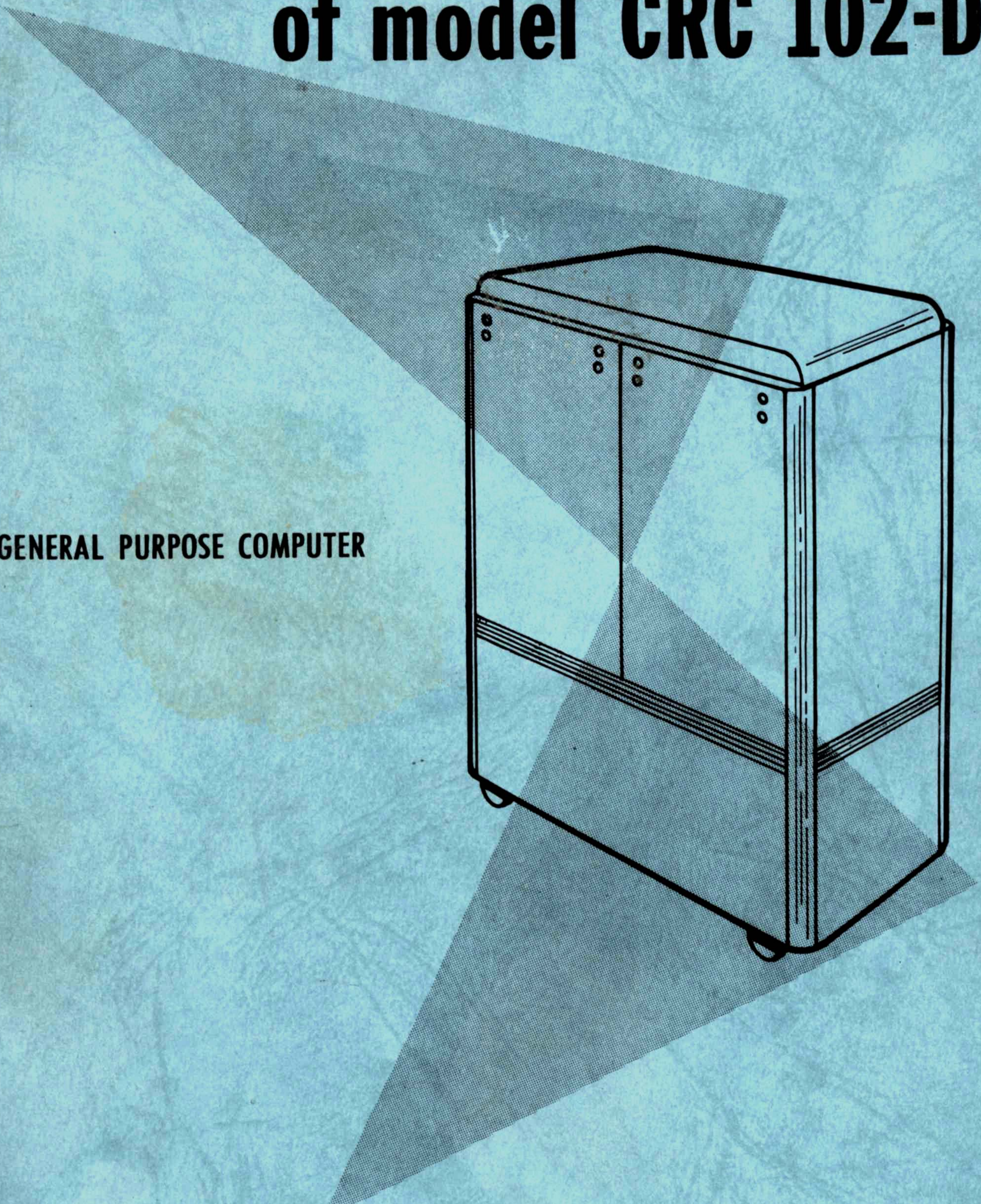
functional specifications of model CRC 102-D



GENERAL PURPOSE COMPUTER

The National Cash Register Company
ELECTRONICS DIVISION
3348 WEST EL SEGUNDO BOULEVARD • HAWTHORNE, CALIFORNIA • OSborne 5-1171

FUNCTIONAL SPECIFICATIONS

OF

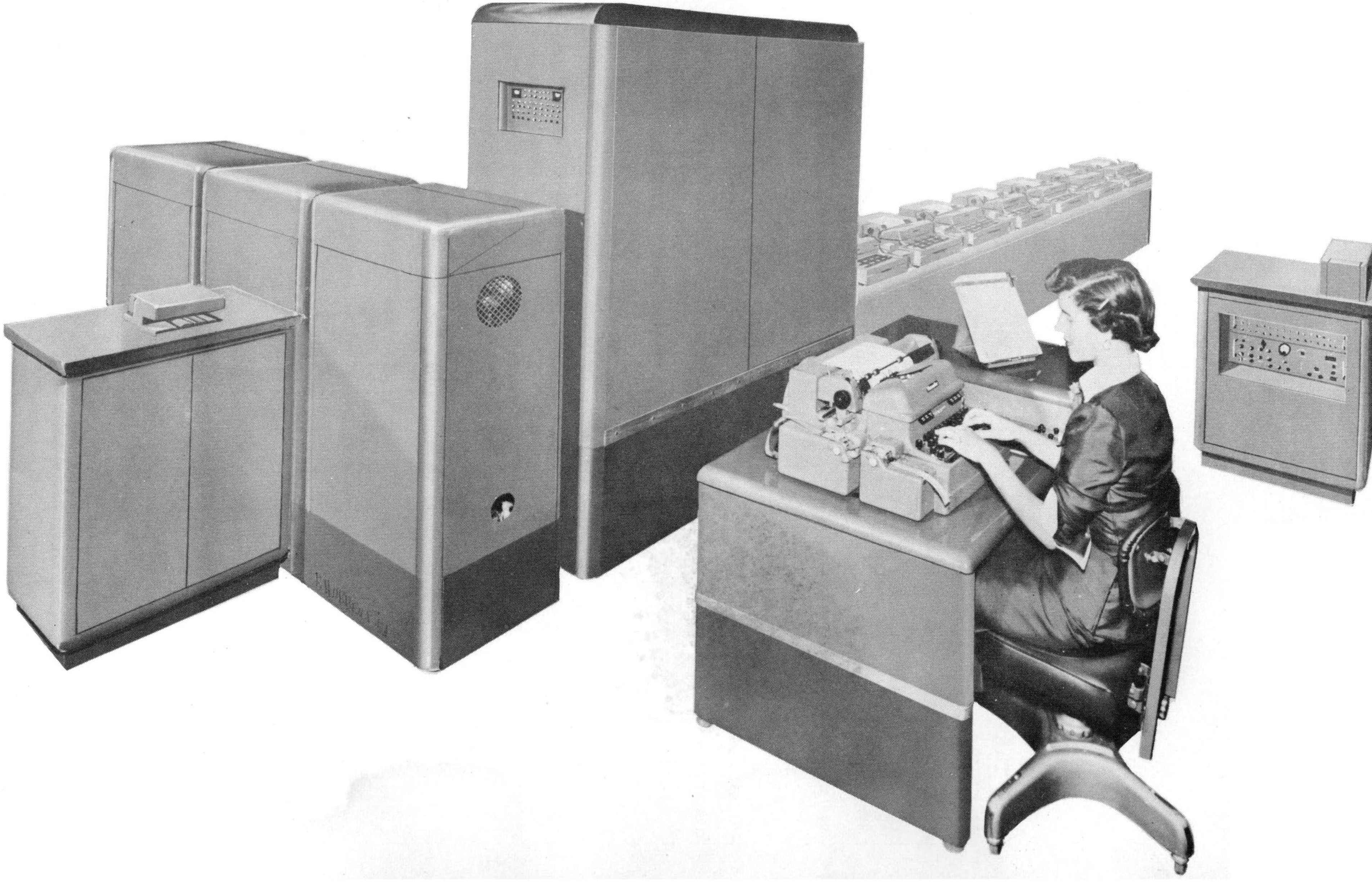
MODEL CRC 102-D

FOREWORD

The material presented herein described the functional characteristics of The National Cash Register Company's electronic data processor, the CRC 102-D. A knowledge of this material is essential for those who prepare programs for the computer. This report has been carefully prepared and is, at the time of this writing, complete and accurate.

A word of acknowledgement is in order. The manuscript was prepared by Mr. B. F. Handy with the cooperation of the other members of the Applications Department and the members of the Product Development Department and the Project Leader of the CRC 102-D project.

A. D. Hestenes
Director, Applications Department



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

TABLE OF CONTENTS

VII (Cont'd)

Page No.

I

INTRODUCTION

Description of the CRC 102-D System

The CRC 102-D Computer 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-D Computer

This machine is the outgrowth of the prototype model 102 developed and built by the NCR Electronics Division. It is a magnetic drum, serial computer housed in a single cabinet, complete with power supply, all logical elements and air-conditioning equipment. It will perform decimal arithmetic on binary-coded decimal numbers.

The CRC 102-D Console

This is a desk of a conventional design into which has been built the operator's console. This console consists of six pushbuttons, seven toggle switches, and seven indicator lights. A Flexowriter, mounted on the desk, is electrically connected to the computer so that signals from either the keyboard or from paper tape 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 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 tape will be explained in Section VIII.

The Ferranti High Speed Tape Reader

This tape reading unit will operate in the same manner as the Flexowriter tape reader. Information will be read from Flexowriter tape under control of the computer.

The Teletype High Speed Tape Punch

This tape punch unit will operate in the same manner as does the Flexowriter typewriter. Information will be punched on Flexowriter tape and may then be fed through a Flexowriter and printed. The printing will not be done during computer operation time.

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-D

The Number System of the CRC 102-D

The fundamental unit of information in the CRC 102-D is defined as a word. The CRC 102-D is a stored program machine, and hence, the word may be a command or a number. Any word, whether a command or a number, is represented inside the computer as a set of 42 binary digits (i. e., O's and 1's. See Appendix B for a complete discussion of binary and octal number systems).

For commands this set of 42 binary digits is represented outside the computer by 14 octal digits, each of which consists of three adjacent binary digits. Table I gives the eight possible groupings of the three binary digits and the corresponding octal digits which represent them.

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

TABLE I

For numbers this set of 42 binary digits is divided into one group of six binary digits which is called the sign section, and into nine groups of four binary digits each called the magnitude section. Each of the nine groups in the magnitude section is a binary-coded decimal digit and may be represented outside the computer as a decimal number. Table II represents the possible groupings and their equivalent decimal digit representation.

The fundamental arithmetic of the CRC 102-D is in the decimal number system. This eliminates the necessity for conversion of data from the decimal to the binary number system and vice versa.

Binary Group	Decimal Digit
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9

TABLE II

NOTE

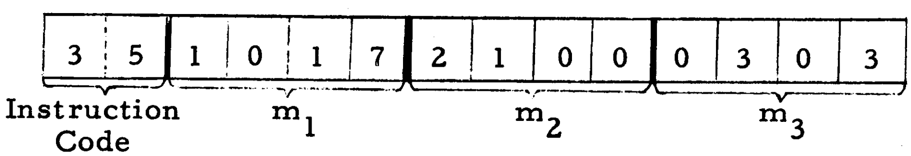
In Table II all possible binary configurations have not been used. The following have no numerical significance but are used for editing purposes when printing results from the computer:

1100	Tab
1101	Space
1110	Period

In addition the configurations 1010, 1011, and 1111 are possible combinations but have no meaning in the computer. If these configurations are in a word they will have no effect on the output. The computer will simply ignore them.

Commands in the CRC 102-D

The CRC 102-D Computer System employs a three-address instruction code. Normally a command consists of an instruction and three memory location references. As was stated in Section II, page 3, commands are represented by fourteen octal digits. The first two octal digits represent the instruction. (For a complete list of the available instructions and their two octal digit representation, see Table III.) The remaining twelve octal digits are divided into three sets of four octal digits each, usually referred to as addresses, and the addresses are labeled m_1, m_2, m_3 respectively in order of their significance.

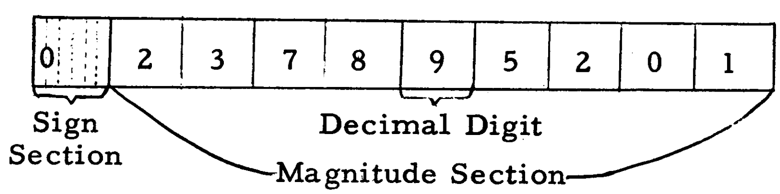


Command in the CRC 102-D

For exact descriptions of the various commands and their use, see Sections VI through X.

Numerical Data in the CRC 102-D

Numbers are usually presented in the decimal notation. The CRC 102-D is built to perform the arithmetic operations with decimal numbers directly without need for conversion. When presented to the computer a number will have two sections: a) A sign section which contains a sign bit and an overflow bit, and b) A magnitude section which contains nine binary-coded decimal digits.



