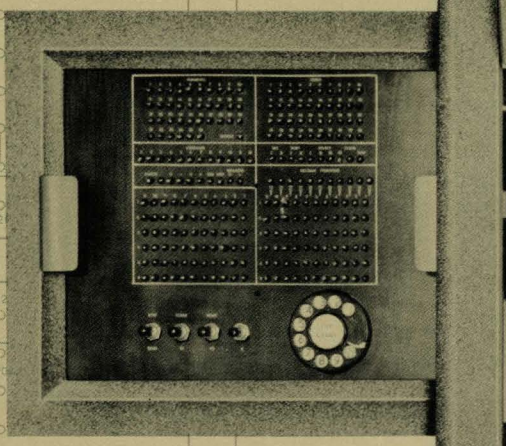
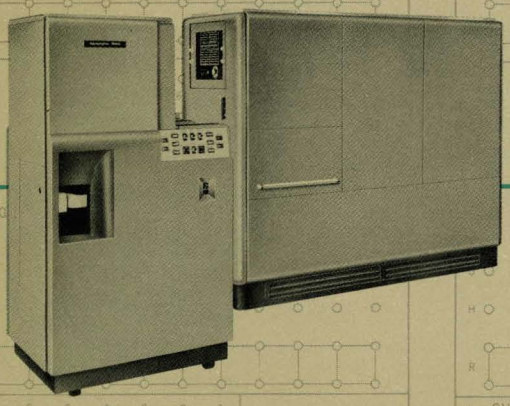
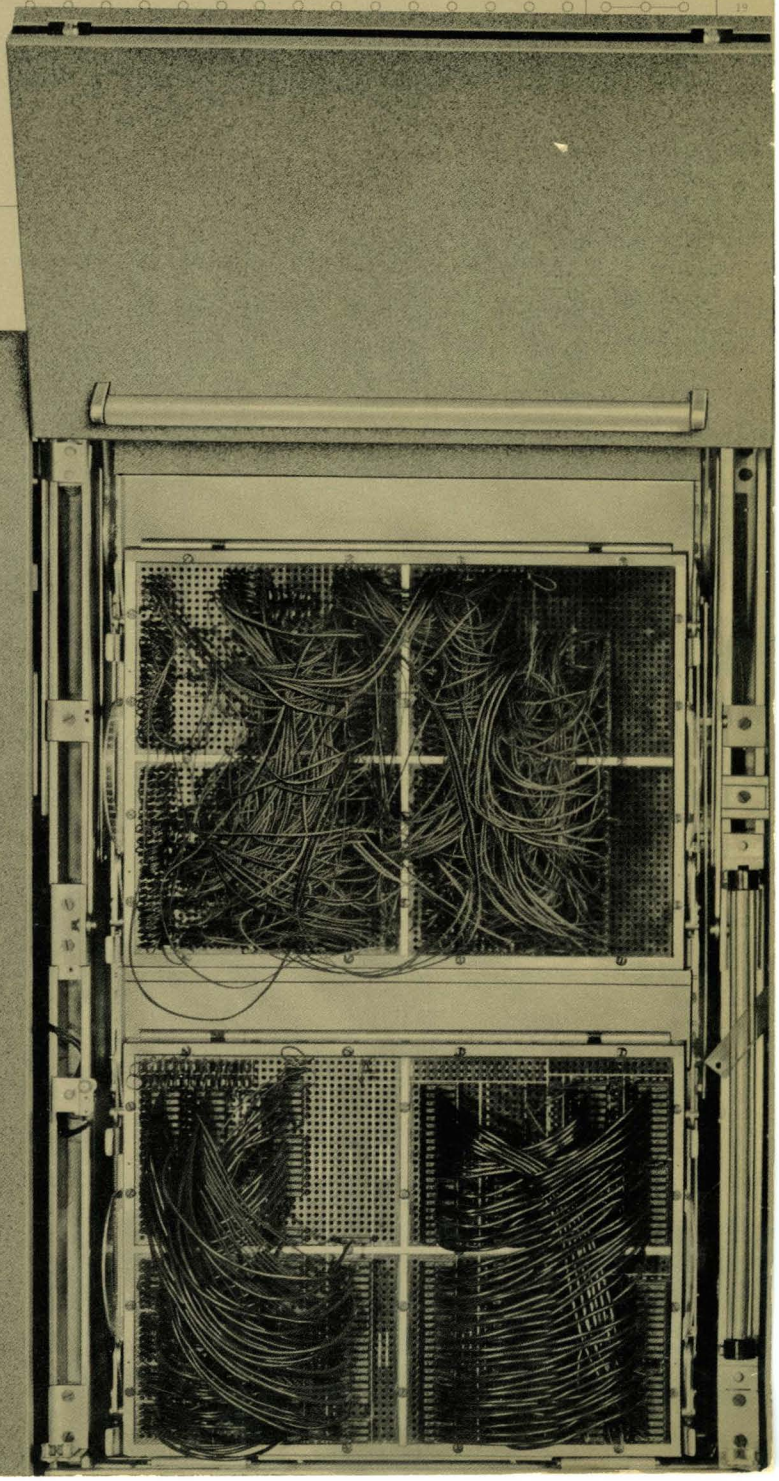
programming

UNIVAC® 60
and
UNIVAC® 120
punched-card
electronic computers







programming

UNIVAC® 60
and
UNIVAC® 120
punched-card electronic computers

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

TM 1038 PRINTED IN U.S.A.

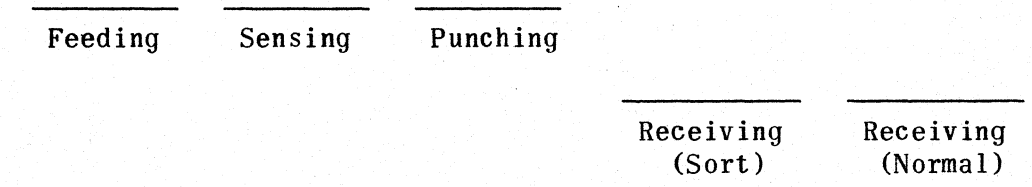
Index

Section I Introduction

The Univac 60 and 120 are electronic computers designed to operate with 90-column punched cards. The two computers are identical, except that the Univac 120 has more input and storage capacity than the Univac 60.

The computer consists of two units - the card sensing-punching unit, and the electronic computing unit. The two units are joined together by electrical cables. Two removable connection panels govern the routine by which the punched cards will be processed. The input-output panel indicates the columns to be sensed and punched; the constant-program panel contains the program which is to be followed - that is, the series of steps through which the computer must go to obtain the desired results. The two connection panels are both inserted in the electronic computing unit; they may be easily removed and others containing a different program inserted, or a panel may be rewired by the operator to handle another program.

The card sensing-punching unit is the portion of the computer through which the cards are fed. The sections of the sensing-punching unit through which a card passes may be diagrammed as follows:



The card is stationary both while values are being sensed in the sensing station and punched in the punching station. The full ninety columns in the card may be sensed simultaneously and used in calculation without being entered in the storage units of the computer. Information punched into the card may be either results calculated by the computer or information reproduced from the same or a preceding card.

The card sensing-punching unit operates at a maximum speed of 150 cards a minute. If, however, due to a particularly long program or one that repeats steps many times, the computer requires more time than this to finish the calculation of a card, the sensing-punching unit will automatically wait until the calculation has been completed before punching the card.

The sensing-punching unit contains a control panel with six switches and eight lights. The switches are used for such operations as turning the computer on, starting and stopping it. The lights are indications to the operator as to the functioning of the computer.

The electronic computing unit handles the computations necessary to complete a program. A program is the series of arithmetic steps through which the computer solves a problem. The maximum number of program steps available on the computer is 40, although this number may be effectively increased by various programming techniques. The computer is basically a "three address" computer - during one step two values are called upon, and the result of the operation upon them is placed in storage. The basic form of a program step, together with the symbols which will be used throughout the manual, is:

Step Number	Value 1	Process	Value 2	=	Result	Branching
	V1	Pr	V2	=	R	-Br, +Br

It is the programmer's responsibility to reduce the problem to a series of steps in this form.

There are four processes available, any one of which may be used on any of the 40 program steps. They are the arithmetic functions of addition, subtraction, multiplication, and division. There are also certain operational functions used in programming, such as start, but these may not be used as the process of a step.

The values in a program step may be of three types: a card-read field, a constant value, or the result of a previous step. Since the sensed set-up is locked in sensing switches during the entire program for that card, any field of the card which has been wired may be called upon at any time. Constant values are wired on the constant-program panel, and are also available at any time. Both card-read fields and constant values are referred to as elements. The maximum number of elements available on the computer is 36. Each element may have a maximum of 10 digits plus sign. The result of a previous step may be used as a value in any following step until the result is cleared, either by the entry of new information into the storage unit or by a clear instruction from the computer.

The result of a program step is placed in a storage unit. There are six storages on the Univac 60, and 12 storages on the Univac 120. Each storage contains 10 columns plus sign. When a result is placed in a storage, it will remain there until cleared as mentioned in the preceding paragraph. The result may be used as necessary in succeeding steps either during the program for the card which is sensed or that of any following cards. It may also be punched into the card which is sensed, or any following cards.

Program steps on the computer are completely non-sequential. It is possible to wire from one program step to any of the other steps, or to any operational function. For example, step 5 could be wired to step 6, to step 39, to step 2, or to clear. This wiring is done on a basis

of whether the result of the step which has just been completed is plus or minus - that is, if the result of step 5 is plus the program could continue with step 6, while if it is minus it could continue with step 2. This is known as the plus or minus branching of a step.

The computation of every program step on the computer is automatically checked before the computer continues to the next step. If it does not check, the computer will repeat the step or the program, until the result does check, or until the cause of the failure is corrected.

Although the computer is basically a numeric computer, it is possible to reproduce both alphabetic and numeric information. The reproducing may be in the same card columns as the original information, or may be to any other columns of the card. It may be done into the same card that is sensed, or any following card or cards.

The computer may be used to summarize information which has been calculated and/or accumulated from the preceding cards. Designating information may be punched into the summary cards from storage units, and reproduced into the summary cards from preceding cards.

The following sections of the manual deal with the various features of the computer, programming techniques, operating instructions, and sample problems.

Section II Machine Functions

APPLICATION:— _____

_____

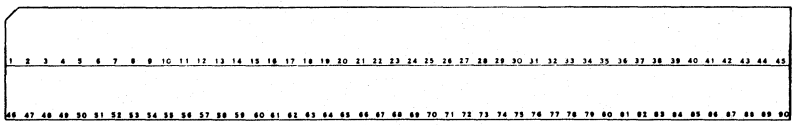
_____

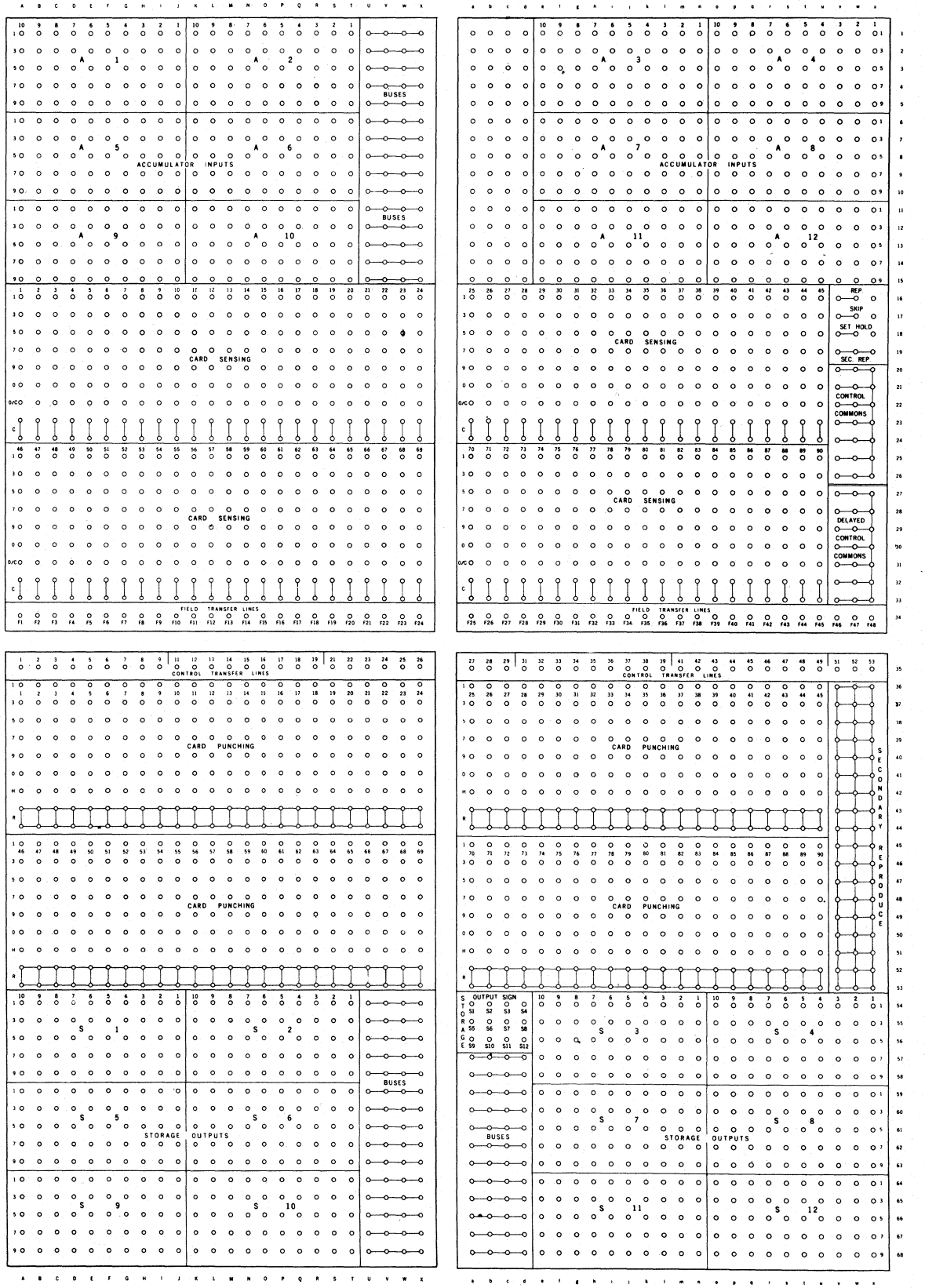
Remington Rand
DIVISION OF SPERRY RAND CORPORATION
TABULATING MACHINES

UNIVAC 60 & 120

PUNCHED-CARD ELECTRONIC COMPUTERS

INPUT & OUTPUT PANEL





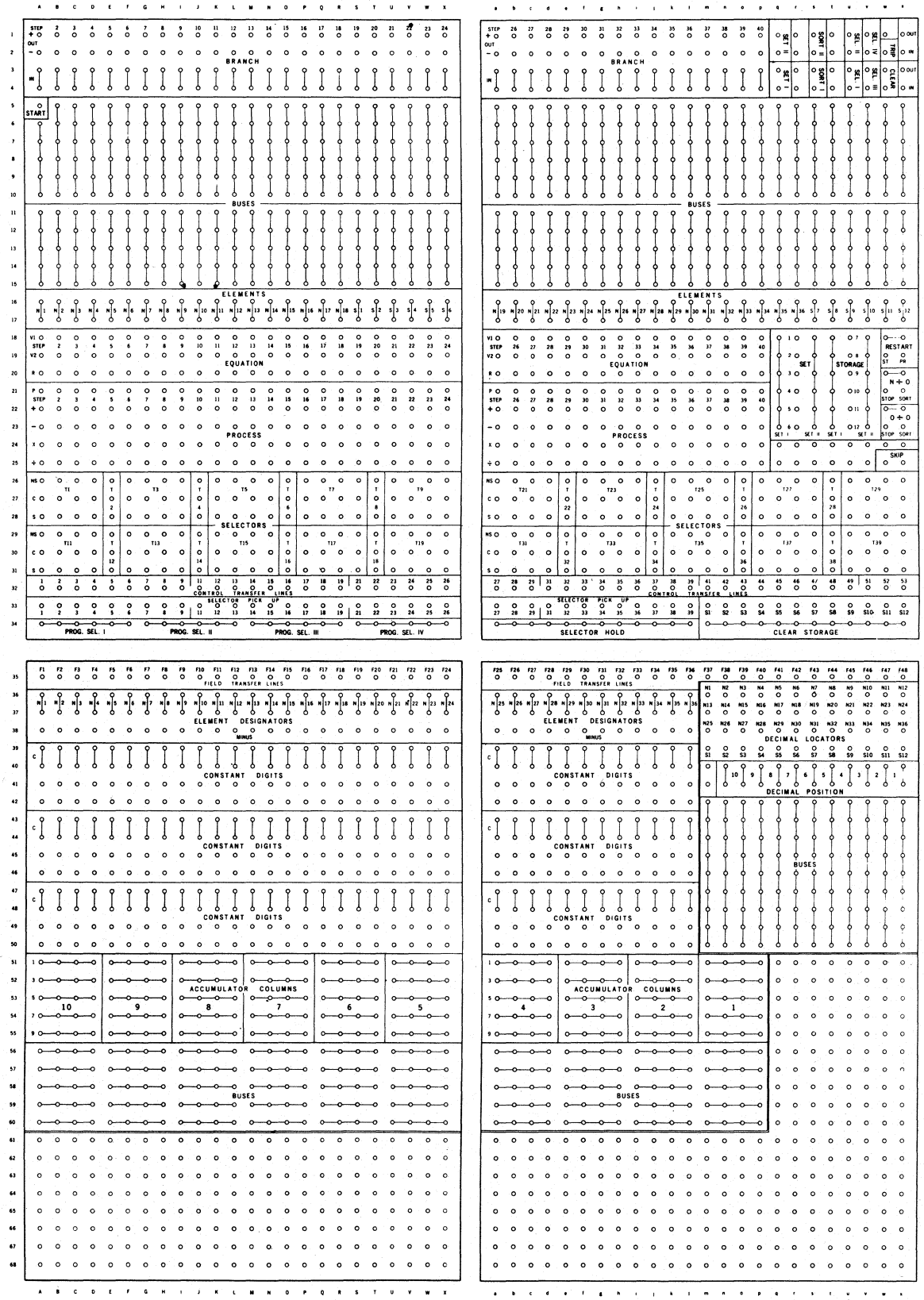
APPLICATION:— _____

Remington Rand
DIVISION OF SPERRY RAND CORPORATION
TABULATING MACHINES

UNIVAC 60 & 120

PUNCHED-CARD ELECTRONIC COMPUTERS

CONSTANT & PROGRAM PANEL



PROGRAM CHART NO.

APPLICATION:

UNIVAC 60 & 120

PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand

TABULATING MACHINES

TABLE OF FACTORS

SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE
N1				N13				N25			
N2				N14				N26			
N3				N15				N27			
N4				N16				N28			
N5				N17				N29			
N6				N18				N30			
N7				N19				N31			
N8				N20				N32			
N9				N21				N33			
N10				N22				N34			
N11				N23				N35			
N12				N24				N36			

CARD DESCRIPTION

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
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																									
32																									
33																									
34																									
35																									
36																									
37																									
38																									
39																									
40																									

SYM	CARD FIELD TITLE	REG. COL	REG. POS	OUTPUT PUNCHING 10	9	8	7	6	5	4	3	2	1	DEC LOC	SET 1	SET 2	CLEAR	SYM
S1																		S1
S2																		S2
S3																		S3
S4																		S4
S5																		S5
S6																		S6
S7																		S7
S8																		S8
S9																		S9
S10																		S10
S11																		S11
S12																		S12

FROM	TO
START	
SET 1	
SET 2	
SORT 1	
SORT 2	
CLEAR	
P.S. I	
P.S. II	
P.S. III	
P.S. IV	

REPRODUCE CONTROLS	COL.	POS.
REP		
SKIP		
SET HOLD		
SEC REP		

	0 ÷ 0	N ÷ 0
STOP		
SORT		

RESTART	
STEP	
PROGRAM	

SELECTOR CHART NO.:

APPLICATION:

UNIVAC 60 & 120
PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand
TABULATING MACHINES

ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD-F1, F2, ETC: CONTROL - C1, C2, ETC.) DECIMAL LOCATION, NEGATIVE CONTROL AND TRANSFER LINE, INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM DESIG	TO	NEG. CONT.			DEC LOC	ACCUMULATOR COLUMNS									
		COL	POS	TRF LINE		10	9	8	7	6	5	4	3	2	1
N1															
N2															
N3															
N4															
N5															
N6															
N7															
N8															
N9															
N10															
N11															
N12															
N13															
N14															
N15															
N16															
N17															
N18															

ELEM. DESIG	TO	NEG. CONT.			DEC LOC	ACCUMULATED COLUMNS									
		COL	POS	TRF LINE		10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															
N22															
N23															
N24															
N25															
N26															
N27															
N28															
N29															
N30															
N31															
N32															
N33															
N34															
N35															
N36															

ACCUMULATOR INPUTS

INDICATE: TOP LINE - TRANSFER LINE (F1, F2, ETC.)
BOTTOM LINE - CARD COLUMNS

SYM.	ACCUMULATED COLUMNS									
	10	9	8	7	6	5	4	3	2	1
A1										
A2										
A3										
A4										
A5										
A6										
A7										
A8										
A9										
A10										
A11										
A12										

REPRODUCE

FROM COLUMN	
TO COLUMN	
CONTROL	

SELECTOR CONTROL CHART

SYMBOLS: SELECTOR . . . T1-1, T1-2, T1-3, T1-4, T2, ETC.
COMMON . . . COM
NON SELECT . . N.S.
SELECT . . . SEL.
BRANCH . . . BR.
PROCESS . . PR.
VALUE 1 . . . V1
VALUE 2 . . . V2
RESULT . . . RES.
ELEMENTS . . . N1, N2, N3, ETC.
STORAGE . . . S1, S2, S3, ETC.
PROGRAM SELECTOR . . . P.S.1, P.S.2, ETC.

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
11-1					
11-2					
11-3					
11-4					
12					
13-1					
13-2					
13-3					
13-4					
14					
15-1					
15-2					
15-3					
15-4					
16					
17-1					
17-2					
17-3					
17-4					
18					
19-1					
19-2					
19-3					
19-4					
31-1					
31-2					
31-3					
31-4					
32					
33-1					
33-2					
33-3					
33-4					
34					
35-1					
35-2					
35-3					
35-4					
36					
37-1					
37-2					
37-3					
37-4					
38					
39-1					
39-2					
39-3					
39-4					

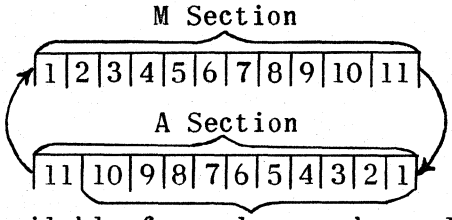
SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1					
1-2					
1-3					
1-4					
2					
3-1					
3-2					
3-3					
3-4					
4					
5-1					
5-2					
5-3					
5-4					
6					
7-1					
7-2					
7-3					
7-4					
8					
9-1					
9-2					
9-3					
9-4					
21-1					
21-2					
21-3					
21-4					
22					
23-1					
23-2					
23-3					
23-4					
24					
25-1					
25-2					
25-3					
25-4					
26					
27-1					
27-2					
27-3					
27-4					
28					
29-1					
29-2					
29-3					
29-4					

S1-1370 RRBH 12456

1. Accumulator

A. DESCRIPTION OF THE ACCUMULATOR

Before discussing the features of the computer from a programming viewpoint it is well to have an understanding of the accumulator. The accumulator is a group of electronic tubes in which all computation occurs. It may be thought of as having 22 columns, as follows:



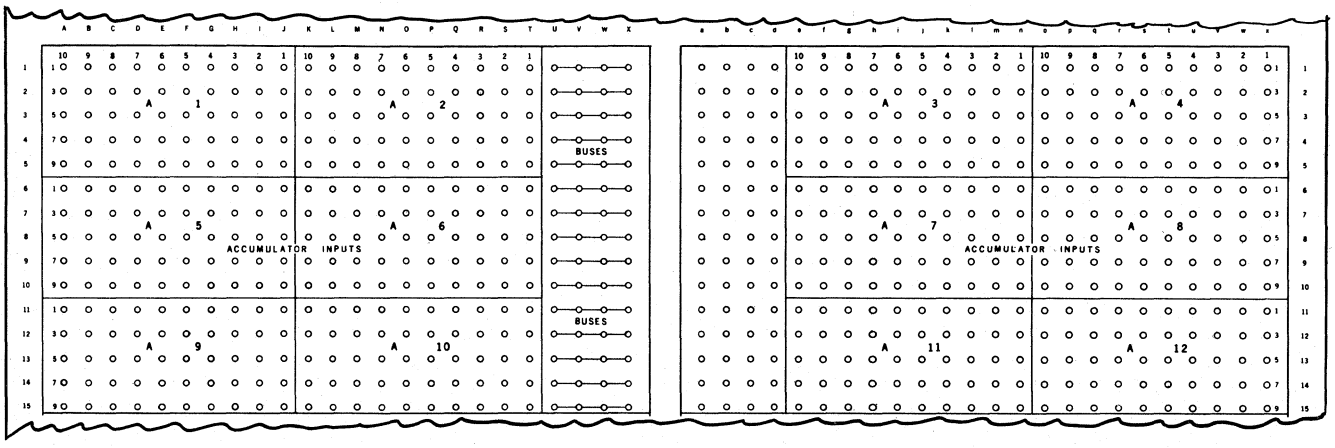
Each column is made up of the necessary electronic tubes to create any numeric digit from 0 to 9.

Although all 22 columns may be used during calculation, value 1 and value 2 of a program step must be placed in the accumulator in columns 10-1 of the A section. When the result of a step is placed in storage, no significant digit may be located in column 11 of the A section. If this happens, the computer will "hang up", stopping on the step in which this occurs. However, it is possible to drop off any number of digits to the right of column 1 of the A section without hanging up the computer. These digits will be placed in the M section, beginning in the 11th column. The use of the M section and column 11 of the A section during computation is automatic with the computer, and need not concern the programmer. It is his responsibility, however, to be certain that no result being placed in storage could have a significant digit in column 11 of the A section.

Paragraphs B and C below explain how card-read fields and constants enter columns 10-1 of the accumulator. Paragraph D explains the relation of storage to the accumulator.

B. ENTERING CARD-READ FIELDS INTO THE ACCUMULATOR

On the input-output connection panel, form S1-1351, are 12 accumulator inputs, lines 1-15, A-T, e-x. They are numbered A1-A12.



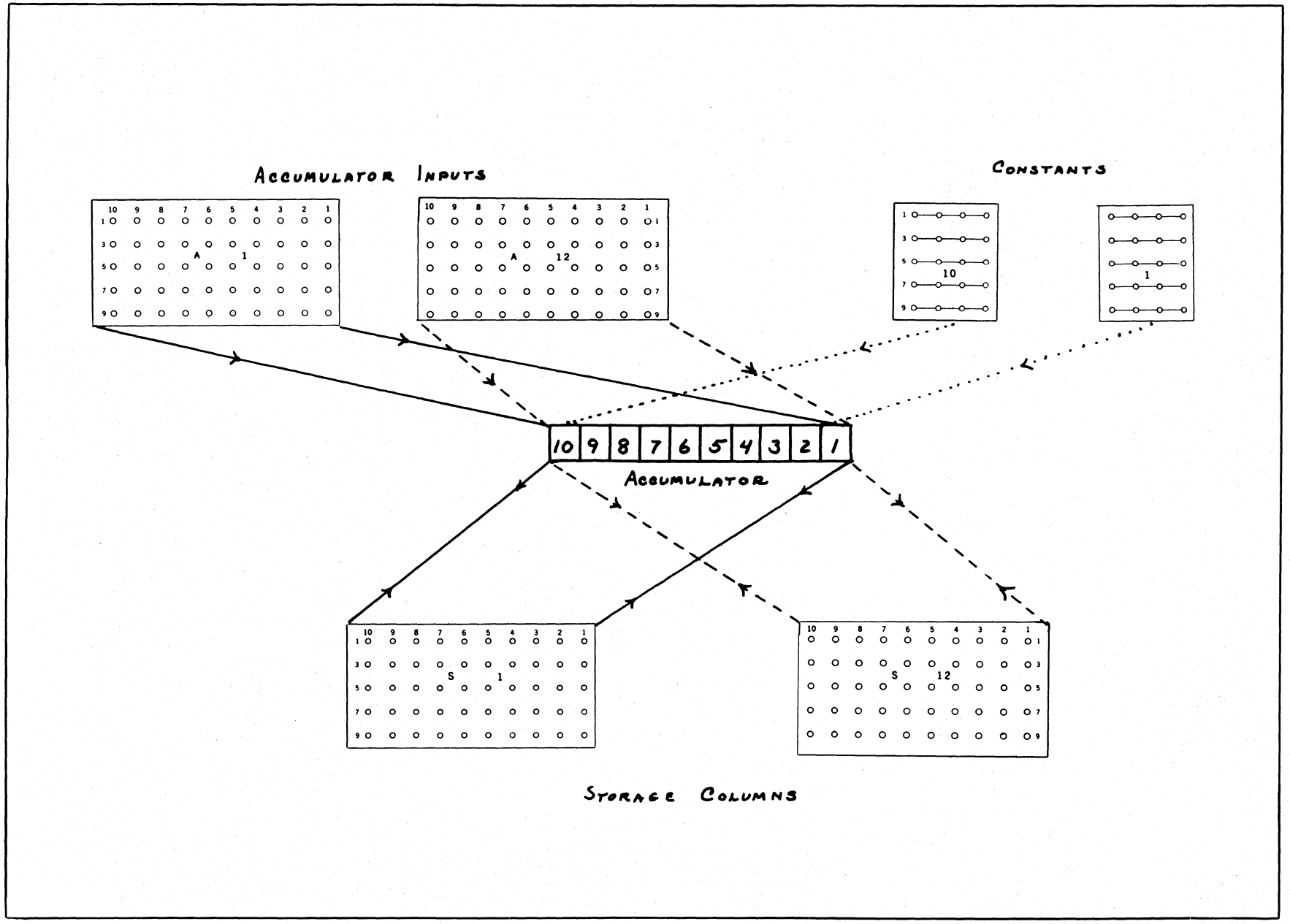
Within each accumulator input are 10 columns of 5 positions each. The column numbers 10-1 correspond to the 10 available columns of the accumulator, while the 5 positions refer to numeric values entering the accumulator columns, coded in the 90-column punching code. Accumulator inputs are the means of wiring card-read fields into the accumulator so that they may be used as values in program steps.

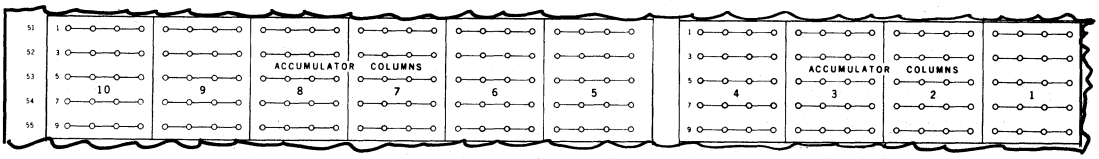
Column 1 of all 12 accumulator inputs is wired to column 1 of the accumulator; column 2 of all 12 is wired to column 2 of the accumulator and so forth. In actuality, the five positions in an accumulator input column are connected to the accumulator column through a decoding section, which changes the punched code into a code compatible with the accumulator. The accumulator inputs are all diode protected, which means that there is no danger of a backfeed - for example, calling on all card columns wired into accumulator column 1 when only that wired through column 1 of A2 is desired. The diagram on page 16 indicates the internal wiring of all 12 accumulator inputs to the accumulator.

All card fields to be used as values in a program must be wired to the accumulator through accumulator inputs. For a complete explanation of this wiring, see page 27.

C. ENTERING CONSTANTS INTO THE ACCUMULATOR

On the constant-program panel, form S1-1352, lines 51-55, A-p are the hubs through which constant values are wired into the accumulator. The horizontal numbers 10-1, between lines 53 and 54, refer to the 10 available columns of the accumulator. The vertical numbers 1, 3, 5, 7, 9 are the actual values to be entered into a particular accumulator column. They are wired in the 90-column punching code - that is, a 2 would be created by wiring a 1 and a 9.


ACCUMULATOR INPUTS
CONSTANTS
A 1
A 12
10
1
10 9 8 7 6 5 4 3 2 1
ACCUMULATOR
S 1
S 12
STORAGE COLUMNS



Since it is possible that a particular digital value, "5" for example, might be used in the same accumulator column as part of more than one constant value, there are four hubs available for each position within each accumulator column. All twenty hubs within an accumulator column are wired to the corresponding accumulator column - that is, all twenty hubs in accumulator column 1 are wired to accumulator column 1. A backfeed does not occur because the constant digits from which the accumulator columns are wired are neon-protected. The diagram on page 16 indicates the internal wiring of the constant accumulator columns to accumulator columns.

A complete explanation of constants is found on page 31.

D. STORAGE AS RELATED TO THE ACCUMULATOR

Storage as related to the accumulator differs from card-read fields and constant values. A number is not only called upon to enter the accumulator from storage to be used as value 1 or value 2 of a program step, but the results of program steps are taken from the accumulator and placed in storage.

Each storage unit has 10 columns, numbered 10-1. As indicated in the diagram on page 16, column 1 of each storage unit is internally wired to column 1 of the accumulator, column 2 of each storage unit is wired to column 2 of the accumulator, and so forth.

As explained below, the programmer has control over the accumulator columns in which a value from storage will be placed through assignment of a decimal location to each storage. No wiring of storage columns to the accumulator is ever done by the programmer - this is all internal wiring. The only wiring of storage columns with which the programmer is concerned is that of storage outputs to punching columns in the card. This is covered on page 39.

E. DECIMAL LOCATIONS

Every card-read field, constant value, and storage used during a program must be assigned a decimal location. The decimal location is wired on the constant-program connection panel and is fixed for a particular program,

except as it may be altered through use of selectors. If the decimal location is not wired, the computer will hang up on the first step in which the element or storage is called upon.

Decimal locations are assigned with reference to the accumulator. For example, assume that a five-digit price, punched in card columns 30-34, is to be entered into accumulator columns 5-1 of accumulator input 1. The decimal point in the card falls between columns 32 and 33. It would be placed as follows:

	10	9	8	7	6	5	4	3	2	1
Accumulator columns (A1)										
Card columns						30	31	32▲	33	34

In this case the decimal point falls between accumulator columns 3 and 2. Price therefore has a 3/2 decimal location. A complete explanation of decimal locations as related to elements and storage is found on pages 29 and 38.

F. OPERATION OF THE ACCUMULATOR

As an example of how the accumulator operates, assume that it is to add 5.837 (V1) and 28.906 (V2). The result is to have only 2 places following the decimal. V1 and V2 are in accumulator inputs as follows:

Accumulator columns:	10	9	8	7	6	5	4	3	2	1	Decimal location
V1:				5▲	8	3	7				7/6
V2:						2	8▲	9	0	6	4/3

Before V1 is called upon, the accumulator and the result storage are cleared. V1 then is entered into the accumulator according to its decimal location.

10	9	8	7	6	5	4	3	2	1
			5▲	8	3	7			

The V2 decimal location is entered, and V1 is shifted so that the decimals of V1 and V2 are aligned.

10	9	8	7	6	5	4	3	2	1
						5▲	8	3	7

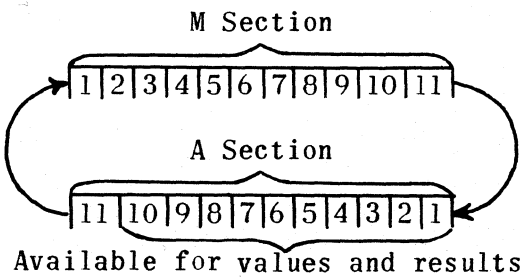
V2 is then entered, and added to V1.

10	9	8	7	6	5	4	3	2	1
					3	4▲	7	4	3

The decimal of the result storage is entered, and the result is shifted so that the two decimals are aligned.

10	9	8	7	6	5	4	3	2	1
						3	4▲	7	4

The result is placed in the result storage and the step is checked.



Referring to the above diagram notice that there is a connection between column 11 of the A section and column 1 of the M section. There is also a connection between column 11 of the M section and column 1 of the A section. As a value is shifted to align with a decimal location, it always moves to the left. If both V1 and V2 have the same decimal location, value 1 shifts entirely around the 22 columns of the accumulator to align itself with the decimal location of value 2.

Assume that V1 has a 3/2 decimal location and V2 has a 1/0 decimal location. When V1 aligns with the decimal location of V2, it shifts 20 places to the left so that its decimal location aligns with the 1/0 decimal location of V2. The last two digits of V1 are therefore in columns 11 and 10 of the M section. If the result is to be placed in a 3/2 storage, the result is shifted two places to the left. The digits which were in columns 11 and 10 of the M section will move to columns 2 and 1 of the A section, and therefore be available. If the result is placed in a 2/1 storage, the result shifts one place to the left, bringing one digit into column 1 of the A section and leaving one digit in column 11 of the M section. This last digit is not placed in storage, and is thereby dropped off.

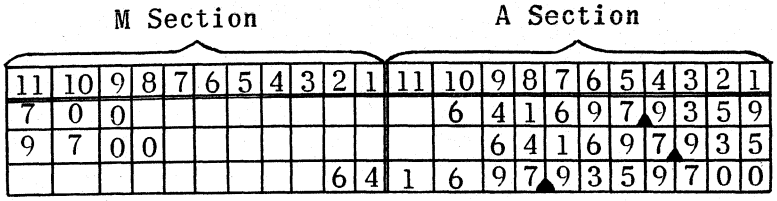
If a decimal location is not assigned to an element or a storage unit the computer hangs up, since it is searching for a decimal position with which to align and cannot find one.

Faster operation is obtained from the computer if the decimal location of V2 is different from the decimal location of V1 and the result.

G. DROPPING DIGITS

As mentioned on page 14, a result may never have a significant digit in column 11 of the A section, or the computer will hang up. However, it is possible to compute an answer of more than 10 digits. For example, when multiplying a 7-digit value times a 6-digit value, a 13-digit answer may be the result. The result of multiplying 843.0925 times 761.124 is 641697.9359700. Some of possibilities for handling this result would be:

Accumulator Columns



Storage decimal location of result
5/4
4/3
8/7 - computer hangs up

As can be seen, by dropping off digits to the right when placing the result in storage, the answer can be adjusted so that the computer will not hang up. The digits which are dropped off are actually computed, but are lost when the result is stored.

H. SUMMARY

1) The accumulator has 22 columns, of which the last 10 are available to the programmer.

2) Accumulator inputs for card-read fields, accumulator columns for constant digits, and storages are all internally wired to the last 10 columns of the accumulator.

3) A decimal location must be assigned to every card-read field, constant value, and storage used during a program. This is done with reference to the accumulator columns. If it is not done, the computer will hang up.

4) A result of more than 10 digits may be computed, though only 10 may be stored. Excess digits above the limit of 10 may be dropped off to the right, but not to the left. The computer will hang up if a result which has a significant digit in column 11 of the A section is placed in storage.

2. Elements

A. GENERAL

A-1 Description of an element

An element is either a card-read field or a constant which is used as a value in a program step. The maximum number of elements available on the 60 or 120 is 36. For convenience, they are referred to in programming as N1, N2, ... N36. Each element may contain a maximum of 10 digits plus sign, due to the fact that only 10 digits of the accumulator are available to the programmer. It may, however, contain fewer digits - even no digits, as in the case of zero, which is usually required as a constant value.

A-2 Decimal location

As mentioned in the section on the accumulator, every element which is used must have a decimal location, even though the element is a whole number. The constant value of zero must also have a decimal location. Without one, the machine will hang up on the first step in which the element is called upon.

A-3 Sign

Any element with the exception of zero may be always plus, always minus, or either plus or minus dependent upon a control hole punched in the card or a determination made during the program. Zero is always plus. An element will always be considered plus by the computer unless it is wired to be minus. If an element is always to be minus, this may be wired on the constant-program panel with no need for control wiring from the card.

A-4 Use as a value

An element may be either V1 or V2 of a program step. In fact, the same element may be both V1 and V2 if desired. Since the sensed information is locked in the sensing switches throughout a program, a card-read element may be called upon as often as necessary during a program. In effect, therefore, there are 90 columns of input storage. Constant elements, which are wired on the constant-program panel, may also be called upon whenever needed.

B. CARD-READ FIELDS

B-1 Columns of input

The Univac 60 may have a maximum of 60 columns of information wired as card-read input, while the 120 may have 120 columns of card-read input. The input columns may be divided among the elements as required by the application - on one application element N1 might contain two columns, while on another it might contain eight.

The number of input columns in each element is determined by the size of the card field which is wired to the element. In general each card column will require one column of input, though if the card field has over-capacity punching an additional column of input is needed. Since each card column is wired to input by individual positions, other types of special wiring are possible which would cause a variation between card columns and input columns. An example of this is wiring alpha into the accumulator, page 132.

B-2 Assignment of element numbers

In assigning element numbers to card fields, it makes no difference which field is assigned to which element. Card fields usually begin with element N1 for the first field required in a problem, and continue through as many as needed.

In a multiple card routine (for example, a payroll application where several cards are required to compute one man's pay), the same card columns may have different titles in different card forms. Columns 50-53 might be "hours today" in a daily summary card and "total hours" in an attendance card; columns 71-75 might be "gross pay" in a daily summary card, and "deduction amount" in a deduction card. In cases like this, only one element number is assigned, since the element number refers to the card columns wired rather than the description of the field.

B-3 Element designators

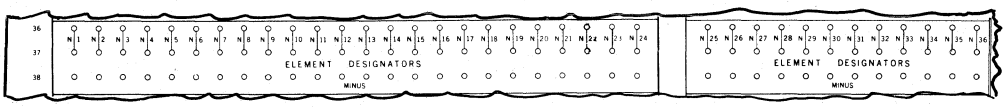
Throughout a program, whenever a particular card field is required, the columns within that field are called upon by the element number to which they have been wired. For example, suppose that columns 20-23, "rate", are wired as element N2; and columns 50-52, "hours", are wired as element N3. On step 5 of the program, hours are to be multiplied by rate. The program step would be:

Step	V1	Pr	V2	=	R
5	N3 (hours)	x	N2 (rate)	=	S1 (labor)

Although three columns are in the hours field and four columns in the rate field, only one wire on the constant-program panel is required to call on each field at the proper time, rather than a wire for each column or for each position.

This is accomplished as follows:

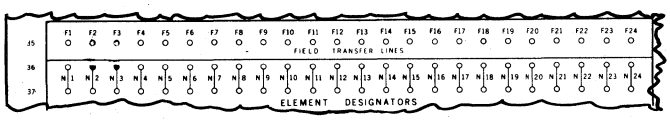
On the constant-program panel are hubs called "element designators" (lines 36-37, A-1). There is a pair of interconnected hubs for each element.



When an element is called upon to be used as a value, the corresponding element designator hubs emit power. The power is transferred to the input-output panel to energize the card columns associated with the element which has been called upon. This is done through field transfer lines or "F-lines".

B-4 Field transfer lines

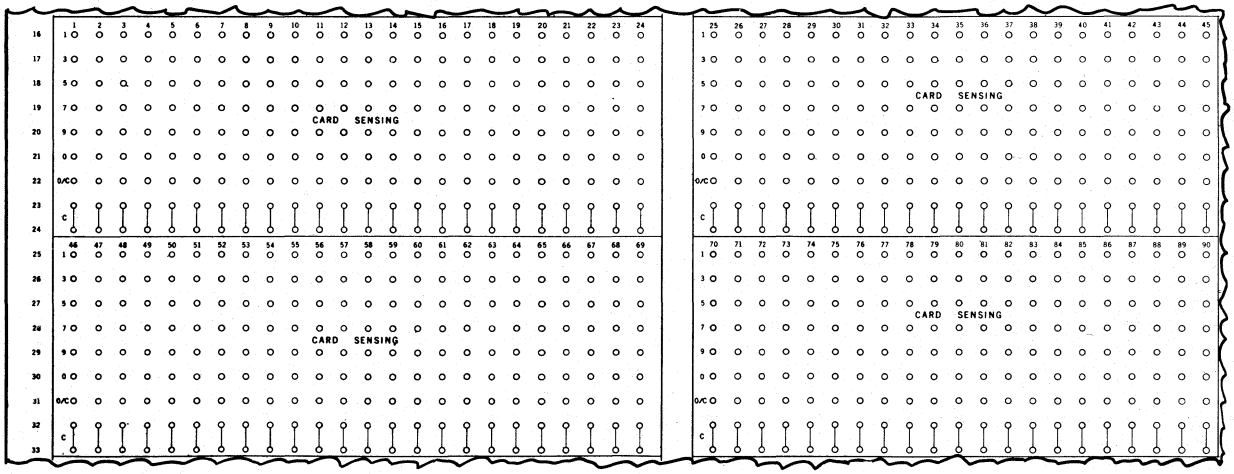
There are 48 field transfer lines on the computer, located on the constant-program panel in line 35, and on the input-output panel in line 34. They are numbered F1 to F48. They are all identical, though not connected, and are merely a means of transferring a pulse from one connection panel to the other. Any field transfer line may be used to transfer any element, though for simplicity in wiring, the F-line corresponding to the element number is usually used. For example, power from element designator N2 would be transferred through F2; from N3 through F3. This would be wired on the constant-program panel as shown at the top of the next page.



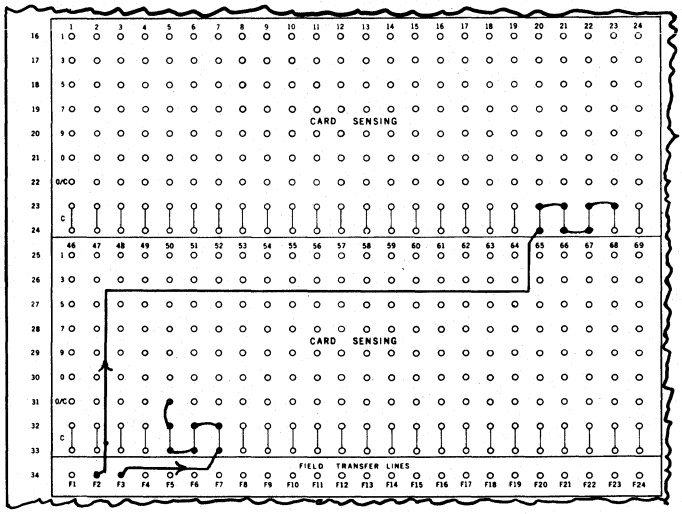
A pulse entering an F-line hub on the constant-program panel will come out of the corresponding F-line hub on the input-output panel. This in turn is wired to energize all of the columns which are associated with the element.

B-5 Card sensing

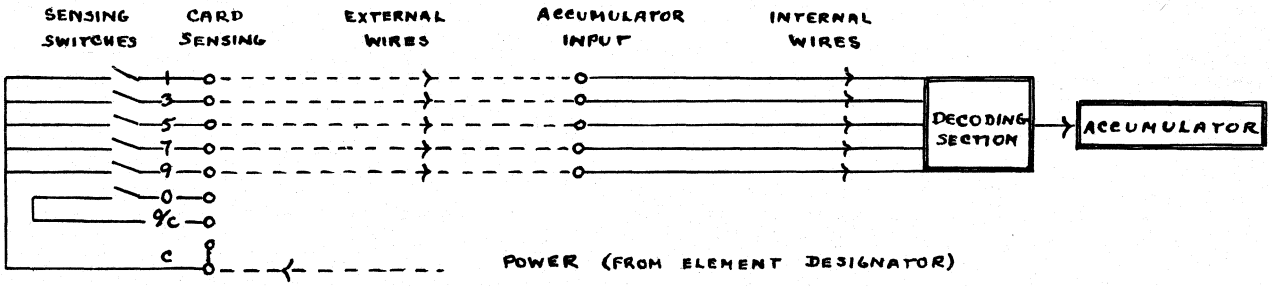
On the input-output panel, each card column is represented by nine hubs (lines 16-33, A-u).



The first 6 hubs (1,3,5,7,9,0) represent the positions in the card. The next hub (0/C), is the zero common, while the last two, which are joined together and may therefore be used interchangeably, are the common hubs for positions 1 through 9. In order to use a card column, power must flow through the common of the column, or the zero-common if zero is the position being used. Therefore the F-line which is transferring power from the element designator on the constant-program panel is wired to the commons of all columns which are to become that element. If a zero is to be used, either as over-capacity, negative control, or for any other purpose associated with that element, the zero common of that particular column must also be wired from the F-line. In the example given on page 22, columns 20-23 are rate, N2; columns 50-52 are hours, N3. The commons of these columns would be wired as follows, assuming that a zero in column 50 is to make element N3 minus:



Each card column has six sensing switches. They may be diagrammed as follows:



When a position is punched in a card column, the corresponding sensing switch closes. For example, if a 4 were punched in the above column, the 3 and 9 switches would close. All closed sensing switches will remain closed until the trip signal is given. Whenever the element is called upon, power will flow from the element designator to the F-line, through the F-line to the common of the column, and out the corresponding positions which are punched. These in turn are wired by the programmer to the desired accumulator input (see page 27.) From the accumulator input the pulses are sent through the decoding section to the accumulator.

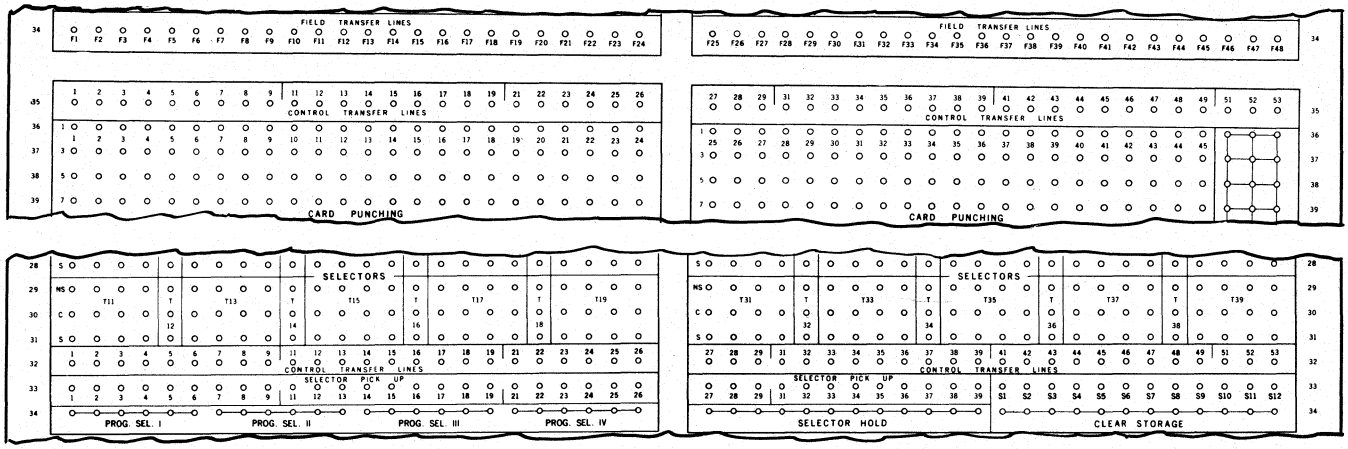
Among the functions of the decoding section is an alpha check. If any combination of punching other than a numeric code enters the decoding section, the computer will stop and the "Input check" light will light on the front of the sensing-punching unit. Since it is at this point that the alpha check is

made, a field which is wired may have alpha punched; if the field is not called on during the program for that particular card, the computer will not stop. This is of significance when certain card columns are used in computation during a detail card routine, yet in preceding heading cards the same card columns are punched with alpha. Note also that certain alpha combinations such as 0 and 7 will not stop the computer, but will read as a 7. There are no zeros in the accumulator inputs, and in this case the remaining punching is correct numeric punching.

When zero is not being used, the zero common is not wired. There is no way of wiring zero as such into the accumulator. When zero is part of a value, as in $2.00 for example, the computer treats the absence of a significant digit as zero; therefore unless zero is being used as over-capacity or some other type of special wiring, it makes no difference whether or not it is punched.

B-6 Control transfer lines

Identical in circuitry to the F-lines are control transfer lines, or C-lines. In fact, the two may be used interchangeably if necessary. There are 48 C-lines, numbered from C1 to C53, omitting C10, C20, C30, C40, and C50. On the input-output panel they are located in line 35, and on the constant-program panel in line 32. They are usually used to transfer a control position from one panel to the other.

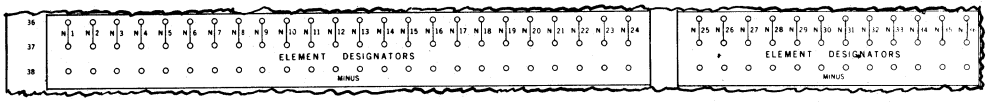


The control positions may be used for negative control, selector pick-up, or any one of several other purposes.

B-7 Negative control

In line 38, A-1 of the constant-program panel are the hubs for making an element negative. Although not shown as such, any hub will actually serve

equally well for making any element negative.

