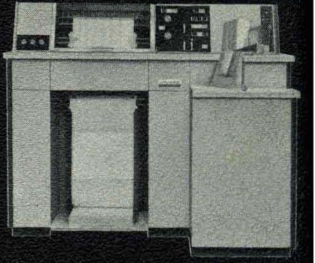
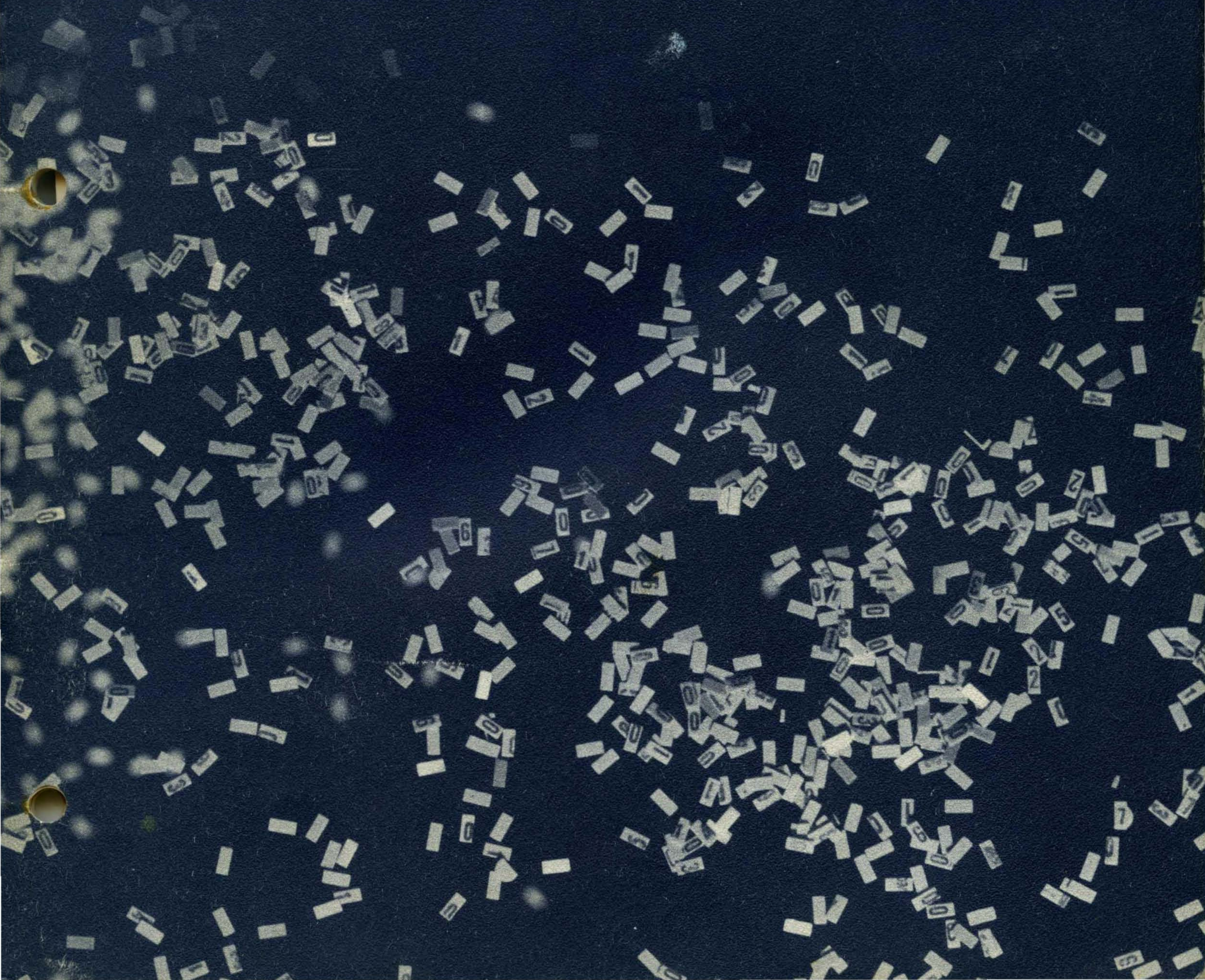
UNIVAC®



1004 CARD PROCESSOR

REFERENCE

UNIVAC® 1004 Card Processor
80 Column



REFERENCE

UNIVAC® 1004

Card Processor

80 Column

CONTENTS

SECTION 1. GENERAL DESCRIPTION

SECTION 2. PROGRAMMING

Section 1

General Description

1. INTRODUCTION

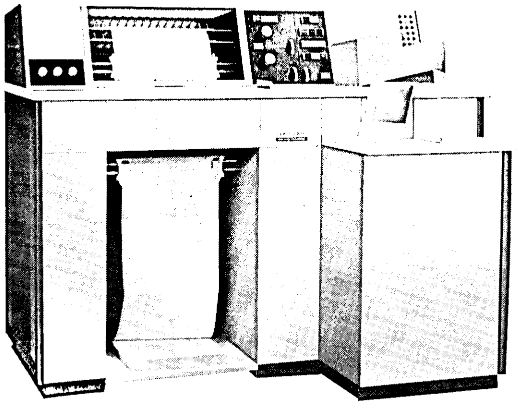


Figure 1-1. UNIVAC 1004 Card Processor.

The UNIVAC® 1004 Card Processor, 80-column edits and accumulates totals from data punched into 80-column cards and prints the results in any desired format. Built-in abilities to perform arithmetic, transfer, and compare operations, and reliable fast-access magnetic core storage provide a high degree of data-processing efficiency.

The 1004 processor consists of a card reader, a processor, and a printer, housed in a single compact unit (Figure 1-1). The unit is built to assure ease of operation with centrally located operating controls. A card punch can be included as an optional output unit (Figure 1-2).

The functions of card reading, data-processing, printing, and punching are user-programmed through wiring of a removable connection panel. To accomplish the desired procedures, the machine follows a series of instructions called STEPS. These steps are defined by connection panel wiring and can be executed in any sequence. The use of magnetic core storage allows the steps to proceed at microsecond speeds.

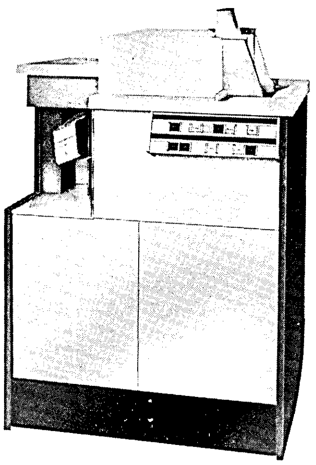


Figure 1-2. Optional Card Punch.

II. READ SECTION

The card reading section is at the right front of the machine (Figure 1-3). Reading is performed column by column at a basic speed of 300 cards per minute. A read station of twelve photocells reads each punching position in each column from 1 through 80. During reading, the card image is transferred to a section of the core storage specifically assigned to card reading. This area of storage is referred to as READ STORAGE.

The input magazine at the front of the read section has a capacity of approximately 1,000 cards, and is angled toward the centrally grouped controls for ease of access.

The card stacker, located above and to the rear of the input magazine, holds approximately 1,500 cards. A supply of cards is placed in the input magazine, face down, with the nine-edge leading. A card is fed to a wait station where the direction of travel is altered to allow the card to pass under the photocells serially. After the card is read, it is deposited "on-end" in the card stacker. Figure 1-4 is a line drawing of card travel through the card reading section of the machine.

A reading speed of up to 400 cards per minute can be obtained if reading is ended before the full 80 columns are read. A read section timing graph is shown in a foldout at the rear of the manual.

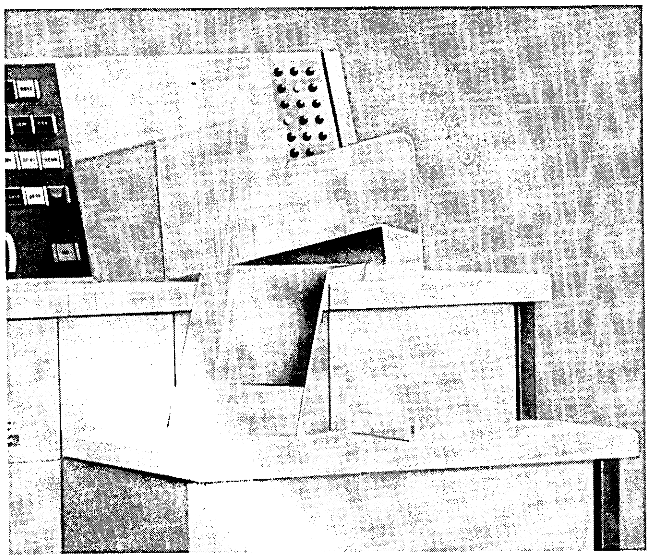


Figure 1-3. Read Section

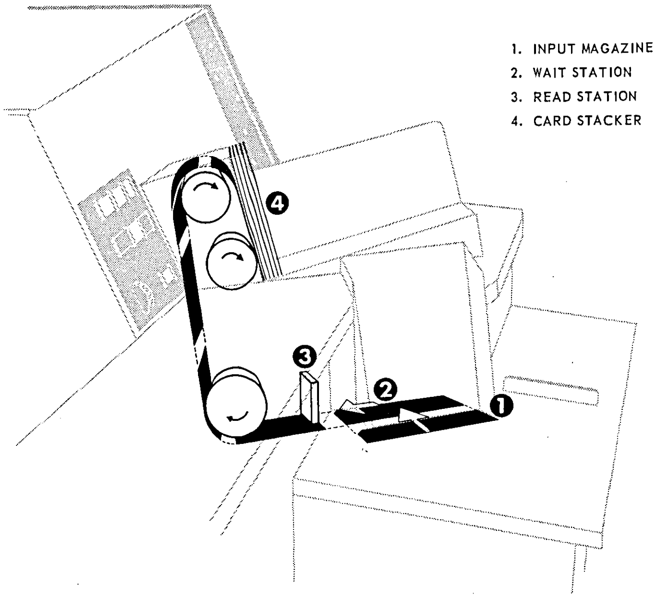


Figure 1-4. Card Travel in Read Section.

III. PROCESSING SECTION

The processing section contains magnetic core storage and the control circuitry necessary to perform the machine operations required. Once input data has entered read storage, the control circuitry performs whatever operations have been programmed for execution. When these are completed, the results are transferred to output storage for printing and/or punching. Output storage areas are referred to as PRINT STORAGE and PUNCH STORAGE.

Magnetic Core Storage

The capacity of core storage is 961 locations. Each location is made up of six magnetic cores, with each of the six cores in a separate core plane. Six 31-by-31 core planes make up the entire storage (Figure 1-5). Any individual six-core storage location is directly addressable, and any number of adjacent locations is directly addressable as a single data unit.

Storage is addressed in segments called OPERANDS. An Operand can consist of any number of adjacent storage locations from 1 to 961. Two operands are required to execute a step.

A portion of storage is allocated to each of the input-output functions of the machine – reading, printing, and punching. The remaining storage area is WORKING STORAGE. Under certain conditions, part or all of the input-output storage areas can be used to expand working storage. For example, if a card punch is not being used, the punch area of storage can be used as working storage.

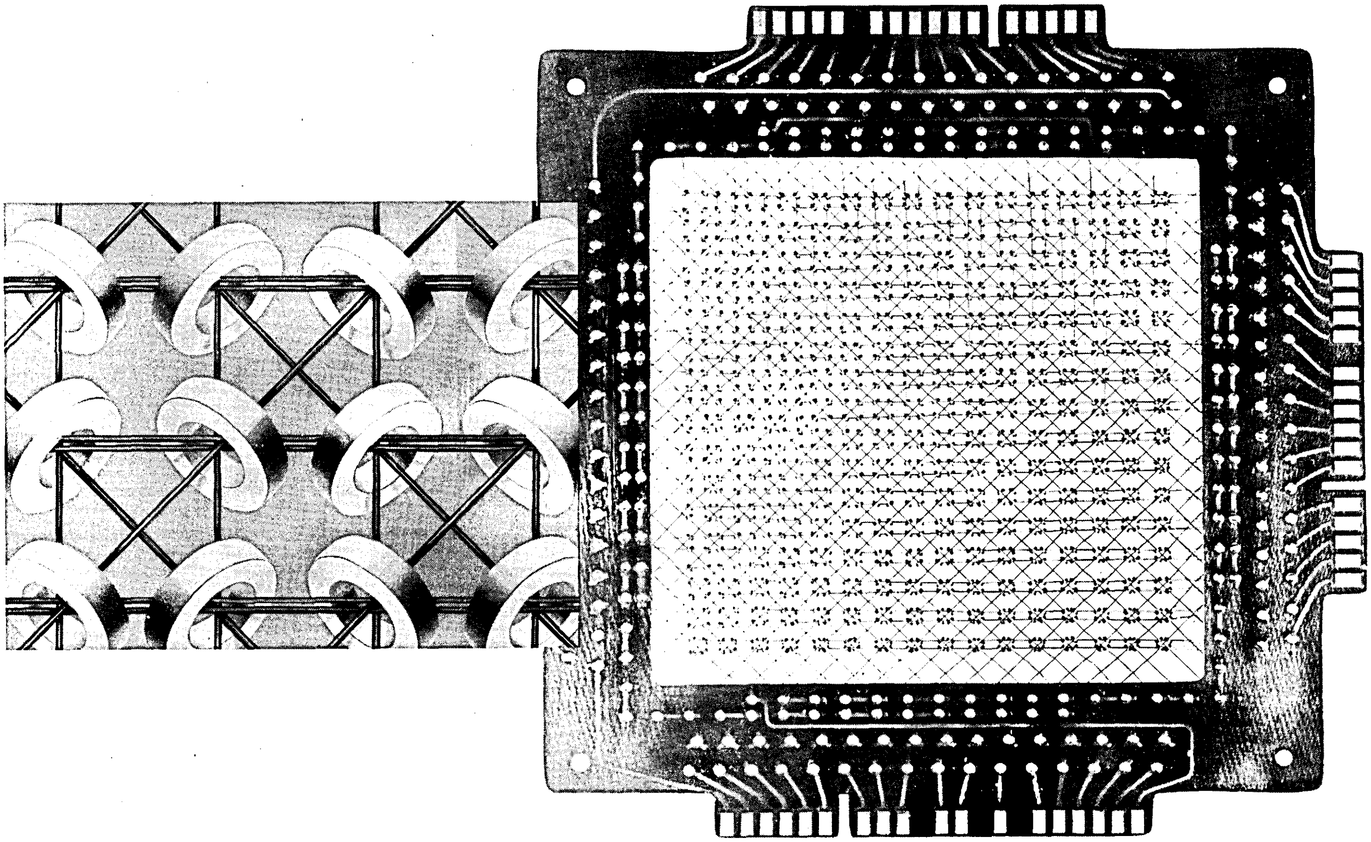


Figure 1-5. Plane of Magnetic Core Storage.

Storage control logic is designed so that "time sharing" can be effected; that is, reading, printing, and punching can occur simultaneously, or punching and processing can occur simultaneously, Processing cannot take place during read or print time.

A maximum of 62 steps can be directly wired on the connection panel. This number can be effectively increased through the use of Selectors and/or Comparators. Only those program steps wired are executed.

Machine Code

The 1004 processor is a character oriented machine. Data is transferred between storage locations one character at a time. The machine code of the Processor is UNIVAC XS-3 code. Cards are read column by column, and each column of data is automatically translated from card code to XS-3 before it is transferred into a 6-core storage location. Figure 1-6 shows the 63 printing characters and their equivalents in card code and in XS-3.

80-COLUMN CODE

80-Col. Card Code	Printable Characters	XS-3 Code	80-Col. Card Code	Printable Characters	XS-3 Code
12–1	A	01 0100	7	7	00 1010
12–2	B	01 0101	8	8	00 1011
12–3	C	01 0110	9	9	00 1100
12–4	D	01 0111	12	&	01 0000
12–5	E	01 1000	11	- (minus)	00 0010
12–6	F	01 1001	12–0	?	01 0011
12–7	G	01 1010	11–0	! (exclam.)	10 0011
12–8	H	01 1011	0–1	/	11 0100
12–9	I	01 1100	2–8	+	11 0011
11–1	J	10 0100	3–8	#	01 1101
11–2	K	10 0101	4–8	@	10 1110
11–3	L	10 0110	5–8	: (colon)	01 0001
11–4	M	10 0111	6–8	>	11 1110
11–5	N	10 1000	7–8	' (apos.)	10 0000
11–6	O	10 1001	12–3–8	. (period)	01 0010
11–7	P	10 1010	12–4–8	⌑	11 1101
11–8	Q	10 1011	12–5–8	[	00 1111
11–9	R	10 1100	12–6–8	<	01 1110
0–2	S	11 0101	12–7–8	=	01 1111
0–3	T	11 0110	11–3–8	$	10 0010
0–4	U	11 0111	11–4–8	*	10 0001
0–5	V	11 1000	11–5–8	]	00 0001
0–6	W	11 1001	11–6–8	;(semi-col)	00 1110
0–7	X	11 1010	11–7–8	Δ	10 1111
0–8	Y	11 1011	0–2–8	≠	11 0000
0–9	Z	11 1100	0–3–8	, (comma)	11 0010
0	0	00 0011	0–4–8	%	11 0001
1	1	00 0100	0–5–8	(	10 1101
2	2	00 0101	0–6–8	\	00 1101
3	3	00 0110	0–7–8	)	11 1111
4	4	00 0111	Blank	Space N.P.	00 0000
5	5	00 1000			
6	6	00 1001			

Figure 1-6. 80-Column Codes and UNIVAC XS-3 Codes for 63 Printable Characters.

IV. PRINTING SECTION

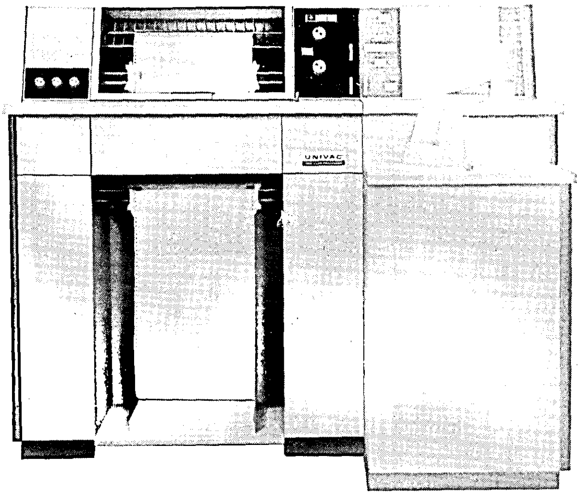


Figure 1-7. Printing Section.

The left portion of the processor is the printing section (Figure 1-7). Printing speed is 300 lines of alphanumeric data per minute, with a maximum of 132 print positions per line. Character spacing is ten characters to the inch horizontally, with an operator option of 6 or 8 lines to the inch vertically. For each print position there are 63 printing characters located around a print drum which rotates at high speed to position each character at the printing line.

Paper movement through the print section, other than one-and two-line spacing, is controlled by a paper tape loop that may be punched in one or more of three available channels. Paper tape loops are easily created and stored for a variety of form designs. Paper forms from 4 to 22 inches in width can be accommodated by the carriage. Form lengths up to 22 inches can be controlled by the paper loop mechanism.

V. CONTROL PANEL AND CONNECTION PANEL

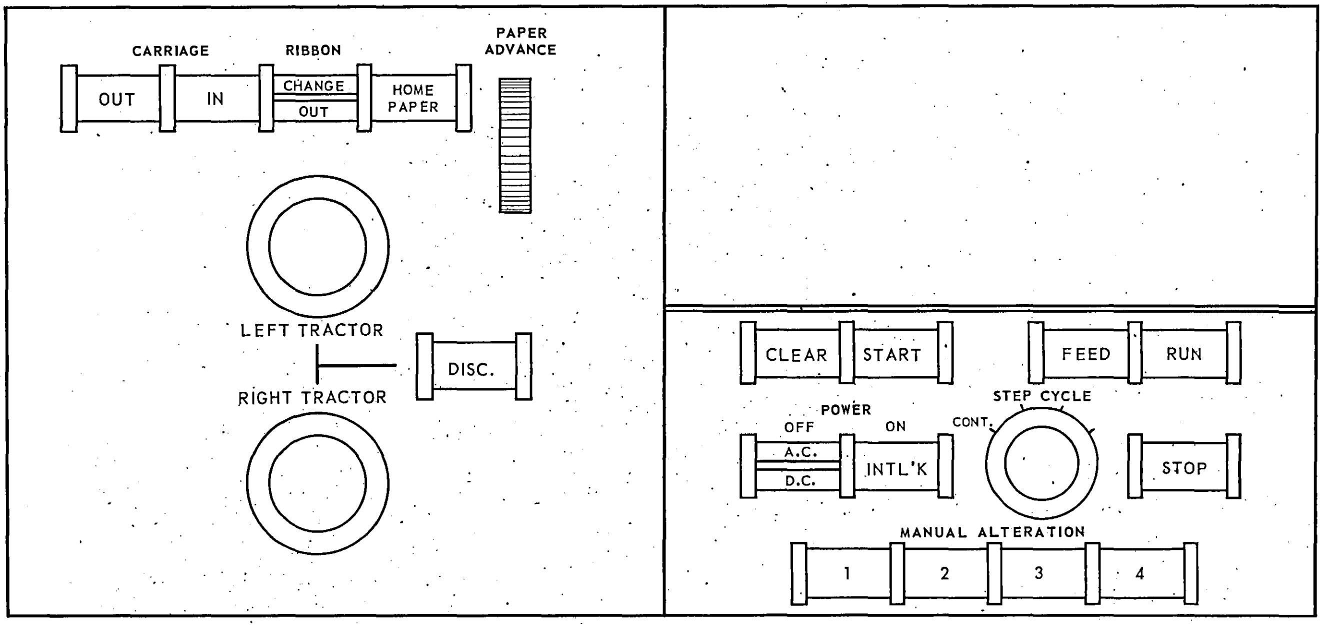


Figure 1-8. Central Control Panel

Operator controls are located on three panels at the front of the machine (Figure 1-8). These panels contain the buttons, switches, and lights with which the operator supervises machine operations. Display lights are provided to allow the operator to monitor program progress.

The connection panel (Figure 1-10) controlling the program in progress is housed in a recess at the right of the Processor. The connection panel can be removed from the recess for wiring, or for storage when other panels are in use.

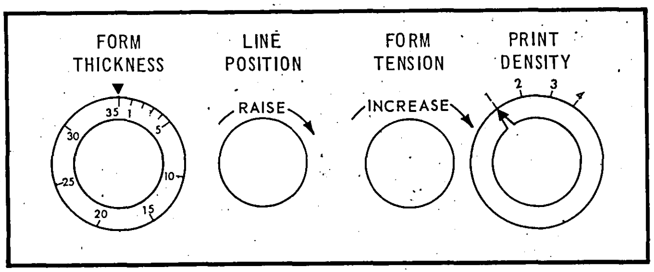


Figure 1-9. Printer Dials

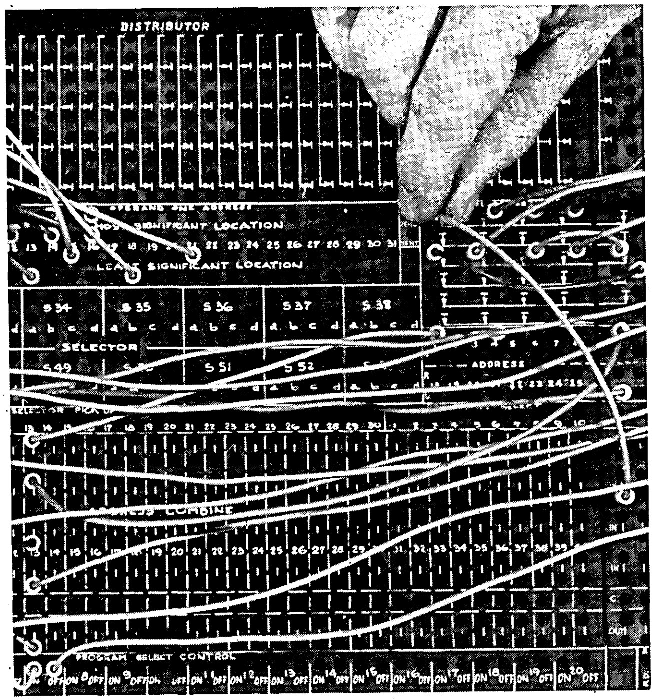


Figure 1-10. Connection Panel

Section 2

Programming

ABBREVIATIONS

The following is a list of the abbreviations used throughout this manual. This list includes those abbreviations used in text, and those which appear on the connection panel.

R Address	*Row designation of storage*
C Address	*Column designation of storage*
OP1	*Operand 1*
OP2	*Operand 2*
MSL	*Most Significant Location in storage*
LSL	*Least Significant Location in storage*
MSL-1	*First location to the right of MSL*
MSL-2	*Second location to the right of MSL*
TRF	*Transfer*
ZDS	*Zone Delete with Sign*
DØB	*Delete Zero Balance*
Ø-Δ	*Zero Suppress with Space Fill*
Ø-*	*Zero Suppress with Asterisk Fill*
CLR	*Clear*
INS	*Insert Transfer*
SI	*Superimpose Transfer*
SP1	*Space 1 line*
SP2	*Space 2 lines*
SK1	*Skip to terminal point 1*
SK2	*Skip to terminal point 2.*
SK4	*Skip to terminal point 4.*
SK1, 2, 4,	*In combination, Skip to terminal points 3, 5, 6, and 7*
RD	*Read*
PR	*Print*
PUN	*Punch*
H	*Hold*
C	*Clear; Cycle; Card; Common*
T	*Test*
CC	*Cycle Clear*
CH	*Cycle Hold*
ISD	*Inhibit Selector Delay*
F.O'F	*Form Overflow*
SENT	*Sentinel*
O'FLOW	*Arithmetic Overflow*
ST	*Start address of function desired*
END	*End address of function desired*
Δ GEN	*Space Generate*
Ø SUPP	*Zero Suppress*
NS	*Non-Select*
SEL	*Select*
S45b	*Selector 45, pole b*
I/O	*Input/Output*
COMP	*Compare*
CMPS	*Compress*

I. STORAGE

The UNIVAC 1004 processor employs magnetic core storage with a capacity of 961 six-core locations. Each character in storage is held in an addressable location. A storage location is designated by an address made up of the horizontal row number, called the R ADDRESS, and the vertical column number, called the C ADDRESS.

Storage consists of 31 horizontal rows of locations, with 31 locations to a row. The rows are identified, top to bottom, as R1, R2, R3, and so on to R31. Storage locations in the same position in each row make a column of locations. There are 31 columns of locations which are identified, left to right, as C1, C2, C3, and so on to C31. Storage can best be visualized as a 31-by-31 array of squares with each square representing a location in which a character may be stored (Figure 2-1).

To understand how storage operates, imagine a circle of storage locations in which storage location R2/C1 follows to the right of location R1/C31;

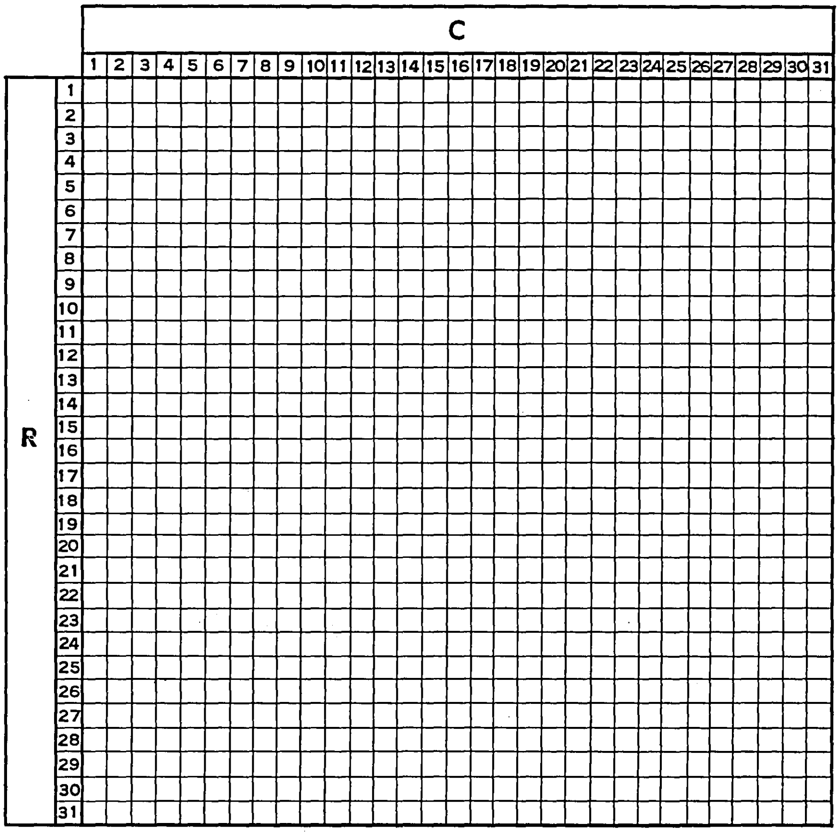


Figure 2-1. Storage Chart

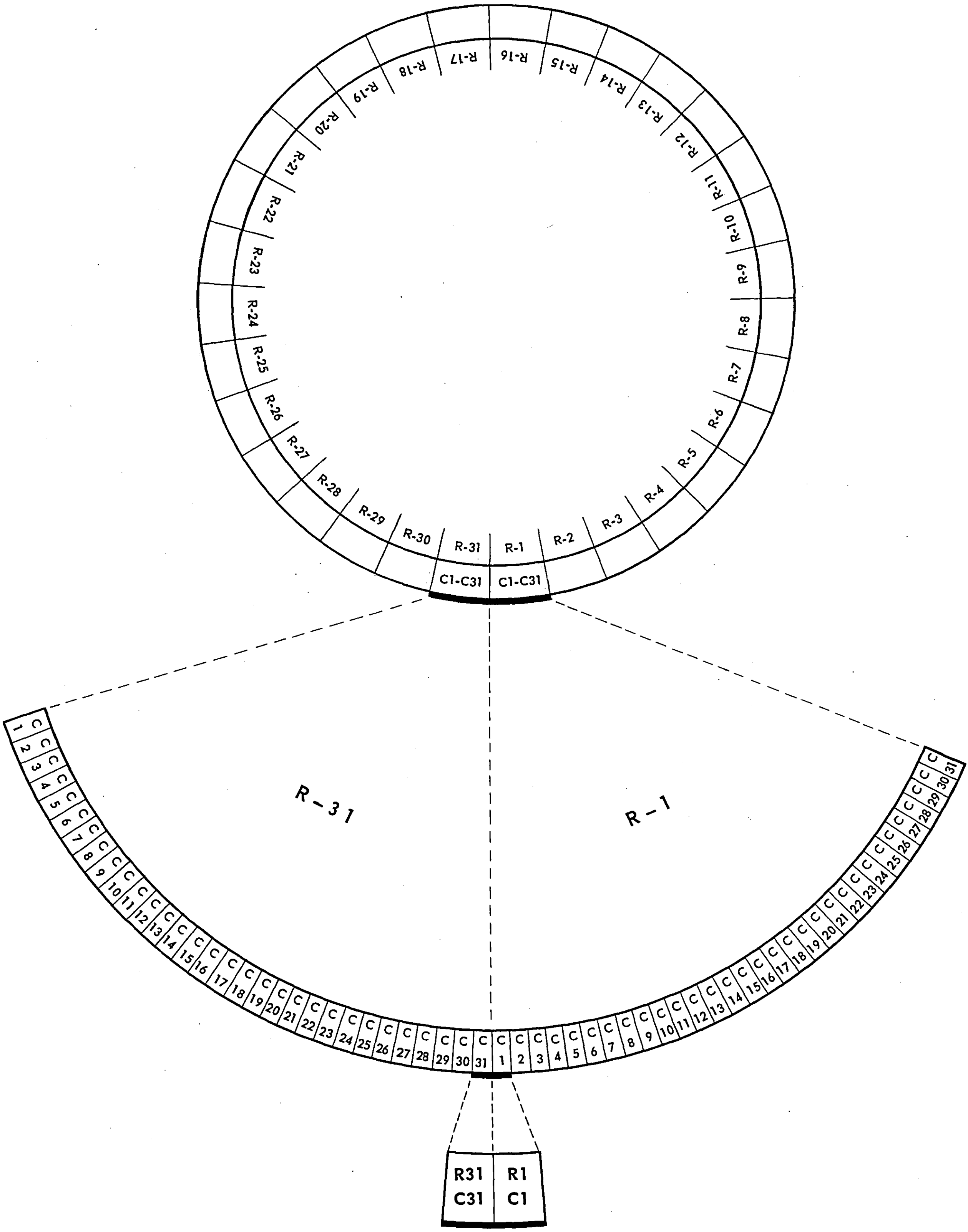
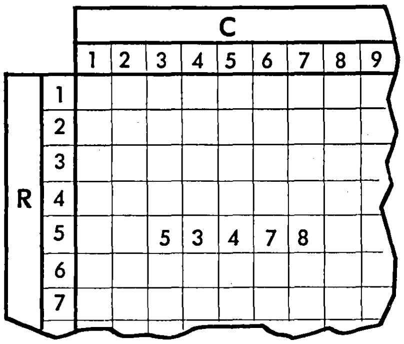


Figure 2-2. Circular Diagram of Core Storage.

