of the block.

(c) If K is equal to one, write the contents of the Buffer Register into the No. 1 section of the block in the following manner: Cell 2000 to first cell on No. 1 section, Cell 2001 on second cell of section No. 1 and so on until cell 2007 has been written on the eighth cell of section No. 1.

(d) If K is equal to two, write the contents of the Buffer Register into the No. 2 section of the tape block in the same fashion as it was written in the No. 1 section.

- There must be a delay of 4 1/2 drum revolutions between successive "Read" or "Write" Instructions

In ~~xxx~~ writing addresses on blank tape - it is necessary to block search "zero" before writing, or to give a dummy read or write command one block search will wipe out ~~xx~~ an immediately previous one which has not yet been completed.

VIII
USE OF IBM EQUIPMENT WITH THE CRC 102-A

General Information

The CRC 102-A is capable of reading and punching IBM cards.* The National Cash Register Company has entered into an agreement with the International Business Machines Corporation whereby certain punched card machines may be modified to NCR Electronics Division specifications so that they may be used in conjunction with the CRC 102-A.

The CRC 102-A will accept signals read from cards by the IBM Gang Summary Punch, Type 523 with IBM Standard Modification #RFQ37624. The CRC 102-A will provide signals to cause punching into cards by the IBM Gang Summary Punch, Type 523 with IBM Standard Modification #RFQ37625. Both of these machines may be rented from IBM under standard IBM contracts with a nominal charge for the modifications. The modifications are electrical in nature and may be switched in or out as desired so that the IBM machines may be used in their normal fashion with other IBM equipment.

The IBM machines are connected by multiple-wire Summary Punch cables to the CRC 102-A. One end of a cable is permanently attached to an IBM 523 Summary Punch and the other end is equipped with a multiple-contact connector unit.

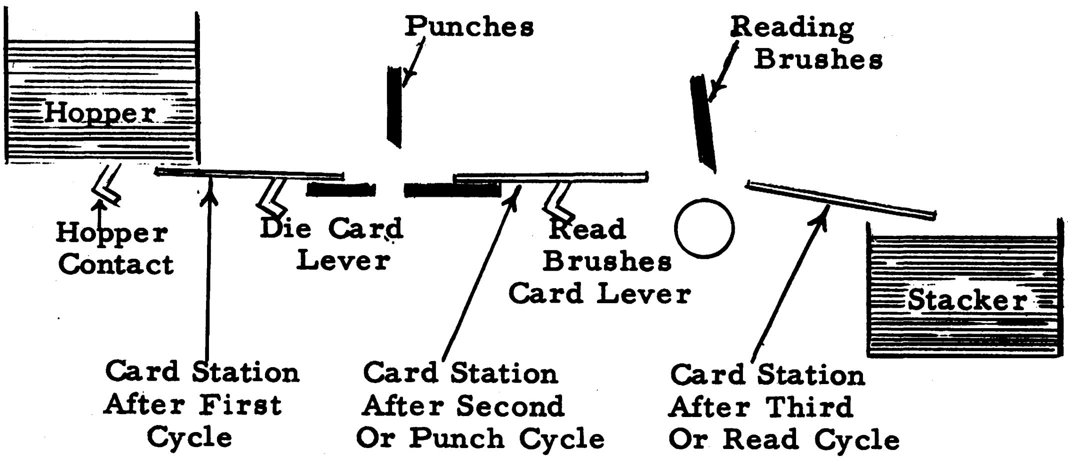
The CRC 102-A is equipped with two stationary cable receptacles, one for the card reader, and one for the card punch, into which the cables from the Summary Punches are inserted.

When preparing to operate the IBM equipment these cables must be plugged into their respective receptacles before the computer is started to insure against computer error.

*The CRC 102-A cannot be used with Remington-Rand cards.

The IBM machines have their own power cables and must be plugged in separately. The power is turned on in the Summary Punches with the Main Line Switch. This must be done before the computer is started.

The IBM machines feed cards face down, the "12" or top edge first at a normal rate of 100 cards per minute. However, when used with the CRC 102-A, the computer controls the punching, and the actual rate of feed depends on several things. A complete discussion of this will be found in Section VIII, page 78. The following illustration demonstrates the path of the card through the Summary Punch Machine.



As can be seen from the diagram, one card must be fed into the computer to the First Card Station before punching can begin, and two cards must be fed before reading starts. As a new card is fed, the cards already in the Summary Punch advance to the next station. The Summary Punches used with the CRC 102-A will do only one operation, either read or punch. Two machines must be used to have both operations.

Reading IBM Cards with the CRC 102-A

Preparing cards in the hopper

The cards to be read are placed in the feed hopper face down, "12" or top edge first. They must be arranged in the order in which the computer is prepared to accept them.

Two cards must be fed to the Summary Punch before computer operations are started so that the first card will be in position to read. (See Section VIII, page 62 .)

The CRC 102-A reads one card at a time and the Summary Punch operates under control of the computer: it does not run continuously unless the computer instructs it to do so.

The Control Panel

The control panel fits into the rack in front of the machine. The control panel shown in figures IV and V together with the following description will indicate the purpose of the various hubs.

a. Punch Brushes: There are 80 outlet hubs for the Punch Brushes. These hubs are wired to the hubs marked with an X on figure IV for reading octal data and to the hubs marked with an X on figure V for reading decimal data.

b. Hubs marked with X (within the Comparing Magnets field):

1) Octal Data (see figure IV).
   There are 56 inlet hubs for transmitting octal data into the 102-A and one inlet hub for transmitting a special signal indicating that a card contains octal data.

   The 56 inlet hubs for data are grouped in four words of fourteen octal digits each. When the words are assembled in the Buffer Register the first word is read from hubs #3-#16, the second from #23-#36, the third from #43-#56, and the fourth from #63-#76 of the comparing magnets field.

The hub #20 of the comparing magnets field must receive a "12" punch signal from every card which is to be read octally. This is necessary for the proper operation of the 102-A card reading process.

2) Decimal Data (see figure V).
There are 40 inlet hubs for transmitting decimal data into the 102-A. They are grouped in four words of ten decimal digits each.

The first decimal digit of each word may be arranged to be a sign digit if desired. A negative sign is indicated as an "X" punch in a column separate from the data columns.* When the words are assembled in the Buffer Register the first word is read from hubs #4-#13, the second from #24-#33, the third from #44-#53, and the fourth from #64-#73 of the comparing magnets field.

Reading Procedure in the 102-A

The complete reading procedure depends on fully automatic apparatus within the computer, the existence or lack of a "12" punch in a particular column, and a parameters specified by the READ CARD

*No column split features are provided on the 102-A; however, if a column split feature is ordered on the Summary Punch it may be used in the normal fashion.