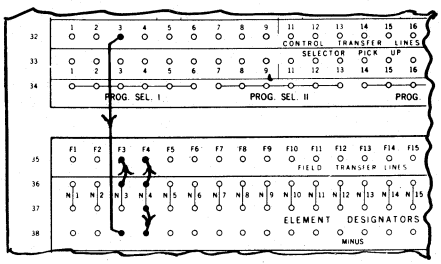


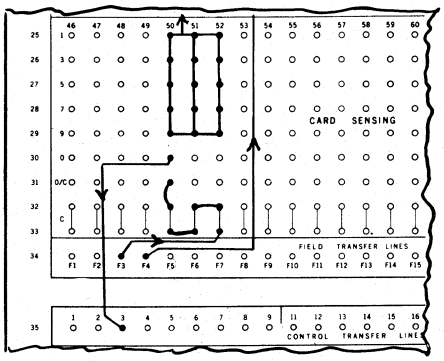
If an element is to be negative at all times, the only wiring necessary is to connect the element designator to the "minus" hub beneath it. Whenever a pulse is emitted from the element designator, the power will flow not only through the F-line to the card columns involved, but also into the "minus" hub. The columns will therefore be read as negative values.

When an element is to be negative dependent upon a control hole, the F-line carrying power from the element designator to the input-output panel is wired to the common associated with the control position. The position is then wired to a C-line, which is wired to the minus hub on the constant-program panel. Whenever the control hole is present, power flows through to make the element negative; otherwise it will not.

The same C-line number may be used for transferring a negative control as the element number and F-line number (C2 for N2). This, however, is not as usual a practice as that of using the same F-line number as element number. When wiring selector pick-ups, the wiring is easier if the pick-up is transferred on the same C-line number as the number of the selector. If C2 is used to pick up selector 2, it cannot also be used to make N2 negative; therefore some other C-line would be used for the negative control.

The following example shows the wiring to make element N4 negative at all times, and N3 negative dependent upon a zero in column 50.





Any position in a card may be used to make an element negative. For the 60 or 120, however, the most convenient position is zero, usually in the card column to the extreme left of the field, as in the example shown on the preceding page. If a zero in a column which does not belong to the card field is used as negative control, the F-line is wired to the zero-common of the column in which the control is punched.

If a position other than zero is used as negative control, it will probably be in a control column rather than the card field. If this is done, the F-line of the element is wired to the common of the column, and the position is wired to a C-line in the same way that the zero was wired. However, no other position in that column except zero, which has a separate common, may be used either for negative control or for selector pick-up unless selectors are used to do so. (See page 77). The reason is that the F-line of more than one element would be wired to the common of the column. If this were done, when calling on either one of the elements, the power would backfeed through the common of the control column, and both elements would be called on simultaneously. The result would be a combination of the two values. Since the common of a column containing selector pick-ups is wired to hubs other than F-lines, both negative control and selector pick-ups in the same column will cause incorrect operation.

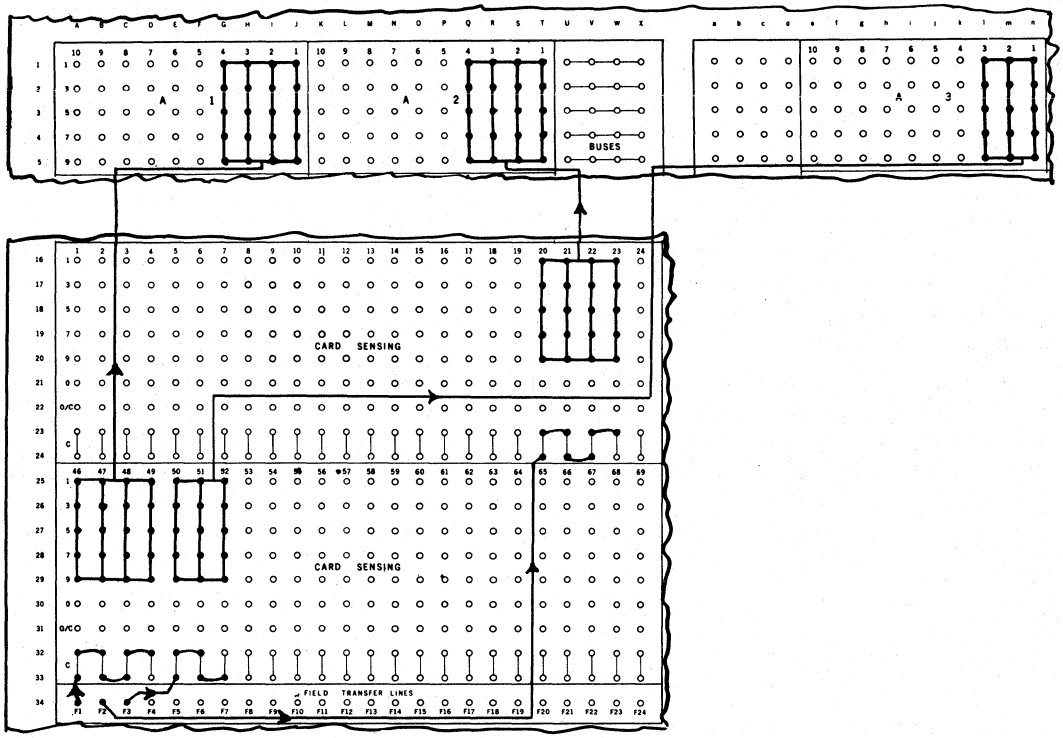
It is also impossible to use one control hole to make more than one element negative without use of selectors. The necessary Y-wiring to accomplish this would result in the same type of backfeed discussed above, with both elements being called on simultaneously.

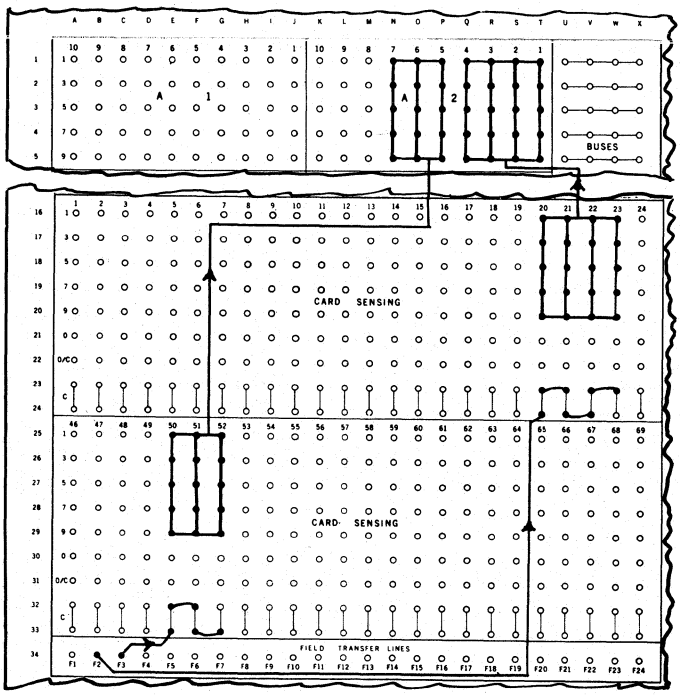
B-8 Accumulator inputs

Accumulator inputs, as mentioned on page 14, are located on the input-output panel in lines 1-15. All card-read fields which are used as elements must be wired to this section.

Each accumulator input contains 10 columns of 5 positions each. Column 1 of all 12 accumulator inputs is wired to column 1 of the accumulator, column 2 of all 12 is wired to column 2, and so forth, with protection so that there will be no backfeed. Because of this "Y-wiring", it does not matter to which accumulator input an element is wired. Most programmers assign element N1 to accumulator input A1, N2 to A2, and so forth. The elements are usually placed in the columns to the right of the accumulator input. For normal wiring, each position in a card sensing column is wired to its corresponding position in the accumulator input. For example, the first illustration at the top of the next page shows the wiring of the rate and hours fields which have been mentioned above, as well as clock number, element N1, which is punched in columns 46-49.

Since there are only 12 accumulator inputs, yet 36 possible elements, more than one element may be placed in one accumulator input. The only limitation as to the number of elements in one input is that the total number of accumulator columns to which the elements are wired may not exceed 10. Of course only the columns wired to the element being called upon at a particular moment will be active, since they are the only ones through which power is flowing. The second illustration on the next page shows how to combine rate and hours in the same accumulator input.


BUSES
CARD SENSING
CARD SENSING
FIELD TRANSFER LINES

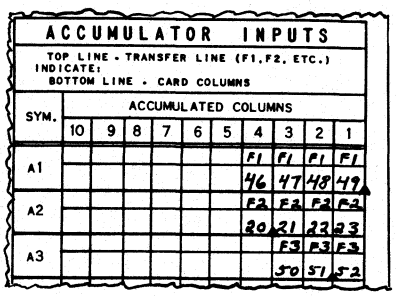

BUSES
CARD SENSING
CARD SENSING
FIELD TRANSFER LINES

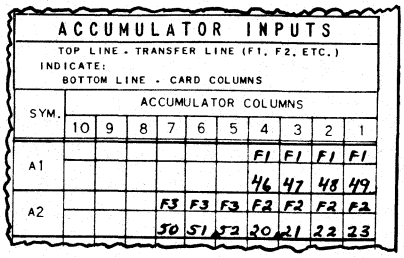
Notice that there are still three available accumulator columns, 8-10, which could be used for another 3-digit field.

"Y-wiring" is possible between card columns and accumulator inputs. One card column may be wired to two or more accumulator columns with no restriction. As an example of this type of wiring, see page 141.

There are limitations on the Y-wiring of two or more card columns to the same column in the same accumulator input. This is possible without use of selectors only if all columns are identical, or if all except one of the columns are known to be blank. If two columns punched with different values are wired to the same accumulator input column, a backfeed might develop, resulting in the reading of both columns when either is called upon. Note that the purpose of having the 12 accumulator input sections is to avoid this type of backfeed. The use of selectors to prevent the backfeed is explained on page 82.

In the upper right corner of the selector chart, form S1-1370, is a section entitled "Accumulator Inputs". There is a box for each of the 10 columns in each accumulator input. In the lower half of the box is written the card column which is wired to the column. In the upper half of the box is written the F-line with which it is associated. The way in which the above wiring would be entered is:





Incidentally, constant values are not entered in this section of the program chart.

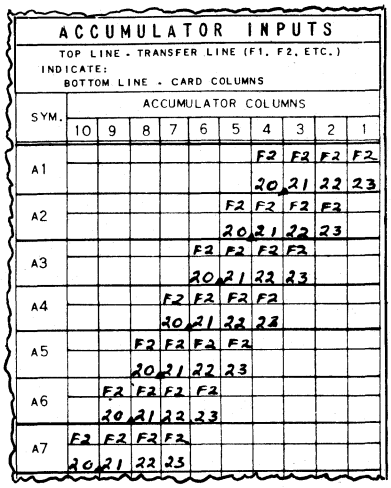
B-9 Decimal locations

Notice that the decimal location of each field is entered in the accumulator input section. Throughout the program the decimal locations of these fields would be referred to as:

Element	Example 1	Example 2
N1	1/0	1/0
N2	4/3	4/3
N3	2/1	6/5

One accumulator input will have as many decimal locations as there are elements wired into it, since the decimal location refers to the element, not the

accumulator input. The decimal location of an element is determined by placing the mark between the card columns where it is located in the card, then checking the accumulator columns between which it falls. Since the decimal locations in the computer are automatically aligned during computation, the accumulator columns into which an element is wired are of little importance as long as the decimal location is correctly indicated. For example, element N2 could be placed in seven different locations within an accumulator input, and the results would always be the same.



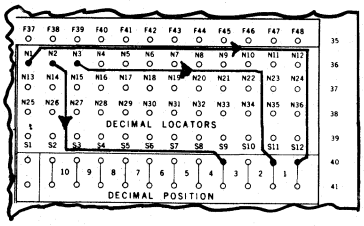
ACCUMULATOR INPUTS

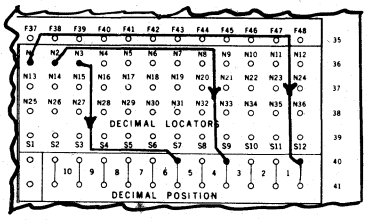
TOP LINE - TRANSFER LINE (F1, F2, ETC.)
INDICATE:
BOTTOM LINE - CARD COLUMNS

SYM.	ACCUMULATOR COLUMNS									
	10	9	8	7	6	5	4	3	2	1
A1							F2	F2	F2	F2
							20	21	22	23
A2						F2	F2	F2	F2	
						20	21	22	23	
A3					F2	F2	F2	F2		
					20	21	22	23		
A4				F2	F2	F2	F2			
				20	21	22	23			
A5			F2	F2	F2	F2				
			20	21	22	23				
A6		F2	F2	F2	F2					
		20	21	22	23					
A7	F2	F2	F2	F2						
	20	21	22	23						

Decimal locations are wired on the constant-program panel in lines 36-41, m-x. A wire is taken from the element number to the decimal position. Usually it will be necessary to use the buses in lines 42-50 for additional hubs to which to wire the most frequently used decimal locations. As is the case with any bus, these are completely neutral and any bus may be used to expand any decimal location.

The wiring of the decimal locations for the elements previously used as examples would be:





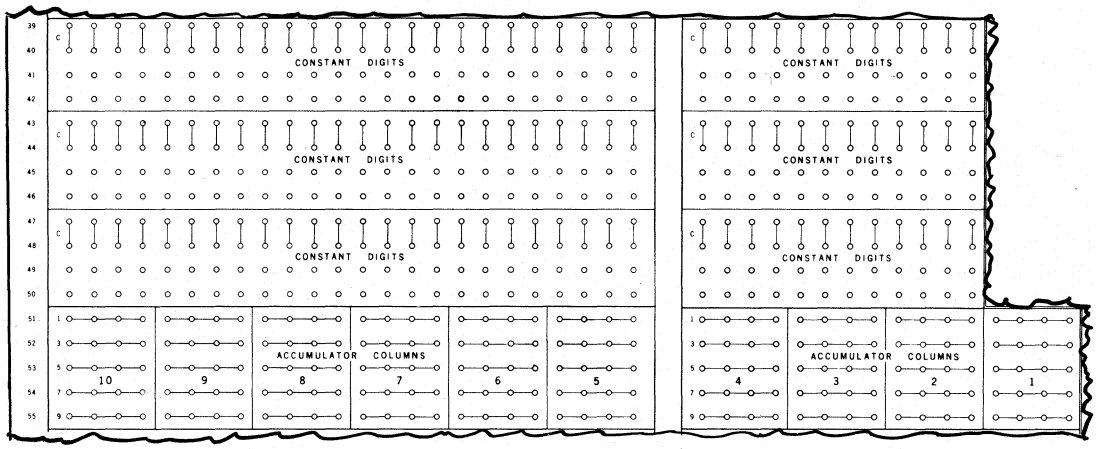
C. CONSTANTS

C-1 Constant digits

A maximum of 108 digits of constants is available on the 60 or 120. Although constants are assigned element numbers, and therefore are included in the 36 elements available, they do not reduce the 60 or 120 columns of input. A constant value may contain as many as 10 constant digits, or as few as none. Zeros are not considered digits; therefore although a constant value of zero would require an element number, usually N36, and a decimal location, it does not require any constant digits.

C-2 Wiring of constant digits

Constants are wired on the constant-program panel, lines 39-55, A-p.



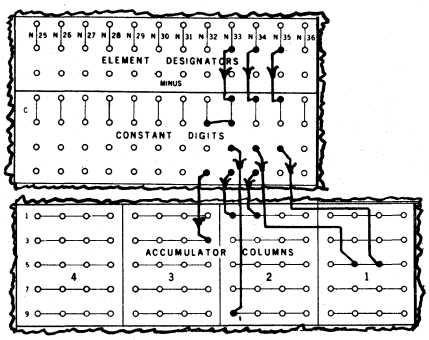
The hubs in lines 39-40, 43-44, and 47-48 are the commons. They correspond to the commons of the card sensing columns, page 23, and in order to activate the hubs beneath each common (lines 41-42, 45-46, and 49-50), power must flow through the commons. As in the case of card-read fields, the source of this power is an element designator. However, since constants are wired on the same panel as the element designators, no F-lines are needed. The element designators are wired directly to the commons of the constant digits which make up that element. If more than one constant digit is required in an element, the commons are joined together in the same way that the commons of card sensing columns are wired. When power enters the common of a constant digit, it will come out of the two hubs beneath that common. These hubs are then wired to the value desired.

In lines 51-55 are 10 sections entitled "Accumulator Columns". These accumulator columns are wired internally to the corresponding columns of the accumulator. Within each accumulator column are 5 rows of hubs, labeled 1,3,5,7,9. Constant digits are wired into the accumulator column in the 90-

column punching code. To make a 5, one of the two hubs beneath the common would be wired to a 5 in the accumulator column desired; to make a 6, the other hub would be wired to a 9. Either hub can be wired to either position.

In creating odd digits, which require only one hub, the other hub can be used to create another digit in the same element. For example, a constant value of .015 would require only one constant digit: the zero is not wired; both 1 and 5 are odd digits, so one hub of the constant digit would be wired to a 1 in accumulator column 2, the other to a 5 in accumulator column 1. This in effect increases the number of constant digits on the computer by one-half of the number of odd digits used in constant values. Note, however, that a hub may be borrowed only within the same element number or constant value. If a constant value of a .1 and a constant value of a .05 are needed, two constant digits are required.

The following example shows the wiring of N33 as 320, N34 as .15, and N35 as .005.



C-3 Decimal locations

When determining the accumulator columns in which to wire constant digits, zeros must be allowed for even though they are not wired. For example:

Constant Value	Constant Digits Required	Accumulator Columns 10 9 8 7 6 5 4 3 2 1	Decimal Location
2000	1	2 - - -	1/0
.016	2	- 1 6	4/3
10.36	2	1 - 3 6	3/2
1,000,000	1	1 - - - - - -	1/0

As in the case of card-read fields, constant digits may be wired into any accumulator columns, provided the decimal location is properly assigned with relation to the value.

A constant value of zero must be assigned a decimal location. It is usually assigned 4/3, although it makes little difference to the operation of the computer what location is assigned.

C-4 Assignment of element numbers

In assigning element numbers to constant values, it is usual to begin with N36 and work backwards as additional constants are needed. Since card-read fields usually begin with N1, a low element number usually refers to a card-read field while a high number refers to a constant.

There are two constants which are almost always required in a program - zero and a "rounding 5" (.005). The use of these two constants will be explained later. Because they are used so frequently, most programmers are in the habit of assigning zero as element N36, and .005 as element N35. If the computer does not have a full complement of elements, zero would be either N24 or N12, while the rounding 5 would be N23 or N11.

D. PROGRAM CHARTS

D-1 Table of factors

Probably the first step in using the program charts for a program is to assign element numbers, using program chart S1-1369. At the top of this chart is a Table of Factors.

PROGRAM CHART NO.
APPLICATION:

UNIVAC 60 & 120
PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand
TABULATING MACHINES

TABLE OF FACTORS

SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE
N1				N13				N25			
N2				N14				N26			
N3				N15				N27			
N4				N16				N28			
N5				N17				N29			
N6				N18				N30			
N7				N19				N31			
N8				N20				N32			
N9				N21				N33			
N10				N22				N34			
N11				N23				N35			
N12				N24				N36			

CARD DESCRIPTION

Both card read fields and constant values are entered here as follows:

Sym: The 36 possible elements which may be assigned.

Sign: The sign of the field or constant value. If the element is always plus, enter a +. If it is always minus, enter a -. If it may be either plus or minus, enter ± .

Card No.: In a multiple card routine, each type of card should be assigned a card number. This may either correspond to a code punched in the card, or merely be an arbitrary numbering system. In either case, the code should be noted beneath the table of factors in "Card Description". In the card number space next to each card-read field on the program chart should be noted the card number of the card or cards in which this field is punched. Card number is not needed for a constant value.

Card field title or constant value: In the case of a card-read field, the title should be written. If the same card columns are used for more than one purpose, each title should be shown. The card columns may also be indicated. For constant values, the actual value should be entered.

The method of entering the elements previously used as examples is shown below:

PROGRAM CHART NO.
APPLICATION:

UNIVAC 60 & 120
PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand
TABULATING MACHINES

TABLE OF FACTORS

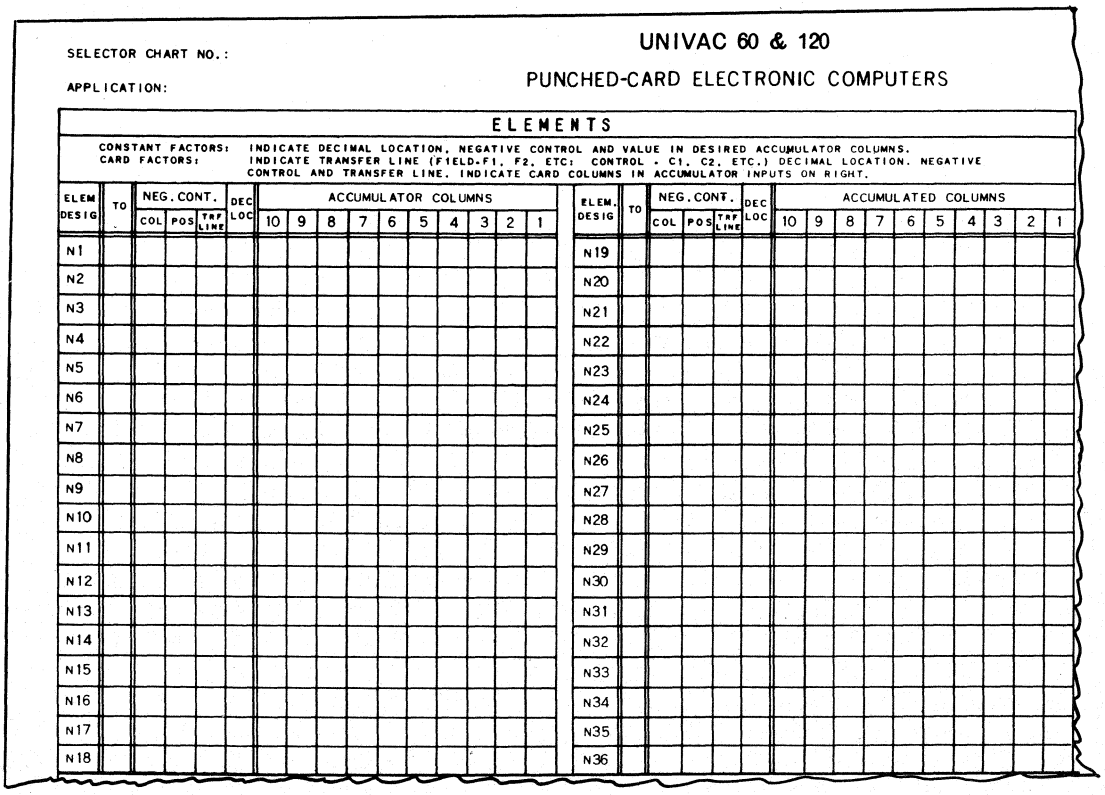
SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE
N1	+		CLOCK NO. (COLS. 46-49)	N13				N25			
N2	+		RATE (COLS. 20-23)	N14				N26			
N3	±		HOURS (COLS. 50-52)	N15				N27			
N4				N16				N28			
N5				N17				N29			
N6				N18				N30			
N7				N19				N31			
N8				N20				N32			
N9				N21				N33			
N10				N22				N34			
N11				N23				N35	+		.005
N12				N24				N36	+		0.000

D-2 Accumulator inputs

Following the assignment of element numbers, the card-read fields should be entered in the accumulator input section of the selector chart, form S1-1370. The use of this section has been explained previously on page 27. Constants are not entered here.

D-3 Elements

To the left of the accumulator input section is a section entitled "Elements". Both card-read fields and constant values are entered here.



SELECTOR CHART NO.:

APPLICATION:

UNIVAC 60 & 120

PUNCHED-CARD ELECTRONIC COMPUTERS

ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD.F1, F2, ETC; CONTROL - C1, C2, ETC.) DECIMAL LOCATION. NEGATIVE CONTROL AND TRANSFER LINE. INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N1															
N2															
N3															
N4															
N5															
N6															
N7															
N8															
N9															
N10															
N11															
N12															
N13															
N14															
N15															
N16															
N17															
N18															

ELEM. DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															
N22															
N23															
N24															
N25															
N26															
N27															
N28															
N29															
N30															
N31															
N32															
N33															
N34															
N35															
N36															

Element Designator: This is a list of the 36 elements.

To: Enter the F-line which is assigned to the element. No F-line is needed for constants.

Negative Control: If an element may be negative, the position in the card which makes it negative is recorded here. If it is always to be negative, write "Always" across these columns.

(a) Column - The column in which the negative control is punched.

(b) Position - The position which represents the negative control.

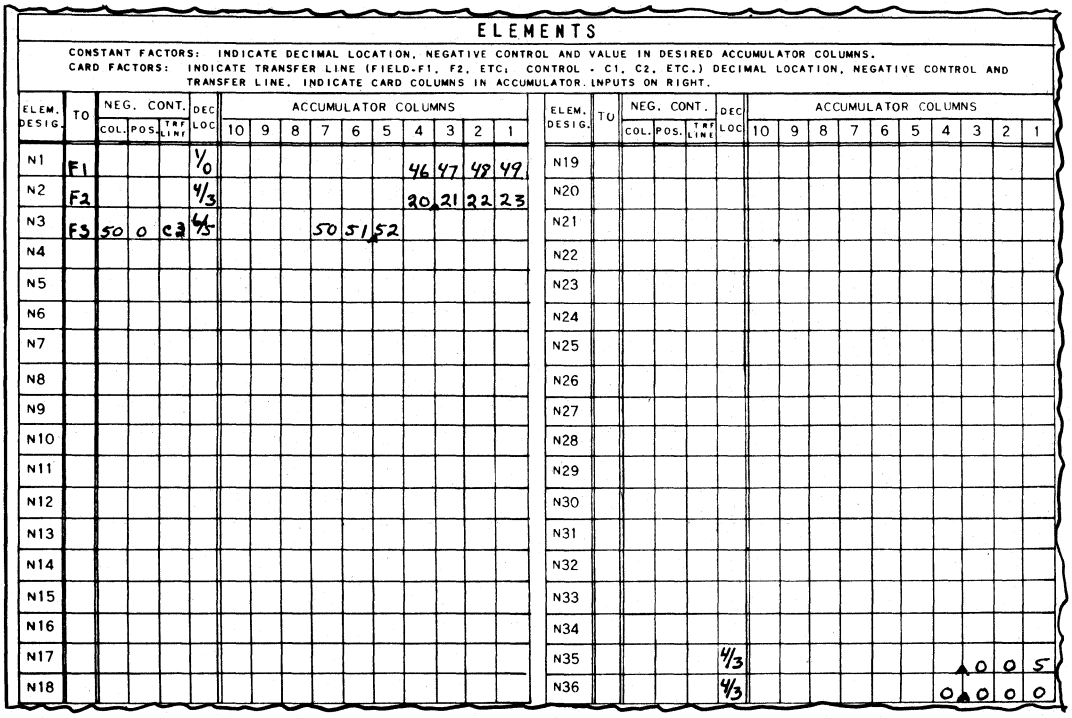
(c) Transfer line - The control transfer line which transfers the negative control from the input-output panel to the constant-program panel.

Decimal Location: The decimal location of the element, given in terms of accumulator columns.

Accumulator Columns: Enter the card columns beneath the accumulator columns to which they will be wired, and the decimal location. For card-read fields this will be a copy of the accumulator input section without the F-lines, except that the card columns are arranged by element number instead of

accumulator input. In the case of constants, the actual constant value is entered. In most cases zeros are written, even though they are not wired.

The following sample shows how the card fields previously used would be entered, using the case of two elements in the same accumulator input.



ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD-F1, F2, ETC; CONTROL - C1, C2, ETC.) DECIMAL LOCATION, NEGATIVE CONTROL AND TRANSFER LINE. INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM. DESIG.	TO	NEG. CONT. COL.	NEG. CONT. POS.	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N1	F1				1/0							46	47	48	49
N2	F2				4/3							20	21	22	23
N3	F3	50	0	C3	4/5				50	51	52				
N4															
N5															
N6															
N7															
N8															
N9															
N10															
N11															
N12															
N13															
N14															
N15															
N16															
N17															
N18															

ELEM. DESIG.	TO	NEG. CONT. COL.	NEG. CONT. POS.	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															
N22															
N23															
N24															
N25															
N26															
N27															
N28															
N29															
N30															
N31															
N32															
N33															
N34															
N35					4/3								0	0	5
N36					4/3							0	0	0	0

E. SUMMARY

1) There is a maximum of 36 elements available, numbered N1-N36. An element is either a card-read field or a constant value.

2) An element may range in size from zero to 10 digits.

3) Every element must have a decimal location.

4) Any element may be either plus or minus. Unless otherwise indicated, an element will be considered plus by the computer.

5) An element may be either V1 and/or V2 of a program step.

6) Both card-read fields and constant values are available at any time during the program for a card, without being placed in storage.

7) The Univac 60 has 60 columns of card-read input available; the Univac 120 has 120 columns of card-read input available. This does not include constant digits.

8) The maximum number of constant digits available is 108, though this may be increased when odd digits are part of a constant value.

9) An alpha input check is made when an element is called upon; the presence of a code which is not numeric will stop the computer.

10) Zeros are not wired into the accumulator; therefore either a zero or a blank will be treated the same.

11) The 1 through 9 positions in one column may be used to make only one element negative without use of selectors. Zero is the preferable negative control.

3. Storage

A. GENERAL

A-1 Storage capacity

The Univac 60 has 60 columns of storage, the Univac 120 has 120 columns of storage. These storage columns are divided into units of 10 columns each, numbered S1-S6 or S1-S12. Each storage unit also carries a sign, plus or minus.

A-2 Use of storage as a value or result

The result of a program step is placed in storage. When this has been done, this result may be used as a value in a succeeding program step, and/or be punched in the card that is in the computer or any succeeding card.

Although a storage unit may be used as either a value or a result in a program step, the same unit cannot be both a value and the result in the same program step. This is due to the automatic self-checking feature of the computer. For an example of what would happen if this were done, see page 47.

Whenever the result of a step is zero, the sign of that result will be plus, even though it should theoretically carry no sign. The computer automatically makes zero a plus value.

The storage which is used as the result of a step is always cleared at the beginning of the step, before computation occurs. For example, suppose storage S1 contains a value of 524.294 at the end of step 5. During step 6, $7.50 is to be multiplied by .02, and the result placed in S1. As soon as a pulse is received at the "in" of step 6, S1 is cleared. $7.50 is then multiplied by .02, and .150 is placed in S1.

A-3 Clearing storage

Once a value has been placed in storage, it will remain there until one of four things occurs:

1. A new result is entered.
2. A clear signal is given to the unit, either during the program or manually.
3. The power is turned off.
4. The temperature of the computer becomes too high.

Therefore a value in storage may be used as often as needed during a program; it may even be punched into a card and still be available for further use. It may be used on as many succeeding cards as necessary.

A-4 Decimal locations

Every storage unit which is used must have a decimal setting, or the computer will hang up during the program step which first calls for the storage unit.

When a result has been computed, its decimal location is aligned with that of the storage unit in which it is to be placed; if none has been assigned, the computer keeps searching for one.

It is a responsibility of the programmer to be certain that no result will be aligned around a decimal location so that any digit of the result extends into column 11 of the A section of the accumulator. To do so will cause the computer to hang up (see accumulator, page 19).

However, digits may be dropped off to the right. Such digits will have no bearing on any further computation. For example, as the result of a multiplication, a value of 185.46312 is computed. This value could be placed in a storage unit as follows, depending upon the decimal location of the storage.

Accumulator Columns										Decimal Location
10	9	8	7	6	5	4	3	2	1	
		1	8	5	4	6	3	1	2	6/5
			1	8	5	4	6	3	1	5/4
				1	8	5	4	6	3	4/3
					1	8	5	4	6	3/2
						1	8	5	4	2/1
							1	8	5	1/0

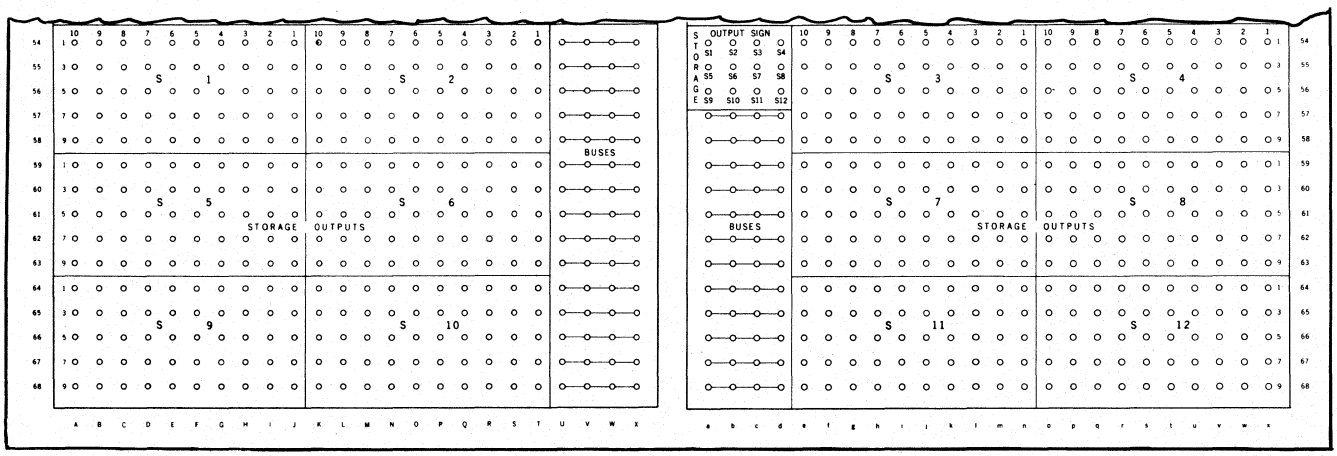
Through use of program selects, it is possible to change the decimal location of a storage unit during the program. This is covered on page 81.

B. PUNCHING FROM STORAGE

B-1 Wiring for punching

It is possible to punch from any or all storages on the computer. As many columns as desired may be wired to punch from each unit, and may punch in any card columns.

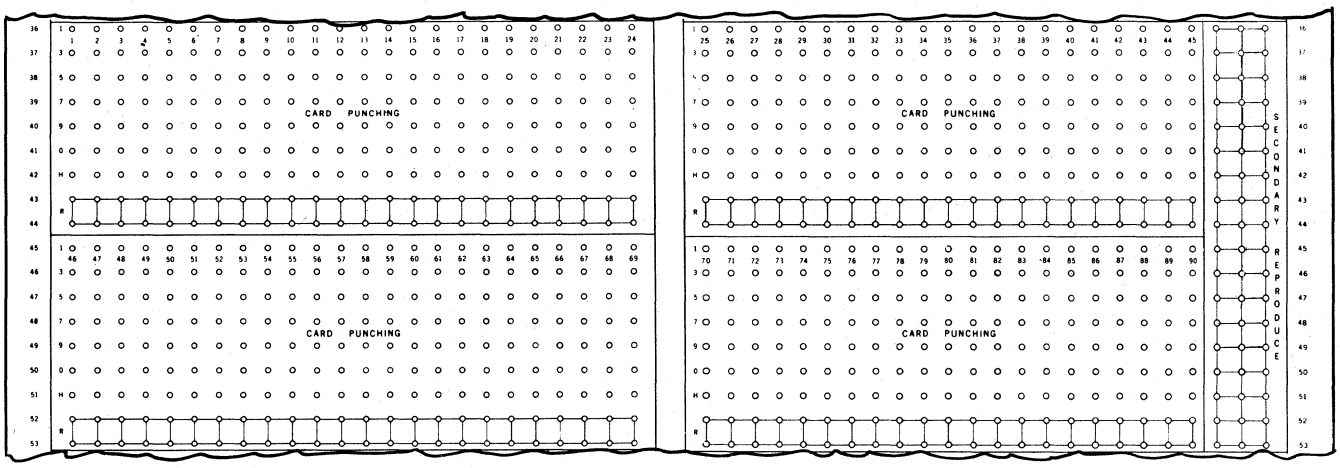
On the input-output panel the storage outputs for punching are located in lines 54-68, A-T and e-x. There are 10 columns of 5 positions each in each unit.



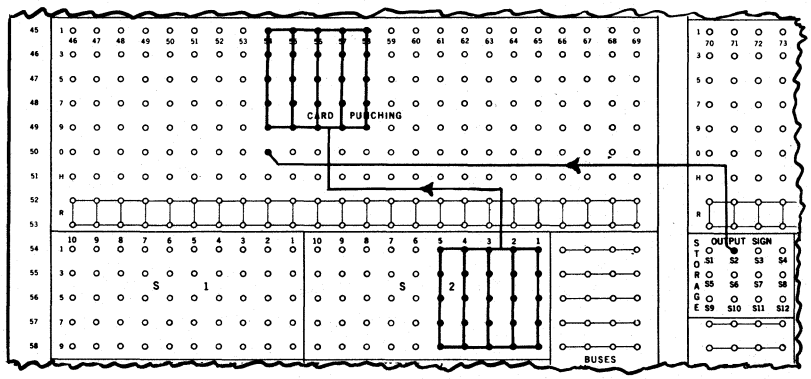
Note that there are no zero positions in storage output, so therefore the computer normally does not punch zeros. Through special programming techniques, however, it is possible to punch zeros if they are necessary (see page 196).

In lines 54-56, a-d, is a box entitled "output sign". If the storage is negative when the value is set in the punching dies, a pulse will be emitted from the sign box. This pulse may be wired to any position in the card, including a zero, as a negative indication.

Storage outputs are wired to the card punching section, lines 36-41, 45-50, A-u.



Lines 42-44 and 51-53 are used in reproducing. Normally the wiring is position for position, but as in the case of input, any position can be wired to any position. In the case of over-capacity punching, for example, a 1 in a storage unit would be wired to a zero in the card. As an example of output punching, the wiring of labor from S2 to card columns 54-58 with a zero in column 54 as negative sign would be:



B-2 Y-wiring of storage output

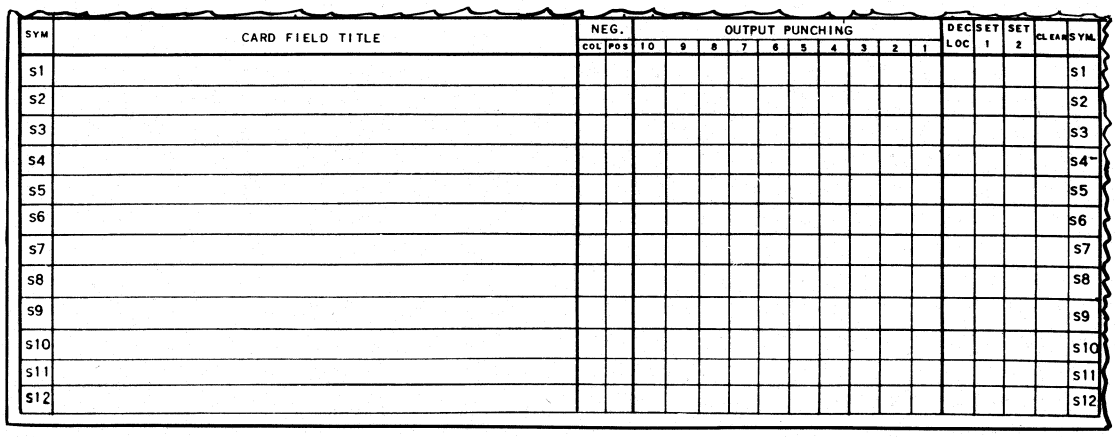
Y-wiring of output punching should not be done. Due to the voltage of the computer during a set operation, the wiring of one storage column or output sign to two or more card columns or positions will not always result in correct punching. The wiring of two storage columns or signs to one card

column or position may result in damage to the punch magnets. However, this type of wiring is possible when only one storage unit or sign contains a value when setting occurs. If all other storage columns or signs wired to the same card column or position are known to be blank, Y-wiring may be done.

One of the most frequent reasons for the punching of two storages into the same card columns is that information accumulated for a summary card must be punched in the same card columns as the detail information. Of course in a case such as this, the Y-wiring of the two storages to the same card column would also result in a backfeed - when the detail information is being set in the punching dies, the wires to the summary storage would permit that information to be set at the same time, with a meaningless result. To avoid this problem, the usual solution is to transfer the summary information to the detail storage as part of the summary card routine (see page 103).

C. PROGRAM CHARTS

At the bottom of program chart S1-1369 is a form for indicating output punching from storage:



SYM	CARD FIELD TITLE	NEG. COL	NEG. POS	OUTPUT PUNCHING 10	9	8	7	6	5	4	3	2	1	DEC LOC	SET 1	SET 2	CLEAR	SYM.
S1																		S1
S2																		S2
S3																		S3
S4																		S4
S5																		S5
S6																		S6
S7																		S7
S8																		S8
S9																		S9
S10																		S10
S11																		S11
S12																		S12

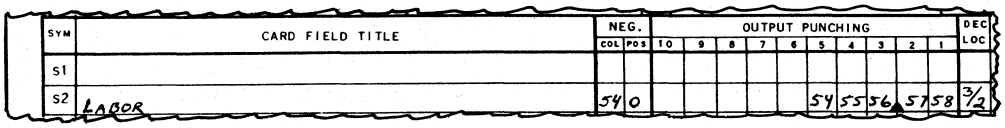
Sym: This is a list of the 12 storage units.

Card Field Title: This space is provided for a description of the field to be punched. It is usual practice to write only the titles of fields to be punched, not those used as "working" or intermediate storages. Such storages usually receive several values during a program, and nothing would be gained by writing each description in this space. However, the word "working" or "intermediate" might be written here.

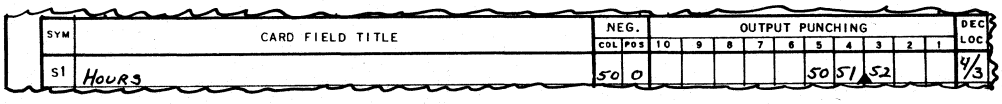
Negative - Column, Position: If a control hole is to be punched in the card indicating that the value in the storage is negative when it is punched,

the column and position into which the control is punched are indicated here.

Output Punching: The 10 columns in a storage unit are listed here. Beneath each column is written the card column into which that storage column is to punch. In doing this, it is well to indicate the decimal location of the storage unit first, and then to align the card columns around the decimal point. The method of entering the example given in the wiring section would be:



A value may be punched from a storage with a decimal location which is not the same as that of the value. For example, "hours" with one place following the decimal, may be placed in a storage with a 4/3 decimal location. If it is to be punched in columns 50-52, the output punching would appear as follows:



Decimal Location: The decimal location of any storage which is used during a problem should be indicated here. It is well to assign a decimal location the first time a storage is used, and to check all succeeding entries into the unit against the decimal location. This is to make certain that digits which should be retained are not dropping off to the right, and that the size of the value does not exceed the capacity of the unit.

The columns headed Set 1, Set 2, and Clear will be covered in their respective sections.

D. SUMMARY

1) The Univac 60 has 60 columns of storage; the Univac 120 has 120 columns of storage. The columns are divided in units of 10 columns each, plus a sign for each unit.

2) The wiring of a storage as the result of a step automatically clears that storage at the beginning of the program step.

3) One storage unit may be used as either a value or result, but not both in the same step.

4) Whenever the result of a step is zero, the sign of the storage unit in which it is placed will be plus.

5) A storage unit may be cleared only by entering a new value, delivering a clear signal, turning off the power, or if the temperature becomes too high.

6) Every storage unit must have a decimal location.

7) Digits may be dropped off the right of a storage unit, but attempting to store a value with a significant digit in the 11th column of the A section of the accumulator causes the computer to hang up.

8) The computer does not normally punch zeros from storage.

9) In general, no Y-wiring should be done from storage, either the punching of two units in the same card columns or the punching of the same unit in two different locations.

4. Program step

A. GENERAL

A program step is the fundamental method of directing the computer through the routine necessary to solve a problem. The maximum number of steps available is 40, though this may be expanded through use of selectors to reuse program steps.

A-1 Sequence of program steps

The 60 and 120 are completely non-sequential. Though the program steps are listed 1 through 40 on the program chart, this does not mean that a program must go from step 1 to step 2 to step 3, etc. The sequence of the steps is completely under the control of the programmer. The "exit" or branching of each step is wired to the next step desired, which may be the next numerical step, several steps further down, or even a step with a number lower than the one which has just been completed. It is possible to reuse steps, or to set up a "loop", which involves going through the same series of steps until the desired result is obtained.

The basic form of a program step is:

Step Number	Value 1 (V1)	Process (Pr)	Value 2 (V2)	=	Result (R)	Branching ± (-Br +Br)

A-2 Process

There are four processes on the computer, any one of which may be used on any step. They are the four arithmetic operations of addition, subtraction, multiplication and division. The computer considers the signs of both values when doing a computation, and delivers the sign to the result storage unit in accordance with algebraic rules.

When doing addition, it makes no difference to the computer which factor is assigned as V1, and which as V2.

	V1	Pr	V2	=	R
	Amount	+	Rounding 5	=	Rounded amount
	S1 (12.375)	+	N35 (.005)	=	S2 (12.38)
or	Rounding 5	+	Amount	=	Rounded amount
	N35 (.005)	+	S1 (12.375)	=	S2 (12.38)

In subtraction, the value to be subtracted is V2, the value from which it is subtracted is V1.

V1	Pr	V2	=	R
Gross Pay	-	Non-taxable pay	=	Taxable pay
S4 (106.95)	-	S3 (52.00)	=	S1 (54.95)

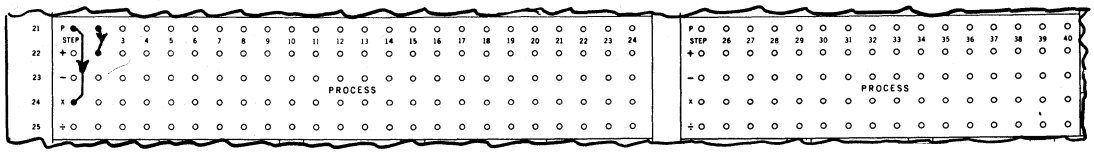
In multiplication, the smaller value should preferably be V1, the larger value V2. This increases the speed of multiplication; however, the correct result will be obtained either way.

	V1	Pr	V2	=	R
	Hours	×	Rate	=	Labor amount
	N3 (1.5)	×	N2 (1.345)	=	S5 (2.0175)
or	Rate	×	Hours	=	Labor amount
	N2 (1.345)	×	N3 (1.5)	=	S5 (2.0175)

In division, the value which is to be divided is V1, the value by which it is divided is V2.

V1	Pr	V2	=	R
Total Pay	÷	Total Hours	=	Average rate
S4 (94.25)	÷	S2 (48.0)	=	S1 (1.9635)

The wiring of the process is done on the constant-program panel, lines 21-25, A-p. In line 21 is a row of hubs labeled "P", one for each of the 40 steps. Each emits a pulse at the process time for the corresponding step. This hub is wired to one of the four hubs beneath it: +, -, ×, or ÷. Actually any of the 40 hubs in the +, -, × and ÷ rows would serve equally well for any program step, but for ease of wiring and checking, it is well to use an arithmetic hub directly beneath the process hub for the step being wired. For example, assume that step 1 is multiplication, and step 2 addition:

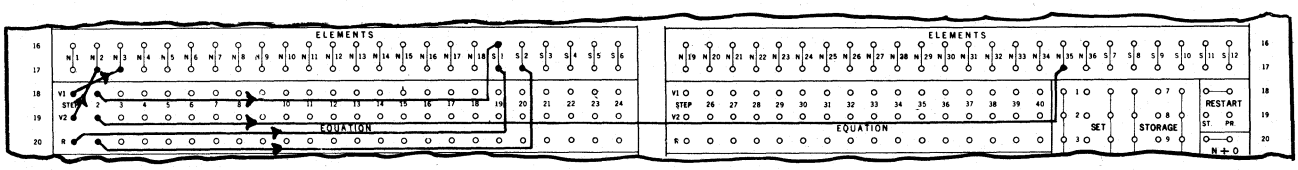


A-3 Value 1, value 2, and result

Value 1 and value 2 of a program step may be either an element or a storage. If desired, both V1 and V2 may be the same element or storage. The result of a program step is placed in a storage. It should not be an element, except as described on page 116. The same storage may not be used as both a value and the result in the same program step (see page 47).

The values and results of program steps are wired on the constant-program panel in lines 18-20, A-p. Line 18 contains a hub for V1 of each of the 40 possible steps, line 19 for V2, and line 20 for R. The V1, V2, and R hubs emit a pulse in turn for the program step being calculated. These hubs are wired to the proper element, lines 16-17, A-R and a-r or storage, lines 16-17, S-X, and s-x. In many cases, of course, the buses above the elements and storages will be used. There is no danger of a back feed in this section. As an example of wiring, assume the following:

Step	V1	V2	R
1.	N3	N2	S1
2.	S1	N35	S2



A-4 Branching

The branching of a program step is the means of associating program steps to form a complete program. For each program step there are two possibilities - plus branching and minus branching. The branching will come from the plus side if value which is placed in storage as the result of the step is plus, from the minus side if the value is minus. It will always come from the plus hub if the result is zero.

The branching is wired to the entry of the next step or operational function desired. If there is a possibility that the result can be either plus or minus, both branchings must be wired. However, if it is known that the result can be only plus or only minus, it is not necessary to wire the other branching. If a branching should occur which is not wired, the computer will hang up for lack of an instruction as to the next step.

Both plus and minus branchings may be wired to the same next step or operation, or they may be wired to different ones. For example, both plus and minus branching of step 3 might be wired to step 4, or plus branching might be wired to step 4 while minus branching is wired to step 16. However, one branching, such as plus branching of step 3, may not be wired to two steps - such as both step 4 and step 16.

As many steps or operational functions as desired may be wired to the entry of a step. For example the entry of step 21 might come from minus branching of step 3, plus and minus branching of step 10, and sort 1.

The various operational functions which may be wired from the exit of a step, and to the entry of a step are (see page 50 for explanation of each function):

Set 1 and Set 2

Sort 1 and Sort 2

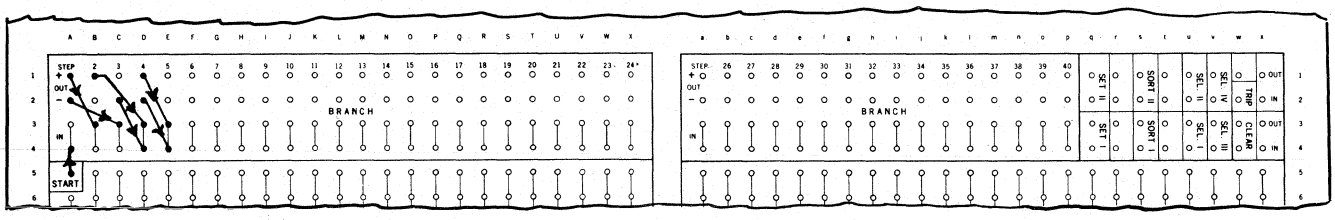
Clear

Program Selects 1, 2, 3, 4

"Start" is wired to the entry of a step or operational function, and "trip" is wired from the exit or branching of a step or operational function. Step branching is wired on the constant-program panel in lines 1-4, A-p. Line 1 contains one hub for each step, which emits a pulse if the sign of the result storage on that step is plus; line 2 emits a pulse if the sign is minus; lines 3 and 4 are the entry hubs for each step.

As an example of wiring, assume the following:

From	To
Start	Step 1
Step 1, +Branching	Step 2
-Branching	Step 3
Step 2, +Branching	Step 4
Step 3, -Branching	Step 4
Step 4, +Branching	Step 5
-Branching	Step 5



A-5 Proof

Every program step on the 60 or 120 goes through a proof of computation before continuing to the next step. This is automatic, cannot be disabled, and is not wired by the programmer. The proof of a step is part of the step, and does not reduce the maximum of 40 available steps. In doing the checking, the computer reverses the indicated process, subtracts V1, and balances to zero.

	Program Step	Check
Addition:	V1 + V2 = R	R - V2 - V1 = 0
	N1 + N2 = S1	S1 - N2 - N1 = 0
	5 + 8 = 13	13 - 8 - 5 = 0
Subtraction:	V1 - V2 = R	R + V2 - V1 = 0
	N1 - N2 = S1	S1 + N2 - N1 = 0
	12 - 4 = 8	8 + 4 - 12 = 0
Multiplication:	V1 × V2 = R	R ÷ V2 - V1 = 0
	N1 × N2 = S1	S1 ÷ N2 - N1 = 0
	3 × 6 = 18	18 ÷ 6 - 3 = 0
Division:	V1 ÷ V2 = R	R × V2 - V1 = 0
	N1 ÷ N2 = S1	S1 × N2 - N1 = 0
	21 ÷ 7 = 3	3 × 7 - 21 = 0

The result used in proof is actually obtained from the accumulator, so even though decimal positions are "dropped off" on the right into the M section, these positions will be considered when proving the step. In division, where the result of the division step may not come out evenly, the remainder is added to the result of multiplication before subtracting V1 and balancing to zero.