R3/C1 follows to the right of R2/C31, and so on until R1/C1 follows to the right of R31/C31 (Figure 2-2).

An information segment in storage is called an operand. An operand consists of one or more consecutive storage locations. The user defines an operand by wiring, on the connection panel, the storage address of the most significant location (MSL) and the storage address of the least significant location (LSL). If, for any operand, the wiring of an MSL or an LSL is omitted, the processor will stall.

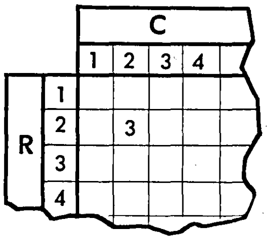
For example, a five-digit employee number is located in storage as shown in Figure 2-3. To use this data as an operand, storage locations R5/C3 – R5/C7 are wired.



	1	2	3	4	5	6	7	8	9
1									
2									
3									
4									
5			5	3	4	7	8		
6									
7									

Figure 2-3. Five-Digit Item in Storage.

A single location in storage can be specified as an operand. In this instance, the MSL and the LSL of the operand are identical. For example, a one-digit code appears in storage as shown in Figure 2-4. To use this code as an operand, wiring would specify R2/C2 – R2/C2.



	1	2	3	4
1				
2		3		
3				
4				

Figure 2-4. One-Digit Item in Storage.

Characters are stored using six magnetic cores per character. Those six cores correspond to the six levels of XS-3 code, designated X, Y, 8, 4, 2, and 1. The cores of a storage location are selectively magnetized to represent one of two conditions: "bit" or "no bit," depending upon which levels of XS-3 code are represented by bits. It is a convention to designate these conditions as "1" and "0," respectively.

The character "6" is recorded in XS-3 code by the presence of bits in the 8 and 1 levels, and the absence of bits in the X, Y, 4, and 2 levels. The same character is stored by magnetizing the cores in the pattern 001001.

In the same way, a "Z" contains bits in the X, Y, 8, and 4 levels and no bits in the 2 and 1 levels. In storage, this is recorded as 111100.

INPUT-OUTPUT STORAGE

Areas of storage have been defined as input-output storage for the reader, the printer, and the punch. These storage areas are referred to as read storage, print storage, and punch storage. These storage areas are fixed, and are always used in connection with the input-output functions. Under certain circumstances, which are explained in following paragraphs, the input-output storage areas can be made available to the programmer for use as working storage.

Figure 2-5 illustrates storage with the input-output storage areas defined.

Read Storage

The read storage area is located in storage locations R1/C1 through R3/C18. Data from column 1 is stored in R1/C1, from column 2 in R1/C2, and so on through the card until data from column 80 is stored in R3/C18.

With this fixed storage area, the data punched in each column and field of the card can be readily located in storage after the card is read. For example, if Department Number is punched in columns 18-20 of the card, a reference to read storage shows the Department Number is in storage locations R1/C18, R1/C19, and R1/C20, after the card is read. The operand defining this data is R1/C18 – R1/C20.

Print Storage

The print storage area contains 132 locations, from R6/C6 through R10/C13. This storage area

80-COLUMN INPUT-OUTPUT STORAGE AREAS

COLUMN

ROW	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62
3	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80													
4																															
5																															
6						1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
7	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
8	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88
9	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127	128	129	130	131	132	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
11	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
12	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

READ

PRINT

PUNCH

Figure 2-5. Storage Chart with Input/Output Storage Areas.

corresponds serially with the 132 print positions available. The character stored in location R6/C6 is printed in print position 1; the character stored in location R6/C7 is printed in print position 2, and so on, until the character in R10/C13 is printed in print position 132.

Punch Storage

The punch storage area consists of locations R10/C14 through R12/C31, corresponding to punching columns 1 through 80.

Defining Input-Output Storage

To allow the user to define only a portion of input-output storage, a two-hub area on the connection panel is provided, and is labeled END. The two hubs are assigned to Read (RD), and Print (PR). Signals to these hubs serve to stop and action with which they are associated.

Signals delivered to the END hubs must always come from the exit hubs of Address Combines. END-RD is signaled from a single storage location through one Address Combine. END-PR is signaled from two storage locations through two Address Combines. Details of wiring each of the END hubs are given in the following paragraphs.

If fewer than 80 columns of information are needed for processing, the read action can be stopped at any column. Cards are read serially one column at a time beginning with Column 1. As the information is read, it is translated and transferred to storage a column at a time. If reading is to be stopped short of column 80, the storage address of the last column to be read is wired through an Address Combine to END-RD.

For example, only the information punched in columns 1 through 60 is needed for processing. END-RD must be signaled by the storage address that is equivalent to column 60, namely R2/C29. Figure 2-6 shows the wiring to end reading at this location.

END-PR need not be wired unless one of the following conditions is present:

1. The printer in use has less than the full capacity of 132 print positions.

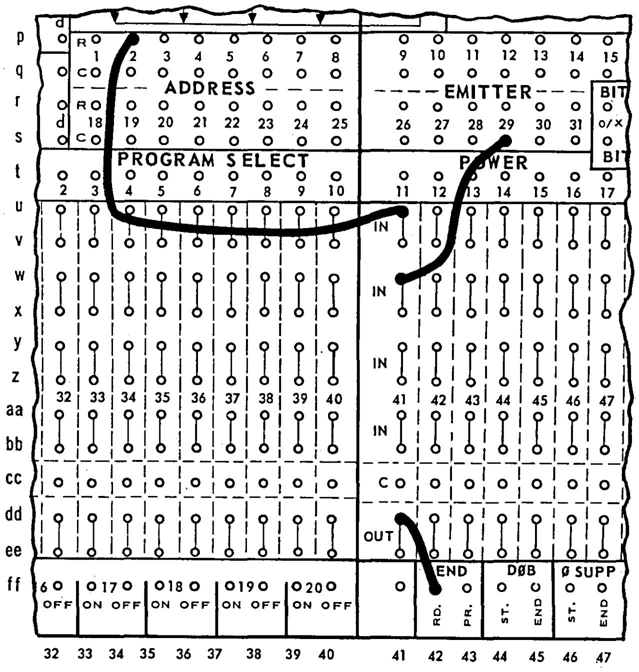


Figure 2-6. Wiring to END-RD at Column 60

2. A print line of less than maximum length is required.

When less than a maximum line is to be printed, END-PR is signaled from the storage locations of the last odd and the last even positions to be printed. Wiring must be through two Address Combines.

For example, if a printer with only 110 print positions is used, and all of the 110 positions are to print, addresses R9/C21 (position 109) and R9/C22 (position 110) are wired to two Address Combines. The OUT hubs of the Address Combines are wired to END-PR (Figure 2-7).

If a maximum printer is to print a line of only 115 positions, addresses R9/C26 and R9/C27 (positions 114 and 115) are wired to END-PR through Address Combines.

Major benefits can be realized through the use of the END-hubs: END-RD releases that portion of read storage that follows the last required location. These released locations become a part of working storage. In addition, the reduction in the number of card columns read saves time that can be used for processing between cards, or can be used to reach a maximum reading speed of 400 cards per minute.

END-PR releases part of usual print storage for working storage.

For purposes of illustration, assume the following input-output requirements:

READ - COLUMNS 1 THROUGH 41

PUNCH - NO PUNCHING IS REQUIRED

PRINT - POSITIONS 1 THROUGH 92

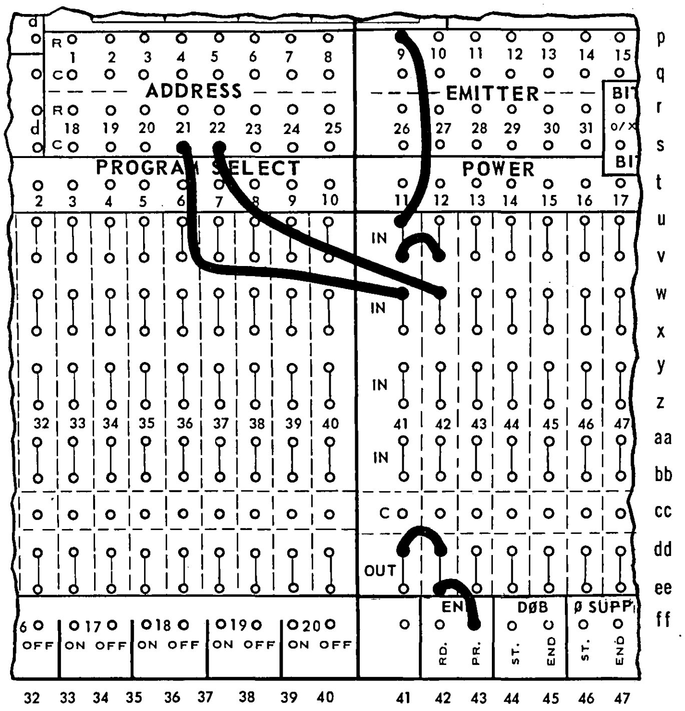


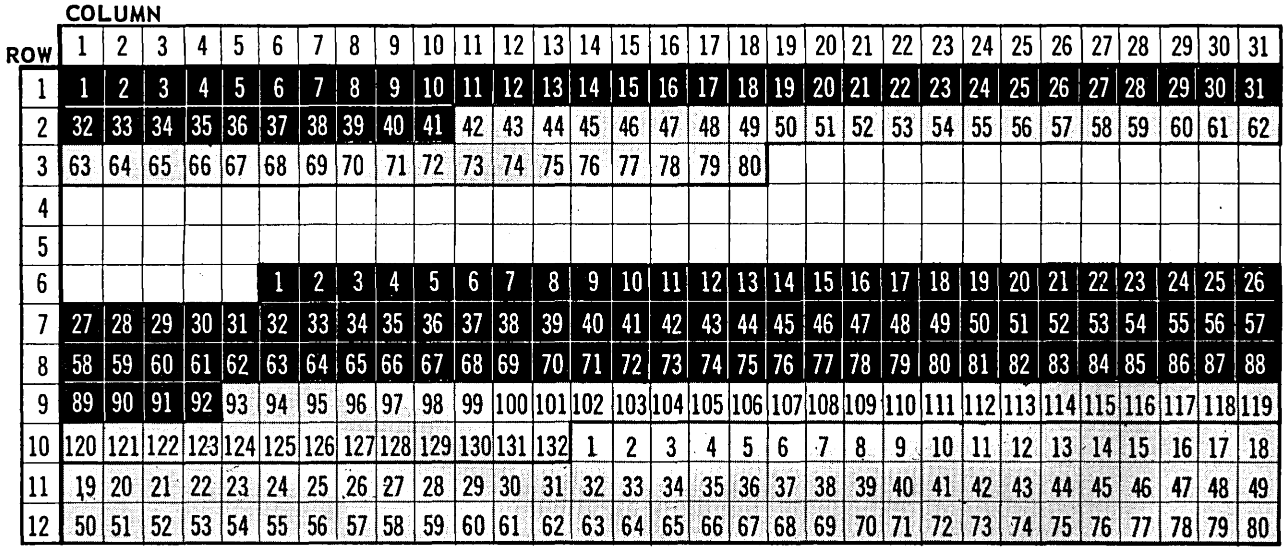
Figure 2-7. Wiring to END-PR at Column 110

Figure 2-9 shows the storage chart modified to illustrate the input-output storage locations that are released through proper wiring of the END-RD and -PR hubs.

Input-output storage areas are automatically protected against the possibility of overlaying data in those areas. This is done in the read storage area by replacing previous data with new information being read. If a column in the card is blank, the related storage location is automatically cleared to a space. When END-RD is ordered at a specific location in the read storage area, clearing and replacing stops after that location is processed. In the same way, a Print-Execute order releases information from print storage for printing and clears the storage area to spaces. Print storage is then ready to accept new information.

Punch storage provides the same automatic clear feature after a punch order unless Punch Hold is used. Punch Hold inhibits clearing of punch storage.

80-COLUMN INPUT-OUTPUT STORAGE AREAS



	COLUMN																													
ROW	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62
3	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80													
4																															
5																															
6						1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
7	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
8	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88
9	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127	128	129	130	131	132	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
11	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
12	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

■ AREA USED

▨ AREA RELEASED

Figure 2-8. Storage Chart with Modified I/O Storage Areas.

II. LOGIC

The operation of the 1004 processor is under the control of a series of instructions wired on the connection panel. This series of instructions is referred to as the PROGRAM and each individual instruction is called a STEP. Each step indicates that data is to be manipulated in some manner – moving it to another location, adding it to a total, and so forth – for purposes of creating the desired content and format of output, either printed or punched. The preparation of this series of steps is called PROGRAMMING.

The 1004 processor uses "two-address logic"; that is, each step specifies two storage areas which are involved in the step, as well as the process to be performed on the data in those locations. The contents of the storage locations specified are referred to as OPERAND 1 and OPERAND 2.

MULTIPLE OPERATIONS

The 1004 processor is designed so that several operations may be performed on the same program step. Knowledge of this concept of multiple operations is essential to the optimum use of the system. With the multiple operations ability, it is possible to transfer information, insert symbols such as dollar signs, periods, and commas, suppress unnecessary zeros, and to advance the paper in the printer a predetermined number of lines, print a line of output, punch a card, and read the next card – all in a single Processor step.

Similarly, it is possible, in a single step, to perform an arithmetic process and to vary the program according to the sign of the result. Also, information may be transferred while suppressing bits or characters in any location in the operand, or stripping off all zone bits of an operand. With each of these multiple-operations abilities, the programmer can combine any or all of the input-output functions of reading, printing, and spacing paper. The multiple-operations feature is especially useful in the editing of print output.

A chart showing all of the permissible combinations of operations within a single processor step is shown in Figure 2-30, page 67.

PROCESSES

The 1004 processor has two categories of processes – arithmetic and logical. Arithmetic processes include add and subtract, both algebraic and absolute, and compare. When performing algebraic add or subtract or numeric compare operations, the machine regards the presence of an X-bit in the least significant location of each operand as a negative sign. When the machine performs absolute add or subtract, both operands are considered positive and any sign indication is ignored.

There are two kinds of comparisons – numeric compare, and alphanumeric compare. Numeric compare is a comparison of relative magnitude. A numeric compare results in one of three indications:

Operand 1 is greater than Operand 2 (>), Operand 1 is equal to Operand 2 (=), or Operand 1 is less than Operand 2 (<). (Zero and space are considered equal during numeric compare.)

Alphanumeric compare is a comparison of bit patterns of the two operands. An alphanumeric compare results in one of two indications: match (M), the bit patterns are identical, or non-match (NM), the bit patterns differ. (Zero is non-match with space during alpha compare.)

The compare circuits can also be used to determine the sign of an arithmetic result. Arithmetic results can be tested for any of three possible indications: the result is positive (+), the result is negative (–), or the result is zero (Ø).

The Clear feature may be used in conjunction with any arithmetic or compare process. When Clear is wired from a Step Output hub in connection with an arithmetic or a compare process, Operand 1 storage locations are cleared to spaces upon completion of that process.

The logical processes are transfer variations. There are seven types of transfer processes which can be performed by the 1004 processor; They are:

1. TRANSFER (TRF)
   Moves data from one location to another.

2. ZERO-SUPPRESS WITH SPACE FILL (Ø – Δ)
   Moves data from one location to another, replacing nonsignificant zeros with spaces.

3. ZERO-SUPPRESS WITH ASTERISK FILL (Ø –*)
   Moves data from one location to another, replacing nonsignificant zeros with asterisks.

4. ZONE-DELETE (ZD)
   Moves data from one location to another, removing all bits in the zone levels of core storage.

5. (a) ZONE-DELETE WITH SIGN (ZDS)
   (Ascending only)
   Moves data from one location to another, removing all bits in the zone levels of cores, except in the least significant location. In this location an X-bit indicates a negative sign.

   (b) DELETE ZERO BALANCE (DØB)
   (Descending Only)
   Moves data from one location to another except when a zero-balance indicator is detected. In that case, space codes are delivered to Operand 2 locations in place of the data.

6. INSERT (IN)
   Moves data from one location to another, inserting specified characters at specified locations.

7. SUPERIMPOSE (SI)
   Moves data from one location to another, superimposing bits or characters on the contents of specified locations.

The Clear feature may be used in conjunction with any of the seven transfer options. When Clear is wired from a Step Output hub in connection with any transfer option, Operand 1 locations are cleared to spaces upon completion of the step.

NOTE: ANY STEP IN THE PROGRAM NOT WIRED TO AN ARITHMETIC OR A LOGICAL PROCESS MUST BE WIRED TO *NO PROCESS*. Any attempt to perform a step without specifying either an arithmetic or logical process or NO PROCESS will cause the processor to stall.

The transfers are discussed in detail in Chapter 4.

Input-Output Functions

The machine uses another class of operations called INPUT-OUTPUT FUNCTIONS. These operations control reading and punching of cards and spacing and printing of forms. The input-output functions are:

READ
: Alerts the reader to perform a Read operation when an Execute order is given. When Execute is ordered, a card is read and the data from the card enters storage.

PRINT
: Alerts the printer to perform a Print operation when an Execute order is given.

EXECUTE
: Causes reading and printing operations previously alerted to be performed. Read and Print may be signaled simultaneously with Execute, or prior to it.

PUNCH HOLD
: Causes information in punch storage to be punched into a card without altering the contents of punch storage.

PUNCH CLEAR
Causes information in punch storage to be punched into a card and clears the punch storage area to spaces.

PUNCH TEST
Tests whether a punching operation is in progress. If the punch is operating, step advance is delayed until punching is completed.

SPACE 1
Advances the paper form in the printer one line.

SPACE 2
Advances the paper form in the printer two lines.

SKIP 1, 2, 4
One or more of these hubs may be signaled simultaneously to advance the paper form in the printer. Paper advances to a line equivalent to a point in the carriage-control paper-tape loop that is punched in the same pattern as the hubs signaled. A total of seven unique terminal points may be defined by paper loop punching.

NOTE: If a Skip or Space instruction is given without a Print instruction, the skip or space is executed immediately. If, however, a Print instruction is given along with the skip or space instruction, the skip or space is delayed until the Print order has been executed.

OPERANDS

Each of the two operands of a step is defined by the row and column addresses of the MSL and LSL of the operand. There is no "fixed word length" in the 1004 processor – an operand may consist of any number of consecutive storage locations from 1 to 961. Only one rule must be observed with regard to the size of the operands:

THE NUMBER OF LOCATIONS SPECIFIED IN OPERAND 2 MUST ALWAYS BE EQUAL TO OR GREATER THAN THE NUMBER OF LOCATIONS SPECIFIED IN OPERAND 1.

In connection with this rule, it is important to consider the INSERT TRANSFER process as a special case. The number of characters to be inserted must be added to the number of storage locations defined as Operand 1 to determine the minimum number of storage locations to be addressed as Operand 2. For example, if a 5-digit operand is to be transferred, inserting a dollar signs and a decimal point, at least seven locations must be specified for Operand 2.

When an add process is performed, the contents of Operand 1 are added to the contents of Operand 2. The sum is placed in the locations specified by Operand 2. This process is represented as:

$$(OP1) + (OP2) \longrightarrow OP2$$

When a subtract process is performed, the contents of Operand 1 are subtracted from the contents of Operand 2. The difference is placed in the locations specified by Operand 2. This process is represented as:

$$-(OP1) + (OP2) \longrightarrow OP2$$

When any transfer process is performed, the contents of Operand 1 are transferred to the locations specified by Operand 2. This relationship is represented as:

$$(OP1) \longrightarrow OP2$$

In each of the above examples, the original contents of Operand 2 are replaced when the process is performed. Therefore, if it is necessary to retain the original contents of Operand 2, a transfer step must be executed in advance of the operation desired to make that information available at another location.

In the compare processes, Operand 1 is compared to Operand 2. Numeric comparisons compare Operand 1 with Operand 2 to find if Operand 1 is greater than, equal to, or less than Operand 2. In alphanumeric comparisons, Operand 1 is compared with Operand 2 to determine if Operand 1 matches Operand 2. The original contents of Operand 1 and Operand 2 are not altered.

Sign

In performing the numeric compare process and algebraic add and subtract, the machine regards the X-bit of the least significant location of the operand as the sign indication for the operand.

The presence of an X-bit in this location is regarded as a minus sign, and the absence of a bit as the plus sign. This fact is important when programming the machine.

It is common practice in 80-column card design to indicate a negative sign by means of an X-punch in the least significant column of the card field. When this is true, the translation of input as the card is read provides the necessary bit in storage automatically. However, when the negative sign for a field is indicated by some punch other than the X-punch in the least significant column, it is necessary to edit the input to insert the appropriate bit in the LSL of the field as it appears in storage. The most common method of accomplishing this editing is to transfer the card field from read storage to working storage, and to superimpose an X-bit on the LSL for a negative field.

STEP SEQUENCE

The basic 1004 processor is equipped with 31 steps. Steps are executed successively unless ordered otherwise. For example, the machine executes the process specified in Step 1 and, unless directed otherwise, proceeds to Step 2. The program advances sequentially through the 31 allotted steps performing the processes requested. After Step 31 is completed, the step sequence automatically returns to Step 1. If, at any time, the programmer wishes to change this standard sequence, Step Sequence Change is signaled on the connection panel to advance to the step the programmer desires. The Step Sequence Change feature is discussed in a chapter of this section entitled CONNECTION PANEL.

A start hub is available to begin a program at any desired step. With this hub wired to a Step Sequence Change hub, the program begins at that step.

The basic machine complement of 31 steps can be optionally increased by the addition of either 16 or 31 steps. If the 16-step increment is included, the additional steps are numbered 32 to 47. If the 31-step increment is included, the steps are numbered 32 to 62. In either case, in order to advance from Step 31 to Step 32, Sequence Change must be wired. Unless otherwise specified, step processing is automatically advanced by one at the completion of each step cycle until the last step in the additional group is completed. In a 47-step machine, if Step 47 is performed without a signal to Step Sequence Change, the machine will stall. In a 62-step machine, the sequence returns automatically from 62 to 32 unless Sequence Change is signaled.

Step Sequence Change may occur in either direction: that is, advancing to a higher step or returning to a lower step. For example, the program may be directed at Step 9 to return to Step 5, or advance to Step 18. Under certain circumstances, the program could be directed to re-execute Step 9. In general, programs tend to follow the pattern of sequential step advance because that line of flow is easiest to create and understand. However, the programmer has complete control over the step sequencing according to the needs of the application.

III. CONNECTION PANEL

The connection panel of the UNIVAC 1004 processor is shown at the back of this manual, Foldout 1. A panel in use is held by a compressor assembly in a recess at the right end of the machine. Data-processing routines are wired on connection panels. A connection panel can only be wired while the panel is removed from the compression device.

Wires are held in place on the connection panel by inserting the terminals in openings called hubs. The hubs are identified according to the internal wiring with which each is associated. There are emitting hubs and receiving hubs. Emitting hubs issue signals which can be wired to receiving hubs to initiate machine functions.

Two types of signals are emitted – high-level signals and low-level signals. For ease of reference, high-level signals are referred to as A-Pulse signals and low-level signals as B-Pulse signals.

Only two hub areas of the connection panel emit B-Pulse signals: Address Emitter and Bit Emitter. All other signals emitted are A-Pulse signals.

There are groupings of hubs that of themselves neither emit signals generated within the machine nor receive signals to be directed to internal circuits. These groupings have various purposes such as: directing a signal along one of two pathways, making a single signal available at a number of hubs, or directing a number of signals to a single destination. These groupings carry such designations as Selectors, Distributors, or Collectors.

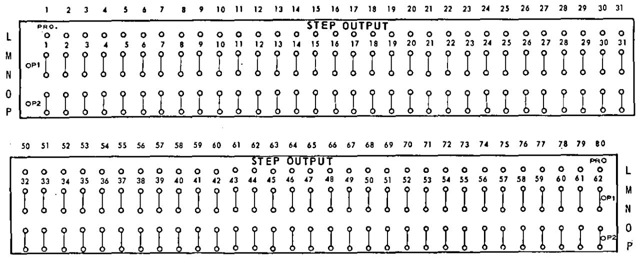
Hub sections are located in the connection panel in the foldout by coordinates made up of letters and numbers printed along the top, bottom, and sides of the diagram. In the following pages, the coordinates are listed for easy reference to the foldout.

STEP OUTPUT

LOCATION: L - P, 1 - 31 and 50 - 80

TYPE: Emitting

There is a maximum of 62 processing steps available in the 1004 processor. A standard machine has 31 steps. The additional 31 steps may be either factory or field installed. For each step there is a column of 5 hubs in the Step Output section of the connection panel with a single hub at the top of the column and the remaining four hubs wired internally in two pairs. The Step Output hub columns are divided into two groups, the first group is numbered from 1 to 31 to identify each column with a processing step. The second group of 31 columns is numbered from 32 to 62, identifying the additional 31 steps.



The single hub at the top of each column is labeled PRO and is wired, usually through Collectors, to the arithmetic or logical processes to be done during that step. The center pair of hubs, labeled OP1, is wired through a Distributor to define Operand 1. The lower pair of hubs, labeled OP2, is wired through a Distributor to define Operand 2.

Step Output hubs can also be wired to other receiving hubs such as Program Select Control hubs, Comparator inputs, or other destinations.

The marking of certain hubs of Step Output as PRO, OP1 and OP2 is arbitrary and only for convenience in wiring. The hubs serve equally as sources of Step Output signals. Back-circuit protection is provided between the PRO hub and the OP1 and OP2 pairs of hubs.

Processing steps occur successively, one at a time. During each step, the processor is directed to perform a part of the total program. The machine recognizes that the Step Outputs are in two groups, and unless otherwise directed, will return to Step 1 after Step 31 is completed, and to Step 32 after Step 62 is completed. The normal order of successive step processing can be interrupted by wiring to Step Sequence Change. This wiring is explained in later paragraphs.

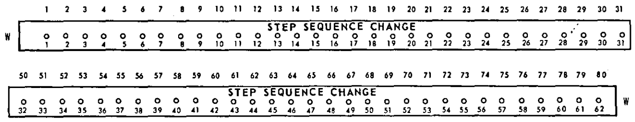
NOTE: THE SIGNAL FROM STEP OUTPUT CANNOT PASS THROUGH MORE THAN TWO DIODES IN SERIES BEFORE REACHING ITS DESTINATION.

STEP SEQUENCE CHANGE

LOCATION: W, 1 - 31 and 50 - 80

TYPE: Receiving

The Step Sequence Change section of the connection panel consists of one hub for each processing step with which the machine is equipped. Each hub is numbered with the step to which it is related. When, during any step, the conditions require a change in the normal serial step sequence of processing, a signal is delivered to the Step Sequence Change hub of the step which is to occur next.



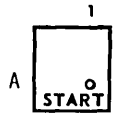
There are no limitations to changing the sequence. Any step can occur next, a higher-numbered step or a lower-numbered step. Changing the step sequence does not in itself cause a delay in processing.

START

LOCATION: A, 1

TYPE: Emitting

The START hub is provided to allow processing to to begin with any step desired. To begin processing, the START hub is wired to the Step Sequence Change hub of the desired initial step. The machine bypasses all preceding steps, regardless of wiring, and begins processing at the step wired from the START hub.

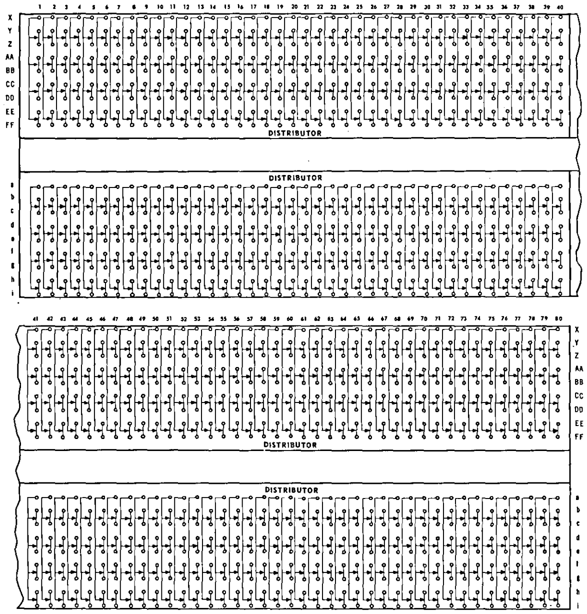


DISTRIBUTOR

LOCATION: X - FF, 1 - 80
a - i, 1 - 80

TYPE: Entrance and Exit

Each of the 160 Distributors consists of a column of 9 hubs: one entrance hub at the top to which a signal may be wired, and four pairs of exit hubs. A signal received at the entrance hub is equally available at each pair of exit hubs. Each pair of exit hubs is diode-isolated from the other pairs.



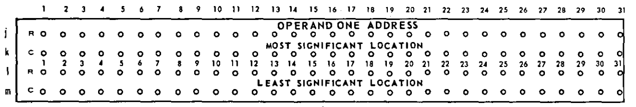
The most common use of a Distributor is to provide four exits for a Step Output signal to four operand address hubs.

OPERAND 1 ADDRESS

LOCATION: j - m, 1 - 31

TYPE: Receiving

The Operand 1 Address section of hubs consists of four rows of 31 hubs each and is used to define the storage location of the first operand of a step. The two upper rows, labeled R and C, are signaled to specify the most significant location (MSL) of Operand 1. The lower two rows, also labeled R and C, are signaled to specify the least significant location (LSL) of Operand 1.



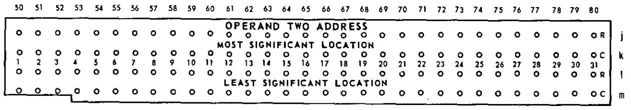
The hub labeled R specified the ROW of an addressed location in storage, and a hub labeled C specifies the COLUMN of the addressed location in storage.

OPERAND 2 ADDRESS

LOCATION: j - m, 50 - 80

TYPE: Receiving

The Operand 2 Address section of the connection panel is used in the same manner as the Operand 1 Address section. Wiring to the Operand 2 Address hubs defines the MSL and LSL of Operand 2 of each step.

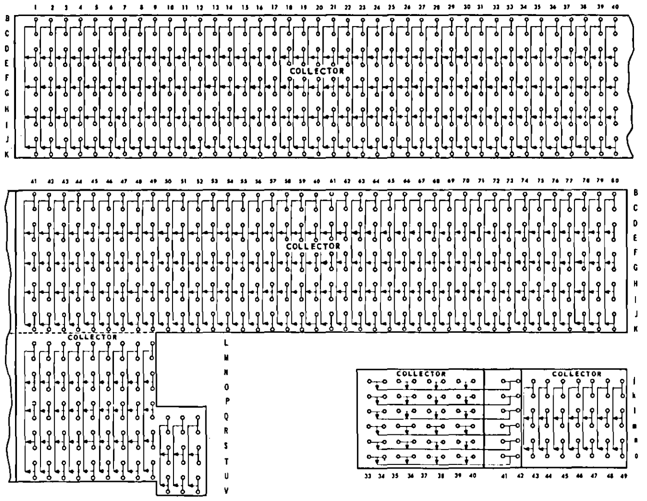


COLLECTOR

LOCATION: B - K, 1 - 80
L - U, 41 - 49
*Q - V, 50 - 52
j - o, 33 - 42
*j - p, 43 - 49

TYPE: Entrance and Exit

Each Collector consists of a related assembly of ten hubs. The units located in the hub areas marked with an asterisk are smaller. A ten-hub assembly provides a pair of exit hubs and four pairs of entrance hubs. A signal received at any one of the entrance pairs is available only at the exit pair. Each pair of entrance hubs is diode-isolated from the other pairs of entrance hubs.



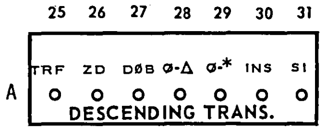
A Collector is most frequently used to expand a single receiving (signal destination) hub so that it may be signaled by more than one source. To provide more than four entrances to a single receiving hub, two or more Collectors can be made common by interconnecting the exit pairs of the Collectors needed.

DESCENDING TRANSFER

LOCATION: A, 25 - 31

TYPE: Receiving

The Descending Transfer section of hubs is normally wired from the PRO hub of Step Output, which may have been sent through a Collector. Any of the seven options can be wired to initiate the type of transfer desired. The combinations of option wiring allowable are shown on the chart in Figure2-30, page 67 .



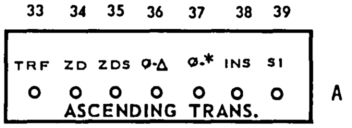
A portion of this manual entitled TRANSFER explains these functions in detail.

ASCENDING TRANSFER

LOCATION: A, 33 - 39

TYPE: Receiving

The Ascending Transfer section of hubs is normally wired from the PRO hub of Step Output, which has been sent through a Collector. Any of the seven options can be wired to initiate the transfer desired. The combinations of option wiring allowable are shown on the chart in Figure 2-30 page 67 .



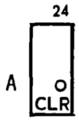
A portion of this manual entitled TRANSFER explains these functions in detail.

CLEAR

LOCATION: A, 24

TYPE; Receiving

The CLEAR hub is wired from a Step Output hub in conjunction with any transfer, arithmetic, or compare process. When signaled the CLEAR hub causes Operand 1 storage locations to be cleared to spaces upon completion of the specified process.