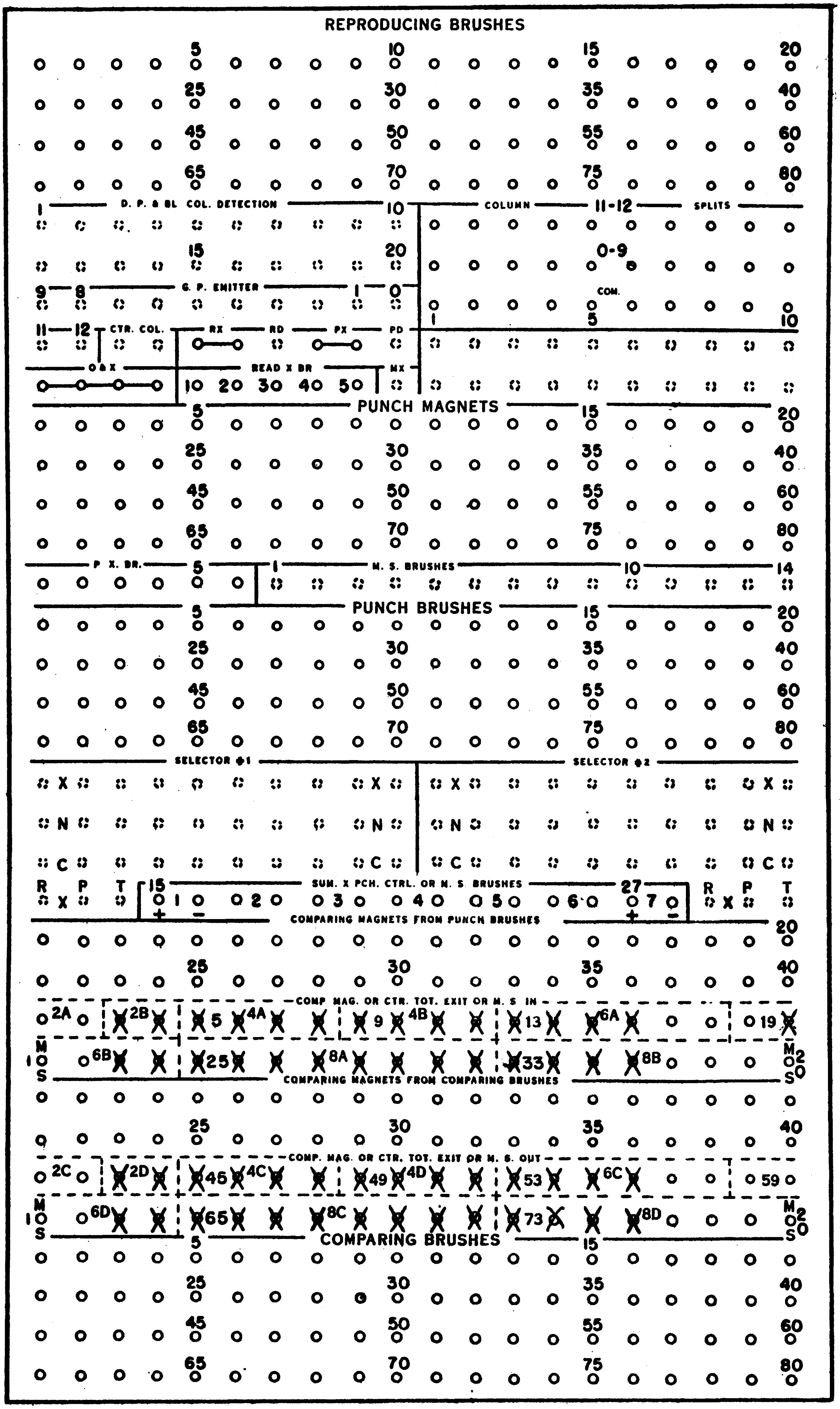

REPRODUCING BRUSHES
D. P. & BL. COL. DETECTION
COLUMN 11-12 SPLITS
0-9
COM.
G. P. EMITTER
CTR. COL.
RX
RD
PX
PD
O & X
READ X BR
MX
PUNCH MAGNETS
P. X. BR.
M. S. BRUSHES
PUNCH BRUSHES
SELECTOR #1
SELECTOR #2
SUM. X PCH. CTRL. OR M. S. BRUSHES
COMPARING MAGNETS FROM PUNCH BRUSHES
COMP MAG. OR CTR. TOT. EXIT OR M. S. IN
COMPARING MAGNETS FROM COMPARING BRUSHES
COMP. MAG. OR CTR. TOT. EXIT OR M. S. OUT
COMPARING BRUSHES

Figure IV

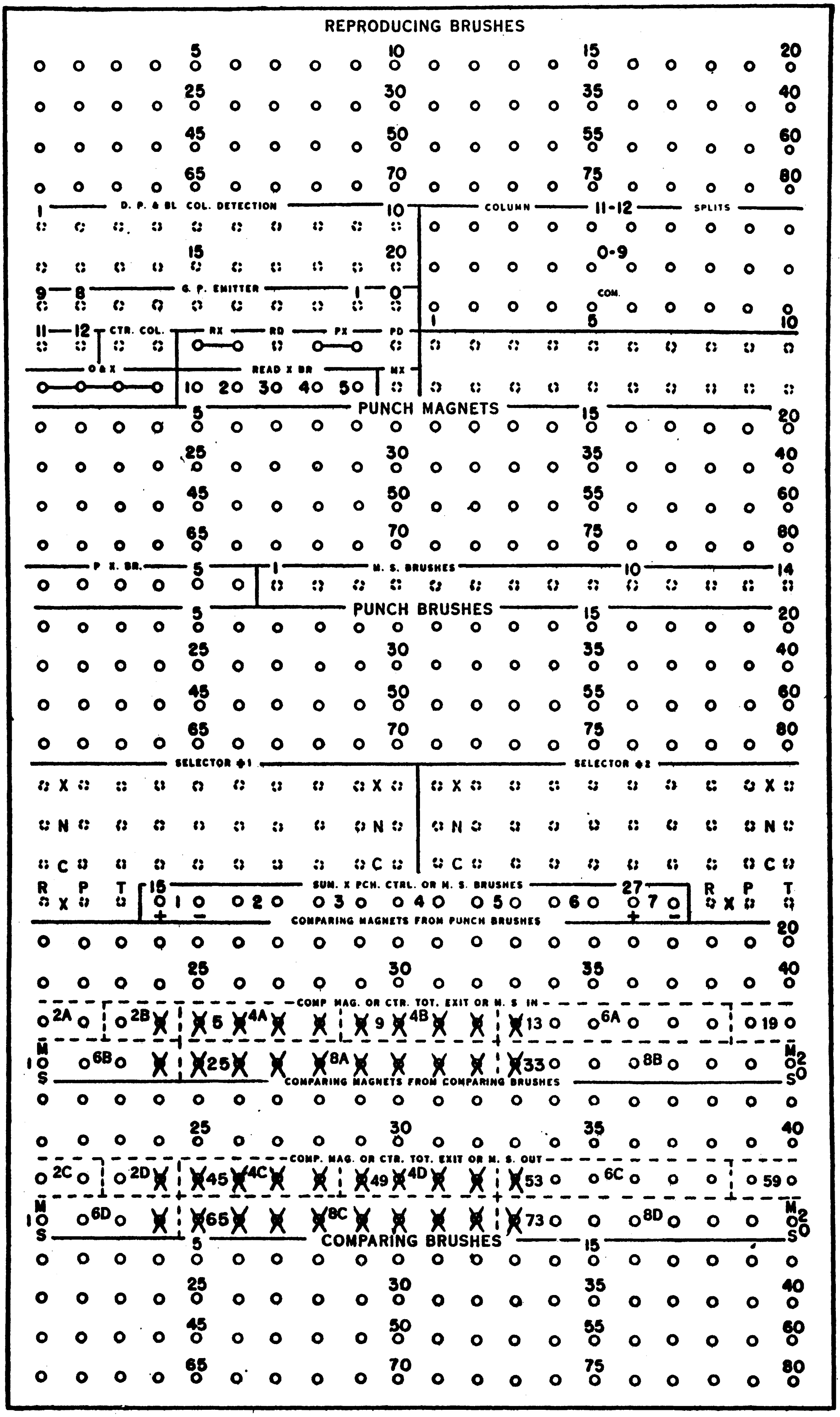

REPRODUCING BRUSHES
D. P. & BL. COL. DETECTION
COLUMN 11-12 SPLITS
G. P. EMITTER
CTR. COL.
RX
RD
PX
PD
O & X
READ X BR
MX
PUNCH MAGNETS
P. X. BR.
M. S. BRUSHES
PUNCH BRUSHES
SELECTOR #1
SELECTOR #2
SUM. X PCH. CTRL. OR M. S. BRUSHES
COMPARING MAGNETS FROM PUNCH BRUSHES
COMP MAG. OR CTR. TOT. EXIT OR M. S. IN
COMPARING MAGNETS FROM COMPARING BRUSHES
COMP. MAG. OR CTR. TOT. EXIT OR M. S. OUT
COMPARING BRUSHES

Figure V

command. There are three components in the automatic apparatus.

a. The Buffer Register assembles words being read in four adjacent cells starting at a cell specified in the READ CARD command. Note that cell 2000 follows cell 2007. Each digit in a word is numbered in a fashion corresponding to the hubs in one row of the control panel.