This method of proof is the reason that it is impossible to use the same storage unit as both a value in a step and the result of the same step. Assuming that the computer is wired to repeat the step in case the proof is not zero, this is what would happen:

	Program Step 3	Proof
	V1 + V2 = R	R - V2 - V1 = 0
	N1 + S1 = S1	S1 - S1 - N1 = 0
1st trial	5 + 0 = 5	5 - 5 - 5 ≠ 0
2nd trial	5 + 0 = 5	5 - 5 - 5 ≠ 0

As indicated in the above example, no matter what the value of S1 before step 3, it is automatically cleared at the beginning of step 3 because S1 is wired as the result storage.

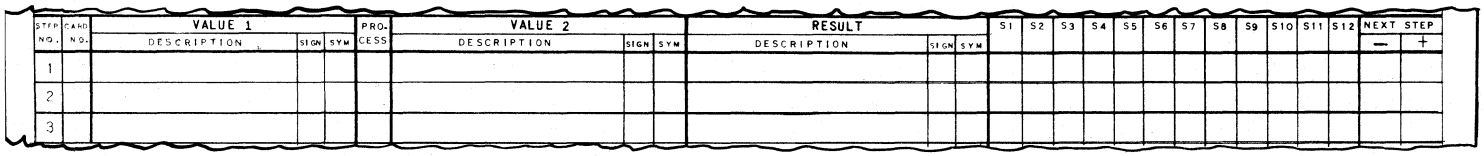
If at any time the proof of a step does not equal zero, the computer will repeat the calculation. Under control of the programmer, either the step will be repeated, or the computer will return to the first step of the program, and recalculate the entire program. Generally the computer is wired for step repeat (see page 60). Whichever it may be, the computer will not continue past a step until the step checks out to zero. If a tube is failing intermittantly, the computer may slow down due to the fact that steps are being repeated frequently; eventually it will probably stop.

A-6 Speed

The base speed of the sensing-punching unit is 150 cards a minute. Probably a majority of the programs will operate at this speed. However, it is possible, due to particularly long or iterative programs, for the calculation to require more time for calculation than that allowed at 150 cards a minute. In such a case, the sensing-punching unit will automatically wait for the end of the calculation before punching and continuing with the next card.

B. PROGRAM CHARTS

Program chart S1-1369 contains a list of 40 steps where the actual program steps are listed. Most programmers find that it is well to flow chart the problem before beginning to work with this section, in order to have a good idea of the steps through which the program must go. When this section has been completed, a final flow chart may be drawn, for ease of explanation to some one else as well as simplifying the change of a program should it become necessary.



Step No.: A list of the 40 possible steps.

Card No.: In a multiple card routine, list the card numbers of the cards which go through this step. The numbers of the cards will be the same as those used in the table of factors (see page 34).

Value 1, Value 2, Result:

(a) Description - Write the title in the case of a card-read field. For a constant value, write the actual value of the constant. In the case of a storage, write the description of the value which is in storage at the time. Certain standard descriptions which are used in the various programming techniques will be shown in connection with the techniques.

(b) Sign - If the value or result is always plus, write +; if always minus, write -. If it may be either plus or minus, write ±. In determining whether a result may be plus or minus, remember to take the signs of value 1 and value 2 into consideration, as well as the process.

(c) Sym - Write the element number (N1-N36) or storage number (S1-S12) which is to be used as the value or result.

Process: Write the symbol for the arithmetic process to be used on this step: +, -, ×, ÷.

S1-S12: This section is for the assistance of the programmer to help him remember which storages contain values and which are free. In the space directly beneath the storage number, enter the decimal location. As a storage is used, the step number on which it is used should be written in the space where the step and storage meet. If the value is to be punched, it should be circled to indicate that nothing else should be put in the storage. As soon as a value is no longer needed, a line should be drawn across the storage on the step in which it becomes free.

Next Step: In these boxes is written the next step or operational function. If there is a possibility that a result can be either plus or minus, both boxes must be filled for that step, even though the same number is written in both. If there is only one possible sign for a result, a small dash should be written in the box which is not to be wired. Frequently in programming when a result can be either plus or minus, it is necessary to follow one of the two branches through a series of steps, then return to pick up the other side. By putting a dash where it is unnecessary to wire the branching, the programmer can tell at a glance that he has considered the possibility of this branching, and knows that it will not occur. However, if there are any blank spaces when a program has been completed, the programmer knows that the condition may exist, yet he has forgotten to wire it.

Two sample steps for which the wiring has been shown in the previous sections are entered below on a program chart.

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
1		HOURS	±	N3	X	RATE	+	N2	LABOR (NR)	±	S1	1												3	2
2		LABOR (NR)	+	S1	+	.005	+	N35	LABOR (RD)	+	S2		②											-	4

C. SUMMARY

1) The maximum number of steps available is 40.

2) The 60 or 120 is a non-sequential computer with regard to program steps.

3) There are 4 processes: addition, subtraction, multiplication, and division.

4) The computer considers the signs of both values and determines the sign of the result in accordance with algebraic laws.

5) V1 and V2 of a program step may be either an element or a storage; the result is placed in a storage.

6) The branching of a step may be either plus or minus, depending on the sign of the result storage.

7) The branchings of as many steps or operational functions as desired may be wired to the entry of a step, but a particular plus or minus branching may be wired to only one step or operational function.

8) Every step of a program goes through an automatic proof where it must balance to zero.

9) The base speed of the 60 or 120 is 150 cards a minute, but if calculation cannot be completed at that speed, the computer will automatically wait for the completion of calculation before punching.

5. Operational functions

A. GENERAL

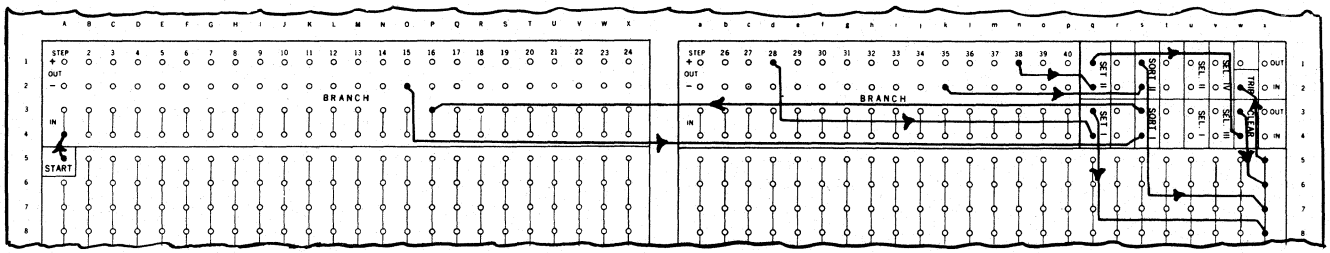
A-1 Description of operational functions

An operational function is an instruction delivered to the computer which does not require a program step. It is similar to a program step in that, with the exception of "start" and "trip", it has an entry and an exit, and the computer does not continue with the program until the function has been completed. Again with the exception of start and trip, it is possible to go into an operational function at any time during a program, and to go from one operational function to another.

A-2 Wiring of operational functions

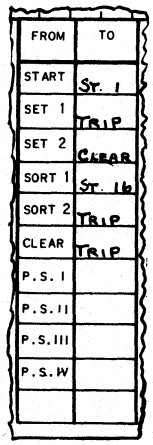
The operational functions, with the exception of start, are wired on the constant-program panel in lines 1-4, q-w. Start is located in line 5,A. Lines 2 and 4 are "in" hubs, while lines 1 and 3 are "out" hubs. The "in" hubs may be wired either from the branching of a step or the "out" of another operational function. The out of as many steps or operational functions as needed may be wired to the entry of one operational function through use of buses. However, the out of an operational function may be wired to only one place, except through use of selectors. As an example of wiring, assume the following:

From	To
Start	Step 1
Step 15, -Branching	Sort 1
Sort 1	Step 16
Step 28, +Branching	Set 1
Set 1	Trip
Step 35, -Branching	Sort 2
Sort 2	Trip
Step 38, +Branching	Set 2
Set 2	Clear
Clear	Trip



A-3 Program Charts

On program chart S1-1369, the operational functions are located on the bottom to the right of the output storage sections. When the "in" of an operational function is wired from the branching of a step, the name of the function is written in the box for the plus and/or minus branching of the step following which the operation is to take place. The "out" of the function is written in the box labeled "to" following the name of the function. The wiring shown above would be entered as follows:



B. START

Start is the impulse which begins a program. A pulse is automatically emitted from start when the card feed switch is raised if a card has entered the sensing station. Start is also internally wired from trip, so that when the program for a card has been completed, a pulse will be emitted from start to begin the program for the next card.

Start is usually wired to the first step of a program. As mentioned earlier, this is not necessarily step number 1, since the computer is nonsequential. In a multiple card routine all cards must go through the same first step (see page 137 for the single exception.) Start cannot be wired to the common of a selector picked up by card control in that card (see page 62), since sufficient time must be allowed for the selector to change from the non-select to the select position before using it. However, it is possible to wire through selectors from the branching of the first step; therefore, following the first step, each type of card may be directed to its own routine.

C. TRIP

All cards must be wired to trip. Upon this signal the card in the punching section is punched and ejected, and the card in the sensing section is sensed and moves to the punching section. A new card is fed from the feeding magazine to the sensing section. Trip is internally wired to start, so that the program will begin for the card which has just been sensed. As many steps or operational functions as necessary may be wired to trip. Note that trip is not listed with the other operational functions on the program chart, since it is possible to go only to start from trip.

D. SET 1 AND SET 2

D-1 Purpose

A set instruction transfers the values in the storages associated with set 1 or set 2 into the punch magnets in the punching code. It does not actually punch the holes, but merely sets the dies preparatory to punching, which takes place when the trip signal is given. When the card has been punched at trip time, all punching dies set from storage are cleared. The value will remain in the storage, and may be used in further computation. The columns into which the values will be punched are determined by the wiring on the input-output panel.

There are two set instructions - set 1 and set 2. Each instruction has separate storages associated with it, determined by the programmer. Any storage may be associated with either set 1 or set 2.

D-2 Use of two set instructions

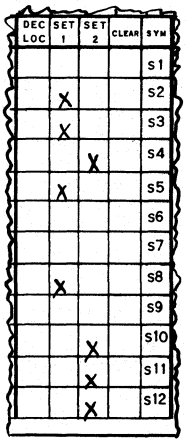
There are at least two ways of using the two set instructions. The most frequent is during a multiple card routine, where certain values are to be punched in detail cards, and others in summary cards. The storages from which the values are to be punched in the detail cards would be associated with set 1, while the summary values would be associated with set 2.

A second use of the two set instructions is during a program in which more storages are needed than the available 6 or 12. In such a case it is often possible to set values which have been completely computed in the punching dies part way through the program, and then use the storages from which they were set for computing other values. Note that when doing this, the storages may be used to compute the other values, but may not be used for punching the second values. They are available for use only as intermediate storage. It is inadvisable to go to the same set instruction twice without going to trip, except through use of selectors, since additional information would be set in the same dies. Therefore the second set instruction would be used to set the additional values computed after use of the first set instruction.

A set instruction does not clear the value from the storage unit from which the value is set. It is therefore possible to set a value from a storage unit and still use the value in further computation.

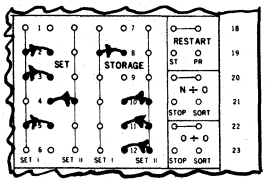
D-3 Program chart

The entry and exit of the set instructions are entered on the program chart as shown in section A of operational functions. In addition to this, set must also be associated with certain storages. This is done to the left of the operational section of the program chart. In the column beneath set 1, check the storages to be associated with set 1; in the column beneath set 2, check the storages to be associated with set 2. For example:

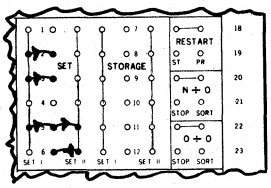


D-4 Wiring

On the constant-program panel the set instruction on which a particular storage is to be set is wired in lines 18-23, q-v. A wire is taken from the center hub, which bears the number of the storage, to the left for set 1 and to the right for set 2. The example given above would be wired as follows:



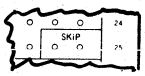
The same storage unit may not be set on both set 1 and set 2 without use of selectors (see page 78). To do so would make a connection which would set all storages associated with either set instruction on both set 1 and set 2. For example, suppose S2, S3 and S5 were wired to set 1, while S5 and S6 were wired to set 2. The connection panel would be wired as follows:



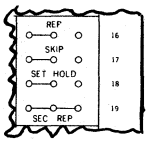
Because S5 connects set 1 and set 2, S2, S3, S5 and S6 will all be set on both set 1 and set 2.

D-5 Skip control

It is possible to set information in the punching dies, yet not punch anything in the card. This is done through use of a skip control. The skip control prevents all punching in the card, and the set information will clear. It must be controlled through a selector, or skipping would occur in all cards. On the constant-program panel the skip hubs are located in line 25, w-x.



To obtain the skip operation, it is merely necessary to connect the two hubs. This is done through the select side of a selector controlled by a program select (see page 66). There are also skip hubs located on the input-output panel, whereby the skip operation can be obtained through card control (line 17, v-x). The use of these hubs is covered in the section on reproducing, page 93.



D-6 Set hold

It is also possible to hold information which has been set in the card from one card to the next, through use of "set hold". This operation is usually used in conjunction with a skip instruction and with reproducing, although not necessarily. The hubs are located in the input-output panel in line 18, v-x, directly beneath the skip hubs (see above). Their use is covered in the reproduce section, page 95.

E. SORT 1 AND SORT 2

E-1 Purpose

When a sort instruction is delivered to the computer during a program, the card which is being computed at the time it is delivered will be sorted into the front receiving magazine instead of falling into the rear receiving magazine as it normally would. The sorting may be done for any desired reason, either based on type of card or something determined during the program. For example, it may be an advantage to have master cards or summary cards separated to avoid a later sorting operation. In a payroll program, it might be desirable to sort the card if a man has insufficient pay to cover his fixed deductions.

A sort instruction may be delivered to the computer at any time during a program, and it will be remembered until the card is fed into the receiving magazine by a trip instruction.

E-2 Use of two sort instructions

There are two sort instructions - sort 1 and sort 2. However, there is only one sort receiving magazine. Wiring to either sort 1 or sort 2 produces the

same result - the card will fall in the front receiving magazine. The purpose of two sort instructions is that there may be two reasons for sorting cards during the same program and it is necessary to go to two different steps following the sort operation. For example, assume a payroll program. On step 15 a minus branching means that the hours total from all of an employee's job tickets do not balance with the total from his attendance card. This card is to be sorted, and the program to continue with step 16.

On step 35 a minus branching means that there is insufficient pay to cover fixed deductions. This card is to be sorted, and tripped immediately. On the program chart this would be entered as shown in the example on page 51.

Note that if in both of these cases the card were to go to the same place, trip for instance, only one sort instruction would be needed. Both steps 15 and 35 could be wired to sort 1, and sort 1 would be wired to trip. However, it would be equally possible to use the two sort instructions.

F. CLEAR

F-1 Purpose

Clear is an instruction to return to zero the specific storages associated with clear. All other storage units will retain their values. The storages which are to be cleared are wired for each program.

As mentioned earlier, a storage unit will retain whatever value is put in it until a new value is placed there, or until an instruction to clear is delivered to it. For many programs clear instructions are not needed, since the entry of new information completely replaces the previous information.

F-2 Use

There are at least two specific instances when clear should be used. One is during the accumulation of information from detail cards to be punched in summary cards. Whenever this is done, the storage unit in which the accumulation is taking place must be cleared when the accumulated value has been set for punching. Without this clearing, the total from the preceding group of cards will be added to the next group.

There are also times when an entry may be made into a storage on one card, yet not made on another. In such cases there is a possibility of punching information from a preceding card into a following card. For example, assume that shift premium is computed in storage S5. It is possible to skip the steps involving computation of shift premium for first shift employees, so for these men S5 is not used. However, if a second shift man precedes a first shift man, the shift premium for the second shift man will remain in S5 and be punched into the card for the first shift man. By always clearing S5 at the completion of a program, this is avoided.

Of course, there are other ways of handling this problem. If shift is calculated by multiplying hours times shift rate, a shift rate of zero could be used for first shift men. The result of this multiplication is zero, which would clear S5. It is also possible, through use of selectors, to prevent setting S5 on first shift men.

F-3 Clearing step

Although the clear instruction is usually used to clear storages, it is possible to clear a storage by use of a step which will place zero in the storage. The usual step used for this purpose is:

V1	Pr	V2	=	R
Zero (N36)	+	Zero (N36)	=	Zero (S5)

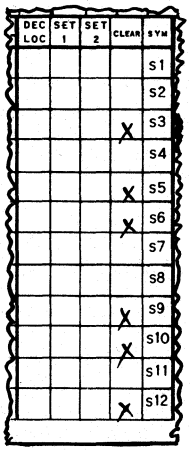
F-4 When to clear

A clear instruction clears storage units, but does not clear the punching dies associated with the storage. Therefore, if clearing is to be done at the end of a program it is usual to set information in punching dies (set 1 or set 2), then go to clear.

Note that it is also possible to go to clear at the beginning of a program, thereby clearing the information from the preceding card.

F-5 Program chart

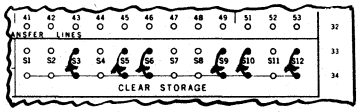
The entry and exit of the clear instruction are entered as shown in section A of operational functions, page 51. However, clear must also be associated with particular storages. To the left of the operational section of the program chart, in the storage section, is a column headed "Clear". In this column, check the storages which are to be cleared during a clear operation. The others, of course, will not be cleared. For example:



If different storages must be cleared on different cards it is possible to wire the storages through selectors (see page 79).

F-6 Wiring

The wiring to clear storage is located on the constant-program panel in lines 33-34, m-x. It is only necessary to wire from the hub for the storage unit to the clear hub beneath it. The wiring for the example given above would be:



G. PROGRAM SELECTS

Since program selects involve the use of selectors, they will be discussed on page 66, following the explanation of selectors.

H. N ÷ 0; 0 ÷ 0

H-1 Description

These two functions of the 60 and 120 are different from the operational functions previously discussed. They cannot be wired from the branching of a step or operational function, nor can they be wired to a step or operational function in the same way as is done with the other functions.

Arithmetically, when a number is divided by zero the result is infinity. To prevent the computer from attempting to compute infinity, if such a condition exists the 60 or 120 will automatically give zero as the answer. If zero is divided by zero, the answer is indeterminate. If this condition exists in a step, the computer will also give zero as the answer.

It may be important to the programmer to know that 0 ÷ 0 or N ÷ 0 has occurred. In fact 0 ÷ 0 is frequently used as an error indicating step. For example, if the computer determines that a card is out of sort, a 0 ÷ 0 step may be used arbitrarily to indicate the fact.

0 ÷ 0 and N ÷ 0 may be wired to stop or to sort, although they need not be wired to either - if they are not wired the program will continue with zero as the result of the step. In all cases either the 0 ÷ 0 or N ÷ 0 light will light.

If the condition is wired to stop, the computer will stop on the step on which it occurs. The test panel will indicate which step this is. If the condition is wired to sort, the program for the card on which it occurs will stop, the card will be tripped out and fall into the front receiving magazine, and the computer will begin the program for the next card.

H-2 Use

As mentioned earlier, 0 ÷ 0 or N ÷ 0, usually 0 ÷ 0, is frequently used as an error indicating step for errors such as a card out of sort, or failure to balance between a total accumulated from a group of detail cards and a predetermined total. The type of error referred to here is often discovered by the computer as the result of a testing step where one value is subtracted from another, and a minus branching indicates an error (see page 117). In such a case the minus branching would be wired to a 0 ÷ 0 step. Frequently, as a programmer begins a program, he designates the last step of the program as 0 ÷ 0. For example:

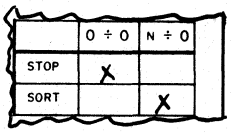
Step	V1	Pr	V2	=	R	Next Step
40	N36 (zero)	÷	N36 (zero)	=	S1 (error step)	

Note that a result storage must be indicated, although it is not necessary to wire the branching. 0 ÷ 0 is wired to stop the computer. In this case, whenever the error occurs for which the test is made, the computer will stop on step 40 and the 0 ÷ 0 light will light.

It is preferable to wire to a 0 ÷ 0 step to indicate an error of this type instead of not wiring the branching of the step which indicates the error, thereby causing the machine to hang up. The use of 0 ÷ 0 gives a positive means of identifying the reason for which the computer stops.

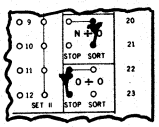
H-3 Program chart

On program chart S1-1369, 0 ÷ 0 and N ÷ 0 are located in the lower right corner. A check is placed in the proper box to indicate the desired result. Of course, if the program is to continue with no indication that the condition has existed, nothing is checked.



H-4 Wiring

On the constant-program panel, N ÷ 0 and 0 ÷ 0 are located in lines 20-23, w-x. The wiring for the above is:



The branching of a program step may not be wired directly to a 0 ÷ 0 or N ÷ 0 stop, but must be wired to a 0 ÷ 0 or N ÷ 0 step to stop the computer.

I RESTART

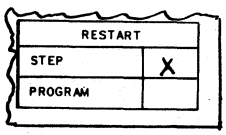
Restart is not an operational function in the same sense as those discussed earlier, since it has no entry or exit. However, it is a function which should be wired on every program.

As discussed on page 47, every program step goes through a proof of calculation and balances to zero before continuing to the next step. Restart is the function which directs the computer to repeat either the step or the program if this proof does not balance to zero.

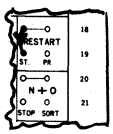
Step repeat is usually wired rather than program repeat. It must always be wired if the program involves accumulating, and almost always during a multiple card routine. With step repeat wired, the computer repeats the step which did not balance to zero, attempting to make it balance. With intermittent tube failure this may gradually slow a program, until eventually the computer will stop.

Although program repeat may be wired, there is no advantage to doing so. The program has already been checked as far as the particular step which did not check out, and repeating the entire program adds time to the checking operation. If the program involves accumulating, when the accumulating steps are repeated the values from the card in the computer are added into the total again, resulting in an incorrect total.

Restart is entered on program chart S1-1369 in the lower right corner.



On the constant-program panel, restart is wired in lines 18-19, w-x.
For example:



J. SUMMARY

1) Operational functions occur between program steps. With certain exceptions they have both an entry and an exit.

2) Start has only an exit, which is wired to the first step of the program. Start may not be wired to the common of a selector picked up by a control position in the same card. All cards must be wired from start.

3) Trip has only an entry, and all cards must be wired to it at the conclusion of their program. It is internally wired to start.

4) a. A set instruction sets the values in the storages associated with the set instruction in the punching dies, but does not punch the values.

b. There are two set instructions, set 1 and set 2, each of which may have any storages associated with it.

c. Setting information does not clear the storages from which it is set.

d. The same storage should not be set on both set 1 and set 2 without use of selectors, unless all of the same storages are to set on both set 1 and set 2.

e. Through use of a skip control, information set in punching dies may be prevented from punching. However, the punching dies will be cleared at trip time.

f. Through use of set hold, information set from storage may be held for punching in following cards, even though skip is used.

5) A sort instruction causes the card for which it is received to fall in the front receiving magazine rather than the rear receiving magazine. Although there are two instructions, sort 1 and sort 2, both cause the card to fall in the same pocket.

6) A clear instruction clears all storages associated with it, but no other storages. Clear does not clear values which have been previously set in punching dies associated with the storage.

7) Program selects are covered on page 66.

8) $N \div 0$ and $0 \div 0$ have no entry or exit. If either of these conditions occurs on a program step, zero will be given as the result and a corresponding light will light. Either $N \div 0$ or $0 \div 0$ may be wired to stop the computer, to sort the card, or the computer may continue the program with a zero result.

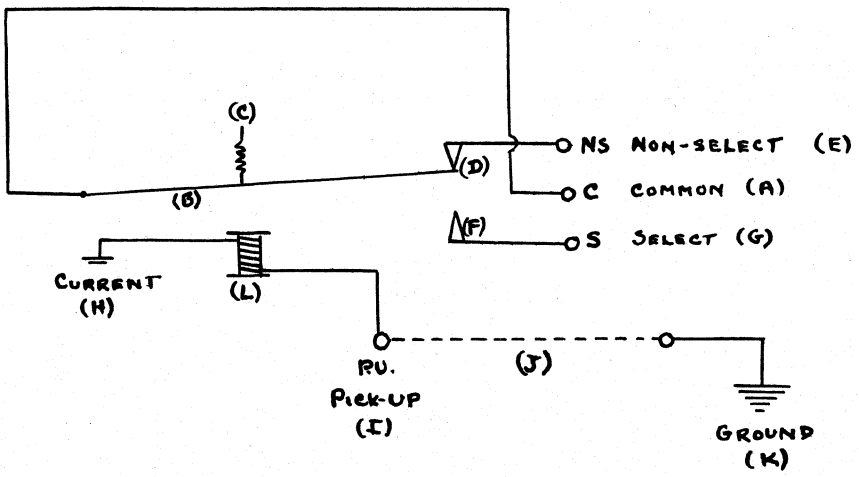
9) The computer may be wired to repeat the program or the step if a step does not check. Step repeat is usually wired.

6. Selectors

A. GENERAL

A selector is a two-way electrically operated switch which allows the programmer to route machine functions in two or more directions. The routing may be based upon the presence or absence of control positions punched in the card, or the plus or minus branching of a step. Any pulse in the computer can be routed through a selector, except the wiring of constant digits to the accumulator columns, and start to a selector picked up by a control hole in the same card. The maximum number of selectors available on the 60 or 120 is 36, referred to as T1 through T39 (there are no selectors numbered T10, T20, or T30).

A selector has no power of its own - it is merely a means of changing the direction of a pulse dependent upon a predetermined condition. In a schematic diagram, a selector may be shown as follows:



The pulse about which a choice is to be made is wired to the "common" of the selector. Normally, the pulse will come out of the "non-select" side of the selector. On the diagram, notice that a pulse entering the common (A) will flow through the bar (B) which is held in place by a spring (C) against a contact (D). Flowing through this contact, the current will emit from the non-select side of the selector (E). As long as nothing disturbs this arrangement, wiring into the common of a selector and out of the non-select side will have the same effect as wiring directly. For example, if the plus branching of step 1 is wired to the common, and the non-select is wired to step 2, as long as the contact at "D" remains unbroken the plus branching of step 1 will always go to step 2.

If, however, the bar (B) is pulled down to the other contact (F), any pulse entering the common will be emitted from the select side (G). In the above example this would be wired to another step, step 27 for example. Therefore, normally the plus branching of step 1 would go to step 2; due to a special circumstance, however, the plus branching of step 1 would go to step 27. Note that it is impossible to go to both step 2 and step 27 simultaneously, since the contact will be made either at point D or point F, but cannot possibly be made at both at the same time.

The means of bringing the bar into contact with point F is by allowing current (H) to reach ground (K). This is done by wiring (J) the pick-up of the selector (I) to ground on the connection panel. Whenever this circuit is complete, the coil (L) is magnetized, overpowering the spring and bringing the bar to point F. As long as the circuit is complete, the selector will remain on the select side. As soon as it is broken, however, the spring will pull the bar back to point D; any pulse entering the common will once more come out of the non-select side.

There are two types of selectors on the 60 and 120 - single-pole and four-pole. A pole refers to the common, non-select and select portion of a selector. There are 16 single-pole and 20 four-pole selectors. The diagram above is a single-pole selector. A four-pole selector differs from this in that there are four commons, four non-selects, and four selects. There is, however, only one pick-up.

The purpose of a four-pole selector is to enable the programmer to make more than one choice based upon the condition at the pick-up. When the coil (L) is magnetized, four bars instead of one are moved; each bar makes contact with its own F point. The circuitry for each pole of a four-pole selector is completely separate; the only relationship is that all four are moved by the same coil. Therefore, all four poles are always in the same position - either select or non-select.

Single-pole selectors are the even numbered selectors, T2, T4, T6, etc. Four-pole selectors are the odd numbered selectors, T1, T3, T5, etc. The individual poles within a four-pole selector are numbered 1, 2, 3, 4; therefore, a specific reference to a pole within selector T1 would be to T1-1, T1-2, T1-3, or T1-4. Any pole of a four-pole selector may be used for any choice; only as many poles need be wired as there are choices to be made.

Although almost any pulse in the 60 or 120 may be wired through the common, select, and non-select portion of a selecter, there are only three methods of controlling the pick-up of a selector - card control, program select, and selector hold.

B. CARD CONTROL

B-1 Pick-up of selectors

The pick-up of a selector may be wired through a control hole in the card. If this is done, as long as a card containing the control hole is being sensed, the selector which is being controlled will be on the select side. This means

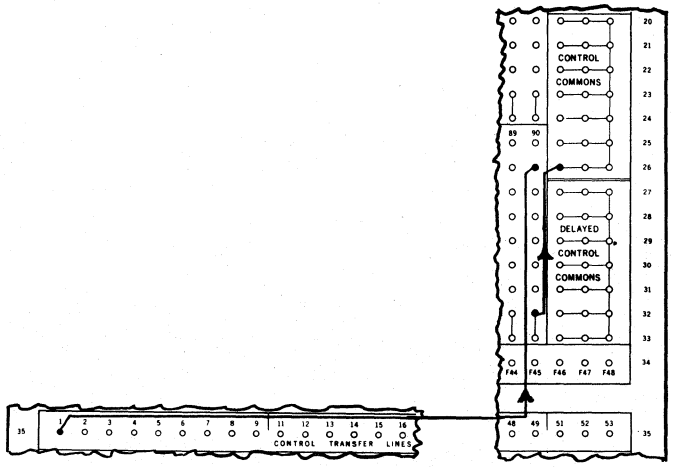
that from the branching of the first step for that card, allowing time for the bar to move from point D to point F, until the trip signal is delivered, any pulse put into the common will come out of the select side. At trip time, however, the connection from the pick-up of the selector through the control hole to ground is broken, since the control hole is no longer locked in the sensing section. The selector then reverts to the non-select side. If the following card does not have the same control hole, the selector remains non-select; if it does have the same hole, it is "picked up" - that is, returned to the select side.

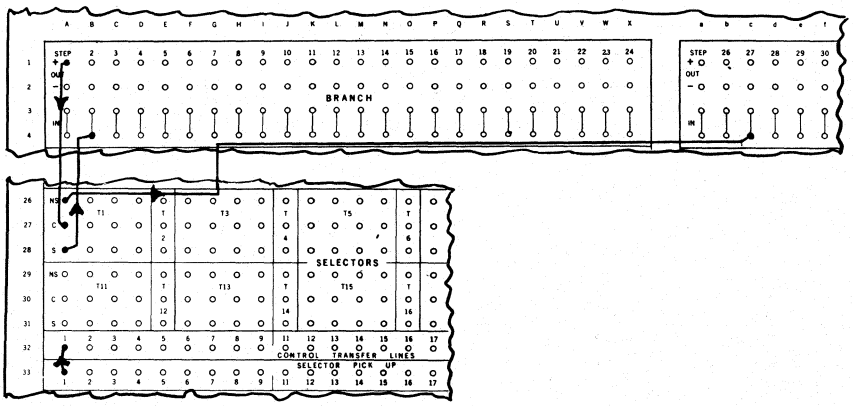
For example, suppose that all cards in a group of detail cards are punched with a 3 in column 90, and summary cards do not have a 3 in column 90. From a plus branching of step 1, detail cards are to continue with step 2, while summary cards continue with step 27. The pick-up of the selector - that is, the condition which determines whether the selector will be select or non-select - is a 3 in column 90. The common of the selector - the pulse about which a choice is to be made - is the plus branching of step 1. The word "common" refers to the fact the plus branching of step 1 is common to or related to both step 2 and step 27. The non-select side of the selector would be wired to step 27, since the selector is in this position when the 3 in 90 is not present. The select side of the selector would be wired to step 2. The presence of a 3 in column 90 will move the selector to the select side; this indicates a detail card, and detail cards will go from the plus branching of step 1 to step 2.

Selectors are wired on the constant-program connection panel. The control hole which is to be sensed is wired on the input-output connection panel. Therefore, a transfer line must be used to transfer the pick-up of the selector from one panel to the other. For this purpose, C-lines are usually used. Each control hole used for selector pick-up must be wired to a separate C-line. However, if one control hole is to pick up several selectors, the pick-ups of all the selectors involved may be bused to the one C-line on the constant-program panel. For ease of wiring, it is usual to use the same C-line number for transferring the pick-up as the number of the selector which it is to pick up - control transfer line C1 to pick up selector T1, C8 to pick up T8. However, since all C-lines are identical in circuitry, any C-line may pick up any selector.

B-2 Wiring

On the input-output panel in lines 20-26, v-x, are hubs called "control commons." These hubs are a direct connection to ground, although the connection is always broken momentarily at trip time. Any one of them may be wired to the common or zero common of the column in which the control hole to pick up a selector is located. The control hole itself is wired to a C-line. On the constant-program panel, the selectors are located in lines 26-31. The selector pick-ups are located in line 33, A-1, with C-lines directly above them in line 32. Using the example mentioned above, assuming the selector to be picked up by a 3 in column 90 is T1, the wiring would be: .

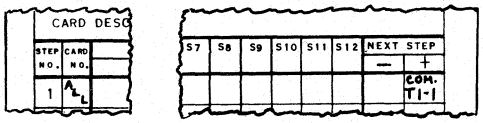




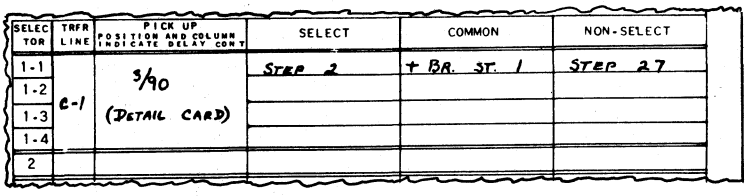
Whenever the 3 in column 90 is present, the current emitted from the pick-up can reach control common ground, and the selector will be on the select side. If the control hole is not present, the current cannot reach ground; therefore the circuit is incomplete and the selector will remain non-select.

B-3 Program charts

The program charts for the above would be filled in as follows. On the chart S1-1369, in the box for the plus branching of step 1, the common of T1-1 would be entered:



On the selector control chart, S1-1370, selector T1 is located at the top of the right column.



SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1	C-1	3/90 (DETAIL CARD)	STEP 2	+ BR. ST. 1	STEP 27
1-2					
1-3					
1-4					
2					

Selector: Indicated here are the selector numbers, with the poles of the four-pole selectors shown separately.

Transfer Line: Enter here the control transfer line used to transfer the control hole from the input-output panel to the constant-program panel. Note that since there is only one pick-up for a four-pole selector, only one C-line is used.

Pick-Up: Enter here the position in the column which is to pick up the selector. Although it is not necessary, many programmers also indicate the reason for which the selector is picked up; it simplifies finding the proper selector if additional choices are made later in the program based upon the same control hole.

Select, Common, Non-Select: In this section the choice to be made is written. Further examples of sample entries will be shown throughout the rest of this section.

C. PROGRAM SELECTS

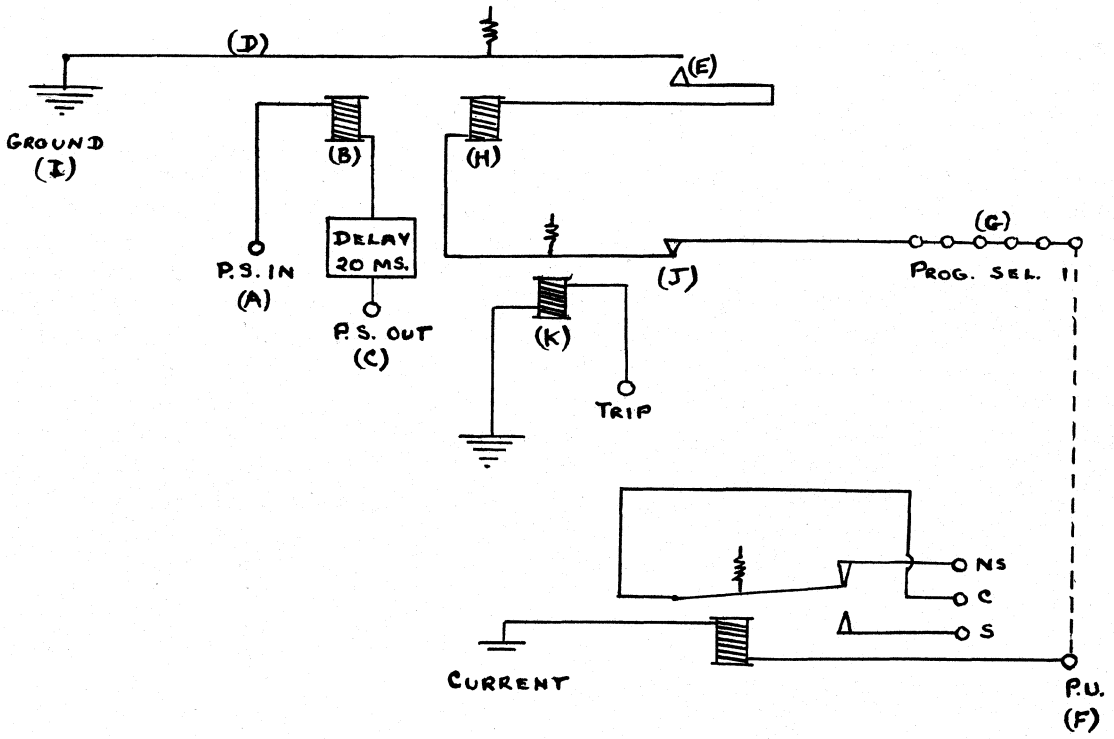
C-1 Purpose

As mentioned earlier there are times when a selector is needed or would facilitate programming, yet the condition requiring the selector cannot be precoded in the cards. In such cases, a program select is used. A program select is an operational function of the 60 and 120, having an "in" and an "out" in the same way as "clear", "set" and "sort". Its purpose is to provide a connection to ground to which the pick-up of a selector can be wired so that at a particular point in a program the corresponding selector will change from non-select to select.

A program select is usually impulsed at the plus or minus branching of a step. It may also be impulsed from the exit of any of the operational functions. The length of any of these pulses is very brief and a selector will remain select only as long as the circuit from its pick-up to ground is unbroken. A program select has the effect of lengthening these pulses until the end of the program, so that a selector will remain select from the time of the pulse which is to pick it up until trip time.

C-2 Internal wiring

There are actually three parts to a program select - "in", "out", and "ground". The "in" is wired from the branching of a step or from the out of another operational function. The "out" is wired to the entry of another step, or to another operational function to continue the program. The "ground" is wired to the pick-up of the selector which the program select is to control. As in the case of a selector controlled by card control, any desired pulse except constant digits to accumulator columns may be routed through a selector controlled by a program select.



When a pulse enters the "in" of a program select (A), current flows through a coil (B) for the duration of the 20 millisecond delay, magnetizing it. A pulse is then emitted from the "out" hub (C), which is wired to the next step so that the program can continue. Coil B is no longer magnetized. While coil B is magnetized during the 20 millisecond delay, bar D is pulled down and contacts point E. Current then can flow from the pick-up of the selector (F) through the program select ground hubs (G), magnetizing coil H as it reaches ground (I). This will continue until the connection to ground is broken at point J by the pulse which magnetizes coil K at trip time.

Whenever the "in" of a program select is wired, the "out" must also be wired or the circuit which originally pulls bar D to contact point E will not be completed, and the program select will not become effective.

The purpose of the 20 millisecond delay is to allow time for the selector associated with the program select to change from non-select to select.

If at any time after a program select has been impulsed, the connection between program select ground and the pick-up of the selector is broken, the program select is "dropped out". This means that coil (H) in the preceding diagram is no longer magnetized. The program select ground hubs cannot get through to ground even though the pick-up and ground hubs are connected again, until the program select "in" has been impulsed again. The corresponding selector, of course, returns to non-select.

C-3 Example of use

As an example of the use of a program select, suppose that to compute withholding tax on a payroll program, exempt income is subtracted from gross pay to determine taxable income in step 25. On step 26 taxable income is to be multiplied by 18%. It is possible that an employee's exemption would exceed his gross pay, so that he would have no tax. The storage unit in which withholding tax is to be placed has previously been used for an intermediate result during the program. Therefore instead of omitting step 26 for those employees with no tax, it is important to clear the storage in which withholding tax will be placed or the intermediate result will be used as the withholding tax. A simple way of doing this is to multiply taxable income by zero instead of 18% for these employees whose exempt income exceeds gross pay. During step 26, therefore, taxable income is multiplied by either 18% or zero, depending on whether the result of step 25 is plus or minus.

Assume the following elements and storages:

S7 - gross pay
S3 - exempt income
N32 - 18%
N36 - zero

Taxable income will be placed in S2, withholding tax in S1, and the choice of 18% or zero will be made through T8.

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
												4/3	3/2												
25		GROSS PAY	+	S7	−	EXEMPT INCOME	+	S3	TAXABLE INCOME	±	S2		25											P51	26
26		TAXABLE INCOME	±	S2	X	.18/ZERO	+	COM. T8	WITHHOLDING TAX (NR)	+	S1	26												−	27

FROM	TO
START	
SET 1	
SET 2	
SORT 1	
SORT 2	
CLEAR	
P.S. I	ST. 26
P.S. II	

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1					
7-4					
8	-	P.S. I	N 36	V2, STEP 26	N 32
9-1					

If step 25 has a minus branching, gross pay is less than the exemption. By going into a program select at this point, the computer will "remember" during step 26 that the branching of step 25 was minus. As soon as the PS 1 instruction is given to the computer, selector T8, which is associated with PS 1, will change to the select side. To continue the program, the "from" of PS 1 is wired to step 26, indicated in the operational function section of the program chart. The use of selector T8 occurs at V2 time of step 26, so in step 26 V2 is indicated as the common of T8. On the selector chart, T8 is indicated as follows:

Transfer Line: There is none, since program selects do not have to be transferred from one panel to another.

Pick-Up: This indicates that the ground of PS 1 is wired to selector T8.

Select, Common, Non-Select: V2 of step 26 is wired to the common of T8. If the branching of step 25 is minus, PS 1 is impulsed, which picks up T8. Therefore, N36, zero, will be used as V2 of step 26. If the branching of step 25 is plus, PS 1 is not impulsed and T8 remains on the non-select side. Therefore N32, 18%, will be used as V2 of step 26.

Note that in this particular case, instead of using a program select it would have been possible to use an additional step to accomplish the same purpose. In other words, step 27 could have been a separate clearing step. In this case, the three steps used would have been:

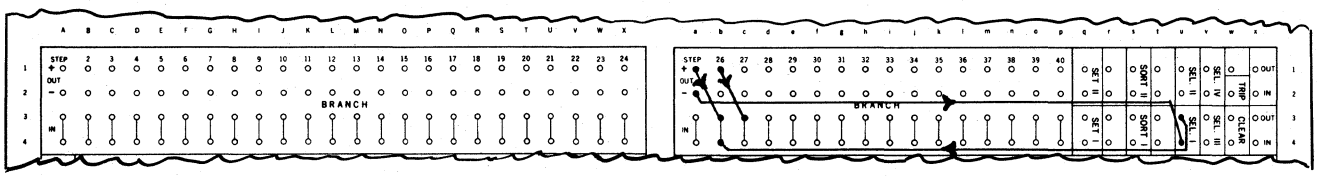
STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP —	NEXT STEP +
												4/3	3/2												
25		GROSS PAY	+	S7	-	EXEMPT INCOME	+	S3	TAXABLE INCOME	±	S2		25											27	26
26		TAXABLE INCOME	+	S2	X	.18	+	N32	WITHHOLDING TAX (NR)	+	S1	26												-	28
27		ZERO	+	N36	+	ZERO	+	N36	CLEAR S1 (NO W/TAX)	+	S1	27												-	28

The advantage of using a program select in this case is that it saves a step, which may be very important in a long routine. Its use does require 20 milliseconds of time, however, which the second method does not require.

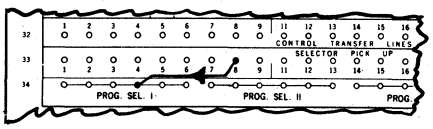
C-4 Connection panel wiring

There are four program selects, referred to as PS 1, PS 2, PS 3, PS 4. Therefore selectors may be picked up during a program for four different reasons. Each program select may be used to pick up as many selectors as needed - that is, if several choices must be made because of one condition which exists, a program select may be wired to the pick-up of several selectors. When a program select is impulsed, all selectors associated with it will become select. At trip time, all selectors associated with all program selects will become non-select. They will not become select until the program select with which they are associated is impulsed again.

On the constant-program panel the "in" and "out" hubs of the four program selects are located with the "in" and "out" hubs of the other operational functions in lines 1-4, q-w. The wiring of the above problem in this section would be:



The ground hubs for the program selects are located in line 34, A-X. The pick-up wiring for the above problem would be:



D. SELECTOR HOLD

D-1 Purpose

The selector hold is a means of retaining a selector on the select side from card to card. Without use of selector hold, a selector picked up by card control returns to non-select at trip time because the sensing switches are unlocked; a selector picked up by a program select returns to non-select because trip is internally wired to drop out program selects.

Selector hold is a direct connection to ground. Therefore if selector hold is wired directly to the pick-up, the selector will always be select. Selector hold should usually be wired through a controlling selector, which will determine when the connection will be made between the pick-up of the selector and selector hold.

D-2 Example of use

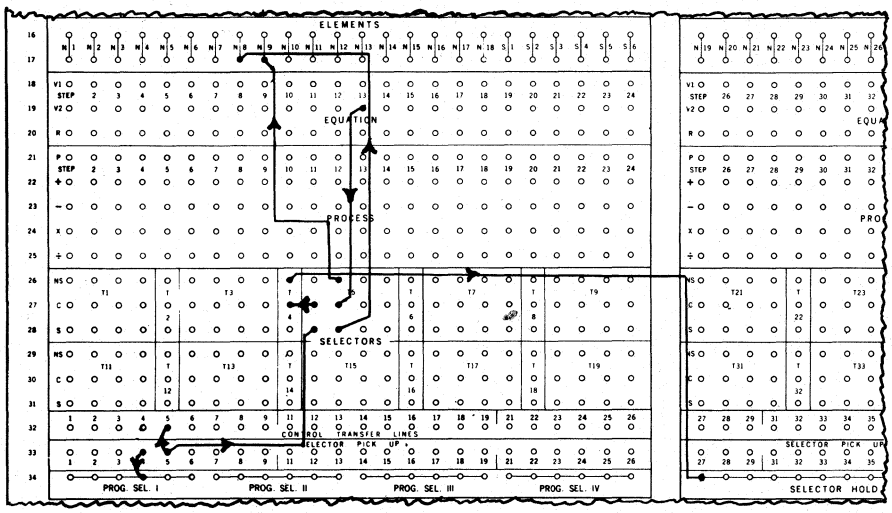
The use of selector hold therefore requires two selectors - a single-pole selector which is used as a controlling selector, and a four-pole selector through which the pulse which is to be changed is routed. As an example, assume an invoicing operation. The cards for each invoice are in sequence, with name and address cards first, followed by detail cards, followed by a summary card. The first name card contains a control hole indicating whether the customer is a retail store or a wholesaler - if a zero is punched in column 45, the customer is retail, otherwise he is wholesale. A retail price (N8) and a wholesale price (N9) are punched in each detail card. The problem is to select the proper price, dependent upon the control hole in the first card.

A selector hold is used in this case to remember whether a 0 in column 45 is present in the first name card. If it is present, and a selector is picked up because of it, that selector must return to non-select when the summary card enters the computer; unless this is done, the selector will remain select for the following customer, regardless of whether he is a retail customer.

D-3 Wiring

Assume that selector T4 is to be the controlling selector, T5 the one through which the selection of price is to be made. On step 13 price will be used as value 2. During the routine for the summary card, which has no control holes, PS 1 will be impulsed to indicate that the card in the computer is a summary card.

Selectors T4 and T5 would be wired as follows:



Selector hold can be connected with the pick-up of T5 only when T4 is non-select and T5 is select. At the beginning of a run, both T4 and T5 are non-select; therefore until a card with a 0 in 45 comes into the computer, V2 of step 13 will be N9, the wholesale price. When a 0/45 enters the sensing section, T5 becomes select, and selector hold is connected to the pick-up of T5. Even though the 0/45 card is tripped out of the computer, T5 remains select because

the selector hold is wired to the pick-up. For the following cards, therefore, N8, which is retail price, will be used as V2 of step 13. T5 will remain select until PS 1, which is wired to the pick-up of T4, causes T4 to become select. This, of course, will happen on a summary card. As soon as the connection between selector hold and the pick-up of T5 is broken, T5 becomes non-select. It will remain non-select until another 0/45 card picks it up. It would, of course, be possible to make two more choices dependent upon the 0/45 control hole by using poles 3 and 4 of selector T5.

D-4 Program chart

The selector chart for entering the above wiring is as follows:

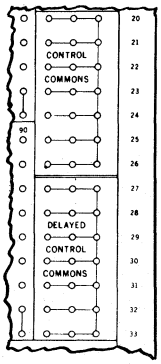
SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1					
1-2					
1-3					
1-4					
2					
3-1					
3-2					
3-3					
3-4					
4		PS1 (SUMM. CARD)	-	COM. T5-1	SELECTOR HOLD
5-1	C5	0/45 (RETAIL)	PU. T5	COM. T4	-
5-2		SEL. T 5-1 (SEL. HOLD)	N8	V2 STEP 13	N9
5-3					
5-4					

Both the controlling selector and the selector to be controlled may be picked up by either card control or a program select. However, the same program select cannot pick up both selectors, nor can a control hole picking up the selector to be controlled, T5, be punched in the same card with a control hole picking up the controlling selector, T4.

E. DELAYED PICK-UP

It is possible to delay the pick-up of a selector by a control hole until trip time of the card in which it is punched. By doing this, the selector will not become select for the program of the card being sensed, but will become select for succeeding cards. To make it effective, however, it must be wired in conjunction with selector hold.

On the input-output connection panel in lines 27-33, v-x, are hubs labeled "delayed control commons". These are direct connections to ground as are control commons. However, the connection is made only at trip time. As in the case of control commons, they are wired to the common or zero common of the column in which the control hole is located.



Selectors associated with delayed control commons become select at trip time, when delayed control commons become effective. However, since the sensing switches become unlocked at trip time, a means must be found of continuing the pulse. Even though delayed control commons remain effective until the sixth or seventh step of the following card, without the sensing of the control hole the selectors return to non-select. The normal means of doing this is through selector hold. The wiring is exactly the same as that shown in the section on the selector hold, except that column 45 would be wired to a delayed control common rather than a control common. The only difference in the program charts would be that the word "delay" would be written after the 0/45 for the pick up of T5.

F. ADDITIONAL EXAMPLES OF THE USE OF SELECTORS

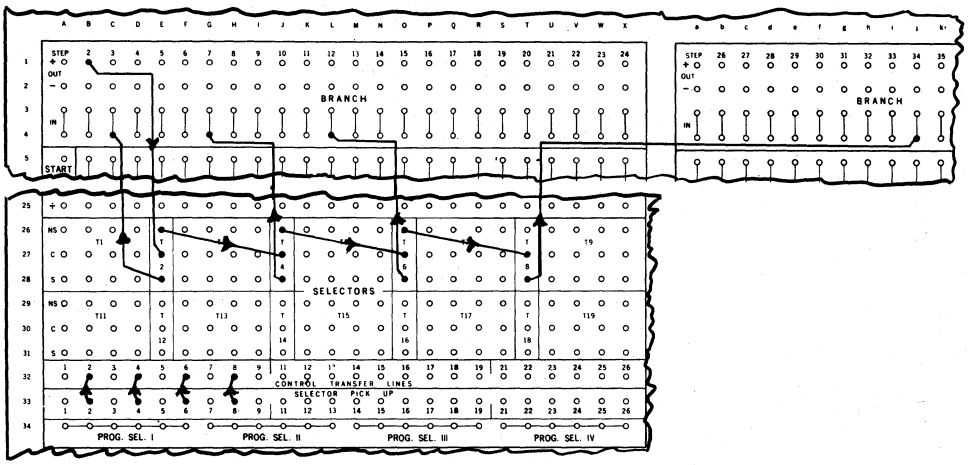
F-1 Differentiating between several codes

Frequently in a multiple card routine, it is necessary to pick up several different selectors, dependent upon the code punched in the card. For example, in a payroll program following the plus branching of step 2, cards are to be sent to different steps as follows, depending on the code punched in column 46.

Code	Type of Card	To Step No.	Selector No.
0	Rate	3	T2
1	Previous year to date	7	T4
3	Gross pay	12	T6
5	Deduction card	34	T8

To do a positive distribution to the proper step requires four selectors, with each selector picked up by one of the control holes. The plus branching of step 2 is wired to the common of the first selector. The selectors are wired in series, with the non-select side of each wired to the common of the next selector. The select side of each is wired to the step to which the program is to go if the control hole is present.