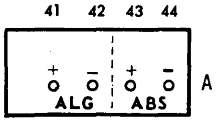


ALG/ABS (Algebraic/Absolute)

LOCATION: A, 41 - 44

TYPE: Receiving

The hub section labeled ALG/ABS contains four hubs for the arithmetic processes which are performed by the processor. Those processes are Add (+) and Subtract (–), Algebraic and Absolute. An arithmetic hub is normally wired from a PRO hub of Step Output. This wiring may be through a Collector.



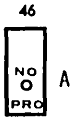
A portion of this manual entitled ARITHMETIC PROCESSES discusses these functions in detail.

NO PRO (NO PROCESS)

LOCATION: A, 46

TYPE: Receiving·

The No Process hub must be wired if an arithmetic or a logical process is not specified for a step. It is normally wired from the PRO hub of Step Output.

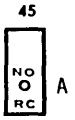


NO RC (No Recomplement)

LOCATION: A, 45

TYPE: Receiving

The No Recomplement hub is wired to suspend automatic recomplementation of a complementary result obtained during an arithmetic step. NO RC is wired from a PRO hub of Step Output which is also wired to the arithmetic process desired. The signal is wired from PRO through a Collector to an ALG/ABS hub and to NO RC.

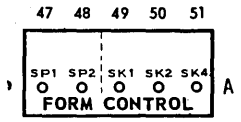


FORM CONTROL

LOCATION: A, 47 - 51

TYPE: Receiving

The Form Control area on the connection panel consists of five hubs. These hubs are signaled to initiate certain paper movements in the printer. Two hubs are labeled SP1 and SP2. A signal to one of these hubs causes paper to line-space one or two lines depending on the option used. Three hubs are labeled SK1, SK2, and SK4. Each of these hubs is associated with one of the channels of the carriage control paper loop. One, two, or all three of these hubs can be signaled by a single pulse. When signaled, paper advances to a point equivalent to the location in the paper loop punched with an identical pattern.



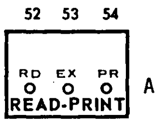
A complete discussion of paper spacing is given in a portion of this manual entitled FORM CONTROL.

READ-PRINT

LOCATION: A, 52 - 54

TYPE: Receiving

The three-hub area on the connection panel marked Read-Print is used to establish communication between the reader section and the processor, and between the printer section and the processor.



The hub labeled RD, when signaled, alerts the reader section of the machine. A signal to this hub alone initiates no read action, however. (See explanation of EX given below.)

The PR hub alerts the printer section of the machine to perform a print operation. No printing action takes place upon receipt of a signal at this hub alone. (See explanation of EX given below.)

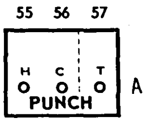
The execution of read and/or print occurs only upon signaling the hub marked EX. When a signal is received at this hub, any alerted operation – Read or Print, or a combination of Read and Print – is executed. The signal to the EX hub may be delivered at the same time that signals are received by the function hubs or it may be delayed until any later program step.

PUNCH

LOCATION: A, 55 - 57

TYPE: Receiving

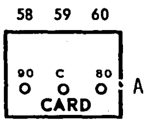
The three Punch hubs labeled H (hold), C (clear), and T (test) are additional output functions. A signal to the Hold hub initiates punching of a card and holds the information in punch storage. A signal to the Clear hub initiates punching while clearing the information from punch storage. The Test hub is used to determine whether or not the punch has completed a previous instruction. If punching is still in progress when the test hub is signaled, the step that delivered the signal is completed, but advance to another step is blocked until punching is completed. The signals to these punch hubs are from the Process hubs of Step Output.



CARD

LOCATION: A, 58 - 60

TYPE: Emitting and Receiving



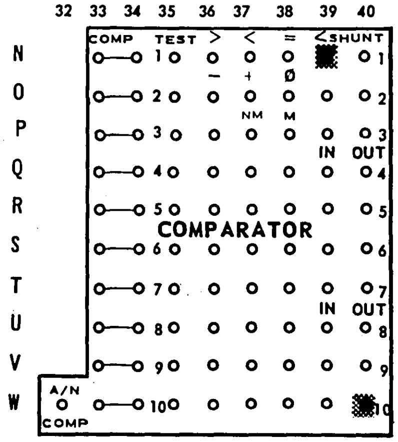
The three hubs of the section are labeled CARD 90, C, and 80. In all instances, the interconnection of the C and 80 hubs is required for the 1004 processor, 80-column.

COMPARATOR

LOCATION: N - W, 33 - 40
*W, 32

TYPE: Emitting and Receiving

The Comparator area of the connection panel contains the hubs for a maximum of ten compare circuits. Each circuit is represented by a row of eight hubs.



The two hubs marked COMP at the left of each row are common receivers. A signal delivered to these hubs causes three actions to be performed in this order:

1) the compare circuit is cleared of any previous result,
2) the comparison is made,
3) the circuit is conditioned to emit the proper result signal when tested.

The circuit remains so conditioned until a subsequent signal is delivered to the COMP hub.

In order to use the result of the comparison, a signal is delivered to the hub marked TEST. The signal delivered to TEST is emitted from one of the three result hubs according to the condition set up by the comparison. COMP and TEST may be impulsed on the same step. In that case, the signal delivered to TEST is emitted only after the circuit has been conditioned by the result of the comparison on this step.

Three kinds of comparison are possible with a compare circuit.

1. NUMERIC – the contents of Operand 1 are compared to the contents of Operand 2. The signs of the operands are considered, but the zone-bits are ignored. (Ø = Δ)

   Results possible: (Op 1) > (Op 2)
   (Op 1) < (Op 2)
   (Op 1) = (Op 2)

2. SIGN COMPARE can be obtained by wiring both Compare and an arithmetic process on the same step.

   Results possible: Sign of result is positive (+)
   Sign of result is negative (–)
   Sign of result is zero (Ø)

3. *ALPHANUMERIC – a signal must be delivered to the alphanumeric Compare hub, marked A/N COMP at panel location W, 32 at the same time a signal is delivered to COMP. The characters in Operand 1 are compared to the characters in Operand 2. This is a bit-for-bit comparison and Ø ≠ Δ.

   Results possible: OP 1 is identical to OP 2 – Match (M)
   OP 1 is unlike OP 2 – Non-Match (NM)

When TEST is signaled, one of the three result hubs emits as long as the signal remains at TEST. The signal emitted can be routed through not more than two diodes before reaching a destination. This is possible regardless of whether or not the signal into TEST passed through a diode.

The two hubs at the right end of a compare row are labeled < SHUNT and consist of one IN hub and one OUT hub. Use of these hubs is to link one characteristic of the compare circuits together. Linkage is made by wiring the OUT of a lower-numbered Shunt to the IN hub of the next higher numbered shunt. The usual application is to control for changes in designation when multiple levels of totals are being accumulated. When two adjacent compare circuits are linked by wiring together the SHUNT hubs, the lower numbered circuit, upon switching to (OP 1) < (OP 2) forces the higher numbered compare to assume the same state. As long as the lower numbered compare is set at <, the higher numbered compare cannot be changed to any other state.

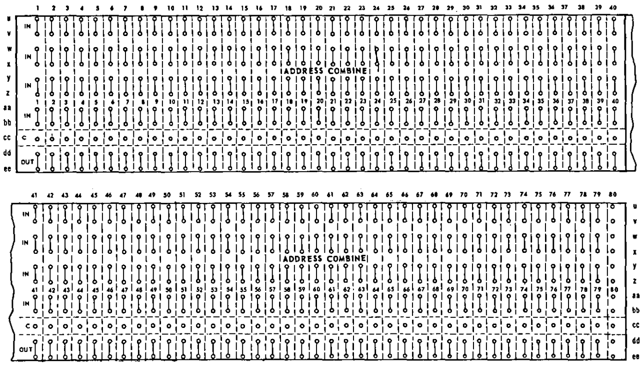
It is possible to link any number of successively numbered compare circuits by wiring the SHUNT pairs together. In this case, each compare circuit controls all linked circuits of a higher number as described above.

ADDRESS COMBINE

LOCATION: u - ee, 1 - 80

TYPE: Emitting and Receiving

There are 80 Address Combines. Each Address Combine consists of a column of 11 hubs; four pairs of IN hubs, a single common (C) hub, and a pair of OUT hubs. The two hubs of each pair are common; that is, each hub of a pair emits or receives in a like manner. The purpose of the Address Combines is to emit an A-pulse when two or more B-pulse signals are received at the IN hubs. To emit the A-pulse signal, an Address Combine must receive as many B-pulse signals as there are IN hubs wired from B-pulse sources. The A-pulse signal is available at the OUT hubs of an Address Combine only as long as all needed B-pulse signals are present.



It is possible to inhibit the A-pulse signal from an Address Combine even though all needed B-pulse signals are present. This block is established if an A-pulse signal is wired to an IN hub of the Address Combine. An A-pulse signal delivered to the IN hub of an Address Combine blocks A-pulse emission from the OUT hubs of that Combine.

An Address Combine can be expanded through the use of Collectors and the common hub of a Combine. If the exit hub of a Collector is wired to the common hub of an Address Combine, the entrance hubs of that Collector become IN hubs for the Address Combine. If a Collector with four pairs of entrance hubs is wired to the common hub of an Address Combine in this manner, the IN hub capacity of the Combine totals eight pairs.

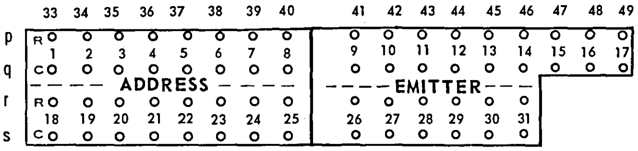
The A-pulse signal from the OUT hub of an Address Combine can be wired to any portion of the panel which accepts A-pulse signals. Frequently, wiring is to ON hubs of Program Select Control or to Bit and Character Generators.

ADDRESS EMITTER

LOCATION: p - s, 33 - 46
p, 47 - 49

TYPE: Emitting (B-Pulse signals)

There are 31 hubs representing the R coordinates of storage and 31 hubs representing the C coordinates of storage in the Address Emitter section of the connection panel. When any operation is in progress, and any specific location within Operand 2 is addressed, the corresponding Address Emitter hubs emit. *The Address Emitters concern only the Operand 2 locations.*



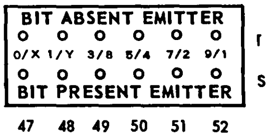
Address Emitter hubs are usually wired to instruct the machine to insert characters, start or stop certain operations, determine control punching, and so forth. *The B-Pulse signals from Address Emitter hubs can be wired only to an Address Combine destination.*

BIT PRESENT EMITTER – BIT ABSENT EMITTER

LOCATION: r - s, 47 - 52

TYPE: Emitting (B-Pulse signals)

Two rows of six hubs each make up the Bit Emitter section of the connection panel. The hubs in each row represent the six-cores of any storage location, and are labeled with the 80- and 90-column values of each core. (The 90-column designations are ignored in the 1004 processor,80 column.)



If a related core in an Operand 2 storage location is magnetized to represent a "no-bit" condition, the hub representing that core in the ABSENT row emits. If a core in an Operand 2 storage location is magnetized to represent a "bit," the related hub in the PRESENT row emits.

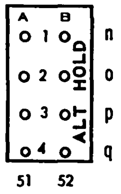
If a storage location is searched to determine if *only* the core that represents a 1 is magnetized to store a bit, the 1 hub of the PRESENT row is wired, and all hubs except the 1 hub of the ABSENT row are wired. If only the 1-value core is storing a bit, only the 1 hub of the PRESENT row emits, and all hubs except the 1 hub of the ABSENT row emit. If a location is searched to determine if the 1-value core is storing a bit, regardless of the condition of the remaining cores, only the 1 hub of the PRESENT row need be wired. The locations searched must be those defined by Operand 2. The Bit Emitter section can work only with Operand 2 locations.

The B-Pulse signals emitted by the Bit Emitter section must be wired to an Address Combine. These signals are usually used to determine control punching in cards in conjunction with Address Emitter signals.

ALT HOLD (Alternate Hold)

LOCATION: n - q, 51 - 52
TYPE: Emitting

The Alternate Hold section of the connection panel consists of four sets of hubs labeled 1 through 4. Each set has a hub labeled A and a hub labeled B. These hubs are related to four switches on the Control Panel of the Processor, which are also numbered 1 through 4. Each switch has two positions, Position A and Position B.

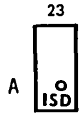


Either the A hub or the B hub of each pair emits a signal at all times, depending on the setting of the four switches on the panel. Wiring from these hubs is usually to Selector Pick-Up hubs.

ISD (Inhibit Selector Delay)

LOCATION: A, 23
TYPE: Receiving

This single hub is wired from the Step Output to eliminate the normal 5 millisecond delay that occurs when a signal is received at an ON or OFF hub of a Program Select.



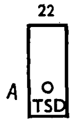
TSD (Test Selector Delay)

LOCATION: A, 22
TYPE: Receiving

If the ISD hub is not signaled, a 5.0 MS delay is started whenever a Program Select is turned ON or OFF. Starting the delay does not in itself cause a delay in step advance. To make the delay effective, it is necessary to test the condition of the delay timer. If the timer is still operating, step advance from the test step is delayed until termination of the 5.0 millisecond period.

Testing the selector delay is accomplished by wiring a Step Output signal to the TSD (Test Selector Delay) hub. The program step on which the program select is signaled or any subsequent step may be wired to TSD. A separate program step is not required for testing the selector delay. The test can be made in conjunction with any other step process or function.

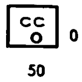
If Program Selects are being turned ON or OFF on a number of steps in a program and the Selector Delay is to be effective on the same steps, TSD can be wired from the H (Hold) hub.



CC (Cycle Clear)

LOCATION: o, 50
TYPE: Emitting

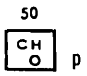
This single hub emits at the start of the execution of a read cycle at the time Cycle Hold is not emitting. Usual wiring is to the OFF hubs of Program Selects.



CH (Cycle Hold)

LOCATION: p, 50
TYPE: Emitting

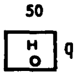
This single hub emits a signal at all times except momentarily at the beginning of each read cycle when Cycle Clear emits. Wiring is usually to Selector Pick-Up hubs to hold the selector relays until the beginning of the next read cycle.



H (Hold)

LOCATION: q, 50

TYPE: Emitting



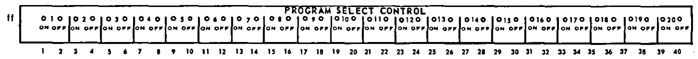
This single hub emits as long as DC power is supplied to the machine. Usual wiring from this hub is through control selectors to Selector Pick-Up hubs.

PROGRAM SELECT CONTROL

LOCATION: ff, 1 - 40

TYPE: Receiving

The Program Select Control section consists of 20 sets of hubs. Each set contains an ON hub and an OFF hub. A signal wired to the ON hub of a Program Select Control set of hubs creates a continuing signal at the related Program Select Power hub located at coordinates t, 31 – 50. Wiring to the ON hub of a Program Select Control pair of hubs is usually from an Address Combine or a Comparator. The signal emits from the related Power hub until a signal is received at the OFF hub of the Program Select Control.



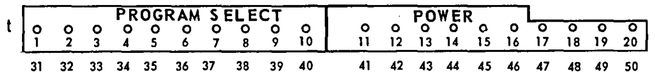
A signal delivered at either the ON or OFF hub initiates a delay which assures that at least 5 milliseconds will elapse before the start of the next processing step. This delay allows time for any affected selector relays to assume a new state. This delay can be eliminated by use of the inhibit delay function, previously explained.

PROGRAM SELECT POWER

LOCATION: t, 31 - 50

TYPE: Emitting

These hubs emit when the associated Program Select Control ON hub is signaled. The signal emitted by these hubs is ended when the associated OFF hub of Program Select Control is signaled.



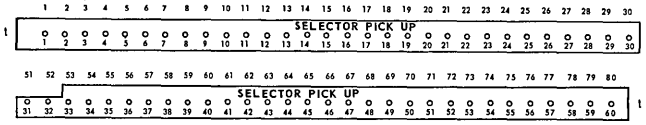
The signal emitted from Program Select Power is usually used to pick up selectors, but can be wired to any receiving hub.

SELECTOR PICK-UP

LOCATION: t, 1 - 30
t, 51- 80

TYPE: Receiving

There are 60 Selectors available for the 1004 processor. Each selector has a hub in the Selector Pick-Up section of the control panel which, when signaled, causes the associated selector to assume the select position. Selector Pick-Ups are numbered 1 – 60 identifying each with its related Selector.



Selector Pick-Up hubs are wired from Hold, Cycle Hold, Alternate Hold, or Program Select Power. Signals from the named hubs are the only signals that can be wired to Selector Pick-Up hubs. As long as a signal is present at the Selector Pick-Up hub, the related Selector remains in the select position.

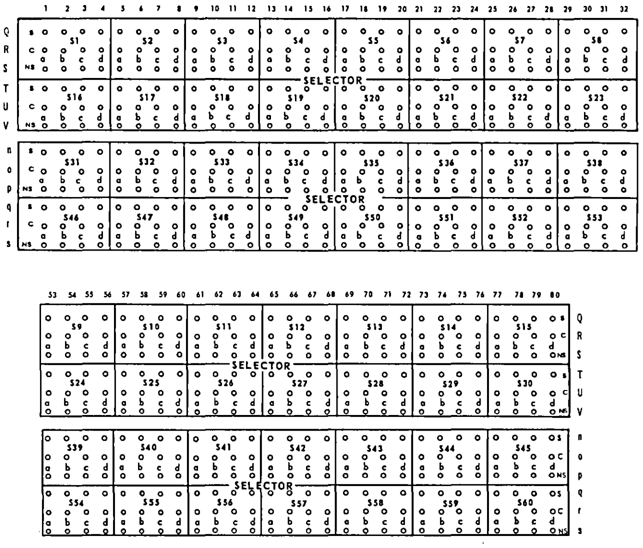
SELECTORS

LOCATION: Q - V, 1 - 32 and 53 - 80
n - s, 1 - 32 and 53 - 80

TYPE: Exit and Entrance

A maximum of 60 Selectors are available with the 1004 processor. The Selectors are numbered from S1 through S60. Each selector consists of four columns of three hubs each. These columns are referred to as POLES, and are labeled a, b, c, and d. The upper hub of each pole is the select (S) hub, each center hub is the common (C) hub, and each lower hub is the non-select (NS) hub. A Selector provides a controllable internal pathway for signal travel. The pathway is normally between the common hub and non-select hub. When a Selector is picked up by a signal at its selector Pick-Up, the signal pathway is altered to exist between the common hub and the select hub.

The main purpose of the Selectors is to allow one machine function if the Selector is picked up, and another machine function if the Selector is not picked up. Each Selector permits two basic types of selection:


SELECTOR
S1 S2 S3 S4 S5 S6 S7 S8
S16 S17 S18 S19 S20 S21 S22 S23
S31 S32 S33 S34 S35 S36 S37 S38
S46 S47 S48 S49 S50 S51 S52 S53
S9 S10 S11 S12 S13 S14 S15
S24 S25 S26 S27 S28 S29 S30
S39 S40 S41 S42 S43 S44 S45
S54 S55 S56 S57 S58 S59 S60
Q R S T U V N O P Q R S
a b c d
C NS

1. *Choosing one of two possible signals*

   In this instance, one signal is wired to the select hub, and another is wired to the non-select hub. If the Selector is not picked up, only the signal to the non-select hub emerges from the common hub. If the Selector is picked up, only the signal to the select hub emerges from the common hub.

2. *Emitting a signal in one of two possible directions.*

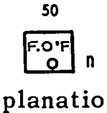
   The signal is wired to the common hub. If the Selector is picked up, the signal is available from the select side of the Selector. If the Selector is not picked up, the signal is available from the non-select side.

F.O'F (Form Overflow)

LOCATION: N, 50

TYPE: Emitting

The F.O'F hub emits a signal during the time a line is being printed if, when the order to execute that line of print was given, the carriage-control loop was positioned so the distinctive overcapacity punching was sensed. The signal from F.O'F may be wired to Program Select Control.



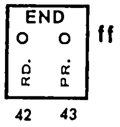
A more detailed explanation of this function is given in the section of this manual entitled FORM CONTROL.

END RD, PR

LOCATION: ff, 42 – 43

TYPE: Receiving

A signal to either of these hubs serves to halt the function with which it is associated. The use of these hubs was explained in the previous Chapter in the Storage section.



1. RD (Read)

   This hub is signaled to end card reading. This hub is wired from an Address Combine to which the storage address of the last column to be read is wired.

2. PR (Print)

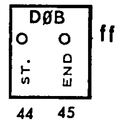
   This hub is signaled to end printing of a line. This hub is wired from Address Combines to which the storage addresses of the last two print positions that are to print are wired.

DØB – ST. – END
(Delete Zero Balance Start and End)

LOCATION: ff, 44 – 45

TYPE: Receiving

These hubs are wired to define the start and end of areas in storage to be cleared selectively by the process Delete Zero Balance. Both hubs receive signals from Address Combines, which are wired from Address Emitter hubs associated with the Operand 2 locations where clearing is to begin and end. Any number of start and end addresses may be signaled within Operand 2. These addresses do not take the place of OP1 and OP2 addresses, which must also be wired.

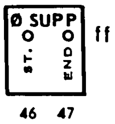


Ø SUPP – ST. – END
(Zero Suppress Start and End)

LOCATION: ff, 46 – 47

TYPE: Receiving

These hubs are wired to define the start and end of a zero suppress field. These hubs are signaled from Address Emitter hubs that define the locations at which zero suppression is to begin and end. Any number of start and end addresses may be given during one program step to accommodate suppressing multiple fields within Operand 2. The Address Emitter signals are wired through Address Combines to the Ø Supp St and End hubs.



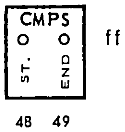
NOTE: The Start and End addresses do not take the place of OP1 and OP2 addresses. The operand addresses must be wired.

CMPS – ST. – END. (Compress Start and End)

LOCATION: ff, 48 – 49

TYPE: Receiving

The two hubs in this area are used to start deletion of Operand 1 locations and to end this deletion. The location following the first location to be deleted is wired to Start Compress, and the location following the last location to be deleted is wired to End Compress. Compress is wired on either a Superimpose or an Insert Transfer, ascending or descending. It is possible to compress without superimposing or inserting, but the wiring must be made through one of these transfer options. Compress is not a process of the 1004 Processor; it is a feature of the Superimpose and Insert Transfer options.

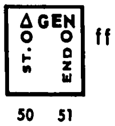


Δ GEN – ST. – END (Space Generate Start and End)

LOCATION: ff, 50 – 51

TYPE: Receiving

The two hubs in this area of the connection panel are used to start the entry of spaces in a series of storage locations and to end such an area. The spaces entered may insert (enter between two characters) or superimpose (enter storage locations replacing characters previously stored). The storage locations to receive the spaces must be within Operand 2 of a step.



Any number of areas may be started and ended within the Operand 2 area defined by the step.

To define the start or the end of an area, a specific address must be defined by wiring from an R and a C hub in the Address Emitter to IN hubs of an Address Combine. The OUT of that Address Combine is wired to the action wanted.

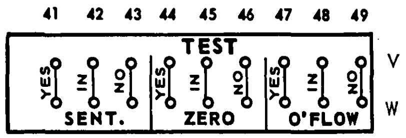
Defining Start and End address for this function does not substitute for operand address definitions for the step.

TEST Ø (Test Zero)

LOCATION: V – W, 44 – 46

TYPE: Entrance and Exit

These hubs are used in conjunction with the Test Location section described above. The IN hub is wired from Step Output causing a signal to exit from either the YES or NO depending on whether or not a zero is present in the location tested. The YES hub exits the signal if a zero is present; the NO hub exits the signal if it is absent. The YES and NO signals can be wired to receiving hubs to initiate desired operations. Δ = Ø for this test.



The YES and NO conditions remain set until the initiation of a new test.

TEST SENT (Test Sentinel)

LOCATION: V – W, 41 – 43

TYPE: Entrance and Exit

These hubs are similar in operation to the TEST ZERO hubs, except that they emit upon the presence or absence of a sentinel in the tested location. Sentinel = X and Y bit combination.

TEST O'FLOW (Test Arithmetic Overflow)

LOCATION: V – W, 47 – 49

TYPE: Entrance and Exit

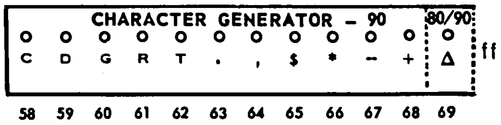
Three pairs of hubs are marked YES, IN, and NO. The two hubs of each pair are interconnected to serve equally.

Each arithmetic step is automatically monitored by the machine to determine if the result contains more digits than there are storage locations defined for Operand 2 of the step. If this condition, called arithmetic overflow, is found, the switch is set to an interconnection between IN and YES. If arithmetic overflow does not occur, the interconnection is between IN and NO. The set condition remains until the next arithmetic (includes compare) step.

A Step Output signal from the same step or a following step may be wired to the IN hub. A signal will then emit from either YES or NO depending on whether or not the overflow did occur. The YES and NO hubs can be wired to receiving hubs to initiate desired operations.

CHARACTER GENERATOR – 90

LOCATION: ff, 58 – 69
TYPE: Receiving

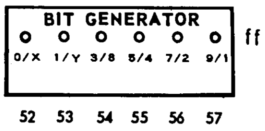


These hubs are used to generate internally 90-column characters. This section of the connection panel is never used with the 80-column processor.

BIT GENERATOR

LOCATION: ff, 52 – 57
TYPE: Receiving

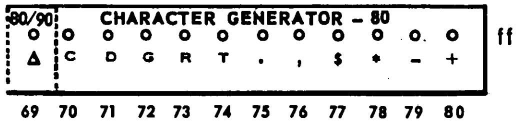
The Bit Generator consists of six hubs representing the six positions of the XS-3 code. (They are also labeled with the 90-column card positions which are ignored when the 80-column Processor is being used.) During an insert or superimpose transfer, the R and C address of the locations of Operand 2 which are concerned with the insert or superimpose are combined and wired to one or more of these hubs. Any character or special code can be internally generated by the proper combination of the Bit Generator hubs. Any number of card code positions can be wired from any number of address locations on one step.



CHARACTER GENERATOR – 80

LOCATION: ff, 69 – 80
TYPE: Receiving

These hubs are wired to generate internally the 80-column characters with which each is labeled. During an insert or superimpose transfer, the R and C addresses of Operand 2 concerned with inserting or superimposing are combined and wired to the hub of this section representing the desired character. The character wired will be inserted or superimposed in the locations wired to the hub. Any number of characters may be inserted or superimposed in any number of locations that are a part of Operand 2 during a single step.

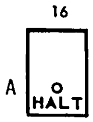


The space character hub at coordinates ff, 69 is common to 80- and 90-column processing, and is so designated. Wiring to this hub for space code generation for either 80- or 90-column processing is permissible.

HALT

LOCATION: A, 16
TYPE: Receiving

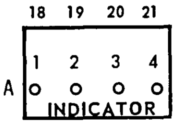
The HALT hub is wired to stop the processor. The HALT hub is usually wired from a Comparator, Test Overflow, Test Zero, or Test Sentinel. When HALT is signaled, the step in progress will be completed but step advance will not occur.



INDICATOR

LOCATION: A, 18 – 21
TYPE: Receiving

These hubs are labeled 1, 2, 3, and 4, and provide a means of indicating the reason for halting. These hubs are associated with display lights on the display panel of the processor. Wiring to these hubs is usually from Step Output which has been routed through a controlling device such as a selector or a comparator. All indicators are normally lit. When signaled, the associated indicator is extinguished.

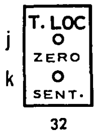


T. LOC (Test Location)

LOCATION: j – k, 32

TYPE: Receiving

Two hubs each of which is associated with a two positions switch. One hub in the T LOC area is marked Ø (zero) and the associated switch is also marked ZERO. The second hub is marked SENT (sentinel) and is associated with a switch marked SENT. The two hubs in this area operate identically and differ only in the stored symbol that operates them. A storage location is defined by wiring from Address Emitter hubs to an Address Combine, the OUT of which is wired to one of these hubs. When this storage location is accessed as a part of Operand 2, the location is tested for the character wanted. If the character is present, the associated switch is set to interconnect IN hubs with YES hubs. If any character except the one specified is present, the switch will be set interconnecting IN with NO hubs.



REPRESENTATIVE WIRING

Example 1

The example given in Figure 2-10 is only to show possible wiring techniques and is neither complete nor part of a useful program. Some, but not all, wires are identified by numbers to aid text reference.

Step 11 is used. The PRO hub of Step 11 is wired (wire 5) to identify the process to be performed, ADD ALG. Wire 1 is connected to a Distributor to identify Operand 1 which is shown to be R9/C1 – R9/C6. Wire 2 goes to a Distributor defining Operand 2 as R5/C3 – R5/C9 (note: Operand 2 must contain as many or more storage locations as Operand 1). Since the process is ADD ALG, it is understood that X-bits in R9/C6 and R5/C9 are used for negative sign indication.

Wire 3 is plugged to the common of Selector 3d. It is assumed that some action (for example, the finding of a control punch) on an earlier step could have picked up Selector 3. If the selector is in a select state, the Step 11 pulse, through wire 4, causes Step 14 to be executed next. If Selector 3 is not select, the machine receives no signal to depart from its usual sequence and Step 12 is the next Step executed.

As Step 11 is executed the machine works with storage location R5/C9, the LSL of Operand 2. The wires numbered 8 from the Address Emitters define the address for a Combine. During this step other locations in Row 5 will be handled but only at R5/C9 time will both emitters provide a signal to Combine 36, and only at this time will a signal emit from the OUT of the Combine. The signal from the OUT is routed thru wire 9 to Test Location for Zero. If R5/C9 contains a zero, the Zero Test switch (hub locations V and W, 44–46) are set for a connection between IN and YES. If a zero is not present, the connection between IN and NO is set. The signal to the IN hub of this switch can be made at any time after it is set.

Step 14 is done if Selector 3 is in a Select condition. This example assumes that this step ends the routine. Wire 6 from Step 14 directs the machine to perform No Process. Wire 7 and its continuation wires further direct the machine to print a line and space one space and to punch a card and clear the punch storage area.

A signal from Step 14 through wire 18 tells the machine to do Step 1 next, restarting the program.