Decimal Number in the CRC 102-D

Instructions in the CRC 102-D

Octal Code	Literal Abbreviation	Function
00 - 03	--	Numbers - cause alarm
04	bo	Buffer Out
05	bl	Buffer Load
06	rc	Read Card
07	sb	Subtract Binary
10	ab	Add Binary
11	fl	Fill
12	pd	Punch Decimal
13	po	Punch Octal
14	bs	Block Search
15	wt	Write Tape
16	rt	Read Tape
17	ts	Test Switch or Test Search
20	pl**	Plot**-cause alarm*
21	pr	Print
22	ht	Halt
23	dr	Divide Decimally and Round
24	dd	Divide Decimally and Save Remainder
25	mr	Multiply Decimally and Round
26	md	Multiply Decimally 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 Decimally
36	sd	Subtract Decimally
37	tb	Test Bit

*when used as a command

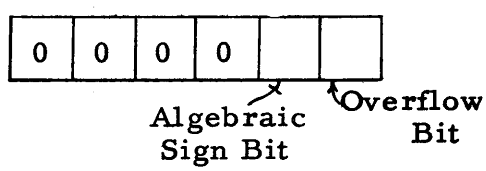
** A Plot command is provided as a special feature when an automatic digital point plotter is to be used with the CRC 102-D.

TABLE III

The sign section consists of six binary digits. The two most significant are always zero, and the four least significant are called the sign digits.

The fifth digit of the sign section indicates the algebraic sign of the magnitude. If it is a zero, the number in the magnitude section is positive; if it is a one, the number is negative.

The sixth digit of the sign section is an overflow marker. When it is zero it indicates no overflow; when it is one it indicates an overflow. It may be filled initially with the magnitude as a special marker to be identified later by the TEST BIT command. 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. The decimal code for the sign section is in Table IV.



Decimal Sign Section

Decimal Sign Section Code

Decimal Sign Fill	Sign Digit Printing	Binary Equivalent	Interpretation
+, 0, Space	Space	00 0000	Positive, no o. f.
-, 2	-	00 0010	Negative, no o. f.
1	p	00 0001	Positive, o. f.
3	n	00 0011	Negative, o. f.

TABLE IV

The magnitude section of the decimal number consists of nine decimal digits as described in Section II, Page 4 . In the CRC 102-D numbers will be treated as if the decimal point were at the extreme left. If numbers greater than unity are to be operated on,a scale factor must be used and care must be exercised to keep this in mind when performing the arithmetic operations. Any number of digits up to nine may be used, merely by inserting zeros in the remaining positions.

Special Codes

It is possible for the CRC 102-D to unscramble data offered to it in a form other than that described above. Such systems as floating point numbers, multiple precision numbers, relative commands and other special forms may be designed and interpreted by the computer by using special programs. For example, if the range of the numbers to be used is so great that single precision operations will not give enough precision or significance, then so-called floating-point operations must be used. In floating arithmetic, a number is scaled so that it lies between .1 and 1. The decimal scale factor is then stored as an integral part of the number, usually on one end of the magnitude section. The program must then separate the number from the scale factor and operate on the two separately, and then affix the proper scale factor to any results of arithmetic operations.

The combination of the binary and the decimal number systems offered in the CRC 102-D simultaneously gives the user the flexibility, the efficiency, the logical operations, and the convenience of the binary system; and the ease of use, the more familiar appearance, and the more easily checked results of the decimal system.

III

OPERATOR'S CONSOLE AND FLEXOWRITER

Operator's Console

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

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

Power Control

1) No. 1, a pushbutton labeled "ON", turns on all power to the computer system. The magnetic tape units, if the AUTO-MANUAL switch is on MANUAL, will not turn on. If on AUTO, they also will be turned on.

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

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

4) No. 4, a pushbutton labeled "OFF", turns off all power to the computer system, except magnetic tape units in the MANUAL mode of operation. If the "MEMORY-SAFE" switch is in the "ON" or "UP" position, this pushbutton is inoperable as far as the computer is concerned.

5) No. 5, a toggleswitch labeled "MEMORY SAFE", performs two functions. When it is in the "UP" or "ON" position, the computer cannot be turned off by means of the "OFF" pushbutton. When this toggleswitch is in the "DOWN" or "OFF" position, the computer may be turned off without impairment of the information on

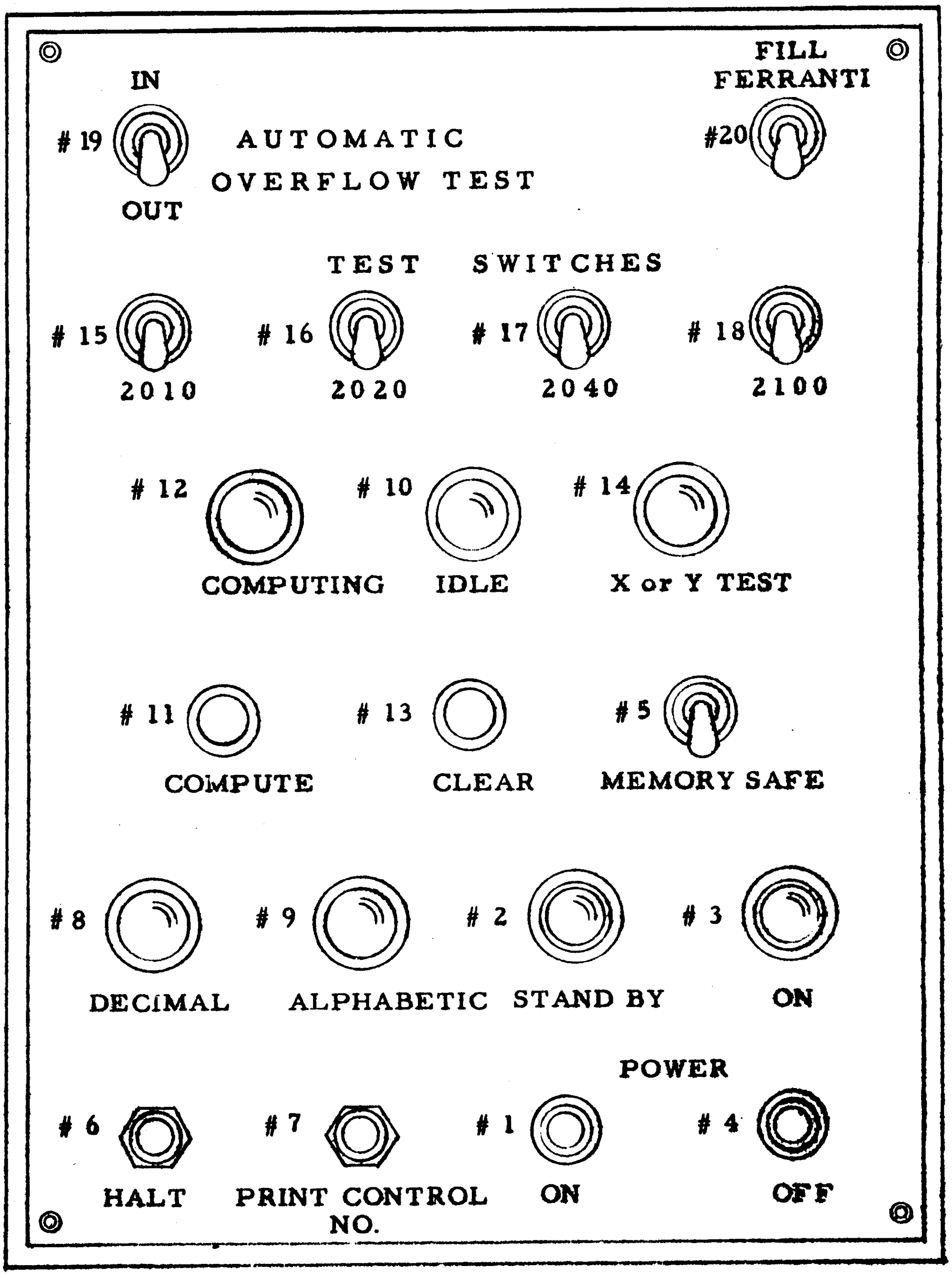
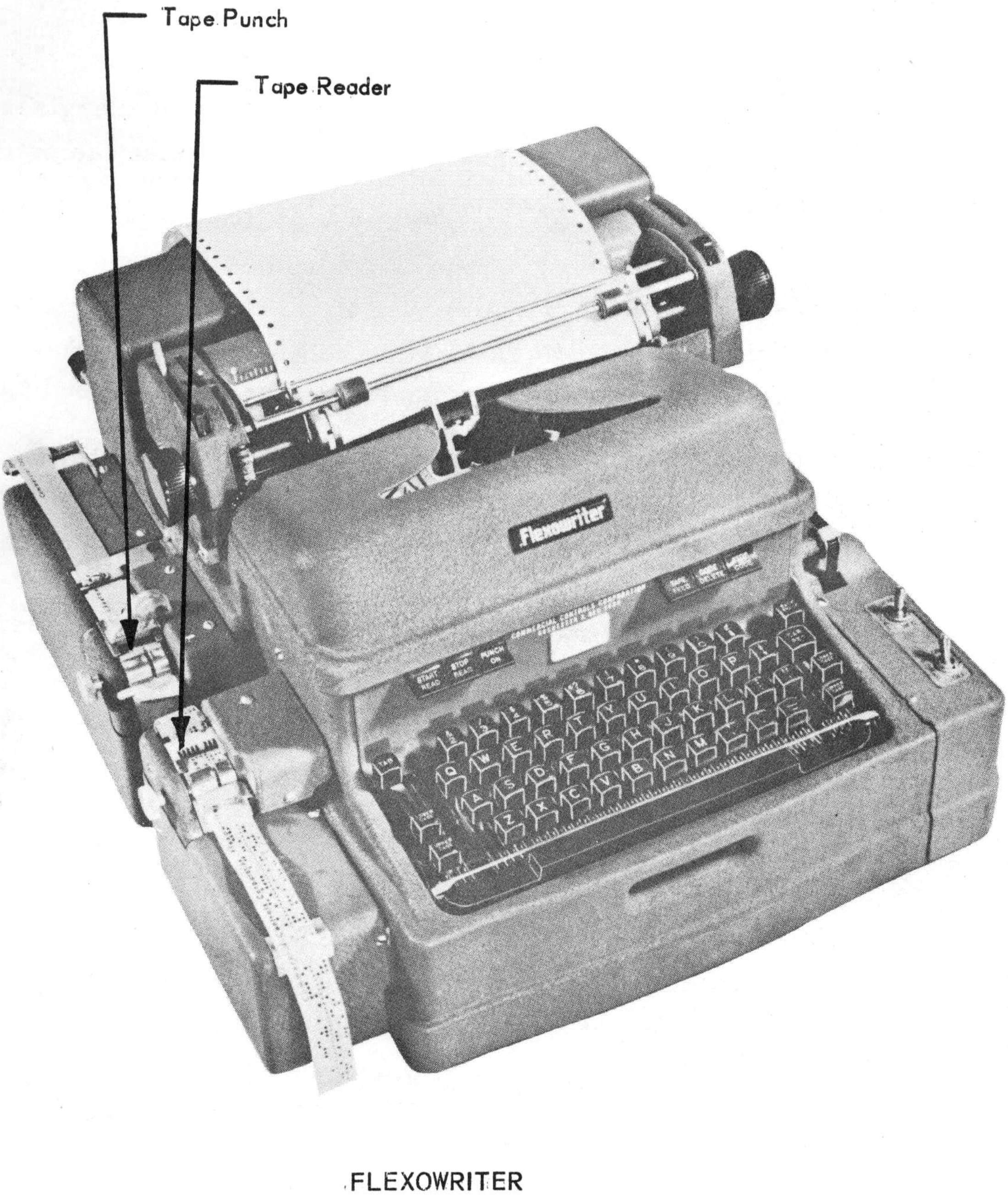


Figure I. Computer Control Panel



FLEXOWRITER

the magnetic drum. However, the "DOWN" position energizes the "CLEAR" circuitry (see pushbutton No. 13) so that computing may not be carried out correctly.

Computer Control

1) No. 6, a pushbutton labeled "HALT", stops computation at the end of a cycle and returns the computer to the "IDLE" condition.

2) No. 7, a pushbutton labeled "PRINT CONTROL NO. ", inhibits adding one to the control register, forces a skip to the error print-out circuitry, and prints the contents of the control register and then returns to the "IDLE" condition. To continue computation re-enter the control number as printed out to the H register and press the "COMPUTE" button.

3) No. 8, an indicator light labeled "DECIMAL", indicates that the computer is prepared to accept information in the decimal mode.

4) No. 9, an indicator light labeled "ALPHABETIC", indicates that the computer is prepared to accept information in the alphabetic mode.

If both indicator lights, No.8 and No.9 are out the computer is prepared to accept fill in the octal mode.

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

6) No. 11, a pushbutton labeled "COMPUTE", starts computation. The "COMPUTE" key on the Flexowriter serves the same purpose.

7) No. 12, an indicator light labeled "COMPUTING", indicates that the computation is in progress.

8) No. 13, a pushbutton marked "CLEAR", stops computation and prepares the computer to accept octal information. This button is not an operator halt. Computation may not be restarted where it was stopped unless the H register is reset. If the button is pressed during computation, possibly during execution of a command, the computer will halt immediately and will go to the "IDLE" condition without completing the normal command flow.