A diagram of the connection panel would look like this:



On the selector chart, the above would be indicated as follows:

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1					
1-2					
1-3					
1-4					
2	C2	0/46 (RATE)	STEP 3	+ BR. STEP 2	COM. T4
3-1					
3-2					
3-3					
3-4					
4	C4	1/46 (PREV. Y.T.D.)	STEP 7	N.S. T2	COM. T6
5-1					
5-2					
5-3					
5-4					
6	C6	3/46 (GROSS)	STEP 12	N.S. T4	COM. T8
7-1					
7-2					
7-3					
7-4					
8	C8	5/46 (DEDUCTION)	STEP 34	N.S. T6	

The plus branching of step 2 is indicated as the common of T2. The selector associated with the code punched in the card will be select, all others will be non-select. If, for example, the code punched in column 46 is 3, the plus branching of step 2 would flow as follows:

Plus branching of step 2 to common of T2
Common of T2 to non-select of T2
Non-select of T2 to common of T4
Common of T4 to non-select of T4
Non-select of T4 to common of T6
Common of T6 to select of T6
Select of T6 to step 12

Note that it would be possible to use only three selectors by wiring the non-select of T6 to step 34, assuming that if the code is not 0, 1 or 3 it must be 5. It is preferable, however, to use T8 and thereby make a positive check. If a card is not coded or a card punched 7 or 9 were to be placed in the deck by mistake, the plus branching of step 2 would end on the non-select side of T8. Since this is not wired, the computer will hang up and the card which is in error can be removed. Otherwise all errors will go to step 34.

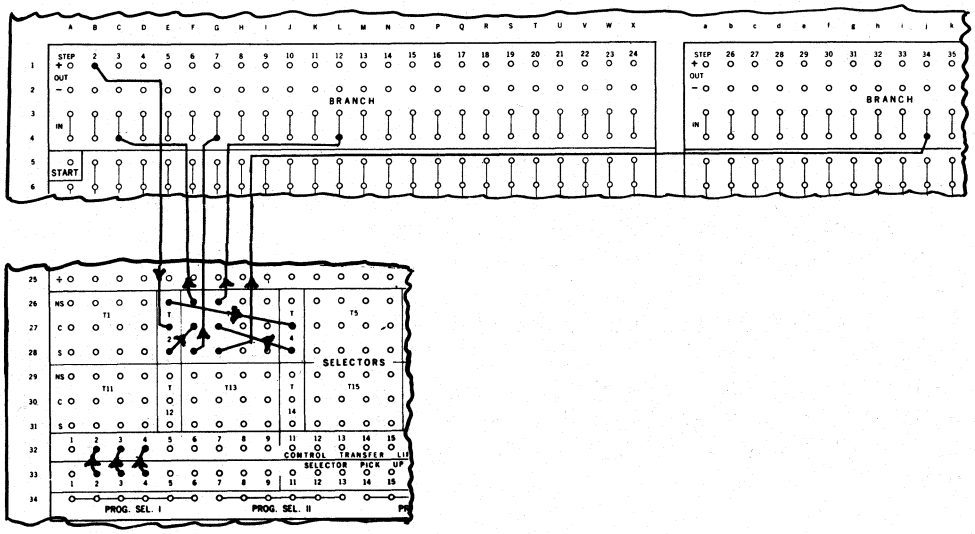
F-2 Differentiating between odd and even codes

In the Remington Rand punching code an even number is punched as an odd number plus the number 9. In the sensing section of the computer, punched positions which are to be used to control selectors are not decoded into their numeric value. Therefore a selector picked up by an odd number will also be picked up by the corresponding even number. For example, a selector picked up by a 1 will also be picked up by a 2. In order to differentiate between a 1 and a 2, therefore, it is necessary to test for a 9 as well as the 1. To do this the select side of the selector picked up by a 1 is wired to the common of a selector picked up by a 9. If the selector is non-select, the code is a 1; if it is select, the code is a 2.

Suppose that in the preceding example the card codes were as follows:

Codes	Type of Card	To Step No.
1	Rate	3
2	Previous year to date	7
3	Gross Pay	12
4	Deduction	34

A diagram of the connection panel would look like this:



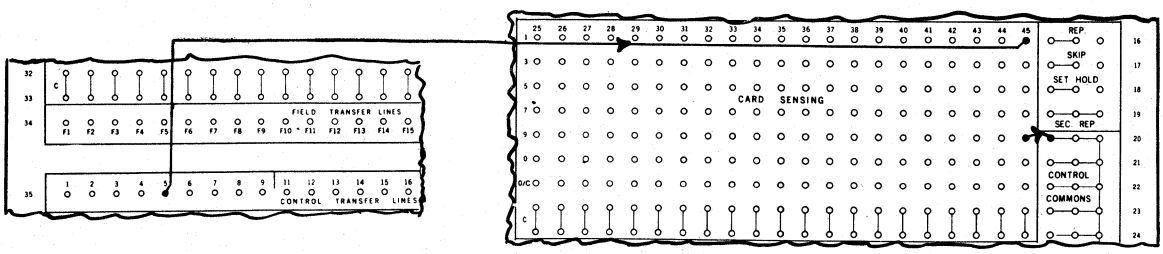
On the selector chart, the above would be entered as follows:

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT.	SELECT	COMMON	NON-SELECT
1-1					
1-2					
1-3					
1-4					
2	C2	1/46	COM T 3-1	+ BR. STEP 2	COM. T4
3-1			STEP 7	SEL. T2	STEP 3
3-2	C3	9/46	STEP 34	SEL. T4	STEP 12
3-3					
3-4					
4	C4	3/46	COM. T 3-2	N.S. T 2	-

In this example, if there is a 1 present, the plus branching of step 2 continues to T3-1 to differentiate between a 1 and a 2. If there is no 1, it continues to T4 to determine if there is a 3. If there is a 3, differentiation is made between a 3 and a 4 in T3-2. If neither a 1 nor a 3 is punched, the computer will hang up.

F-3 Pick-up of a selector by an even number.

Occasionally it is possible to save selectors by wiring two positions in a card to pick up one selector instead of wiring each position to a separate selector. For example, suppose that both a 1 and a 2 are punched in column 44, but only the 2 is to pick up selector T5. To accomplish this, the wiring would be:



Note that the common of column 44 is not wired. In order for the current which is emitted from the pick-up of selector T5 to reach control common ground, both the 1 and the 9 must be present. If either is punched without the other, the circuit will not be completed. If the common of the column were wired, the presence of either control hole would pick up the selector.

More than one selector can be picked up in this way from control punching in the same column. For example, with the "9" position wired to control common, the "1" could be wired through a C-line to pick up one selector while the "3" could be wired through another C-line to pick up a different selector. With this wiring a "2" and a "4" would pick up selectors, but the "1" and "3" would not.

This wiring is not confined to numeric combinations. If necessary, for example, the "5" could be wired to the control common and the "3" wired to the pick-up of a selector. The presence of both would cause the selector to become select, whereas either one alone would not.

F-4 Wiring more than one position in a column as minus.

It was mentioned on page 27 that more than one position in the same column may not be used as minus positions without the use of selectors. When wiring several positions in a column for negative control using selectors, each position is wired as the pick-up of a different selector, using normal wiring for this purpose. The element designators are then wired through the selectors to be minus.

Assume that column 89 has negative controls as follows:

Position	Element	Selector
1	N2.	T2
3	N5	T4
5	N3	T6

On the selector chart, this would be entered as follows:

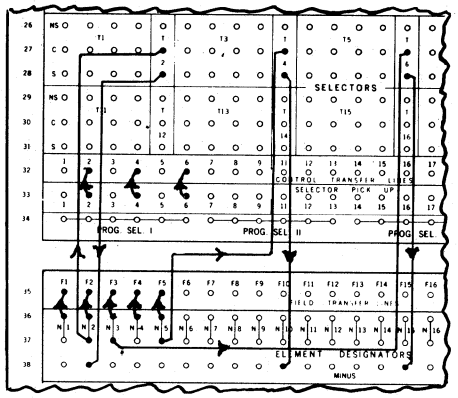
SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1					
1-2					
1-3					
1-4					
2	C2	1/89 (N2)	MINUS	ELEM. DES. N2	–
3-1					
3-2					
3-3					
3-4					
4	C4	3/89 (N5)	MINUS	ELEM. DES. N5	–
5-1					
5-2					
5-3					
5-4					
6	C6	5/89 (N3)	MINUS	ELEM. DES. N3	–

In the element section at the top of the same program chart, the negative controls would be indicated as:

CONSTANT FACTORS: I
CARD FACTORS: INDIC TRANS

ELEM. DESIG.	TO	NEG. CONT. COL.	NEG. CONT. POS.	NEG. CONT. TRF LINE	DEC LOC
N1	F1				
N2	F2	(COM. T2) 89	1	C2	
N3	F3	(COM. T6) 89	5	C6	
N4	F4				
N5	F5	(COM. T4) 89	3	C4	
N6					

The constant-program panel wiring for this would be:



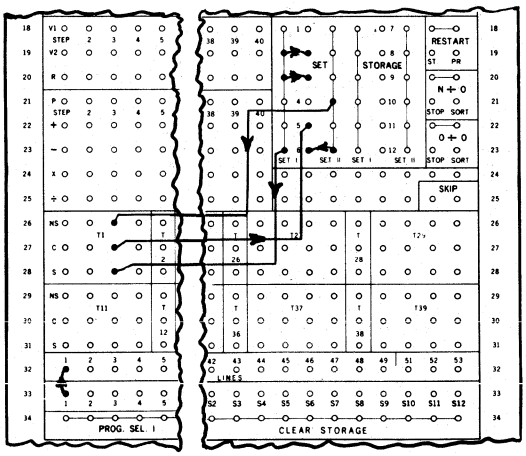
F-5 Setting the same storage on both set 1 and set 2.

As mentioned on page 54, it is impossible to set the same storage on both set 1 and set 2 without setting all other storages associated with set 1 and set 2 whenever either is called upon.

To avoid this, the storage which is to be set on both set instructions should be wired through a selector. In the example given on page 54, S2, S3 and S5 were to be set on set 1; S5 and S6 on set 2. Assuming that set 1 occurs in a card punched with a 7 in column 45, S5 could be routed through selector T1-3 as follows:

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1					
1-2	C1	7/45			
1-3			SET 1	SET S5	SET 2
1-4					

Therefore, on the connection panel the wiring would be:



With this wiring there is no time at which set 1 power and set 2 power are connected.

F-6 Using "clear" more than once.

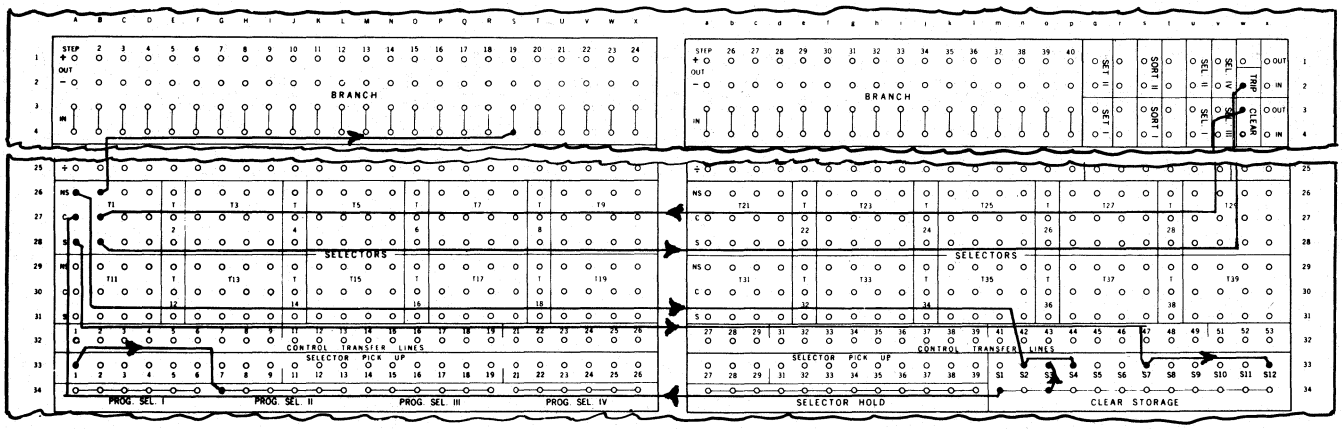
Occasionally it is necessary to use the operational function of clear more than once, clearing certain storages at one time and others at another time. Since there is only one clear instruction, a selector must be used to prevent clearing all storages which have been wired whenever a clear instruction is given.

Suppose that in one type of card it is necessary to clear S2, S3 and S4, while in another card S3, S7 and S12 must be cleared. Note that S3 is to be cleared in both cards. From the first clear instruction the program is to continue with step 19, while from the second clear the program goes to trip. The second type of card can be identified because PS 2 is impulsed during its program.

On the selector chart this would be shown as:

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1			S7, S12	CLEAR STORAGE	S2, S4
1-2		PS2	TRIP	FROM "CLEAR"	STEP 19
1-3					
1-4					

The connection panel wiring for this would be:



S3 will be cleared whenever clear is impulsed. S2 and S4 can be cleared only when the selector is non-select, while S7 and S12 can be cleared only when the selector is select.

F-7 Reusing steps through use of a program select.

Sometimes additional steps are needed for a program, more than the complement which is on the computer. Through use of a program select, it is possible to reuse program steps.

To do so, it is advisable to use program steps which are similar. For example, suppose that there are two sets of accumulating steps (see page 103). They are:

STEP NO.	CARD NO.	VALUE 1	SIGN	SYM	PROCESS	VALUE 2	SIGN	SYM	RESULT	SIGN	SYM	S1 4/3	S2 3/2	S3	S4	S5	S6	S7 3/2	S8 3/2	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
8		Σ HOURS	+	S7	+	HOURS	+	N9	NEW Σ HOURS	+	S1	8												−	9
9		NEW Σ HOURS	+	S1	+	ZERO	+	N36	NEW Σ HOURS	+	S7							(9)						−	10
10		Σ LABOR	+	S8	+	LABOR	+	S2	NEW Σ LABOR	+	S1	10												−	11
11		NEW Σ LABOR	+	S1	+	ZERO	+	N36	NEW Σ LABOR	+	S8								(11)					−	12

In combining steps 8 with 10, and 9 with 11, the following similarities would exist:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
8					+				NEW Σ	+	S1	8												−	9
9		NEW Σ	+	S1	+	ZERO	+	N36																−	

By impulsing a program select following the first time through step 9, the four points of difference (V1 and V2 of step 8, R and plus branching of step 9) can be routed through a selector. The first time through steps 8 and 9 the factors which were in the original steps 8 and 9 would be called upon; the second time through steps 8 and 9 the values from the original steps 10 and 11 would be called upon. Assuming that **PS** 1 picks up selector T5, this would be entered on the program and selector charts as follows:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2	S3	S4	S5	S6	S7 3/2	S8 3/2	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
8		Σ HOURS / LABOR	+	COM TS-1	+	HOURS / LABOR	+	COM TS-2	NEW Σ HOURS / LABOR	+	S1	8												−	9
9		NEW Σ HOURS / LABOR	+	S1	+	ZERO	+	N36	NEW Σ HOURS / LABOR	+	COM TS-3							(9)	(9)					−	COM TS-4

FROM	TO
START	
SET 1	
SET 2	
SORT 1	
SORT 2	
CLEAR	
P.S. I	STEP 8
P.S. II	
P.S. III	
P.S. IV	

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
5-1		PS1	S8	V1 STEP 8	S7
5-2			S2	V2 STEP 8	N4
5-3			S8	R STEP 9	S7
5-4			STEP 10	+ BR. STEP 9	PS1

The first time through steps 8 and 9 the values on the non-select side of T5 are called upon. From the plus branching of step 9 PS 1 is impulsed, which changes T5 to the select side. From PS 1 the program returns to step 8. This time the values on the select side are called upon, with the plus branching of step 9 going to step 10.

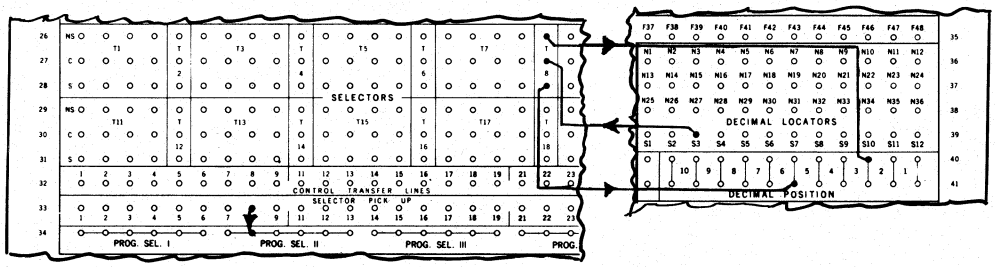
Since both the non-select and the select side of T5-1 and T5-3 are identical, it would be possible to wire both V1 of step 8 and R of step 9 to the common of T5-1, thus saving one pole of the selector. The selector would then read:

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
5-1			S8	V1, STEP 8 R, STEP 9	S7
5-2		PS1	S2	V2 STEP 8	N4
5-3			STEP 10	+ BR. STEP 9	PS1
5-4					

As many pulses as necessary may be wired to the common of a selector, provided the select and non-select sides are to be identical in all cases.

F-8 Changing the decimal location of storage

It is possible to change the decimal location of a storage unit during the program by impulsing a program select. This in turn picks up the selector through which the decimal location is wired. If the decimal location of S3 is to be 3/2 for steps 1-15, and 6/5 for the remaining steps, the program select would be impulsed at the end of step 15. Assuming that PS 2 picks up T8, the wiring would be:



There are two completely different purposes in changing the decimal location of a storage during the program. One is merely the necessity of a particular decimal location when the storage into which the result of a step will be placed has already been assigned a different decimal for use earlier in the problem. For example, a division to determine average rate might require a 5/4 decimal, yet all available storages have previously been assigned 4/3 or 3/2 decimal locations, and none can be changed.

The second use of the changing of a decimal location is to change the decimal location of a value in storage without the use of a step. When the decimal location is changed after a value has been placed in the storage, the value remains in the same storage columns although the decimal location changes. For example, if the number .123 is in a 4/3 storage, when the decimal is changed to 1/0, the value becomes 123. This has the effect of multiplying the number by 1000. If the decimal location were changed from 4/3 to 6/5, the value would become .00123, in effect dividing the value by 100.

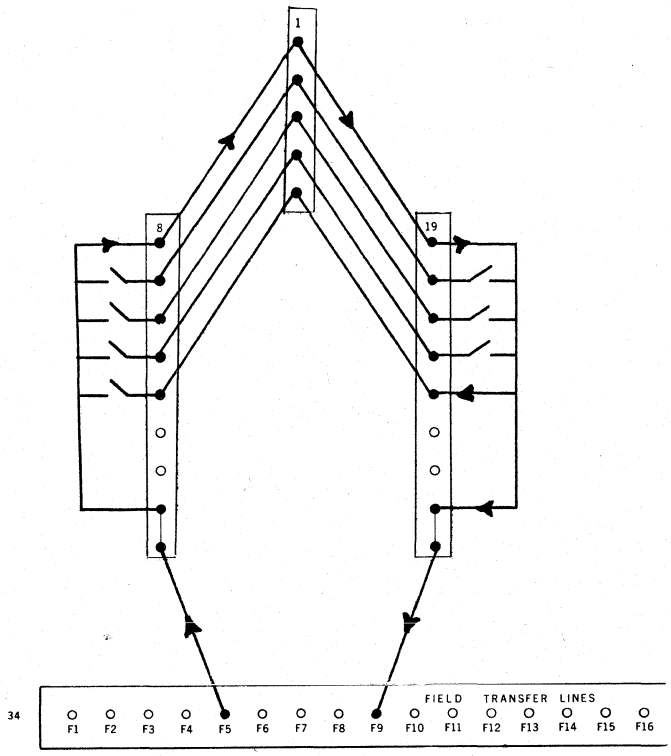
F-9 Y-wiring of accumulator input

On a Univac 60 it may be necessary to Y-wire more than one element to the same accumulator input columns due to the fact that there are only 60 columns of accumulator input available. To do this requires the use of selectors - one pole for each 9 position and one pole for the common of each column. The selectors which are being used may be picked up either by a program select or by card control.

Assume that element N5, punched in card columns 6-8, and element N9, punched in columns 17-19, are both to be wired to accumulator input A6, columns 3-1. If elements N5 and N9 are both punched in the same card, a program select would be used to control the selectors. All reference to one of the elements, N5 for example, should be completed before the other, N9, is called upon. When the last reference to N5 has been made, the program select would be impulsed to change the selector so that N9 may be called upon.

If N5 and N9 are punched in different cards, the use of a selector is unnecessary if columns 17-19 (N9) are blank when columns 6-8 (N5) are called on, and columns 6-8 are blank when columns 17-19 are called on. If this is not the case, however, selectors must be used.

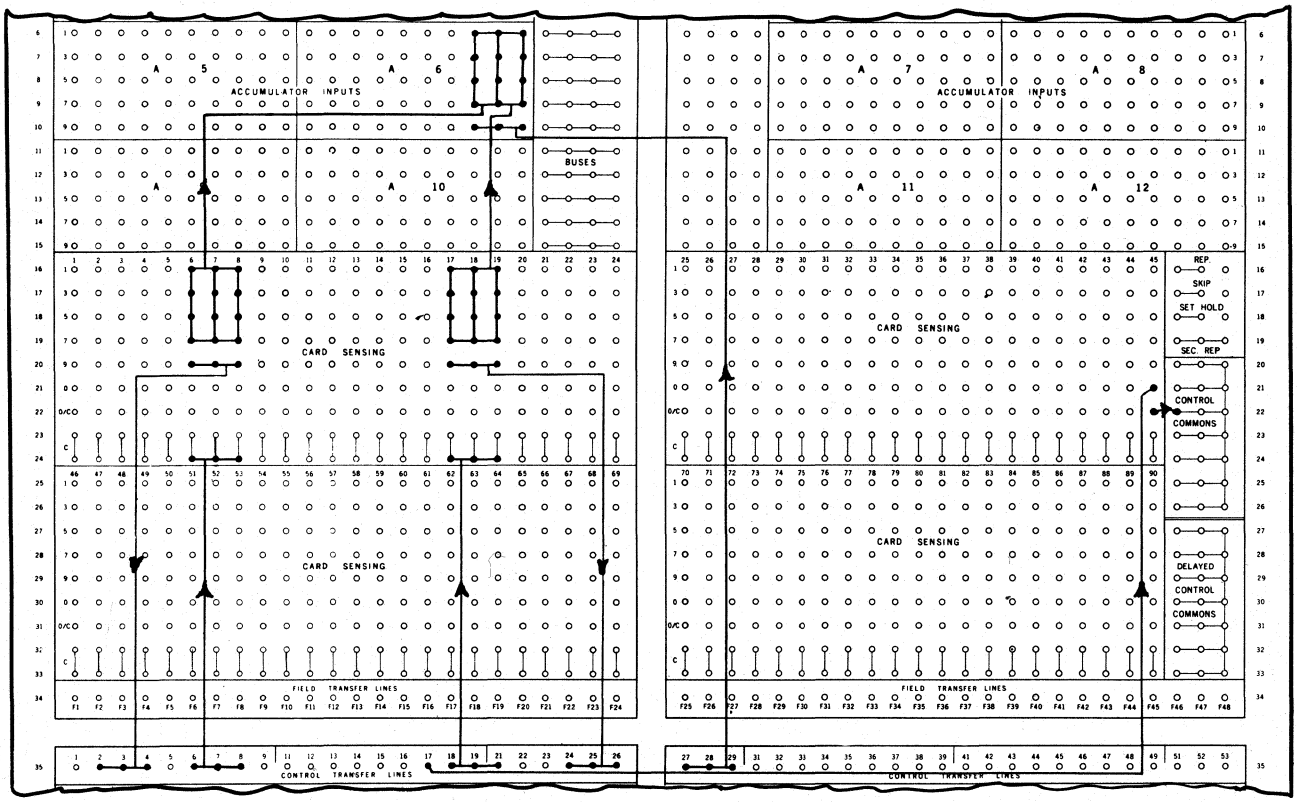
The reason for the selection of the 9 position and the commons is related to the sensing switch. The following sensing switch diagram shows what would happen if the 9 position is not selected, and column 8 is punched with a 1 while column 19 is punched with a 2.

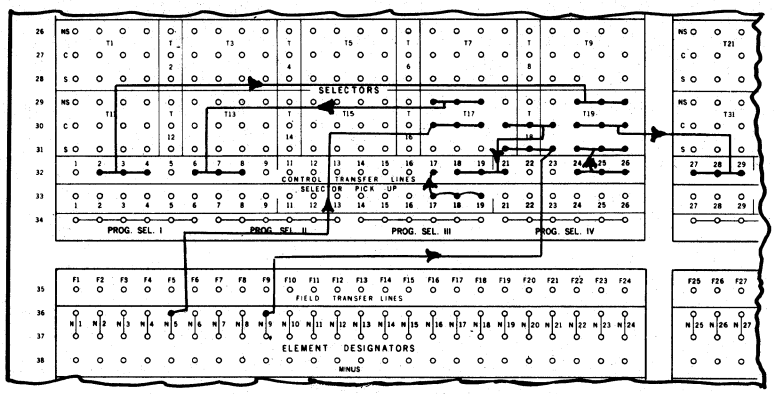


In the above example, power would emit from element designator N5 and be transferred on F5 to the common of column 8. Internally the power would flow from the common through the closed sensing switch of position 1, and into accumulator column 1, position 1. Because of the Y-wire it would continue to position 1 of column 19, across the closed sensing switch of position 1, through the closed sensing switch of the 9 position in column 19 and into accumulator column 1, position 9. Therefore, the 1 in column 8 would actually be sensed as a 2. Note that if only a 1 were punched in both columns there would be no difficulty in these two columns.

However, a back feed could develop through the common of the columns. If the commons associated with an element are wired together as they normally are, power would flow across the closed sensing switches where the same digit is punched as in columns 8 and 19 above, from one common into the common of the next column, thereby impulsing all columns associated with the element. Therefore, if only one position in the two fields is identical, all columns in both fields will be sensed simultaneously. To avoid this, the common of each column must be wired through a selector.

As an example of the necessary wiring, assume that a 0 is punched in column 45 when element N9, columns 17-19, is to be used; at other times N5, columns 6-8, is to be used. The wiring would be:





SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
17-1	C17	0/45		N5 ELEM. DES.	C6 (COM. COL. 6)
17-2				N5 ELEM. DES.	C7 (COM. COL. 7)
17-3				N5 ELEM. DES.	C8 (COM. COL. 8)
17-4			N9 ELEM. DES	C18 (COM. COL 17)	
18	C17	0/45	N9 ELEM. DES.	C19 (COM. COL. 18)	
19-1	C17	0/45	N9 ELEM. DES.	C21 (COM. COL. 19)	
19-2			C24 9/COL 17	C27 ACC NO. 9	C2 9/COL 6
19-3			C25 9/COL 18	C28 ACC NO. 9	C3 9/COL 7
19-4			C26 9/COL 19	C29 ACC NO. 9	C4 9/COL 8

Note that in the above example three selector poles could be saved by referring to both card fields by the same element number, N5 for example. In this case the N5 element designator would be wired to the common of three selector poles. The transfer lines for the commons of columns 6-8 would be wired to the non-select of these poles, and the transfer lines for the commons of columns 17-19 would be wired to the select of the same poles.

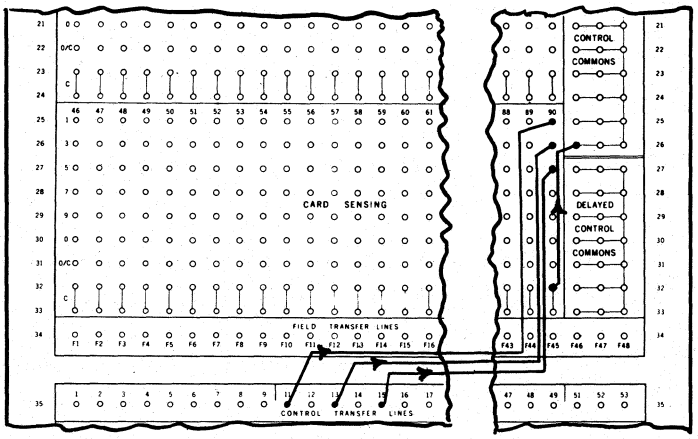
F-10 Wiring selector pick-up through constants

It is possible to use constant digits to act as a one-way bus in selector pick-up. The purpose of this wiring is to pick up one selector from several control holes, while at the same time to pick up a separate selector for each control hole. In many cases this will save several selector poles, and may permit the use of single-pole selectors instead of four-pole selectors.

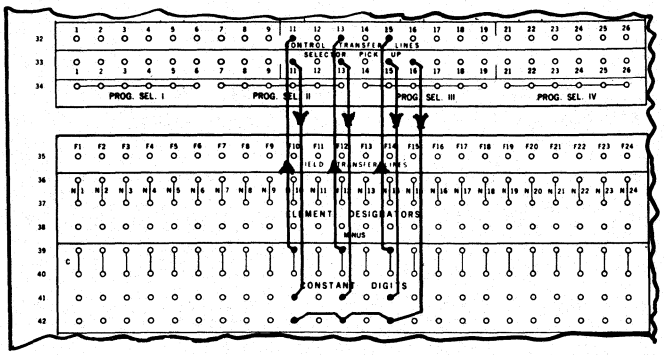
For example, assume a multiple card routine in which there are cards punched with a 1, 3 or 5 in column 90, as well as blanks. As of the plus branching of step 5, all cards with a 1, 3 or 5 in column 90 are to go to step 6, while cards without punching are to go to step 20. Suppose that a 1 picks up T11, a 3 picks up T13, and a 5 picks up T15. The plus branching of step 5 could be wired to the common of T11-2, the non-select of T11-2 to the common of T13-2, and the non-select of T13-2 to the common of T15-2. The select of T11-2, T13-2, and T15-2 would all be wired to step 6, while the non-select of T15-2 would be wired to step 20. However, with the wiring shown below, this choice could be made in one pole of another selector, while T11, T13, and T15 are used for other choices.

To accomplish this, the wiring on the input-output panel is normal. The control holes are wired to control transfer lines as follows:

Control Hole	Transfer Line
1	C11
3	C13
5	C15



On the constant-program panel each control transfer line is to pick up its corresponding selector, as well as selector T16. If this is wired through a normal bus, a back feed will develop whereby the presence of any control hole will pick up all four selectors. To avoid this, each control transfer line is wired to a constant digit. One of the hubs beneath the constant digit is wired to pick up the related selector, while the other is wired to pick up T16. Since a back feed cannot develop through the constant digits, a 1 in column 90 will pick up only T11 and T16, a 3 in 90 will pick up T13 and T16, and a 5 in 90 will pick up T15 and T16. The wiring is as follows:



In the above example, the plus branching of step 5 would be wired to the common of T16. The select of T16 would be wired to step 6, while the non-select would be wired to step 20.

G. SUMMARY

1) A selector is a two way switch permitting the rerouting of almost any pulse on the 60 or 120.

2) The pulse to be changed is wired to the common of the selector. The common is internally connected to the non-select side unless the selector is picked up. When it is picked up, the common is internally connected to the select side.

3) A selector may be picked up by card control, a program select, or selector hold.

4) Unless selector hold is wired, a selector picked up either by card control or a program select will return to non-select at trip time.

5) The maximum number of selectors available on the 60 or 120 is 36, 16 single-pole and 20 4-pole selectors.

6) There are four program selects on the 60 or 120. They are used when a selector is to change from non-select to select due to a factor discovered during the program rather than a condition which can be precoded in a card.

7) A program select is impulsed only from the plus or minus branching of a step, or from the exit of an operational function. However, the selector associated with it can be used to reroute almost any pulse on the computer.

8) Selector hold is a means of keeping a selector select from card to card. It should be wired through a controlling selector.

9) The pick-up of the selector which controls selector hold must be in a different card from the pick-up of the selector which is being controlled.

10) The card controlled pick-up of a selector may be delayed so that the selector will not be picked up until the following card.

11) Several selectors may be associated in a "chain" so that one pulse may be routed to one of several places instead of only two, as is the case when only one selector is used.

12) Program steps may be reused through use of program selects.

7. Reproduce

A. GENERAL

Reproduce is a function of the 120 which allows the transposition of information within a card, or the reproduction of information from one card to another. Since reproduce does not involve the use of the calculating section of the 60 or 120, alphabetic as well as numeric information may be reproduced. The reproduction may be within the same card - that is, transposing from one group of card columns to another group; it may also be from one card into succeeding cards, either in the same card columns or different columns. Reproducing may be done at the same time that other values are being calculated and punched.

However, fields to be reproduced may not be Y-wired for reproducing and input for calculating, nor may output storages be wired into the same columns into which information is being reproduced, even though the output punching and reproducing occurs in different cards.

There is a function called secondary reproduce. This permits the picking up of information for reproducing from an additional card without the clearing of the first information picked up.

Reproduction is done on a position for position basis. This means that it is possible to reproduce any position as any other position. It is more usual, of course, to reproduce a position as the same position in another column. The basic wiring for reproducing is done on the input-output panel. However, if reproducing is to be done in a separate operation, with no calculation necessary, it is still necessary to use a constant-program panel. This is due to the fact that start and trip are on this panel. The only wiring necessary would be a connection between start and trip.

B. CARD COLUMN WIRING

The columns which are to be reproduced are wired from the card sensing section (lines 16-21, 25-30, A-u) to the columns in which they are to punch (lines 36-41, 45-50, A-u). The zeros may or may not be wired, according to the application; however, they must be wired if the information is alphabetic.

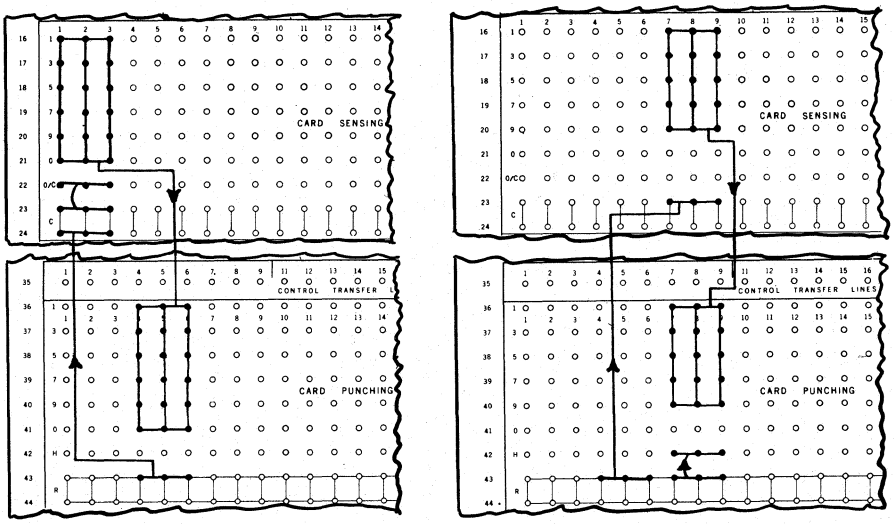
There are certain restrictions as to the number of card columns which may be reproduced. In checking to see whether a particular application exceeds the limits, multiply the number of numeric columns to be reproduced by 2.5 and the number of alphabetic columns to be reproduced by 3.5. Multiply the number of columns to be punched from storage by 1.5. The sum of the results of these three multiplications may not exceed 200.

As is the case when using the card sensing columns as elements, the commons of the columns to be reproduced (lines 22-24, 31-33, A-u) must be impulsed. This is done by wiring reproduce hubs (lines 43-44, 52-53, A-u) to the commons. Since reproduce hubs are connected to each other, it makes no difference which hubs are wired to the commons. It is important to remember that if zero is to be reproduced, the corresponding zero common hub must also be connected with reproduce.

It is possible to take a single wire from an R hub to the common of one of the columns to be reproduced, and then jack the other commons in the same way that they are jacked when an F-line is wired to one of the commons (page 24). However, no more than 10 commons may be associated with one wire from the R hubs; for each additional 10 columns another wire must be brought up from an R hub.

If the reproduction is to be transposition within a card, no additional connections are needed in these two sections. If, however, the information is to be reproduced from one card into following cards, connections must be made to prevent the clearing of the punching dies until the next card from which information is to be picked up. This is done by wiring the H or "hold" hubs beneath the columns into which the information is to punch (lines 42 and 51, A-u) to the reproduce hubs.

Two examples are shown below. The first transposes columns 1-3 to 4-6 in the same card -- therefore the hold is not wired. The second shows the reproduction of columns 7-9 into columns 7-9 of following cards:



The above wiring would be shown on the selector program chart beneath the element section.

FROM COLUMN	1	2	3	
TO COLUMN	4	5	6	
CONTROL	R	R	R	

FROM COLUMN	7	8	9	
TO COLUMN	7	8	9	
CONTROL	RH	RH	RH	

The R under "control" indicates that the columns are to be reproduced as opposed to secondary reproduce. The H following the R indicates that the hold hubs are to be wired.

Y-wiring is possible when reproducing. A card sensing column may be reproduced into as many punching columns as necessary, without restriction. Reproducing from two or more sensing columns into one punching column is possible if all the columns are identical, or if all except one are blank.

Through use of selectors it is possible to reproduce into the same columns one or the other of two fields, both of which are punched in the same card. If the information is numeric, the same rules apply as in the Y-wiring of input to the accumulator input columns (page 82); that is, each 9 position and the common of each sensing column must be wired through a selector. If the information is alphabetic, the 0, 1, 3, 5, 7, and 9 positions in each column as well as each common and zero common must be routed through selectors.

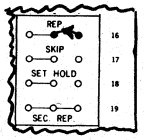
C. CONTROL OF REPRODUCE

C-1 Reproducing into the same card.

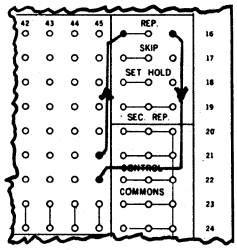
Reproducing may occur automatically, so that every card will reproduce, or it may be controlled through a control hole. The automatic reproduction of every card would be used mainly for transposition punching within the same card, while punching into a following card would involve some type of control.

When reproducing, any information previously set up is cleared from the punching dies at the beginning of the card cycle on which calculation takes place. Following the clearing, the new information to be reproduced is set into the punching dies. For transposition reproducing, the setting of new information may be either automatic in every card or controlled by a control hole. The control hole will permit the setting of new reproduce information only in those cards containing the control hole. However, by wiring through a selector, it is possible to reproduce in all cards except those containing the control punching. Examples of each type follow.

On the input-output connection panel, the control of reproducing is located in line 16, v-x. To obtain automatic reproduction, the only connection required is:

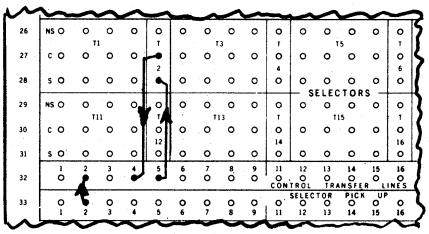


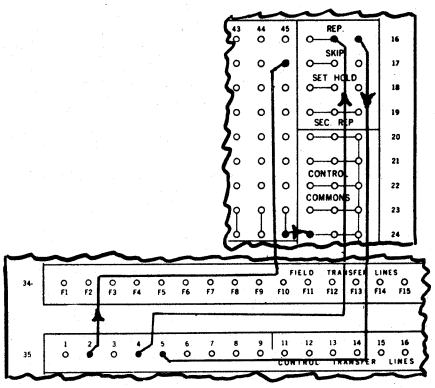
To reproduce whenever a control hole is present, either the single hub or the two joined hubs may be wired to the common of the column in which the control hole is located. The control position is wired to either the single or double hub, whichever is not wired to the common. Although it makes no difference whether the common is wired to the single or double hub, for ease of explanation it will be assumed that the single hub is always wired to the common. The following example shows the use of a 0 in 45 to cause reproducing.



If a position other than zero is used as a reproduce control hole, no other positions in that column may be used for any other type of control, with the exception of certain additional reproduce functions listed in the following paragraphs. The reason for this is that the reproduce hub wired into the common of the column carries a different voltage from that used when a control common or element designator (through an F-line) is wired to the common; attempting to use the other positions for other purposes will result in blown fuses.

It is possible, however, to wire the reproduce control through a selector, thereby wiring the common of the column to a control common so that the other positions may be used to pick up other selectors. Assuming that a 3 in 45 is to pick up selector T2, the wiring would be as follows:

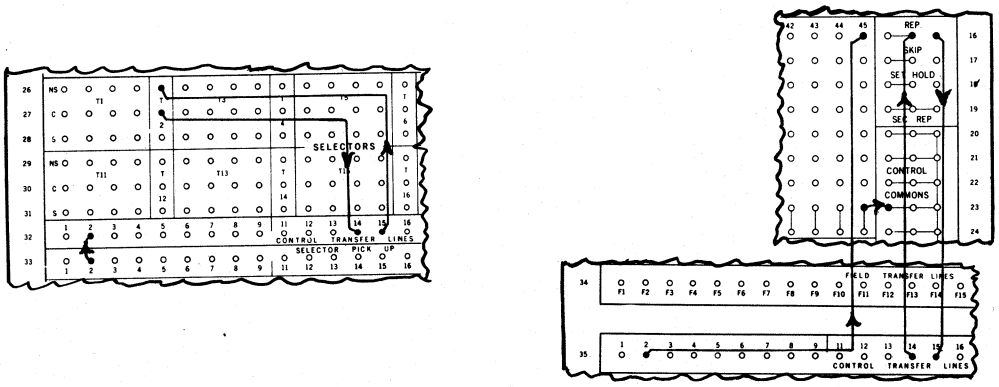




On the selector chart this would be entered:

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1					
1-2					
1-3					
1-4					
2	C2	3/45 (Reproduce)	C5 (Rep)	C4 (Rep)	—

To reproduce except when a control hole is present, the reproduce function must be wired through a selector. Using a 1 in 45 to pick up selector T2, the wiring would be as follows.



SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1					
1-2					
1-3					
1-4					
2	C2	1/45	—	C14 (REP)	C15 (REP)

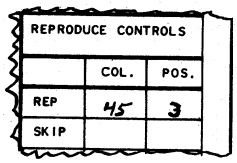
In this example the 1 in 45 is used to pick up selector T2 with normal selector pick-up wiring. A reproduce hub is wired through a control transfer line to the common of T2, and from the non-select of T2 through a control transfer line to the other reproduce hub. As long as selector T2 is not picked up, the wiring is effectively the same as that shown in the first example - the two reproduce hubs are connected. However, when selector T2 is picked up the connection is broken, and reproducing will not take place.

On program chart S1-1369 the reproduce controls are located in the lower right corner. The above three examples would be indicated here as follows:

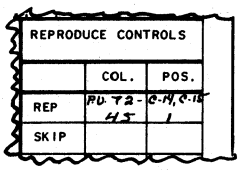
1. Automatic

REPRODUCE CONTROLS		
	COL.	POS.
REP	AUTO	
SKIP		

2. Reproduce when control hole is present



3. Reproduce except when control hole is present



C-2 Reproducing into following cards

Reproducing into following cards will always involve the use of control, usually a position in a card although a program select may be used. When a program select is used to control the reproduce operation, the program select must be impulsed no later than the branching of the second step of the program. If it is impulsed any later, the previous information will not clear.

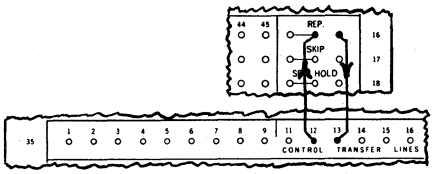
In reproducing into following cards, the H (hold) hubs in lines 42 and 51, A-u, must be wired in those columns in which the reproduce information is to be held from card to card. The wiring of these hubs prevents the clearing of these columns at the beginning of the calculating cycle (see page 89) unless the reproduce hubs in line 16, v-x, are connected, either through a control hole or a selector. When these hubs are connected, the following occurs:

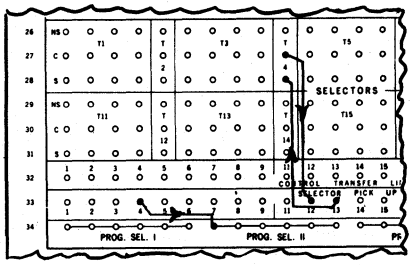
1. Clearing of punching dies wired for reproducing.

2. Set up of new reproducing information from the card being sensed. This new information will be punched into the card being sensed and all following cards until reproduce is impulsed once more. With automatic reproduction, of course, the reproduce hubs are impulsed on every card; therefore information cannot be held from card to card.

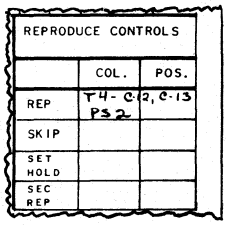
The wiring of a control position to impulse the reproduce function for following cards is exactly the same as that given in example 2 above. The only additional wiring needed is the wiring of the H (hold) hubs to the columns into which reproducing will take place.

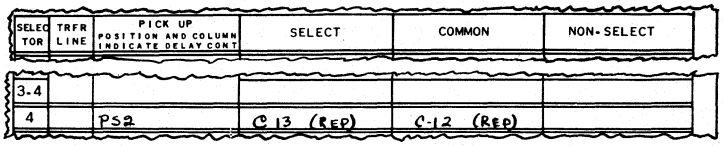
The wiring of a program select to impulse reproducing is as follows: Assume that whenever PS 2 is impulsed at the minus branching of step 1, the reproduce function is also to be impulsed.





On the program charts this would be indicated as:





Note that the above wiring could also be used for transposition reproducing.

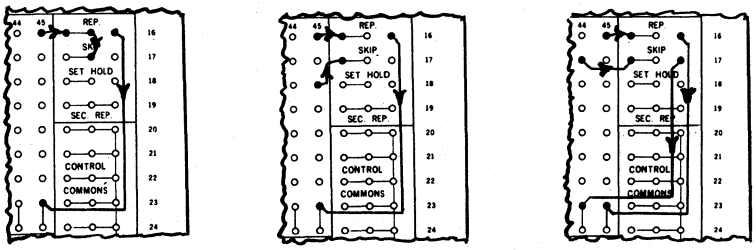
D. SKIP CONTROL

There are many times when it is desirable not to punch in the card from which reproduce information is picked up. Even though the reproducing is to be in identical columns of following cards, it is preferable not to overpunch the original card because of the possibility of "half-moon chips". These are caused by minute differences in the positioning of the card in the punching section when originally punched and when overpunched. If the punching in the following cards is to be in different columns from the original card, it is usually preferable not to punch the information a second time in the original card, particularly if the fields are overlapping. Therefore, a "skip" control may be wired in conjunction with a reproduce control.

When the skip control is wired, it prevents all punching in the card in which the control is sensed -- not only reproduce punching, but also output punching from storages. It is usually controlled by a control hole in the card, though it may be controlled by a program select. It may be controlled from the same control hole as that governing the reproduce operation; in this case skip will occur in the card from which new reproduce information is picked up. However it may also be operated from a separate control hole so that skipping would occur in cards other than these causing reproducing. This would be particularly useful in a billing application where all the cards used in writing an invoice are placed in the 60 or 120 by customer number in sequence for the invoice writing. The order number could be picked up for reproducing from the first card for each customer, a card with information about the order. Punching would be skipped in the following name and address cards, yet the order number would be punched in all detail cards.

Although skip cycle prevents all punching, both reproduce and that set from storage, it does not clear reproduce information. Reproduce information will be punched in any following card which does not contain a skip control until it is cleared by a reproduce control. However, any values set from storage in a card containing a skip control will be cleared, unless set hold is also wired (see below).

The wiring of skip is done on the input-output panel on the line beneath the wiring for reproducing. Three examples follow:



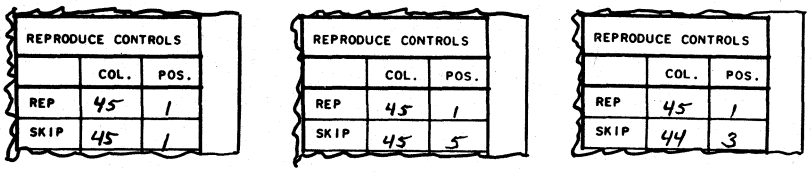
1. This shows the wiring of the skip control from the same position as that used for reproducing.

2. This shows the wiring of the skip control from a separate position in the same column as the reproduce control.

3. This shows the wiring of the skip control from a different column from that used for the reproduce control.

Notice that the hub to the right of the double skip hub is used when skip control is in a separate column. It could also be used to impulse a zero common if the zero is used for skip while one of the other digits in the same column is used for reproducing. If this hub is wired to the common of a column, no position in that column may be used for selector pick-up or any control function other than those associated with reproduce.

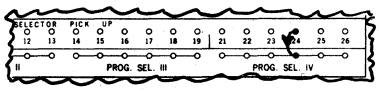
The above three examples would be indicated in the program chart as follows:

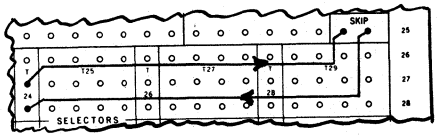


The skip control is not necessarily associated with reproduce. By wiring the skip control as in example 3, with no reproduce wiring at all, whenever a 3 in 44 is punched in the card no punching at all will occur. This is true since skip always eliminates all punching, both reproduce and output from storage. If information is set into the punching dies from storage on a card with a skip control, at trip time these dies will be cleared without punching.

Skip may be controlled by a program select as well as by card control. In this case the wiring may all be done on the constant-program panel. On the constant-program panel in line 25, w-x, are two hubs labeled "skip". These correspond to the double hub and the single hub labeled "skip" on the input-output panel. These hubs must be wired through a selector, for if they are wired together directly, skipping will occur in all cards.

Assume that if PS 4 is impulsed, no punching is to occur in that card. PS 4 will pick up T24. The wiring would be:





On the program charts this would be shown as:

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
24		PS4	Skip	Skip	

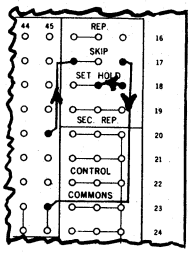
It makes no difference, of course, which hub is wired to the common and which is wired to the select side.

E. SET HOLD

Set hold provides a means of holding information set from storage in the punching dies even though skip cycle has been impulsed and therefore no information is punched in the card. It is very seldom used without skip cycle, although skip cycle is frequently used without set hold.

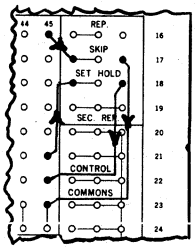
Set hold and reproduce are two completely separate functions, since set hold affects only information set from storage. Therefore if both control holes are punched in the same card, reproduce information will clear at the beginning of the calculating cycle and the new information will be set up at the end of the cycle, just as if the set hold were not punched.

On the input-output panel, the control for set hold is located in line 18, v-x, directly beneath the skip controls. To make set hold operative whenever skip cycle is operative, it is merely necessary to wire the double set hold hub to the single set hold hub. No other wiring is necessary, except the normal skip cycle wiring.



REPRODUCE CONTROLS		
	COL.	POS.
REP		
SKIP	45	9
SET HOLD	45	9
SEC REP		

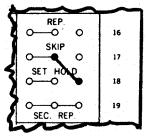
It is possible to wire the computer so that set hold will occur only on certain cards when skip cycle is impulsed. If the control positions have different commons, the wiring would be:



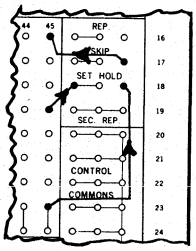
REPRODUCE CONTROLS		
	COL.	POS.
REP		
SKIP	45	1
SET HOLD	45	0
SEC REP		

In this case, set hold would operate only when both 0 and 1 in column 45 are punched. If either is punched separately, set hold will not operate.

If the positions used as skip and set hold have the same common, the wiring is slightly different. This is due to the fact that there is an internal connection between the double hubs of skip and the single hub of set hold:



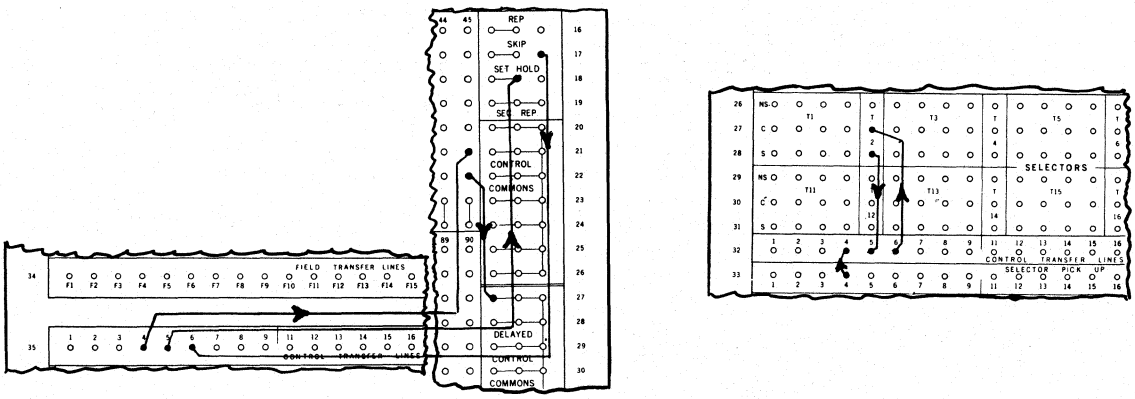
To complete the circuits correctly, therefore, the following wiring would be used:



REPRODUCE CONTROLS		
	COL.	POS.
REP		
SKIP	45	1
SET HOLD	45	7
SEC REP		

Either the double skip hub or the single set hold hub is wired to the common; the skip control is wired to the single skip hub, and the set hold control is wired to the double set hold hub.