Step 12 is done if Selector 3 is in a Non-select condition. The PRO hub of Step 12 is wired (wire 16) to perform another ADD ALG process. Note

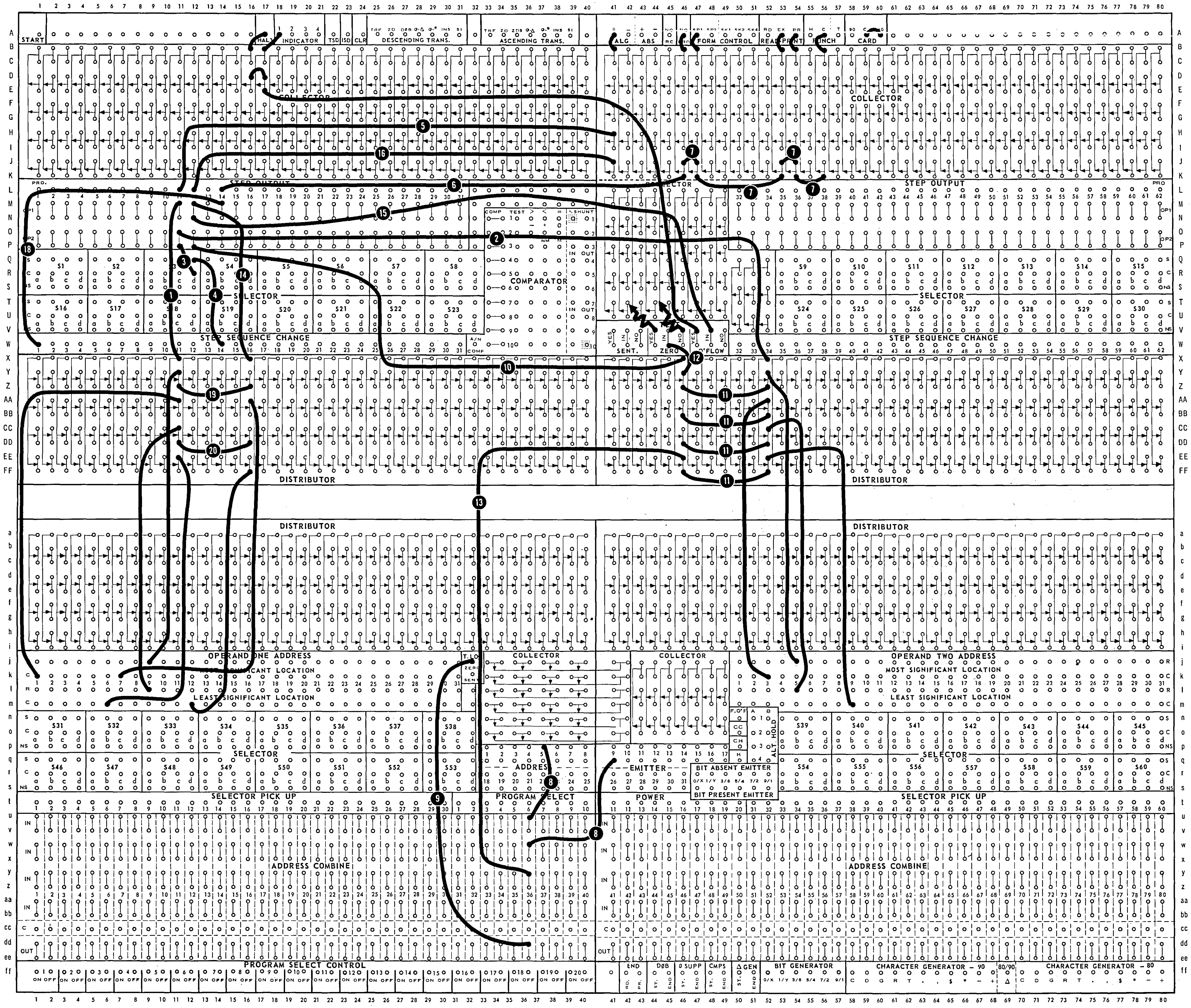


Figure 2-10. Representative Wiring – Example 1

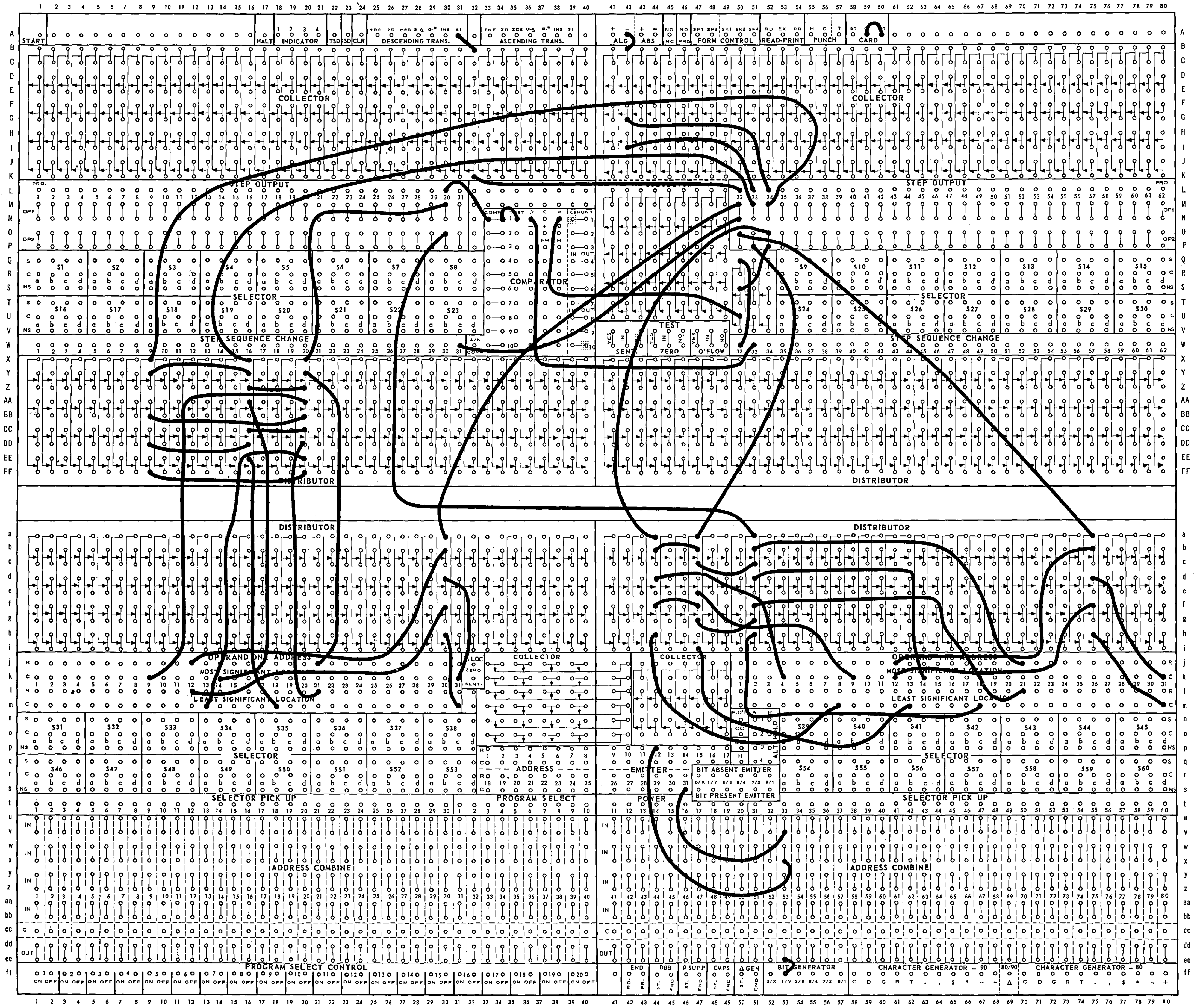


Figure 2-11. Representative Wiring – Example 2.

that the same collector that was used on Step 11 is available. No "back circuit" can develop due to the protection of the diodes that are indicated in the connection panel marking. Current can flow only in the direction of the arrow.

Wire 14 is used to define Operand 1 of Step 12. This operand, R9/C7 – R9/C12, has a common row number with Operand 1 of Step 11. Since the two R9 hubs in the Operand One Address area of the connection panel are single hubs, it is necessary to use the dual hubs of the distributor to avoid use of Y-wires. This simplified system of wiring is illustrated by the wires numbered 19 and 20.

Wire 10 leads to the defining of Operand 2 of Step 12. Since this operand is identical to the storage locations used as Operand 2 of Step 11, complete use of dual hub facility is possible. The wires numbered 11 illustrate this possibility.

Wire 12 routes the Step 12 signal to the IN of the Zero Test switch that was set on Step 11. This signal emits from either the YES or NO hub depending on the condition found on Step 11. The wiring from this switch can be to Program Select Controls, Step Sequence Change hubs, or to any destinations needed by the programmer.

The same storage locations are handled as Operand 2 of Step 12, as were used during Step 11. To stop the machine from again testing storage location R5/C9 for a zero, it is necessary on Step 12 that a Step Output pulse (an A-pulse) be routed to the combine that receives the B-pulses from the Address Emitters when R5/C9 is handled. This insures that no signal from the OUT hubs of the combine will be generated on Step 12. Wire 13 carries this inhibiting signal to the combine.

Wire 15 carries a Step 12 signal to the IN of the Arithmetic Overflow switch. If the addition performed on Step 12 results in a sum having more digits than there are storage locations in Operand 2, this switch is set to connect IN with YES. The signal entering IN then goes to HALT and INDICATOR 1. The operator can then take corrective action.

Example 2

Figure 2-11 is a portion of a program that is used only for illustrative purpose and is not a unique approach nor is it complete.

The problem is stated:

FICA YEAR-TO-DATE, including this week, has been computed and is stored in R21/C9 – R21/C13.

FICA this week has been computed and is stored in R21/C4 – R21/C8.

Maximum FICA, a constant, is stored in locations R21/C14 – R21/C18.

When an employee has paid in maximum FICA, punch a Y in Column 80. Other punches may also be made in Column 80.

The program is:

Step 30 Compare FICA Year-to-date: FICA Maximum for the possible results

OP1 = OP2 Go to Step 32

OP1 < OP2 Go to Step 31 (automatic)

OP1 > OP2 Go to Step 33 to determine amount FICA

Step 32 Superimpose a Y bit in R12/C31. Go to Step 31 (must be signaled).

Step 33 Subtract FICA maximum from FICA Y.T.D. to determine the amount by which FICA this week must be reduced. Go to Step 34.

Step 34 Subtract the amount over FICA maximum from FICA this week. Go to Step 32 since employee has reached maximum.

In Summary

	OPERAND 1	OPERAND 2	PROCESS
Step 30	R21/C9 – R21/C13	R21/C14 – R21/C18	Compare
Step 31	Not detailed here; begin edit routine		
Step 32	R12/C31 – R12/C31	R12/C31 – R12/C31	Transfer-Superimpose Superimpose a Y-bit
Step 33	R21/C14 – R21/C18	R21/C9 – R21/C13	Subtract Algebraic
Step 34	R21/C9 – R21/C13	R21/C4 – R21/C8	Subtract Algebraic

Note that Step 31 begins automatically if the compare on Step 30 results in Operand 1 is less than Operand 2. This step must be forced, however, by wiring to Step Sequence Change 31 from Step 32.

Step 32 may be entered from either of two points in the program: (1) if the comparison in Step 30 produces an equal result, or (2) after Step 34 has been performed. Both of these conditions require Step Sequence Change 32 to receive a signal.

Since one destination must receive signals from two sources, a Collector must be used. The two source signals are wired to separate, diode-protected IN pairs, and the OUT of the Collector is wired to Step Sequence Change 32.

IV. PROGRAMMING AND WIRING TECHNIQUES

ARITHMETIC PROCESSES

Adder

The 1004 processor performs addition and subtraction through the use of a one-character serial adder. Information is moved one character at a time from the operands to the adder, beginning with the least significant locations. One digit from Operand 1 is added to one digit from Operand 2, and the result digit is stored before the next pair of digits is added. Any carry generated by adding a pair of digits is delayed so that it enters the adder with the next pair of digits handled. The result of the addition of each digit-pair is stored in the location from which the Operand 2 digit was obtained. Subtraction is performed by adding the complement of Operand 1 to Operand 2.

Add Algebraic

When the ADD ALG process is performed, an X-bit in the LSL of each operand is recognized as a negative sign indication. The presence of an X-bit in the LSL of either operand conditions the machine to treat this operand as a negative value. The sum carries the true algebraic sign.

When a negative value is added to a smaller positive number, the original result will be the complement of the true answer. The machine automatically recomplements this answer to give a true value result. Automatic recomplementing of a negative answer is prevented if, on the same step as the add order is given, NO RC (no recomplement) is signaled.

The adder operates at microsecond speed. Since no special intermediate accumulator is needed, values of any size can be accumulated.

Figure 2-12 graphically illustrates how the single-digit adder obtains the sum of two 3-digit fields. The handling of sign is not shown.

The UNIVAC 1004 processor permits many very powerful processes to be profitably used by an installation. The ability to address operands of any length and the use of a single-digit adder combine to supply one of these unique strengths.

If a number of numeric fields of the same sign must be grouped for printing and/or accumulating, the user can accumulate all fields into their respective sums in ONE program step.

ADDITION

EXAMPLE: (OP1) + (OP2) → OP2

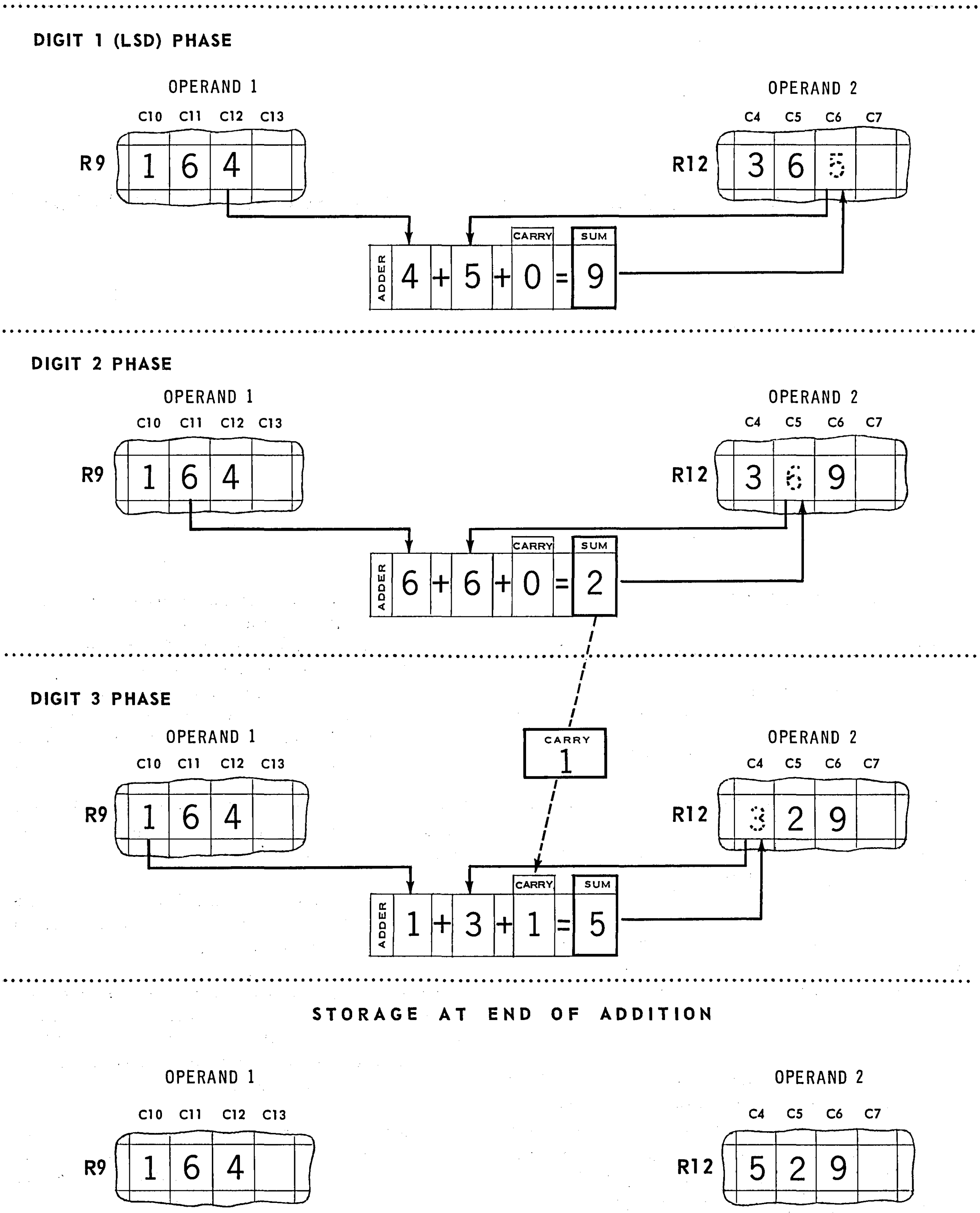


Figure 2-12. Adding Two 3-Digit Quantities.

For example, a card form is used that contains the following values;

	CARD COLUMNS	READ STORAGE AREA
Factor 1	19–23	R1/C19 - R1/C23
Factor 2	24–29	R1/C24 - R1/C29
Factor 3	30–37	R1/C30 - R2/C6
Factor 4	38–45	R2/C7 - R2/C14
Sign of all 4 fields		X-bit of R2/C14

In working storage these four factors from each card read are being accumulated in the storage locations:

Σ Factor 1's	R21/C1 - R21/C5
Σ Factor 2's	R21/C6 - R21/C11
Σ Factor 3's	R21/C12 - R21/C19
Σ Factor 4's	R21/C20 - R21/C27
Sign of all four factors	X-bit of R21/C27

One program step makes this accumulation:

OPERAND 1	PROCESS	OPERAND 2
R1/C19 - R2/C14	ADD ALG	R21/C1 - R21/C27

In the example given it was assumed that the number of digits in the total for each factor does not exceed the number of columns in each factor's card field. If this condition is not true, a single step before addition can edit the factors in storage, inserting spaces to the left of each factor to allow the total field to be larger than the value field. Of course, the result storages for each factors must be established with enough locations to contain the expanded total.

Subtract Algebraic

THE MACHINE SUBTRACTS OPERAND 1 FROM OPERAND 2

A true value result, with its proper sign, is delivered to Operand 2 storage locations.

Since the machine operates in a way counter to pencil and paper procedure, where the second value written down is subtracted from the first value, this rule must be emphasized.

Usual school room notation is: 4 − 3 = 1
Machine Notation is: −3 + 4 = 1

This is only a matter of notation, and reasonable care will make this method habitual for the user.

Add Absolute

Addition can be done with this order exactly as with Add Algebraic except that the least significant location is NOT used to determine the sign of the operands. The machine assumes both operands to be positive, and the sum does not carry a sign indication.

This instruction is vital to the efficient programmed multiplication routine used with this machine. This order will have frequent use in many tabulating runs such as those to obtain certain control totals when sign implications are of no concern.

An example of adding multiple factors in one step was shown in the Add Algebraic section of this chapter. Since the process was Add Algebraic, which considers the X-bit of the LSL of each operand as the stored sign location, a restriction was made that the X-bits of R2/C14 and R21/C27 must contain the sign that applies to all four factors. If the Add Absolute process had been used instead, the machine would simply have ignored those X-bits and regarded both operands as positive values.

Subtract Absolute

Subtraction can be performed with this order exactly as with Subtract Algebraic except that the least significant location is NOT used to determine the sign of the operands. The machine assumes both operands to be positive, and the result does not carry a sign indication.

Wiring a compare process on the same step provides the result sign indication. The sign is determined on the basis of the relative magnitude of the operands. Therefore:

If OP1 > OP2; result is minus.
If OP1 < OP2; result is plus
If OP1 = OP2; result is zero

CLEARING ON AN ARITHMETIC STEP

The clear feature may be used with any of the arithmetic processes. When Clear is wired from a Step Output hub in connection with an arithmetic process, Operand 1 storage locations are cleared to spaces at the completion of that step.

TRANSFER

Data is transferred from one storage location to another by means of logical commands called TRANSFER. A variety of changes can be made in the information during transfer, depending upon which optional transfer hubs are signaled.

Transfers are effected serially, from the source location of the information, Operand 1, to the destination location, Operand 2. Movement can be from the least significant location of Operand 1 to the least significant location of Operand 2; then from the next location to the left of the LSL of Operand 1 to the location to the left of the LSL of Operand 2, and so forth, until the transfer is complete. This method of transfer is referred to as ASCENDING TRANSFER, and is initiated when any of the Ascending Transfer hubs of the connection panel is signaled.

For example, if an ascending transfer is ordered by signalling the TRF hub on a step that defines Operand 1 as R2/C2 – R2/C5, four locations, and Operand 2 as R10/C2 – R10/C6, five locations, the transfer proceeds as follows:

1. Before the transfer step starts, storage appears as:

	C1	C2	C3	C4	C5	C6
R2		J	O	H	N	
		OP1				

	C1	C2	C3	C4	C5	C6	C7
R10		Δ	Δ	Δ	Δ	Δ	
		OP2					

2. After one character is transferred, storage appears as:

	C1	C2	C3	C4	C5	C6
R2		J	O	H	N	
		OP1				

	C1	C2	C3	C4	C5	C6	C7
R10		Δ	Δ	Δ	Δ	N	
		OP2					

3. After the second character is transferred, storage appears as:

	C1	C2	C3	C4	C5	C6
R2		J	O	H	N	
		OP2				

	C1	C2	C3	C4	C5	C6	C7
R10		Δ	Δ	Δ	H	N	
		OP2					

4. After the ascending transfer is complete storage appears as:

	C1	C2	C3	C4	C5	C6
R2		J	O	H	N	
		OP1				

	C1	C2	C3	C4	C5	C6	C7
R10		Δ	J	O	H	N	
		OP2					

At the end of the ascending transfer, any unused storage locations in Operand 2, to the LEFT of the most significant location, are space filled. If Operand 2 contains data before the transfer, that data is destroyed and replaced by the transferred data, and any unused locations in Operand 2 will contain spaces.

A transfer starting with the most significant location is referred to as a DESCENDING TRANSFER. If the example given previously is executed as a descending transfer, it results,when completed, in storage as:

	C1	C2	C3	C4	C5	C6
R2		J	O	H	N	
		OP1				

	C1	C2	C3	C4	C5	C6	C7
R10		J	O	H	N	Δ	
		OP2					

In a descending transfer, unused locations in Operand 2, to the RIGHT of the data transferred, are filled with spaces.

As previously mentioned, there are seven different transfer options which may be executed singly or in combination on one program step. *When several transfer operations are performed by a single step, they must be specified as either all ascending or all descending transfers.*

The seven transfer options are:

1. TRF (Transfer)

   This option is a standard transfer. Operand 1 is duplicated character for character in the Operand 2 locations. The transfer can be made in either ascending order or descending order. The data in Operand 1 locations is not altered.

2. Ø – Δ (Zero Suppress with Space Fill)

   With this command, preceding or trailing zeros in a storage location are eliminated and replaced with spaces. This option is frequently needed in preparing data for printing. Zero Suppress with Space Fill is usually wired in descending order to eliminate preceding zeros, but can be used in ascending order to eliminate trailing zeros.

   The Zero Suppress Start and End (Ø Supp-ST and Ø Supp-END) hubs must be used in conjunction with the Ø-Δ option. The Ø Supp-ST hub must be signaled from an Address Combine that is wired from Address Emitter hubs defining the storage location within the Operand 2 area where zero suppression is to begin. The Ø Supp-END hub must be signaled from an Address Combine that is wired from Address Emitter hubs that define the storage location of the last character to be considered in the zero suppress area.

   The Ø Supp-ST and END addresses do not replace the OP1 and OP2 addresses. The operand addresses must be wired.

   Within the Operand 2 storage area any number of smaller areas may be successively zero-suppressed on one step by proper signaling of the Ø Supp-ST and Ø Supp-END hubs while the step is in progress.

   Zero suppression begins when the serial transfer of data to Operand 2 reaches the storage location that is wired through an Address Combine to Ø Supp-ST. The machine examines the character being transferred to the Operand 2 location. If a zero is found, a space replaces it in the destination location. The suppress process continues until one of the following four conditions is met:

   1. A character other than space, comma, or zero is TRANSFERRED FROM OP1.
   2. A period (decimal point) is INSERTED.
   3. The LSL of Operand 2 is reached in descending zero-suppression; the MSL of Operand 2 is reached in ascending zero-suppression.
   4. Ø Suppress-END is signaled.

   When Zero Suppress is wired in combination with Insert or Superimpose, the following facts must be observed. Once zero suppression has begun and while it is *actively* in progress, any character or bit pattern except a comma can be inserted; if a comma is wired to be inserted, a space is delivered to that Operand 2 location. If a period is inserted, zero suppression stops at that location. The insertion of any other character does *not* stop zero suppression.

   The fact that a comma cannot be inserted during active zero suppression does not prevent the insertion of commas when zero suppress is inactive (for example, after a significant digit has been detected in a zero suppress field).

   An ascending Ø – Δ transfer operates the same as a descending transfer except that the storage locations of Operand 2 are accessed serially from the least significant location to the most significant location. In this instance, zeros to the right of the least significant character in the zero suppress area are replaced by spaces.

   Reaching the end of a zero suppress field before the end of Operand 2 does not stop the transfer of data from Operand 1.

3. Ø – * (Zero Suppress with Asterisk Fill)

   This transfer operates the same as a Ø – Δ and in conjunction with Ø Supp-ST-END. However, the code for an asterisk is used as the replacement character instead of a space (Δ).

4. ZD (Zone-Delete)

When a transfer is made using the ZD option the 8, 4, 2, and 1 bits of the characters in Operand 1 locations are duplicated in Operand 2 locations. No zone bits are duplicated. An X-bit in the sign location, indicating a negative sign, is also eliminated. The data in Operand 1 locations is not altered.

5a. ZDS (Zone-Delete with Sign)

The ZDS option is used when it is desired to delete the zone bits from an operand but retain the X-bit in the LSL as a sign indication. A ZDS TRANSFER CAN ONLY BE PERFORMED IN ASCENDING ORDER. The character in the LSL of Operand 1 including both zone bits is duplicated in the LSL of Operand 2. The remaining characters of Operand 1 are duplicated serially in Operand 2 locations without zone bits. The data in Operand 1 locations is not altered.

5b. DØB (Delete Zero Balance)

The Delete Zero Balance transfer option may be performed in the Descending mode only. It is used to clear zero-balance areas in storage to spaces. Like all other transfer options, its function is to move data from Operand 1 locations to Operand 2 locations, but in this case with a selective clearing to spaces.

The process operates in conjunction with the wiring to DØB – ST and END. When a character is about to be transferred to an OP 2 location which is wired to DØB-ST, that character is examined for the presence of a Y-bit, indicating a zero balance. (Any arithmetic operation which results in a zero balance causes a Y-bit to be inserted in the MSL of Operand 2 automatically.) If a Y-bit is present, a space is delivered to the OP 2 location and to all subsequent locations until a location wired to DØB-END is reached. If the Y-bit is not present in the location wired to DØB-ST, normal transfer from OP 1 to OP 2 is performed.

Any number of locations may be wired to DØB-ST and END within the limits of the locations specified as OP 2. All such wiring is from the appropriate Address Emitter hubs through Address Combines to DØB-ST and END.

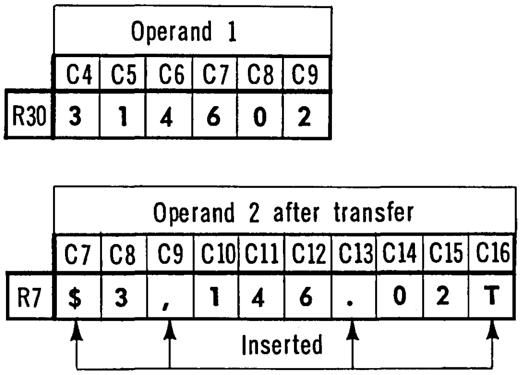
6. IN (Insert)

The Insert transfer option is used to place any character or bit configuration into any storage location within Operand 2 of the step. When the machine is ordered to do an insert, it stops moving characters from Operand 1 to Operand 2, makes the insertion in the Operand 2 storage location, and then restarts moving characters from Operand 1 to Operand 2.

When the Insert option is used, Operand 2 must be wired to contain at least as many storage locations as Operand 1 plus the number of characters to be inserted.

The addresses of the storage locations in which insertions are to be made must be specified by proper wiring of Address Emitters, and must be within the Operand 2 storage locations.

For example, a six-digit value has been accumulated in storage. Before it is printed, it is necessary to insert a "T" to the right of the value, a decimal point to separate dollars from cents, a comma to separate the most significant digit from the next digit to the right, and a dollar sign to the left of the most significant digit (Figure 2-13).



	Operand 1					
	C4	C5	C6	C7	C8	C9
R30	3	1	4	6	0	2

	Operand 2 after transfer									
	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16
R7	$	3	,	1	4	6	.	0	2	T

Figure 2-13. Result of Insert Transfer

7. SI (Superimpose)

The Superimpose transfer option is used when it is necessary to add bits to specific storage locations in Operand 2 without eliminating the bits already stored. Whereas the Insert option spread apart two characters in storage and adds a character between them, the Superimpose option does not change the relative locations of characters. Bits are overlaid on the characters from Operand 1 as they are transferred to Operand 2 locations.

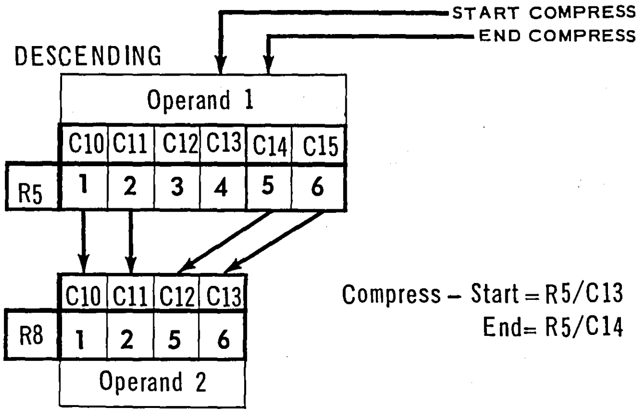
Thus, if a storage location contains the code for 1 (000100) and a Y-bit is superimposed, that storage location will then contain a 010100 that will be recognized throughout the machine as an A.

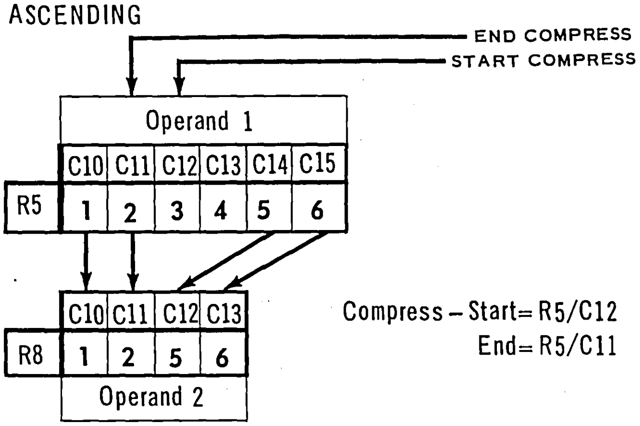
A common use of this feature is to superimpose zone-bits in punch storage locations where it is desired to create X and Y punches for control.

IF A SPACE IS SUPERIMPOSED, A UNIQUE RESULT IS THE ELIMINATION OF ALL BITS IN THE STORAGE LOCATION SPECIFIED. Superimposing spaces can be used to clear selected locations within Operand 2.

COMPRESS FEATURE

The Compress feature is used in conjunction with either the Insert transfer option or the Superimpose transfer option. Use of the compress feature results in the deletion of specified locations within Operand 1 upon transfer to Operand 2. It is possible to delete OP 1 locations through Compress without performing an insert or a superimpose operation; however, the wiring must be made through one of these transfer options. Data can be inserted or superimposed (one or the other, not both) along with Compress on the same transfer step. If, during an insert or superimpose transfer, it is desired that the transfer of data from specific locations within Operand 1 is to be prevented, the Compress feature is made effective for these locations. The address of the OP 1 location immediately following the first location to be deleted is wired to the Compress START hub. The OP 1 location immediately following the last location to be deleted is wired to Compress END. For example, if six locations were specified for OP 1, and the third and fourth locations were to be deleted during a descending insert transfer, START Compress would be signaled from the fourth location, and END Compress from the fifth location.





Any number of locations, from a minimum of 1 to a maximum of 1 less than the number of locations specified for OP 1, may be deleted during a single active compress operation. Compress can be started and ended any number of times during a single transfer according to program requirements. If a single location is to be deleted, START and END would be signaled from the same OP 1 address. The last location to be transferred from OP 1 cannot be deleted during a compress operation; neither the LSL of OP 1 during a descending transfer, nor the MSL during an ascending transfer.

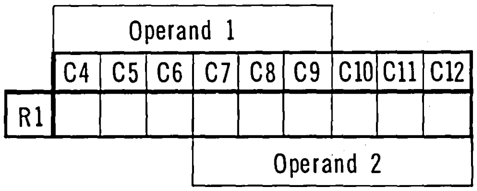
The Compress feature is the exception to two previously stated programming rules. The exceptions to these rules are:

1. Operand 2 may contain fewer location than Operand 1. However, Operand 2 must contain as many locations as Operand 1 plus inserts, minus the locations to be deleted.
2. Operand 1 addresses are used to start and end Compress. OP 1 addresses are wired from the Address Emitter in the same manner as OP 2 addresses which are used for all other features.

Transfer With Overlapping Operands

Operand 1 and Operand 2 may be defined as different areas in storage when a transfer process is executed. They may also be defined as the same storage areas. In addition, the two operands may be defined so that the storage areas partially overlap.

An example of partially overlapping operands might be:



Or, the overlap might be:

			Operand 1					
	C9	C10	C11	C12	C13	C14	C15	C16
R12								
	Operand 2							

Usually, when operand-overlap is specified, one of the transfer options used is Clear (CLR) althrough this is not a requirement. The examples given later show the result of NOT using the Clear option.

There are, however, two requirements that must be met to obtain a result that is meaningful when transfer is done with partially overlapping operands.

1. *When the MSL of Operand 1 is to the left of the MSL of Operand 2, the transfer executed must be in the ASCENDING Mode.*

2. *When the MSL of Operand 1 is right of the MSL of Operand 2, the transfers executed must be in the DESCENDING Mode.*

For example, given the operands illustrated below:

	Operand 1								
	C4	C5	C6	C7	C8	C9	C10	C11	C12
R10	Δ	9	7	1	3	6	Δ	Δ	Δ
				Operand 2					

In accordance with Rule 1, an ascending transfer is performed. The TRF option is ordered but the CLR option is not ordered. After execution, the result appears in storage as:

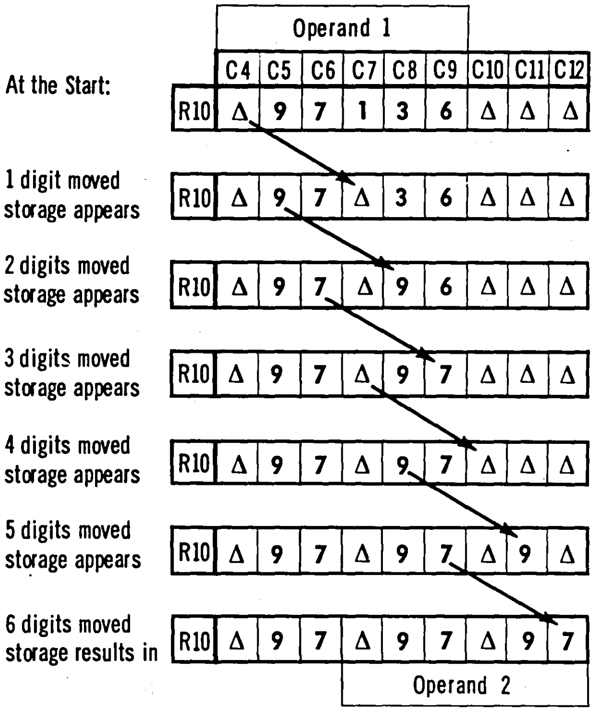
	Operand 1								
	C4	C5	C6	C7	C8	C9	C10	C11	C12
R10	Δ	9	7	Δ	9	7	1	3	6
				Operand 2					

If, however, an ascending transfer with the Clear option is ordered, the result appears in storage as:

	Operand 1								
	C4	C5	C6	C7	C8	C9	C10	C11	C12
R10	Δ	Δ	Δ	Δ	9	7	1	3	6
				Operand 2					

If the rules stated above are not followed and a descending transfer is ordered, the error that results is illustrated in Figure 2-14. Each line shows storage as it appears after each character is moved, beginning with the MSL since this is a descending transfer. Note that the following example demonstrates an ERROR in programming and is included for illustration purposes only.