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

10) No. 20, a toggleswitch labeled "FILL FERRANTI", starts the Ferranti Reader.

Test Control

1) Nos. 15, 16, 17 and No. 18, switches labeled "TEST SWITCHES", and marked "2010", "2020", "2040", and "2100" are used with the TEST SWITCH COMMAND. For more details see Section V and Section VII.

2) No. 19, a toggleswitch labeled "AUTOMATIC OVERFLOW TEST", controls the overflow checking feature. When this switch is in the position labeled "OUT", the overflow checking feature is not in operation.

The Flexowriter

The Flexowriter is the primary input-output device for the CRC 102-D Computer System.

The Flexowriter keyboard appears as in Figure II. The "COMPUTER

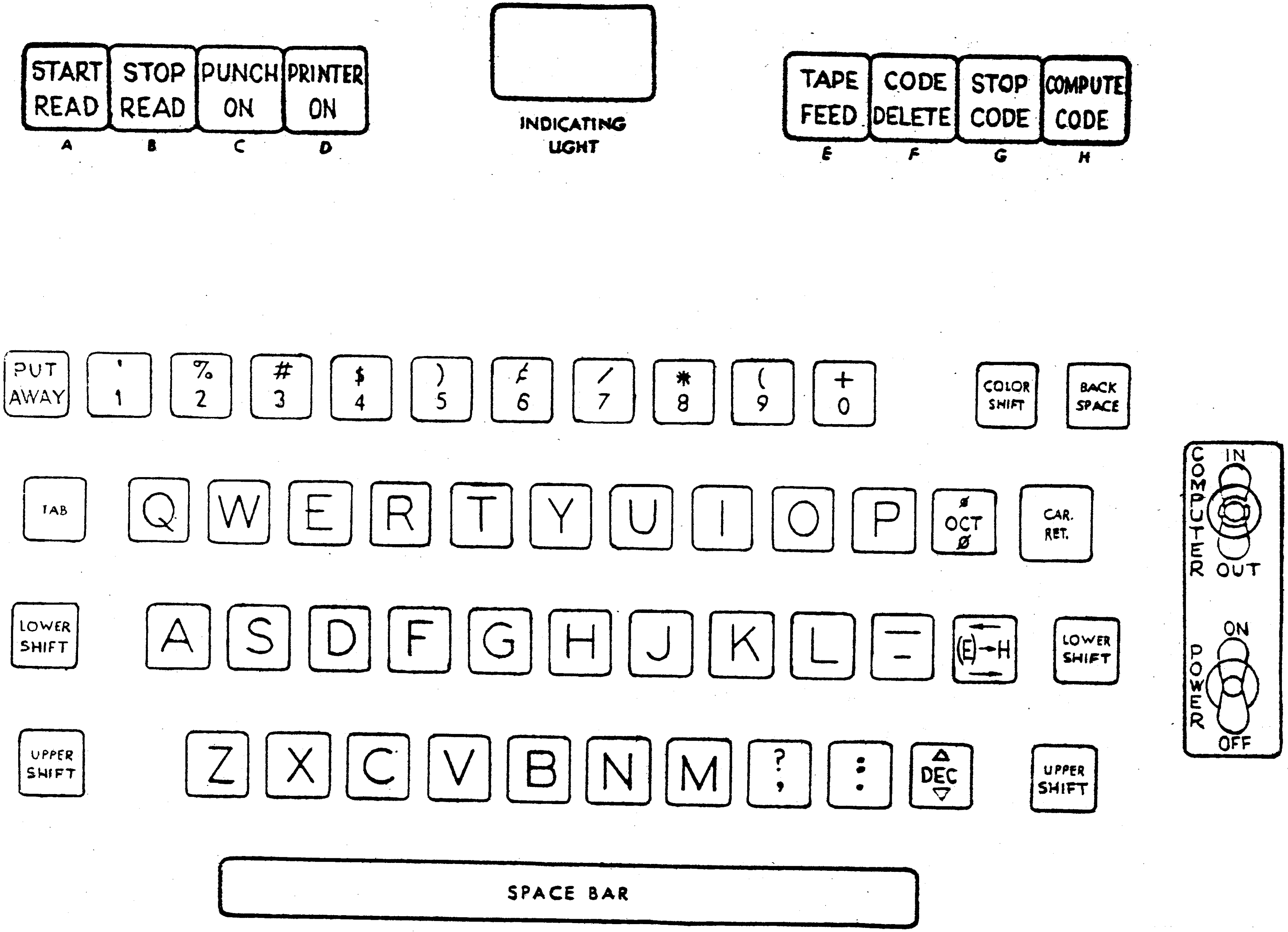


Figure II. CRC 102-D Flexowriter

IN-OUT" switch controls filling, printing and punching with the Flexowriter. The effect of this switch on the execution of FILL and PRINT commands is shown in Table V below:

Computer In	Print	Proceeds if Punch or Printer "ON"; otherwise waits.
	Fill (Flex)	Fills
Computer Out	Print	Light Appears, waits
	Fill (Flex)	Tape Reader starts; no filling occurs.

TABLE V

Along the top of the Flexowriter are eight buttons which perform the following functions:

1) "Start Read" starts the paper tape reader.

2) "Stop Read" stops the paper tape reader.

3) "Punch On" turns the paper tape punch on and controls the punching from tape reader, computer, or keyboard.

4) "Printer On" controls printing from the tape reader or from the computer. Printing from the keyboard will occur regardless of this button's position.

5) "Tape Feed" causes the paper tape in the Flexowriter tape punch to run out punching sprocket holes in the center of the tape.

6) "Code Delete" punches a configuration which, when the computer is filling automatically, causes the computer to ignore the punched tape.

7) "Stop Code" punches a configuration which, when the computer is filling automatically causes the Flexowriter to stop reading tape, until the START READ button is depressed again.

8) "Compute Code" punches a configuration which starts the computation when filling automatically, or when it is desired to begin computation from the Flexowriter.

The following keys on the Flexowriter keyboard perform control functions:

1)	Δ Dec ∇	Prepares the computer to accept decimal digits and clears the Input Register.
2)	ϕ Oct ϕ	Prepares the computer to accept octal digits and clears the Input Register.
3)	Put Away	Transfers a word from the Input Register to a preselected memory location
4)	$\leftarrow$ (E) $\rightarrow$ H $\rightarrow$	Transfers a word from the Input Register to the Control Register.
5)	A	Prepares the computer to accept alphabetic information.
6)	Z	Fills four octal or decimal zeros.

Flexowriter Code in the CRC 102-D

The CRC 102-D is capable of receiving and printing information in the decimal, octal, or alphabetic modes. The following tables give the binary representation of the Flexowriter keys when filling in the respective modes.

Decimal Mode

Flexowriter Key	Binary Configuration
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
Tab	1100
Space	1101
Period	1110
Code Delete	1111
Z	Fills four decimal or sixteen binary zeros

TABLE VI

None of the alphabetic keys except the "Z" key and the "A" key have any significance while filling in the decimal mode.

When using the alphabetic mode for filling or printing, each key on the Flexowriter keyboard is referred to by two octal digits. In this mode, then, each CRC 102-D word holds six characters, either alphabetic, numeric, or editorial. The alphabetic code may be entered into the computer directly from the Flexowriter typewriter by first striking the "A" key to set up the alphabetic control, then striking six characters then a tab, then six more characters and a tab, etc., until all the

Alphabetic Mode

Flexowriter Key	Octal Configuration	Flexowriter Key	Octal Configuration
0 +	00	A	46
1 ' ,	01	B	47
2	02	C	50
3 #	03	D	51
4 $	04	E	52
5)	05	F	53
6 ¢	06	G	54
7 /	07	H	55
8 *	10	I	56
9 (	11	J	57
Back Space	20	K	60
Color Shift	21	L	61
ϕ ϕ (Oct)	25	M	62
Δ ∇ (Dec)	26	N	63
Carriage Return	30	O	64
Stop Code	33	P	65
Tab	34	Q	66
Space	35	R	67
Period	36	S	70
Code Delete	37	T	71
, ?	41	U	72
- _	42	V	73
Upper Shift	43	W	74
Lower Shift	44	X	75
← → (E) → H)	45	Y	76
Put Away	24	Z	77

TABLE VII

alphabetic information is properly entered. Each character will be represented internally by two octal digits according to the above table.

Octal Mode

Flexowriter Key	Binary Configuration
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111
Z	Fills 12 binary zeros

TABLE VIII

None of the alphabetic keys except the "Z" key and the "A" key have any significance when filling in the octal mode.

IV

FILLING AND OPERATING THE CRC 102-D

Filling the Computer

The following instructions for filling the computer apply to both paper and manual filling. A paper tape must be provided with all the symbols which a human operator would use to enter the data. There are three filling modes on the CRC 102-D: decimal, octal, and alphabetic; 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.

1. The input Register, also called the "E" register, accepts and assembles individual digits directly from the Flexowriter or the Ferranti Reader. It should be clear before filling. Each new punch configuration or striking of the Flexowriter key causes the "E" register to shift its contents to the left. In the case of octal or decimal digits, this shift is one digit, (three or four binary positions respectively). In the case of an alphabetic character, this shift is two octal digits. The new character is then entered into the right-most position. For example, consider filling the decimal number 314160000.

   a. First the input register is cleared and set up for decimal input by striking the $\boxed{\Delta \text{ dec } \nabla}$ key.

0 0	0	0	0	0	0	0	0	0	0

   b. Strike the "3" key

0 0	0	0	0	0	0	0	0	0	3

c. Strike the "1" key

0 0	0	0	0	0	0	0	0	3	1

d. Strike the "4" key

0 0	0	0	0	0	0	0	3	1	4

e. Strike the "1" key

0 0	0	0	0	0	0	3	1	4	1

f. Strike the "6" key

0 0	0	0	0	0	3	1	4	1	6

g. Strike the "Z" key

0 0	3	1	4	1	6	0	0	0	0

2. The Control Register, also called the "H" 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 portion is the address of the next command in order, and the m_3 portion 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 0735, or if computation were started, the computer would take its first command from cell 1642.

Information entered into the Input Register may be transferred to the Control Register by striking the $\boxed{\overleftarrow{\ }\ (E)\rightarrow H\ \overrightarrow{\ }}$ key.

Information entered into the Input Register may be transferred to the proper Main Memory location by hitting the "Put Away" key.

The complete filling process proceeds as follows:

1. Select the octal mode in order to enter the address of the first cell to be filled into the control register.

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

   b. The "CLEAR" button on the console sets the computer to fill in octal.

   c. The computer may be set to fill octal specifically by striking the $\boxed{\phi\ \text{oct}\ \phi}$ key.

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

   a. If the control register has been cleared, input will automatically start at cell 0000 unless the Control Register is modified.

   b. If the computer has stopped at a HALT or FILL command, the m_3 portion of those commands 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, type the desired octal address and then strike the $\boxed{\overleftarrow{\ }\ (E)\rightarrow H\ \overrightarrow{\ }}$ key on the Flexowriter.

3. Select the filling mode necessary for the data to be entered. Once a mode is selected it is preserved until changed.

a. Commands must be entered in the octal mode. Strike the $\boxed{\begin{array}{c}\phi\\ \text{oct}\\ \phi\end{array}}$ key before entering octal information.

b. Decimal data is entered in the decimal mode. Strike the $\boxed{\begin{array}{c}\Delta\\ \text{dec}\\ \nabla\end{array}}$ key before entering decimal information.

c. Alphabetic data is entered in the alphabetic mode. Selection of this mode is accomplished by striking the "A" key.

4. Enter data.

a. Commands - octal system
Enter the two digits identifying the command. Enter the three addresses.

b. Decimal numbers
Enter the sign. If the sign is positive without overflow, it need not be entered. If any of the other signs (see TABLE IX) are intended, the sign digit must be entered. Table IX outlines the fill for the possible sign configurations.

Desired Sign	Fill
Positive, no overflow	Nothing, or 0 (zero), or space
Negative, no overflow	- (Minus) or 2
Positive, overflow	1
Negative, overflow	3

TABLE IX

Enter the 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.

c. Alphabetic information
Enter six characters only to magnitude section, including any editorial characters such as shift up, shift down, space, tab, etc. Since each character is represented by two octal digits, six characters fill the magnitude section of one word.

5. Transfer the word to the designated location.

When the word has been typed into the Input Register, it can be transferred to the previously selected destination cell by hitting the "Put Away" key. This moves the typewriter carriage just as a tab does, 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.

6. In the Case of an Error.

An error before the "Put Away" key is struck can be corrected by entering the entire correct command or number, always including the sign digit and all left most zeros, and then striking the "Put Away" key, or by clearing the Input Register and entering the correct number. An error detected after "Put Away" can be corrected by selecting the octal mode and setting the Control Register to the address of the cell where the incorrect data was entered and then reentering the corrected data.

7. 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 outlined above.

Operating the Computer

The following are the rules to be obeyed when operating the computer:

1. Set the "MEMORY SAFE" switch in the OFF or DOWN position, and then turn computer on with pushbutton #1. Wait until indicator or light #3 comes on, then turn "MEMORY SAFE" switch to ON.