It is possible, through use of a selector, to perform set hold without skip cycle. To do so, the single hub of skip cycle is wired to the double hub of set hold through the select side of a selector. The selector may be picked up either by a control hole in the card or by a program select. If it is picked up by a control hole, a delayed control common is wired to the common of the column in which the control hole is punched. If the selector is picked up by a program select, the program select may not be impulsed during the program until a period of time has passed equal to 18 addition or subtraction steps or 6 multiplication or division steps. Assume a 0 in 45 is to operate set hold:



On the selector chart this would be entered:

REPRODUCE CONTROLS		
	COL.	POS.
REP		
SKIP		
SET HOLD	C5, C6	
SEC REP		

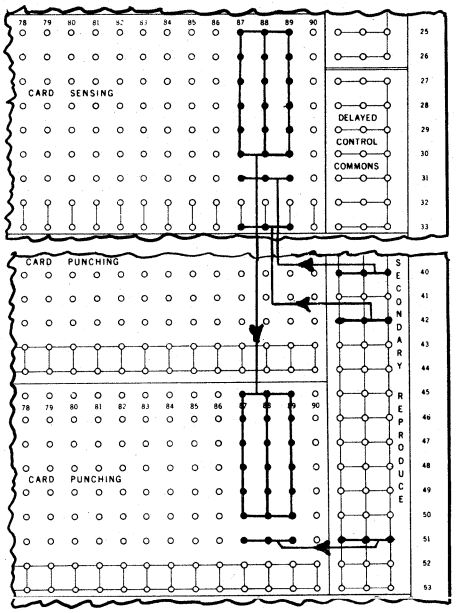
SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
4	C-4	0/45 (DELAY)	C-5	C-6	

F. SECONDARY REPRODUCE

Secondary reproduce is a means of picking up additional information to be reproduced from cards other than those containing a reproduce control. It is operated by a control hole. The same card may not contain a control hole for both reproduce and secondary reproduce.

The sensing of a secondary reproduce control does not clear any values, either reproduce or secondary reproduce. If H (hold) connections are made to the columns in which secondary reproduce information is to be punched, this information will remain set up until the next sensing of a reproduce control, which clears both reproduce and secondary reproduce columns. If the H connections are not made, the secondary reproduce information will be punched only in the card from which it is sensed.

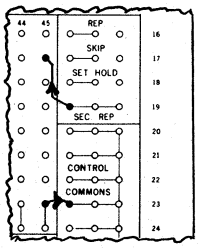
The sensing-punching portion of secondary reproduce connections are similar to those for reproduce, except that the secondary reproduce hubs (lines 36-53, v-x) are used instead of the hubs beneath each punching column.



FROM COLUMN	87	88	89			
TO COLUMN	87	88	89			
CONTROL	SH	SH	SH			

Note that both the hold and commons of the columns are wired to secondary reproduce hubs. On the program chart, in the "control" section, S is used to represent secondary reproduce.

The control hole for secondary reproduce is wired on the input-output panel beneath the control hole for set hold. A control common is wired to the common of the column, and the position is wired to one of the three secondary reproduce hubs.



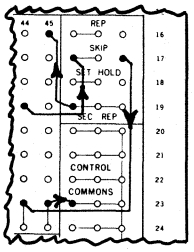
REPRODUCE CONTROLS		
	COL.	POS.
REP		
SKIP		
SET HOLD		
SEC REP	45	3

Since a control common is wired to the common of the column in which the secondary reproduce control is punched, it is possible to use other positions in that column for selector pick-up. It is not possible, however, for a skip control to be punched in that column without use of a selector.

It is possible for a secondary reproduce control to be punched in more than one card of a group. It is important to remember when doing this that a secondary reproduce control does not clear; therefore, if other information is punched in the same columns, it will pile up in the punching dies. However, if two fields are to be secondary reproduced, with one blank in one card while the other is blank in the other, there will be no difficulty.

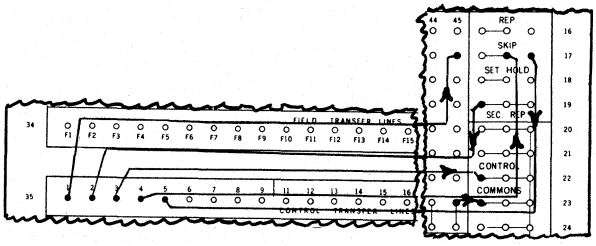
Skip cycle and set hold may be used in conjunction with secondary reproduce. The set hold wiring is identical to that explained earlier.

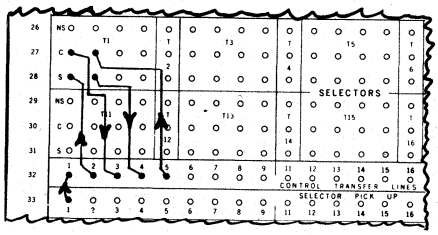
If the control holes for secondary reproduce and skip cycle have separate commons, the wiring is as shown below.



REPRODUCE CONTROLS		
	COL.	POS.
REP		
SKIP	44	7
SET HOLD		
SEC REP	45	1

If the same position is to control both secondary reproduce and skip, this wiring must be done through selectors.



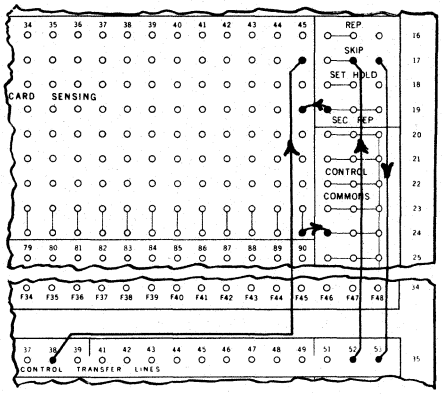


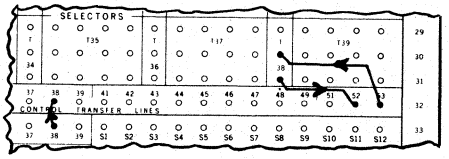
On the selector chart this would be entered:

REPRODUCE CONTROLS		
	COL.	POS.
REP		
SKIP	T1-2 C4, C5	
SET HOLD		
SEC REP	T1-1 C2, C3	

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1	C-1	3/45	C2	C3	—
1-2			C4	C5	—
1-3					
1-4					

If the secondary reproduce and skip cycle control holes have the same common, the skip cycle control must be used to pick up a selector through which the skip hubs are joined. The secondary reproduce control, however, may be wired directly.





G. SUMMARY

1) Up to 80 columns of numeric information or 57 columns of alphabetic information may be reproduced. These totals will be reduced if information is also punched from storage.

2) Information may be reproduced into the same columns or different columns, in the same card or any number of following cards.

3) Reproducing may be done into the same cards in which calculated values are also punched.

4) A card field may not be Y-wired for accumulator input and reproducing, nor for storage output and reproducing.

5) Any position may be reproduced as any position.

6) For transposition punching within the same card the reproducing may occur automatically on every card, whenever a control hole is present, or when a control hole is absent.

7) Reproducing into following cards always involves the use of either a control hole or a program select to control the reproduction.

8) Through use of a skip control the computer is prevented from punching all information, both reproduced and calculated. Skipping may occur either in the card from which new reproduce information is picked up, or any other card.

9) A skip cycle does not clear reproduce values, although values set in the punching dies from storage will clear.

10) The skip function may be controlled either by card control or by a program select.

11) Unless the reproduce and skip control are zero positions, no other control hole punched in the same column may be used except through selectors.

12) Set hold provides a means of holding information set from storage from card to card. It usually operates in association with skip control.

13) A set hold control hole does not affect a reproduce control.

14) Secondary reproduce is a means of picking up additional information for reproducing without clearing that set up originally.

15) Reproduce and secondary reproduce control holes may not be punched in the same card.

16) Secondary reproduce clears neither the columns wired for reproduce nor those wired for secondary reproduce.

17) Skip cycle and set hold may be used in conjunction with secondary reproduce.

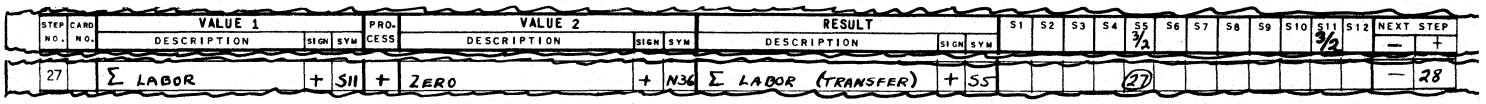
Section III Programming Techniques

1. Transfer into storage

The only processes available for a program step are the four arithmetic operations, yet the only way in which a value can be placed in storage is through a program step. Therefore, the method of transferring the value in one storage to another storage is by adding zero to it and placing the result in the desired storage. This is also the method used for storing an element sensed from a card to be used during the program for a following card.

One reason for the transfer of a value from one storage to another storage is a program in which both detail information and the summary information accumulated from it are to be punched in the same card columns. The two values must be handled in separate storages; yet because a backfeed would occur, the two storages cannot be wired to the same card columns for punching. As part of the summary routine, therefore, the accumulated value is transferred to the storage from which the detail information is punched.

Assume that on step 27 the accumulated labor total, usually indicated on a program chart as Σ or "sum of", from storage S11 is to be transferred to storage S5. The program step would read:



A second frequent use of transfer is during accumulation from card to card. This is discussed below.

2. Accumulation

Frequently during a program, detail information is accumulated to be punched in summary cards. The detail information may be either an element in the detail cards, or a value which is computed during the program for the detail card.

The first step in planning an accumulation routine is to select a storage in which the value will be accumulated. This storage unit must be clear when the accumulation occurs for the first card of each group, or the preceding total will be added to the new total. This is done by manually clearing the computer before the program starts, and programming the clearing of the storage unit at the end of the routine for each summary card.

Accumulation requires two steps for each value accumulated. During the first step the new detail information is added to the accumulated total. Since the same storage unit may not be used as both a value and the result in the same step, the new total must be placed temporarily in some other storage. On the second step of the routine, the new total is transferred to the original accumulating storage.

As an example, assume that labor is computed in S5, and as the accumulating storage S11 has been selected. S1 is used as a working storage. The steps required to accumulate labor would be:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 1/3	S2	S3	S4	S5 3/2	S6	S7	S8	S9	S10	S11 3/2	S12	NEXT STEP —	NEXT STEP +
10		Σ LABOR	±	S11	+	LABOR	±	S5	NEW Σ LABOR	±	S1	10												11	11
11		NEW Σ LABOR	±	S1	+	ZERO	+	N36	NEW Σ LABOR	±	S11											(11)		12	12

Note the use of the symbol Σ to represent "sum of".

Assume that in the first two cards of a group labor is calculated as $9.45 and $8.73. When the first card comes in, S11 is clear. The accumulation would take place as follows:

10) 0.00 + 9.45 = 9.45
11) 9.45 + 0.00 = 9.45 } Card #1

10) 9.45 + 8.73 =18.18
11) 18.18 + 0.00 =18.18 } Card #2

The labor amount in the first card of the group should not be transferred directly into S11. To do so would require a step such as S5 + N36 = S11, which would mean an extra program step. There would also have to be some means of identifying the first card so that all succeeding cards in the group could be prevented from going through this step. If any of the other cards did go through it, the accumulated total would be replaced with the labor amount in that card.

3. Rounding

A. HALF-CENT ROUNDING OF MULTIPLICATION

The majority of multiplications should be rounded. On the 60 and 120, this requires one step in addition to the multiplication.

For multiplications with a dollar and cent result, the usual rounding is half-cent. In other words, a 5 is added to the digit to the right of the final penny. If this digit is less than 5, nothing will be added to the last penny. If it is 5 or more, there will be a carry-over of 1 to the last penny. For example:

```
  4.233                4.236
+ .005               + .005
  4.238 or 4.23        4.241 or 4.24
```

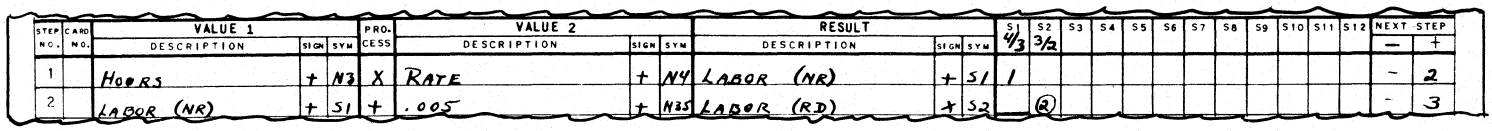
In many multiplications such as this, the result is computed to more than three places following the decimal. For example, if V2 with 3 places following the decimal is multiplied by V1 with 2 places following the decimal, the result is

computed to 5 places. If this is to be rounded to two places, no digit past the third digit can have any possible bearing on the rounded result. Therefore in such a case the result of the multiplication can be placed in a 4/3 storage, dropping off the unused digits to the right. For example:

```
  1.275                    11.793
× 9.25                   + .005
------                   ------
  6375                   11.798 or 11.79
 2550
11475
--------
11.79375 or 11.793
```

The general rule for determining the decimal location of the result storage is that there must be one more place following the decimal than is to be used in the rounded result; there need be no more. Into the last position the rounding value is added; the result of this step should be placed in a storage which will drop off the extra digit, a 3/2 if the original result is placed in a 4/3.

Frequently, S1 is assigned a 4/3 decimal, and the results of all multiplications are placed in it. From S1 they are rounded into their final storages. A typical multiplication and rounding would be:



STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2 3/2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
1		HOURS	+	N3	X	RATE	+	N4	LABOR (NR)	+	S1	1												-	2
2		LABOR (NR)	+	S1	+	.005	+	N35	LABOR (RD)	+	S2		(2)											-	3

Note the use of (NR) following the result of the multiplication, indicating that the result is not rounded. On the following step the symbol (RD) indicates that the result has been rounded. Once this symbol has been used for a result, it is not necessary to continue using it in later steps when the value is called upon. However, (NR) should always be used until the result has been rounded.

Note also the use of N35 as .005. Since the majority of 60 and 120 programs include multiplications which must be rounded, element N35 is usually designated as the rounding factor.

Although a value which is being rounded is usually placed in a storage which will drop off the digit to which the rounding factor has been added, this is not absolutely necessary. If the value is not to be used in further computation, the last digit can be dropped off when wiring from storage for punching. However, if the value is to be used further, the last digit must be dropped off when rounding or the computation will be distorted by the amount of the additional 5 in the last digit. It is preferable always to drop off the digit if possible.

B. ROUNDING OF DIVISION - "ROUNDING UP"

Division may or may not be rounded, depending upon the use to which the quotient is put. An average hourly rate, which is the result of dividing total pay by total hours, is frequently carried out far enough so that there is no need to round. On the other hand, the division of total quantity by pack to determine the number of cartons would be rounded to the nearest carton.

In the last example, a value of 9 rather than 5 would probably be used as the rounding factor. Any decimal remainder would require an additional carton, so the result of the division would be placed in a 2/1 storage. A .9 would be added to the storage, and the final answer would be placed in a 1/0 storage.

For example, if an item is packed three to a carton, and 16 have been ordered, the computation would be:

$$16 \div 3 = 5.3 \qquad \begin{array}{r} 5.3 \\ +\ .9 \\ \hline 6.2 \end{array} \text{ or } 6.$$

C. NEGATIVE ROUNDING

If the result of a step can be minus, this must be taken into account when rounding. If the same rounding step is used as with positive results, the following would occur:

$$\begin{array}{r} -\ 4.233 \\ +\ +\ .005 \\ \hline -\ 4.228 \end{array} \text{ or } -\ 4.22 \qquad \begin{array}{r} -\ 4.236 \\ +\ +\ .005 \\ \hline -\ 4.231 \end{array} \text{ or } -\ 4.23$$

Both of these results are incorrect. The first should be - 4.23 and the second should be - 4.24. If the result of a step is minus, either a - .005 should be added, or a + .005 should be subtracted.

With the subtraction of a + .005, the two above examples would be computed as follows:

$$\begin{array}{r} -\ 4.233 \\ -\ +\ .005 \\ \hline -\ 4.238 \end{array} \text{ or } -\ 4.23 \qquad \begin{array}{r} -\ 4.236 \\ -\ +\ .005 \\ \hline -\ 4.241 \end{array} \text{ or } -\ 4.24$$

If one of the elements to be used in the multiplication is a card-read field which would have a negative control, this negative control may be used both to make the element minus and the .005 minus. To do this, the negative control is wired to the pick-up of a four-pole selector, and both element designators are wired to minus hubs through the selector. This is the same type of wiring that is explained on page 77. For example, suppose that quantity, element N2 which may be plus or minus, is to be multiplied by price, element N3. N35 is the .005 rounding factor. The element chart would be entered as follows:

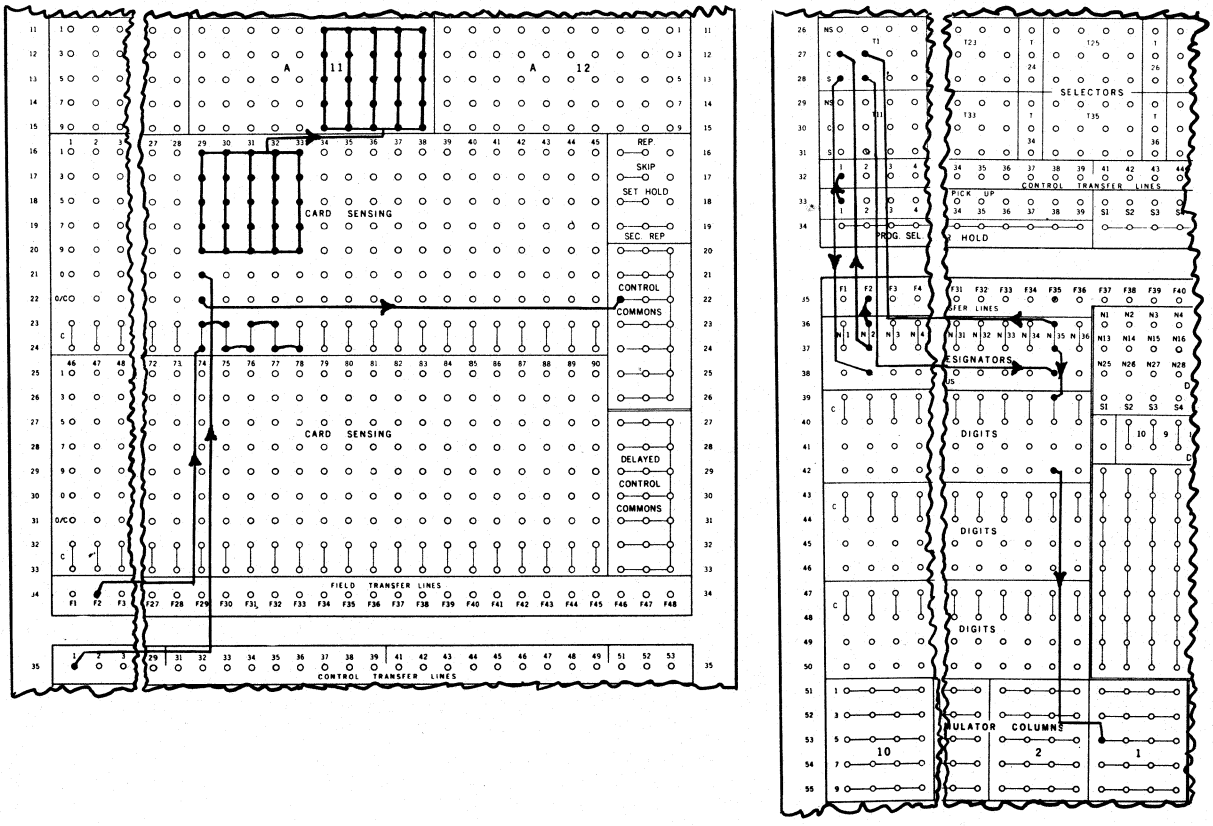
ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N1															
N2	F2	T1+1 29	0	C-1	3/1						29	30	31	32	33
N3	F3				4/3						34	35	36	37	38	39
N17															
N18															

ELEM. DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															
N35		T1+2 29	0	C-1	4/3									0	0 5
N36															

The two steps required for multiplication and rounding would be:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2 3/2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
5		QUANTITY	±	N2	X	PRICE	+	N3	AMOUNT (NR)	±	S1	5												6	6
6		AMOUNT (NR)	±	S1	+	.005	±	N35	AMOUNT (RD)	±	S2		(6)											7	7

The wiring for elements N2 and N35 would be:



There are times when this method is not practical. For example, there may be other multiplications which must be rounded, yet they are not to be rounded negatively dependent upon the control hole which makes the .005 negative. In such instances it is preferable to subtract +.005 rather than adding -.005, since the use of -.005 would involve setting up an additional constant. If sufficient steps are available, the negative rounding should be done in a separate step.

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2 3/2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
5		QUANTITY	±	N2	X	PRICE	+	N3	AMOUNT (NR)	±	S1	5												7	6
6		AMOUNT (NR)	+	S1	+	.005	+	N35	AMOUNT (RD)	+	S2		⑥											-	8
7		AMOUNT (NR)	-	S1	-	.005	+	N35	AMOUNT (RD)	-	S2		⑦											8	-

However, this is an excellent place to combine steps through use of a program select.

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2 3/2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
5		QUANTITY	±	N2	X	PRICE	+	N3	AMOUNT (NR)	±	S1	5												PS1	6
6		AMOUNT (NR)	±	S1	COM T2	.005	+	N35	AMOUNT (RD)	±	S2		⑥											7	7

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1					
1-2					
1-3					
1-4					
2		PS1	−	PR. STEP 6	+

FROM	TO
START	
SET 1	
SET 2	
SORT 1	
SORT 2	
CLEAR	
P.S. I	STEP 6
P.S. II	
P.S. III	
P.S. IV	

D. PROGRESSIVE ROUNDING

There is a technique called "progressive rounding", frequently used in an application involving labor tickets, or when a number of detail cards should come out to a predetermined total. Assume an application where in each job ticket an hourly rate is multiplied by hours. A multiplication of total hours by hourly rate might result in an answer which differs from the accumulated total of the job tickets by a penny or two, due to the rounding of each job ticket. This variance may be over or under, and over a period of time should balance out. However, the problem can be avoided by progressive rounding.

As an example of what can happen if the individual tickets are not rounded progressively:

	Hours		Rate		Amount (NR)		Rounding		Full Amount	Amount (RD)
(a)	2.5	×	1.255	=	3.1375	+	.005	=	3.1425	3.14
	3.2	×	1.255	=	4.0160	+	.005	=	4.0210	4.02
	2.3	×	1.255	=	2.8865	+	.005	=	2.8915	2.89
	8.0				10.0400					10.05
(b)	8.0	×	1.255	=	10.0400	+	.005	=		10.04

In the above example the employee gains a penny.

(a)	1.7	×	1.255	=	2.1335	+	.005	=	2.1385	2.13
	.8	×	1.255	=	1.0040	+	.005	=	1.0090	1.00
	1.9	×	1.255	=	2.3845	+	.005	=	2.3895	2.38
	.8	×	1.255	=	1.0040	+	.005	=	1.0090	1.00
	2.8	×	1.255	=	3.5140	+	.005	=	3.5190	3.51
	8.0				10.0400					10.02
(b)	8.0	×	1.255	=	10.0400	+	.005	=		10.04

In the above example the employee loses 2 cents.

The principle of progressive rounding is to add the rounding .005 only once, as is done in the (b) portion of the two examples. The .005 is added only on the first card. Thereafter the remaining decimals are used for rounding, rather than .005. To do this it is necessary to retain all digits which have been computed following the decimal. The multiplication of a two decimal value times a three decimal value will require a 6/5 storage for the result of the multiplication. It is also necessary to assign another storage with a 6/5 decimal location in which the rounding factor will be stored. A third storage with a 6/5 decimal location is used as a working storage.

Assume the following factors:

Element or Storage	Decimal Location	Description
N2	2/1	Hours
N3	4/3	Rate
N35	4/3	.005
N36	4/3	0.000
S1	5/4	Working
S2	5/4	Working
S4	3/2	Labor
S10	5/4	Rounding Factor

The program steps required would be:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 5/4	S2 5/4	S3	S4 3/2	S5	S6	S7	S8	S9	S10 5/4	S11	S12	NEXT STEP −	NEXT STEP +
1																									
2																									
3		.005	+	N35	+	ZERO	+	N36	ROUNDING FACTOR	+	S10										③			−	4
4		HOURS	+	N2	X	RATE	+	N3	LABOR (NR)	+	S1	4												−	5
5		LABOR (NR)	+	S1	+	ROUNDING FACTOR	+	S10	LABOR (RD - XX.XXXX)	+	S2		5											−	6
6		LABOR (RD XX.XXXX)	+	S2	+	ZERO	+	N36	LABOR (RD - XX.XX)	+	S4				⑥									−	7
7		LABOR (RD XX.XXXX)	+	S2	−	LABOR (RD - XX.XX)	+	S4	ROUNDING FACTOR	+	S10										⑦			−	8

Step 3: Only the first detail card of a group goes through this step, where .005 is stored as a rounding factor.

Step 4: The multiplication takes place, with the result placed in a storage with a decimal location equivalent to the sum of the decimals of the two values.

Step 5: The rounding factor is added to the result, and placed in a storage with a decimal location the same as that in which the result of the multiplication is stored. The purpose is to retain the remaining decimal values so that they can be used as the next rounding factor. The maximum size of the value, including all decimals, is indicated by ×'s.

Step 6: The rounded labor is placed in its final storage, with the remaining decimals dropped off.

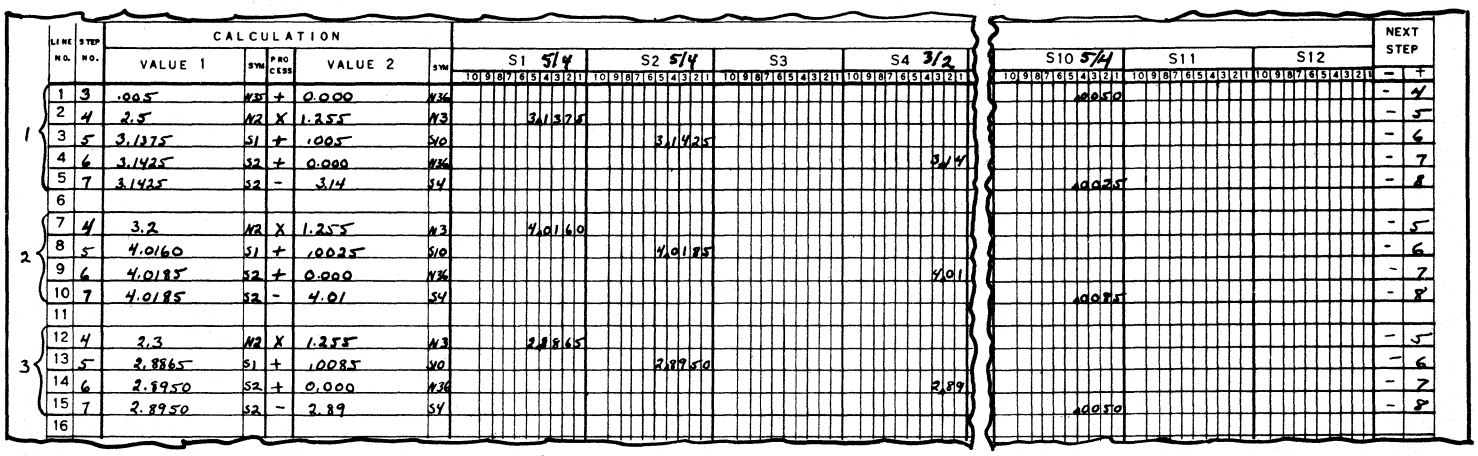
Step 7: The actual labor is subtracted from the labor with the full rounded value. The result is the rounding factor to be used in the next card.

Using the above examples with this program, rounding will take place as follows.

Hours		Rate		Amount (NR)		Rounding		Full Amount	Amount (RD)
2.5	×	1.255	=	3.1375	+	.005	=	3.1425	3.14
3.2	×	1.255	=	4.0160	+	.0025	=	4.0185	4.01
2.3	×	1.255	=	2.8865	+	.0085	=	2.8950	2.89
8.0				10.0400					10.04
1.7	×	1.255	=	2.1335	+	.005	=	2.1385	2.13
.8	×	1.255	=	1.0040	+	.0085	=	1.0125	1.01
1.9	×	1.255	=	2.3845	+	.0025	=	2.3870	2.38
.8	×	1.255	=	1.0040	+	.0070	=	1.0110	1.01
2.8	×	1.255	=	3.5140	+	.0010	=	3.5150	3.51
8.0				10.0400					10.04

Note that this method of rounding requires three additional program steps, as well as the additional storages with high decimal locations.

The first example on preceding page would be computed as follows:



LINE NO.	STEP NO.	CALCULATION VALUE 1	SYM	PROCESS	VALUE 2	SYM	S1 5/4	S2 5/4	S3	S4 3/2	S10 5/4	S11	S12	NEXT STEP −	NEXT STEP +
1	3	.005	N35	+	0.000	N36					.0050				4
2	4	2.5	N12	X	1.255	N13	3.1375								5
3	5	3.1375	S1	+	.005	S10		3.1425							6
4	6	3.1425	S2	+	0.000	N36				3.14					7
5	7	3.1425	S2	−	3.14	S4					.0025				8
6															
7	4	3.2	N12	X	1.255	N13	4.0160								5
8	5	4.0160	S1	+	.0025	S10		4.0185							6
9	6	4.0185	S2	+	0.000	N36				4.01					7
10	7	4.0185	S2	−	4.01	S4					.0085				8
11															
12	4	2.3	N12	X	1.255	N13	2.8865								5
13	5	2.8865	S1	+	.0085	S10		2.8950							6
14	6	2.8950	S2	+	0.000	N36				2.89					7
15	7	2.8950	S2	−	2.89	S4					.0050				8
16															

4. Range testing

A. DETERMINING IF ONE NUMBER IS GREATER THAN ANOTHER

The problem frequently arises of determining whether one number is larger than another number. For example, in computing social security an employee's FICA deductions must stop when his gross earnings to date reach $4200.00, at which time his FICA to date is $84.00. In computing utility bills, the rate may decrease at certain specified consumption amounts. To handle problems such as this the computer must be able to compare numbers.

Since the only processes available are the arithmetic functions, the means of doing this is through subtraction. The actual value of the result is usually unimportant - the sign of the result will indicate which number is larger.

For example, in comparing an employee's FICA to date with $84.00, there are three possible results: it is less than $84.00, equal to $84.00, or greater than $84.00. Either value can be subtracted from the other. The programmer can determine whether there is a preference as to which is subtracted from which by trying examples. If FICA to date is subtracted from $84.00, the possibilities are as follows:

Constant:	84.00	84.00	84.00
FICA to date:	-83.99	-84.00	-84.01
	+ .01	+ 0.00	- .01

When an employee has paid $84.00, social security no longer should be deducted; if the program is properly designed, $84.00 will be the maximum FICA to date. Eliminating the last possibility above, this subtraction has not solved the problem of differentiating between those employees for whom an FICA deduction is required, and those for whom it is not. If the subtraction is reversed, the possibilities are:

FICA to date:	83.99	84.00	84.01
Constant:	-84.00	-84.00	-84.00
	- .01	+ 0.00	+ .01

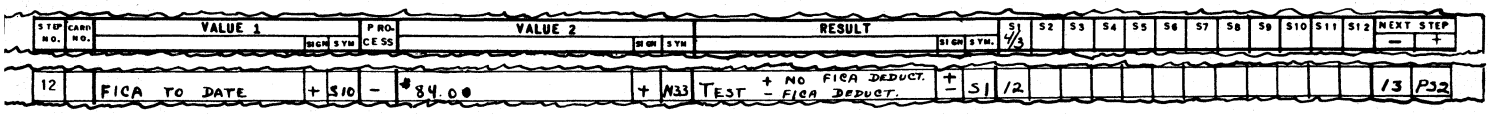
In this case as soon as the result of the subtraction is plus, the employee has reached $84.00.

By using a constant of 83.99 instead of 84.00 the first form of the equation would be satisfactory.

Constant:	83.99	83.99	83.99
FICA to date:	-83.99	-84.00	-84.01
	+ 0.00	- .01	- .02

This is not the preferable method, since more wires are required to wire a constant of 83.99 than 84.00.

Assuming that FICA to date is located in storage S10, and $84.00 is a constant wired as N33, the step to test for an FICA deduction would be:



Note the use of "Test $\pm$" as the result of the step. The best possible description of what has been accomplished by the step should be entered here. The plus and minus branchings of the step will then go to the proper routine for handling the condition.

B. DETERMINING INTO WHICH GROUP A NUMBER FALLS

Sometimes several determinations must be made instead of one as in the above example. Frequently a discount rate is based upon the total amount of the invoice. In such a case successive subtractions must be made until the proper range is found.

For example, assume the following discount rates:

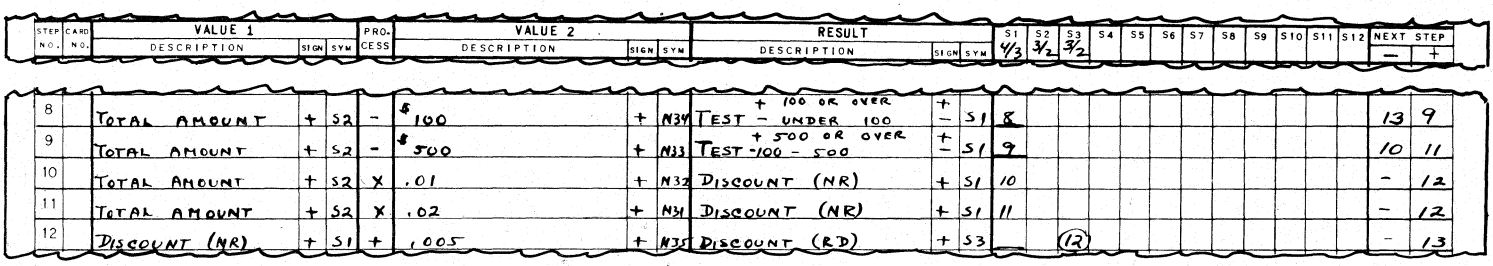
Amount of Invoice	Discount Rate
Under $100.00	---
100.00 - 499.99	.01
500.00 and over	.02

Since the discount rate applies to the entire amount of the invoice, the portion of the total at each rate is unimportant. The subtraction must be stated so that $100 falls in the .01 discount class, and $500 falls in the .02 discount class. Therefore the best method would be:

Amount of Invoice - $100 = Test - under $100
+ $100 or over

Using the following elements, with total amount in storage S2, the program would be:

N31	.02
N32	.01
N33	500.
N34	100.
N35	.005
N36	0.00



STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2 3/2	S3 3/2	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP —	NEXT STEP +
8		TOTAL AMOUNT	+	S2	-	$100	+	N34	TEST - UNDER 100 (+ 100 OR OVER)	+	S1	8												13	9
9		TOTAL AMOUNT	+	S2	-	$500	+	N33	TEST -100 - 500 (+ 500 OR OVER)	+	S1	9												10	11
10		TOTAL AMOUNT	+	S2	X	.01	+	N32	DISCOUNT (NR)	+	S1	10												-	12
11		TOTAL AMOUNT	+	S2	X	.02	+	N31	DISCOUNT (NR)	+	S1	11												-	12
12		DISCOUNT (NR)	+	S1	+	.005	+	N35	DISCOUNT (RD)	+	S3			(12)										-	13

If there are several ranges into which an amount may fall, it is well to test for the most frequently occuring range first. In the above example, if more invoices were over $500 than under $100, a step would be saved on the majority of the cards if the first test were made for $500 instead of $100.

C. DETERMINING AMOUNTS AT VARIOUS RANGES

In computing utility bills, such as electricity, there is usually a sliding rate scale dependent upon consumption. For example, assume a rate schedule such as this:

Kilowatt Hours		Rate for Kilowatt Hour
1st	50	.045
Next	50	.037
Next	100	.027
Over	200	.018

It would be desirable to do the range testing in such a way that the consumption in the final rate bracket is computed at the same time that the bracket is determined. This value can then be multiplied by the corresponding rate. To determine the total amount of the bill, the result of this multiplication is added to a constant value which is the charge for the consumption in the first brackets. For example, if consumption is 175, 75 hours are charged at .027 or $2.03. The first 50 hours cost $2.25 and the next 50 cost $1.85, so the constant value would be $4.10. The total amount of the bill is $6.13.

If the range testing is begun at the highest breaking point, 200, and the breaking point is subtracted from total consumption, the desired results will be obtained. A plus answer indicates that the range has been found, a minus answer indicates that further testing is needed. In the following example a number within each range is used to show the steps through which it would go:

	Breaking Point	Total Consumption			
		235	175	65	30
(1)	200	235 -200 + 35	175 -200 - 25	65 -200 -135	30 -200 -170
(2)	100		175 -100 + 75	65 -100 - 35	30 -100 - 70
(3)	50			65 - 50 + 15	30 - 50 - 20

Three steps are required to test for four ranges. When a result is plus, the range has been found, and the consumption within that range has been placed in storage. For a consumption of less than 50, indicated by a minus answer on the last step, the total consumption is used.

If total consumption equals one of the breaking points, the result will be a plus zero. When this is multiplied by the rate for this bracket, the result will again be zero. When this is added to the constant value representing the first brackets, the correct answer will be obtained. For example:

$$\begin{array}{r} 50 \\ -50 \\ \hline +\ 0 \end{array}$$

$$0 \times .037 = 0$$

$$0 + 1.85 = \$1.85$$

The program to do the range testing and calculation of the bill, assuming that total consumption is in storage S2, would be:

TABLE OF FACTORS

SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE
N1				N13				N25	+		2.25
N2				N14				N26	+		4.10
N3				N15				N27	+		6.80
N4				N16				N28	+		.045
N5				N17				N29	+		.037
N6				N18				N30	+		.027
N7				N19				N31	+		.018
N8				N20				N32	+		50
N9				N21				N33	+		100
N10				N22				N34	+		200
N11				N23				N35	+		.005
N12				N24				N36	+		ZERO

CARD DESCRIPTION

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2 1/0	S3 4/3	S4 4/3	S5 3/2	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
1																									
2																									
3		TOTAL CONSUMPTION	+	S2	−	200	+	N34	TEST + OVER 200 − LESS THAN 200	+ −	S1	3												4	6
4		TOTAL CONSUMPTION	+	S2	−	100	+	N33	TEST + 100-200 − LESS THAN 100	+ −	S1	4												5	8
5		TOTAL CONSUMPTION	+	S2	−	50	+	N32	TEST + 50-100 − LESS THAN 50	+ −	S1	5												12	10
6		CONS. OVER 200	+	S1	X	.018	+	N31	AMOUNT (NR)	+	S3			6										-	7
7		AMOUNT (NR)	+	S3	+	6.80	+	N27	TOTAL AMOUNT (NR)	+	S4				7									-	13
8		CONS. 100-200	+	S1	X	.027	+	N30	AMOUNT (NR)	+	S3			8										-	9
9		AMOUNT (NR)	+	S3	+	4.10	+	N26	TOTAL AMOUNT (NR)	+	S4				9									-	13
10		CONS. 50-100	+	S1	X	.037	+	N29	AMOUNT (NR)	+	S3			10										-	11
11		AMOUNT (NR)	+	S3	+	2.25	+	N25	TOTAL AMOUNT (NR)	+	S4				11									-	13
12		CONS. LESS THAN 50	+	S2	X	.045	+	N28	TOTAL AMOUNT (NR)	+	S4				12									-	13
13		TOTAL AMOUNT (NR)	+	S4	+	.005	+	N35	TOTAL AMOUNT (RD)	+	S5					(13)								-	14

Note that the rounding step, step 13, may be avoided for those consumptions of 50 or more. This can be accomplished by adding the rounding 5 to the constant value. Elements N25, N26, and N27 would therefore be as follows:

N25: 2.255
N26: 4.105
N27: 6.805

Steps 7, 9, and 11 would have S5, the total amount storage with a 3/2 decimal location, as the result of the step. Step 13, however would still be included in the program for the rounding of those consumptions under 50.

A series of steps such as this is an excellent place to combine steps through use of program selects. Steps 6, 8, 10 and 12 could be stated as one step, as could steps 7, 9 and 11. The program for this is shown on the following page.

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2 1/0	S3 4/3	S4 4/3	S5 3/2	S6	S7	S8	S9	S10	S11	S12	NEXT STEP —	NEXT STEP +
1																									
2																									
3		TOTAL CONSUMPTION	+	S2	-	200	+	N34	TEST + OVER 200 - LESS THAN 200	+ −	S1	3												4	PS1
4		TOTAL CONSUMPTION	+	S2	-	100	+	N33	TEST + 100-200 - LESS THAN 100	+ −	S1	4												5	PS2
5		TOTAL CONSUMPTION	+	S2	-	50	+	N32	TEST + 50-100 - LESS THAN 50	+ −	S1	5												6	PS3
6		CONSUMPTION	+	COM T1-1	X	RATE	+	COM T1-2	AMOUNT (NR)	+	S3			6										-	7
7		AMOUNT (NR)	+	S3	+	6.80/4.10/2.25/6.00	+	COM T1-3	TOTAL AMOUNT (NR)	+	S4				7									-	8
8		TOTAL AMOUNT (NR)	+	S4	+	.005	+	N35	TOTAL AMOUNT (RD)	+	S5					(8)								-	9

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1		PS1 (OVER 200)	S1	V1 STEP 6	COM T 3-1
1-2			N31	V2 STEP 6	COM T 3-2
1-3			N27	V2 STEP 7	COM T 3-3
1-4					
2					
3-1		PS2 (100-200)	S1	N.S. T 1-1	COM T 5-1
3-2			N30	N.S. T 1-2	COM T 5-2
3-3			N26	N.S. T 1-3	COM T 5-3
3-4					
4					
5-1		PS3 (50-100)	S1	N.S. T 3-1	S2
5-2			N29	N.S. T 3-2	N28
5-3			N25	N.S. T 3-3	N36
5-4					

FROM	TO
START	
SET 1	
SET 2	
SORT 1	
SORT 2	
CLEAR	
P.S. I	STEP 6
P.S. II	STEP 6
P.S. III	STEP 6
P.S. IV	

In this example step 8 could be completely avoided by adding the rounding 5 to the constant value. N25, N26, and N27 would be as mentioned above. Instead of using zero, N36, to transfer the consumptions under 50, N35 (.005) would be used. The result of step 7 would be placed in the final result storage with a 3/2 decimal location.

5. Use of an element as the result of a step

It is possible to use an element as the result of a step, provided it is known that the step so wired will prove. For example, if a positive value is subtracted from itself, the result is always zero. If element N36, zero, is wired as the result of such a step, it will prove. For example:

```
N1 - N1 = N36
 5 -  5 =   0      Step

N36 + N1 - N1 = 0
 0  +  5 -  5 = 0  Proof
```

If at any time the result of the step is not the same as the element which is the result, the computer will hang up. For example:

N1 - S1 = N36 5 - 4 = 0	Step
N36 + S1 - N1 = 0 0 + 4 - 5 ≠ 0	Proof - Computer Hangs Up

One use of a step such as this is a "selector delay", to allow the 20 milliseconds following start which is necessary before selectors may be used. The selector delay step may be stated as:

N36 + N36 = N36
0 + 0 = 0

Two other uses of an element as a result storage are in connection with verification. Its use in connection with verification of designating information from card to card is found on page 123, and in connection with verification of a recalculated value on page 151.

6. Testing designating information (sequence check)

A. DETERMINING IF TWO NUMBERS ARE IDENTICAL

Similar to range testing in principle is the programming technique of determining whether one number is the same as another. This technique is almost always used in a multiple card routine to test the designating information (clock number, part number, customer number, etc.) to be certain that all cards within a group contain the same designation.

To do this testing, the designation is placed in storage on the first card of a group. Each succeeding card of the group is tested against the storage to be certain that it is identical. If it is not the same, the computer is usually stopped by wiring a 0 ÷ 0 step to stop.

Throughout the following explanation it will be assumed that the designating information which is punched in the cards is element N1, and that it will be stored from the first card in S12. S1 will always be used as the working storage in which the testing is done. Step 40 will be a 0 ÷ 0 step to stop the computer. Since this testing routine is used so frequently, it is well to be consistent in the element and storages used.

To determine whether one number is the same as another, two subtraction steps are necessary. On the first step, one number is subtracted from the other. If 25 is element N1 as punched in the card the sign possibilities when subtracting S12 from it would be:

Punched:	25	25	25
Stored:	-25	-24	-26
	+ 0	+ 1	- 1

If the result is minus, the value which is stored must be larger than that which is punched. If the result is plus, the value in storage is either equal to or less than that which is punched. A minus result may be wired directly to step 40 to stop the computer. The computer is usually stopped if the two values do not agree so that the card which is out of sort may be placed in its proper location. There are times, of course, when the 0 ÷ 0 step may be wired to sort - particularly if the determination of whether the two values are equal is not for the purpose of checking the sequence of the cards. A plus result on the step is routed to the next step. In this step the values are reversed for subtraction:

Stored:	25	24
Punched:	-25	-25
	+ 0	- 1

If the result of this step is minus, the value in storage is less than that which is punched. If the result is plus, the stored and punched values are equal.

Assuming that a clock number has been stored in S12 from the first card of a group, the two steps required to do this testing would be:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12 1/0	NEXT STEP −	NEXT STEP +
1	ALL	CLOCK NO. (PCH)	+	N1	−	CLOCK NO. (ST.)	+	S12	TEST + MAY BE OK − ERROR	±	S1	1												40	2
2	ALL	CLOCK NO. (ST.)	+	S12	−	CLOCK NO. (PCH)	+	N1	TEST ± OK ERROR	±	S1	2												40	3
40	ERR	ZERO	+	N36	÷	ZERO	+	N36	ERROR STEP	+	S1	40													

Note that for ease of reference the description is followed by "PCH", meaning punched, or "ST", meaning stored. It is possible during the first step to subtract either S12 from N1 or N1 from S12. To be consistent, however, this manual will always use N1-S12 as the first step of the pair.

B. IDENTIFICATION OF FIRST CARD OF A GROUP

As mentioned earlier, it is always necessary to identify the first card of a group when doing a checking routine like this. Not only must the designation be stored from that card, but the computer must not stop because the new designation is higher in value than the number in storage.

In some cases there is a control hole which will identify the first card, as in the case of a rate or pricing card, or a previous balance card. Before this card can be separated from the others, however, it must go through one step, which will act as a selector delay. Since it will do no harm to send it through the first step of the testing routine, this step is usually used as a selector delay instead of a zero + zero step. Assuming that T1 is picked up by a zero in column 90, which is punched only in the first card of a group, the routine for testing the cards and storing clock number in S12 would be:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
												4/3											1/0		
1	ALL	CLOCK NO (PCH)	+	N1	−	CLOCK NO (ST.)	+	S12	TEST − ERROR + MAY BE OK, 1ST CARD	+ −	S1	1												40	COM. T1-1
2	ALL	CLOCK NO (ST)	+	S12	−	CLOCK NO (PCH)	+	N1	TEST − ERROR + OK	+ −	S1	2												40	5
3	0	CLOCK NO.	+	N1	+	ZERO	+	N36	STORE CLOCK NO.	+	S12												③	-	4
40	ERR	ZERO	+	N36	÷	ZERO	+	N36	ERROR STEP	+	S1	40													

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1	C-1	0/90 (RATE CARD)	STEP 3	+ BR. STEP 1	STEP 2
1-2					
1-3					
1-4					

Step 1: All cards go through this step. Since the cards are sorted in ascending sequence, both cards which are equal to the stored clock number and the first card of a new group will branch on the plus side. Only cards of descending sequence, which are errors, will branch on the minus side. The plus branching is routed through selector T1-1. If the card is the first of a group, the selector will be select and the program continues with the storing of the new clock number. If the selector is non-select, the next step is a continuation of the testing. Note that for the first card of the first group into the computer, S12 will be zero; therefore steps 1 and 3 will still be followed.

Step 2: Since it is well to keep the two basic testing steps together, this is used as step 2 instead of the storing of clock number. No first cards enter this step, so all cards with a minus branching on this step are errors. Cards with a plus branching are correct, and are routed to the first step of the detail routine.

If there are several types of cards in this category, each type is routed through selectors from this step to its own routine. It is preferable to separate the types at the end of step 2 rather than the end of step 1, when the first cards are separated. If they had been identified at the end of step 1, the first step of each routine would have been identical to step 2; by delaying identification until testing has been completed, several steps are saved.

Step 3: The first card of each group is sent through this standard storing step. It is unimportant whether S12 is cleared at the end of the group, since the old value is cleared when the new is entered.

In many cases it is impossible to identify the first card of a group by means of a control hole. For example, when sorting a group of labor cards there is no way of knowing which of the cards will be the first for a particular employee. Therefore the computer itself must identify the first card.

This is done by clearing S12 on the final card of a group. The final card is usually easy to identify, since one of the two fundamental reasons for checking the designating information is to be certain that only cards belonging to the same group are summarized into a summary card. The other reason for checking is the use of information from a first card in the routine for succeeding cards; in this case the first card can be identified and it is unnecessary to identify the last card.

If S12 is cleared on the final card of a group, one step in addition to those listed above is required to determine the first card of a group. Such cards would branch on the minus branching of step 2:

1) N1 - S12 = S1
   25 - 0 = +25

2) S12 - N1 = S1
   0 - 25 = -25

Instead of sending the minus branching of step 2 to step 40, it should first be sent to a step to determine whether S12 is clear, which would be the subtraction of S12 from zero. If it is clear, this card is the first of a group; if it is not clear, the card is an error.

N36 - S12 = S1
0 - 0 = +0

The routine of four steps necessary for the testing of a clock number and storing it would be:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
1	ALL	CLOCK NO. (PCH)	+	N1	−	CLOCK NO (ST)	+	S12	TEST + MAY BE OK, 1ST CARD − ERROR	±	S1	1/3											1/0	40	2
2	ALL	CLOCK NO (ST)	+	S12	−	CLOCK NO (PCH)	+	N1	TEST ± OK ERROR, 1ST CARD	±	S1	2												3	5
3	ERR 1ST	ZERO	+	N36	−	CLOCK NO (ST)	+	S12	TEST + 1ST CARD − ERROR	±	S1	3												40	4
4	1ST	CLOCK NO (PCH)	+	N1	+	ZERO	+	N36	STORE CLOCK NO	+	S12												(47)	−	5

Note that all cards could be sent through step 3, N36 - S12, as the first step of the program, and from the minus branching continue with the two normal testing steps. However, the majority of the cards will probably be detail cards which are not the first of their group, and all of these cards therefore would be routed through an extra step.

If alpha columns are wired as part of the designation, the test for a first card may change slightly when selectors are used to identify the first card. This will occur only when S12 is not cleared following the last card of a group - for example, when the first card of a group is a rate card and there is no summary card. Alpha punching is arbitrarily changed to a numeric value,

and although the alpha punching itself is in ascending sequence, the numeric values have no relation to each other. For example, although a "C" follows an "A" in sorting, the "C" would have a lower numeric value.

	Punched Code	Accumulator Columns
A	1, 5, 9	3 2 1 / 1 1 1
C	0, 7	9 9

see page 132 for alpha wiring

Because of this, the first card of a group may be in either ascending or descending sequence. Therefore both the plus and minus branching of step 1 must be wired to the selector which is picked up by the control hole. They would, of course, be wired to separate poles, since the step to which each branching is wired after the first card has been separated will differ. For example:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/8	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12 1/0	NEXT STEP −	NEXT STEP +
1	ALL	CLOCK NO (PCH)	+	N1	−	CLOCK NO (ST)	+	S12	+ MAY BE OK, 1ST CARD TEST − ERROR, 1ST CARD	+ −	S1	1												COM T1-1	COM T1-2
2	ALL	CLOCK NO (ST)	+	S12	−	CLOCK NO (PCH)	+	N1	+ OK TEST − ERROR	+ −	S1	2												40	5
3	0	CLOCK NO (PCH)	+	N1	+	ZERO	+	N36	STORE CLOCK NO	+	S12												③	−	4

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1	C-1	0/90 (RATE CARD)	STEP 3	− BR STEP 1	STEP 40
1-2			STEP 3	+ BR. STEP 1	STEP 2
1-3					
1-4					

Note that if S12 is cleared at the end of the routine for a summary card, this problem does not occur. The first step is then the subtraction of zero from a punched clock number (N1-S12); it therefore does not matter whether the groups are in ascending or descending sequence. The first card of a group will always branch on the minus branching of step 2.