It is obvious that this result is not one that usual business problems could use. The first rule in transferring overlapping operands was not followed since the MSL of Operand 1, R10/C4, is to the left of the MSL of Operand 2, R10/C7. When this condition exists, the transfer MUST BE ASCENDING.



		Operand 1								
		C4	C5	C6	C7	C8	C9	C10	C11	C12
At the Start:	R10	Δ	9	7	1	3	6	Δ	Δ	Δ
1 digit moved storage appears	R10	Δ	9	7	Δ	3	6	Δ	Δ	Δ
2 digits moved storage appears	R10	Δ	9	7	Δ	9	6	Δ	Δ	Δ
3 digits moved storage appears	R10	Δ	9	7	Δ	9	7	Δ	Δ	Δ
4 digits moved storage appears	R10	Δ	9	7	Δ	9	7	Δ	Δ	Δ
5 digits moved storage appears	R10	Δ	9	7	Δ	9	7	Δ	9	Δ
6 digits moved storage results in	R10	Δ	9	7	Δ	9	7	Δ	9	7
					Operand 2					

Figure 2-14. Incorrect Transfer Operation

When the MSL of Operand 1 is right of the MSL of Operand 2, storage appears as:

			Operand 1					
	C9	C10	C11	C12	C13	C14	C15	C16
R12	Δ	Δ	4	1	3	5	2	7
	Operand 2							

As stated in Rule 2, a descending transfer, TRF option, is ordered but the CLR option is not ordered. After execution, the result appears in storage as:

			Operand 1					
	C9	C10	C11	C12	C13	C14	C15	C16
R12	4	1	3	5	2	7	2	7
	Operand 2							

A descending transfer, CLR option, results in storage as:

			Operand 1					
	C9	C10	C11	C12	C13	C14	C15	C16
R12	4	1	3	5	2	7	Δ	Δ
	Operand 2							

The rules of transfer with partially overlapping operands must be observed. If an ascending transfer is ordered with operands overlapped so the MSL of Operand 1 is to the right of the MSL of Operand 2, a violation of rule 2 is made. Result storage, assuming Clear option is not used, will be:

			Operand 1					
	C9	C10	C11	C12	C13	C14	C15	C16
R12	2	7	2	7	2	7	2	7
	Operand 2							

CLEARING ON TRANSFER

When performing all other transfer options, the data in Operand 1 locations is not altered. Because of specific programming considerations, it may be desirable to delete the information contained in Operand 1 locations during the transfer. When the Clear transfer option is used, Operand 1 locations are cleared to spaces.

Operand 1 and Operand 2 may be partially or completely overlapped during a transfer operation. If a Clear transfer is ordered with overlapping operands, only those locations of Operand 1 which are not overlapped by Operand 2 are cleared to spaces.

For example, data in R4/C3 – R4/C10 is to be transferred to R4/C7 – R4/C14. In this instance, the operands partially overlap, and the data in the Operand 1 locations that are not overlapped has no value after transfer is completed. For this operation, the CLR-Ascending Transfer hub is signaled.

Before this transfer is started, storage appears as:

	OP1												
	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
R4	J	O	H	N	Δ	L	E	E	Δ	Δ	Δ	Δ	Δ
					OP2								

After the transfer, storage appears as:

	OP1												
	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
R4	Δ	Δ	Δ	Δ	J	O	H	N	Δ	L	E	E	
					OP2								

NOTE: Locations C3 – C6 contain spaces.

MULTIPLICATION

Multiplication is performed by a subroutine, and the programmer can minimize time required for this operation by limiting the result to only meaningful, useful digits.

Multiplication is done by first completing two data positioning steps – a procedure that has a counterpart in most computing systems. Once the two values have been positioned for the process, the two steps that actually perform multiplication begin.

The explanation given here will be more meaningful for most readers if continuous reference is made to the step-by-step example given in Figure 2-15. Three areas of storage are defined for use in multiplication. For convenience in this explanation, these areas are referred to as A, B, and C. In general terms these areas store the multiplier (A), the product (B), and the multiplicand (C). A and B must be made up of adjacent, consecutive storage locations. Area C may be located anywhere in working storage (Figure 2-15, line 1).

The three areas must contain a definite number of storage locations; the number is related to the number of digits in the values to be handled. Determining the number of storage locations needed for each area is done with these rules:

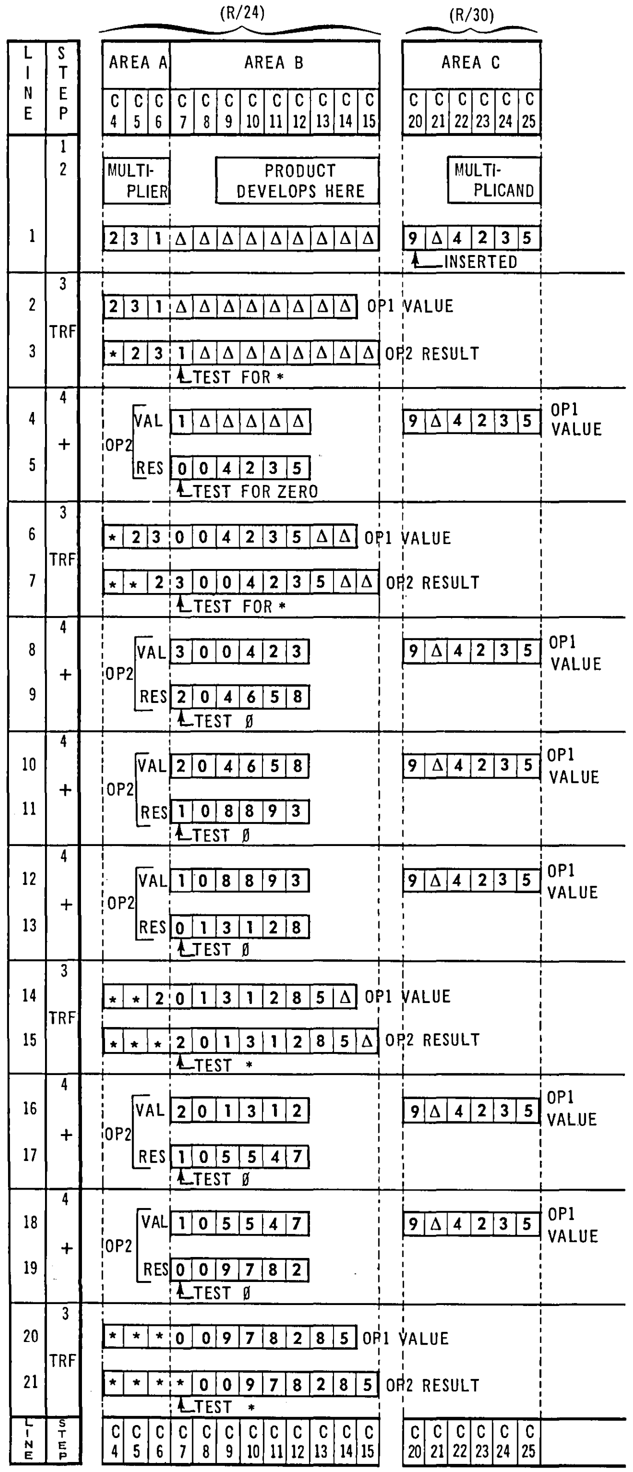

(R/24)
(R/30)
LINE
STEP
AREA A
AREA B
AREA C
C C C C C C C C C C C C
4 5 6 7 8 9 10 11 12 13 14 15
C C C C C C
20 21 22 23 24 25
1 2
MULTIPLIER
PRODUCT DEVELOPS HERE
MULTIPLICAND
1 2 3 1 Δ Δ Δ Δ Δ Δ Δ Δ Δ 9 Δ 4 2 3 5
INSERTED
2 3 TRF 2 3 1 Δ Δ Δ Δ Δ Δ Δ Δ Δ OP1 VALUE
3 * 2 3 1 Δ Δ Δ Δ Δ Δ Δ Δ Δ OP2 RESULT
TEST FOR *
4 4 + VAL 1 Δ Δ Δ Δ Δ 9 Δ 4 2 3 5 OP1 VALUE
OP2
5 RES 0 0 4 2 3 5
TEST FOR ZERO
6 3 TRF * 2 3 0 0 4 2 3 5 Δ Δ OP1 VALUE
7 * * 2 3 0 0 4 2 3 5 Δ Δ OP2 RESULT
TEST FOR *
8 4 + VAL 3 0 0 4 2 3 9 Δ 4 2 3 5 OP1 VALUE
OP2
9 RES 2 0 4 6 5 8
TEST Ø
10 4 + VAL 2 0 4 6 5 8 9 Δ 4 2 3 5 OP1 VALUE
OP2
11 RES 1 0 8 8 9 3
TEST Ø
12 4 + VAL 1 0 8 8 9 3 9 Δ 4 2 3 5 OP1 VALUE
OP2
13 RES 0 1 3 1 2 8
TEST Ø
14 3 TRF * * 2 0 1 3 1 2 8 5 Δ OP1 VALUE
15 * * * 2 0 1 3 1 2 8 5 Δ OP2 RESULT
TEST *
16 4 + VAL 2 0 1 3 1 2 9 Δ 4 2 3 5 OP1 VALUE
OP2
17 RES 1 0 5 5 4 7
TEST Ø
18 4 + VAL 1 0 5 5 4 7 9 Δ 4 2 3 5 OP1 VALUE
OP2
19 RES 0 0 9 7 8 2
TEST Ø
20 3 TRF * * * 0 0 9 7 8 2 8 5 OP1 VALUE
21 * * * * 0 0 9 7 8 2 8 5 OP2 RESULT
TEST *
LINE
STEP
C C C C C C C C C C C C
4 5 6 7 8 9 10 11 12 13 14 15
C C C C C C
20 21 22 23 24 25

Figure 2-15. Multiplication Example

*NOTE: This example uses the symbol * to represent the sentinel for which machine code is 110000.*

AREA A

Storage locations equal to the maximum number of digits that can be in the multiplier. Multiplication is performed by using Add Absolute; therefore, the sign indication is ignored.

AREA B

Both of the following rules must be observed in determining the size of area B.

1. The number of locations in area B must equal the number of digits wanted in the product, plus 2. (If the number of product digits desired is less than the maximum that can be developed, the less significant digits will be dropped off.)

2. The number of locations in area B must be equal to or greater than the number in area C.

Before the multiplication is begun, area B must be known to contain nothing but zeros or spaces. The presence of any other codes in this area will result in an incorrect product.

AREA C

Two more storage locations than there are digits in the multiplicand.

Figure 2-15 shows an example of multiplication using a three digit multiplier (231), a four digit multiplicand (4235), and provision for a full seven-digit product.

Program Sequence

Two data-positioning steps are used to load values into the storage areas selected by the programmer as areas A and C. These steps are:

STEP 1

Transfer the multiplier to area A (Figure 2-15, line 1).

STEP 2

Transfer (Ascending) the multiplicand to area C. Use the Insert option of the transfer order. During the transfer, insert a 9 in the most significant location of area C. Since the transfer is ascending, the least significant digit of the multiplicand will be in the LSL of area C (Figure 2-15, line 1).

When the two positioning steps just described are completed, the machine must be directed to begin the two steps of multiplication. These steps are:

STEP 3

Transfer (Ascending), using the Insert option of the transfer order. In this step Operand 1 includes all of the storage locations of area A and all but the LSL of area B. See Figure 2-15, line 2. Operand 2 includes ALL storage locations of area A and area B. See Figure 2-15, line 3. During this transfer a number of actions take place almost simultaneously. Of necessity these are described here individually.

a) Combined areas A and B are shifted one place to the right and the LSD of the original area B is eliminated.

b) A sentinel (110000) is inserted in the Operand 2 storage location that is the MSL.

c) A test is made on the Operand 2 storage location that is the MSL of area B. This test is to discover if the storage location is storing either a zero or a sentinel. The results of this test will determine the next action.

1. If the storage location is storing a sentinel, multiplication is complete. Go to the next program step to be done in the over-all procedure.

2. If the storage location is storing a zero, repeat Step 3.

3. If neither a zero or a sentinel is detected, go to Step 4.

The result of the first Step 3 transfer is shown in Figure 2-15, line 3.

NOTE: Ø = Space in Ø Test.

STEP 4

Add Absolute. In this step, Operand 1 is area C. Operand 2 is part or all of area B. Use as many storage locations of area B as there are storage locations in area C. The storage locations of area B that are used are the most significant locations. Therefore, the six most significant locations of area B are used for Operand 2. See Figure 2-15, lines 4, 5, and 6.

The most significant location of Operand 2 is, of course, a digit that was originally a part of the multiplier, area A. Shifting in Step 3 moved this digit to area B. As the multiplication process continues, Step 3 is repeatedly used, and on each use another digit from the multiplier is positioned in this location. When Add Absolute is done, the 9 that was stored in the MSL of area C is added to the MSL of area B, and the effect is to reduce that number by 1. (Because of the nature and size of the operands, Arithmetic Overflow will occur each time this step is performed. The routine is designed to utilize this fact, so the programmer should ignore Arithmetic Overflow when multiplying.)

During the Add Absolute step, the MSL of the result, Operand 2, is tested to determine if it is storing a zero. As a result of this test, the program will proceed one of two ways:

1. If zero is stored, go to Step 3.
2. If zero is not stored, repeat Step 4.

For example, at line 5 of the Figure, the test indicated a zero in C7. On line 11, the test indicated no zero; therefore, Step 4 was repeated.

The multiplication process continues through repetitive addition and shifting until a sentinel (110000) is detected in C7 as the result of a transfer step. See Figure 2-15, line 21. When the sentinel is detected, multiplication is complete and the program is directed to proceed to the next operation.

A complete example of the steps and data movements and accumulations that occur in the sample used is given as Figure 2-15.

DIVISION

As an aid to understanding division, the following definitions are repeated:

1. An OPERAND consists of adjacent storage locations defined by the programmer through connection panel wiring. The number of storage locations included in an operand may vary from 1 to 961.
2. The storage location defined as the MOST SIGNIFICANT LOCATION (MSL) is always at the extreme left end of a group of locations making up an operand. The storage location immediately to the right of the MSL is referred to as MSL-1 (Most Significant Location minus one). The next location to the right is MSL-2. This "naming" of locations in the operand from the left continues through MSL-3, MSL-4, MSL-5, and so forth, to the rightmost location of the operand.
3. The storage location defined as the LEAST SIGNIFICANT LOCATION (LSL) is always at the extreme right end of a group of locations making up an operand. This location can be referred to either as the LSL or by its position relative to the MSL. For example, in a six-location operand:

MSL	MSL -1	MSL -2	MSL -3	MSL -4	LSL or MSL -5

Division is done with a sub-routine within a program. A number of approaches may be taken to accomplish this operation; one of these is discussed in the following paragraphs. This approach is not presented as the most effective for every programming situation – it is a typical method.

The method of division described here has only one restriction – the divisor cannot exceed ten digits. The definitions of value areas given in the example do not permit the divisor to contain decimal fractions (digits to the right of the decimal point) but rules for using such divisor values can be formulated from this example.

The routine requires that a single constant, a one (1), be available somewhere in storage. A second requirement is to clear the quotient area in storage before starting the subroutine.

Continued reference to the detailed example, Figure 2-16, printed on the foldout at the end of this section, is helpful in understanding the explanation.

Certain areas in storage are established by the programmer to contain the factors involved in division. Figure 2-16, line 1, shows these areas. The Row and Column designations are arbitrarily chosen; there are no limitations on the storage locations that may be used in any particular program. The areas are:

AREA	CONTENTS
A	Quotient
B	Dividend
C	Divisor
D	Shift Count

Areas A and B must be located in consecutive, adjacent storage locations.

The rules for calculating the number of storage locations that must be established for each area are as follows:

AREA A (Quotient) – May be located anywhere in storage.

The number of storage locations equals:
The number of digits to the left of the decimal point in the dividend;
PLUS – number of digits wanted to the right of the decimal point in the quotient;
PLUS – number of digits in the divisor, minus one (1).

AREA B (Dividend) – MSL of this area must be immediately to the right of the LSL of Area A.

The number of storage locations equals:

The number of digits in the dividend (regardless of digit position in relation to the decimal point);
PLUS – four (4);
AND – the total obtained by the above computation must be equal to or greater than the number of digits in the divisor PLUS four (4). If it is not, add a number to make the sum this large.

AREA C (Divisor) – May be located anywhere in storage.

The number of storage locations equals:

The number of digits in the divisor (regardless of digit position in relation to the decimal point);
PLUS – three (3);

AREA D (Shift Count) – May be located anywhere in storage.

One (1) storage location is needed for Area D.

Editing The Values

Four steps are required to edit the values and locate them properly for division to begin:

STEP 1

Process: Insert transfer, ascending.

Operand 1: Storage locations containing the value that is the dividend.

Operand 2: Area B
Insert a sentinel (110000) in the MSL of Operand 2. Go to Step 2.

STEP 2

Process: Insert transfer, ascending.

Operand 1: Storage locations containing the value that is the divisor.

Operand 2: Area C.
Insert a "9" in the MSL of Operand 2.
Test MSL-3 of Area C for a Ø:
If NO, go to Step 5 – start division,
If YES, go to Step 3 – adjust shift count.

STEP 3

Process: Add Absolute.

Operand 1: Storage location containing a 1 constant.

Operand 2: Area D.
Test for arithmetic overflow:
If NO, go to Step 4,
If YES, ten divisor digits have been tested and all found to be zero. Division cannot be done; this situation usually indicates an error. Go to a subroutine that correctly handles this error condition.

STEP 4

Process: Transfer, descending.

Operand 1: Area C storage locations MSL-4 through LSL.

Operand 2: Area C storage locations MSL-3 through LSL.
Test the MSL of Operand 2 (that is MSL-3 of Area C) for a zero:
If NO, go to Step 5; start division.
If YES, go to Step 3.

Dividing

Three additional steps are required for the actual division operation.

STEP 5 (TRIAL SUBTRACTION)

Process: Subtract Absolute; No Recomplement.

Operand 1: Area C.

Operand 2: Part of Area B to include MSL-1 of Area B as the MSL of this operand, and enough locations to equal the number of locations in Area C.
Test the storage location that is MSL-2 of Area B, for a zero: this is MSL-1 of Operand 2 of this step.
If NO, go to Step 6,
If YES, repeat Step 5.

STEP 6 (ADD-BACK)

Process: Add Absolute

Operand 1: Area C, except the MSL of Area C.

Operand 2: The same as Operand 2 of Step 5, except the MSL of that operand.
Go to Step 7.

STEP 7

Process: Transfer, descending.

Operand 1: Area A and Area B storage locations, except the MSL of Area A.

Operand 2: Area A and Area B storage locations.
Test a storage location in Operand 2 that is the MSL minus the number of digits in the divisor minus one (1) for a Sentinel (110000):
If NO, go to Step 5,
If YES, go to Step 8: Division completed, align decimal.

Positioning The Decimal

Two steps follow which are used to position the decimal in the quotient.

STEP 8

Process: Superimpose transfer, descending.

Operand 1: All of Area A except the MSL, plus the MSL of Area B.

Operand 2: All of Area A, plus the MSL of Area B. Superimpose a space (Δ) in the storage location to the left of the storage location tested for a Sentinel in Step 7. To accomplish this the user must wire to an Address Combine:

a. The Row number of the storage location to receive the space.
b. The Column number of the storage location to receive the space.
c. The "X" hub of the Bit Present Emitter.
d. The "Y" hub of the Bit Present Emitter.

Go to Step 9.

STEP 9

Process: Subtract Absolute; Compare for Sign.

Operand 1: Storage location containing a 1 constant.

Operand 2: Area D.
Proceed as directed by sign of result determination made by the comparator:
If MINUS, exit from division subroutine.
If PLUS, go to Step 8.

INPUT EDIT

The editing of card input is largely a matter of rearranging information into a form suitable for machine processing. Several of the more common considerations of input editing are the recognition of control holes and the recognition of negative indicators. Each of these considerations is usually approached in the same fashion; that is, through the use of selectors. Wherever possible, as the card is read, the punching positions in question are wired to pick up selectors which are later used to direct the program along different paths, or to caused a sign or a value to be stored.

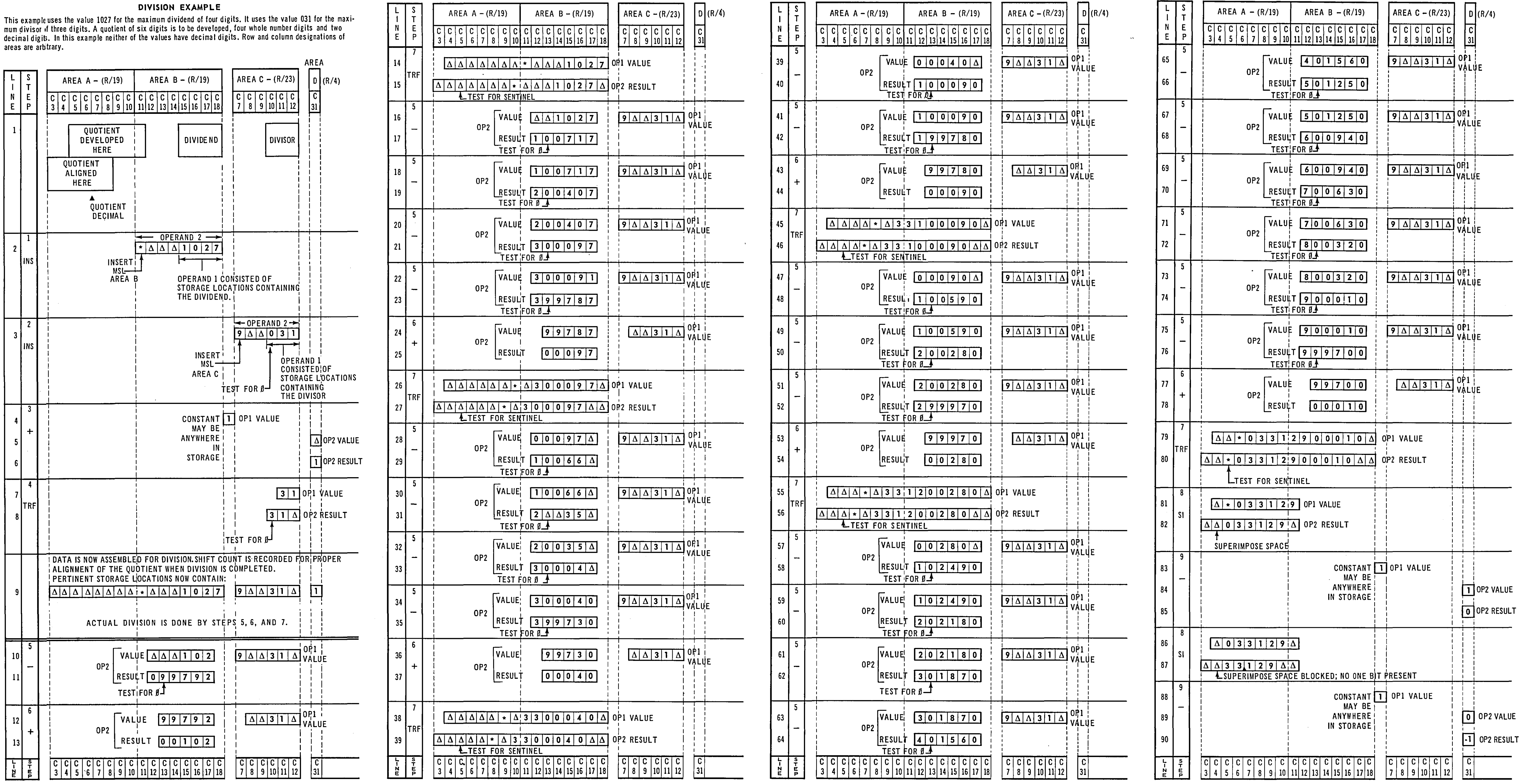
The presence of a specific punch in a card is usually detected by the program in a standard fashion. Assume that we wish to detect the presence of an X-punch in column 45, as the card is being read into storage. Column 45 is always placed in location R2/C14 in read storage, so that this is the location where we must detect the X-punch. The following chart lists the required wir-

DIVISION

DIVISION

DIVISION EXAMPLE

This example uses the value 1027 for the maximum dividend of four digits. It uses the value 031 for the maximum divisor of three digits. A quotient of six digits is to be developed, four whole number digits and two decimal digits. In this example neither of the values have decimal digits. Row and column designations of areas are arbitrary.


LINE
STEP
AREA A – (R/19)
AREA B – (R/19)
AREA C – (R/23)
AREA D (R/4)
QUOTIENT DEVELOPED HERE
DIVIDEND
DIVISOR
QUOTIENT ALIGNED HERE
QUOTIENT DECIMAL
OPERAND 2
INSERT MSL AREA B
OPERAND 1 CONSISTED OF STORAGE LOCATIONS CONTAINING THE DIVIDEND.
INSERT MSL AREA C
TEST FOR Ø
OPERAND 1 CONSISTED OF STORAGE LOCATIONS CONTAINING THE DIVISOR
CONSTANT MAY BE ANYWHERE IN STORAGE
OP1 VALUE
OP2 VALUE
OP2 RESULT
DATA IS NOW ASSEMBLED FOR DIVISION. SHIFT COUNT IS RECORDED FOR PROPER ALIGNMENT OF THE QUOTIENT WHEN DIVISION IS COMPLETED. PERTINENT STORAGE LOCATIONS NOW CONTAIN:
ACTUAL DIVISION IS DONE BY STEPS 5, 6, AND 7.
INS
TRF
SI
OP2
VALUE
RESULT
TEST FOR SENTINEL
SUPERIMPOSE SPACE
SUPERIMPOSE SPACE BLOCKED; NO ONE BIT PRESENT

NOTE:

*This example uses the symbol * to represent the sentinel for which machine code is 110000.*

Figure 2-16. Division Example.

ing, referenced by Wire Number to the accompanying plugboard segments on which the wiring is illustrated in Figure 2-17.

WIRE NO.	FROM	TO
1	Address Emitter R2	IN, Address Combine 1
2	Address Emitter C14	IN, Address Combine 1
3	Bit Present Emitter X	IN, Address Combine 1
4	OUT, Address Combine 1	ON, Program Select 3
5	Power, Program Select 3	Pick-up, Selector 5

Through this combination of wiring, Selector 5 is picked up when a card is read having an X-punch in column 45; that is, the three INs of the Address Combine are impulsed simultaneously, the OUT hub emits a signal to the Program Select, which in turn supplies power to the selector pick-up. When an X-punch is not present, only the two INs wired from the Address Emitter are impulsed, the OUT does not emit, and consequently the selector is not picked up. By routing control through Selector 5, we can cause whatever alteration of the program is required by the presence or absence of an X-punch in column 45.

This same form of selector pick-up wiring is used regardless whether the punch in question is a control hole to indicate card type, a negative sign control, or some other type of control. However, the wiring of impulses THROUGH the selector differs in each case.

Control Punches

In the broadest sense, control punches are used to alter the performance of the program in some manner. For instance, it is customary to identify different types of cards with significant control punches, and to use these punches to route the program in different directions for the various types of cards. One example of this distinction might be to identify a header card with a 1-punch in column 1 and a detail card with a 2-punch in column 1. This example assumes that only these two punches are possible in column 1. The wiring to accomplish selector pick-up from these two control punches is shown in Figure 2-18.

WIRE NO.	FROM	TO
1	Address Emitter R1	IN, Address Combine 1
2	Address Emitter C1	IN, Address Combine 1
3	Bit Present Emitter 4	IN, Address Combine 1
4	Bit Absent Emitter 1	IN, Address Combine 1
5	OUT, Address Combine 1	ON, Program Select 4
6	Power, Program Select 4	Pick-up, Selector 4
7	Address Emitter R1	IN, Address Combine 3
8	Address Emitter C1	IN, Address Combine 3
9	Bit Present Emitter 4	IN, Address Combine 3
10	Bit Present Emitter 1	IN, Address Combine 3
11	OUT, Address Combine 3	ON, Program Select 5
12	Power, Program Select 5	Pick-up, Selector 5

It is especially important to note the effect of Wire 4, which is vital to the correct identification of the two punches. Since the XS-3 codes for the values 1 and 2 both contain a bit in the 4-level, the omission of Wire 4 would cause both selectors to be picked up when the 2-punch was detected. To prevent this unwanted selector pickup, Wire 4 is included to deliver a pulse from the Bit Absent Emitter to Address Combine 1 and thereby to cause Selector 4 to be picked up only when the 1-punch is present.

Note that this wiring is effective to pick-up the selectors *at the time the card is read.*

Let us assume that the program requires all cards to go through Steps 1 through 5, and thereafter, the header card to be routed to Step 7 and the detail card to Step 14. The wiring required to accomplish this is as follows:

WIRE NO.	FROM	TO
13	Process, Step 5	Common, Selector 4a
14	Select, Selector 4a	Step Sequence Change 7
15	Non-Select, Selector 4a	Common, Selector 5a
16	Select, Selector 5a	Step Sequence Change 14
17	Non-Select, Selector 5a	Indicator 1 and Halt

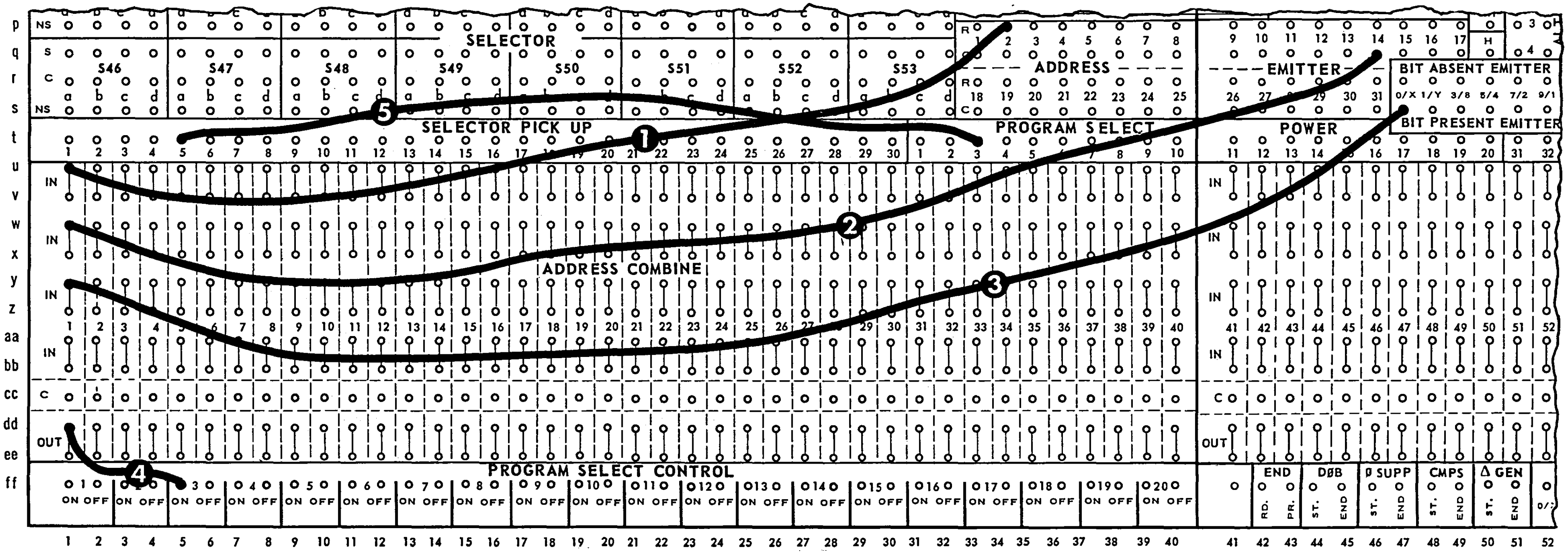


Figure 2-17. Selector Pick-Up Wiring.