2. Clear computer with pushbutton #13.

3. Enter commands and data according to the instructions in the section immediately preceding (pages 21 through 25).

4. Select address of the first instruction:

   a. Prepare the computer to accept octal numbers by striking the [ϕ oct ϕ] key. To specify a particular starting address to the control register, type the desired address followed by striking the "Z" key on the Flexowriter. Then hit the [← (E)→H →] key on the Flexowriter.

   b. If the Control Register was cleared just previously, computation will start with cell 0000 if the control is not modified as outlined in (a) above.

   c. If the computer has been halted by a HALT or FILL command, computation will restart at the command specified by the HALT or FILL command (the next command).

5. Start computation by depressing the "Compute" button on the Flexowriter or by depressing pushbutton #11 on the console. (See Figure I and Figure II).

6. Program control will bring the computer to a stop after the process of computation is over. If the operator wishes to halt before

the end of the computing cycle, he can depress pushbutton #8 on the console (see Figure I page 9). To halt the computation and to examine the control register setting, push button #6 marked "PRINT CONTROL NO." on the computer console is depressed. To restart enter the number into the control register which was typed out and press the "Compute" button.

7. To turn off the computer, first place "MEMORY SAFE" switch in OFF position, then depress pushbutton #4.

Further operating instructions for the entire system are included in the sections on TEST SWITCH command, Magnetic Tape Unit description, IBM and Flexowriter operation.

V

MEMORY OF THE CRC 102-D

General Description

The CRC 102-D has a magnetic drum memory. The drum rotates on a vertical axis at approximately 40 rps (revolutions per second) 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 circumference 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.

The 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 restrictions on the respective positions of commands or data. The main memory can be filled directly: either manually, or from paper tape, or from the magnetic tape.

The Buffer Register

The buffer register is an 8 word delay line synchronized with the least significant digits of the word channels. It is customarily referred to by the addresses 2000 through 2007, but other addresses may also refer to it (see Table X and XI for their structure). The buffer 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 read out onto magnetic tape or to punch cards, the buffer register is first loaded from the main memory and then the external unit is read to from the buffer register. The buffer register may be filled either manually or from the paper tape.

The buffer register may also be used as an eight word extension of the main memory. The access time to any cell in it is a maximum of one-eighth of a drum revolution; whereas it is one drum revolution in the main memory.

Cell 2100

Cell 2100 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 ever contain the results of an operation. Other addresses may also refer to cell 2100; see the table at the end of this section for their structure.

Cell 3000

This cell is provided as a convenience for use with relative coding 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, Table III, page 6) bo, bl, cr, pd, bs, dr, dd, sl, sm, sf, ex, or a test command to, ta, tm, or ts, when the test does not work then cell 3000 will contain:

a. Zeros in the two octal instruction digits.

b. In m_1, the m_1 portion of the previous 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 previous command.

2. If the command preceding the command which refers to cell 3000 was mr or md, cell 3000 will contain:

a. 37 in the two octal instruction digits.

3. 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 address 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, zeros except the least significant octal digit will equal the most significant octal digit of the m_1 of the test command which worked.

4. If the command which refers to cell 3000 is the starting command immediately after filling, and assuming that the computer has halted

with a fl or ht command and that during the filling operation the $\boxed{(E) \rightarrow H}$ key was <u>not</u> struck, nor read from tape, then cell 3000 will contain:

a. Zeros in the two octal instruction digits.

b. In m_1, the m_1 portion of the fl or ht 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.

5. If the command which refers to cell 3000 is the starting command immediately after filling and the $\boxed{(E) \rightarrow H}$ key was struck or 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 $\boxed{(E) \rightarrow H}$ key was struck or read from tape.

b. In m_1, the m_1 portion of the word entered just before the last $\boxed{(E) \rightarrow H}$ key 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 $\boxed{(E) \rightarrow H}$ key was struck or read from tape after the last cell was filled, in which case m_3, will contain the m_3 portion of the word entered just before the last $\boxed{(E) \rightarrow H}$ key was struck or read from tape.

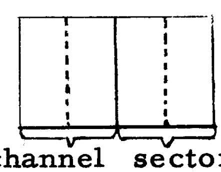
When any of the three commands rt, wt, or pr precede a reference to cell 3000, the contents of cell 3000 is determinable in a fashion distinct from any other command. The information in it is derived directly from the command and does not provide the programmer with useful data. Therefore, its forms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

Test Switches

The 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 those addresses have different meanings when referred to by any other command. Associated with the Test Switch command is another address - 2400. This refers to the magnetic tape units; see the TEST SEARCH command for the significance of this.

Interpretation of Address

Each cell in the internal memory of the computer has a unique address defined by four octal digits. The two most significant digits specify the particular channel in which the cell lies, and the two least significant digits select the sector.



Memory Address

Reference to the memory for both channel and sector selection is distinct in the two operations of reading and writing. As an example cells 2100 and 3000 may be read from but not written into. Any reference to these two cells for writing will result in cell 2000 being written into.

The computer selects the channel to be addressed by considering the two channel digits of the address. Table X shows the channel the computer will select when reading from the memory; Table XI shows the channel the computer will select when writing into the memory.

(Reading from Memory)

Channel Digits are:	Computer will select
00 - 17 or 40 - 57	Main Memory Channels 00 - 17
20, 22, 24, 26 or *60, 62, 64, 66	Buffer Register Channel 20
21, 23, 25, 27 or *61, 63, 65, 67	Cell 2100
30 - 37 or *70 - 77	Cell 3000

TABLE X

The Computer selects the proper sector by comparing the sector digits with the word channel. When the sector digits are exactly equal

*Adding the octal number, 40, to the channel has no effect on the computer channel selection. This fact may be useful when coding in a relative notation.

(Writing into Memory)

Channel Digits are	Computer will select
00 - 17 or 40 - 57	Main Memory Channels 00 - 17
*20 - 37 or 60 - 77	Buffer Register Channel 20

TABLE XI

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 both sector digits for reference to the main memory. For reference to the Buffer Register the comparison takes place with respect to the least significant octal digit of the address only.

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 Section XII.

*The least significant digit in the channel selection has no significance when the Buffer Register is desired.

VI

ARITHMETIC COMMANDS

The arithmetic commands are used to perform the bulk of all 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 binary 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.

Name and octal code digits

1. Abbreviation and address structure.
2. Equivalent algebraic and overflow sign when command is operated on as a number by the ADD binary or SUBTRACT binary commands.
3. 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.
4. Description of the function of the command.
5. Examples where required.

ADD DECIMALLY 35

1) ad m_1 m_2 m_3

2) Positive, overflow

3) 7.8 milliseconds

4) Obtains the algebraic sum of two binary-coded decimal numbers stored in m_1 and m_2 as a binary-coded decimal number and stores it in m_3. If the magnitude of the sum requires more than 9 decimal digits an overflow digit is recorded in the overflow bit of the sign section of m_3. When an overflow digit is generated by this command, the computer will cause an overflow alarm to be printed out unless the next command is a Test Bit or Shift Logically command. (This feature may be suppressed by placing the console switch #19, marked "Automatic Overflow Test" in the "Out" position. c.f. Section III, page 15). If an overflow bit existed in the m_1 before the command was obeyed, an overflow marker will record in m_3 but this will not cause an overflow alarm unless the sum of the numbers in m_1 and m_2 exceeds nine decimal digits. An overflow bit in the sign section of the number in m_2 is not recorded in m_3.

No check is made to verify that the numbers in m_1 and m_2 are coded decimal numbers. If they are not, the results will be meaningless.

5) Examples

(m_1) + 500000000	(m_1) p 314159265	(m_1) + 271828183
(m_2) + 700000000	(m_2) + 271828183	(m_2) p 314159265
(m_3) p 200000000	(m_3) p 585987448	(m_3) + 585987448

SUBTRACT DECIMALLY 36

1) sd m_1 m_2 m_3
2) Negative, Overflow
3) 7.8 milliseconds
4) Obtains the algebraic difference of two binary coded decimal numbers stored in m_1 and m_2 and stores the binary coded decimal result in m_3 by changing the algebraic sign of the number in m_2 and adding. See the ADD DECIMALLY command on previous page.

MULTIPLY AND ROUND DECIMALLY 25

1) mr m_1 m_2 m_3
2) Positive, overflow
3) 21.1 milliseconds minimum; 49.1 milliseconds maximum. The time for execution depends on the digits of the multiplier. The time in milliseconds may be computed by use of the formulae

$$t = 17.55 + .39 \; (\Sigma \, d_i + n)$$

where $\Sigma \, d_i$ is the sum of the digits of the multiplier and n is the number of zeros in the multiplier.

4) Obtains the algebraic product of two binary coded decimal numbers stored in m_1 and m_2 as an 18 digit binary coded decimal number. This result is rounded off to 9 binary coded decimal digits and stored in m_3 with proper algebraic sign.

MULTIPLY DOUBLE LENGTH DECIMALLY 26

1) md m_1 m_2 m_3

2) Negative, no overflow

3) The execution time is the same as for the MULTIPLY and ROUND command. (see MULTIPLY and ROUND DECIMALLY command above).

4) Obtains the algebraic product of two binary coded decimal numbers stored in m_1 and m_2 as an 18 digit binary coded decimal number. The most significant half is stored in m_3, the least significant half is stored on the next cell adjacent to m_3 on the magnetic drum. Each product part has the algebraic sign of the product.

DIVIDE AND ROUND DECIMALLY 23

1) dr m_1 m_2 m_3

2) Negative, overflow

3) 23.0 milliseconds minimum; 54.2 milliseconds, maximum. The time for execution depends on the digits of the quotient. The time in milliseconds may be computed by use of the formulae:

$$t = 18.11 + .39 (\Sigma d_i + n)$$

where Σd_i is the sum of the digits in the ten digit quotient before round off and n is the number of zeros in the ten digit quotient before round-off.

4) Obtains the algebraic quotient of two binary coded decimal numbers, the divisor stored in m_1 and the dividend stored in m_2 as a 10 digit binary coded decimal number. This result is rounded off to 9 binary coded decimal digits and

stored in m_3. This command has no regard for arbitrarily assigned scale factors; it assumes that the decimal point of the numbers to be operated on is to the extreme left of the magnitude.

If the absolute quotient is greater than one, it is meaningless in this computer. However, when the quotient is greater than one, a one is put in the overflow bit of the sign section of m_3 and the computer will cause an overflow alarm unless the next command is TEST BIT or SHIFT LOGICALLY.

DIVIDE DECIMALLY AND SAVE REMAINDER 24

1) dd m_1 m_2 m_3
2) Positive, no overflow
3) 21.1 milliseconds, minimum; 49.1 millisecond, maximum. The time for execution depends on the digits of the quotient. The time in milliseconds may be computed by use of the formulae:

$$t = 17.6 + .39\ (\Sigma d_i + n)$$

where Σd_i is the sum of the digits in the quotient and n is the number of zeros in the quotient.

4) Obtains the algebraic quotient of two binary coded decimal numbers, the divisor stored in m_1 and the dividend stored in m_2, as a nine digit binary coded decimal number. The nine digit binary coded decimal remainder is stored in m_3 and the quotient is stored in the next cell adjacent to m_3 on the magnetic drum.

The overflow feature of this command is the same as for the DIVIDE AND ROUND command above.

SCALE FACTOR DECIMALLY 31

1) sf m_1 m_2 m_3

2) Positive, overflow

3) 9.8 milliseconds, basic time, plus 1.95 milliseconds for each decimal shift.

4) Shifts the binary coded decimal number stored in m_2 to the left until the first non-zero binary coded decimal digit is in the most significant digit position. If the number in m_2 is zero, does not shift.

The number of decimal shifts multiplied by 4 and expressed as an octal integer is subtracted from the number stored in m_1 and the result is stored in m_3. If the number in m_2 is zero, 64 (Octal 100) is subtracted from the number in m_1.

The shifted number is stored in the next cell adjacent to m_3 on the magnetic drum surface.

5) Example

+006324159	The original decimal number stored in m_2
+0000000000044	The original octal scale factor stored in m_1
+632415900	The shifted result stored in the cell adjacent to m_3
+0000000000034	The new octal scale factor stored in m_3

ADD BINARY

1) ad m_1 m_2 m_3

2) Negative, overflow

3) 10.8 milliseconds to add two binary numbers; 9.8 milliseconds to add two commands.

4) Obtains the sum of two binary numbers stored in m_1 and m_2 and records the result in m_3. If the magnitude of the sum requires more than thirty-six digits to be represented in the computer, an overflow bit is recorded in the sign overflow position and the computer causes an overflow alarm to print out unless the next command is a TEST BIT or a SHIFT LOGICALLY command. If an overflow bit existed in the number stored in m_1 before the addition, and overflow bit will be recorded in m_3 without causing an overflow alarm. An overflow bit in m_2 has no effect on the sum.

If m_1 is a command the computer adds the absolute value of the number in the magnitude section of m_1 to the algebraic value of the number in m_2 and records the sum in m_3. The two octal digits identifying the command in the number 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.