C. IDENTIFICATION OF BLANK SUMMARY CARDS

Frequently in a 60 or 120 routine a blank card into which summary information is to be punched is collated following a group of cards. A blank will branch on the minus branching of step 1, since nothing is punched as element N1:

$$N1 - S12 = S1$$
$$0 - 25 = -25$$

Therefore, if blank cards are included in the routine, a step must be included to determine whether the cards on the minus branching of the first step are summary cards or errors. These cards are routed to a step which subtracts N1 from zero. If the result is plus, the program continues with the routine for the summary card; if it is minus, it is routed to the error step.

A program for a complete test of designating information including both a first card without a control hole and a blank summary card follows:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP —	NEXT STEP +
												4/3											1/0		
1	ALL	CLOCK NO (PCH)	+	N1	−	CLOCK NO (ST)	+	S12	TEST − + MAY BE OK, 1ST CARD ERROR, S/C	+ −	S1	1												3	2
2	ALL	CLOCK NO (ST)	+	S12	−	CLOCK NO (PCH)	+	N1	TEST − + OK ERROR, 1ST CARD	+ −	S1	2												4	6
3	ERR S/C	ZERO	+	N36	−	CLOCK NO (PCH)	+	N1	TEST − + S/C ERROR	+ −	S1	3												40	27
4	ERR 1ST	ZERO	+	N36	−	CLOCK NO (ST)	+	S12	TEST − + 1ST CARD ERROR	+ −	S1	4												40	5
5	1ST	CLOCK NO (PCH)	+	N1	+	ZERO	+	N36	STORE CLOCK NO	+	S12												(5)	−	6
40	ERR	ZERO	+	N36	÷	ZERO	+	N36	ERROR STEP	+	S1	40													

For all testing routines, steps 1, 2, 5 and 40 should be included. If there is a blank summary card, step 3 must be included. If the first detail card has no control hole, step 4 must be included. Note that when step 4 is used, S12 must have been cleared on the preceding summary card.

D. DESIGNATIONS OF MORE THAN SEVEN DIGITS

The designating information sometimes exceeds seven digits, yet the storage in which it is to be tested has been given a 4/3 decimal location for use in other steps. It is important to remember that if the designation is assigned a 1/0 decimal and is subtracted from zero into a 4/3 storage, the computer will hang up because the result exceeds the capacity of the storage. In such a case both the designation and S12 should also be assigned a 4/3 decimal.

E. REUSE OF DESIGNATION STORAGE

Occasionally in a particularly complicated routine there do not seem to be sufficient storages. In such a case it is possible to use S12, the storage in which the designation is placed, as a working storage during the program. When this is done the testing steps are the same, except that as the final step for every card the designation must be replaced in S12. The first card of a group must still be determined as previously outlined, and the designation stored, preferably as the last step of the routine for the card. For each following card, after the testing is completed, S12 is available for any other purpose. Since the testing at the beginning of the program proved that the columns punched as N1 are correct for that group, the final step for these succeeding cards is to return to N1 + N36 = S12, so that the designation will be available in S12 for testing the next card.

F. USE OF AN ELEMENT AS RESULT OF TESTING STEP

A second method of conserving storages, and also of conserving steps, is the use of an element as the result of a testing step. This method should be used with special care by the programmer, since it is not as flexible as those discussed above. On page 116 the use of an element as a result for a selector delay step is explained. It is pointed out that if the step will prove, an element may be used as a result. A testing step can therefore be developed which combines the two basic testing steps, N1 - S12 = S1 and S12 - N1 = S1, into one step:

```
N1 - S12 = N36
25 -  25 =   0
```

The proof of this step would be:

```
N36 + S12 - N1 = 0
  0 +  25 - 25 = 0
```

If the two values are not identical, the following would occur:

```
N1 - S12 = N36
25 -  24 =   0          Step

N36 + S12 - N1 = 0
  0 +  24 - 25 ≠ 0      Proof - computer hangs up
```

The above method has saved the testing storage (S1), the second basic testing step, and the 0 ÷ 0 error step. However, since this step cannot be directed to the 0 ÷ 0 step in case of error, the 0 ÷ 0 light will not light to indicate why the computer has stopped, and the test panel must be used instead.

Note that when using this method the first card of a group should have an identifying code punched so that it can be prevented from entering this step; otherwise it would hang up the computer as shown in the second example above. Therefore the selector delay step of N36 + N36 = N36 (see page 116) should be the first step of the routine, with the first card of the group going from there to a step storing the identifying information (N1 + N36 = S12) while the remaining cards go to the testing step.

If blank summary cards are included in the routine, they will not check:

```
N1 - S12 = N36
 0 -  24 =   0          Step

N36 + S12 - N1 = 0
  0 +  24 -  0 ≠ 0      Proof - computer hangs up
```

They could, of course, be removed prior to the testing by the standard step of N36 - N1 = S1. However, this would require the use of S1 as a working storage, and one of the reasons for using this method is to eliminate the use of S1. If the summary cards are blank except for a control hole, they can be prevented from entering the step at the end of the selector delay step when the first card of each group is separated.

Note that with this method S12 can still be used as a working storage as explained on page 122.

G. USE OF N ÷ 0 AND 0 ÷ 0 AS THE SECOND STEP OF A TESTING ROUTINE

On programs in which first cards of a group and summary cards may be identified by a control hole, the 0 ÷ 0 step of a testing routine may be saved by using a division step as the second step rather than the normal subtraction step. This method involves the division of the difference between N1 and S12 by zero. If the two numbers are identical, the difference is zero, zero is automatically given as the result of the step, and the computer is allowed to continue. However, if there is any digital value as the difference between N1 and S12, either positive or negative, the computer signals a number divided by zero, which is wired to stop. Note that neither first cards nor blank summary cards can be permitted to enter this step, since in both these cases there will be a difference between N1 and S12.

Assuming that first cards, rate cards, may be identified by a 1 in column 45, which picks up T2, and all other cards are labor cards, the program would be:

CARD DESCRIPTION 1= RATE CARD 3= LABOR CARD

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
1	1,3	CLOCK NO (PCH)	+	N1	−	CLOCK NO (ST)	+	S12	DIFFERENCE	±	S1	1												2	COM T2
2	3	DIFFERENCE	±	S1	÷	ZERO	+	N36	TEST 0 ÷ 0 - OK N ÷ 0 ERROR	±	S2		2											-	5

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1					
1-2					
1-3					
1-4					
2	C2	1/45 (RATE)	ST. 3	+ BR. ST. 1	ST. 2

	0 ÷ 0	N ÷ 0
STOP		X
SORT		

At the plus branching of step 1 will be first cards of a new group, correct detail cards, and incorrect detail cards in which the punched clock number is greater than the stored clock number. The rate cards are directed immediately to step 3, the first step of the rate card routine, by means of selector T2. All other cards which branch plus are routed to step 2. Although the cards branching minus must be errors since the punched clock number is less than the stored clock number, they are routed to step 2 so that the computer will stop with the N ÷ 0 light lit.

Since 0 ÷ 0 is not wired to stop or sort, all cards in which N1 is equal to S12 continue with step 5, the first step of the detail card routine. Note that the plus branching of step 2 could be wired to the common of a selector to separate various types of cards. Note also that the 0 ÷ 0 light will light on every correct card, since this is an automatic feature regardless of whether the computer is wired to stop on 0 ÷ 0.

If there is any difference between N1 and S12, the computer will stop with the N ÷ 0 light lit. It is unnecessary to wire the minus branching of step 2 since the only cards which do not stop the computer are those in which this step is 0 ÷ 0, the result of which is always a plus zero.

7. Placing more than one value in a storage

A technique to conserve storages is the use of the same storage for more than one value. This will usually require additional steps, but steps may be available when storages are not.

In order to combine two values in the same storage, both must carry the same sign - that is, they may both be positive or both be negative, but one may not be positive when the other is negative. The only exception to this is explained on page 166. There must also be sufficient space in the ten digits of the storage so that certain digits can be assigned to each value with the knowledge that neither will exceed the allotted number of digits.

Two values may be placed in the same storage either as elements or from other storages. The techniques for doing so are slightly different.

A. ENTERING TWO ELEMENTS IN STORAGE

If the two values are elements, and the problem is merely to accumulate these values from card to card for punching in a summary card, the problem is relatively simple. Note that in this example the two card fields are not to be used as individual elements during the program for the card in which they are punched, but merely to be accumulated for a summary card. In such a case the two values may be assigned only one element number and one decimal location. For example, assume the following card fields in daily payroll summary cards which are to be accumulated for the gross pay card:

Description	Columns	Decimal
Total Hours	21-23	22/23
Overtime Hours	28-30	29/30

These two fields could be entered into accumulator input 2 as element N2 as follows:

ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD-F1, F2, ETC. CONTROL - C1, C2, ETC.) DECIMAL LOCATION, NEGATIVE CONTROL AND TRANSFER LINE. INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N1															
N2	F2				2/1					21	22	23	28	29	30
N3															

ELEM. DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															

ACCUMULATOR INPUTS

TOP LINE - TRANSFER LINE (F1, F2, ETC.)
INDICATE:
BOTTOM LINE - CARD COLUMNS

SYM.	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
A1										
A2					F2	F2	F2	F2	F2	F2
					21	22	23	28	29	30

The decimal location assigned is usually the correct decimal location of one of the two elements, preferably the one on the right.

To accumulate the two values from card to card, only two steps are required instead of the usual four required to accumulate two values. If the two are to be accumulated in S11, the necessary steps would be:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP -	NEXT STEP +
												4/3										2/1			
15		HOURS / OVERTIME HRS	+	N3	+	Σ HOURS / OVERTIME HOURS	+	S11	NEW Σ HRS / O.T. HRS.	+	S1	13												-	16
16		NEW Σ HRS / O.T. HRS.	+	S1	+	ZERO	+	N36	NEW Σ HRS / O.T. HRS	+	S11											(18)		-	17

Using the actual values for total hours and overtime hours, the accumulation would take place as follows:

Card #1

N2	+	S11	=	S1
9001.0	+	0.0	=	9001.0
S1	+	N36	=	S11
9001.0	+	0.0	=	9001.0

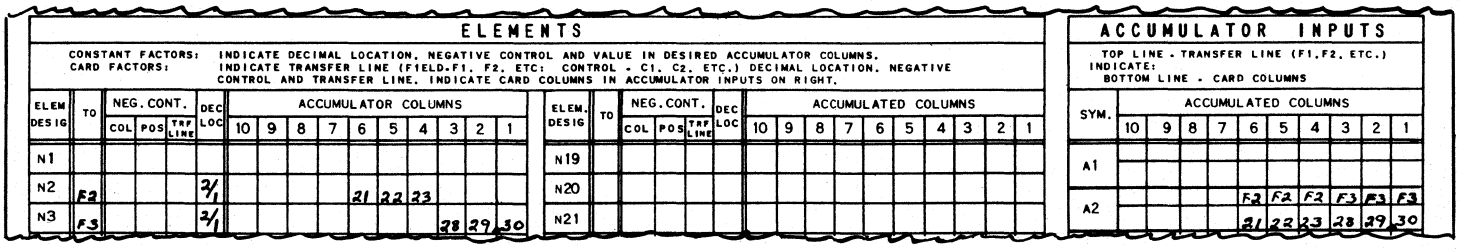
Card #2

N2	+	S11	=	S1
8500.5	+	9001.0	=	17501.5
S1	+	N36	=	S11
17501.5	+	0.0	=	17501.5

The total hours are 17.5 and overtime hours are 1.5. Through the technique explained on page 128, it is possible to separate the two values and use them individually for computation in the summary card.

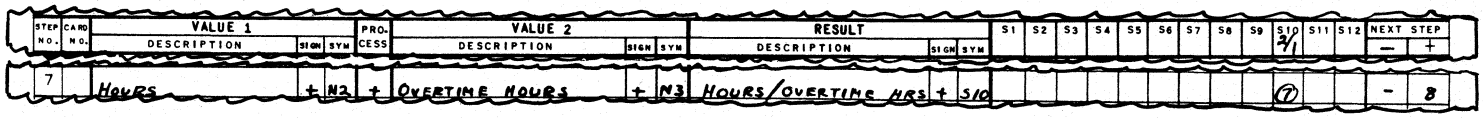
Note that this technique may be used when more elements are required than are present on the computer. Two separate values may be placed in storage as one element, using the standard storing step of the element plus zero instead of the accumulating steps shown above. The values may be separated either in the same card or some following card.

A similar technique for entering two card-read fields into storage, yet one which retains the fields as separate elements, is that of assigning the same decimal point to two elements but offsetting them in the accumulator input. Using the same two card fields as on the preceding page, they would be entered as follows:



Note that in the above case, element N2 could not be used in the card in which it is punched without adjustment of the decimal point. This may be done in a step by dividing N2 by 1000, or by using a selector to change the decimal location after N2 has been combined with N3. N3, however, may be used without any change in decimal location.

To place the values in the same storage unit, the following step would be used.



For example, with hours of 8.5 and overtime hours of .5, the step would be:

$$85000.0 + .5 = 85000.5$$

From this storage unit they could be accumulated in two standard accumulating steps, or held in storage to be separated or punched in a following card.

B. COMBINING VALUES FROM TWO STORAGES IN ONE STORAGE.

If the two values which are to be combined are located in storages, it is usually necessary to change the decimal location of one of the values. Assume the following:

	10	9	8	7	6	5	4	3	2	1	Dec.	Sym.
Withholding Tax							x	x	x	x	3/2	S2
FICA								x	x	x	3/2	S3
FICA/ Withholding Tax			x	x	x		x	x	x	x	3/2	S4

To combine these two values in S4 so that FICA is located in positions 8-6, FICA should be multiplied by 100,000 with the result placed in an intermediate storage (S1, for example.) The addition of S1 and S2 will then result in the placing of the two values side by side in S4. The constant value by which FICA is multiplied is determined by the number of digits it is to be moved to the left.

Assuming that element N33 is a value of 100,000, the program for placing withholding tax and FICA in the S4 would be:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2 3/2	S3 3/2	S4 3/2	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
26		FICA	+	S3	X	100,000	+	N33	FICA (XXX 000.00)	+	S1	26												−	27
27		FICA (XXX 000.00)	+	S1	+	WITHHOLDING TAX	+	S2	FICA/WITHHOLDING TAX	+	S4	—		(27)										−	28

C. SEPARATION OF TWO VALUES IN THE SAME STORAGE - THREE STEPS.

To separate two values which have been combined in the same storage, three steps are usually required. Using the example with FICA and withholding tax in S4, FICA can be obtained in one step by dividing the value in S4 by 100,000 and placing the result in a storage with a 3/2 decimal location. To obtain withholding tax, this result is multiplied by 100,000, to compute the value which was originally added to the withholding tax. This is then subtracted from the value in S4, resulting in the withholding tax.

The program for this would be:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2 3/2	S3 3/2	S4 3/2	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
31		FICA/WITHHOLDING TAX	+	S4	÷	100,000	+	N33	FICA	+	S3			(31)										−	32
32		FICA	+	S3	X	100,000	+	N33	FICA (XXX 000.00)	+	S1	32												−	33
33		FICA/WITHHOLDING TAX	+	S4	−	FICA (XXX 000.00)	+	S1	WITHHOLDING TAX	+	S2	—	(33)											−	34

Assuming that the FICA is $1.72, and withholding tax is $8.46 the values would be separated as follows:

31) 172008.46 ÷ 100,000 = 1.72 (FICA)

32) 1.72 x 100,000 = 172000.00

33) 172008.46 - 172000.00 = 8.46 (Withholding Tax)

Note that the left of the two values can be obtained in one step, while the value to the right requires two additional steps.

If the decimal location of the storage is assigned in relation to the left value instead of the right value, 8/7 instead of 3/2 in this case, both values can still be obtained in three steps. However, to do so would require the use of two storages with 8/7 decimal locations, which are not as useful for other purposes as the 3/2 decimal locations which can be used with the other method.

D. SEPARATION OF TWO VALUES IN THE SAME STORAGE - TWO STEPS.

With the proper arrangement of values in storage, it is sometimes possible to separate them in two steps instead of three. To do this, the storage column preceding the right value must be blank at all times. If the right value contains four digits, column 5 must be blank, and the left value may be placed in columns 6-10.

Assume the same arrangement of FICA and withholding tax as previously discussed. To separate FICA the step would be the same as in the previous example.

S4	÷	N33	= S1
172008.46	÷	100,000	= 1.72

To separate withholding tax, FICA may be dropped off to the left. To do this, FICA must be transferred completely out of the A section of the accumulator and the 11th column of the A section must be zero. When the computer hangs up because of the presence of digits to the left of the 10th storage column, a significant digit in the 11th column is actually the cause of it. If the 11th column of the A section is blank, digits can be present in any columns of the M section without hanging up the computer.

Therefore, in the above problem, withholding tax may be transferred to a storage with a 9/8 decimal location. The two step program would be:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2 3/2	S3 3/2	S4 3/2	S5 9/8	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
31		FICA/WITHHOLDING TAX	+	S4	÷	100,000	+	N33	FICA	+	S3			(31)										−	32
32		FICA/WITHHOLDING TAX	+	S4	+	ZERO	+	N36	WITHHOLDING TAX	+	S5					(32)								−	33

The result of step 32 would be computed in the accumulator as follows:

M Section											A Section										
11	10	9	8	7	6	5	4	3	2	1	11	10	9	8	7	6	5	4	3	2	1
								1	7	2			8	4	6						

The withholding tax in S5 may be used immediately in further computation, since FICA has been dropped off to the left. Note that if for any reason withholding tax extended into column 11 of the A section, the computer would hang up. .

8. Crossfooting

Occasionally the problem arises of crossfooting a large number of small fields. By conventional methods, this requires one less step than the number of fields to be crossfooted. With 15 two-digit fields assigned as elements N1-N15, the usual approach would be:

1) N1 + N2 = S1
2) S1 + N3 = S2
3) S2 + N4 = S1

14) S1 + N15= S2

However, by reading several of the fields as one element, it is possible to reduce the number of addition steps. By multiplication it is then possible to do the crossfooting of the final storage.

Assume that the maximum total of the 15 two-digit fields will never exceed three digits - 999. In this case three two-digit fields can be considered one element - the number of digits equal to the size of the maximum total must be allowed for each of the individual fields. The same field assignment of accumulator columns should be made in each element. If the fields are punched from columns 11-40, they might be arranged as follows:

ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD-F1, F2, ETC: CONTROL - C1, C2, ETC.) DECIMAL LOCATION. NEGATIVE CONTROL AND TRANSFER LINE. INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N1	F1				1/0			11	12		13	14		15	16
N2	F2				1/0			17	18		19	20		21	22
N3	F3				1/0			23	24		25	26		27	28
N4	F4				1/0			29	30		31	32		33	34
N5	F5				1/0			35	36		37	38		39	40
N6															
N7															
N8															

ELEM. DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															
N22															
N23															
N24															
N25															
N26															

ACCUMULATOR INPUTS

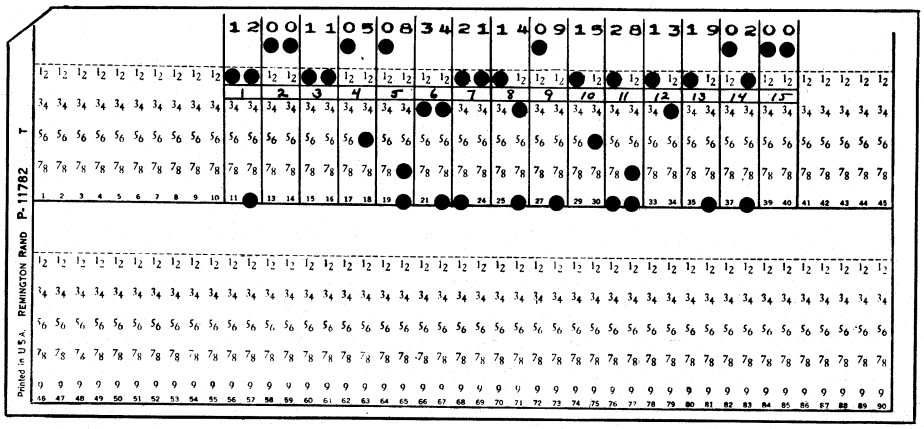
TOP LINE - TRANSFER LINE (F1, F2, ETC.)
INDICATE:
BOTTOM LINE - CARD COLUMNS

SYM.	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
A1			F1	F1		F1	F1		F1	F1
			11	12		13	14		15	16
A2			F2	F2		F2	F2		F2	F2
			17	18		19	20		21	22
A3			F3	F3		F3	F3		F3	F3
			23	24		25	26		27	28
A4			F4	F4		F4	F4		F4	F4
			29	30		31	32		33	34
A5			F5	F5		F5	F5		F5	F5
			35	36		37	38		39	40

The accumulation of these five elements into one storage would require the following steps:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	VALUE 1 SIGN	VALUE 1 SYM	PROCESS	VALUE 2 DESCRIPTION	VALUE 2 SIGN	VALUE 2 SYM	RESULT DESCRIPTION	RESULT SIGN	RESULT SYM	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP -	NEXT STEP +
												1/0	1/0												
1		FIELDS 1, 2, 3	+	N1	+	FIELDS 4, 5, 6	+	N2	Σ FIELDS 1-6	+	S1	1													-	2
2		Σ FIELDS 1-6	+	S1	+	FIELDS 7, 8, 9	+	N3	Σ FIELDS 1-9	+	S2		2													-	3
3		Σ FIELDS 1-9	+	S2	+	FIELDS 10, 11, 12	+	N4	Σ FIELDS 1-12	+	S1	3														-	4
4		Σ FIELDS 1-12	+	S1	+	FIELDS 13, 14, 15	+	N5	Σ FIELDS 1-15	+	S2		4													-	5

With a card punched as follows:



the calculation would be:

CALCULATION

LINE NO.	STEP NO.	VALUE 1	SYM	PROCESS	VALUE 2	SYM	S1 1/0	S2 1/0	S3
1	1	12000011	N1	+	5008034	N2	17008045		
2	2	17008045	S1	+	21014009	N3		38022054	
3	3	38022054	S2	+	15028013	N4	53050067		
4	4	53050067	S1	+	19002000	N5		72052067	
5									

To crossfoot the value in S2, S2 is multiplied by .001001001. This constant is composed of a 1 in the right digit of each field. The decimal is so arranged that in the result the decimal point will immediately precede the total. For example:

```
      72052067
  .001001001
  ---------------
     72052067
   72052067
72052067
---------------------
72124191119067
```

The result of this multiplication is placed in a 4/3 storage, which will drop off all digits to the right of the desired answer. Three steps are then required to break out the total. In the first step the value is placed in a 1/0 storage to determine the digits preceding the desired total. In the next step these digits are subtracted from the entire value. In the last step the desired total is multiplied by 1000 to correct the decimal point. Note that these steps are unnecessary if the final total is only to be punched, not used in further calculation.

These four steps would be entered on the program charts as follows:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
5		Σ FIELDS 1-15	+	S2	X	.001001001	+	N39	TOTAL 1-15 (XXXXXX.XXX)	+	S3			5											−	6
6		TOTAL 1-15 (XXXXXX.XXX)	+	S3	+	ZERO	+	N36	PRECEDING DIGITS (XXXXXX.)	+	S1	6													−	7
7		TOTAL 1-15 (XXXXXX.XXX)	+	S3	−	PRECEDING DIGITS (XXXXXX.000	+	S1	TOTAL (.XXX)	+	S4				7										−	8
8		TOTAL (.XXX)	+	S4	X	1000	+	N33	FINAL TOTAL (XXX.)	+	S1	(8)													−	9

Continuing with the sample values:

LINE NO.	STEP NO.	CALCULATION VALUE 1	SYM	PROCESS	VALUE 2	SYM	S1 1/0	S2 1/0	S3 4/3	S4 4/3
5	5	72052067	S2	X	.001001001	N39			72134191	
6	6	72134.191	S3	+	0.000	N36	72134			
7	7	72134.191	S3	−	72134	S1				191
8	8	.191	S4	X	1000	N33	191			

With this example, a total of 8 steps is required to crossfoot the 15 fields punched in the card, while it would require 14 if each field had been assigned a separate element number.

9. Wiring alpha into the accumulator

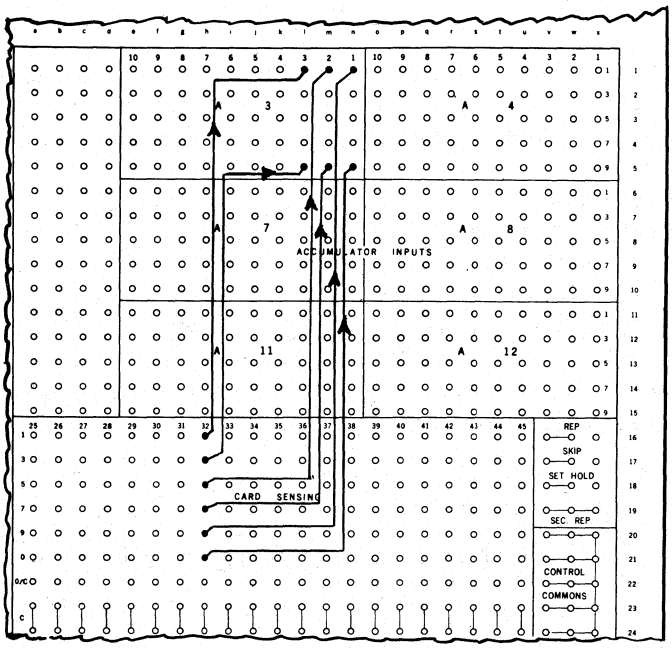
There are times when it is necessary to wire a limited amount of alpha into the accumulator of the 60 or 120. The most common use of this is the test from card to card of a part number or similar designating field containing one or two columns of alpha, to be certain that no cards are out of sort.

If such a column is wired normally, position for position, the alpha will cause the computer to stop with the "input" light lit. However, by expanding one column of alpha into 2 1/2 or 3 accumulator columns, any possible alpha punching can be changed to a numeric combination which will be accepted by the computer. The numeric value thus created is meaningless, but it can be compared with a preceding card to determine whether it is identical. (See page 117).

If only one column of alpha is to be wired, three accumulator columns are required. Assuming that the alpha is punched in column 32, the general principle of the wiring is:

Acc. Pos.	Accumulator Column		
	3	2	1
1	1/32	5/32	9/32
3			
5			
7			
9	3/32	7/32	0/32

The above would appear on the input-output panel as follows:



Some sample alpha combinations would enter the computer as:

Letter	Code	Col. 3	Col.2	Col. 1
A	1,5,9	1	1	1
J	1,3,5	2	1	-
Q	3,5,7	9	2	-
C	0,7	-	9	9

As long as each accumulator column is wired so that all possible combinations of punches result in a numeric code, which is accomplished by having one of the two positions in each accumulator column a "9", it does not matter which card position is wired to which accumulator position. For example, card position 5 could be wired to accumulator position, 1, 3, 5, or 7; however, the other card position wired to the same column must be wired to accumulator position 9.

There is no alpha code which combines a zero and a 1. Therefore, it is possible to wire two columns of alpha in five accumulator columns instead of six. To do this, the zero and 1 in a card column must be wired to the same accumulator input column. With the alpha in columns 32 and 33, the wiring might be:

Acc. Pos.	Accumulator Column 5	4	3	2	1
1	1/32	5/32	1/33	3/33	7/33
3	0/32		0/33		
5					
7					
9	3/32	7/32	9/32	5/33	9/33

With the above wiring, it is possible to enter into accumulator columns 5 and 3 the following values: 1, 2, 3, 4, 9, -; but since 0 and 1 are never punched in the same column, a 1 and 3 combination could never occur.

By wiring storage outputs to card punching positions in the same pattern as the alpha is entered, it is possible to punch the designating information.

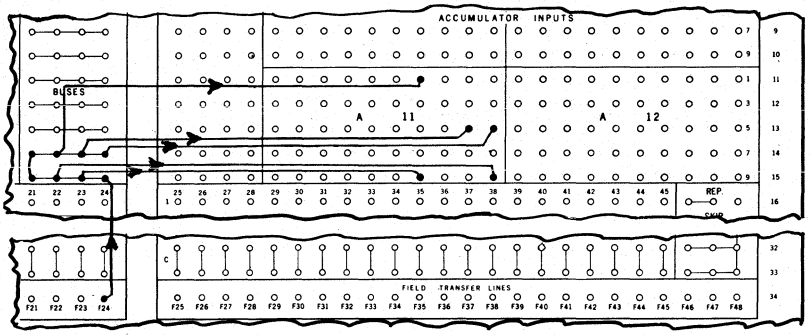
Whenever alpha wiring is done, a note should be attached to the program chart giving a complete description of the wiring used.

The use of alpha wiring changes slightly the procedure used in testing card sequence. This change is covered on page 120.

10. Wiring constants through input

Occasionally the allotment of constants ordered with the 60 or 120 may be insufficient for a particular problem. In such a case accumulator inputs may be used as constants.

To do this, the power from the element designator for the constant value (see page 31) is transferred from the constant-program panel to the input-output panel. This is easier to wire if it is done through an F-line rather than a C-line. The transfer line on the input-output panel is wired to the accumulator positions representing the constant value. Any accumulator input may be used for this purpose. For example, to create the number 20.56 as element N24, the following wiring would be done:



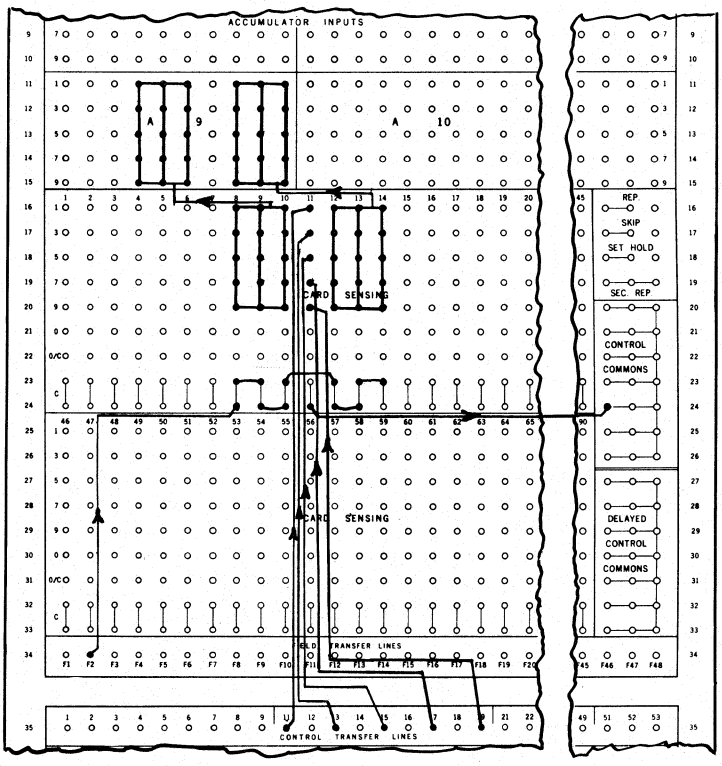
Element N24 may be called upon at any time during the program. It will always have a value of 20.56, just as if it had been wired in the constant digits section of the constant-program panel.

A note should be attached to the program chart indicating the special wiring which has been done.

11. Wiring a card column as an element and to pick up selectors

Occasionally a problem may arise in multiple card routines in which a card column is part of a field in one card, yet is a control column to pick up selectors in another card form. It is also possible that a column in a card might be used to pick up selectors yet is also needed as an element, perhaps so that it may be stored for punching in a later card. Both of these problems can be solved with the same type of wiring.

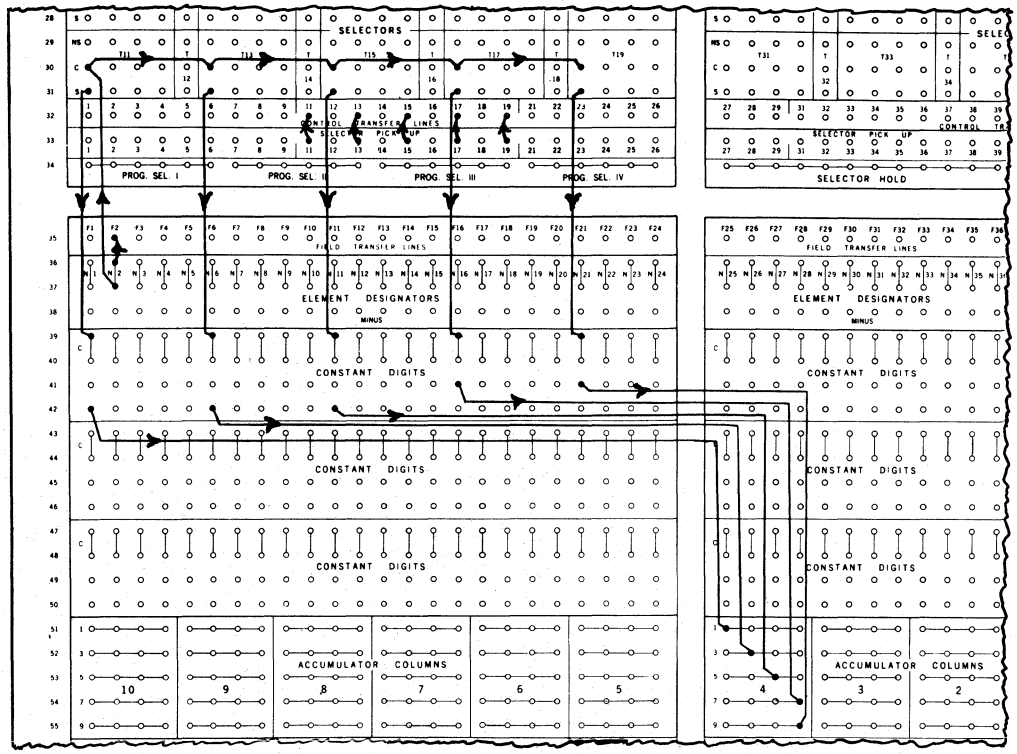
Assume that card columns 8-14 are to be wired as element N2 in one card form, while column 11 is to pick up selectors in another card form. Columns 8-10 and 12-14 are wired as normal accumulator input, with their commons wired to F2. Column 11, however, is wired to pick up selectors. Each position in the column is wired to a control transfer line, with the common of the column wired to a control common. This would be diagrammed as follows:



On the constant-program panel F2 is wired to N2 in the normal way. This provides for using columns 8-10 and 12-14 as element N2. Each control transfer line is wired to pick up a selector, also normal wiring. If the position involved is to be used for selection as well as being part of element N2, the selector must be a 4-pole selector. It is possible, however, that not all positions in the column are used for control purposes; these selectors may be single-pole selectors.

Three poles in each 4-pole selector are available for whatever choices are to be made dependent upon control punching in column 11. The fourth pole is used to create the value which will be used as the element. To do this, the element designator N2 is wired to the common of one pole in each selector. The select side of the pole is wired to create the same constant digit as the position which picked up the selector - if T11 is picked up by a 1 in column 11, the select side is wired to become a constant 1. The accumulator column into which the constant digit is wired is determined by its relationship to the rest of the value. In this example column 11 normally would be wired into accumulator input column 4; therefore the constant digits are all wired to accumulator column 4 in the constant section.

Wiring for this would be:



On the selector chart, this would appear as follows:

ELEME

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONT
CARD FACTORS: INDICATE TRANSFER LINE (FIELD-F1, F2, ET CONTROL AND TRANSFER LINE. INDICATE CARD

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N1															
N2	F2				3/2				8	9	10		12	13	14
N3															

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
11-1	C11	1/11	Constant 1, Acc 4	Elem. Des. N2	—
11-2					
11-3					
11-4					
12					
13-1	C13	3/11	Constant 3, Acc 4	Elem. Des. N2	—
13-2					
13-3					
13-4					
14					
15-1	C15	5/11	Constant 5, Acc 4	Elem Des. N2	—
15-2					
15-3					
15-4					
16					
17-1	C17	7/11	Constant 7, Acc 4	Elem Des. N2	—
17-2					
17-3					
17-4					
18					
19-1	C-19	9/11	Constant 9, Acc 4	Elem. Des. N2	—
19-2					
19-3					
19-4					

Note that whether a particular card column is to be used as an element or to pick up selectors, the selectors corresponding to the positions in the card will always be picked up. Therefore, it is possible in the same card to use the column both as an element and to pick up selectors. An example of this would be a shift code which must be stored for punching in a later card, as well as used during the card in which it is punched to determine what shift premium rate must be applied.

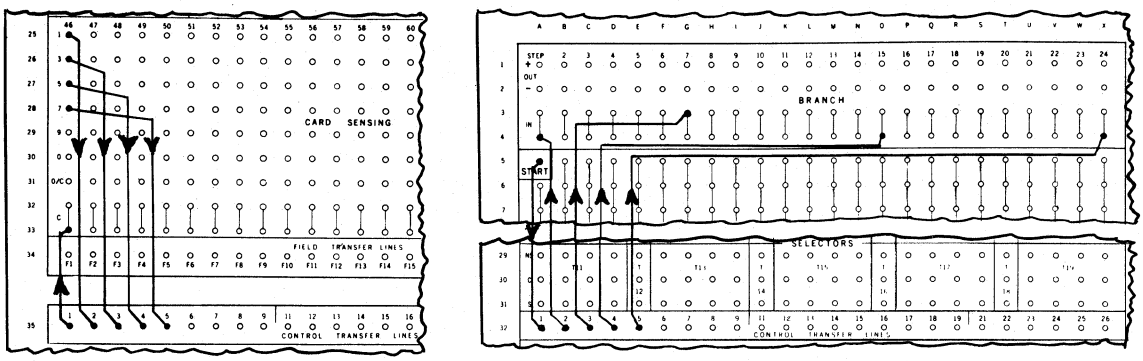
12. Card position selection

Card position selection involves the routing of a pulse through one of several positions in a card rather than directly to its destination. It provides a means of selection by card positions without the use of selectors. With the exception of obtaining zero or a significant value as an element, (page 138), in all cases a control hole must be present in every card. If there is no control hole, the computer will hang up for lack of an instruction. Any pulse on the computer may be routed through a card position, except from the hubs beneath a constant digit to an accumulator column.

As an example of card position selection, assume that start is to be routed to one of four steps, depending on the position punched in column 46 of the card:

Position	To Step
1	1
3	7
5	15
7	24

The wiring would be:



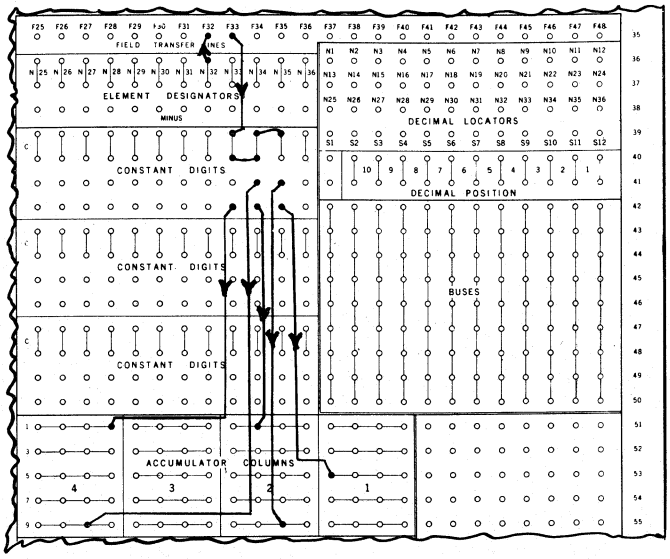
Note that the pulse which is to be selected is routed through a control transfer line to the common of the column, through the various positions, and returned to the constant-program panel through additional C-lines.

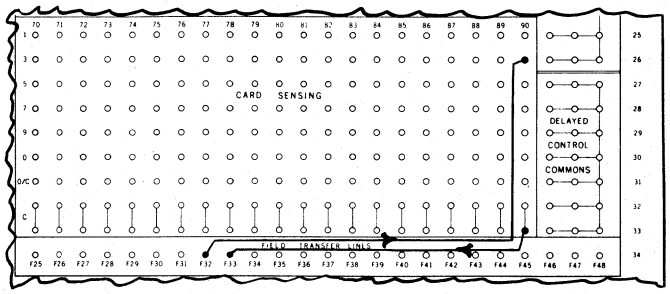
An explanation or diagram of such wiring should be attached to the program charts.

13. Making an element zero or a significant value

It is possible to make a card-read field or a constant value either zero or a significant value without a selector, by using a control hole for the purpose. To do this the element designator is wired to the commons of the columns or of the constant digits through the control hole. If the control hole is present, a connection will be made so that the significant value will be used as the element; if the control hole is not present, the power will be unable to reach the commons of the columns and the value will be zero.

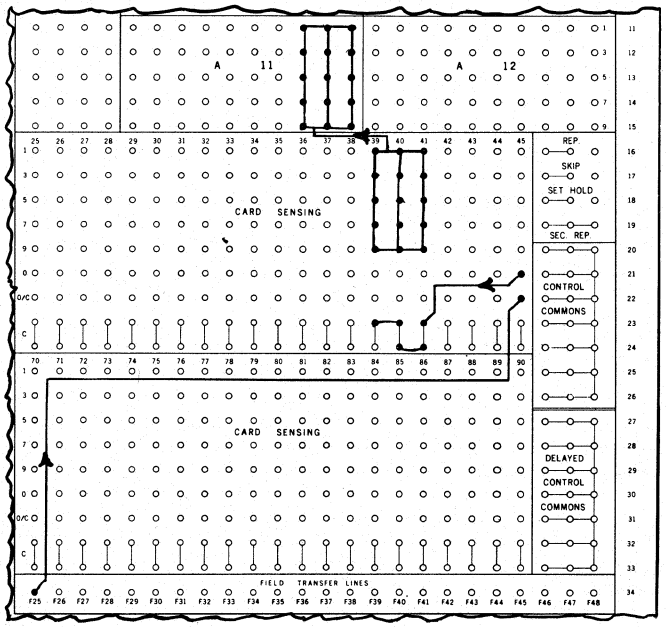
To obtain a constant value of 20.25 as element N32 whenever there is a 3 in column 90, the wiring would be:

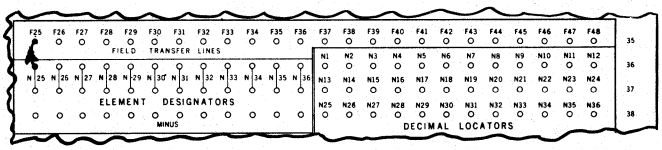




If the 3 is not present in column 90, N32 will be zero.

To obtain card columns 39-41 as element N25 whenever there is a zero in column 45, the wiring would be:





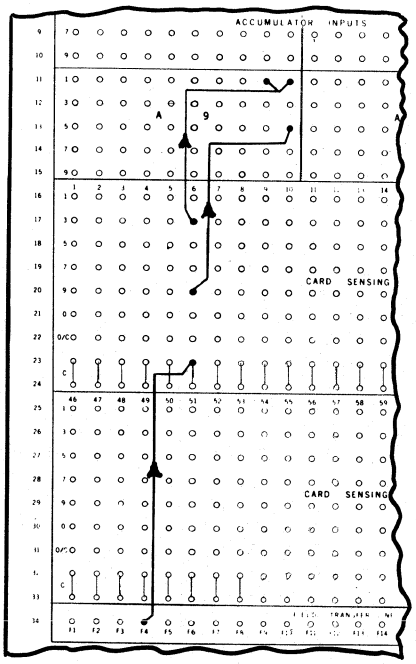
If the zero is not present in column 45, N25 will be zero.

Note that with this type of wiring, the control hole must be present in the card in which the significant value is to be used and not present when zero is used. It would be impossible to use this wiring if the value is to be zero when the control hole is present and significant when the control hole is not present.

This wiring should be indicated by a note attached to the program chart.

14. Conversion of codes to constants

A code punched in a card may, in some applications, be converted directly to a constant value at accumulator input. For example, assume a shift code of 1, 2, or 3 punched in column 6. Second shift employees receive .05 an hour shift premium, third shift employees receive .11, while first shift employees receive nothing. Instead of using shift code to pick up selectors to call on constant values of .05 and .11, the shift code may be wired as a card-read element, with the punching converted to the desired values. In the above example, the 9 in column 6 would be wired to 5 in accumulator input, while the 3 is wired to 11. Note that the 9 must be used for second shift employees, since 1 is punched in both first and second shift cards, with the 9 distinguishing the two. If shift code is to be element N4, the wiring would be as follows:

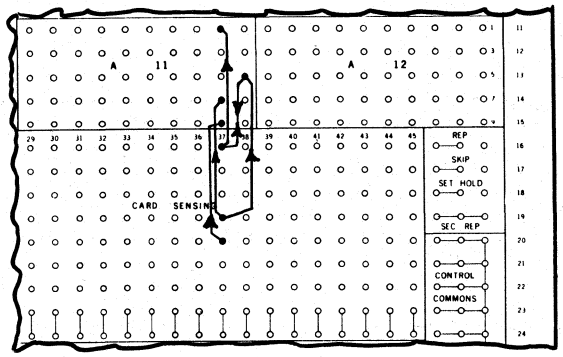


Note that in the preceding example, if N4 is called upon for first shift employees, it will have a value of zero, since the 1 position in column 6 is not wired. Therefore the program step for computing shift premium in the problem could be N4 times total hours.

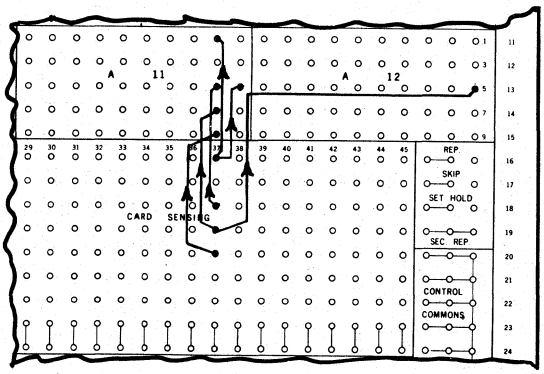
When converting codes to constants, it may be necessary to wire the same position in the same accumulator column from more than one code. For example, assume a payroll application with quarters of an hour, or a billing application with only quarters of a dozen required. The quarters could be punched in one column as follows:

Code	Value
2	.25 (1/4)
5	.50 (1/2)
7	.75 (3/4)

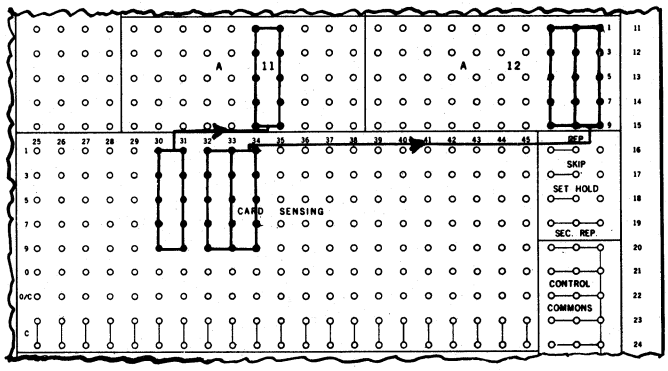
In wiring this, both the 2 and the 7 must be wired to a 5 in the adjacent column. If they are wired to a 5 in the same accumulator input, a backfeed will develop so that when a .25 is read into the accumulator, a .75 will also be read; when a .75 is read, a .25 will also be read. The following diagram with the Y-wiring of the 5 position shows the reason:



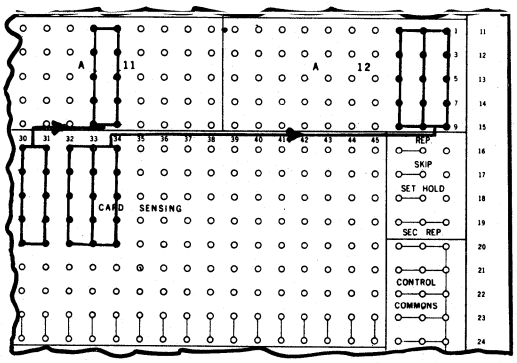
To avoid this, the 5 should be wired to two different accumulator inputs. Since the input positions are diode protected, no back feed can develop. The full wiring for fourths punched in this way would be:

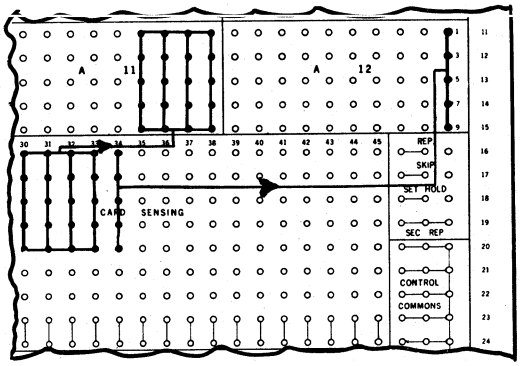


Note that any value can be read into as many accumulator inputs as desired, as long as the card columns retain their relative location in the accumulator. For example, a five digit field punched in columns 30-34 with the decimal between 32 and 33, could be wired to input as follows:



It could not however, be wired as in either of the following examples:





In the first example the field would become six digits with a zero between columns 31 and 32; in the second example columns 33 and 34 would be read into the same accumulator column. The resultant value would be a combination of the two, probably stopping the computer since the two digits together might well be an alpha combination.

15. Overlapping card fields

In a multiple card routine in which different card forms are used, card fields which are to be read as input may overlap. For example, the card field being wired as N2 from one card may consist of columns 5 to 9; the card field being wired as N3, which is punched in a different card, may extent from 7 to 11. Through use of selectors, it is possible to differentiate between the two. There are two approaches to the problem, however, depending on whether the total number of card columns involved is ten or less, or more than ten.

If the total number of card columns involved does not exceed 10, only one selector pole is required. The two card fields are wired to the same accumulator input, and the element designation of the columns is controlled through a selector. Using the above example, the element section of the program chart would appear as follows. Assume the card containing element N3 is punched with a 0 in column 45.

ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD.F1, F2, ETC: CONTROL - C1, C2, ETC.) DECIMAL LOCATION, NEGATIVE CONTROL AND TRANSFER LINE. INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N1															
N2	F2 F4				5/4				5	6	7	8	9		
N3	F3 F4				3/2						7	8	9	10	11
N4															

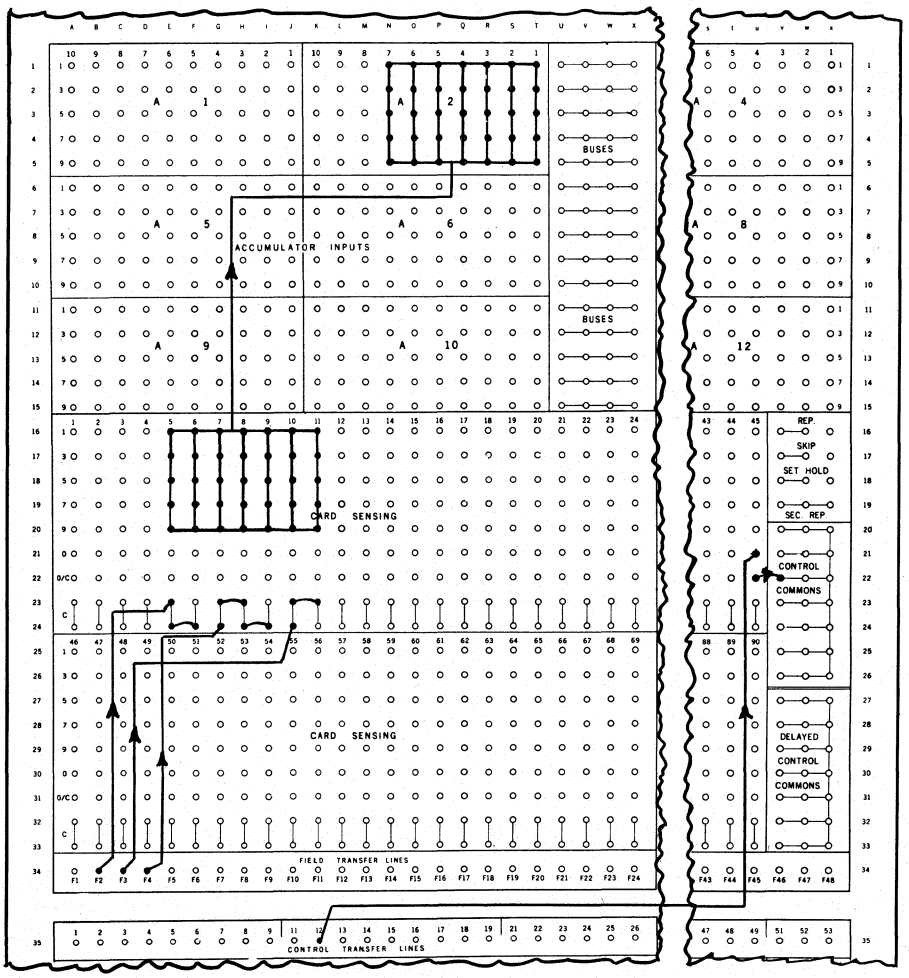
ELEM. DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															
N22															

ACCUMULATOR INPUTS

TOP LINE - TRANSFER LINE (F1, F2, ETC.) INDICATE:
BOTTOM LINE - CARD COLUMNS

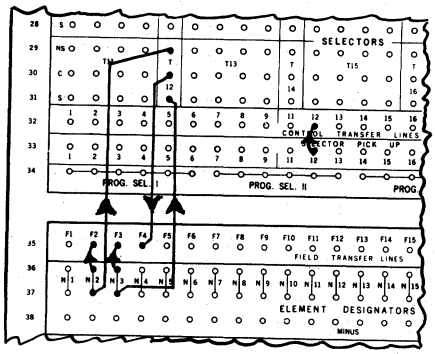
SYM.	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
A1										
A2				F2	F2	F4	F4	F4	F3	F3
				5	6	7	8	9	10	11
A3										

The wiring on the input-output panel would be:



The selector chart and constant-program panel would be:

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
11-1					
11-2					
11-3					
11-4					
12	C12	0/45	N3	F4	N2



When the card punched with element N2 is being sensed, columns 5 and 6 are called on through F2. Through the non-select side of T12, F4 calls on columns 7 to 9. There is no possibility of also calling on columns 10 and 11. When the card punched with element N3 is being sensed, columns 10 and 11 are called on through F3. Through select side of T12, F4 calls on columns 7-9, with no possibility of also calling on columns 5 and 6.