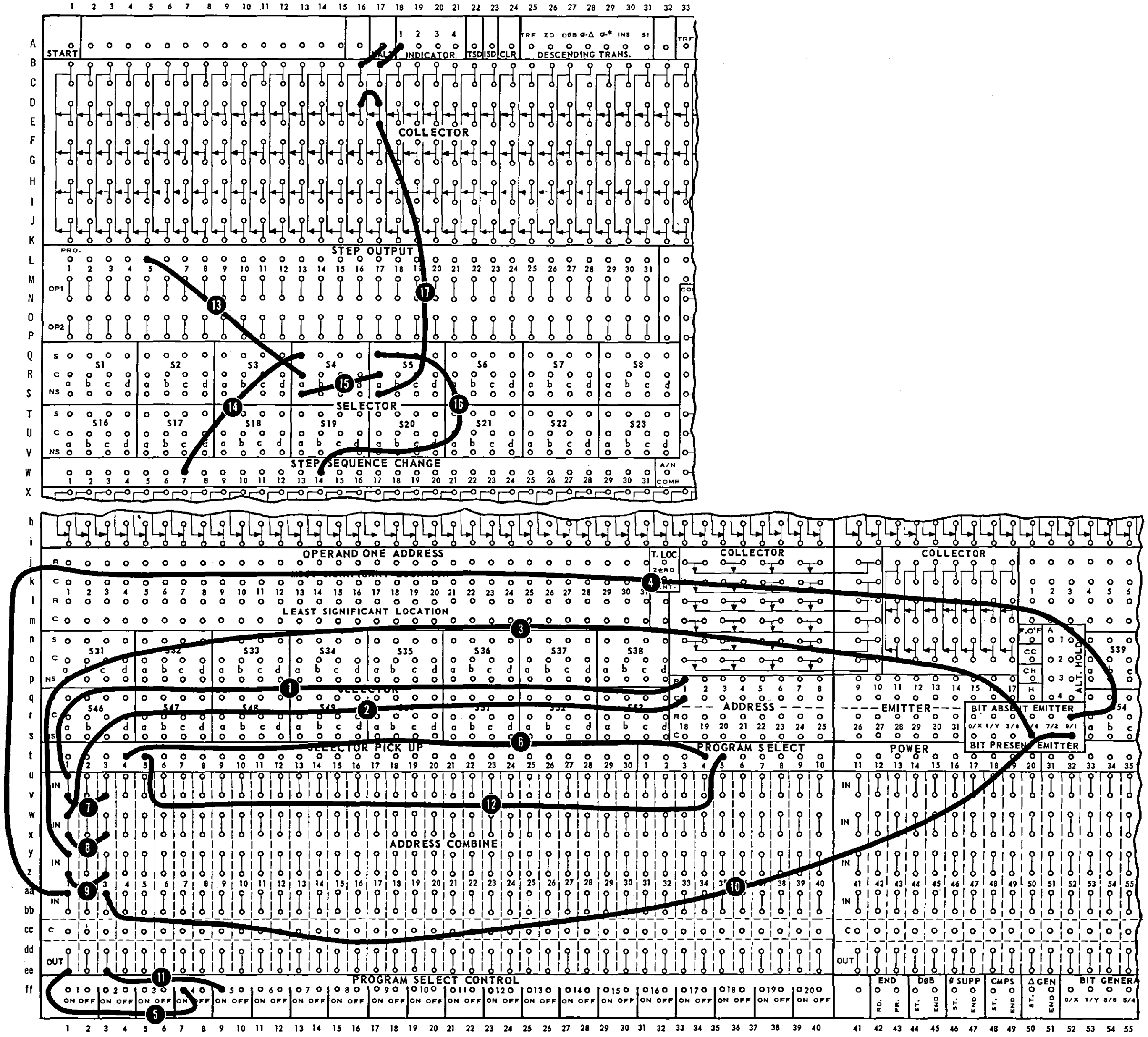


Figure 2-18. Wiring to Identify Card Type by Control Punching.

Note that the non-select side of Selector 5a is wired to indicate an error and to stop the machine. An impulse can be directed through the non-select side of 5a ONLY when neither Selector 4 nor Selector 5 is picked up, which means that column 1 of the card contains neither a 1- nor a 2-punch. Since this example specifies that each card must contain one or the other of these punches, an error condition exists and must be corrected.

The preceding example specified the possibility of only a 1- or a 2-punch in column 1. If this example is extended to included the possibility of punches from 1 to 9 in column 1, it is advisable to identify the punching in a different manner but also through the use of selectors.

Four Address Combines are wired to pick up selectors from each of the numeric bits in column 1, as follows:

WIRE NO.	FROM	TO
1	Address Emitter R1	IN, Address Combine 25
2	Address Emitter C1	IN, Address Combine 25
3	Bit Present Emitter 8	IN, Address Combine 25
4	OUT, Address Combine 25	ON, Program Select 12
5	Power, P.S. 12	Pick-up, Selector 5

Similarly, Bit Present Emitter hubs 4, 2, and 1 are wired to pick up Selectors 6, 7, and 8 respectively. Then the wiring shown in Figure 2-19, serves to identify each of the possibilities. Note that this wiring is representative – no attempt is made to show where each of the impulses is routed. Whatever purpose is to be accomplished by the identification of the various punches can be effected by appropriate wiring of the impulses. The circled numbers shown merely indicate the identification of the specific value punched in column 1 of the card.

Negative Sign Indication

As previously stated, it is customary to indicate a negative sign for a card field by the presence of a control punch, usually an X-punch in the least significant column of the card field. When this is true, the automatic translation of input information as the card is read causes the X-bit to be inserted in the appropriate location in storage to be used as a negative sign for the card field. However, it is possible that some other punch may be used as a negative sign indication. In that case, it is necessary to perform an editing operation to insert the X-bit in storage.

For example, a card field in columns 46 to 50 is to be considered negative if there is an X-punch in column 79. In this case, the X-punch in column 79 is wired to pick up a selector as the card is being read, using the following wiring (Figure 2-20):

WIRE NO.	FROM	TO
1	Address Emitter R3	IN, Address Combine 22
2	Address Emitter C17	IN, Address Combine 22
3	Bit Present Emitter X	IN, Address Combine 22
4	OUT, Address Combine 22	ON, Program Select 12
5	Power, Program Select 12	Pick-up, Selector 39

Then, after the card has been read, Selector 39 is used to indicate the sign of the card field. One way to do this is to transfer the information in this card field to a working storage location and to superimpose the sign indication, if any, at the same time. This transfer must, of course, be accomplished by a program step.

It is important to keep the following distinction in mind when inserting or superimposing characters or bits based on punches in cards.

1. The CHARACTER OR BIT to be inserted or superimposed depends upon one or more selectors, which may be picked up from any position in the card, preferably at the time the card is read.

2. The LOCATION into which the character or bit is to be inserted or superimposed depends upon the storage designated as Operand 2 on the program step which performs transfer.

In the case of this sign detection example, the wiring described above accomplishes the selector pick-up during the read operation. Thus, the remainder of the problem consists of specifying the wiring to perform the actual entry of the sign bit into storage.

If it is desirable to leave the card field information in its original locations in read storage, the

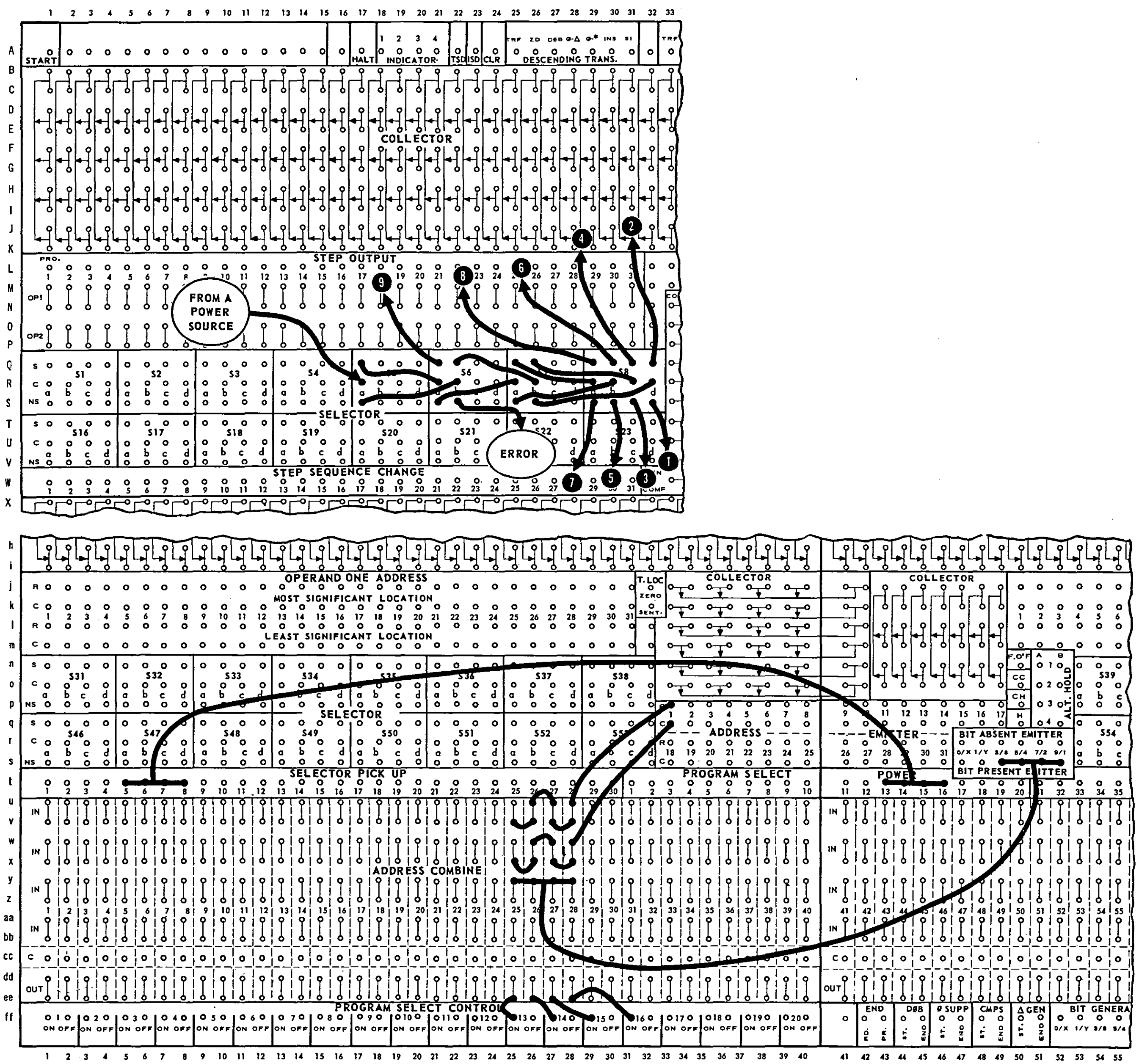


Figure 2-19. Wiring to Identify Punches.

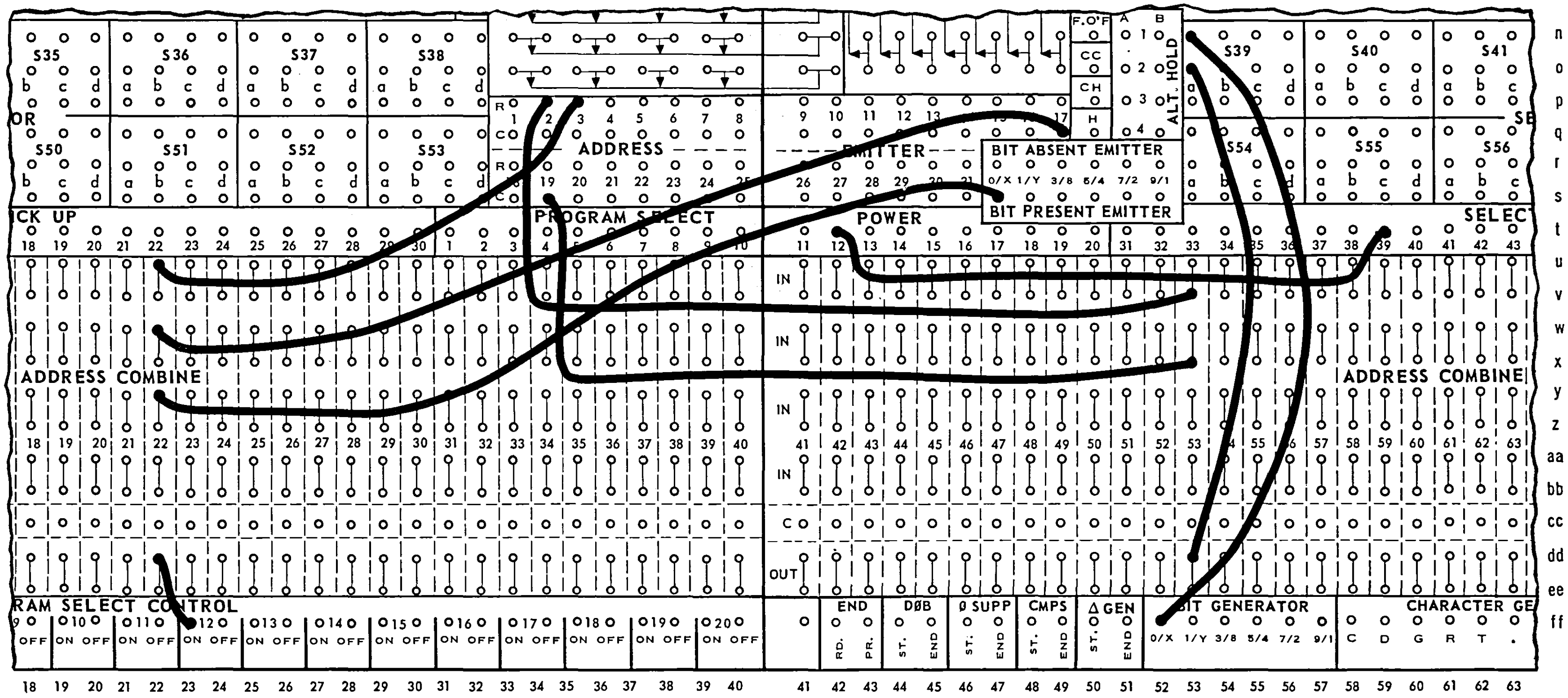


Figure 2-20. Negative Sign Control Wiring.

fastest method of including the sign bit is to perform a transfer of only the LSL of the field to the same location in which it is presently located, superimposing an X-bit through the use of the selector. In the example cited above, the step would be wired using location R2/C19 as both Operand 1 and Operand 2, the Superimpose option of transfer, and the following additional wiring (Figure 2-20):

FROM	TO
Address Emitter R2	IN, Address Combine 53
Address Emitter C19	IN, Address Combine 53
OUT, Address Combine 53	Common, Selector 39a
Select, 39a	Bit Generator X

Thus, when Selector 39 is picked up by the presence of an X-punch in column 79, the wiring shown above will cause an X-bit to be superimposed on the information in the LSL of the card field. Subsequent use of this field in arithmetic processes can then insure that the sign indication will be effective. Note that if the X-punch in column 79 is not present in the card, Selector 39 will not be picked up, and the step shown above will merely transfer the data originally stored in R2/C19.

If it is desirable to transfer the card field information to some other storage locations, the same type of wiring permits the X-bit to be superimposed. In that case, the wiring to Address Combine 53 should be from the Address Emitter hubs associated with the LSL of Operand 2.

OUTPUT EDIT

Printing

Several important considerations in editing for printout include the alignment of information into proper order for printing, suppression of non-significant zeros, insertion of symbols and punctuation, and conversion of a machine minus code to an appropriate printer symbol.

Alignment of information may involve rearrangement of fields or possibly only the insertion of spaces between fields. For example, in a straight listing which simply prints out the contents of a number of cards, it is desirable to insert spaces between fields for ease in reading the printed result. Similarly, while there are a number of reasons for storing data in consecutive locations (to accumulate totals of several adjacent card fields in one step, for instance), it is seldom desirable to print this data in consecutive printing positions. In each of these instances the solution of the problem is to insert spaces when transferring the data into print storage.

If only one space is to be inserted, the desired result is achieved by use of the Insert transfer process and the Character Generator Space hub. Assume two card fields in consecutive storage locations, the first field in R1/C15–R1/C17 and the second in R1/C18–R1/C22. The first field is to print in positions 2–4 and the second in positions 6–10, with a space in position 5. It is possible to transfer both fields and to insert the space on one program step by means of the following wiring:

Process: Insert

Operand 1: MSL R1/C15
LSL R1/C22

Operand 2: MSL R6/C7 (Print Storage 2)
LSL R6/C15 (Print Storage 10)

Print position 5 is represented in print storage by location R6/C10. The following wiring will cause a space code to be inserted in this position during the insert step described above.

WIRE NO.	FROM	TO
1	Address Emitter R6	IN, Address Combine 55
2	Address Emitter C10	IN, Address Combine 55
3	OUT, Address Combine 55	Character Generator Space

This wiring is illustrated with solid lines in Figure 2-21. The Character Generator Space hub should always be used where only one space is to be inserted.

When printing areas are to be separated by the insertion of several consecutive spaces, a slightly different technique is used. This method makes use of the Space Generate – Start and End hubs. The wiring of these hubs defines the beginning and ending locations in Operand 2 where spaces are to be inserted. It is possible to insert successive spaces in several locations within the Operand 2 area by wiring more than one location to each of

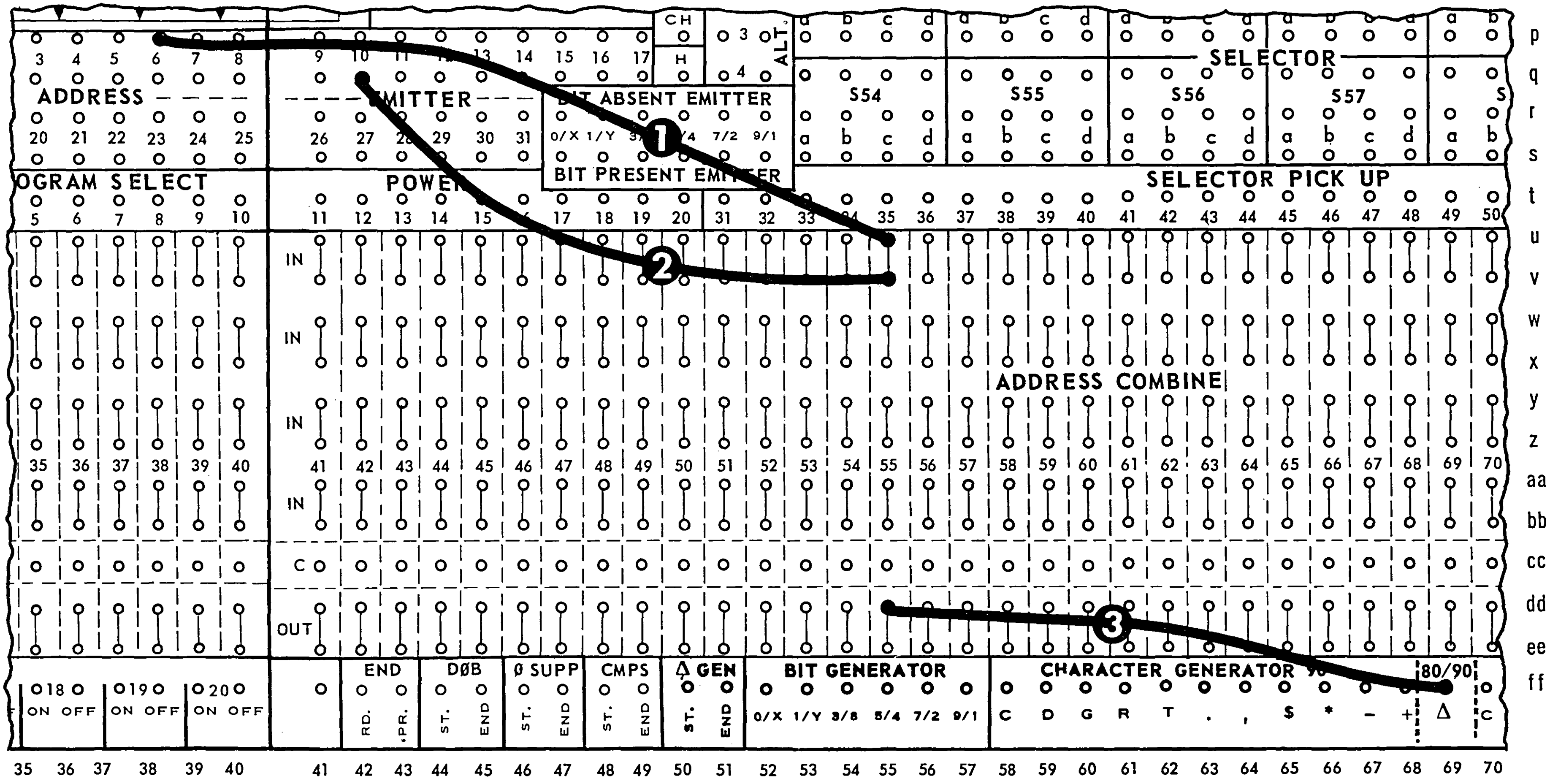
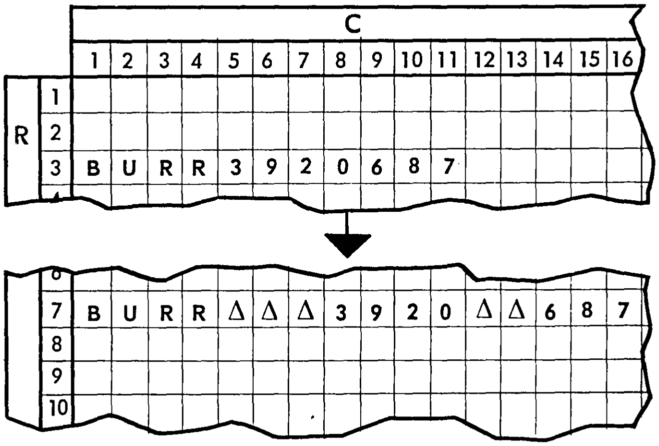


Figure 2-21. Wiring to Insert a Space.

these hubs. For example, data in locations R3/C1 – R3/C11 is to be transferred to print storage areas as shown in Figure 2-22.



R \ C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1																
2																
3	B	U	R	R	3	9	2	0	6	8	7					

R																
7	B	U	R	R	Δ	Δ	Δ	3	9	2	0	Δ	Δ	6	8	7
8																
9																
10																

Figure 2-22. Space Insert.

One program step can transfer all this data and insert the correct number of spaces, by means of the following wiring:

Process: Insert

Operand 1: MSL R3/C1
LSL R3/C11

Operand 2: MSL R7/C1
LSL R7/C16

Additional wiring to define the start and end of the two areas in which spaces are to be generated is as follows (Figure 2-23):

WIRE NO.	FROM	TO
1	Address Emitter R7	IN, Address Combine 41
2	Address Emitter C5	IN, Address Combine 41
3	OUT, Address Combine 41	Space Generate - Start
4	Address Emitter R7	IN, Address Combine 42
5	Address Emitter C7	IN, Address Combine 42
6	OUT, Address Combine 42	Space Generate - End
7	Address Emitter R7	IN, Address Combine 44
8	Address Emitter C12	IN, Address Combine 44
9	OUT, Address Combine 44	Space Generate - Start
10	Address Emitter R7	IN, Address Combine 45
11	Address Emitter C13	IN, Address Combine 45
12	OUT, Address Combine 45	Space Generate - End

Any number of Space Generate Start and End addresses may be specified within the limits of Operand 2 on a single step. It is also possible to combine the insertion of single spaces and multiple spaces (as illustrated in these two examples) in a single step.

Zero Suppression

The suppression of non-significant zeros is an important consideration of output editing. The 1004 processor provides two zero-suppress transfer instructions that ,replace non-significant zeros with either spaces or asterisks at the option of the programmer. The rules governing the operation of these processes are detailed in the section of this chapter called "Transfer."

Removing non-significant zeros from totals is probably the most frequent use of zero suppression in editing print output. The result of an arithmetic step is "packed" with zeros; that is, any location not containing a significant digit is filled with a zero. Thus, any total accumulated in the machine is zero-filled. For ease of reading the printed output, most applications require the elimination of of zeros preceding the most significant digit of a total amount.

Areas in storage to be zero-suppressed are indicated by the wiring of Zero-Suppress Start and End hubs in addition to one of the zero-suppress processes. As in any step, two operands must be defined by wiring. Zero-suppress start and end addresses are always wired with reference to Operand 2 locations. Any number of zero-suppress start and end addresses may be specified within the limits of Operand 2 on a single step.

For example, assume a totals-only operation in which the totals have been accumulated in print storage in the locations indicated in Figure 2-24. Before printing these totals, it is necessary to suppress zeros preceding the first significant digit of each of the totals. Since these totals are already located in the appropriate locations in print storage, the step to perform the zero-suppression specifies the same locations for Operand 1 and Operand 2.

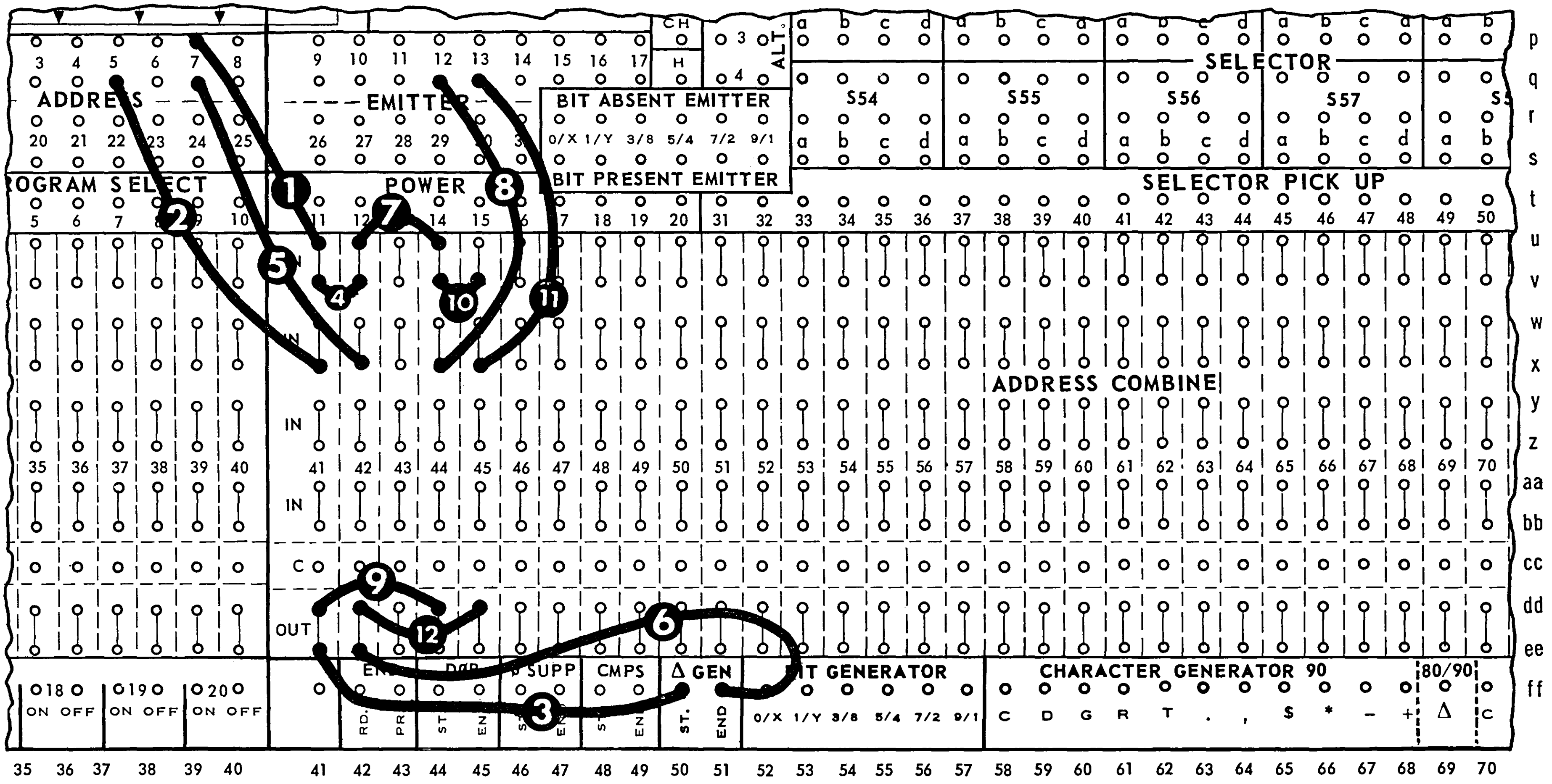


Figure 2-23. Space Generate Wiring.

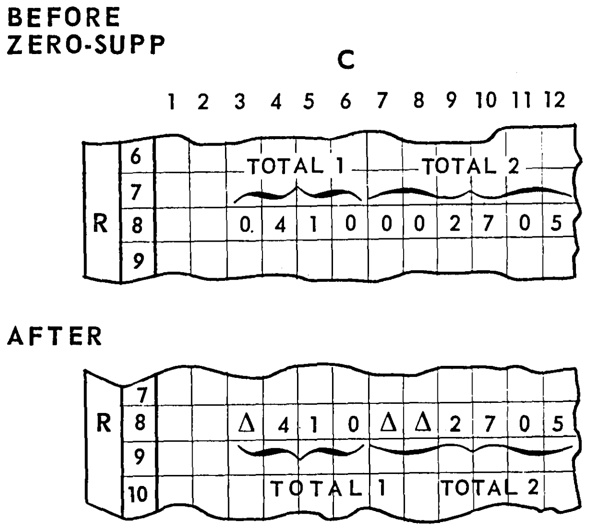


Figure 2-24. Totals in Storage Before and After Zero-Suppression.

Step wiring is as follows:

Process: Zero-Suppress Space – Descending

Operand 1; MSL R8/C3
LSL R8/C12

Operand 2: MSL R8/C3
LSL R8/C12

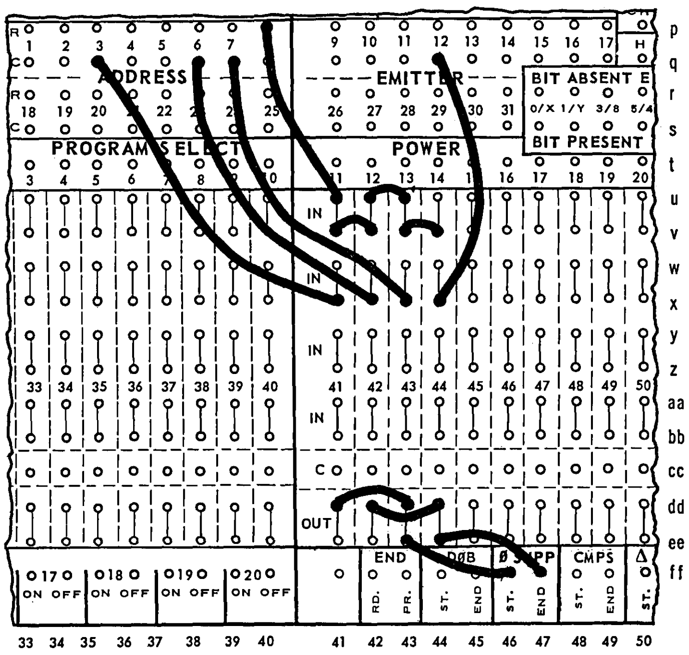


Figure 2-25. Zero-Suppress Wiring.

Additional wiring necessary to create the desired output is as follows (Figure 2-25):

FROM	TO
Address Emitter R8	IN, Address Combine 41
Address Emitter C3	IN, Address Combine 41
OUT, Address Combine 41	Zero-Suppress Start
Address Emitter R8	IN, Address Combine 42
Address Emitter C6	IN, Address Combine 42
OUT, Address Combine 42	Zero-Suppress End
Address Emitter R8	IN, Address Combine 43
Address Emitter C7	IN, Address Combine 43
OUT, Address Combine 43	Zero-Suppress Start
Address Emitter R8	IN, Address Combine 44
Address Emitter C12	IN, Address Combine 44
OUT, Address Combine 44	Zero-Suppress End

Note that this combination of wiring removes zeros BEFORE the most significant digit in each zero-suppress field, but leaves any zeros unaltered in the body of a field as in locations R8/C6 and R8/C11 in the example in Figure 2-24.

The replacing of zeros with asterisks is most frequently used in check-writing applications, where the asterisks serve the function of a built-in check-protector. The Zero-suppress Start and End hubs are wired from the OUT hubs of Address Combines as illustrated in the preceding example. The only difference would be the use of Zero-Suppress Asterisk as the process of the step.

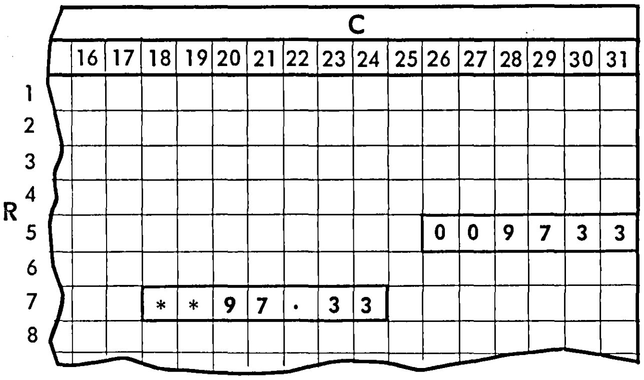
Assume a payroll program in which the net pay amount has been developed in working storage locations R5/C26 – R5/C31. The problem specifications state that net pay will never exceed five digits (though it may be less than five) and that the net pay amount must be preceded by at least one asterisk. It is also desired to insert a period between the dollars and cents portions of the net pay figure when the check is printed. (The dollar sign is pre-printed on the checks.) (See Figure 2-26).