1	2	3	4	5	6	7	8	9	10

1	2	3	4	5	6	7	8	9	10	11	12	13	14

b. The ~~Input~~ E Register is initially filled by the READ CARD command with a group of digits which are inserted into the corresponding digit positions of the proper words in the Buffer Register when signals are received at the inlet hubs. The ~~input~~ E register is indexed by the ~~addend~~ F register as each row of the card passes under the reading brushes so that holes in each succeeding row insert the next value into the Buffer Register.

c. The ~~Addend~~ F Register is filled by the READ CARD command with a group of digits which are added to the ~~Input~~ E Register while the card moves between holes. The normal configuration will usually be a group of decimal or octal "ones".

d. READ CARD command

1) Read Card 06

2) rc m_1 m_2 m_3

3) Negative, no overflow.

4) Approximately 550 milliseconds for octal reading, 650 milliseconds for decimal reading.

5) a. The first cell in the Buffer Register of the four into which data is to be entered is specified by the least significant digit of m_3. The first cell is the one whose address has the same least significant digit as m_3. Thus, if m_3 is 0000, the first cell would be 2000 and the following three would be 2001, 2002, and 2003. If m_3 is 1726, the first cell would be 2006 and the following three would be 2007, 2000, and 2001.

b. The contents of the ~~input~~ E register is specified by (m_1). This is a different value for the normal octal reading process than for the normal decimal reading process.

(m_1) when reading octal numbers or commands

1	5	5	5	5	5	5	5	5	5	5	5	5	6

(m_1) when reading decimal numbers with sign digit used in X punch.

0	1	6	7	3	5	6	7	3	5	6	7	3	6

(m_1) when reading decimal numbers with sign digit used for numerical data.

1	5	6	7	3	5	6	7	3	5	6	7	3	6

c. The contents of the ~~Addend~~ F Register is specified by (m_2). This is a different value for the normal octal reading process than for the normal decimal reading process.

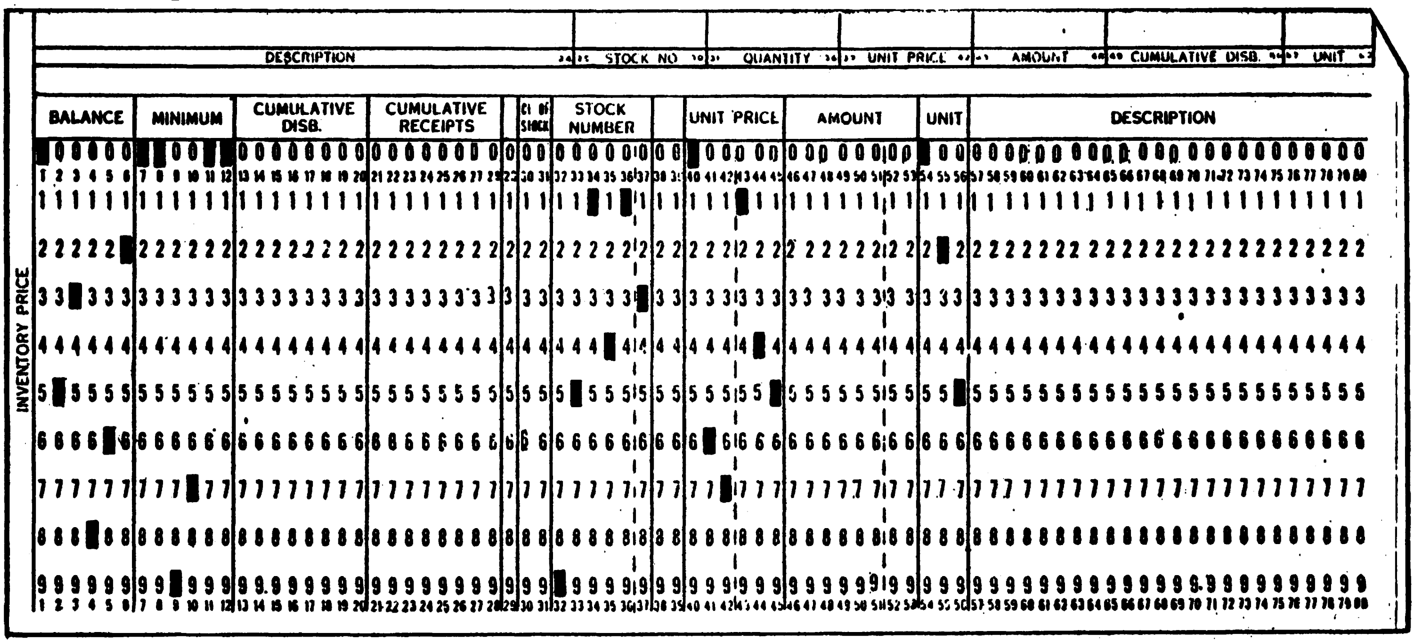
(m_2) when reading octal numbers or commands.

1	1	1	1	1	1	1	1	1	1	1	1	1	1

(m_2) when reading decimal numbers.

0	1	0	4	2	1	0	4	2	1	0	4	2	1

As an example of the complete reading process consider every step in reading data from the following inventory control card into the computer.



a. The following fields will be read into the computer

CARD		COMPUTER		VALUE
Name	Columns	Word	Digits	
Stock No.	32-37	2003	5-10	951413
Balance	1-6	2004	5-10	053862
Minimum	7-12	2005	5-10	009700
Unit	54-56	2006	2-4	025
Unit Price	40-45	2006	5-10	067.145

b. 1) The control panel is first wired as in Figure VI.

2) The Summary Punch is prepared for operation by connecting the Summary Punch Cable to the receptacle on the 102-A, inserting the control panel in its rack, throwing control switches to computer operation position (these switch in the modified circuits) and turning the Main Line Switch "ON".

3) The cards to be read are placed in the feed hopper face down, "12" edge first.

4) The "START" button on the IBM machine is depressed for two card cycles to place first card under reading brushes.

5) The program containing the READ CARD command(s) may now be started.

c. Just prior to the execution of the READ CARD command the Buffer Register should be cleared by executing bl 2100 2100 2100.

d. For the sake of this example the READ CARD command is rc 0201 0202 2003: $(m_1) = (201) = 15673567356736$ (octal).

This in binary is:

00	1101	1101	1101	1101	1101	1101	1101	1101	1101	1110

which may be rather loosely considered as being the decimal number

13	13	13	13	13	13	13	13	13	14

$(m_2) = (202) = 01042104210421$ (octal).