When the total number of card columns involved in overlapping fields exceeds 10, the two fields must be placed in different accumulator inputs. The columns which overlap cannot be Y-wired, since a back feed would develop. Therefore, each position of the overlapping columns, as well as the commons, must be wired through selectors. This results in the use of five selector poles for each card column, plus one for the common of the value.

For example, suppose that element N1 is punched in columns 10-16, and N2 is punched in 16-22. The total number of columns involved is 13, with one overlapping. When a 1 is present in column 90, element N2 is to be used. The two fields might be placed in accumulator input as follows:

ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD-F1, F2, ETC: CONTROL - C1, C2, ETC.) DECIMAL LOCATION, NEGATIVE CONTROL AND TRANSFER LINE. INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N1	F1 F3				3/2				10	11	12	13	14	15	T17 T18
N2	F2 F3				4/3				T17 T18	17	18	19	20	21	22
N3															

ELEM. DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															

ACCUMULATOR INPUTS

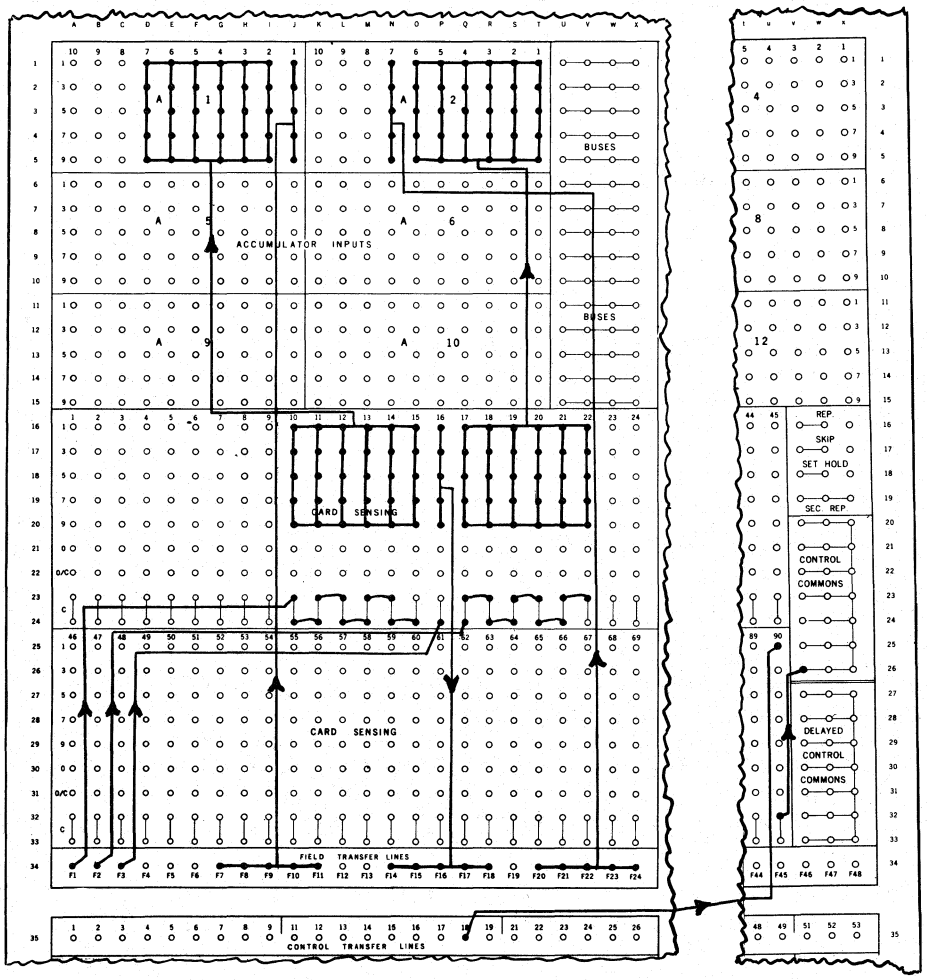
TOP LINE - TRANSFER LINE (F1, F2, ETC.) INDICATE:
BOTTOM LINE - CARD COLUMNS

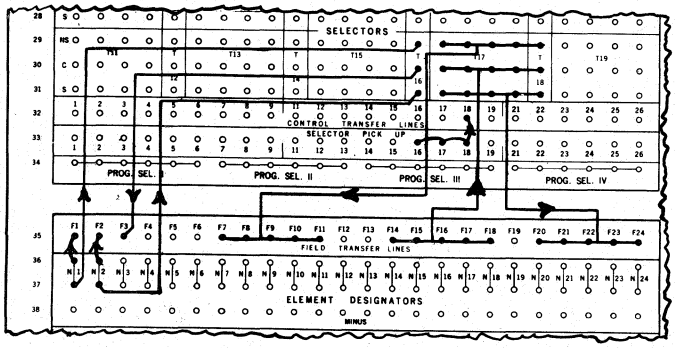
SYM.	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
A1				F1	F1	F1	F1	F1	F1	F3
				10	11	12	13	14	15	T17 T18
A2				F3	F2	F2	F2	F2	F2	F2
				T17 T18	17	18	19	20	21	22

On the connection panels, each position in an overlapping card column is wired through a transfer line to the common of a selector pole. The non-select side of the selector is wired through a transfer line to the corresponding position in accumulator input to be used when the control hole is not present. The select side of the selector is wired through a transfer line to the corresponding position in accumulator input to be used when the control hole is present.

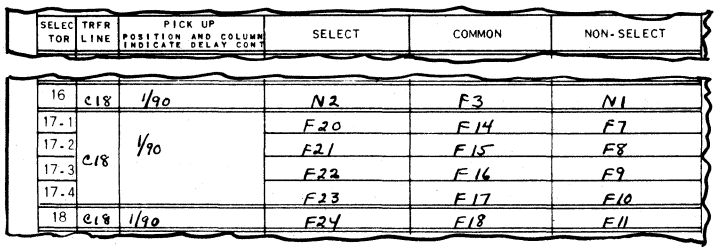
Therefore one selector pole and three transfer lines are used for each position in an overlapping card column. An additional transfer line and selector pole are required for the commons of the overlapping columns.

The wiring on the connection panel for the above would be:





The selector chart would be filled in as follows:



SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
16	C18	1/90	N2	F3	N1
17-1	C18	1/90	F20	F14	F7
17-2			F21	F15	F8
17-3			F22	F16	F9
17-4			F23	F17	F10
18	C18	1/90	F24	F18	F11

If two columns are overlapping, 10 selector poles using 30 transfer lines would be needed, plus one pole for the commons. The commons would be wired in the same way as the commons are wired when the total number of card columns is ten or less.

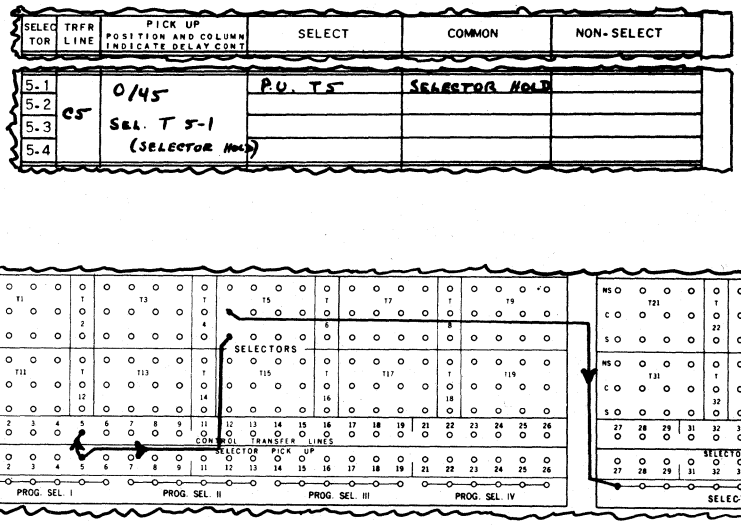
A note should be attached to the program chart showing the wiring of card positions and accumulator positions to the F-lines.

16. Wiring more than one program on a panel

Frequently the programs on a 60 or 120 are relatively short, leaving sufficient program steps for the wiring of additional problems on the same input-output panel. By careful programming it is often possible to develop programs which will require only the change of the wire from start to the first step of the routine to make one or the other program effective. Even this may be made automatic, dependent upon control punching in the card.

When combining programs, it is well to remember that it may be necessary to use the same storages, set instructions, clear instruction, etc. for both programs. It may therefore be necessary to change decimal locations and the storages associated with either set or clear. To prevent much rewiring when changing from one program to the other, this should probably be done through selectors.

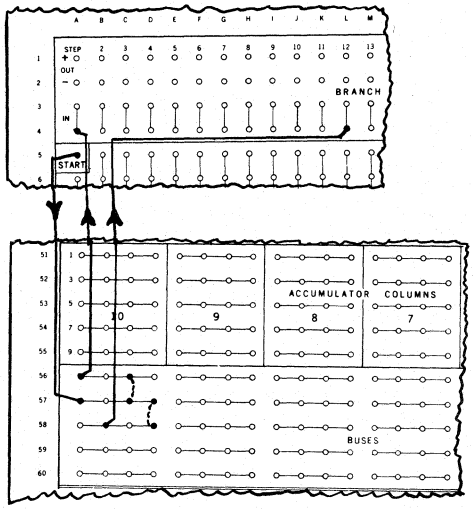
One method of controlling the selectors would be to precondition the computer for a run by use of a selector hold. To do this, a control hole not present in the other run should be used to pick up the selector through which the selector hold is wired. Since proper operating procedure would involve the clearing of the computer before each run, the control selector for breaking the connection from selector hold to the pick-up of the selector would be unnecessary. If selector T5, picked up by a "0" in column 45, is the one through which the choices will be made, the wiring would be as follows:



The choices would be made through the second, third, and fourth poles of the selector. One of the choices would be the step on which each routine starts.

It is also possible to direct one of the routines directly to a program select, which would pick up the necessary selectors. In this case the wire from start would be manually changed before beginning each routine.

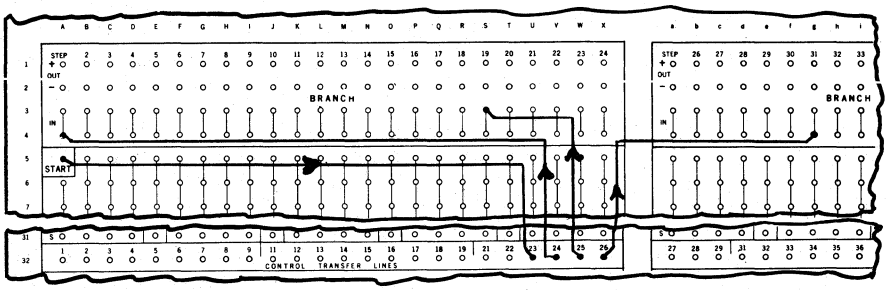
A method of wiring which is very simple for the operator to change is to wire start to an unused bus beneath the accumulator column section (lines 56-60). The bus beneath it is wired to one step, while the bus above it is wired to another. For example, suppose start may be wired to either step 1 or step 12:

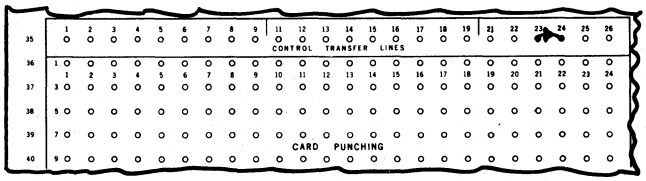


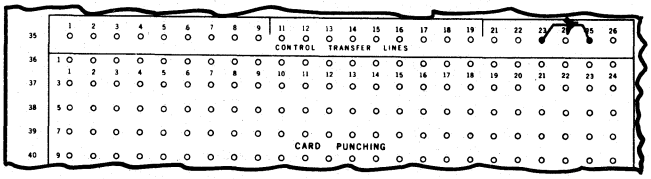
With this wiring, the only change necessary is the wiring of a jack in a position of the board which probably has very few wires. If this is not done, it is well to wrap the wire which is to be changed in colored tape, so that it can be easily identified.

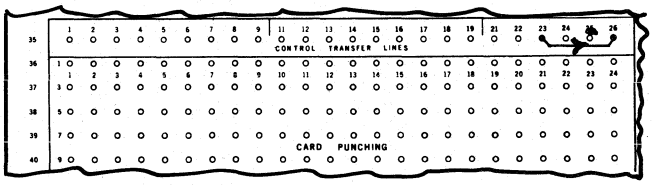
Sometimes several problems can be placed on a constant-program panel, yet each problem requires its own input-output panel. In such a case it is possible to wire the input-output panel to make the selection of the step on which each routine will begin.

Suppose that there are three routines in the same constant-program panel, one beginning on step 1, one on step 19, and one on step 31. The start hub would be wired to a control transfer line, C23 for example. The entry of each step which begins a routine would also be wired to a control transfer line - C24, C25, and C26 in this case. On the input-output panels C23 would be wired to the control transfer line which is wired to the first step of the routine associated with the panel. The wiring would be:









With this wiring the use of the proper input-output panel automatically controls the step on which the program will start.

17. Verification

As discussed earlier, the 60 and 120 automatically check each computation before continuing to the next step. Changes in voltage, intermittant failure of tubes, and similar sources of error are thereby prevented from affecting the accuracy of the computer. It may be assumed that the computation is correct.

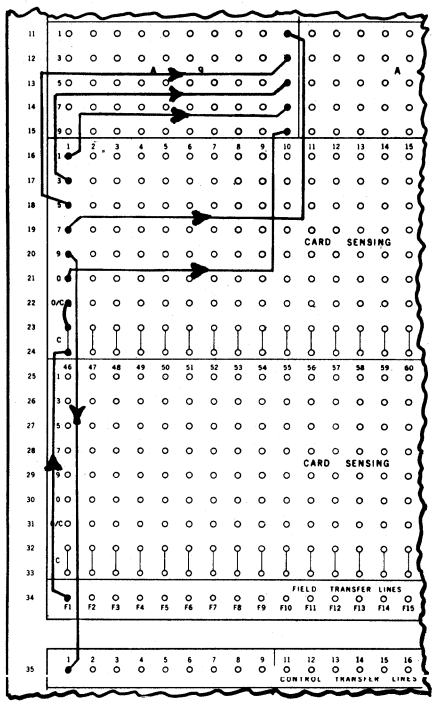
However, no machine is perfect, and it is possible for errors in sensing and punching to occur. This section deals with various methods of verification. The type which is used will depend entirely upon the problem involved.

Balancing to established controls is a method familiar to tabulating personnel. Frequently the use of controls established prior to or subsequent to a computer run, in conjunction perhaps with crossfooting of totals from the run, will be sufficient verification.

If it is necessary to rerun the cards a second time to insure accuracy, several methods are available. The best and most complete method is inserting the cards so that the "9" edge is the leading edge, and column 1 is where column 90 is usually located. Although this method becomes rather complicated from a wiring viewpoint, it completely changes the sensing switches through which each column is read, and thereby provides a complete check on sensing.

When the cards are reversed in this way, a "9" is read as a "0", a "7" as a "1", and so forth. Therefore both the zero common and the common of each column must be wired. It is also necessary to reverse the wiring in the card sensing section. Instead of using the 5-prong wires which are normally used to wire this portion of the board, each position must be wired separately.

As an example, assume that column 1, in which the zero position acts as a negative control, is to be element N1, transferred on F1. The negative control is to be transferred on C1. The wiring would be as follows:

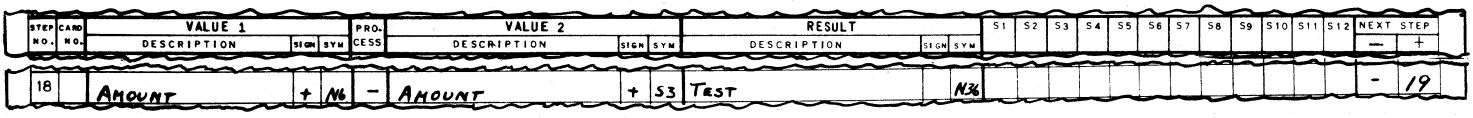


In order to check the punching, it is necessary to read as elements the values which have been punched on the first run. The calculation is done a second time, and the second result compared with the first result. This can be done in one of two ways. Suppose the amount calculated on the first run is now element N6, and the amount as calculated the second time is in storage S3. The two can be compared with two steps as follows:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
18		AMOUNT (PCH)	+	N6	−	AMOUNT (ST)	+	S3	TEST + MAY BE OK − ERROR	±	S1	18												40	19
19		AMOUNT (ST)	+	S3	−	AMOUNT (PCH)	+	N6	TEST + OK − ERROR	±	S1	19												40	20

If the result is negative on either step, the two values are not identical. This is the same method of testing as was used in checking the designating information, page 117. Note that in this example the negative branching is wired to step 40. By doing this, the computer will stop with a zero ÷ zero light lit. The operator can go through the program step by step using the test panel to determine the step on which the error occurred.

The second method of testing requires only one step for each value to be checked. It involves the use of zero as the result of the step. The basic use of this technique was covered on page 116. In this example the step would be:



In both cases it is well to attach a list of the testing steps to the program, with an indication of which value is being verified. This will simplify the work of the operator when an error does occur.

Another method of verification that is sometimes considered is the tipping over of the cards so that the zero edge is still leading, but column 1 is where column 45 is normally located. This method has certain disadvantages:

1. Columns 23 and 68 are still in the same location, and therefore the sensing of these columns is not verified.

2. The warp of the cards may make feeding more difficult.

3. In a multiple card routine, information which should be read from the first card of a group cannot be read until the last card, since the cards are in reverse sequence.

4. Overlapping machine operations are difficult, since the card which would normally go in the tabulator or other machine first now becomes the last card through the computer. The next run of the cards therefore cannot begin until the verification run has been completed.

Although punching could be verified by reading the cards in the normal way, recalculating, and checking the results in the same way as the first type of machine verification mentioned, this method does not provide any checking of sensing. If special verification boards are to be wired, it is well to verify as much as possible, which would include sensing.

18. Card code checking

A. DESCRIPTION

In some applications it may be desirable to check card codes to ascertain whether certain conditions have been met - for example whether a particular

card is present, whether the cards are in sequence by card code within a particular group, or whether there are duplications. Several examples of this type of checking are given. For purposes of comparison it is assumed that there are four types of cards in each run, with each type of card containing a single position code. Using these examples as a basis, however, the programmer can adapt these principles to fit other problems.

In all cases, two four-pole selectors are needed for each type of card. One program select is also required, and incorrect cards are directed to step 40, zero divided by zero, to stop the computer.

Note that this check does not test the designating information, but merely the card codes. Therefore the usual steps for testing designating information (page 117) should be included, usually before the card code test. In the following examples it will be assumed that all cards enter the card code test as of the plus branching of step 2. In actual programming the first card of a group could enter either from the select side of its selector at the plus branching of step 1, or following a testing step which identified the first card of the group and the step which stored the designating information.

In the following examples, the card codes used and the steps to which each will be directed will be:

Code (column 46)	Card Sequence	To Step
1	1st Card	4
3	2nd Card	12
5	3rd Card	21
7	4th Card	31

B. ALL CARDS PRESENT, IN ORDER, WITH NO DUPLICATIONS

The conditions of this card code check are:

1) All cards must be present
2) All cards must be in order
3) There may be no duplications

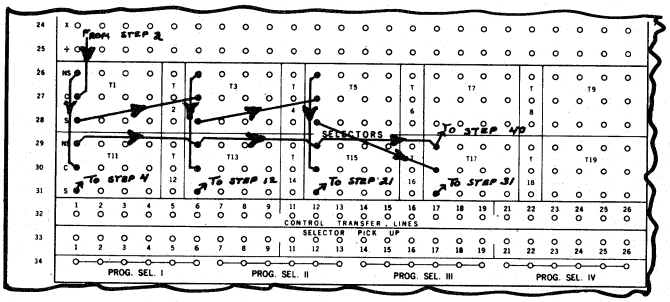
The types of cards are associated with selectors as follows:

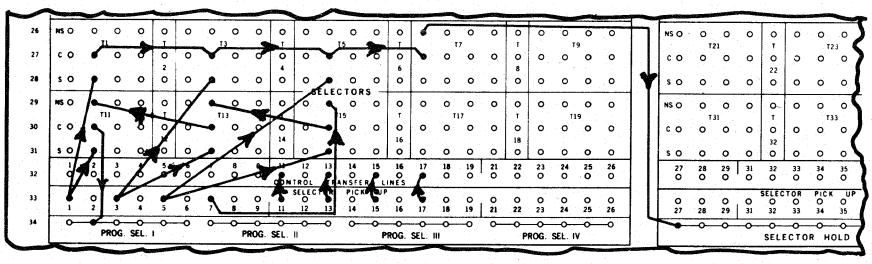
Card Code	Picks Up	Related To
1	T 11	T 1
3	T 13	T 3
5	T 15	T 5
7	T 17	T 7

The pick-up shown above is wired directly, and therefore the selector drops out at the end of the card. The related selector is picked up by PS 1 during the program for the card, and is held select by selector hold until the last

card of the group, when all selectors are returned to non-select. PS 1 is impulsed in every card, and directed through selectors to the selector which it is to pick up in the card which is present. The impulsing of PS 1 should be done in each card immediately before the wiring to trip.

The first diagram below shows the wiring of the step branching, while the second diagram shows the method of picking up the selectors. The two diagrams are consolidated on the selector chart which follows:





SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
11-1	C11	1/46	STEP 4	NS T1-1	STEP 40
11-2			P.U. T1	PS1 GROUND	COM. T 13-2
11-3					
11-4					
12					
13-1	C13	3/46	STEP 12	NS T 3-1	STEP 40
13-2			P.U. T3	NS T11-2	COM. T 15-2
13-3					
13-4					
14					
15-1	C15	5/46	STEP 21	N.S. T 5-1	STEP 40
15-2			P.U. T5	NS T 13-2	P.U. T7
15-3					
15-4					
16					
17-1	C17	7/46	STEP 31	SEL. T 5-1	STEP 40
17-2					
17-3					
17-4					

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1		SEL. T11-2 (PS 1) SEL. T1-2 (SEL. HOLD)	COM. T3-1	+ BR. STEP 2	COM. T11-1
1-2			P.U. T1	COM. T 7-1	
1-3					
1-4					
2					
3-1		SEL. T13-2 (PS 1) SEL. T3-2 (SEL. HOLD)	COM. T5-1	SEL. T1-1	COM. T13-1
3-2			P.U. T3	COM. T 7-1	—
3-3					
3-4					
4					
5-1		SEL. T15-2 (PS 1) SEL. T5-2 (SEL. HOLD)	COM. T17-1	SEL. T3-1	COM T15-1
5-2			P.U. T5	COM. T7-1	—
5-3					
5-4					
6					
7-1		N.S. T 15-2 (PS 1)	—	COM T1-2, T3-2, T5-2	SELECTOR HOLD
7-2					
7-3					
7-4					

B-1 First card

Branching: When the first card of a group enters the computer, all selectors are non-select except the one picked up by the card code. This should be T11. The plus branching of step 2 is wired through the non-select of T1-1 to the common of T11-1. If T11 is non-select the card is not coded 1, and is routed to step 40. If T11 is select, the program continues with step 4.

Selector pick-up: When PS 1 is impulsed, it picks up T1 through the select of T11-2. Although PS 1 drops out at the trip signal for the card, T1 remains select because selector hold is wired to the pick-up of T1 through the non-select of T 7-1 and the select of T1-2. T11 becomes non-select at trip time because it is card controlled.

B-2 Second card

Branching: When the second card enters the computer, the only selectors which are select are T1 and the selector picked up by the card code. This should be T13. The plus branching of step 2 is wired through the select of T1-1 and non-select of T3-1 to the common of T13-1. If T13 is non-select, the card is not coded 3, and is routed to step 40. If T13 is select, the program continues with step 12.

Selector pick-up: When PS 1 is impulsed, it picks up T3 through the non-select of T11-2 and the select of T13-2. It remains select after the card is tripped out because selector hold is wired to the pick-up of T3 through the non-select of T7-1 and the select of T3-2. T13 drops out at trip time because it is card controlled.

B-3 Third card

Branching: When the third card enters the computer, the only selectors which are select are T1, T3, and the selector picked up by the card code. This should be T15. The plus branching of step 2 is wired through the select of T1-1, the select of T3-1, and the non-select of T5-1 to the common of T15-1. If T15 is non-select, the card is not coded 5, and is routed to step 40. If T15 is select, the program continues with step 21.

Selector pick-up: When PS 1 is impulsed, it picks up T5 through the non-select of T11-2 and T13-2, and the select of T15-2. It remains select after the card is tripped out because selector hold is wired to the pick-up of T5 through the non-select of T7-1 and the select of T5-2. T15 drops out at trip time because it is card controlled.

B-4 Fourth card

Branching: When the fourth card enters the computer, T1, T3, T5 and the selector picked up by the card code are select; the others are non-select. The card controlled selector should be T17. The plus branching of step 2 is wired through the select of T1-1, T3-1, and T5-1 to the common of T17-1. Note that since this is the last card, it is unnecessary to wire the branching through the non-select of T7-1. If T17 is non-select, the card is not coded 7, and is routed to step 40. If T17 is select, the program continues with step 31.

Selector pick-up: When PS 1 is impulsed, it picks up T7 through the non-select of T11-2, T13-2, and T15-2. When T7 becomes select, the connection from selector hold through the non-select of T7-1 to the pick-up of T1, T3, and T5 is broken. These selectors therefore return to non-select. T7 will drop out at the end of the card because it is picked up by a program select, and T17 drops out because it is card controlled. At trip time, therefore, the computer is ready for the next group of cards with all selectors non-select.

C. ALL CARDS PRESENT, IN ORDER, WITH DUPLICATIONS EXCEPT THE LAST

The conditions of this card code check are:

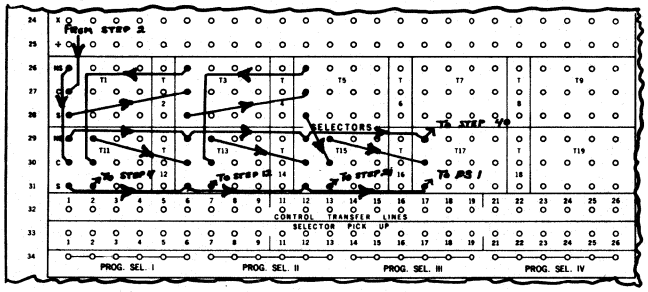
1) All cards must be present
2) All cards must be in order
3) There may be duplications except for the last card

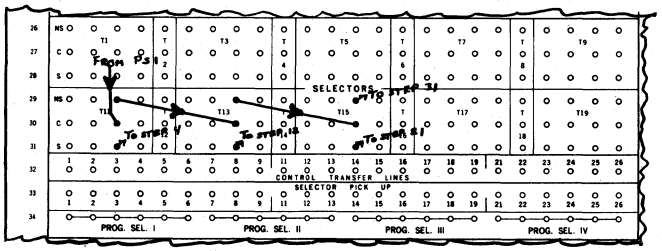
The types of cards are associated with selectors as follows:

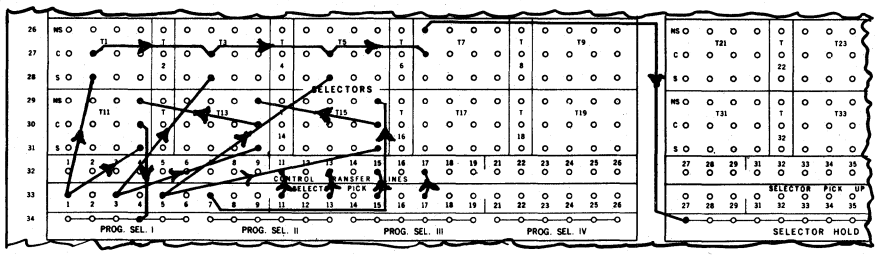
Card Code	Picks Up	Related To
1	T11	T1
3	T13	T3
5	T15	T5
7	T17	T7

The pick-up shown above is wired directly, and therefore the selector drops out at the end of the card. The related selector is picked up by PS 1 during the program for the first card of each type, but not in duplicate cards. A selector picked up by PS 1 is held select by selector hold until the last card of the group, when all selectors are returned to non-select.

The first diagram below shows the wiring of the step branching. The second diagram shows the branching of PS 1, which in this routine must be controlled through selectors. The third diagram shows the method of picking up the selectors. The three diagrams are consolidated on the selector chart which follows:







SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
11-1	C11	1/46	To PS1	NS T1-1	STEP 40
11-2			STEP 4	NS T3-1	COM T13-1
11-3			STEP 4	FROM PS1	COM. T13-3
11-4			P.U. T1	PS1 GROUND	COM. T13-4
12					
13-1	C13	3/46	To PS1	NS T11-2	STEP 40
13-2			STEP 12	NS T5-1	COM T15-1
13-3			STEP 12	NS T 11-3	COM T15-3
13-4			P.U. T3	NS T 11-4	COM T15-4
14					
15-1	C15	5/46	To PS1	NS T13-2	STEP 40
15-2			STEP 21	SEL. T5-1	COM. T17-1
15-3			STEP 21	NS T13-3	STEP 31
15-4			P.U. T5	NS T13-4	P.U. T7
16					
17-1	C17	7/46	To PS1	NS T15-2	STEP 40
17-2					
17-3					
17-4					

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1		SEL. T11-4 (PS 1)	COM. T 3-1	+ BR. STEP 2	COM. T11-1
1-2			P.U. T1	COM T 7-1	—
1-3		SEL. T1-2 (SEL. HOLD)			
1-4					
2					
3-1		SEL. T13-4 (PS 1)	COM. T5-1	SEL. T1-1	COM. T11-2
3-2			P.U. T3	COM. T 7-1	—
3-3		SEL. T3-2 (SEL. HOLD)			
3-4					
4					
5-1		SEL. T15-4 (PS1)	COM. T15-2	SEL. T 3-1	COM. T13-2
5-2			P.U. T5	COM. T 7-1	—
5-3		SEL. T5-2 (SEL. HOLD)			
5-4					
6					
7-1		NS T 15-4 (PS 1)	—	COM. T1-2, T3-2, T5-2	SELECTOR HOLD
7-2					
7-3					
7-4					

C-1 First card

Branching: When the first card of a group enters the computer, all selectors are non-select except the one picked up by the card code. This should be T11. The plus branching of step 2 is wired through the non-select of T1-1 to the common of T11-1. If T11 is non-select the card is not coded 1 and is routed to step 40. If T11 is select, the program continues with PS 1. From PS 1 the program is routed through the select of T11-3 to step 4.

Selector pick-up: When PS 1 is impulsed, it picks up T1 through the select of T11-4. T1 remains select after the card is tripped out because selector hold is wired to the pick-up of T1 through the non-select of T7-1 and the select of T1-2. T11 becomes non-select at trip time because it is card controlled.

C-2 Duplicates of first card or second card

The second card which enters the computer may be coded in one of three ways:

1) 1 in column 46 - a duplicate of the first card.
2) 3 in column 46 - the first card of the second type.
3) Not 1 or 3 in column 46 - error.

When this card enters the computer, the only selectors which are select are T1 and the selector picked up by the card code. This should be T11 if it is a duplicate, or T13 if it is the first of the second type.

Branching: The plus branching of step 2 is wired through the select of T1-1 and the non-select of T3-1 to the common of T11-2.

1) If T11 is select, the card is a duplicate of the first type. The select of T11-2 is wired to step 4.

2) If T11 is non-select, the card may be the first card of the second type. The non-select of T11-2 is wired to the common of T13-1. If T13 is select, the card is the first of the second type, and the select of T13-1 is wired to PS 1. From PS 1 the card is routed through the non-select of T11-3 and the select of T13-3 to step 12.

3) If neither T11 nor T13 is select, the card is an error: through the non-select of T11-2 and T13-1 it is directed to step 40.

Selector pick-up:
1) Since PS 1 is not picked up if the card is a duplicate, the selectors remain unchanged except that T11 drops out at trip time because it is card controlled. They will remain unchanged as long as cards which are duplicates of the first enter the computer.

2) If the card is the first of the second type, it is directed to PS 1. PS 1 picks up T3 through the non-select of T11-4 and the select of T13-4. T3 remains select after the card is tripped out because selector hold is wired to the pick-up of T3 through the non-select of T7-1 and the select of T3-2. T13 drops out at trip time because it is card controlled.

3) If the card is incorrectly coded, the computer is stopped; therefore there is no change to the selectors.

C-3 Duplicates of second card or third card

The card which enters the computer after the first card of the second type may be coded in three ways:

1) 3 in column 46 - a duplicate of the second type.
2) 5 in column 46 - the first card of the third type.
3) Not 3 or 5 in column 46 - error.

When this card enters the computer, the only selectors which are select are T1 and T3, and the selector picked up by the card code. This should be T13 if it is a duplicate, or T15 if it is the first card of the third type.

Branching: The plus branching of step 2 is wired through the select of T1-1 and T3-1 and the non-select of T5-1 to the common of T13-2.

1) If T13 is select, the card is a duplicate of the second type. The select of T13-2 is wired to step 12.

2) If T13 is non-select, the card may be the first card of the third type. The non-select of T13-2 is wired to the common of T15-1. If T15 is select, the card is the first of the third type, and the select of T15-1 is wired to PS 1. From PS 1 the card is routed through the non-select of T11-3 and T13-3 and the select of T15-3 to step 21.

3) If neither T13 nor T15 is select, the card is an error; through the non-select of T13-2 and T15-1 it is directed to step 40.

Selector pick-up:
1) Since PS 1 is not picked up if the card is a duplicate, the selectors remain unchanged, except that T13 drops out at trip time because it is card controlled. They will remain unchanged as long as cards which are duplicates of the second enter the computer.

2) If the card is the first of the third type, it is directed to PS 1. PS 1 picks up T5 through the non-select of T11-4, T13-4, and the select of T15-4. T5 remains select after the card is tripped out because selector hold is wired to the pick-up of T5 through the non-select of T7-1 and the select of T5-2. T15 drops out at trip time because it is card controlled.

3) If the card is incorrectly coded the computer is stopped; therefore there is no change to the selectors.

C-4 Duplicates of third card or fourth card

The card which enters the computer after the first card of the third type may be coded in three ways:

1) 5 in column 46 - a duplicate of the third type.
2) 7 in column 46 - the only card of the fourth type.
3) Not 5 or 7 in column 46 - error.

When this card enters the computer, T1, T3, T5, and the selector picked up by the card code are select; the others are non-select. The card controlled selector should be T15 if it is a duplicate, or T17 if it is the fourth card.

Branching: The plus branching of step 2 is wired through the select of T1-1, T3-1, and T5-1 to the common of T15-2.

1) If T15 is select, the card is a duplicate of the third type. The select of T15-2 is wired to step 21.

2) If T15 is non-select the card may be the fourth card. The non-select of T15-2 is wired to the common of T17-1. If T17 is select, the card is the fourth card and the select of T17-1 is wired to PS 1. From PS 1 the card is routed through the non-select of T11-3, T13-3, and T15-3 to step 31.

3) If neither T15 nor T17 is select, the card is an error; through the non-select of T15-2 and T17-1 it is directed to step 40.

Selector pick-up:
1) Since PS 1 is not picked up if the card is a duplicate, the selectors remain unchanged, except that T15 drops out at trip time because it is card controlled. They will remain unchanged as long as cards which are duplicates of the third enter the computer.

2) If this is the fourth card, it is directed to PS 1. PS 1 picks up T7 through the non-select of T11-4, T13-4, and T15-4. When T7 becomes select, the connection from selector hold through the non-select of T7-1 to the pick-up of T1, T3 and T5 is broken. These selectors therefore return to non-select. T7 will drop out at the end of the card because it is picked up by a program select, and T17 drops out because it is card controlled. At trip time, therefore, the computer is ready for the next group of cards with all selectors non-select.

3) If the card is incorrectly coded the computer is stopped; therefore there is no change to the selectors.

D. LAST CARD MUST BE PRESENT, OTHERS MAY OR MAY NOT BE PRESENT, ALL CARDS IN ORDER, NO DUPLICATION

The conditions of this card code check are:

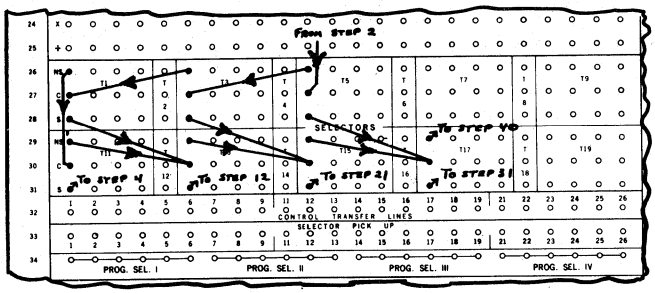
1) The last card of each group must be present.
2) Other cards may or may not be present.
3) The cards must be in order.
4) There may be no duplication.

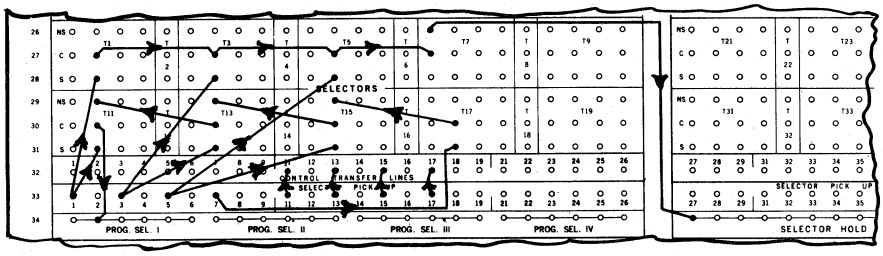
The types of cards are associated with selectors as follows:

Card Code	Pick-up	Related to
1	T11	T1
3	T13	T3
5	T15	T5
7	T17	T7

The pick-up shown above is wired directly, and therefore the selector drops out at the end of the card. The related selector is picked up by PS 1 during the program for the card, and held select by selector hold until the last card of the group, when all selectors are returned to non-select. PS 1 is impulsed in every card, and directed through selectors to the selector which it is to pick up in the card which is present. This impulsing of PS 1 is done in each card immediately before the wiring to trip.

The first diagram below shows the wiring of the step branching, while the second diagram shows the method of picking up the selectors. The two diagrams are consolidated on the selector chart which follows:





SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
11-1					
11-2	C11	1/46	STEP 4	NS T1-1	COM. T13-1
11-3			P.U. T1	PS1 GROUND	COM T13-2
11-4					
12					
13-1			STEP 12	SEL. T1-1, NS T11-1	COM. T15-1
13-2	C13	3/46	P.U. T3	NS T11-2	COM. T15-2
13-3					
13-4					
14					
15-1			STEP 21	SEL. T3-1, NS T13-1	COM. T17-1
15-2	C15	5/46	P.U. T5	NS T13-2	COM. T17-2
15-3					
15-4					
16					
17-1			STEP 31	SEL. T5-1, NS T15-1	STEP 40
17-2	C17	7/46	P.U. T7	NS T15-2	—
17-3					
17-4					

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1					
1-2		SEL. T11-2 (PS 1)	COM. T13-1	NS T3-1	COM. T11-1
1-3		SEL. T1-2 (SEL. HOLD)	P.U. T1	COM. T7-1	—
1-4					
2					
3-1					
3-2		SEL. T13-2 (PS 1)	COM. T15-1	NS T5-1	COM. T1-1
3-3		SEL. T3-2 (SEL. HOLD)	P.U. T3	COM. T7-1	—
3-4					
4					
5-1					
5-2		SEL. T15-2 (PS 1)	COM. T17-1	+ BR. STEP 2	COM. T3-1
5-3		SEL. T5-2 (SEL. HOLD)	P.U. T5	COM. T7-1	—
5-4					
6					
7-1					
7-2		SEL. T17-2 (PS 1)	—	COM T1-2, T3-2, T5-2	SELECTOR HOLD
7-3					
7-4					

The principle of the loop is that as soon as a particular code has been accepted by the computer, only a higher code may be accepted thereafter. Any duplicate code or lower code will stop the computer.

D-1 First card

Branching: When the first card of a group enters the computer, all selectors are non-select except the one picked up by the card code. This may be T11, T13, T15, or T17. The plus branching of step 2 is wired through the non-select of T5-1, T3-1, and T1-1 to the common of T11-1.

If T11 is select, the card is code 1 card, and is directed to step 4 through the select of T11-1. The non-select of T11-1 is wired to the common of T13-1.

If T13 is select the card is a code 3 card and is directed to step 12 through the select of T13-1. The non-select of T13-1 is wired to the common of T15-1.

If T15 is select the card is a code 5 card and is directed to step 21 through the select of T15-1. The non-select of T15-1 is wired to the common of T17-1.

If T17 is select the card is a code 7 card, and is directed to step 31 through the select of T17-1. If T17 is non-select, none of the codes are punched and the card is directed to step 40.

Selector pick-up: PS 1 is impulsed immediately before trip. PS 1 ground is wired to the common of T11-2.

If T11 is select T1 is picked up. It is held from card to card by selector hold, which is wired through the non-select of T7-1 and the select of T1-2. The non-select of T11-2 is wired to the common of T13-2.

If T13 is select, T3 is picked up. It is held from card to card by selector hold, which is wired through the non-select of T7-1 and the select of T3-2. The non-select of T13-2 is wired to the common of T15-2.

If T15 is select, T5 is picked up. It is held from card to card by selector hold, which is wired through the non-select of T7-1 and the select of T5-2. The non-select of T15-2 is wired to the common of T17-2.

If none of the other selectors are select, T17 must be select or the computer would have stopped following the branching of step 2. PS 1 picks up T17 through the non-select of T11-2, T13-2, T15-2, and the select of T17-2. When a code 7 card is the only card of the group this is of no importance, since its purpose is to drop out any selectors picked up in previous cards. When this card is tripped out, all selectors will be non-select, ready for the next group of cards.

D-2 Second card

Branching: When the second card of a group enters the computer, T1, T3, or T5 will be select, as well as the selector which is picked up by card control. This could be T13, T15, or T17, but should not be T11. Since one card of the group has already gone through the computer, a card coded 1 would be either a duplication or in descending sequence.

The plus branching of step 2 is wired to the common of T5-1. If T1 has been picked up by the preceding card, it is routed through the non-select of T5-1 and T3-1 and the select of T1-1 to the common of T13-1. From there it is routed to step 12, step 21, or step 31 depending on whether T13, T15 or T17 is select. If none of them are select, the computer is stopped on step 40.

If T3 has been picked up by the preceding card, the plus branching of step 2 is routed through the non-select of T5-1 and the select of T3-1 to the common of T15-1. Unless this card is coded 5 or 7, picking up T15 or T17, the computer will stop on step 40.

If T5 has been picked up by the preceding card, the plus branching of step 2 is routed through the select of T5-1 to the common of T17-1. Unless this card is coded 7, picking up T17, the computer will stop on step 40.

Selector pick-up: When PS 1 is impulsed, PS 1 ground is routed through the non-select of T11-2 and the select of T13-2, T15-2 or T17-2, whichever is picked up by card control, to pick up the related selector. This selector is held by selector hold unless the card is coded 7. If T7 becomes select, it breaks the connection between selector hold and the selector which was being held select, thereby returning the selector to non-select. The computer is therefore ready for the next group of cards.

D-3 Third card

Branching: When the third card of a group enters the computer, two of the three selectors T1, T3 and T5 will be select, as well as the selector which is picked up by card control. This could be T15 or T17, but should not be T11 or T13. Since two cards of the group have already gone through the computer, a card coded 1 or 3 would be either a duplication or in descending sequence.

The plus branching of step 2 is routed through the select of T3-1 or T5-1 to T15-1 or T17-1, depending on whether the preceding card was coded 3 or 5. If T15 is select the program continues with step 21; if T17 is select the program continues with step 31. If neither is select, the computer stops on step 40.

Selector pick-up: When PS 1 is impulsed, PS 1 ground is routed through the non-select of T11-2 and T13-2 and the select of T15-2 or T17-2 to pick up the related selector. If this is T5, it is held by selector hold. If it is T7, it breaks the connection between selector hold and the selectors which were being held select, thereby returning them to non-select. The computer is therefore ready for the next group of cards.

D-4 Fourth card

Branching: When the fourth card of a group enters the computer, T1, T3 and T5 will be select, as well as the selector which was picked up by card control. This should be T17, but not T11, T13, or T15. Since three cards of the group have already gone through the computer, a card coded 1, 3 or 5 would be either a duplication or in descending sequence.

The plus branching of step 2 is wired through the select of T5-1 to the common of T17-1. If T17 is select, the program continues with step 31. If T17 is non-select, the computer stops on step 40.

Selector pick-up: When PS 1 is impulsed, PS 1 ground is routed through the non-select of T11-2, T13-2, and T15-2 to the common of T17-2. The select of T17-2 picks up T7. When T7 becomes select, the connection from selector hold through the non-select of T7-1 to the pick-up of selectors T1, T3, and T5 is broken; these selectors therefore become non-select. The computer is now ready for the next group of cards, with all selectors non-select at trip time.

E. LAST CARD MUST BE PRESENT, OTHERS MAY OR MAY NOT BE PRESENT, ALL CARDS IN ORDER, DUPLICATIONS EXCEPT THE LAST

The conditions of this card code check are:

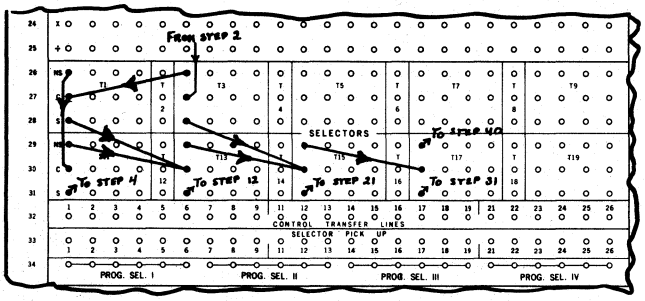
1) The last card must be present.
2) Other cards may or may not be present.
3) The cards must be in order.
4) There may be duplications except the last.

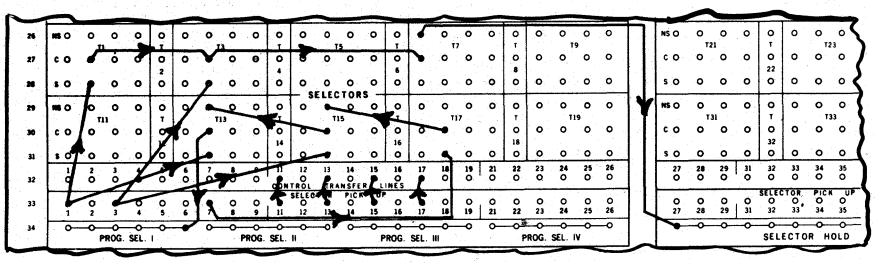
The types of cards are associated with selectors as follows:

Card Code	Picks Up	Related To
1	T11	
3	T13	T1
5	T15	T3
7	T17	T7

The pick-up shown above is wired directly, and therefore the selector drops out at the end of the card. The related selector is picked up by PS 1 during the program for the card, and held select by selector hold until the last card of the group, when all selectors are returned to non-select. PS 1 is impulsed in every card, and directed through selectors to the selector which it is to pick up in the card which is present. The impulsing of PS 1 is done in each card immediately before the wiring to trip.

The first diagram below shows the wiring of the step branching, while the second diagram shows the method of picking up the selectors. The two diagrams are consolidated on the selector chart which follows:





SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
11-1	C11	1/46	STEP 4	NS T1-1	COM. T13-1
11-2					
11-3					
11-4					
12					
13-1	C13	3/46	STEP 12	SEL. T1-1, NS T11-1	COM. T15-1
13-2			P.U. T1	PS1 GROUND	COM. T15-2
13-3					
13-4					
14					
15-1	C15	5/46	STEP 21	SEL. T3-1, NS T13-1	COM. T17-1
15-2			P.U. T3	NS T13-2	COM. T17-2
15-3					
15-4					
16					
17-1	C17	7/46	STEP 31	NS T15-1	STEP 40
17-2			P.U. T7	NS T15-2	—
17-3					
17-4					

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1		SEL. T13-2 (PS 1)	COM. T13-1	NS T3-1	COM. T11-1
1-2			P.U. T1	COM. T7-1	—
1-3		SEL. T1-2 (SEL. HOLD)			
1-4					
2					
3-1		SEL. T15-2 (PS 1)	COM. T15-2	+ BR. STEP 2	COM. T1-1
3-2			P.U. T3	COM. T7-1	—
3-3		SEL. T3-2 (SEL. HOLD)			
3-4					
4					
5-1					
5-2					
5-3					
5-4					
6					
7-1		SEL. T17-2 (PS 1)	—	COM. T1-2, T3-2	SELECTOR HOLD
7-2					
7-3					
7-4					

The principle of the loop is that as soon as a particular code has been accepted by the computer, either duplicates or higher codes will be accepted, but no lower codes.

E-1 First card

Branching: When the first card of a group enters the computer, all selectors are non-select except the one picked up by the card code. This may be T11, T13, T15, or T17. The plus branching of step 2 is wired through the non-select of T3-1 and T1-1 to the common of T11-1. As in the preceding example of card code testing, the branching is routed to step 4, step 12, step 21, or step 31, depending on which of the card code selectors is select. If none are select, the card is routed to step 40.

Selector pick-up: PS 1 is impulsed immediately before trip. PS 1 ground is wired to the common of T13-2.

If the card is a code 1 card, selectors T13, T15, and T17 are non-select; therefore no additional selectors are picked up. The computer will accept any of the four codes which may be punched in the next card.

If the card is a code 3 card, PS 1 ground is routed through the select of T13-2 to pick up T1. T1 is held select from card to card by selector hold, which is wired through the non-select of T7-1 and the select of T1-2 to the pick-up of T1.

If the card is a code 5 card, PS 1 ground is routed through the non-select of T13-2 and the select of T15-2 to the pick-up T3. T3 is held select from card to card by selector hold, which is wired through the non-select of T7-1 and the select of T3-2 to the pick-up of T3.

If the card is a code 7 card, PS 1 ground is routed through the non-select of T13-2 and T15-2 and the select of T17-2 to pick up T7. Since this is the

first card of the group and no selectors are being held from a previous card, this is of no importance. When this card is tripped out, all selectors will be non-select, ready for the next card.

E-2 Second card

The second card which enters the computer may be coded in one of three ways:

1. A duplicate of the first card
2. A higher order card
3. A lower order card - error

When this card enters the computer, the card code selector will become select. If the preceding card was coded 1, no others will be select. If the preceding card was coded 3, T1 will be select. If the preceding card was coded 5, T3 will be select.

Branching: The plus branching of step 2 is wired to the common of T3-1.

If the preceding card was coded 1, the branching is routed through the non-select of T3-1 and T1-1 to the common of T11-1. It proceeds to step 4, step 12, step 21, or step 31, depending on which of the card code selectors is select.

If the preceding card was coded 3, the branching is routed through the non-select of T3-1 and the select of T1-1 to the common of T13-1. From there it is routed to step 12, step 21, or step 31, depending on whether T13, T15, or T17 is select. If there is not a 3, 5, or 7 punched in the card, the computer will stop on step 40.

If the preceding card was coded 5, the branching is routed through the select of T3-1 to the common of T15-1. From there it is routed to step 21 or step 31, depending on whether T15 or T17 is select. If neither a 5 nor a 7 is punched in the card, the computer will stop on step 40.