5) Examples; (Octal Notation)

Contents of		Contents of	
m_1	36 0364 0277 1630	m_1	36 0364 0277 1630
+ m_2	00 0001 0010 0054	m_2	00 7413 7500 6160
m_3	36 0365 0307 1704	* m_3	36 0000 0000 0010

*No overflow generated, since (m_1) is a command

SUBTRACT BINARY 10

1) sb m_1 m_2 m_3

2) Positive, no overflow

3) 10.8 milliseconds to subtract two binary numbers; 9.8 to subtract a number from a command.

4) The SUBTRACT BINARY command operates exactly like the ADD BINARY command except that the algebraic sign of the number in m_2 is inverted before the addition takes place. (see the ADD BINARY Command)

SHIFT MAGNITUDE 30

1) sm m_1 m_2 m_3

2) Positive, no overflow.

3) 8.6 milliseconds, basic time plus 0.78 milliseconds for each shift.

4) Shift the magnitude 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.

   (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.

5) 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

SHIFT LOGICALLY 27

1) sl m_1 m_2 m_3

2) Negative, overflow.

3) 9.0 milliseconds, basic time plus 0.78 milliseconds for each shift.

4) 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 3. See example.

   (m_2) is interpreted in SHIFT LOGICALLY exactly as in SHIFT MAGNITUDE.

5) 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 6 has been recorded as octal number 2 because of the restriction of the binary digit at the left end of the word.

VII

LOGICAL AND TRANSFER COMMANDS

EXTRACT 32

1) ex m_1 m_2 m_3

2) Negative, no overflow

3) 32.4 milliseconds if m_3 is in main memory; 10.6 milliseconds if m_3 is in buffer register.

4) The extract command moves 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 inserted into the corresponding positions of m_3. Where there are binary "zeros" in (m_2) the corresponding positions of (m_3) remain unchanged.

5) Examples:

   (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

   (m_1) 10723635214263

   (m_2) 30236241253637

   m_3 = 2103

   Result in 2003 10222201210223

TEST BIT 37

1) tb m_1 m_2 m_3

2) Negative, overflow

3) 6.6 milliseconds when test works

4) Tests whether there is a binary one in the word in m_1 for every binary one in the word in m_2. If the test works, the next command is taken from m_3. If the test fails, i.e. at least one binary position of m_1 does not have a one when there is a one in that position of m_2, the computer proceeds normally to the next command.

This test may be used to test for overflow by putting in m_2 a number which has a one only in the overflow bit. All the other bits of this word must be zero.

5) Examples:

The command is in 0300:

```
        tb 1001 1005 0306
a. 1001 = 36 5217 0321 7982
   1005 = 00 4010 0100 1002
```

Control will go to 0306 for the next command:

```
b. 1001 = 36 5207 0321 7682
   1005 = 00 4010 0100 1002
```

Control will proceed normally to 0301 for the next command:

```
c. 1001 = 01 3265 4217 6135
   1005 = 01 0000 0000 0000
```

Control will jump to 0306 for the next command because of overflow.

TEST MAGNITUDE

1) tm m_1 m_2 m_3

2) Positive, no overflow.

3) 7. 8 milliseconds when test works.

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

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

TEST ALGEBRAICALLY 33

1) ta m_1 m_2 m_3

2) Negative, overflow.

3) 8. 2 milliseconds when test works.

4) 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.

TEST SWITCH or TEST SEARCH (See Section VII) 17

1) ts m_1 3000 m_3

2) Negative, overflow.

3) 6. 3 milliseconds when test works.

4) 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.

If m_1 is 2400, 2500, 2600, or 2700, and any Magnetic Tape Unit attached to the computer is block searching (see Section VII) take the next command from cell m_3, if no Magnetic Tape Unit is searching, proceed normally.

HALT 22

1) ht 3000 3000 m_3
2) Negative, no overflow.
3) 9.3 milliseconds (meaningless).
4) Stop computation and return machine to idle. Prepare computer to accept octal. Leave control register prepared to place the first word entered from the Flexowriter in the normal filling operation into cell m_3.

BUFFER OUT 04

1) bo 3000 3000 m_3
2) Positive, no overflow.
3) 10.2 milliseconds.
4) 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 in the buffer cell which 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 the contents into the main memory. The Buffer cells are treated in a cyclic manner, with cell 2000 considered as following cell 2007.

BUFFER LOAD 05

1) bl 3000 3000 m_3
2) Positive, overflow.
3) 9.8 milliseconds.
4) 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 are lost. The Buffer Register may be cleared with the command: bl 3000 3000 2100.

VIII

MAGNETIC TAPE OPERATION

Magnetic Tape Unit Model CRC 126

The CRC 102-D 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-D for this purpose. The CRC 102-D is capable of having seven such tape handling units connected to it at any one time. Each individual unit handles one tape at a time, but tapes may be manually changed 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 lines on the plugs are common to all of the tape units The computer does not have separate wires 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 particular configuration to energize a relay. The recognizing mechanism may be changed in a tape unit by the repositioning of only three crystal diodes.

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, page 54.

The various controls accomplish the following:

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

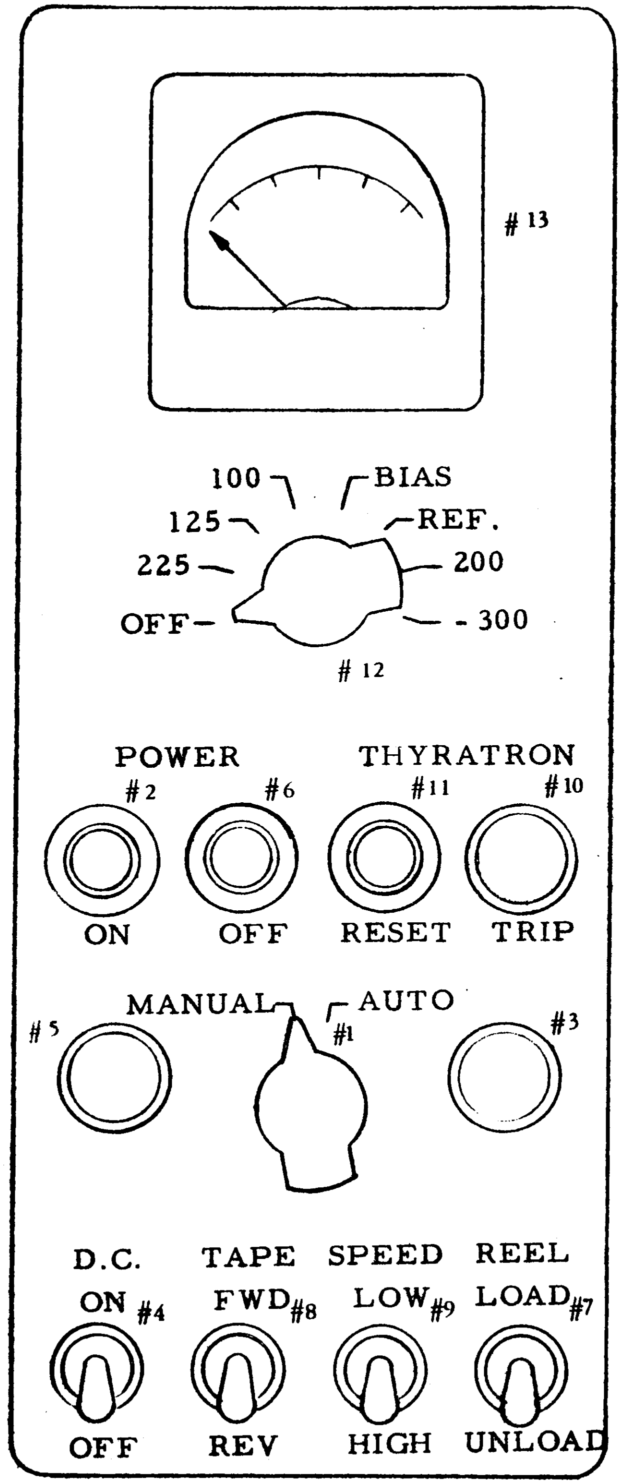


Figure III. Tape Unit Control Panel

marked "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.

2) Power control in both AUTO and MANUAL states.

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

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

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

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

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

3) Tape Control for manual operation.

   a) 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.

   b) Toggle switch No. 8 marked "TAPE" is a three position 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 "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.

c) Toggle switch No. 9 marked "SPEED" is a two position switch which controls the moving speed of the tape. 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 ninety inches per second. This switch can be used only when switch No. 1 is on "MANUAL".

4) Testing Controls - For Use by Maintenance Man.

a) 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.

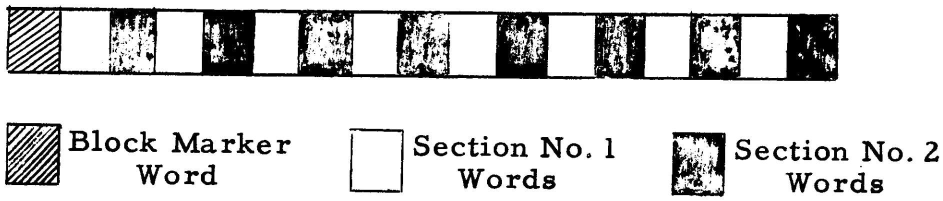
b) Push button No. 11 will reset DC voltage alarm and permit the DC voltage to come back on.

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

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

The magnetic tapes used with the 126 Magnetic Tape Unit are manufactured by the Minnesota Mining Corporation. They consist of a thin transparent plastic strip, one inch wide, twelve hundred 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 reels 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).

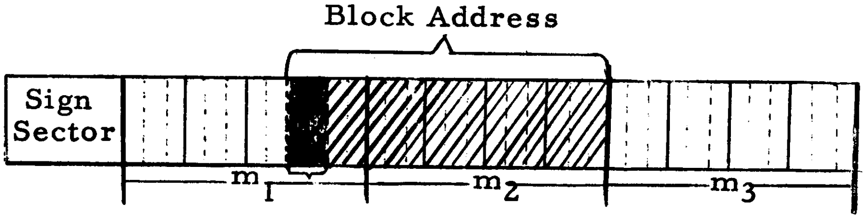
Data is arranged on the magnetic tape in blocks of seventeen words, in a physical space on the tape of 2.04 inches per block. Each block is divided into three sections: 1) the Block Marker word; 2) eight words of storage called Section No. 1: and 3) eight words of storage called Section No. 2. The Block Marker word is the first word in the block, then the Section No. 1 and Section No. 2 words alternate, with the first Section No. 1 word following immediately after the Block Marker word. The computer can read from or record on only one of the three sections at any one time.



Magnetic Tape Block Layout

The Block Marker

The Block Marker Word stores the block address assigned to a given block. Sixteen binary digits are set aside for the block address; the only restriction being that the two most significant bits cannot both be ones. The reason for this will be seen when the BLOCK SEARCH command is discussed. The units bit of the block address is in the 13th binary digit position counting from the least significant end of the Block Marker word. If the user wishes these sixteen bits may be expressed as four binary-coded decimal digits. However care must be exercised when addressing the block with the BLOCK SEARCH command since commands are always written in the octal notation (c. f. Section II, page 4).



Block Marker Word (Bits)

BLOCK SEARCH 14

1)

		m_1				m_2			m_3
b	s	3	0	0	0	3	A_c		A_t

Bits Determining Mode of Block Search

2) Positive, no overflow

3) The time of execution depends on the location of the block on the tape. However since the searching is independent of the computer, the actual computer time is less than 20 milliseconds.

4) 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 one.

A_t is a sixteen bit number which specifies the address of the block on the tape to be search for. The only restriction on this address is that the two most significant bits cannot both be one when used as a block address.

There are two modes of the BLOCK SEARCH command.

a) When the two most significant bits of A_t are not both ones, the tape unit block searches for the block specified by A_t. The motor of the tape unit specified by A_c accelerates to a speed of 90 inches per second with a time expense of about .75 seconds. After the acceleration is completed, the tape unit then searches and finds the block on the tape corresponding to the block specified by A_t. After the block is found the tape halts and the motor decelerates to a speed of 15 inches per second. The deceleration process takes about .75 second. It is impossible to tell the tape unit to BLOCK SEARCH when a read or write is taking place; however it is possible to tell a tape unit to perform this mode of 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.

b) When the two most significant bits of A_t are both ones, the tape will back up for the one block and halt without any change in motor speed which means no time lost due to acceleration or deceleration. The "Back up one block" mode of BLOCK SEARCH may be followed by either mode of the BLOCK SEARCH command provided sufficient time delay exists so that the tape unit will have completed the first command before attempting to perform the next BLOCK SEARCH command. A delay of approximately 4 drum revolutions should intervene. The TEST SEARCH command may be used to check if a normal BLOCK SEARCH is completed, but will have no effect on the "Back up one Block" mode of search.

5) 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.

READ TAPE

			m_1			16		m_2			m_3		
r	t	x	x	x	x	x	A_c	0	k	x	x	x	n

1) (see instruction word above)

2) Negative, no overflow.

3) The execution time depends on the number of tape blocks to be read. The first tape block takes 137.6 milliseconds and approximately 133.3 milliseconds for each tape block following.

4) A_c is an octal digit which specifies which tape unit is to be read. If the CRC 102-D system has only one magnetic tape unit attached, A_c will be one. 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.