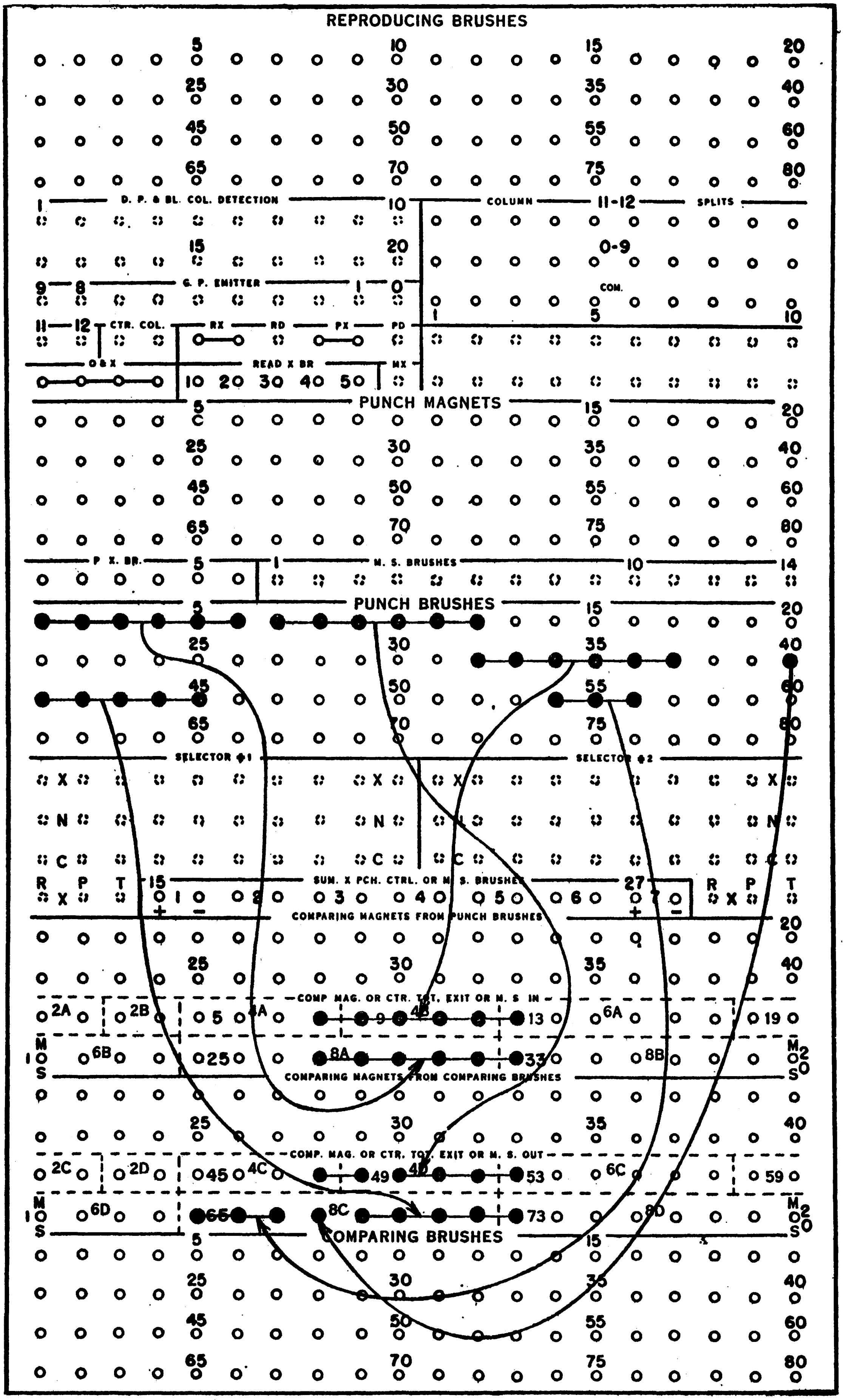

REPRODUCING BRUSHES
D. P. & BL. COL. DETECTION
COLUMN
11-12
SPLITS
0-9
COM.
G. P. EMITTER
CTR. COL.
RX
RD
PX
PD
O & X
READ X BR
MX
10 20 30 40 50
PUNCH MAGNETS
P. X. BR.
M. S. BRUSHES
PUNCH BRUSHES
SELECTOR #1
SELECTOR #2
SUM. X PCH. CTRL. OR M. S. BRUSHES
COMPARING MAGNETS FROM PUNCH BRUSHES
COMP MAG. OR CTR. TOT. EXIT OR M. S. IN
COMPARING MAGNETS FROM COMPARING BRUSHES
COMP. MAG. OR CTR. TOT. EXIT OR M. S. OUT
COMPARING BRUSHES

Figure VI

This in binary is:

00	0001	0001	0001	0001	0001	0001	0001	0001	0001	0001

which is the decimal number

1	1	1	1	1	1	1	1	1	1

m_3 = 2003 which indicates that cell 2003 will be the first cell filled and the other three in order will be 2004, 2005, and 2006. This section of the Buffer Register is clear.

e. The READ CARD command is used with both octally and decimally coded cards. (m_1) and (m_2) are specified in each case by the programmer, but the computer treats a card as decimal unless there is a "12" punch in a particular column. The octal mode inserts each octal digit into 3 binary digits and the decimal mode inserts each decimal digit into 4 binary digits. In this example there is no wire connected to hub #20 so that the data will be assembled in the decimal fashion.

f. The progressive contents of the Buffer Register and the Input Register, as the card moves across the reading brushes and each row of holes is read, is outlined in Table XVI.

g. The termination of the automatic process depends on the contents of the input register. After each row of the card is scanned, before the Addend Register is added onto the Input Register, the least significant digits of the Input Register are examined by the machine. If the reading mode is decimal, the automatic reading process will terminate when the least significant decimal digit is 9, i.e., the binary configuration is 1001. If the reading mode is octal, the automatic reading process will terminate when the two least significant octal

TABLE XVI

(Decimal numbers are used throughout. See note after X row.)

	First Word										Second Word	Third Word	Fourth Word
Start	1	2	3	4	5	6	7	8	9	10	1 2 3 4 5 6 7 8 9 10	1 2 3 4 5 6 7 8 9 10	1 2 3 4 5 6 7 8 9 10
Buffer	0	0	0	0	0	0	0	0	0	0			
Input	13	13	13	13	13	13	13	13	13	14	Input register repeats for each word of buffer.		
Addend	1	1	1	1	1	1	1	1	1	1			
After 12 row:	Lack of special punch has indicated decimal mode.												
Buffer	0	0	0	0	0	0	0	0	0	0			
Input	14	14	14	14	14	14	14	14	14	15			
After X row:	Note decimal 15 equals binary 1111 which causes carry when indexed.												
Buffer	0	0	0	0	0	0	0	0	0	0			
Input	0	0	0	0	0	0	0	0	0	0			
After O row:	Since Buffer Register was clear, O's are not visible.												
Buffer	0	0	0	0	0	0	0	0	0	0			
Input	1	1	1	1	1	1	1	1	1	1			
After 1 row:													
Buffer	0	0	0	0	0	0	1	0	1	0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1 0 0
Input	2	2	2	2	2	2	2	2	2	2			
After 2 row:													
Buffer	0	0	0	0	0	0	1	0	1	0	0 0 0 0 0 0 0 0 0 2	0 0 0 0 0 0 0 0 0 0	0 0 2 0 0 0 0 1 0 0
Input	3	3	3	3	3	3	3	3	3	3			
After 3 row:													
Buffer	0	0	0	0	0	0	1	0	1	3	0 0 0 0 0 0 3 0 0 2	0 0 0 0 0 0 0 0 0 0	0 0 2 0 0 0 0 1 0 0
Input	4	4	4	4	4	4	4	4	4	4			
After 4 row:	0	0	0	0	0	0	1	4	1	3	0 0 0 0 0 0 3 0 0 2	0 0 0 0 0 0 0 0 0 0	0 0 2 0 0 0 0 1 4 0
After 5 row:	0	0	0	0	0	5	1	4	1	3	0 0 0 0 0 5 3 0 0 2	0 0 0 0 0 0 0 0 0 0	0 0 2 5 0 0 0 1 4 5
After 6 row:	0	0	0	0	0	5	1	4	1	3	0 0 0 0 0 5 3 0 6 2	0 0 0 0 0 0 0 0 0 0	0 0 2 5 0 6 0 1 4 5
After 7 row:	0	0	0	0	0	5	1	4	1	3	0 0 0 0 0 5 3 0 6 2	0 0 0 0 0 0 0 7 0 0	0 0 2 5 0 6 7 1 4 5
After 8 row:	0	0	0	0	0	5	1	4	1	3	0 0 0 0 0 5 3 8 6 2	0 0 0 0 0 0 0 7 0 0	0 0 2 5 0 6 7 1 4 5
After 9 row:	0	0	0	0	9	5	1	4	1	3	0 0 0 0 0 5 3 8 6 2	0 0 0 0 0 0 9 7 0 0	0 0 2 5 0 6 7 1 4 5