Step wiring is specified as follows:

Process: Zero-Suppress Asterisk and Insert – both Descending

Operand 1: MSL R5/C26
LSL R5/C31

Operand 2: MSL R7/C18
LSL R7/C24



R \ C	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1																
2																
3																
4																
5											0	0	9	7	3	3
6																
7			*	*	9	7	.	3	3							
8																

Figure 2-26. Asterisk and Decimal Insert.

Additional wiring to create the desired output is follows (Figure 2-27):

FROM	TO
Address Emitter R7	IN, Address Combine 41
Address Emitter C18	IN, Address Combine 41
OUT, Address Combine 41	Zero-Suppress Start
Address Emitter R7	IN, Address Combine 43
Address Emitter C21	IN, Address Combine 43
OUT, Address Combine 43	Zero-Suppress End
Address Emitter R7	IN, Address Combine 45
Address Emitter C22	IN, Address Combine 45
OUT, Address Combine 45	Character Generator-Period

Note that the same Zero Suppress Start and End hubs are wired as in the previous example. The machine determines *on the basis of the PROCESS that is wired* whether to insert spaces or asterisks in place of non-significant zeros.

Note also that, to place at least one asterisk ahead of the net pay amount, Operand 1 begins with R5/C26, which always contains a zero, and the zero suppress field begins with R7/C18. In the example shown, because net pay is only a four-digit number, the result is the insertion of two asterisks before the first significant digit.

Insertion of Punctuation and Symbols

The insertion of punctuation and symbols is another important consideration of editing print output. The rules governing the Insert transfer process are detailed in the section of this Chapter entitled "Transfer." Remember that the wiring of locations for insertions is always done with reference to Operand 2 locations, and that the number of characters to be inserted must be included in determining the number of locations required in Operand 2.

Insert transfer is frequently combined with zero suppression, especially in the case of machine-accumulated totals. These two processes used in combination make a powerful edit instruction. To illustrate this fact, consider the following problem and the various examples of printing which result from differences in the actual value in storage.

A money total accumulated in storage may contain any number of digits, from 1 to 7. This total is to be printed with two marks always present – a dollar sign preceding the field and a decimal point to separate dollars and cents. If the total is $1000 or more, a comma is to be printed between the hundreds-of-dollars digit and the thousands-of-dollars digit. All non-significant zeros are to be replaced with spaces.

The total is accumulated in locations R16/C1 – R16/C7. It is to be printed in positions associated with locations R9/C16 – R9/C25. The wiring for this step is as follows:

Process: Insert and 0-Δ,both Descending

Operand 1: MSL R16/C1
LSL R16/C7

Operand 2: MSL R9/C16
LSL R9/C25

Wiring to insert the necessary characters is as follows (Figure 2-28):

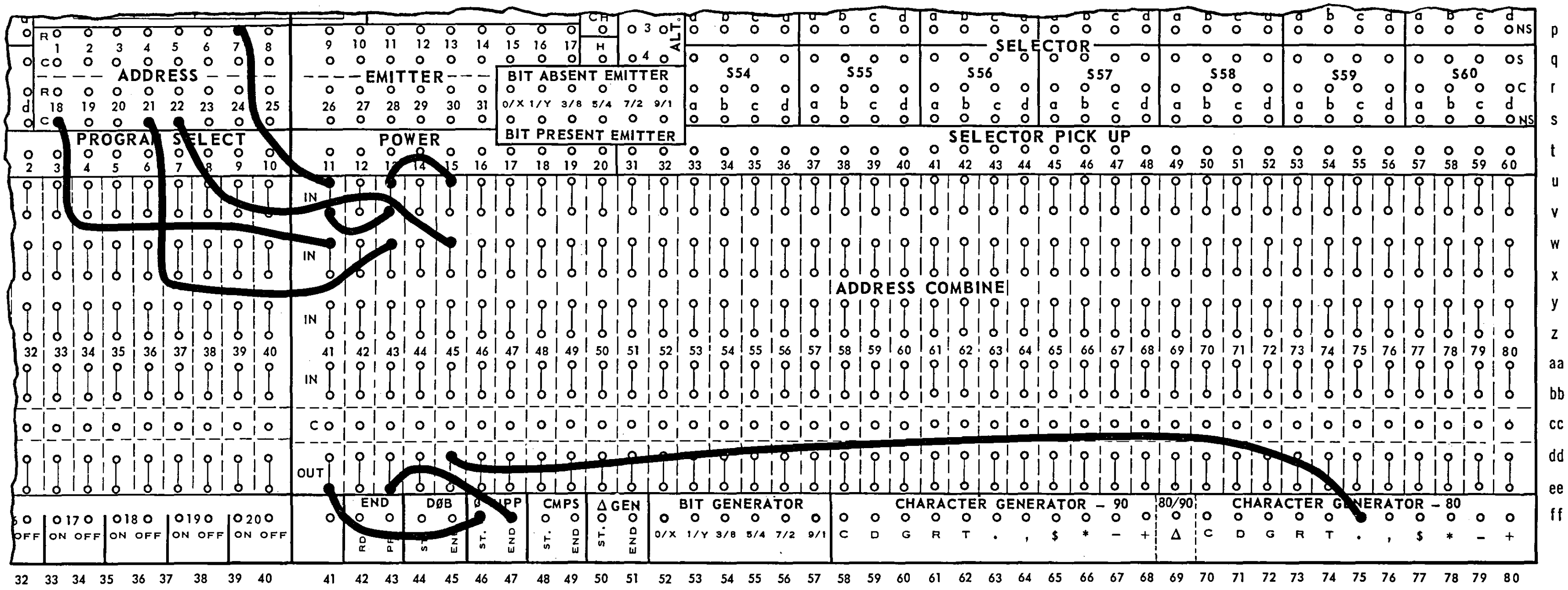


Figure 2-27. Zero-Suppress and Insert Wiring.

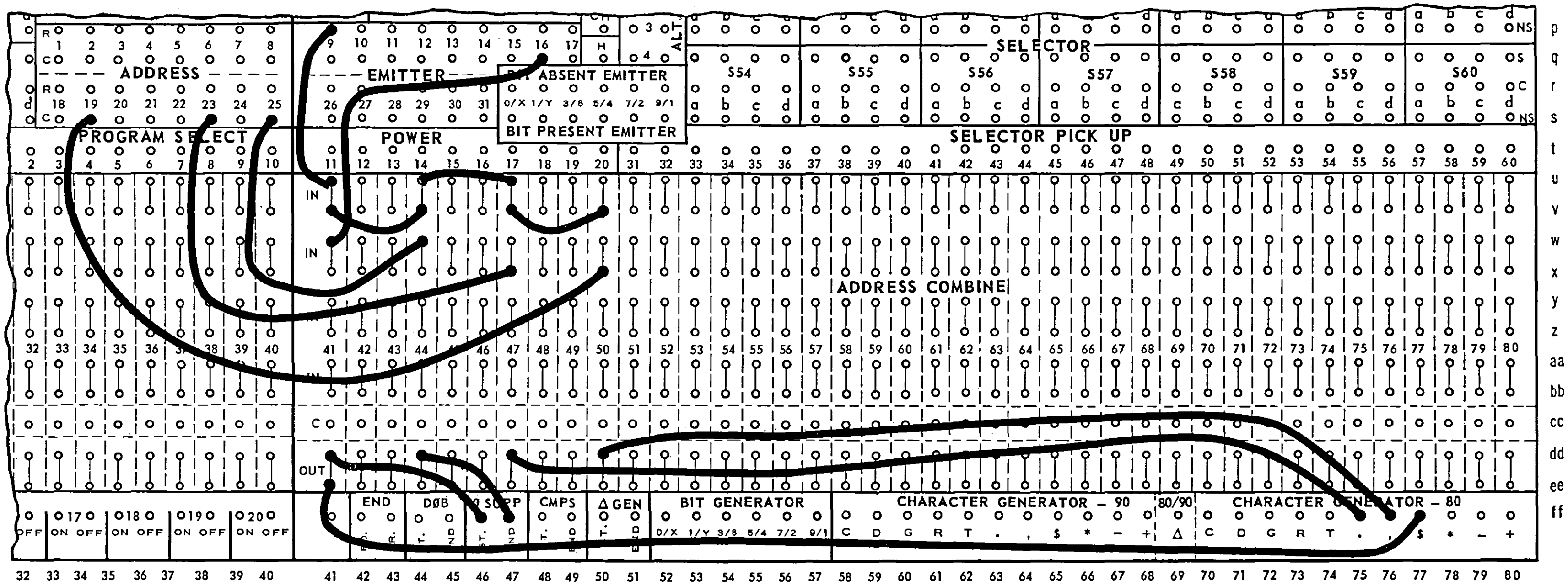
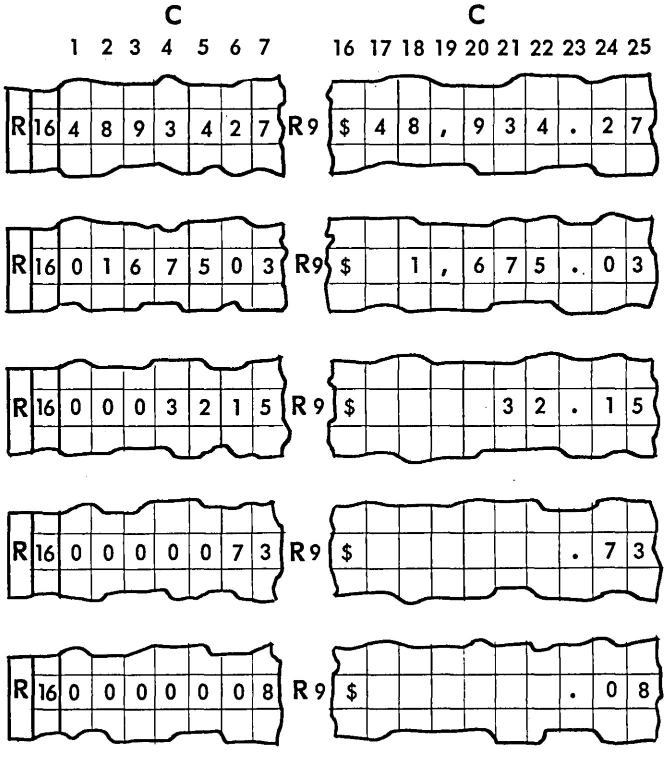


Figure 2-28. Wiring to Insert Punctuation and Symbols.

FROM	TO
Address Emitter R9	IN, Address Combine 41
Address Emitter C16	IN, Address Combine 41
OUT,Address Combine 41	Zero-Suppress Start
OUT,Address Combine 41	Character Generator $
Address Emitter R9	IN, Address Combine 50
Address Emitter C19	IN,Address Combine 50
OUT,Address Combine 50	Character Generator ,
Address Emitter R9	IN, Address Combine 47
Address Emitter C23	IN,Address Combine 47
OUT,Address Combine 47	Character Generator
Address Emitter R9	IN, Address Combine 44
Address Emitter C25	IN, Address Combine 44
OUT,Address Combine 44	Zero-Suppress End



	C 1	2	3	4	5	6	7		C 16	17	18	19	20	21	22	23	24	25
R16	4	8	9	3	4	2	7	R9	$	4	8	,	9	3	4	.	2	7
R16	0	1	6	7	5	0	3	R9	$		1	,	6	7	5	.	0	3
R16	0	0	0	3	2	1	5	R9	$					3	2	.	1	5
R16	0	0	0	0	0	7	3	R9	$							.	7	3
R16	0	0	0	0	0	0	8	R9	$							.	0	8

Figure 2-29. Representative Total in Storage.

Figure 2-29 shows the results which are created by this same wiring depending upon the number of digits actually developed in the total. (Space codes have been omitted from the diagram to emphasize the printing format.)

Line 1 shows the insertion of all marks when the total contains the maximum number of digits.

Line 2, a six digit total, shows the suppression of one non-significant zero after the insertion of the dollar sign; that is, the dollar sign is inserted but zero suppression continues.

Line 3 illustrates the selective insertion of the comma. In this case, the comma is NOT inserted in Operand 2 because no significant digit has been encountered.

Line 4 shows that the period IS inserted even though it is not preceded by a significant digit.

Line 5 shows that, not only is the period inserted, but that it also causes the end of zero suppression. Note that the zero in the ten-cents position is transferred to the Operand 2 location.

Note that, while the example shows the expected wiring of Zero-Suppress End, the last three connections are actually unnecessary since the insertion of the decimal point ALWAYS stop zero suppression.

Sign Indication

A vital consideration in editing print output is printing an indication that a field is negative. As stated before, the machine code for a negative sign is an X-bit in the LSL of an operand. It is necessary to edit the sign location in order to print a negative indication. Most applications indicate negative quantities by either a minus sign, the letter C, or the letters CR, any of which may be provided when the data is transferred to print storage.

Although common practice is to designate only negative values, the following example illustrates the printing of either a plus or a minus sign. Assume that a total has been accumulated in working storage locations R12/C14 – R12/C19. This total and its sign are to be printed in print position 124 to 130 (R10/C5 to R10/C11). Print position 130 is to print either a plus or a minus sign, depending upon the sign of the total.

The editing of this information can be accomplished in several ways. One possible method is outlined below.

This method assumes that the arithmetic step which caused the total to be accumulated also delivered a signal to Comparator 10 to establish the sign of the result. However, the setting of the comparator is not tested until the print-editing step which performs the transfer.

The step to accomplish the transfer is wired as follows:

Process:	Insert and Zone-Delete, both Descending.
Operand 1:	MSL R12/C14 LSL R12/C19
Operand 2:	MSL R10/C5 LSL R10/C11

In addition to this wiring, an Address Combine is wired as follows:

FROM	TO
Address Emitter R10	IN, Address Combine 42
Address Emitter C11	IN, Address Combine 42
OUT, Address Combine 42	TEST, Comparator 10
– OUT, Comparator 10	Character Generator –
+ OUT, Comparator 10	Character Generator +

Note the inclusion of the process Zone-Delete in this step. It is included to prevent the printing of an alphabetic character in the LSL when the minus sign is present.

> *Zone-Delete should always be specified in editing for print-out of fields which may carry a negative sign.*

The same type of wiring illustrated above could be used to print the letter "C" to indicate a negative amount. In that case, the – OUT of Comparator 10 is wired to Character Generator C and the + OUT to Character Generator Δ.

Punching

In some respects, editing for output punching is similar to editing for printing. Data must be arranged in punch storage in card format, and certain machine codes must be converted to other codes for output. The principles of data arrangement by means of the various transfer orders has already been covered in detail in previous sections of this manual. Hence, this section will deal with a consideration of code conversion – conversion of a machine minus code to a punching position code.

Punching Negative Sign Controls

As noted in the section on input edit, the common practice is to punch an X in the least significant column of a card field to indicate a negative sign. When this is true, no special editing is required since the X-bit in the LSL of a field, which indicates the negative sign in machine code, is automatically translated to an X-punch as output.

However, many applications require that a negative sign for an amount be punched out as a control hole in a column not in the card field assigned to contain that amount. There are various methods of achieving this result; the specific method used depends upon the program. The following example illustrates one possible method.

Assume a total has been accumulated in locations R12/C11 – R12/C16 (sign in LSL) to punch in columns 60-65. If the total is negative, two control punches are required – a X-punch in column 65 and a 12-punch in column 78. Column 78 may or may not contain other control punches. The step which accumulates the total is wired to sign compare (Comparator 9) as well as to the arithmetic process.

As noted above, no special editing is required for the field in columns 60-65. However, the creation of the 12-punch in column 78 requires that an edit step be performed. One typical method of performing this editing function is described below.

Assume that after Step 25 of the program has been completed, it is desired to perform Step 42 to edit the Y-bit into column 78 if a negative total has been developed. A Step Output pulse from Step 25 is routed to the TEST hub of Comparator 9, and the OUT of Comparator 9 is wired to Step Sequence Change 42. (If the total is a positive number, the program proceeds by normal step advance to Step 26.) Step 42 is wired to perform the process Superimpose, using location R12/C29 (associated with column 78) as both Operand 1 and Operand 2. Address Emitter hubs R12 and C29 are wired to two IN's of an Address Combine, and the OUT of the Combine is wired to Bit Generator Y.

A pulse from Step 42 is also wired to Step Sequence Change 26 to return the program to the normal line of flow.

If no comparator is available to route the signal, the same effect could be achieved through the use of a selector, picked up from the presence of the X-bit in location R12/C16. This procedure would require an additional transfer step to pick up the selector.

FORM CONTROL

The control of forms to be printed by the 1004 processor is accomplished by a combination of the input-output functions of Print, Execute, Space 1 and 2, and Skip 1, 2, and 4. The execution of these functions, alone or in combination, determines the physical appearance of the printed form.

As previously stated, the Print order by itself merely alerts the processor that a print operation is to be performed when the Execute order is received. Of course, it is possible to impulse Print and Execute on the same program step, in which case the printing operation immediately follows the execution of whatever arithmetic or logical process is specified by that step.

It is important to remember that Print-Execute by itself does not involve any movement of the form in the print section. This form advance is controled by either a Space or a Skip order, both of which function in much the same manner.

Space

The orders Space 1 and Space 2 may be signaled on any step in the program. When either Space 1 or Space 2 is signaled on the same step which impulses Print, the form advance of either one or two lines is delayed until *after* the printing of that line. When either space order is signaled alone; that is, not accompanied by Print – the spacing operation is initiated immediately.

The maximum number of lines which can be advanced in a single step through a Space order is two. Even if Space 1 and Space 2 are impulsed on the same step, the form is advanced only two lines. If more than two lines are to be spaced between lines of printing, it is necessary either to give another space order on another step or to use a skip.

Skip

The operation of the Skip functions is similar to that of Space 1 and 2. A skip function may be signaled on any step. When Skip is signaled on a step which also impulses Print, the Skip function is effective *after* the line has been printed. If Skip is signaled on any other step, the skipping action is initiated immediately.

Form Advance

The delayed and immediate form advance features may be utilized by the programmer to space the form according to the requirements of the program. For instance, when the form is to be single-spaced throughout, Space 1 should be signaled on the same step which signals Print. Similarly, for a form to be double-spaced throughout, Space 2 should be signaled.

However, when a form is to be variably spaced, according to program-detected relationships between the various cards, Print-Execute could be given without an accompanying Space order. The space order could then be given later, after the relationship between the cards has been determined. For example, in the case of a form to be double-spaced between lines relating to the same total but single-spaced before the total is printed, this ability can be utilized to good effect. As each card is processed and a line is printed, no space order accompanies the Print-Execute. After each line is printed and the next card is read, the designating information from the new card is compared to that stored from the previous card. Where an "equal" result is detected by the comparison, a Space 2 order is given. When a change of designation is signaled by an unequal result (usually "greater than"), a Space 1 order is signaled to position the form for printing the total.

Where possible, it is advisable to signal Skip or Space simultaneously with Print. However, if either Skip or Space is signaled separately, it should be impulsed as far in advance of the next Print-Execute as possible, since an automatic interlock will prevent the print section from executing another Print order until the form advance has been completed.

It is important to distinguish between the operation of the Space and Skip functions. Space 1 and 2 perform exactly as the names state; that is, they cause the paper form to be advanced either one or two line-spaces. When Skip 1, 2, or 4 is impulsed, however, the form is advanced to a pre-determined line which may be many lines distant.

In connection with the Skip functions, the operation of the paper-tape carriage-control loop must be understood. The size of the form to be printed determines the length of the paper-tape loop. Punches in the three channels of the loop indicate various pre-determined stopping places for form advance. The channels of the loop are numbered 1, 2, and 4, corresponding to the three Skip hubs on the connection panel. When Skip 1 is impulsed, both the loop and the form move until a punch is detected in only the 1-channel of the loop. Movement of the form and the loop stops when this condition is detected. When two Skip functions are impulsed simultaneously, the form and the loop advance until punches are detected in both of the corresponding channels of the loop, at which time movement is stopped.

There are a total of seven unique stopping-points available through the use of single and combination punches in the paper-tape loop. Two of these seven positions (the 1 punch alone and the combination of all three punches – 1, 2, and 4) have been assigned specific meanings although their use is not limited to these meanings; the remaining five may be used in any manner required by the program. These two pre-assigned skips are described below.

Skip 7 has been assigned as the home paper position, which is usually the first line to be printed on each form. The first line to be printed should always be indicated in the carriage-control loop by a punch in all 3 channels.The Home Paper button on the control panel, when pressed, always causes the form to advance to this point, thus making it easy for the operator to position the form correctly before beginning a run.

Skip 1 has been assigned the function of signaling the completion of one page of a form. Most applications require an indication when the last desired line of a form has been printed, either to prevent printing over the perforations between forms or to allow for the possibility of multiple-page items such as invoices. The carriage-control loop should always be punched in the first channel at the line corresponding to the last line desired for printing in the body of a form. The mechanism of the carriage, sensing that the first channel is punched, causes a signal to be delivered from the Form Overflow hub on the connection panel. The Signal is emitted only when the form is being printed on a line where Skip 1 punching is present in the paper loop. Advancing the form *past* this line will not cause Form Overflow to emit. Thus, if variable spacing in the body of the form means that the final line of printing may occur at one of several lines on the form, all such possible lines of the paper loop should contain Skip 1 punching.

It is important to remember that the Form Overflow signal is delivered *during* the printing operation. Therefore, the Form Overflow signal must be "remembered" in some manner because the signal itself will no longer be present after the print operation is completed. The method of remembering this signal is to deliver it to the ON of a Program Select, which picks up a Selector. Then the condition of this Selector can be tested by the program, using the non-select side to continue the normal flow, and the select side to direct the program into a sub-routine to accomplish any desired results (print page totals, advance the form, print second-page headings, and so forth).

Note that the function of Form Overflow is to signal the last desired line in the body of a form. After the condition is detected, it is still possible to print additional lines on the form – such as page totals, or balance forward sub-totals – at the discretion of the programmer. The Form Overflow signal is simply a means of recognizing that the condition exists; the programmer has complete control over what use is made of the signal.

V. MULTIPLE OPERATIONS AND TIMING

MULTIPLE OPERATION PROGRAM STEP

The chart in Figure 2-30 shows the various machine operations which can be performed during a single program step. To determine allowable combinations, find the principal operation of the step in the Principal Operations column. Reading across the page, any auxiliary operation marked with an X may be combined with the principal operation on any one step.

Any number of operations marked with an X in any one line may be wired to one program step, with the following considerations:

1) Only one auxiliary operation with a like sub-number (1-5) can be used on one step.

2) Group I (Arithmetic) operations CANNOT be combined with Group II (Logical) operations.

3) Group III (Input-Output) operations can be combined with either Group I or Group II operations.

When multiple-operation steps are used, the following sequence of execution occurs:

A. Group I (Arithmetic) combined with Group III (I/O).

1. Group I (Arith.) executed first.
2. Group III (I/O) executed last.

B. Group II (Logical) combined with Group III (I/O).

1. Group II (Logical) executed first.
2. Group III (I/O) executed last.

Any allowable combination of operations within Group I or within Group II are executed simultaneously on a multiple-operation step.

All specified input-output functions are normally executed simultaneously during a multiple-operation step. However, if a form advance is still in progress from a previous instruction, a Print-Execute or a new form advance instruction will be delayed until completion of the previous form advance. If a form advance instruction is given with a Print instruction, the form advance is delayed until the print has been executed.

A Punch Test feature is provided. When Punch Test is interrogated on any given step, step advance is delayed until any punching operation in progress has been completed.

If an Arithmetic or Logical process is not included as part of a step, NO PRO (No Process) must be wired to prevent stalling the processor.

PRINCIPAL OPERATIONS	AUXILIARY OPERATION																			
	GROUP 1 ARITHMETIC						GROUP II LOGICAL								GROUP III INPUT-OUTPUT					
	ADD ALG	ADD ABS	SUBT ALG	SUBT ABS	SIGN COMP	COMP	TRF	ZD	ZDS (Ascending only)	DØB (Descending only)	Ø-Δ	Ø-*	IN	SI	SP (1-2)	SK (1-4)	PR	RD	EX	PU (H-C)
ADD. ALG.					X										X	X	X	X	X	X
ADD. ABS.					X										X	X	X	X	X	X
SUBT. ALG.					X										X	X	X	X	X	X
SUBT. ABS.					X										X	X	X	X	X	X
SIGN COMP.	X1	X1	X1	X1											X	X	X	X	X	X
COMP.															X	X	X	X	X	X
TRF.								X2	X2	X	X3	X3	X4	X4	X	X	X	X	X	X
ZD							X			X	X3	X3	X4	X4	X	X	X	X	X	X
ZDS							X				X3	X3	X4	X4	X	X	X	X	X	X
DØB							X	X			X3	X3	X4	X4	X	X	X	X	X	X
Ø-Δ							X	X2	X2	X			X4	X4	X	X	X	X	X	X
Ø-*							X	X2	X2	X			X4	X4	X	X	X	X	X	X
IN							X	X2	X2	X	X3	X3			X	X	X	X	X	X
SI							X	X2	X2	X	X3	X3			X	X	X	X	X	X
SP (1-2)	X1	X1	X1	X1	X5	X5	X	X2	X2	X	X3	X3	X4	X4		X	X	X	X	X
SK (1-4)	X1	X1	X1	X1	X5	X5	X	X2	X2	X	X3	X3	X4	X4	X		X	X	X	X
PR	X1	X1	X1	X1	X5	X5	X	X2	X2	X	X3	X3	X4	X4	X	X		X	X	X
RD	X1	X1	X1	X1	X5	X5	X	X2	X2	X	X3	X3	X4	X4	X	X	X		X	X
EX	X1	X1	X1	X1	X5	X5	X	X2	X2	X	X3	X3	X4	X4	X	X	X	X		X
PU (H-C)	X1	X1	X1	X1	X5	X5	X	X2	X2	X	X3	X3	X4	X4	X	X	X	X	X	
NO PRO.															X	X	X	X	X	X

NOTES: A. Only one operation with a like sub-number (1 through 5) can be performed on any one step.

B. Group I and Group II operations can not be combined on the same step. Groups I and III, or Groups II and III are allowable combinations.

Figure 2-30. Multiple Operations Chart.

TIMING

Because of variable-operand-length logic, and the ability to perform multiple operations during a single program step, the actual time required to complete a step is variable. The following basic timing enables the programmer to determine the time required for any given program step or series of steps.

Add or Subtract

56 microseconds for the first character in Operand 2.

16 microseconds for each additional character in Operand 2.

16 microseconds for recomplementing each character of a complementary result in Operand 2.

16 microseconds per program step when the signs of the operands are NOT alike.

Transfer

48 microseconds for the first character in Operand 2.

24 microseconds for each additional character in Operand 2.

8 microseconds for each Start or End of Zero-Suppress.

8 microseconds for each Start or End of Space-Generate or Superimpose.

16 microseconds for each character inserted.

Compare

56 microseconds for the first character in Operand 2.

16 microseconds for each additional character in Operand 2.

Read

61 milliseconds for the first column. Processor interlocked.

1.32 milliseconds for each additional column. Processor interlocked.

Print-Execute

150 milliseconds maximum. Processor interlocked.

Space 1

20 milliseconds. Processor NOT interlocked. Printer interlocked.

Space 2

28 milliseconds. Processor NOT interlocked. Printer interlocked.

Skip

20 milliseconds for one line space. Processor NOT interlocked. Printer interlocked.

8 milliseconds for each additional line space. Processor NOT interlocked. Printer interlocked.

On multiple-process and/or function steps the following rules govern the sequence of execution:

1) Multiple logical processes are executed simultaneously.

2) Multiple I/O functions are executed simultaneously, with the exception stated in (4) below.

3) When arithmetic or logical processes are combined with I/O functions, they are executed before the I/O functions.

4) When a space or skip function is combined with a Print instruction, the Print will be executed before the Space or Skip.

In Summary

Classification of processes and functions.

ARITHMETIC PRO.	LOGICAL PRO.	I/O FUNCTIONS
1. Add Algebraic	1. Transfer (All options)	1. Read
2. Add Absolute	2. No Process	2. Print
3. Subtract Algebraic		3. Execute
4. Subtract Absolute		4. Punch Hold
5. Compare		5. Punch Clear
6. Sign Compare		6. Punch Test
		7. Space (1–2)
		8. Skip (1, 2, 4)

For purposes of explaining machine timing, a program step may be considered to consist of four sub-steps:

1) Initiate

2) Execute

3) Special Results (Arithmetic and Compare Steps Only)

4) Step Advance

The chart in Figure 2-31 illustrates the sequence of events within a step.

SEQUENCE OF EVENTS WITHIN A PROGRAM STEP

TYPE	INITIATE	EXECUTE		SPECIAL RESULTS	ADVANCE
GENERAL	■ Test Selectors and Comparators for process change. ■ Processes and functions to be performed are determined. ■ Order of execution of multiple processes and functions is determined. ■ Arithmetic Overflow is cleared. Delay for step control circuit switching.	■ Processes and functions are executed. ■ Form Over-Capacity is signaled. ■ Data in Operand 2 may be interrogated.		■ Arithmetic Overflow is signaled. ■ Arithmetic Result Sign is determined. ■ Comparison Result is determined.	■ Test for Step Sequence Change. ■ Step advance.
ADD or SUBTRACT	■ Operand signs examined for algebraic determination of process.	■ Arithmetic is completed.		■ Result Sign is determined. ■ Arithmetic Overflow is signaled.	■ Test for Step Sequence Change. ■ Step advance.
COMPARE	■ Numeric or Alphanumeric process determined. ■ Comparator cleared.	■ Comparison is made.		■ Comparison Result is set.	■ Test for Step Sequence Change. ■ Step advance.
TRANSFER	■ Processes determined. TRF, ZD, ZDS, Ø–Δ, Ø–*, IN, SI, DØB	■ (1) OP1 transferred to OP2. (2) Zeros deleted from OP2. (3) Zeros suppressed. (4) OP1 cleared. (5) Insertions made. (6) Bits superimposed. ■ Control positions examined. ■ Program Select turned on from Combine Out.			■ Test for Step Sequence Change. ■ Step advance.
PUNCH CLEAR (NP must be wired)		■ Punching starts. ■ Punch Storage area starts clearing.		■ Processor interlock is removed.	■ Test for Step Sequence Change. ■ Step advance. (Punching continues)
PUNCH HOLD (NP must be wired)		■ Punching starts. ■ Punch Storage area is not cleared.		■ Processor interlock is removed.	■ Test for Step Sequence Change. ■ Step advance. (Punching continues)
READ EXECUTE		■ Processor interlocked. ■ Data in the card is read and stored in the Read Storage area. ■ Read Storage area may be scanned for control or Over-Capacity bits. ■ Program Selects may be turned "ON" from Combines.		■ Reading is terminated when Read End address is reached. ■ Processor interlock is removed.	■ Test for Step Sequence Change. ■ Step advance.
PRINT EXECUTE		■ Processor Interlocked. ■ Printing occurs. ■ Form Overflow position is signaled. ■ Program Select may be turned "ON" from Over-Capacity.		■ Processor interlock is removed.	■ Test for Step Sequence Change. ■ Step advance.
PRINT and/or READ EXECUTE with ARITHMETIC or LOG. PRO.	■ Processes are determined. ■ Order of execution is established.	■ Arithmetic or Logical process is executed.	■ Print and/or Read is executed. ■ Form Overflow position is signaled.	■ Arithmetic Overflow and Result Sign is indicated. ■ Compare Result is set. ■ Processor interlock is removed.	■ Test for Step Sequence Change. ■ Step advance.
PRINT EXECUTE with SPACE or SKIP (NP must be wired)	■ Processes and functions are determined. ■ Order of execution is established.	■ Printing takes place. ■ Form Overflow position is signaled.	■ Space or Skip is started.	■ Processor interlock is removed.	■ Test for Step Sequence Change. ■ Step advance. (Spacing or Skipping continues to completion.)
PUNCH with ARITHMETIC and/or LOGICAL PROCESS	■ Processes and functions are determined. ■ Order of execution is established.	■ Arithmetic or Logical processes are executed.	■ Punching is started.	■ Special Results of Arithmetic and/or Logical processes are indicated. ■ Processor interlock is removed.	■ Test for Step Sequence Change. ■ Step advance. (Punching continues)

Figure 2-31. Sequence Chart.

VI. OPERATOR CONTROLS

The operator controls of the 1004 processor are located in three areas across the front of the machine. A single, centrally located panel houses the major machine controls. Four additional controls are located at the left of the processor, and are associated with the paper forms used by the printer. A display panel is mounted at the extreme right of the machine, and is referred to by the operator to monitor the progress of the program through the machine.

CENTRAL CONTROL PANEL

The central control panel is located to the right of the printer carriage. Its location, in relation to the input magazine, card stacker, and printer carriage, augments the one-work-station principle built into the 1004 processor to assure ease of operation. The panel contains various switches, buttons, and button-lights which govern reading, processing, and printing operations.