In the CRC 102-D system up to 1032 words (129 tape blocks) may be read from the CRC 126 Tape Unit to the main memory and buffer register of the CRC 102-D computer with a single READ TAPE command. Each tape block is transferred to the buffer register and then emptied out to the main memory into cells designated by the READ TAPE command.

Only one of the three sections of the tape block may be read with any one READ TAPE command. Which section is read depends upon the octal digit, K.

a) If K = 0, the Block Marker word is read. The number in the m_3 section of the READ TAPE command gives the cell in the main memory to which it should go minus 10 in octal. The least significant octal digit of m_3, N, chooses the buffer register cell, 200N, into which the Block Marker word is to be read. After the Block Marker word is read to the buffer, the whole buffer, beginning with cell 200N + 1, is emptied out to the eight main memory cells beginning with cell m_3 + 1. Only one Block Marker word will be read to the memory with one READ TAPE command.

b) If K = 1, Section No. 1 is read; if K = 2, Section No. 2 is read. Whenever either Section No. 1 or Section No. 2 are to be read, the first cell of the main memory to be entered is m_3 + 10 in octal. The least significant digit of m_3, N chooses the buffer cell into which the first word of the section is to be read. The following seven words are read into the buffer into 200N + 1, 200N + 2, etc., cyclically. The buffer is then emptied to the main memory into the cells beginning with m_3 + 10. Succeeding tape blocks are then read to the buffer and in turn emptied out to the memory.

The address specified by the m_3 of the word located in the cell designated by the m_1 of the READ TAPE command is the last buffer out address.

5) Examples

a) rt m_1 3101 1073

contents of m_1 = 35200020031303

This command will read from tape unit No. 1 and Section No. 1 to the main memory cells beginning with 1103 and ending with 1312.

b) rt m_1 3002 1770

contents of m_1 = 35200120022000

This command will read from tape unit No. 0 and Section No. 2 to the buffer cells beginning with 2000 and ending with 2007.

WRITE TAPE 15

1)

		m_1						m_2			m_3		
w	t	x	x	x	x	x	A_c	0	k	x	x	x	0

2) Positive, overflow

3) The execution time depends on the number of tape blocks to be recorded on. The first tape block takes 137.6 milliseconds; each succeeding block takes approximately 133.3 milliseconds.

4) A_c is an octal digit which specifies which tape unit is to be written on. If the CRC 102-D system has only one magnetic tape unit attached, A_c will be one. 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.

In the CRC 102-D system, up to 129 CRC 126 blocks (1032 words) may be written on from the main memory and the buffer of the CRC 102-D computer, with a single WRITE TAPE command.

Information can be transmitted to only one of the three sections of a tape block with a single WRITE TAPE command. The octal digit, K, which is the least significant digit of m_2 of the TAPE command, designates the particular section.

a) K = O: The Block Marker word is transmitted from the main memory to the tape. The m_3 of the WRITE TAPE command designates the cell in which the Block Marker word is to be found. The least significant octal digit of m_3 must be a zero.

b) The CRC 102-D computer loads groups of eight words into the buffer and then records them on the magnetic tape unit as specified by the A_c of the WRITE TAPE command. When K = 1, the data is transcribed onto Section No. 1; when K = 2, onto Section No. 2. m_3 of the WRITE TAPE command specifies the address of the first word of the group to be recorded on the tape. The word in the cell specified by m_1 of the WRITE TAPE command supplies the necessary information to stop the recording process. The m_3 section of this word should be equal to the last cell to be written on the tape plus 02 in octal.

5) Examples:

If cells 1370 through 1527 were to be transferred to Section No. 1 of a tape unit, the command would be

wt m_1 2001 1370

The contents of m_1 = 35 2000 2000 1531

NOTE

It is not possible to give two READ TAPE or WRITE TAPE commands in a row. There must be at least 4 drum revolutions delay between any two commands calling for a READ or WRITE TAPE. If the program is such that this is not inherent in the code, a SHIFT MAGNITUDE command calling for the shift of a dummy cell 200 (octal) left is usually sufficient. If this is not done the computer will hang up in program counter 605 for READ TAPE or in program counter 650 for WRITE TAPE.

IX

AUTOMATIC INPUT AND OUTPUT ON THE CRC 102-D

Input - Output Devices

The Flexowriter is the primary input-output device associated with the CRC 102-D. It has been described in detail in Section III. This section describes the Automatic use of the Flexowriter and also auxiliary equipment for input and output under control of the program.

The Teletype High Speed Tape Punch may be attached to the CRC 102-D as auxiliary output equipment. Data punched on the Flexowriter tape by the Teletype Punch may be printed on any Flexowriter Typewriter modified for use with the CRC 102-D. The Teletype Punch operates at a speed of 60 characters per second and is controlled by the PRINT command.

The Ferranti High Speed Tape Reader may be attached to the CRC 102-D as auxiliary input equipment. Information may be fed to the computer under control of the program at a much higher rate of speed than that achieved by the Flexowriter and may be intermediate results punched by the Teletype Tape Punch. The Ferranti Reader operates at an effective rate of 170 characters per second in the octal mode, 160 characters per second in the decimal mode, and 147 characters per second in the alphabetic mode.

The Print Command

Results of computations, previously stored data, program monitoring information, or alphabetic expressions are printed or punched under the control of the PRINT COMMAND. Alarms are printed out but not under control of the program (see Section XI).

The printed data may be commands in the octal notation, numbers in either the decimal or octal mode, or alphabetic information;

the PRINT COMMAND mode will determine how the binary digits of the words to be printed will be interpreted by the output.

PRINT

1) pr m_1 m_2 m_3
2) Positive, overflow
3) Approximately 100 milliseconds for each character when using the Flexowriter; approximately 16 milliseconds, depending on the print mode, when using the Teletype Punch.
4) The total number of words to be typed or punched with the one print command is m_3 as an octal integer; m_3 must be greater than zero.

   The first word to be typed will be the contents of m_1. If more than one word is to be typed,the words following the first one are the contents of $m_1 + 1$, of $m_1 + 2$, etc. The last word to be typed will be in cell $m_1 + m_3 - 1$. All words are typed in the same mode.

   The octal sign digits of the word in m_2 control the mode of printing and the choice of Flexowriter or Teletype Tape Punch. Table XII gives the mode as designated by the two octal sign digits of the word in m_2.

   When printing in the decimal mode the following binary combinations have no numerical significance. However, they may be used to edit the output;

1010, 1011	Ignored
1100	Tab
1101	Space
1110	Period
1111	Code Delete

The number of digits which will be printed from the magnitude section of each word is controlled by the magnitude section of the word in m_2. This is effected by placing a binary one in the least significant bit position of the octal or decimal or alphabetic digit position corresponding to the last digit to be typed or punched from a single word.

TABLE XII

Octal Sign Digits of Words in m_2	Mode of Printing or Punching
00	Type the octal sign digits and the magnitude as a 14 octal digit number.
20	Punch paper tape using the Teletype High Speed punch as described for 00.
01	Type the sign digit as space, -, p, n according to Table IV and type the magnitude as a 9 digit decimal number.
21	Punch paper tape using the Teletype High Speed Punch as described for 01.
02	Type the octal address of the cell being printed followed by a space and type its contents as described in 00.
22	Punch paper tape as described in 02 using Teletype High Speed Punch.
03	Type the octal address of the cell being printed followed by a space and type its contents as in 01 above.
23	Punch paper tape as described in 03 using Teletype High Speed Punch.

NOTE: The above modes of typing, after printing each word, automatically tab the Flexowriter carriage to the next tab stop preset by the operator.

Octal Sign Digits of Words in m_2	Mode of Printing or Punching
10	This mode allows the program to activate all the keys on the Flexowriter under control of the alphabetic code. The magnitude of the word to be printed is divided into six pairs of octal digits and is coded as in Section III. There is no automatic tabbing after the printing of each word. Any desired editorial characters must be coded. This includes shifting up, shifting down, backspace, tabbing, etc. It must be remembered when printing in the alphabetic mode that the automatic carriage return must be placed so as not to cause a carriage return while printing.
30	Punch paper tape with the Flexowriter code for the alphabetic, editorial and numeric characters desired to be printed out using the Teletype tape Punch.

5) Examples:

a) pr 1726 2100 0002

(1726) = 35200020012003

(1727) = 26200313252003

(2100) = 00000000000000

Result: 35 2000 2001 2003 26 2003 1325 2003

b) pr 1001 0204 0001

(1001) = +314159782 (decimal notation)

(0204) = 03000000000400 (octal notation)

Result: +3141597

c) pr 0000 0133 0010

(0000) = 00 4371 4455 5235
(0001) = 00 4363 4446 7156
(0002) = 00 6463 4771 3543
(0003) = 00 5044 4670 5535
(0004) = 00 4367 4452 5456
(0005) = 00 7071 5267 3543
(0006) = 00 5044 6462 6546
(0007) = 00 6376 3635 3535
(0133) = 10 0000 0000 0000
Result: The National Cash Register Company.

The Fill Command

Paper tapes containing the initial program, data, or previously computed results may be fed to the computer without manual intervention by means of the FILL command. However, certain special characters must appear on the tape. These are "Compute Code" - starts the computation again after the fill is completed; and "Stop Code" - stops the paper tape reader.

All necessary control information may be included on the paper tape.

The Flexowriter Tape Reader may be started manually by depressing the Start Read Key on the Flexowriter Key Board. The Ferranti Tape Reader may be started by putting the "START FERRANTI" switch on the console in the "ON" position.

FILL

1) fl 3000 300N m_3

2) Positive, overflow

3) Indeterminate

4) Stop computation and return computer to IDLE.
Prepare computer to accept octal.
Leave Control Register prepared to place the first word entering the Input Register, into cell m_3.

If the least significant octal digit of m_2, N, is 0, start the Ferranti High Speed Tape Reader; if the least significant octal digit of m_2, is a 1, start the Flexowriter Paper Tape Reader.

When the read-in process is completed and if the computer is not started with a "Compute Code" punched in the tape, the computer will remain in the IDLE position until started manually.

X

USE OF IBM EQUIPMENT WITH THE CRC 102-D

General Information

The CRC 102-D 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-D.

The CRC 102-D will accept signals read from cards by the IBM Gang Summary Punch, Type 523 with IBM Standard Modification #RFQ37624. The CRC 102-D 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-D. 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-D 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-D 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-D, 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 X, page 89. The following illustration demonstrates the path of the card through the Summary Punch Machine.

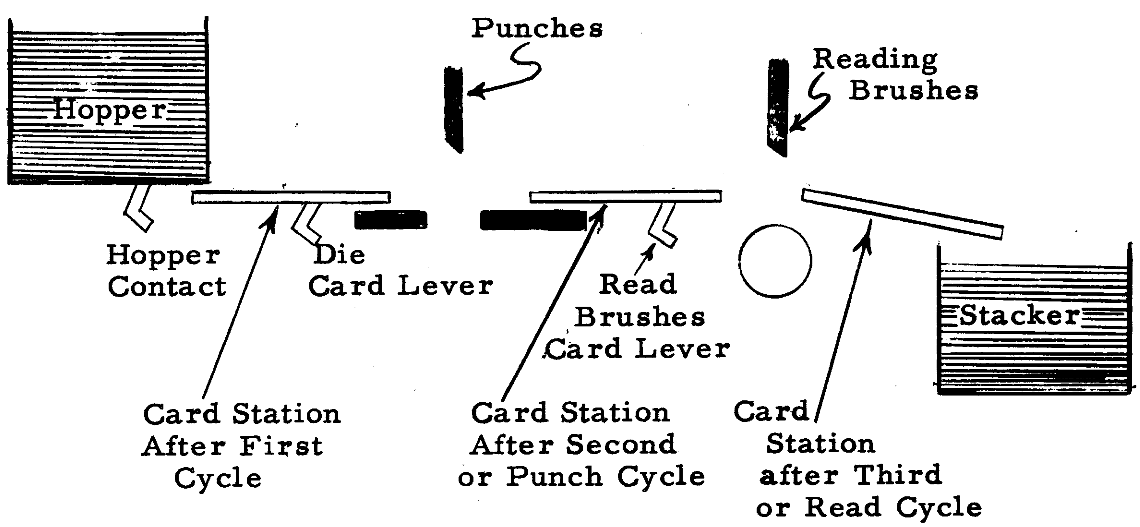


Diagram of IBM 523 Summary Punch

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-D 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-D

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 X, page 72.)

The CRC 102-D 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-D 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-D card reading process.

2) Decimal Data (see Figure V).
There are 40 inlet hubs for transmitting decimal data into the 102-D. 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-D

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-D; however, if a column split feature is ordered on the Summary Punch it may be used in the normal fashion.