digits are both 7's, i.e., the binary configuration is 111 111.

h. If a card is multiple punched in a column being read, the largest digit will be entered into the Buffer Register. Actually, each successive digit enters its configuration with the last configuration remaining in the end. Note that an X punch cannot be used for a negative sign in the octal mode.

Punching IBM Cards with the CRC 102-A

Preparing cards in the hopper.

The cards to be punched are placed in the feed hopper face down, "12" or top edge first. If they are prepunched with data which is related to the data in the computer in a particular order, they must be arranged in the order the computer is prepared to punch them.

One card must be fed to the Summary Punch before the computer operations are started, so that the first card will be in position to be punched. (See Section VIII, page 62).

The 102-A punches cards one at a time and the Summary Punch operates under control of the computer; it does not run continuously unless the computer instructs it to do so.

The Control Panel

The control panel fits into the rack in front of the machine. The control panel shown in Figures IV and V together with the following description will indicate the purpose of the various hubs.

Punch Magnets: There are 80 inlet hubs for the punch magnets. These hubs are wired from the hubs marked with an X on Figure IV for reading octal data and to the hubs marked with an X on Figure V for reading decimal data.

Hubs marked with an X:

1) Octal data (see Figure IV).

The outlet hubs for punching are identical to the inlet hubs for reading. They are grouped in the same fashion as for reading and refer to the same respective cells in the Buffer Register.

The hub #20 emits a "12" punch signal when punching octal cards. Most often octal cards will be punched for later reëntry into the 102-A. This hole can then be used to control the reading process.

2) Decimal data (see Figure V).
The outlet hubs for punching are identical to the inlet hubs for reading. They are grouped in the same fashion as for reading and refer to the same respective cells in the Buffer Register.

Punching Procedure in the 102-A

The complete punching procedure depends on fully automatic apparatus within the computer, whether the command is PUNCH OCTAL or PUNCH DECIMAL, and on parameters specified by the commands. There are three components in the automatic apparatus.

a. The Buffer Register must contain the words to be punched in four adjacent cells starting with the cell specified by the punch command. Each digit in a word is numbered in a fashion corresponding to the hubs in one row of the control panel. (See description of READ CARD command).

b. The Comparison Register is initially filled by the punch commands with a group of digits which are compared to the digit positions of the proper words in the Buffer Register. When a digit in the Buffer Register compares to a digit in the Comparison Register a signal is transmitted to the hub on the control panel corresponding to the digit position in the Buffer Register. The Comparison Register is indexed as each row of the card passes under the punch dies so that each digit is punched in the proper row of the card.

c. The Addend Register is filled by the punch commands with a group of digits which are added to the Comparison Register while the card moves between holes. The normal configuration will be a group of decimal or octal "ones".

d. PUNCH OCTAL command.

1) Punch Octal 13

2) po m_1 m_2 m_3

3) Negative, overflow.

4) Approximately 550 milliseconds.

5) a. The first cell in the Buffer Register of the four whose contents are to be punched is specified by the least significant digit of m_3. The first cell is the one whose address has the same least significant digit as m_3. (See READ CARD command.)

b. The contents of the Comparison Register are specified by (m_1). For the normal octal punching process this value is:

2	6	6	6	6	6	6	6	6	6	6	6	6	7

c. The contents of the Addend Register is specified by (m_2). For the normal octal punching process this value is:

1	1	1	1	1	1	1	1	1	1	1	1	1	1

d. The PUNCH OCTAL command causes a special signal to be emitted from hub #20 to punch the "12" row in a particular column.

e. Punch Decimal Command:

1) Punch Decimal 12

2) pd m_1 m_2 m_3

3) Negative, no overflow.

4) Approximately 650 milliseconds.

5) a. The first cell of the Buffer Register of the four whose contents are to be punched is specified by the least significant digit of m_3. The first cell is the one whose address has the same least significant digit as m_3. (See READ CARD command).

b. The contents of the Comparison Register is specified by (m_1). For punching an X in a particular column when the negative sign is coded as a two, this value is:

0	2	7	3	5	6	7	3	5	6	7	3	5	7

For punching numerical data in all ten columns this value is:

1	6	7	3	5	6	7	3	5	6	7	3	5	7

c. The contents of the Addend Register is specified by (m_2). For punching the sign digit in the X row this value is:

0	0	0	4	2	1	0	4	2	1	0	4	2	1

For punching numerical data in all ten columns this value is:

0	1	0	4	2	1	0	4	2	1	0	4	2	1

f. The termination of the automatic process for punching cards depends on the contents of the Comparison Register. After each row of the card is punched, before the Addend Register is added onto the Comparison Register, the least significant digits of the Comparison Register are examined by the machine. If the command was PUNCH DECIMAL the automatic punching process will terminate when the least significant decimal digit is nine, i.e., the binary configuration is 1001. If the command was PUNCH OCTAL, the automatic punching process will terminate when the two least significant octal digits are 7's, i.e., the binary configuration is 111 111.

Time Relationship between CRC 102-A and IBM Equipment.

The IBM Summary Punch, Type 523, which is recommended for use as a card reader or card punch with the CRC 102-A system, has a basic card feeding speed of 100 cards per minute. It will feed cards at this rate only if the start clutch is continuously energized. This clutch is arranged so that if it is de-energized while the machine is feeding cards, it will wait until the present card has passed beyond the punch dies or reading brushes, as the case may be, and then stop in a position such that all reading brushes or punch dies are between two cards. The clutch always stops in this position, and, when re-energized, starts from this position. When the clutch is re-energized after being at rest, it waits again until a dog on the motor shaft rotates to the proper position to engage and begin to drive the clutch. Thus the motor shaft, when it is driving the clutch, is always in the same position relative thereto. When the clutch is disengaged, after having been running, the clutch itself stops in the position referred to above. The motor shaft, however, is not positively stopped, and continues to coast a bit after having been disconnected from the clutch. Accordingly it may come to rest in any random position. But when being re-energized, the clutch must wait for the motor shaft before it can engage; thus there is always some delay before the clutch starts to move after it is energized, if it was formerly stopped. This delay depends on the position that the motor shaft was in just before energizing the clutch. If the motor shaft happened to be in the most favorable position, there would be almost no delay; but if the motor shaft happened to be in the least favorable position, the clutch might have to wait almost a full revolution of the motor shaft, or 600 milliseconds, before it begins to move.