Power

At center left is the main power control of the processor. The control consists of two button-lights labeled *Power – On/Off*. The button-light labeled Power On serves two purposes. It is pressed once to connect AC and DC power to the machine. When all interlocks are complete the button light glows green.

The button-light labeled Power Off disconnects power to the processor when pressed. This button-light also serves to indicate that AC and DC power is being supplied. When the ON button is pressed AC power is supplied and the upper portion of the OFF button glows white. After a short warm-up period DC power is supplied and the lower portion of the OFF button glows white.

General Clear

The General Clear button performs four functions simultaneously. These functions are:

1. Clear all internal controls to an initial state.
2. Clear the Program Step Outputs to their no-output state.
3. Clear all error and fault indicators.
4. Turn off all Program Selects.

This button should be pressed before the start of each new program.

NOTE: General Clear does *not* clear storage.

Initial Start

The Initial Start button is pressed at the start of a run. Pressing this button causes the Step Outputs to clear to their no-output state, and provides a signal at the START hub of the connection panel.

Manual Feed

The Manual Feed Button-light is pressed to deliver a card from the input magazine to the wait station. Feeding is disallowed if a card is in the wait station, and/or a stop condition exists.

A feed control switch in the upper portion of the panel is provided to clear the read section and is used primarily for maintenance and program checking. When this control switch is operated and the Manual Feed button is pressed, card feeding is allowed regardless of stop conditions.

Alternate Hold

The Alternate Hold control consists of four button-lights. These buttons are related to the four Alternate Hold pairs of hubs on the Connection panel. When a button is OFF, it does not light, and constant power is available at the A hub of the associated Hold on the connection panel. Pressing the button turns on the light and provides constant power at the B hub of the associated Hold on the panel.

An additional depression of the button returns the button to the OFF condition.

Run

The Run button is pressed to initiate execution of the program steps wired on the connection panel. If the Run button is pressed after a depression of the Initial Start button, the program starts at the step wired from the START hub of the connection panel. If the Run button is pressed without a prior depression of the Initial Start button, the program resumes at the step on which it was stopped.

If processing was stopped by an abnormal condition which has been corrected, the abnormal indication is cleared by the Run button.

Stop

The Stop button-light is pressed to halt the processor. When pressed, the Stop button causes the processor to halt at the completion of the current step, and the button glows red.

Mode

The Mode switch is a four-position control which defines the manner in which processing is performed. The four positions are as follows:

CONTINUOUS: The processor operates continuously until a programmed halt is encountered, or an abnormal condition arises, or until the operator introduces a manual stop.

STEP: The processor is brought to a halt at the end of each programmed step.

CYCLE: The processor stops at the end of each memory cycle, except during the execution of Input/Output operations.

TEST: Provided for maintenance purposes.

Paper Advance

The Paper Advance control is a thumb wheel type control, and is used by the operator to advance the paper form manually. When the wheel is turned in an upward direction, the form can be advanced one line at a time, or any number of lines.

Left Tractors

The Left Tractors control knob is associated with the left-hand paper drive tractors of the printer. Movement of this knob in a clockwise direction moves the associated tractors to the right. Movement of the knob in a counter-clockwise direction moves the tractors to the left. The left tractors of the printer are positioned for varying widths of paper by rotation of this knob.

Right Tractors

The Right Tractors control knob operates in the same manner as the Left Tractors Knob, and is associated with the right-hand paper drive of the printer.

Disconnect Tractors

This control is a combination button and light. When the Disconnect Tractors button is pressed the right and left tractor assemblies are disconnected, and the button-light glows red. To link the assemblies the button is again pressed and the light is extinguished.

Home Paper

Depressing the Home Paper button causes the paper form in the printer to advance until the home paper code punched in the paper tape control loop is read.

Carriage In

The Carriage In button is pressed to cause the carriage to move toward the rear of the printer. When the button is released, carriage movement stops.

Carriage Out

Depressing the Carriage Out button causes the carriage to move toward the front of the printer. When the button is released, carriage movement stops. The carriage is positioned OUT to facilitate inserting or removing the paper forms. If the carriage is not fully positioned in the IN position, the Carriage Out light glows red.

Change Ribbon/Positioned

The Change Ribbon control is a combination button and light and is provided to facilitate replacement of the ribbon in the printer. A ribbon is changed most easily when the used ribbon is fully wound on the supply reel. When the Change Ribbon button is pressed, the upper portion of the button lights while the ribbon is winding onto the supply reel. After the ribbon is fully wound, the processor is brought to a halt, and the lower portion of the button, labeled *Positioned,* lights. Printing can continue during rewinding time.

PAPER FORM CONTROLS

Four dials are located at the front of the processor, to the left of the printer carriage. These dials serve to position the paper form in the printer carriage.

Form Thickness

The setting of this dial determines the distance between the surface of the print drum and the print hammers. This space accommodates varying thicknesses of paper forms, and is adjustable to allow free movement of the forms during skipping and spacing. With this dial, optimum print quality can be obtained for any thickness of paper.

Print Line Position

The Print Line Position dial permits the operator to make minor adjustments to the vertical location of the printed line. Such adjustment allows the operator to position the horizontal printed line in proper relationship to the imprinting of the form.

Form Tension

Setting the Form Tension dial increases or decreases the distance between the upper and lower tractors that hold the paper form in correct alignment. Proper vertical tension can be maintained on the forms by proper settings of this dial.

Density Selector

The Density Selector is a four-position control which allows the operator to vary the striking force of the print hammers. Position 1 provides minimum striking force; position 4 provides maximum striking force. The Carriage must be OUT to make this control effective.

DISPLAY PANEL

The Display Panel consists of four rows of indicator lights with eight lights in each row. The indicating lights are overlaid with a translucent mask bearing the labels which identify the feature or operation being displayed. Any one of 'nine masks can be positioned over the indicators by the operation of a manual control, providing 288 individual displays (4x8x9 = 288)

The Display Panel has been designed as an aid for machine operation, program debugging and system maintenance. Various connection panel features, abnormal tests, internal circuitry points, etc., are assigned to the Display positions to indicate the state of the assigned points.

The following list shows the labeling and a brief description of the indicated conditions for each of the nine masks. The first three masks are intended primarily for system maintenance. The last six masks will be particularly useful for program debugging as well as a system maintenance aid. Mask four is normally displayed when a program is in progress.

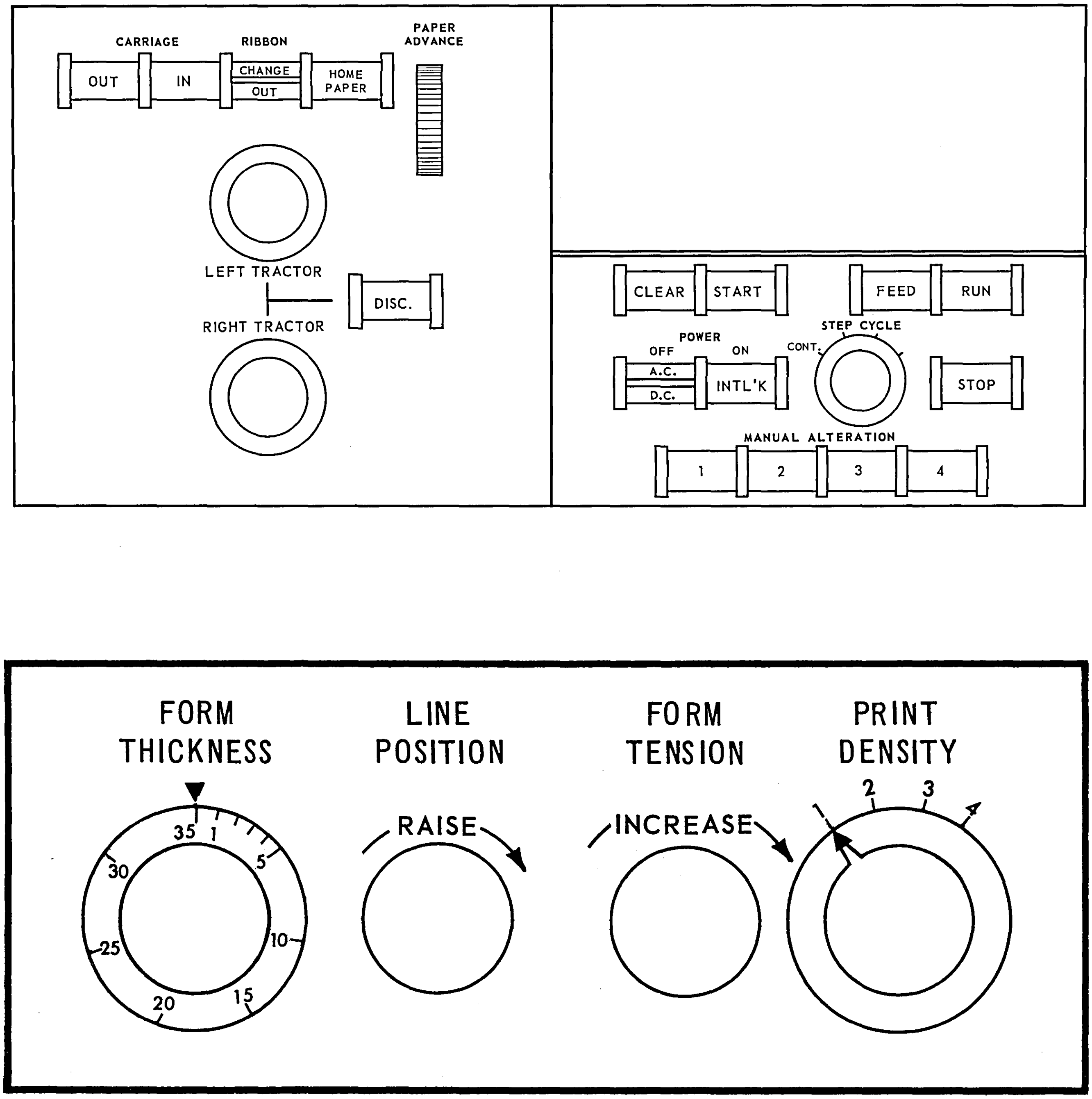


Figure 2-32, Central Control Panel and Printer Dials

MASK 1

MRL 1	MRL 2	MRL 3	MRL 4	MRL 5	MRL 6	MRL 7	MRL 8
MRB 1	MRB 2	MRB 3	MRB 4	MCB 1	MCB 2	MCB 3	MCB 4
MCL 1	MCL 2	MCL 3	MCL 4	MCL 5	MCL 6	MCL 7	MCL 8
CEBX/0	CEBY/1	CEB8/3	CEB4/5	CEB2/7	CEB1/9		

MRL 1 to 8	Memory Address Decode Row
MRB 1 to 4	Memory Address Decode Row
MCB 1 to 4	Memory Address Decode Column
MCL 1 to 8	Memory Address Decode Column
CEB X/0 to 1/9	Character Generator Bits

MASK 2

F 1 R 5	F 1 R 4	F 1 R 3	F 1 R 2	F 1 R 1	M R R 5	M R R 4	1 E C

F 1 C 5	F 1 C 4	F 1 C 3	F 1 C 2	F 1 C 1	M R R 3	M R R 2	M R R 1

F 2 R 5	F 2 R 4	F 2 R 3	F 2 R 2	F 2 R 1	M R C 5	M R C 4	2 E C

F 2 C 5	F 2 C 4	F 2 C 3	F 2 C 2	F 2 C 1	M R C 3	M R C 2	M R C 1

F1R 5 to 1	Field Count One, Row Bit
MRR 5 to 1	Memory Address Register Row Bit
1 EC	Field Count One Equal
F1C 5 to 1	Field Count One, Column Bit
F2R 5 to 1	Field Count Two, Row Bit
MRC 5 to 1	Memory Address Register Column Bit
2 EC	Field Count Two Equal
F2C 5 to 1	Field Count Two, Column Bit

MASK 3

BPX/0	BPY/1	BP8/3	BP4/5	BP2/7	BP1/9	SPFF	CMPS
TR 6	TR 5	TR 4	TR 3	TR 2	TR 1	ØBFF	ØSFF
AR 6	AR 5	AR 4	AR 3	AR 2	AR 1	EQFF	CTDS
	ACC1	AS 4	AS 3	AS 2	AS 1	RECP	NESN

BP x /0 to 1/9	Memory Data Register Bit
SPFF	Space Flip Flop
CMPS	Compress Flip Flop
TR 6 to 1	Transfer Register Bit
ØBFF	Zero Balance Flip Flop
ØSFF	Zero Suppress Flip Flop
AR 6 to 1	Adder Register Bit
EQFF	Equal Flip Flop
CTDS	Descending Flip Flop
ACC 1	Carry Flip Flop
AS 4 to 1	Sum Bit
RECP	Recomplement Flip Flop
NESN	Non Equal Signs

MASK 4

HOPPER	FEED	RD JAM	TSPJAM	STACKR	FORM	ADV √	PUNCH
HALT	IND1	IND2	IND3	IND4	RD I/0	PR I/0	PCHI/0
SP 1	SP 2	SK 1	SK 2	SK 4	END RD	END PR	R/P EX
MAINTA	MAINTB	MAINTC				PCHHLD	PCHCLR

Hopper	Last Card
Feed	Misfeed
Rd Jam	Reader Jam
TSP Jam	Card Transport Jam
STACKR	Full Stacker
FORM	Out of Paper
ADV √	Paper Runaway
PUNCH	Punch Abnormal
HALT	Halt Signalled
Ind 1-4	Programmed Indicators

RD I/0	Read Instruction
PR I/O	Print Instruction
PCH I/0	Punch Instruction
SP 1 and 2	Space 1 and 2
SK 1, 2 & 4	Skip 1, 2 & 4
END RD	End Read
End PR	End Print
R/P EX	Read +/or Print Execute
Maint A, B & C	Module A, B & C Test
PCHHLD	Punch Hold
PCH CIR	Punch Clear

MASK 5

STEP 1	STEP 2	STEP 3	STEP 4	STEP 5	STEP 6	STEP 7	STEP 8
STEP 9	STEP 10	STEP 11	STEP 12	STEP 13	STEP 14	STEP 15	STEP 16
STEP 17	STEP 18	STEP 19	STEP 20	STEP 21	STEP 22	STEP 23	STEP 24
STEP 25	STEP 26	STEP 27	STEP 28	STEP 29	STEP 30	STEP 31	

Step 1 to 31 Step Output

MASK 6

STEP32	STEP33	STEP34	STEP35	STEP36	STEP37	STEP38	STEP39
STEP40	STEP41	STEP42	STEP43	STEP44	STEP45	STEP46	STEP47
STEP48	STEP49	STEP50	STEP51	STEP52	STEP53	STEP54	STEP55
STEP56	STEP57	STEP58	STEP59	STEP60	STEP61	STEP62	

Step 32 to 62   Step Output

MASK 7

+ALG	-ALG	+ABS	-ABS	CPR	CPRA/N	NO REC	NO PRO
CLEAR	TRF D	ZD D	DØB D	ØSUP△D	ØSUP*D	INS D	SI D
TPCH	TRF A	ZD A	ZDS A	ØSUP△A	ØSUP*A	INS A	SI A
ISELD	TSELD	TØ YES	TØ NO	TS YES	TS NO	TO'F YES	TO'F NO

+ALG	Algebraic Add. Process	TPCH	Test Punch Descending
- ALG	Algebraic Subtract Process	TRF A	Transfer Ascending
+ABS	Absolute Add Process	ZD A	Zone Delete Ascending
- ABS	Absolute Subtract Process	ZDS A	Zone Delete with sign Ascending
CPR	Compare Process	Ø SUP △ A	Zero Suppress Space Ascending
CPR A/N	Alpha/Numeric Compare Process	Ø SUP * A	Zero Suppress Asterisk Ascending
No REC	No Recomplement	INS A	Insert Ascending
NO PRO	No Process	SI A	Superimpose Ascending
CLEAR	Clear	I SEL D	Inhibit Selector Delay
TRF D	Transfer Descending	T SEL D	Test Selector Delay
ZD D	Zone Delete Descending	T Ø YES	Test Zero Yes
DØB D	Delete Zero Balance Descending	T Ø NO	Test Zero No
Ø SUP △ D	Zero Suppress Space Descending	TS YES	Test Sentinel Yes
Ø SUP * D	Zero Suppress Asterisk Descending	TS NO	Test Sentinel No
INS D	Insert Descending	T O'F YES	Test Overflow Yes
SI D	Superimpose Descending	T O'F NO	Test Overflow No

MASK 8

PSP 1	PSP 2	PSP 3	PSP 4	PSP 5	PSP 6	PSP 7	PSP 8
PSP 9	PSP 10	PSP 11	PSP 12	PSP 13	PSP 14	PSP 15	PSP 16
PSP 17	PSP 18	PSP 19	PSP 20	CCB	CCA	CBB	CBA
CBA	CAA	ICA	ICB	ICC	ICD	OCA	OCB

PSP 1 to 20	Program Select Power
CCB	Instruction Cycle Control
CCA	Instruction Cycle Control
CBB	Instruction Cycle Control
CBA	Instruction Cycle Control
CAB	Instruction Cycle Control
CAA	Instruction Cycle Control
ICA	Instruction Cycle Control
ICB	Instruction Cycle Control
ICC	Instruction Cycle Control
ICD	Instruction Cycle Control
OCA	Instruction Cycle Control
OCB	Instruction Cycle Control

MASK 9

C 1 > –	C 1 ≠ < +	C 1 = Ø	C 2 > –	C 2 ≠ < +	C 2 = Ø	C 3 > –	C 3 ≠ < +

C 3 = Ø	C 4 > –	C 4 ≠ < +	C 4 = Ø	C 5 > –	C 5 ≠ < +	C 5 = Ø	C 6 > –

C 6 ≠ < +	C 6 = Ø	C 7 > –	C 7 ≠ < +	C 7 = Ø	C 8 > –	C 8 ≠ < +	C 8 = Ø

C 9 > –	C 9 ≠ < +	C 9 = Ø	C 10 > –	C 10 ≠ < +	C 10 = Ø		

C1 (to 10) > -	Comparator 1 (to 10) Greater or Minus.
C1 (to 10) < +≠	Comparator 1 (to 10) Less than, plus, or non match.
C1 (to 10) Ø =	Comparator 1 (to 10) Zero or match.

VII OPTIONAL EQUIPMENT

CARD PUNCH

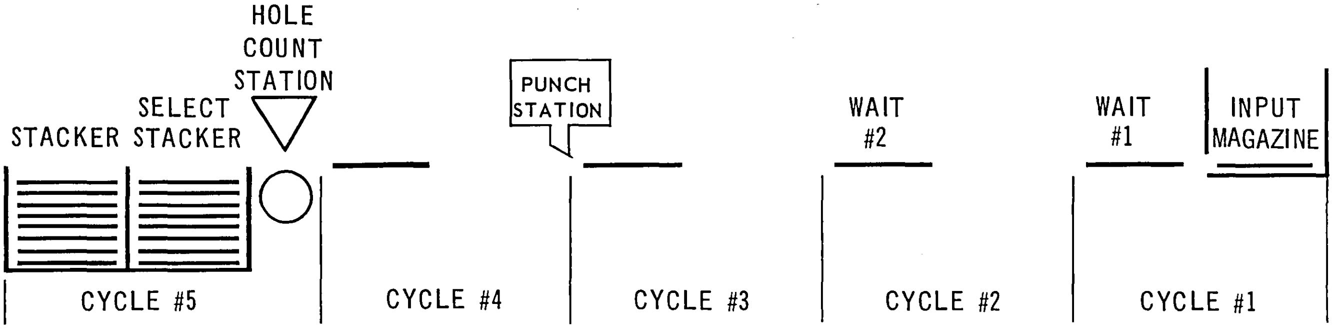
A card punch is available as an optional output unit. The punch is directly connected to the 1004 Processor through an electrical cable. As previously explained, a section of storage is set aside to be used as punch storage. Data to be punched is transferred to this section of storage prior to punching.

The speed of the card punch is 200 cards per minute regardless of the amount of information to be punched into a card. Data to be punched is set up and punched into the card one row at a time. Cards are placed in the input magazine face down with the 9 edge leading into the machine. The capacity of the input magazine is 1000 cards. Five machine cycles are required to transport a card through the punch. These cycles are:

1. From input magazine to the first unit station.
2. From the first wait station to the second unit station.
3. From the second wait station to the punch station.
4. The card is punched and goes to a hole count checking station.
5. Hole count check is made and the card is delivered to one of two card stackers.

An automatic hole count check is made on all cards punched. The number of punches in a card is compared to the number of bits transferred from storage. If there is a discrepancy in the comparison, the card is delivered to the select stacker. If the comparison matches, the card is delivered to the normal stacker. The capacity of each of the card stackers is 1000 cards. Under operator control, the processor can be halted when a hole count check error is found.

CARD TRAVEL



In the event of a card jam, if a punch operation is initiated without a card in the punching station, the punch instruction is retained. The processor halts on the current step, and clearing the jam reinitiates the punch instruction. Depressing the Run button allows the program to proceed.

SHORT CARD FEEDING DEVICE

A short card feeding device is available to allow the read section to accept stub cards. These stub cards contain 51 columns. The device consists of two inserts for the input magazine and a filler for the card stacker. The input magazine inserts are easily positioned with a thumb screw at each side of the magazine. The filler for the card stacker reduces the depth of the stacker to receive the stub cards. Card feeding, reading, and ejecting proceeds in the normal manner with the use of stub cards. The device can be easily removed to allow feeding of standard-size cards.

The use of the 51-column stub card allows for a maximum operating speed in the reader of 400 cards per minute with ample time for a variety of programming operations.

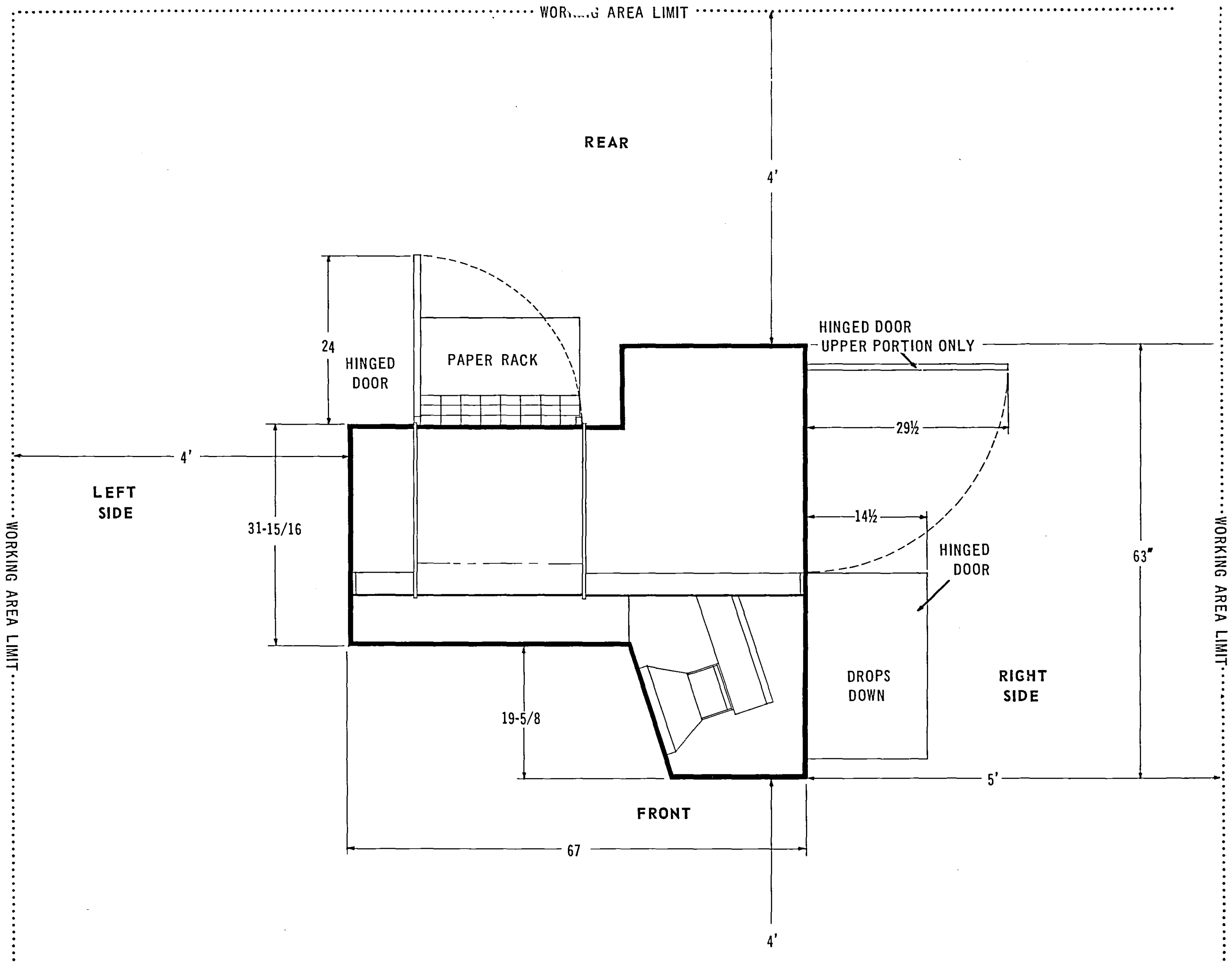
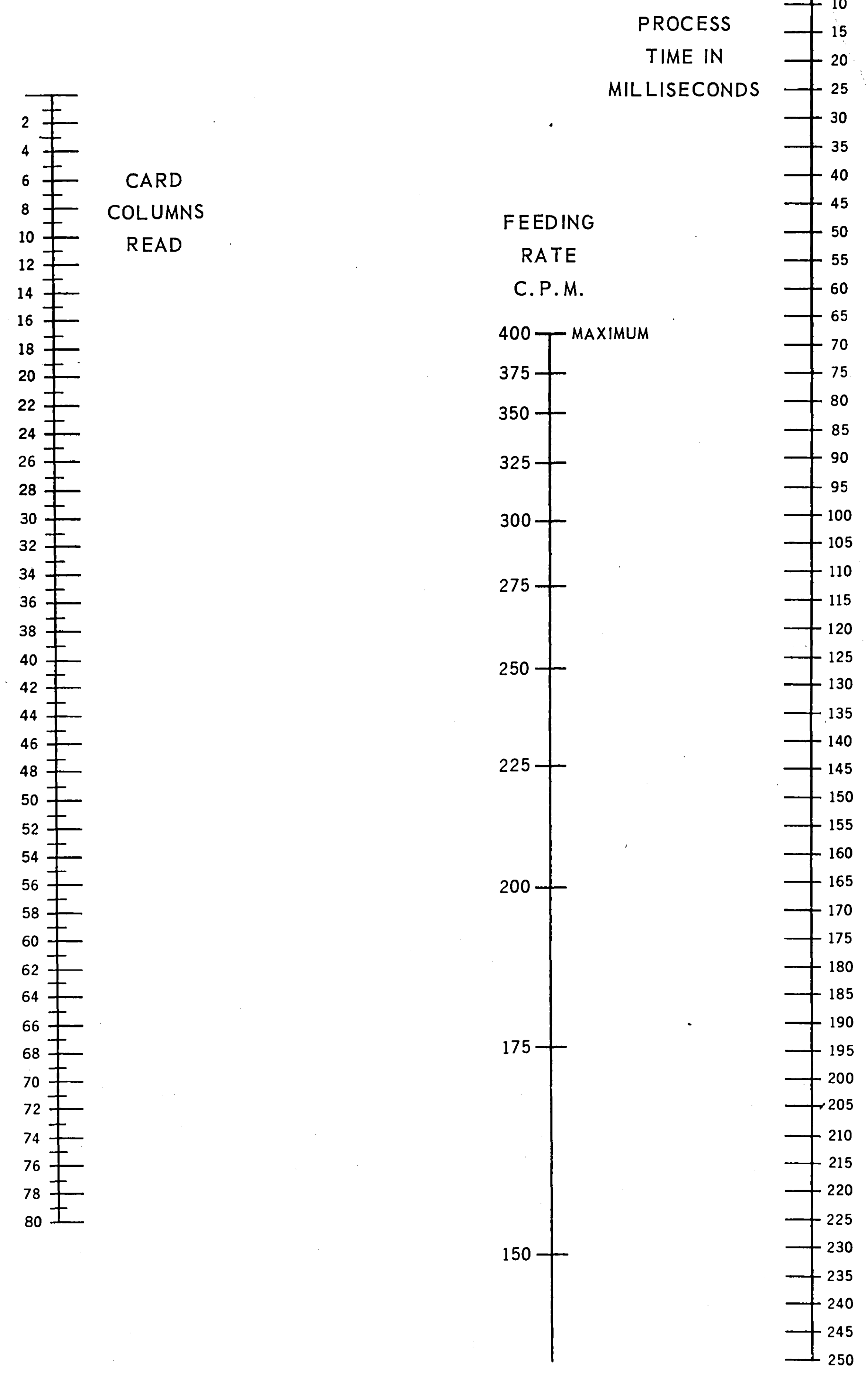


Figure 2-33. This diagram illustrates the required floor measurements for installation of the 1004 processor. The additional area required for work and maintenance purposes is outlined by the dotted lines.

READ SECTION TIMING CHART

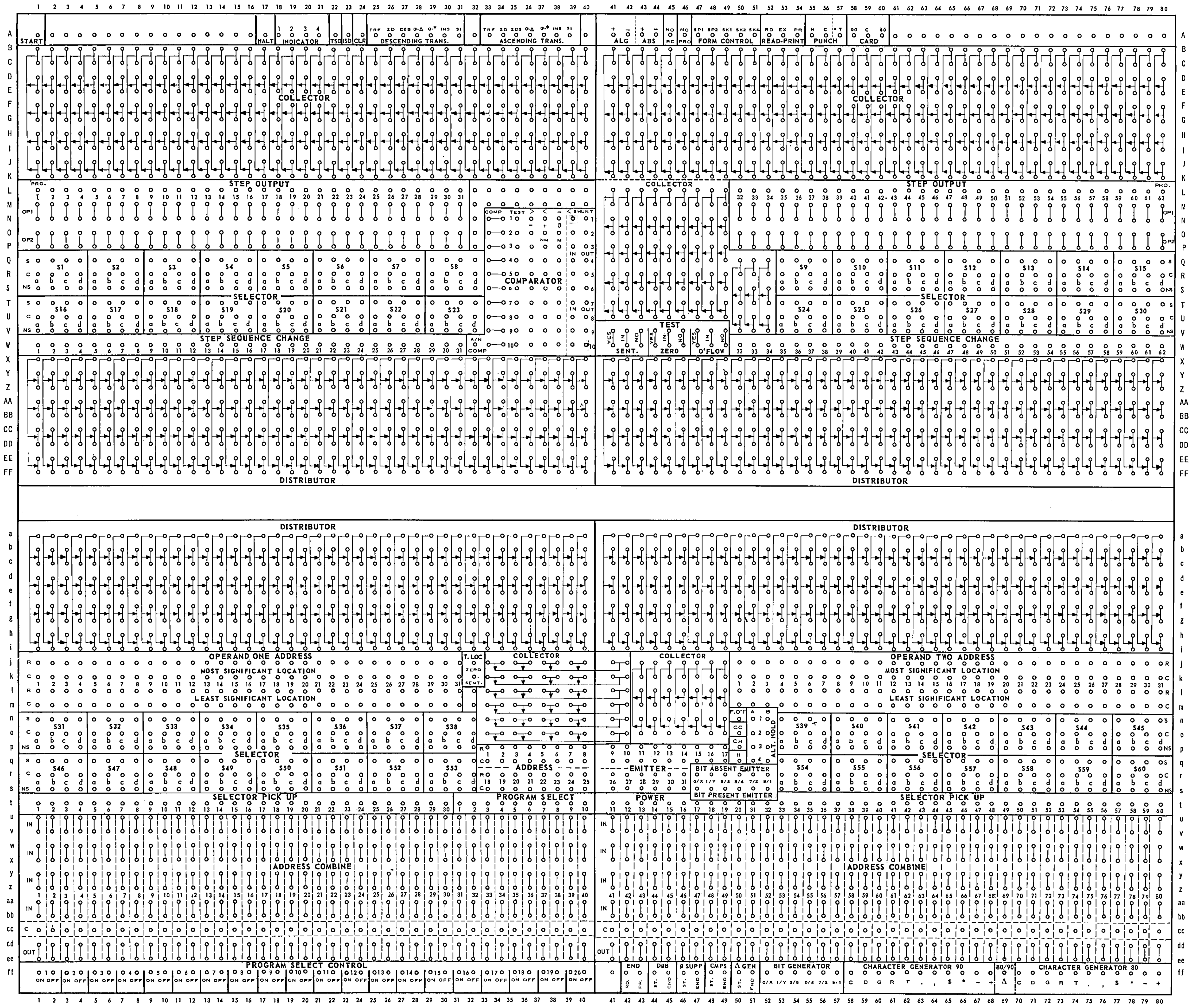
80-COLUMN



INSTRUCTIONS FOR TIMING CHART

When 2 factors are known, the third can be found by aligning a straight-edge through the 2 known points. For example, when reading 40 columns at 400 cards per minute, about 35 milliseconds of processing time are available.

UNIVAC 1004 CARD PROCESSOR



INDEX

UNIVAC

DIVISION OF SPERRY RAND CORPORATION