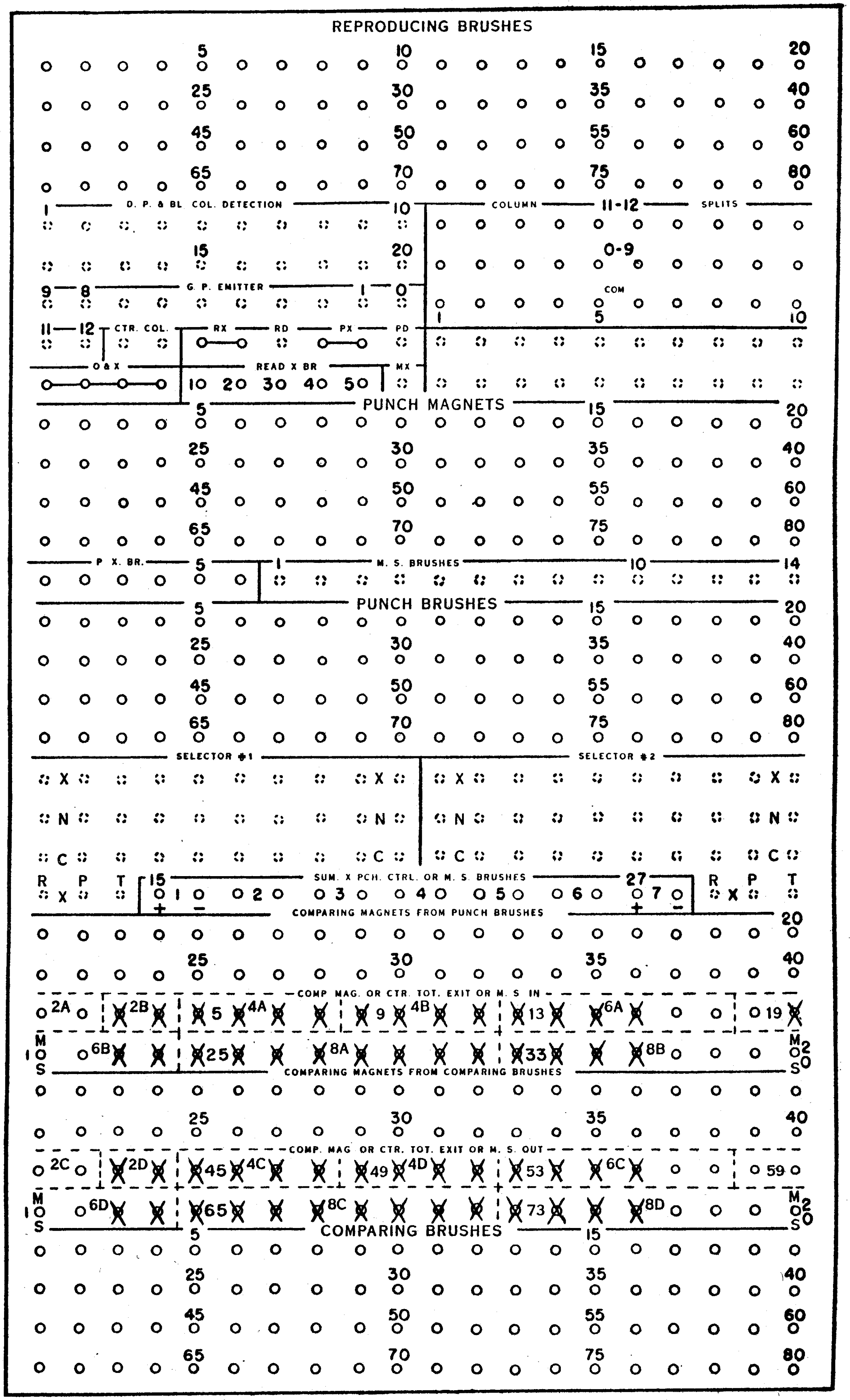

REPRODUCING BRUSHES
D. P. & BL. COL. DETECTION
COLUMN 11-12 SPLITS
0-9
COM
G. P. EMITTER
CTR. COL.
RX
RD
PX
PD
O & X
READ X BR
MX
PUNCH MAGNETS
P. X. BR.
M. S. BRUSHES
PUNCH BRUSHES
SELECTOR #1
SELECTOR #2
SUM. X PCH. CTRL. OR M. S. BRUSHES
COMPARING MAGNETS FROM PUNCH BRUSHES
COMP. MAG. OR CTR. TOT.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

Figure IV

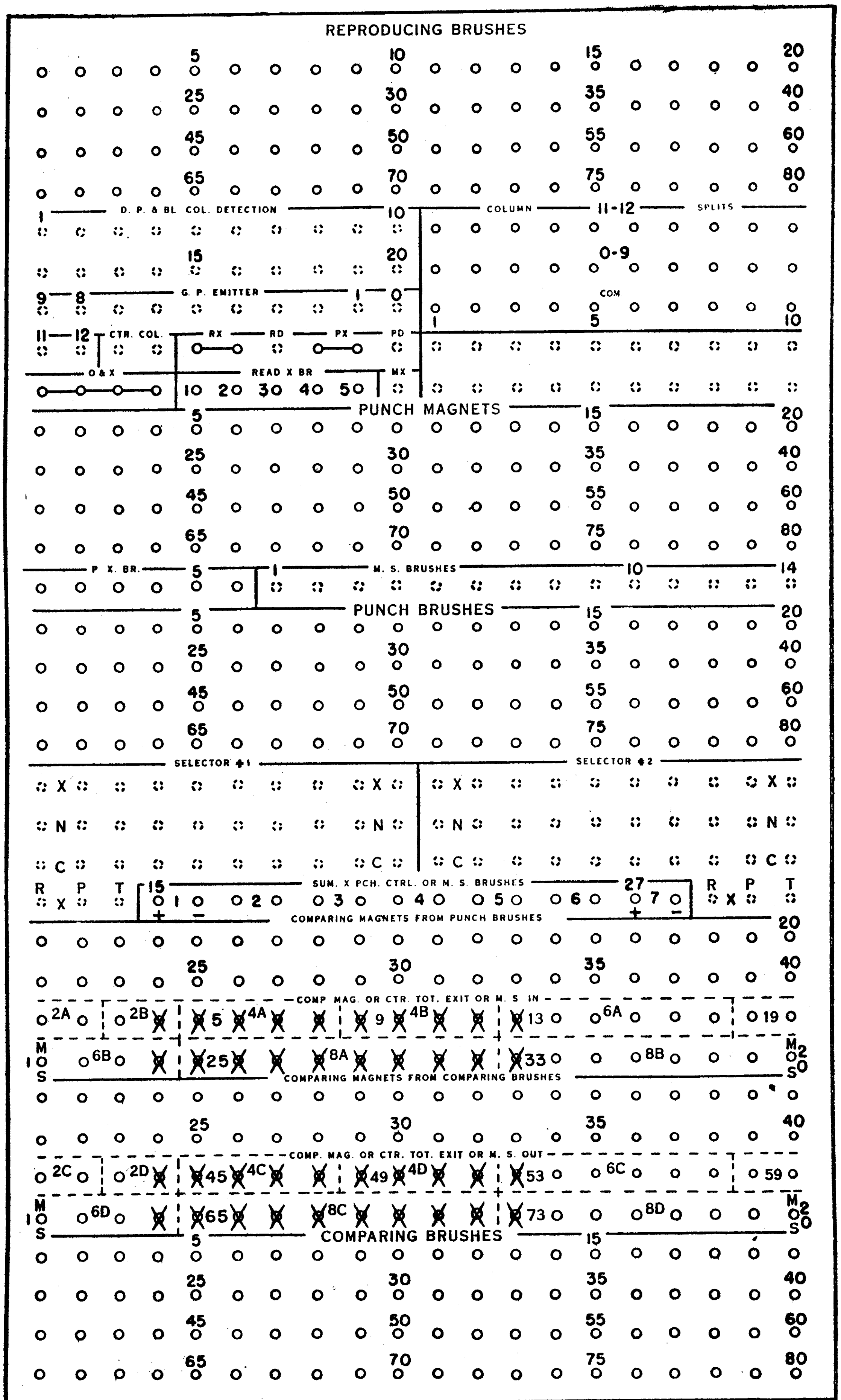

REPRODUCING BRUSHES
D. P. & BL COL. DETECTION
COLUMN
11-12
SPLITS
0-9
G. P. EMITTER
COM
CTR. COL.
RX
RD
PX
PD
O & X
READ X BR
MX
PUNCH MAGNETS
P. X. BR.
M. S. BRUSHES
PUNCH BRUSHES
SELECTOR #1
SELECTOR #2
SUM. X PCH. CTRL. OR M. S. BRUSHES
COMPARING MAGNETS FROM PUNCH BRUSHES
COMP MAG. OR CTR. TOT. EXIT 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
0	3	5	8	2	9	1	7	6	4

Decimal Digits

1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	3	2	7	6	4	5	1	0	2	5	3	1

Octal Digits

b. The Input 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 register is indexed by the addend 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 Register is filled by the READ CARD command with a group of digits which are added to the Input 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 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 Register is specified by (m_2). This is a different value for the normal octal reading process than for the normal decimal reading process.

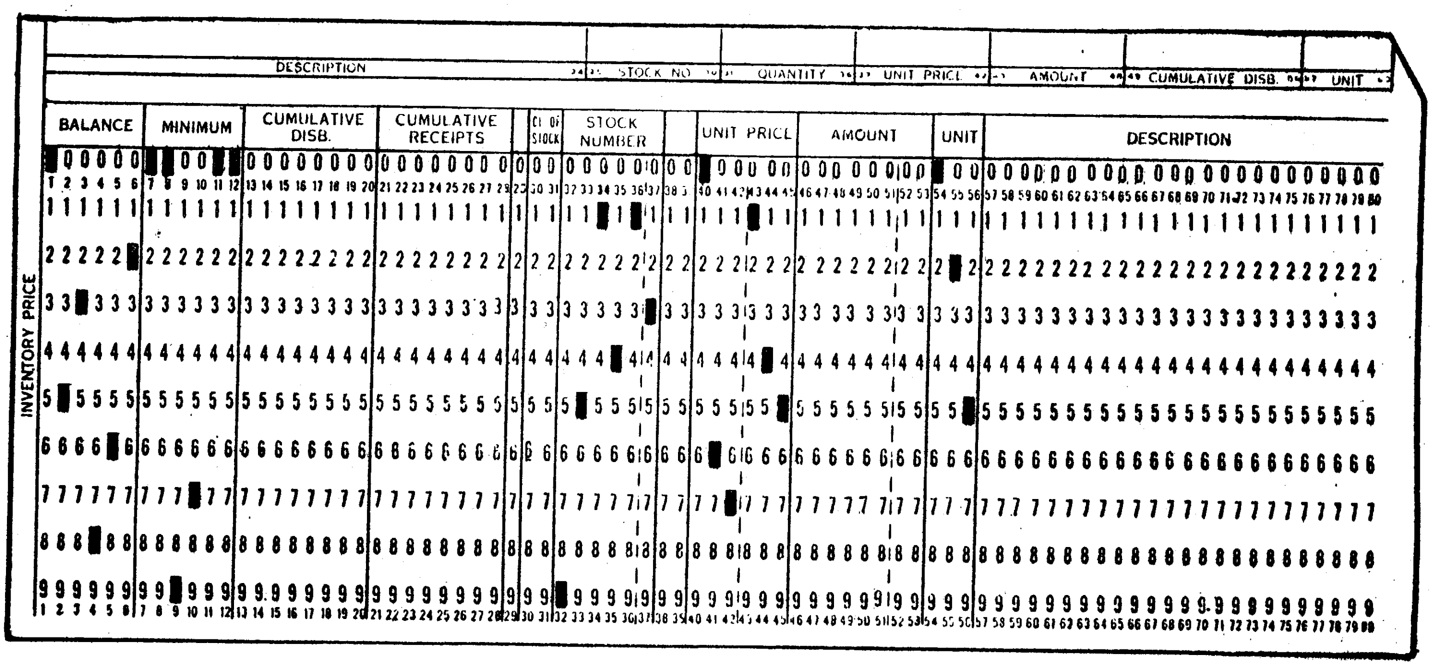
(m_2) when reading octal numbers or commands.

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.



I. B. M. Card

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	053962
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-D, 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).