This relationship between clutch and motor shaft is important, also, when feeding cards continuously. The clutch is energized momentarily by the 102-A at the beginning of a card cycle, but the card continues through the entire cycle since the clutch cannot stop except in a position between cards. If the card machine is to continue to the next card, the clutch must be re-energized before the completion of the card cycle; otherwise it will stop. If the clutch is energized only a millisecond after the completion of the card cycle, the dog on the motor shaft will have passed beyond the position in which it can engage the clutch, and the clutch will remain stationary until the motor shaft makes another revolution, thus wasting a card cycle. In a practical case, the time delays inherent in the operation time of the clutch and its associated relays dictate that the clutch must be re-energized a minimum of about 20 milliseconds <u>before</u> the end of a card cycle; that is, about 580 milliseconds after the card cycle begins.

Reading in the Octal Mode: A READ CARD command in the octal mode takes approximately 487.5 milliseconds to complete. Commands taking no less than $2\frac{1}{2}$, and preferably 3, drum revolutions <u>must</u> be inserted between two READ CARD (octal) commands. If less than $2\frac{1}{2}$ drum revolutions are used, the computer will not obey the second READ CARD command. If more than 3 drum revolutions are used, the computer will cause the card feed rate to be 50 per minute instead of 100 per minute.

Reading in the Decimal Mode: A READ CARD command in the Decimal Mode takes approximately 562.5 milliseconds. In order to achieve the maximum rate of feed of the IBM cards of 100 cards per minute, each READ CARD command must immediately follow the original. If so much as a one-half of a drum revolution command, such as ADD, is inserted between two commands in the decimal mode, the computer will call for cards at the rate of 50 per minute.

Punching Cards: The PUNCH OCTAL and PUNCH DECIMAL commands operate in the same way as the READ CARD commands.

A delay of from $2\frac{1}{2}$ to 3 drum revolutions must be inserted between each Punch Octal command to insure continuous feed at the rate of 100 cards per minute.

The computer will not punch decimal cards at a speed of 100 cards per minute even if the PUNCH DECIMAL commands are consecutive. If a delay of even one-half of a drum revolution is inserted, the speed drops to 50 per minute.

IX

THE AUTOMATIC USE OF THE FLEXOWRITER

The Flexowriter is the primary input-output device associated with the CRC 102-A and has been described above in Section III, page 11 . This chapter describes the PRINT and FILL commands which are used to control the Flexowriter in a fully automatic fashion.

Printing out on the Flexowriter.

Results of computation, previously stored permanent data, program monitoring information, and computer alarms (see Section X) are printed out on the Flexowriter. Except for the Alarms all printing is under control of a program.

Data may be encoded for printing in any one of the four normal forms. The printer automatically interprets the binary coding of the data and types out the proper character indicated by the code. There is a single print command used for all printing operations.

1) PRINT 21

2) pr m_1 m_2 m_3

3) Positive, overflow

4) Approximately 100 milliseconds for each character printed.

5) a. The total number of words to be typed out with the one print command is m_3 taken as an octal integer. m_3 must be greater than zero.

   b. The first word to be typed is (m_1). If more than one word is typed, the following words are (m_1+1), (m_1+2), . . . , (m_1+m_3-1). All words are typed in the same mode as the first word.

c. The printing mode; i.e., octal printing, decimal printing, or alphabetic printing (for all words printed with the same print command), is controlled by the sign digits of (m_2) according to the following table.

No Address	Address	
00	02	OCTAL
01	03	DECIMAL
10,11	12,13	ALPHABETIC

m_2 sign	Mode of printing
00	- Type the octal sign digits and the magnitude as an octal number.
01	- Type the sign digit according to Table XVII below and type the magnitude as a decimal number according to Table IV.
02	- Type the octal address of the cell being printed followed by a space and type its contents as in 00 above. OCTAL NUMBER
03	- Type the octal address of the cell being printed, followed by a space and type its contents as in 01 above. DECIMAL NUMBER

These four modes of typing automatically tab the Flexowriter carriage to the next tab stop preset by the operator after printing each word. A special adjustable tab stop is fitted on the Flexowriters which automatically returns the carriage at the end of the printed line as part of the automatic tabbing operation.

10	- The alphabetic printing mode allows the program to activate any key on the Flexowriter. The

magnitude of the word to be printed is divided into six pairs of octal digits and is coded as in Section III, page 20, Table VIII. There is no automatic tabbing after the printing of each word. Any desired editorial characters must be coded as any other alphabetic characters. This includes shifting up and down, color shifting, backspace, tabbing, etc. It must be remembered, though, when printing in the alphabetic mode that the automatic carriage return tab stop must be moved far enough to the right to avoid returning the carriage while printing.

Decimal

TABLE XVII

Flexowriter Printing of Sign Digit in 01 Mode + 03 mode

BINARY CONFIGURATION	octal print out	SIGN PRINT OUT (decimal)
00 0000	-0	+
00 0001	-1	p X
00 0010	-2	-
00 0011	-3	n X
00 0100	-4	4 X
00 0101	-5	8 X
00 0110	-6	6 X
00 0111	-7	5 X
00 1000	-8	
00 1001	-9	Ignores (Prints nothing)
X 00 1010	- delete	
X 00 1011	- ignore	
00 1100	- space	Space
X 00 1101	- back	Back space
X 00 1110	- ignore	Ignores
00 1111	- period	

see p 21

X - cannot fill

d. The number of digits which will be printed from the magnitude section of each word is controlled by the magnitude section of (m_2). This is effected by placing in (m_2) a binary one in the least significant binary digit position of the octal, decimal, or alphabetic digit position corresponding to the last digit which is to be typed from a single word, except that, if no bit is stored, all digits will print. This feature affects only the magnitude section of all ~~three~~ five printing modes.

6. Examples:

	$\boxed{\text{pr 1726 2100 0002}}$	(1726) = 35 2000 2001 2003 ~~(1727)~~
(1727) =	26 2003 1325 2003	(2100) = 00 0000 0000 0000

result: 35 2000 2001 2003
26 2003 1325 2003

pr 1001 0204 0001 (1001) = 14 1761 2025 4400
(0204) = 03 0000 0000 0400

result: 1001 3.14159

pr 0000 0133 0010 (0133) = 10 0000 0000 0000
(0000) = 00 3675 3274 4564
(0001) = 00 3666 3241 7560
(0002) = 00 7766 4171 6436
(0003) = 00 6232 4144 7464
(0004) = 00 3663 3245 7360
(0005) = 00 4475 4563 6436
(0006) = 00 6232 7776 7041
(0007) = 00 6654 1212 1234

result: The National Cash Register Company

Automatic Reading of Flexowriter Tapes

Tapes which are prepared as described in Section III, page 20, may be read into the computer by manually starting the paper tape reader or by having the computer automatically start the paper tape reader with the FILL command. When the paper tape is to be used with no manual intervention, after being started either manually or automatically, it must contain certain special characters which control it and the computer. Specifically, these characters are:

s - Start computation, and Stop Code - stop paper tape reader.

If the Flexowriter is to be used for output data after reading in, it will usually be necessary to edit the sheet with carriage returns which may be included on the tape before the Stop Code. The tape may continue to run after computation has begun for purposes of editing and preparing column headings, titles, etc.

1. Fill 11
2. fl 3000 3000 m_3
3. Positive, overflow
4. Indeterminate
5. a. Stop computation and return machine to idle.
   b. Clear the input register.
   c. Prepare computer to accept octal.
   d. Leave Control Register prepared to place the first word entered from the Flexowriter in the normal filling operation into cell m_3.
   e. Start Flexowriter Paper Tape Reader.
   f. When the read-in process is finished and if computer is not restarted with an "s" code punched in the tape, the computer will remain in the idle condition.

X

AUTOMATIC ALARM CHECKS

The CRC 102-A is provided with two automatic alarm checks as indications of programming blunders and as some protection against improper machine operation.

The Alarm proper on the 102-A consists of the spontaneous unexpected printing of cell 3000 immediately followed by the computer returning itself to the idle condition. The contents of cell 3000 is completely described in Section IV, page 30 . The special interpretations as an alarm are indicated below.

No Command Alarm

If during computation the computer tries to execute as a command, a number, or a word whose sign digits do not represent a command, the computer will cause a No Command Alarm. This helps to check the continuity of the program, the proper transfer of control in a test command, and failure of the memory reference circuitry of the computer.

Overflow Alarm

If an ADD, SUBTRACT, or DIVIDE command causes an overflow as described in their respective sections, and the command is not immediately followed by a TEST FOR OVERFLOW MARKER or a SHIFT LOGICALLY command, and switch #19 on the console marked "AUTOMATIC OVERFLOW TEST" is in the "IN" position, the computer will cause an Overflow Alarm.

It is not necessary that the Test for Overflow or Shift Logically commands operate on the overflow result produced by the guilty command.

Interpretation of cell 3000 when printed as an alarm

The alarm occurs because of one of two things happening:

1. Either (m_2-2) generated an overflow; or
2. m_2-1 is not a command.

Whenever 3000 prints out as an alarm, print out cells (m_2-2) and (m_2-1) and examine these. If m_2-1 is a good command, then examine the results of the command executed by m_2-2. This should have generated an overflow.

Skill in diagnosing program alarms will come from experience in operating the computer.

XI

TECHNIQUES OF MINIMUM ACCESS CODING

As previously mentioned (Section V) the minimum execution times bear little relationship to the speed of the computer. This is due to the nature of the addressing of commands and their positioning on the drum. When a command is read from the memory, regardless of the execution time of the arithmetic proper, the next command can be read from its position on the drum only when it is under a read-write head. Then, between two commands, there must always be a delay at least equal to the time it takes for the drum to carry the next command under its read-write head. For a drum numbered consecutively around the periphery, and with each succeeding command in the next adjacent cell, and allowing some time for the execution of the command, the minimum time between two commands is the time of one drum revolution.

The delay is further aggravated by the need to refer to three other positions on the drum during the execution time, each operand and the destination cell would have to be in optimum peripheral positions on the drum so that they would be under a read-write head just when required in the execution of the command. The Buffer Register simplifies this process of placing the operands and destinations in optimum positions so as to make possible the achievement of near minimum execution times. With judicious use of the Buffer Register combined with careful placing of constants on the drum it is possible to execute most commands in one-half revolution of the drum or less. A renumbering of the drum has been developed which allows the machine to take advantage of this saving in time. This renumbering process places each successive address on the opposite side of the drum so that the drum revolves one-half of a revolution to reveal the next command.

Table XVIII indicates this numbering principle.

TABLE XVIII

First Quadrant	Second Quadrant	Third Quadrant	Fourth Quadrant
00	20	40	60
41	61	01	21
02	22	42	62
43	63	03	23
04	24	44	64
45	65	05	25
06	26	46	66
47	67	07	27
10	30	50	70
51	71	11	31
12	32	52	72
53	73	13	33
14	34	54	74
55	75	15	35
16	36	56	76
57	77	17	37

There are certain commands which require attention when used with a renumbered drum. They are specifically: md, dd, sf, bo, and bl. These commands refer to adjacent cells in the main memory, which are not consecutively addressed on the renumbered drum. Where a great deal of input-output data must be used with IBM cards or magnetic tape, it may be more desirable to have a drum numbered as in Table XIX. This is an example to illustrate the possibilities of combining a regularly numbered drum with a renumbered drum.

TABLE XIX

First Quadrant	Second Quadrant	Third Quadrant	Fourth Quadrant
00	40	20	60
21	41	01	61
02	42	22	62
23	43	03	63
04	44	24	64
25	45	05	65
06	46	26	66
27	47	07	67
10	50	30	70
31	51	11	71
12	52	32	72
33	53	13	73
14	54	34	74
35	55	15	75
16	56	36	76
37	57	17	77

The time required to execute a given command consists of certain arithmetic and latent times as well as the access time to the drum. In order to assign optimum positions to the operands these latent times must be known. The following Table Number XX, defines all of the relative positions for the operands and destinations. All numbers in the table are in the octal notation. The relative positions define space around the drum regardless of the numbering of the Word Channel. Column titles are defined as follows:

1. Command - the various commands including their variations which may be minimum access coded.

2. Minimum Revolution - to be more accurate this figure should be 33/64 and 1-33/64 where it reads 1/2 and 1-1/2. If all minimum access restrictions outlined in the remaining columns of the table are adhered to, this is the execution time. This figure is more

TABLE XX

Command	Min Rev	Position of m_1 Early	Position of m_1 Late	Position of m_2 Early	Position of m_2 Late	Position of m_3 Early	Position of m_3 Late	with cmd	Min Execution Time Word Time	Min Execution Time Ms
		I+	I+	m_1+	I+	m_2+	I+	m_3+		
Add										
#'s same sign	½	3	21	3	24	12	36	3	23	7.4
#'s diff. sign	½	3	17	3	22	14	36	3	25	8.2
(m_1) command	½	3	21	3	24	12	36	3	23	7.4
Subtract										
#'s same sign	½	3	17	3	22	14	36	3	25	7.4
#'s diff. sign	½	3	21	3	24	12	36	3	23	8.2
(m_1) command	½	3	21	3	24	12	36	3	23	7.4
Multiply Round	1½	3	44	3	47	67	136	3	100	25.0
Multiply Double	1½	3	44	3	47	66	135	4	100	25.0
Divide Round	1½	3	42	3	45	71	136	3	102	25.8
Divide Double	1½	3	42	3	45	70	135	4	102	25.8
Shift Magnitude *3	½	3	16-2n	3	21-2n	15+2n	36	3	26+2n	8.6+.78n
Shift Logically *3	½	3	15-2n	3	20-2n	16+2n	36	3	27+2n	9.0+.78n
Scale Factor *3	½	3	11-2n	3	14-2n	21+2n	35	4	33+2n	10.6+.78n
Extract										
Into Main Memory	1½	3	21	3	24	12	36	3	123	32.4
Into Buffer *4	½	3	11	3	14	12	26	13	33	10.6
Into 210X *4	½	3	~~[illegible]~~ 15	3	~~[illegible]~~ 20	~~12~~ 16	36 ~~26~~	3	33	10.6
Buffer Load	½	*1		*1		I+25	27	12	37	12.1
Buffer Out	½	*1		*1		I+26	27	12	32	10.2
Test Overflow										
Test Works		3	m_3-16	*1		m_1+16			21	6.6
Test Fails	½	3	26	*1						
Test Magnitude										
Test Works		3	m_3-21	3	m_3-16	16			24	7.8
Test Fails	½	3	23	3	26					
Test Algebraically										
Test Works		3	m_3-22	3	m_3-17	17			25	8.2
Test Fails	½	3	22	3	25					
Test Switch										
Test Works		*2		*1		I+20			20	6.3
Test Fails	½	*2		*1						

*1 Use 3000 or 2100

*2 Use 3010, 3020, 3040, 3100 instead of 2010, 2020, 2040, 2100

*3 Depends on number of shifts -n

*4 Minimum access coding restrictions when extracting into the Buffer Register or with cell 210X are independent of any other than the least significant octal digits of the address; others, though specified in this table, may be ignored.

realistic than the quotations of exact execution time because of the numbering of the drum; see comment in Introduction of Section V, page 37 .

3. Earliest position for m_1 is three words after the location of the command. This is not significant in bl or bo where some perfectly minimum access coded cell like 3000 or 2100 may be used for convenience.