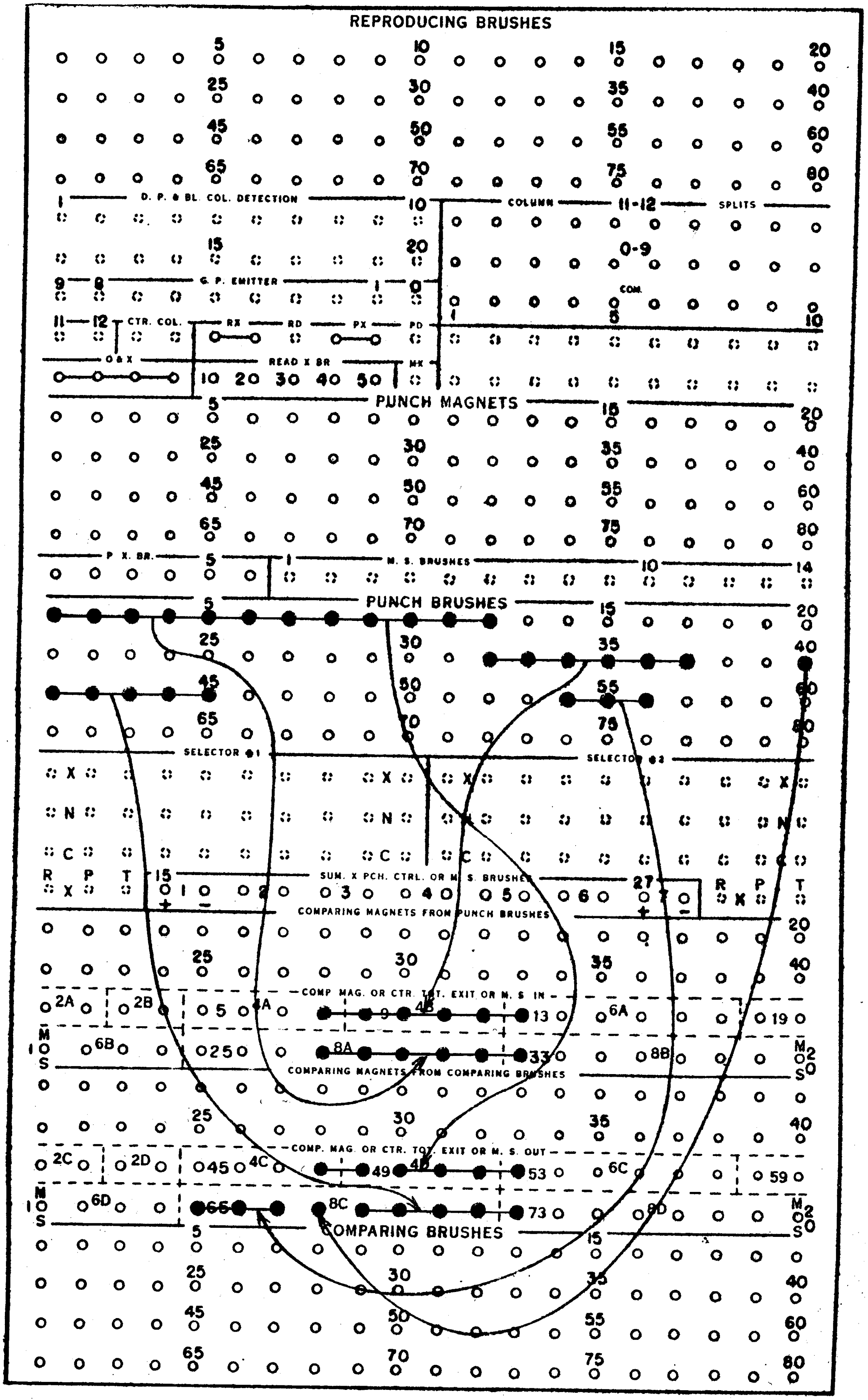

REPRODUCING BRUSHES
D. P. & BL. COL. DETECTION
COLUMN 11-12 SPLITS
0-9
COM.
G. P. EMITTER
CTR. COL.
RX
RD
PX
PD
O & X
READ X BR
MX
PUNCH MAGNETS
P. X. BR.
M. S. BRUSHES
PUNCH BRUSHES
SELECTOR #1
SELECTOR #2
SUM. X PCH. CTRL. OR M. S. BRUSHES
COMPARING MAGNETS FROM PUNCH BRUSHES
COMP. MAG. OR CTR. TOT.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

Figure VI

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) = 010421042104 21 (octal)

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 XII.

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 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-D

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 X, page 72.)

The 102-D 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 reentry into the 102-D. 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.

TABLE XII

(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 13	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 14			
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 0 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

Punching Procedure in the 102-D

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 balue 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 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-D 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-D 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 shift is important, also, when feeding cards continuously. The clutch is energized momentarily by the 102-D 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 1/2, and preferably 3, drum revolutions <u>must</u> be inserted between two READ CARD (octal) commands. If less than 2 1/2 drum revolutions are used, the computer will not obey the second READ CARD command but will hang up. 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 cammand, 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 almost the same way as the READ CARD commands.

A delay of from 2 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. In fact, the speed will not exceed 50 cards per minute.

XI

AUTOMATIC ALARM CHECKS

The CRC 102-D 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 CRC 102-D 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. 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, a 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 BIT, or a SHIFT LOGICALLY command, and switch No. 19 on the console marked "AUTOMATIC OVERFLOW TEST" is in the "IN" position, the computer will cause on 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. This helps to check the arithmetic rigor of the program.

Interpretation of Cell 3000 when printed as an alarm

The Alarm occurs because one of two things happened:

1) Either (m_2-2) generated an overflow; or

2) m_2-1 is not a command.

If (m_2-2) generated an overflow, the instruction portion of the cell 3000 print-out will contain a 71; if (m_2-1) is not a command, the instruction portion of the cell 3000 print-out will contain 70.

XII

TECHNIQUES OF MINIMUM ACCESS CODING

The minimum execution time for commands bears 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 of an arithmetic command. To realize the minimum 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 XIII indicates this numbering principle.

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

TABLE XIII

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 XIV. This is an example to illustrate the possibilities of combining a regularly numbered drum with a renumbered drum.

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.

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

TABLE XIV

APPENDIX A

SUMMARY OF CRC 102-D FUNCTIONAL SPECIFICATIONS

I. General

a) Serial Operation-one word time = .39 milliseconds

b) Decimal Arithmetic - 9 decimal digit capacity

II Word Size

a) Numbers: 9 binary-coded decimal digits plus a six bit xection containing a sign and overflow bit.

b) Commands: 14 octal digits

III Instructions

a) Three address system - each address is 4 octal digits

b) 27 distinct instructions - 2 octal digit code

List of Instructions in CRC 102-D

Code	Title	Code	Title
	Arithmetic		**Logical and Transfer**
35	Add Decimally	32	Extract
36	Subtract Decimally	34	Test Magnitude
25	Multiply Decimally and Round	33	Test Algebraically
26	Multiply Double Length Decimally	37	Test Bit
23	Divide Decimally and Round	17	Test Search and Test Switch
24	Divide and Save Remainder Decimally	05	Buffer Load
31	Scale Factor Decimally	04	Buffer Out
07	Add Binary	22	Halt
10	Subtract Binary		**Input-Output**
27	Shift Logically	11	Fill (from Flexowriter or Ferranti Reader)
30	Shift Magnitude	06	Read Card (Octal or Decimal)
	Magnetic Tape	21	Print (Flexowriter) or Punch (Flexowriter or Teletype High Speed Tape Punch)
14	Block Search or Back up One Block	13	Punch Octal (I. B. M.)
16	Read Tape	12	Punch Decimal (I. B. M.)
15	Write Tape		

IV Magnetic Drum Storage

a) Physical Specifications

1. Diameter = 12 inches
2. Length = 6 inches
3. Speed of rotation = 40 revolutions per second

b) Memory Specifications

1. 1024 words of permanent storage: 16 channels with 64 words per channel.
    a. Average access time = 12.5 milliseconds
2. 8 word minimum access buffer register
    a. Average access time = 1.5 milliseconds
3. Permanent zero storage - cell 2100
4. 4 one word registers - 3 arithmetic, 1 control

V. Magnetic Tape Storage

a) Physical Specifications

1. Tape is 1200 feet long, 1 inch wide
2. Up to 7 units may be attached to one CRC 102-D Computer
3. Tape Speed: High or Searching, 90 inches per second; Low or Read-Write, 15 inches per second.
4. Total Acceleration and Deceleration Time: 1.75 seconds.

b) Memory Specifications

1. Records in 10 channels, 5 effective channels plus duplicates
2. Records in blocks of 17 words, 2.04 inches to the block
3. Stores upwards of 100,000 words per reel of tape
4. Searches for Block address independently of computer
5. Able to back up one block at a time under control of computer
6. Reads or Writes up to 1032 words with one command.

VI Input

a) Three Fill Modes: Decimal, Octal, Alphabetic

b) Flexowriter Keyboard and Paper Tape Reader

1. Effective Rate: Approximately 6 characters per second

c) Ferranti High Speed Tape Reader

1. Effective Rate: 170 characters per second octally
160 characters per second decimally
140 characters per second alphabetically

d) I. B. M. Card Reader

1. Four decimal or octal words per card

2. Speed: .65 seconds per card

VII Output

a) Flexowriter Typewriter and Tape Punch

1. Effective Speed: Approximately 6 characters per second

b) Teletype High Speed Tape Punch

1. Effective Speed: Approximately 60 characters per second

c) I. B. M. Card Punch

1. Punches four octal or decimal words per card

2. Effective Speed: .65 seconds per card

APPENDIX B

INTRODUCTION TO THE BINARY AND OCTAL NUMBER SYSTEMS

A number system is defined as the method by which a set of symbols (usually called digits) are used to represent numerical quantities. Normally this representation is expressed as the sum of a set of these digits each multiplied by a power of some numerical quantity called the base. The name of the system, and the total number of symbols used by the system is determined by the numerical size of the base.

A number in a given system is written as a sequence of digits, the integral and fractional portions of the number being separated by a point. The position of the digits in the sequence denotes the power of the base by which the digit is to be multiplied. Digits to the left of the point are multiplied by positive powers of the base, while digits to the right of the point are multiplied by negative powers of the base. The following diagram shows the powers of the base and the corresponding digit positions relative to the point.

	Positions to Left						Pt.	Positions to Right					
Position	. .	5	4	3	2	1	.	1	2	3	4	5	. . .
Power	. .	4	3	2	1	0		-1	-2	-3	-4	-5	-. -. -.

Digit Positions and Powers of Base

The most common number system in use today is the decimal (or base ten) system. It is so called because there are ten symbols in the system represented by the familiar arabic numerals : 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. If the rule stated above is applied, then the sequence of digits

1234.56 represents

$1x10^3 + 2x10^2 + 3x10^1 + 4x10^0 + 5x10^{-1} + 6x10^{-2} = 1x1000 + 2x100 + 3x10$
$+ 4x1 + 5x^1/10 + 6x^1/100.$

Because of engineering considerations, however, the internal arithmetic of most automatic computers is in the binary (or base two) system. In this system only two symbols are employed, 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. The sequence of binary digits, 110101.1, represents the following:

$1x2^5 + 1x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0 + 1x2^{-1} = 1x32 + 1x16 + 1x4 + 1x1 + 1x^1/2.$

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

$$\begin{array}{lr} & 11000011 \\ & -\ \underline{1000110} \\ \text{Difference} & 1111101 \end{array}$$

c. The multiplication table for binary digits is

1. $0 \times 0 = 0$
2. $1 \times 0 = 0$
3. $1 \times 1 = 1$

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

$$\begin{array}{r} 1101.1 \\ \times\ \underline{11001.1} \\ 11011 \\ 11011 \\ 1101100 \\ \underline{11011} \\ 101011000.01 \end{array}$$

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, these numbers are very difficult to manipulate outside the computer, so a number system which is exactly convertible to the binary system is usually substituted. One system which is commonly used in computer work is the octal (or base eight) system; since $8 = 2^3$, any combination of three binary digits can be represented by an octal digit. These digits are usually represented by the arabic numerals: 0, 1, 2, 3, 4, 5, 6, and 7. Table I, Section II, gives the binary representation of the various octal integers. A number in the octal system, expressed by the sequence of octal digits 1257.4, represents the following:

$1x8^3 + 2x8^2 + 5x8^1 + 7x8^0 + 4x8^{-1} = 1x64 + 2x16 + 5x8 + 7x1 + 4x1/8.$

Arithmetic in the octal system is similar to the decimal system. The following examples give an idea of the method

a. Addition

$$\begin{array}{r} 12375 \\ +\ \underline{20611} \\ 33206 \end{array}$$

NOTE: In octal $7+1 = 10$, i, e, $1 \times 8^0 + 0 \times 8^0 = 8$.

b. Subtraction

$$\begin{array}{r} 33206 \\ -\ \underline{20611} \\ 12375 \end{array}$$

c. Multiplication

$$\begin{array}{r} 127 \\ \times\ \underline{301} \\ 127 \\ \underline{4050} \\ 40627 \end{array}$$

Because all business records and transactions and most scientific data are secured and kept in the decimal number system, and because binary numbers and their usual substitutes can not easily be converted to decimal numbers, the problem of converting data from decimal to the binary and of converting computer results from binary to decimal becomes a very arduous task. Hence a pseudo-number system called the binary coded number system has come into general use with high speed digital computers. In this system four binary digits are used to represent each decimal digit. Table II, Section II, gives the binary representation of each decimal digit.

For example, the number 8203.6594 in the decimal notation becomes in the binary coded decimal notation

1000 0010 0000 0011 . 0110 0101 1001 0100

Arithmetic carried out in the binary coded decimal system is equivalent to arithmetic carried out in the decimal number system. The arithmetic section of the CRC 102-D carries out all arithmetic operations in the binary coded decimal number system. Hence no time is wasted in conversion and results are printed in the familiar decimal number system.

The decimal number, 33.125 may be expressed in the CRC 102-D System in the following manner:

INPUT-OUTPUT		INTERNAL OPERATIONS	
Number System	Sequence Of Digits	Number System	Sequence Of Digits
Octal	41.1	Binary	100001.001
Decimal	33.125	Binary Coded Decimal	00110011.000100100101

APPENDIX C

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.
Coded Decimal Number	- A decimal number expressed in binary notation, each decimal digit being represented by 4 binary digits.
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 CRC 102-D 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 control of the program.
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 standarized 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.