The ts constants of 2010, 2020, 2040, and 2100 may be replaced by 3010, 3020, 3040, and 3100 for minimum access coding.

4. Latest position for m_1 also depends on the position of m_2. If the earliest position for m_2 is used, there is only one place for m_1.

5. Earliest position of m_2 is three words after m_1. This is not significant in bl, bo, to, or ts, where 3000 or 2100 may be used.

6. Latest position of m_2 also depends on the position of m_3. If the earliest position for m_3 is used, there is only one place for m_2.

7. Earliest position for m_3 depends on the arithmetic time of the commands and the position of m_2 except in the bl, bo, and ts commands where m_1 and m_2 are always perfectly minimum access coded.

8. Latest position of m_3 depends on whether the command records a single or double length result except in bl and bo and assumes the drum is numbered as in Table XVIII

9. The Minimum Execution Time - Word Time is the number of words (counted octally) passing under the head of the drum while the command is being executed if the earliest positions are used for m_1, m_2, and m_3. The last column "ms" translates this figure into milliseconds.

APPENDIX I

Introduction to the Binary and Octal Number Systems

The most common number system in use today is the decimal number system. It is so called because of its dependence on the number 10 (ten). There are ten digits: 0, 1, 2, 3, ..., 9, in the system, and every number may be expressed as a sum of a set of these integers each multiplied by some power of 10. The number is written as a set of these integers with the integral and fractional parts of the number separated by a point called the decimal point. The position of the integers in the set denoted the power of 10 by which the number is to be multiplied. Numbers to the left of the point are multiplied by positive powers of 10 and numbers to the right of the point are multiplied by negative powers. The following diagram shows the powers and digit positions relative to the point.

	Positions [redacted]. LEFT						Pt	Positions [redacted]. RIGHT						
Position	6	5	4	3	2	1		1	2	3	4	5	6	
Power	5	4	3	2	1	0	.	-1	-2	-3	-4	-5	-6	.....

For example the number 13529.723 is interpreted

$$1x10^4 + 3x10^3 + 5x10^2 + 2x10^1 + 9x10^0 + 7x10^{-1} + 2x10^{-2} + 3x10^{-3}$$

In a like manner the binary number system may be defined with the number two replacing the number 10 in the above discussion. There are only two digits in the binary system, 0 and 1; and every number may be expressed as a sum of a set of these digits each multiplied by a power of 2. Thus the number 101101 is interpreted

$$1x2^5 + 0x2^4 + 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 1x32 + 1x8 + 1x4 + 1x1$$

In the decimal system this number would be equal to 45.

In the binary system fractions are expressed as combinations of the two digits 0 and 1 multiplied by negative powers of two. The fraction .1101

$$1x2^{-1} + 1x2^{-2} + 1x2^{-4} = 1x.5 + 1x.25 + 1x.0625$$

In the decimal system this fraction would be equal to .8125.

The point which separates the integral from the fractional part of a binary number is called the binary point. The diagram above is applicable to the binary system.

Arithmetic in the binary system is similar to arithmetic in the decimal system.

a. The rules for addition are

1. 0 + 0 = 0
2. 1 + 0 = 1
3. 1 + 1 = 10

As an example the sum of the two numbers, 1011101 and 1100110

```
Carries 11111
         1011101
        +1100110
  Sum   11000011
```

b. The rules for subtraction are

1. 0 - 0 = 0
2. 1 - 1 = 0
3. 1 - 0 = 1
4. 0 - 1 = 1 with one borrowed

As an example, the difference of the two numbers 11000011 and 1000110

```
            11000011
          -  1000110
Difference   1111101
```

c. The multiplication table for binary digits is

1. 0 x 0 = 0
2. 1 x 0 = 0
3. 1 x 1 = 1

The rules for multiplication and division in longhand are exactly similarly to those in the decimal system. For example, the multiplication of 1101.1 by 11001.1

```
         1101.1
      x 11001.1
      ---------
          11011
         11011
        00000
       00000
      11011
     11011
   ------------
   101011000.01
```

In the binary system a number of any size or a fraction of any precision requires a long string of zeros and ones. In practice this becomes very difficult to work with outside the computer so a substitute system which incorporates the binary system exactly is usually substituted. One system which lends itself to this definition is the octal system. The octal system has as a base the number eight, which is equal to 2^3. Thus any combination of three binary digits gives a digit in the octal system. For all the possible combinations and their octal equivalent see the Table I, Section II, page 3 . Thus in the discussion of the computer commands and numbers, the octal system has been used throughout. As an example of a number expressed in the three modes

DECIMAL	252.859375
BINARY	11111100.110111
OCTAL	374.67

As another example consider the largest integer which may be used directly in the CRC 102-A, $2^{36}-1$. This number, in binary, is expressed by 36 ones. In decimal it requires eleven decimal digits: 68 719 476 736; and in octal it requires twelve digits: 777 777 777 777. Therefore it is easy to see that the octal notation is almost as economical as the decimal but has the advantage of being exactly convertible to the binary with which the computer operates.

GLOSSARY

Access Time - The time required for a certain cell on the surface of the rotating drum to be carried under a read-write head after the computer has called for it.

Address - The number which refers to a certain cell.

Arithmetic Time - The time required to execute an arithmetic operation after the operands have been located on the drum.

Bus - A wire or group of wires over which information common to a number of units is carried. The unit which finally uses the information connects itself to the bus through special switching circuitry.

Cell - A position in the main memory or in a register which can hold a word. Each cell has a distinguishing address.

Command - A command is an instruction followed by the addresses of its operands. A command completely specifies a single computer operation.

Delay Line - A memory device whereby information continually reappears under a read-write head once each cycle of a certain number of word times.

Flexowriter - A trade name for an automatic electric typewriter manufactured by the Commercial Controls Corporation.

Input - Information which has been prepared for entry into the computer.

Instruction - A specific arithmetic, logical, transfer, input, or output operation which the computer is capable of performing.

Latent Time	- The time used by the computer in internal operation other than arithmetic or access times.
Logical Circuitry	- Electronic circuitry designed so that the behavior of electrical quantities in the circuit follow relationships predicted by logical equations.
Memory	- The storage facility within the computer for commands and data. The 102-A has a magnetic drum memory.
Minimum Access Coding	- The coding process which locates operands in a fashion which minimizes their access time and permits faster overall operation of the computer.
Negative Zero	- Due to the nature of the computer, a word containing zero in its magnitude section may have either sign affixed. They are treated as mathematically identical but logically distinct by the machine.
Operation Time	- The total execution time of a command, which is the sum of the latent time, the access time, and arithmetic time.
Output	- Information produced by the computer under automatic control of the program.
Overflow	- The indication that arithmetic operation has exceeded the capacity of the arithmetic register.
Positive Zero	- See "Negative Zero."
Program	- The list of commands which are entered into the computer to accomplish a certain complete problem.
Read-Write Head	- The piece of electronic hardware which reads information from the drum and writes information onto the drum.

Register	- A cell or group of cells outside of the main memory which are used for a particular purpose. A register is usually a delay line.
Relative Programming	- A programming technique whereby addresses assigned to commands do not refer to a particular cell but rather to the relative position with regard to some subsequently assigned arbitrary cell.
Scale Factor	- A factor, usually some integral power of two or ten, which a value has been multiplied by so that the value may be treated in a standardized fashion within the computer.
Sub-routine	- A short program used separately from the body of the main program to do frequently needed operations which are too complicated to completely specify each time they are required.
System	- The complete group of equipment around the computer and the computer, which is used as a unit by the operator.
Word	- The basic data unit within the computer. A word is contained in a cell within the memory.