Selector pick-up: PS 1 is impulsed before trip. If the card is a code 1, 3, or 5 card, the same action will take place as described above in connection with the first card. If the card is a code 7 card, the pick-up of T7 will drop out any selectors which might have been picked up on the first card, since this action breaks the connection from selector hold to the pick-up of the selectors. When this card is tripped out, therefore, all selectors will be non-select, ready for the next group.

E-3 Remaining cards

In all succeeding cards the branching and selector pick-up will occur as described for the second card.

19. Accumulating positive and negative values in one storage

It is possible in a program in which there are insufficient storages but ample steps, to accumulate two values, either of which may be positive or negative, in the same storage unit. The program requires additional storages before the values are set for punching, but these may be available at set time although not earlier.

The total of neither of the fields to be accumulated may exceed four digits. Both may be assigned a normal negative control; in the following example, it is the zero position in the column to the left of the card field. If the fields are placed in the extreme right in accumulator input, the value which will accumulate on the left side of the storage is assigned a 1/0 decimal location, while the value which will accumulate on the right side of the storage is assigned a 6/5 decimal location. The storage in which accumulation takes place, as well as two working storages, are assigned 6/5 decimal locations. A third working storage is assigned a 1/0 decimal location. Note that the decimal locations of these storages may be changed from a previous location to 6/5 by selectors picked up either by card control or a program select.

The element sections of the program charts, including the necessary constants, would be filled in as follows:

TABLE OF FACTORS

SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE
N1				N13				N25			
N2	±	D	QUANTITY 1	N14				N26			
N3	±	D	QUANTITY 2	N15				N27			
N4				N16				N28			
N5				N17				N29			
N6				N18				N30	+		.2
N7				N19				N31	+		.1
N8				N20				N32	+		20000.
N9				N21				N33	+		10000.
N10				N22				N34	+		10000.10000
N11				N23				N35			
N12				N24				N36	+		0.000

CARD DESCRIPTION D= DETAIL CARDS S= SUMMARY CARDS

ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD-F1, F2, ETC: CONTROL - C1, C2, ETC.) DECIMAL LOCATION. NEGATIVE CONTROL AND TRANSFER LINE. INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N1															
N2	F2	33	0	C2	1/0							33	34	35	36
N3	F3	37	0	C3	4/5							37	38	39	40
N4															
N5															
N6															
N7															
N8															
N9															
N10															
N11															
N12															
N13															
N14															
N15															
N16															
N17															
N18															

ELEM. DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															
N22															
N23															
N24															
N25															
N26															
N27															
N28															
N29															
N30					2/1										2
N31					2/1										1
N32					1/0						2	0	0	0	0
N33					1/0						1	0	0	0	0
N34					4/5	1	0	0	0	0	1	0	0	0	0
N35															
N36					4/3							0	0	0	0

ACCUMULATOR - INPUTS

TOP LINE - TRANSFER LINE (F1, F2, ETC.) INDICATE:
BOTTOM LINE - CARD COLUMNS

SYM.	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
A1										
A2							F2	F2	F2	F2
							33	34	35	36
A3							F3	F3	F3	F3
							37	38	39	40
A4										
A5										
A6										
A7										
A8										
A9										
A10										
A11										
A12										

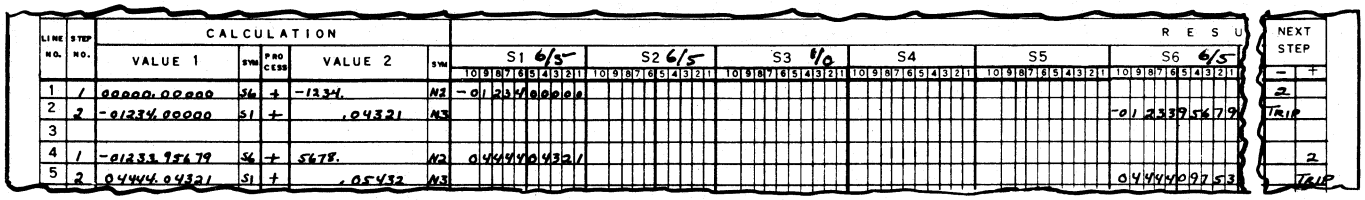
The accumulation of the two values in the detail cards may be done in the following two steps:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	VALUE 1 SIGN	VALUE 1 SYM	PROCESS	VALUE 2 DESCRIPTION	VALUE 2 SIGN	VALUE 2 SYM	RESULT DESCRIPTION	RESULT SIGN	RESULT SYM	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
1	D	Σ QTY 1 + Σ QTY 2	±	S6	+	QTY. 1	±	N2	NEW Σ QTY 1 + QTY 2	±	S1	1												2	2
2	D	NEW Σ QTY 1 + QTY 2	±	S1	+	QTY. 2	±	N3	NEW Σ QTY 1 + QTY 2	±	S6						(2)							TRIP	TRIP

Four planning charts follow, each of which will be carried forward throughout the program. They show the various possibilities of positive and negative combinations. The charts immediately following show the accumulation of two detail cards. The headings indicate the signs of the two values when the detail accumulation has been completed.

1. Both values are positive.

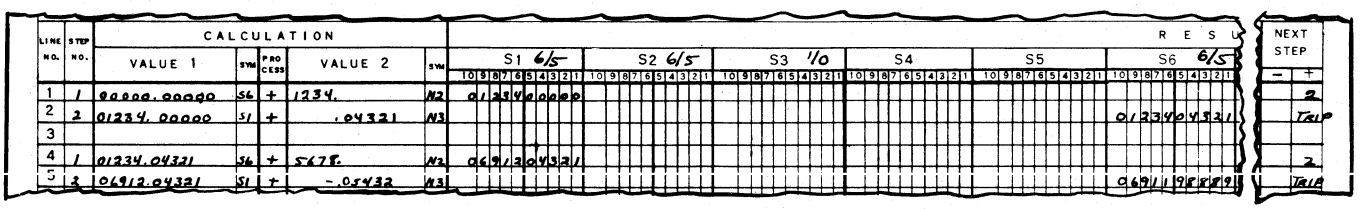
	Left value	Right value
Card #1	-1234	+4321
Card #2	+ +5678	+ +5432
	+4444	+9753



LINE NO.	STEP NO.	CALCULATION VALUE 1	SYM	PROCESS	VALUE 2	SYM	RESULT S1 6/5	S2 6/5	S3 1/0	S4	S5	S6 6/5	NEXT STEP −	NEXT STEP +
1	1	00000.00000	S6	+	-1234.	N2	-01234.00000							2
2	2	-01234.00000	S1	+	.04321	N3						-01233.95679	TRIP	
3														
4	1	-01233.95679	S6	+	5678.	N2	04444.04321							2
5	2	04444.04321	S1	+	.05432	N3						04444.09753		TRIP

2. Left value is positive, right value is negative.

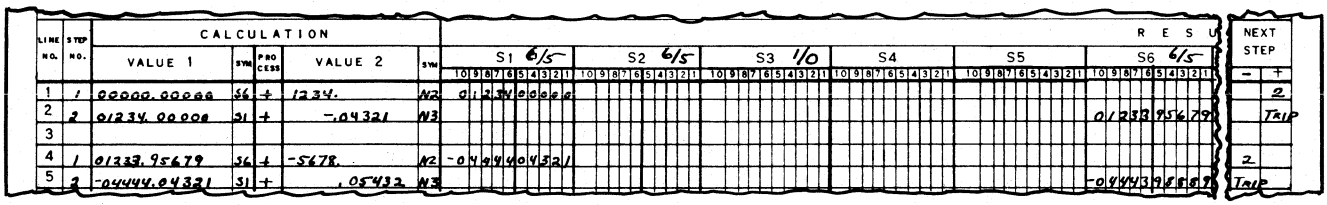
	Left value	Right value
Card #1	+1234	+4321
Card #2	+ +5678	+ -5432
	+6912	-1111



LINE NO.	STEP NO.	CALCULATION VALUE 1	SYM	PROCESS	VALUE 2	SYM	RESULT S1 6/5	S2 6/5	S3 1/0	S4	S5	S6 6/5	NEXT STEP −	NEXT STEP +
1	1	00000.00000	S6	+	1234.	N2	01234.00000							2
2	2	01234.00000	S1	+	.04321	N3						01234.04321		TRIP
3														
4	1	01234.04321	S6	+	5678.	N2	06912.04321							2
5	2	06912.04321	S1	+	-.05432	N3						06911.98889		TRIP

3. Left value is negative, right value is positive.

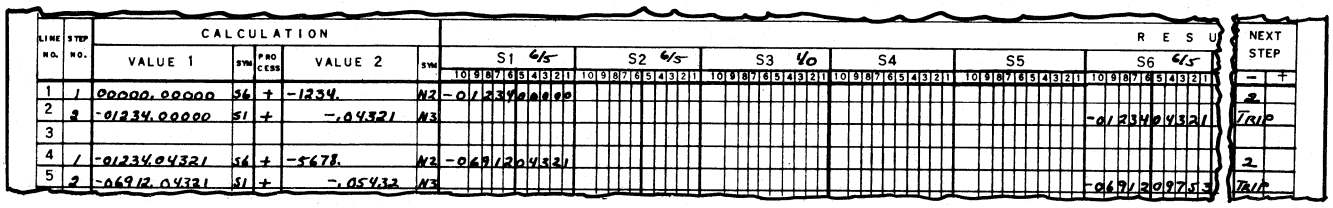
	Left value	Right value
Card #1	+1234	-4321
Card #2	+ -5678	+ +5432
	-4444	+1111



LINE NO.	STEP NO.	CALCULATION VALUE 1	SYM	PROCESS	VALUE 2	SYM	RESULT S1 6/5	S2 6/5	S3 1/0	S4	S5	S6 6/5	NEXT STEP −	NEXT STEP +
1	1	00000.00000	S6	+	1234.	N2	01234.00000							2
2	2	01234.00000	S1	+	-.04321	N3						01233.95679		TRIP
3														
4	1	01233.95679	S6	+	-5678.	N2	-04444.04321						2	
5	2	-04444.04321	S1	+	.05432	N3						-04443.98889	TRIP	

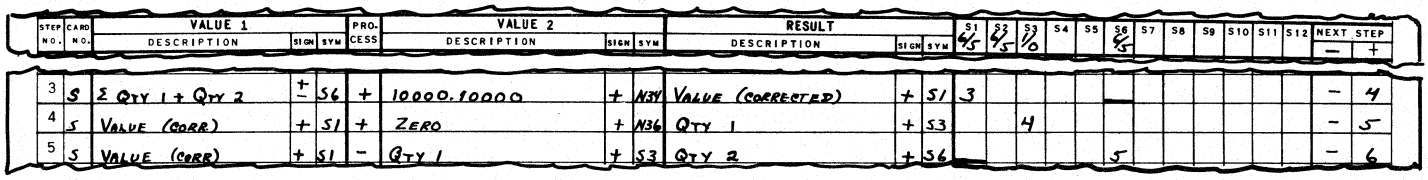
4. Both values are negative.

	Left value	Right value
Card #1	-1234	-4321
Card #2	+ -5678	+ -5432
	-6912	-9753



LINE NO.	STEP NO.	CALCULATION VALUE 1	SYM	PROCESS	CALCULATION VALUE 2	SYM	RESULT S1 6/5	S2 6/5	S3 10	S4	S5	S6 6/5	NEXT STEP −	NEXT STEP +
1	1	00000.00000	S6	+	-1234.	N2	-01234.00000							2
2	2	-01234.00000	S1	+	-.04321	N3						-01234.04321		TRIP
3														
4	1	-01234.04321	S6	+	-5678.	N2	-06912.04321							2
5	2	-06912.04321	S1	+	-.05432	N3						-06912.09753		TRIP

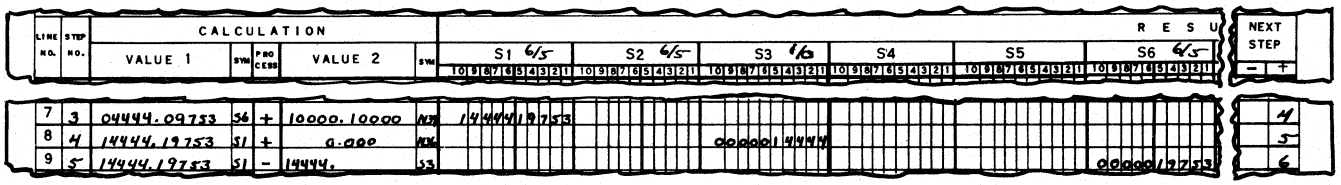
In the summary card, the first step is the addition of a constant value of 10000.10000 to the accumulating storage. This automatically makes the storage plus. It also places a one in the storage position to the left of any positive value, and a zero in the position to the left of any negative value. Note that all negative values will be in the form of a modified 9's complement with the final digit a 10's complement. During the following two steps the values are separated into different storages. The first of the two steps transfers the value to a 1/0 storage, which drops off the right value, leaving quantity 1. The second step subtracts quantity 1 from the full value, leaving quantity 2. These three program steps are:



STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 6/5	S2 6/5	S3 1/0	S4	S5	S6 6/5	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
3	S	Σ QTY 1 + QTY 2	±	S6	+	10000.10000	+	N34	VALUE (CORRECTED)	+	S1	3												−	4
4	S	VALUE (CORR)	+	S1	+	ZERO	+	N36	QTY 1	+	S3			4										−	5
5	S	VALUE (CORR)	+	S1	−	QTY 1	+	S3	QTY 2	+	S6						5							−	6

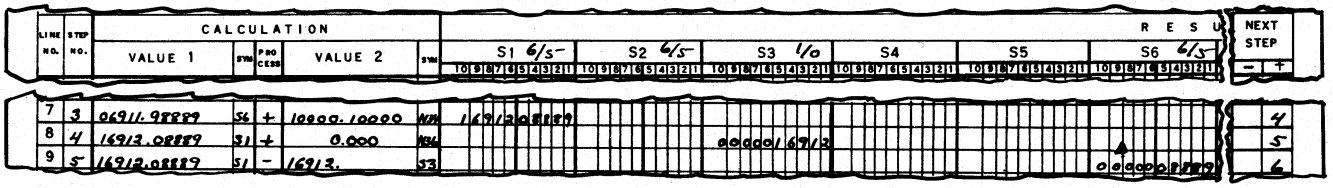
The planning charts are:

1. Both values are positive.



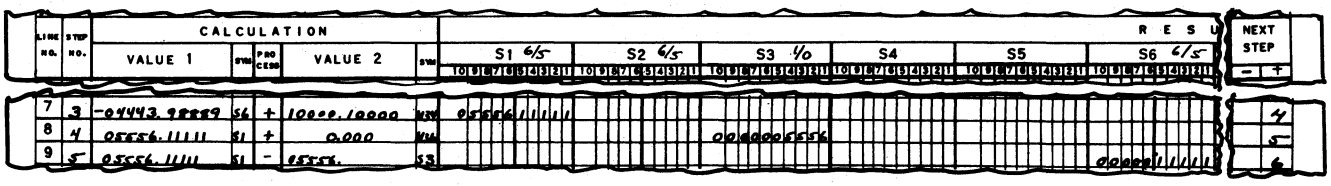
LINE NO.	STEP NO.	CALCULATION VALUE 1	SYM	PROCESS	CALCULATION VALUE 2	SYM	RESULT S1 6/5	S2 6/5	S3 1/0	S4	S5	S6 6/5	NEXT STEP −	NEXT STEP +
7	3	04444.09753	S6	+	10000.10000	N1	14444.19753							4
8	4	14444.19753	S1	+	0.000	N36			00000.14444					5
9	5	14444.19753	S1	−	14444.	S3						00000.19753		6

2. Left value is positive, right value is negative.

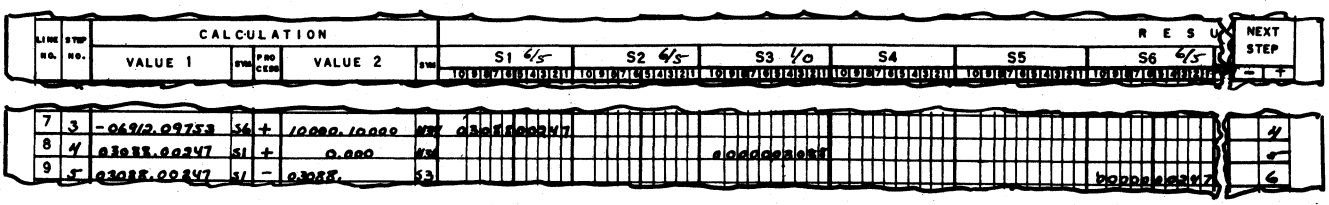


LINE NO.	STEP NO.	CALCULATION VALUE 1	SYM	PROCESS	CALCULATION VALUE 2	SYM	RESULT S1 6/5	S2 6/5	S3 1/0	S4	S5	S6 6/5	NEXT STEP −	NEXT STEP +
7	3	06911.98889	S6	+	10000.10000	N1	16912.08889							4
8	4	16912.08889	S1	+	0.000	N36			00000.16912					5
9	5	16912.08889	S1	−	16912.	S3						00000.08889		6

3. Left value is negative, right value is positive.



4. Both values are negative.



The next portion of the program varies dependent upon whether two storages are available for punching quantities 1 and 2 or whether they must be combined into one storage for punching. The program as illustrated below assumes that they must be combined. Note that through use of program selects this program could be repeated for several quantities, and it may therefore be necessary to punch two quantities from one storage. The variation in the program if they need not be combined is explained in page 175.

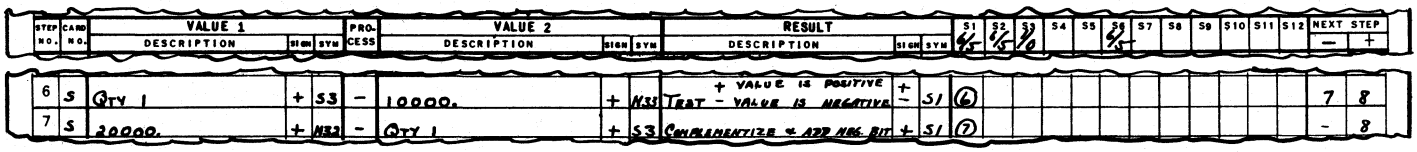
During the next step quantity 1 is tested to determine whether it is positive or negative. This is done by subtracting 10000. from the value. Since a positive value contains a 1 in the storage position to the left of the value, the sign of the result will be plus if the value is positive. This subtraction also removes the 1, leaving the true value in storage. The program therefore continues with the testing of quantity 2. For example:

$$\begin{array}{r} 14444 \\ -\ 10000 \\ \hline 4444 \end{array}$$

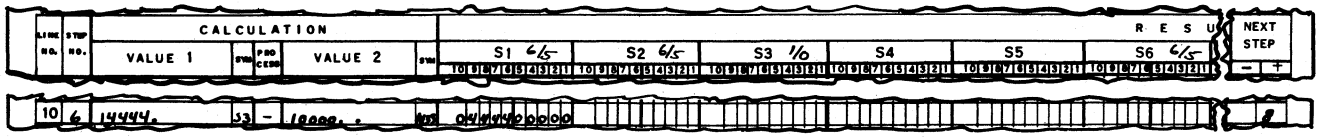
If quantity 1 is negative, there is a zero in the storage position to the left of the value, and the value itself is in the form of a modified 9's complement, with the last position a 10's complement. If the result of the subtraction of 10000. is negative, an additional step is required to change the value to its true value (complementize), and add a 1 in the position to the left of the value, which will be used to punch a negative control in the card. This result may be obtained by subtracting the value from 20000. For example:

$$\begin{array}{rr} 1) & \begin{array}{r} 05556 \\ -\ 10000 \\ \hline -04444 \end{array} \end{array} \qquad \begin{array}{rr} 2) & \begin{array}{r} 20000 \\ -\ 05556 \\ \hline 14444 \end{array} \end{array}$$

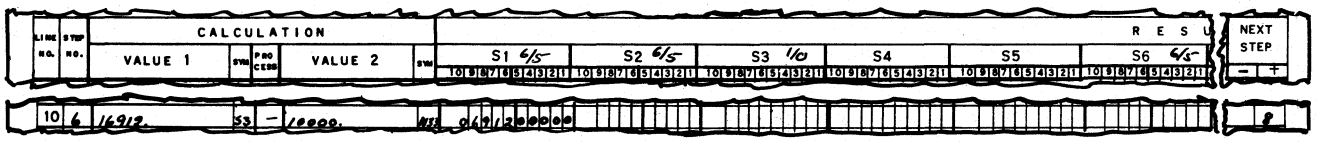
The two program steps required for dealing with quantity 1 are:



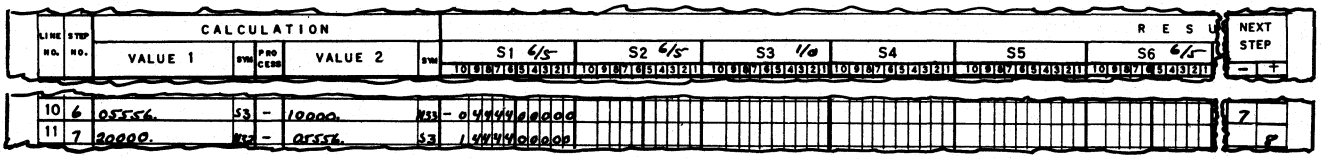
1. Both values are positive.



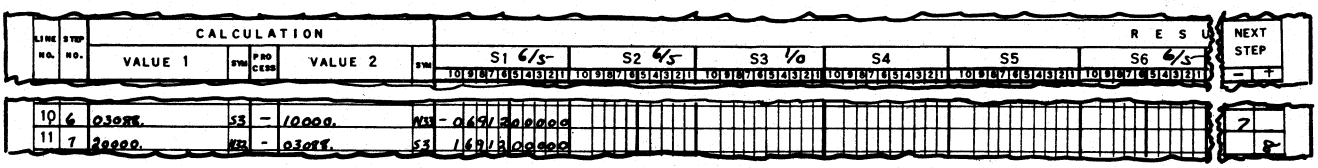
2. Left value is positive, right value is negative.



3. Left value is negative, right value is positive.



4. Both values are negative.



During the next step quantity 2 is tested to determine whether it is positive or negative. This is done by subtracting .10000 from the value, since the value is in the form .xxxxx. A positive value contains a 1 in the storage position immediately following the decimal location, so the sign of the result will be plus if the value is positive. This subtraction also removes the 1, leaving the true value in storage. The program continues with the final step, which combines quantity 1 and quantity 2.

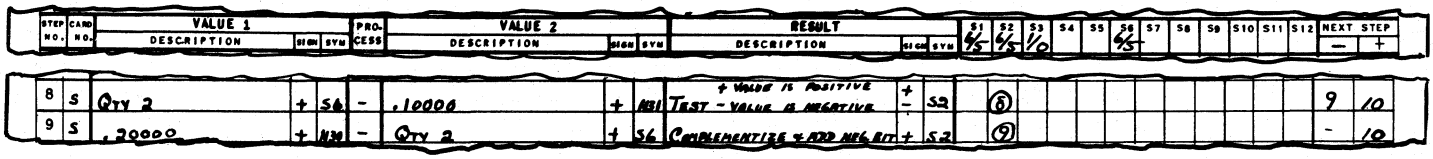
$$\begin{array}{r} .11111 \\ -\ .10000 \\ \hline .01111 \end{array}$$

If quantity 2 is negative, there is a zero in the storage position to the right of the decimal location, and the value itself is in the form of a

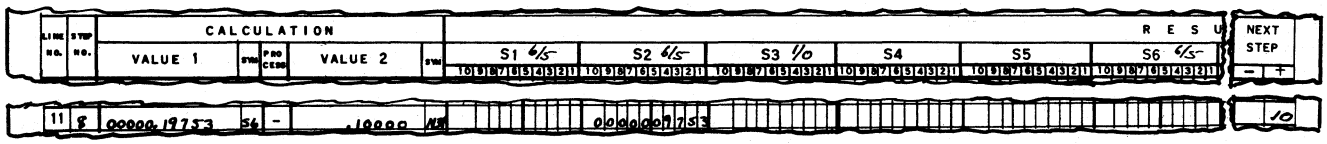
modified 9's complement, with the last position a 10's complement. If the result of the subtraction is negative, an additional step is required to change the number to its true value and add a 1 in the position to the left of the value. The 1 will be used to punch a negative control in the card. This result may be obtained by subtracting the value from .20000. For example:

1)	.08889	2)	.20000
	- .10000		-.08889
	- .01111		-.11111

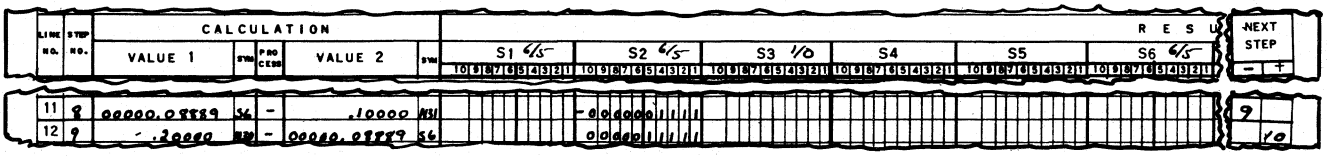
The two program steps required for dealing with quantity 2 are:



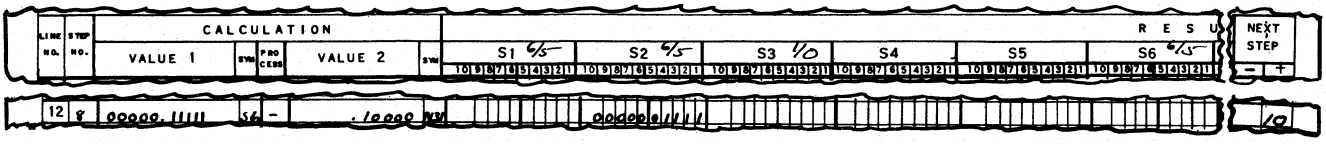
1. Both values are positive.



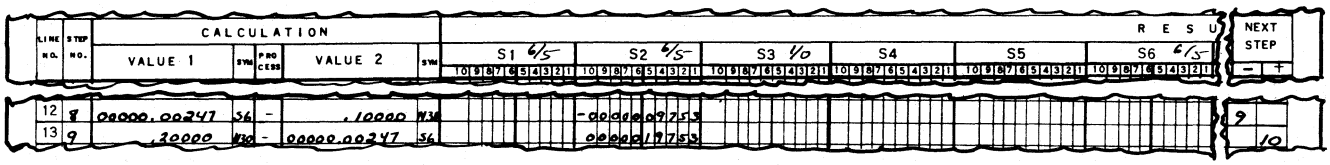
2. Left value is positive, right value is negative.



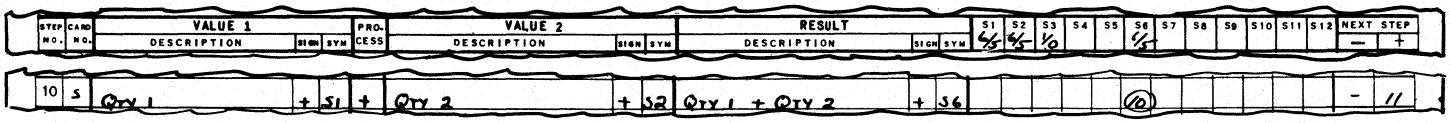
3. Left value is negative, right value is positive.

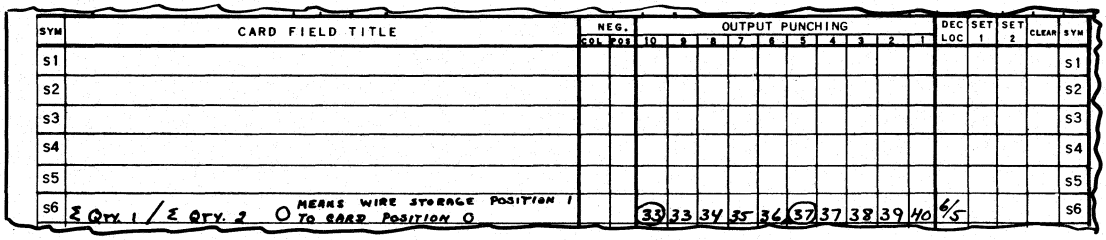


4. Both values are negative.

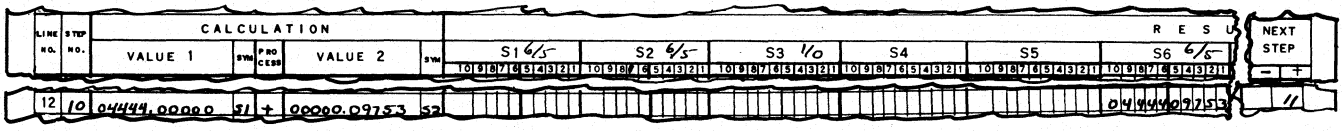


The final step of the program is the combination of quantity 1 and quantity 2 into the same storage. The 1 position to the left of each value is then wired to punch the negative control. With the two values combined in S6, the 1 position in storage column 10 is wired to punch a zero in column 33, while the 1 position in storage column 5 is wired to punch a zero in column 37. Storage columns 9-6 are wired normally to card columns 33-36, while storage columns 4-1 are wired normally to card columns 37-40.

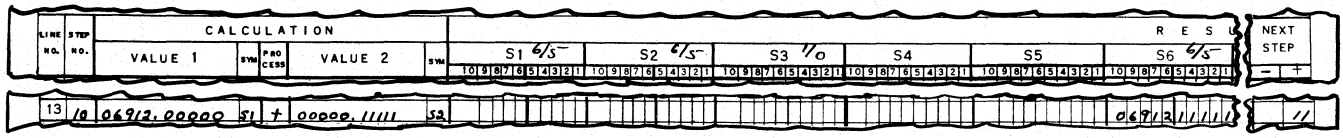




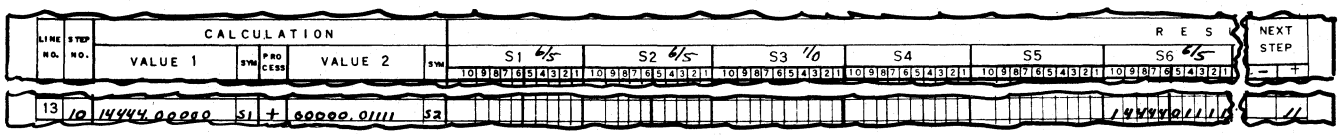
1. Both values are positive.



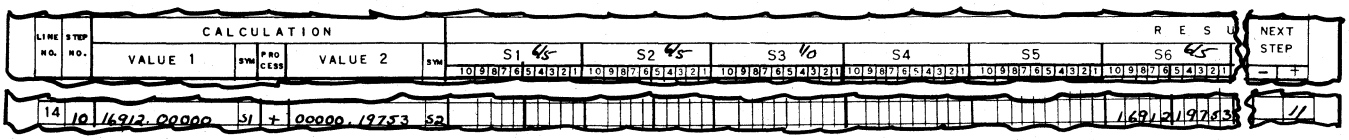
2. Left value is positive, right value is negative.



3. Left value is negative, right value is positive.



4. Both values are negative.



The entire program of 10 steps for the combining of two values, either positive or negative in one storage is:

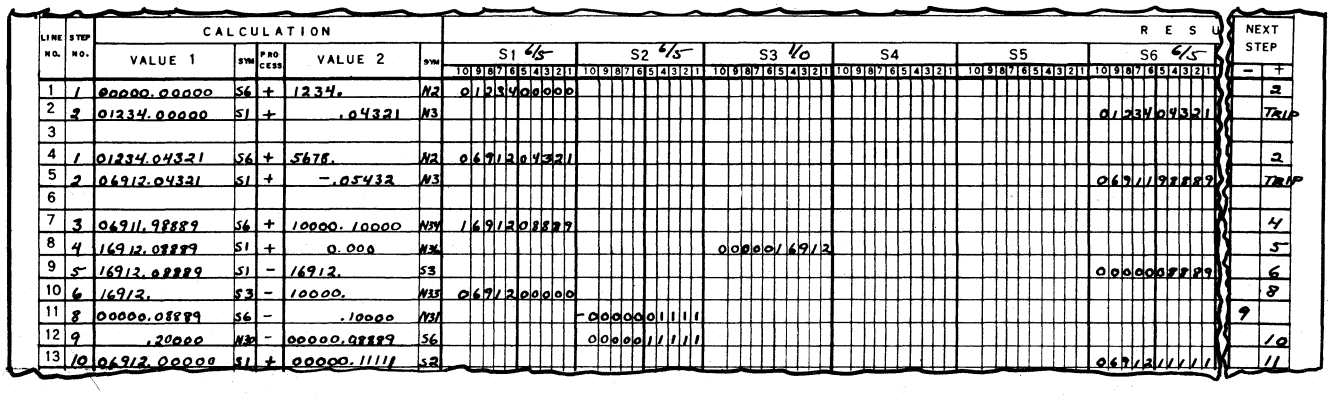
CARD DESCRIPTION D= DETAIL CARDS S= SUMMARY CARDS

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 6/5	S2 6/5	S3 1/0	S4	S5	S6 6/5	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
1	D	Σ QTY 1 + QTY 2	±	S6	+	QTY 1	±	N2	NEW Σ QTY 1 + QTY 2	±	S1	1												2	2
2	D	NEW Σ QTY 1 + QTY 2	±	S1	+	QTY 2	±	N3	NEW Σ QTY 1 + QTY 2	±	S6	—					(2)							TRIP	TRIP
3	S	Σ QTY 1 + QTY 2	±	S6	+	10000.10000	+	N34	VALUE (CORRECTED)	+	S1	3					—							-	4
4	S	VALUE (CORR.)	+	S1	+	ZERO	+	N36	QTY 1	+	S3			4										-	5
5	S	VALUE (CORR.)	+	S1	−	QTY 1	+	S3	QTY 2	+	S6						5							-	6
6	S	QTY 1	+	S3	−	10000	+	N33	TEST + VALUE IS POSITIVE − VALUE IS NEGATIVE	±	S1	(6)												7	8
7	S	20000	+	N32	−	QTY 1	+	S3	COMPLEMENTIZE & ADD NEG. BIT	+	S1	(7)		—										-	8
8	S	QTY 2	+	S6	−	.10000	+	N31	TEST + VALUE IS POSITIVE − VALUE IS NEGATIVE	±	S2		(8)											9	10
9	S	.20000	+	N30	−	QTY 2	+	S6	COMPLEMENTIZE & ADD NEG. BIT	+	S2		(9)				—							-	10
10	S	QTY 1	+	S1	+	QTY 2	+	S2	QTY 1 + QTY 2	+	S6						(10)							-	11

1. Both values are positive.

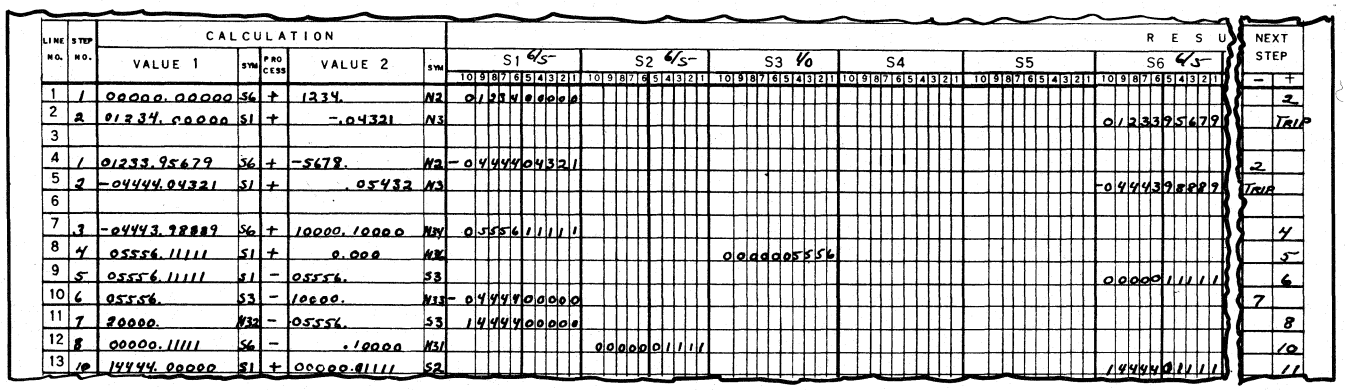
LINE NO.	STEP NO.	VALUE 1	SYM	PROCESS	VALUE 2	SYM	S1 6/5	S2 6/5	S3 1/0	S4	S5	S6 6/5	NEXT STEP −	NEXT STEP +
1	1	00000.00000	S6	+	-1234.	N2	-01234.00000						2	
2	2	-01234.00000	S1	+	.04321	N3						-01233.95679	TRIP	
3														
4	1	-01233.95679	S6	+	5678.	N2	04444.04321							2
5	2	04444.04321	S1	+	.05432	N3						04444.09753		TRIP
6														
7	3	04444.09753	S6	+	10000.10000	N34	14444.19753							4
8	4	14444.19753	S1	+	0.000	N36			00000.14444					5
9	5	14444.19753	S1	−	14444.	S3						00000.19753		6
10	6	14444.	S3	−	10000.	N33	04444.00000							8
11	8	00000.19753	S6	−	.10000	N31		00000.09753						10
12	10	04444.00000	S1	+	00000.09753	S2						04444.09753		11

2. Left value is positive, right value is negative.



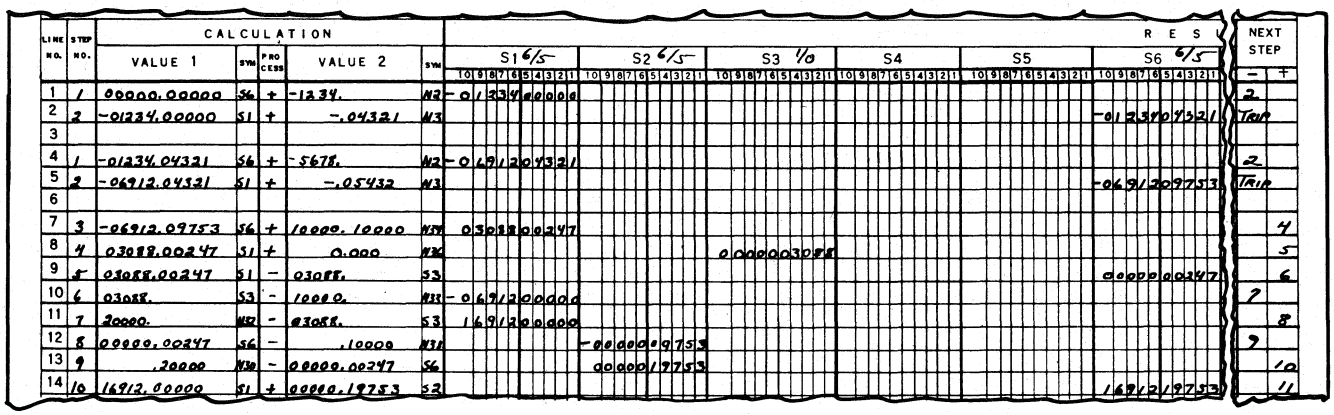
LINE NO.	STEP NO.	CALCULATION VALUE 1	SYM	PROCESS	VALUE 2	SYM	RESULT S1 6/5-	S2 6/5-	S3 1/0	S4	S5	S6 6/5-	NEXT STEP -	NEXT STEP +
1	1	00000.00000	S6	+	1234.	N2	0123400000							2
2	2	01234.00000	S1	+	.04321	N3						0123404321		TRIP
3														
4	1	01234.04321	S6	+	5678.	N2	0691204321							2
5	2	06912.04321	S1	+	-.05432	N3						0691198889		TRIP
6														
7	3	06911.98889	S6	+	10000.10000	N3Y	1691208887							4
8	4	16912.08889	S1	+	0.000	N3X			0000016912					5
9	5	16912.08889	S1	-	16912.	S3						0000008889		6
10	6	16912.	S3	-	10000.	N3S	0691200000							8
11	8	00000.08889	S6	-	.10000	N3I		-0000001111					9	
12	9	.20000	N3Z	-	00000.98889	S6		0000011111						10
13	10	06912.00000	S1	+	00000.11111	S2						0691211111		11

3. Left value is negative, right value is positive.



LINE NO.	STEP NO.	CALCULATION VALUE 1	SYM	PROCESS	VALUE 2	SYM	RESULT S1 6/5-	S2 6/5-	S3 1/0	S4	S5	S6 6/5-	NEXT STEP -	NEXT STEP +
1	1	00000.00000	S6	+	1334.	N2	0133400000							2
2	2	01234.00000	S1	+	-.04321	N3						0123395679		TRIP
3														
4	1	01233.95679	S6	+	-5678.	N2	-0444404321						2	
5	2	-04444.04321	S1	+	.05432	N3						-0444398889	TRIP	
6														
7	3	-04443.98889	S6	+	10000.10000	N3Y	0555611111							4
8	4	05556.11111	S1	+	0.000	N3X			0000005556					5
9	5	05556.11111	S1	-	05556.	S3						0000011111		6
10	6	05556.	S3	-	10000.	N3S	-0444400000						7	
11	7	20000.	N3Z	-	05556.	S3	1444400000							8
12	8	00000.11111	S6	-	.10000	N3I		0000011111						10
13	10	14444.00000	S1	+	00000.81111	S2						1444481111		11

4. Both values are negative.



LINE NO.	STEP NO.	CALCULATION VALUE 1	SYM	PROCESS	VALUE 2	SYM	RESULT S1 6/5-	S2 6/5-	S3 1/0	S4	S5	S6 6/5-	NEXT STEP -	NEXT STEP +
1	1	00000.00000	S6	+	-1234.	N2	-0133400000						2	
2	2	-01234.00000	S1	+	-.04321	N3						-0123407521	TRIP	
3														
4	1	-01234.04321	S6	+	-5678.	N2	-0691204321						2	
5	2	-06912.04321	S1	+	-.05432	N3						-0691209753	TRIP	
6														
7	3	-06912.09753	S6	+	10000.10000	N3Y	0308800247							4
8	4	03088.00247	S1	+	0.000	N3X			0000003088					5
9	5	03088.00247	S1	-	03088.	S3						0000000247		6
10	6	03088.	S3	-	10000.	N3S	-0691200000						7	
11	7	20000.	N3Z	-	03088.	S3	1691200000							8
12	8	00000.00247	S6	-	.10000	N3I		-0000009753					9	
13	9	.20000	N3Z	-	00000.00247	S6		0000019753						10
14	10	16912.00000	S1	+	00000.19753	S2						1691219753		11

If the two quantities need not be combined in one storage, steps 7, 9 and 10 are unnecessary. Quantity 1 may be wired normally from storage S1, with the negative sign wired to punch a zero in column 33. Quantity 2 may be wired normally from storage S2, with the negative sign wired to punch a zero in column 37.

20. Square root

The computation of square root on the 60 or 120 is an excellent example of a looping or iterative routine. An approximation is made of the root, which is then tested to determine whether it actually is the root. If not, a new approximation is made and tested. The same series of steps is used to do this until the root is found.

The testing of the approximation is made by dividing it into the number of which the root is being computed. When the quotient of the division equals the approximation, the root has been found:

$64 \div 8 = 8$ $\quad N \div A = Q$

N = number of which to compute root
A = approximation
Q = quotient

The root may be carried to as many places as desired, within the 10 digit limit of storage, by the setting of the decimal locations of the storages in which the results of the division are placed. If the root is to be carried to 4 places, the storages would be 5/4.

The first approximation which is made must always be larger than the true root. This may be accomplished by using the number plus one divided by two:

$$\frac{(N+1)}{2} = \frac{(64+1)}{2} = 32.5$$

This value is then divided into the number.

$$64.0 \div 32.5 = 1.9$$

To determine whether this quotient is the same as the approximation, the approximation is subtracted from it.

$$1.9 - 32.5 = -30.6$$

As long as the result is minus, the approximation is larger than the quotient. When the two values are equal, a plus zero will be the result.

Since the result in the above example is minus, a new approximation must be made. One which is closer to the true root may be obtained by averaging the two values:

$$\frac{1.9 + 32.5}{2} = 17.2$$

This process is repeated until the true root is found.

The program for handling the above routine on the computer follows. The elements used are:

N1 - Number of which to take the root
N33- 2
N34- 1

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 2/1	S2 2/1	S3 2/1	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
1		N	+	N1	+	1	+	N34	N+1	+	S1	1												−	2
2		N+1 OR Q+A	+	S1	÷	2	+	N33	APPROXIMATION	+	S2		2											−	3
3		N	+	N1	÷	APPROXIMATION	+	S2	QUOTIENT	+	S3			3										−	4
4		QUOTIENT	+	S3	−	APPROXIMATION	+	S2	TEST + ROOT − NOT ROOT	+	S1	4												5−	6
5		QUOTIENT	+	S3	+	APPROXIMATION	+	S2	Q + A	+	S1	5												−	2

Step 1: Used only for the first approximation.

Step 2: For all loops except the first, the sum of the quotient plus the approximation is averaged. For the first loop, this step determines half of the number plus 1. In either case, the result is the new approximation.

Step 3: The number of which the root is being taken is divided by the approximation.

Step 4: The approximation is subtracted from the quotient of step 3. If the result is minus, the two are not equal and another approximation must be made. If the two are equal, the square root has been found.

Step 5: As the first step towards the new approximation, the approximation and quotient are added together. They will be averaged in the next step, which is step 2.

Using 64 as the number of which the root is to be taken, the calculation would be as follows:

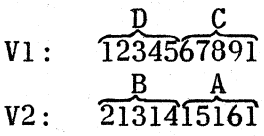
LINE NO.	STEP NO.	CALCULATION VALUE 1	SYM	PROCESS	VALUE 2	SYM	S1 2/1	S2 2/1	S3 2/1	NEXT STEP −	NEXT STEP +
1	1	64	N1	+	1	N34	65			−	2
2	2	65	S1	÷	2	N33		325		−	3
3	3	64	N1	÷	32.5	S2			19	−	4
4	4	1.9	S3	−	32.5	S2	306			5−	−
5	5	1.9	S3	+	32.5	S2	344			−	2
6	2	34.4	S1	÷	2	N33		172		−	3
7	3	64	N1	÷	17.2	S2			37	−	4
8	4	3.7	S3	−	17.2	S2	135			5−	−
9	5	3.7	S3	+	17.2	S2	209			−	2
10	2	20.9	S1	÷	2	N33		104		−	3
11	3	64	N1	÷	10.4	S2			61	−	4
12	4	6.1	S3	−	10.4	S2	43			5−	−
13	5	6.1	S3	+	10.4	S2	165			−	2
14	2	16.5	S1	÷	2	N33		82		−	3
15	3	64	N1	÷	8.2	S2			78	−	4
16	4	7.8	S3	−	8.2	S2	4			5−	−
17	5	7.8	S3	+	8.2	S2	16			−	2
18	2	16	S1	÷	2	N33		8		−	3
19	3	64	N1	÷	8	S2			8	−	4
20	4	8	S3	−	8	S2	000			−	6

21. Obtaining a product of more than ten digits

Occasionally it may be necessary on the computer to obtain a product of more than 10 digits. This is obviously impossible in one step, since the maximum result which can be placed in one storage is 10 digits. Although the first 10 digits of a product may be obtained in one step, the remaining digits cannot be obtained by doing the multiplication a second time and shifting the first 10 digits to the left. This would place a significant digit in the 11th column of the A section of the accumulator. It is therefore necessary to do a split multiplication and add the products, as outlined below.

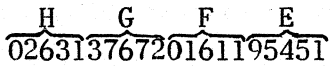
The example given below is the multiplication of two 10-digit numbers to obtain a 20-digit product. If the two values are smaller in size, it may in certain cases be possible to reduce the number of steps and storages required.

When doing the split multiplication, it is essential that none of the individual multiplications have a product of more than 10 digits. This can be accomplished by making both V1 and V2 five digits in each multiplication. With a 10-digit multiplier and a 10-digit multiplicand, four multiplications will be required. For example:



Multiplication 1:	A × C	1029295451
Multiplication 2:	B × C	1447028774
Multiplication 3:	A × D	0187162545
Multiplication 4:	B × D	0263121330

The actual 20-digit product of the multiplication is 02631376720161195451. It may be obtained from the four products shown above in 4 groups of 5 digits:



To do this it is necessary to think of each of the four products as being divided into two groups of 5 digits. These groups are then added together as follows to obtain the four final product groups:

Group E: Last 5 digits of multiplication 1

Group F: First 5 digits of multiplication 1, plus
Last 5 digits of multiplication 2, plus
Last 5 digits of multiplication 3

Group G: Carry from group F, plus
First 5 digits of multiplication 2, plus
First 5 digits of multiplication 3, plus
Last 5 digits of multiplication 4

Group H: Carry from group G, plus
First 5 digits of multiplication 4

1029295451	E
1447028774	F
0187162545	G
0263121330	H
02631376720161195451	Product

The following example assumes that one of the two 10-digit amounts is punched in card columns 21-30, and the other is punched in card columns 31-40. The 20-digit product is to be punched in columns 46-65.

Each of the two amounts is read into the computer as two 5-digit elements. Because of the decimal locations necessary for the addition of the products of the multiplications, these four elements are assigned decimal locations as follows:

Amount 1: First 5 digits - 5 whole numbers

Amount 1: Last 5 digits - 5 decimal positions

Amount 2: First 5 digits - 5 whole numbers, changed by a selector to 5 decimal positions

Amount 2: Last 5 digits - 5 decimal positions

These elements may be entered in the element sections of the program charts as follows:

TABLE OF FACTORS

SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE
N1	+		AMOUNT 1 - FIRST 5 DIGITS	N13				N25			
N2	+		AMOUNT 1 - LAST 5 DIGITS	N14				N26			
N3	+		AMOUNT 2 - FIRST 5 DIGITS	N15				N27			
N4	+		AMOUNT 2 - LAST 5 DIGITS	N16				N28			
N5				N17				N29			
N6				N18				N30			
N7				N19				N31			
N8				N20				N32			
N9				N21				N33			
N10				N22				N34			
N11				N23				N35			
N12				N24				N36			0.000

ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD-F1, F2, ETC: CONTROL - C1, C2, ETC.) DECIMAL LOCATION, NEGATIVE CONTROL AND TRANSFER LINE. INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N1	F1				6/5	21	22	23	24	25	▲				
N2	F2				6/5						▲	26	27	28	29	30
N3	F3				COM T1-1	31	32	33	34	35					
N4	F4				6/5						▲	36	37	38	39	40
N5																

ELEM. DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															
N22															
N23															

ACCUMULATOR INPUTS

TOP LINE - TRANSFER LINE (F1, F2, ETC.)
INDICATE:
BOTTOM LINE - CARD COLUMNS

SYM.	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
A1	F1	F1	F1	F1	F1	F2	F2	F2	F2	F2
	21	22	23	24	25	26	27	28	29	30
A2	F3	F3	F3	F3	F3	F4	F4	F4	F4	F4
	31	32	33	34	35	36	37	38	39	40
A3										

The program is as follows:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 11/10	S2 6/5	S3 6/5	S4 1/0	S5 6/5	S6 COM T1-2	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
1		AMOUNT 1 - LAST 5 DIG.	+	N2	X	AMOUNT 2 - LAST 5 DIG.	+	N4	PRODUCT 1 $\{1_1$ = FIRST 5 DIG. 1_2 = LAST 5 DIG.	+	S1	①												−	2
2		AMOUNT 1 - LAST 5 DIG.	+	N2	X	AMOUNT 2 - FIRST 5 DIG.	+	N3	PRODUCT 2 $\{2_1$ = FIRST 5 DIG. 2_2 = LAST 5 DIG.	+	S2		2											−	3
3		PRODUCT 1	+	S1	+	PRODUCT 2	+	S2	PRODUCT $2_1 + \Sigma 1_1 + 2_2$	+	S3			3										−	4
4		AMOUNT 1 - FIRST 5 DIG.	+	N1	X	AMOUNT 2 - LAST 5 DIG.	+	N4	PRODUCT 3 $\{3_1$ = FIRST 5 DIG. 3_2 = LAST 5 DIG.	+	S2		4											−	5
5		PRODUCT 3	+	S2	+	ZERO	+	N3G	PRODUCT 3_1	+	S4				5									−	6
6		PRODUCT 3	+	S2	−	PRODUCT 3_1	+	S4	PRODUCT 3_2	+	S5					6								−	7
7		PRODUCT $2_1 + \Sigma 1_1 + 2_2$	+	S3	+	PRODUCT 3_2	+	S5	PRODUCT $2_1 + \Sigma 1_1 + 2_2 + 3_2$	+	S2		⑦											−	8
8		PRODUCT $2_1 + \Sigma 1_1 + 2_2 + 3_2$	+	S2	+	PRODUCT 3_1	+	S4	Σ PRODUCTS $2_1 + 3_1$	+	S6						8							−	PS1
9		AMOUNT 1 - FIRST 5 DIG.	+	N1	X	AMOUNT 2 - FIRST 5 DIG.	+	N3	PRODUCT 4 $\{4_1$ = FIRST 5 DIG. 4_2 = LAST 5 DIG.	+	S3			9										−	10
10		PRODUCT 4	+	S3	+	Σ PRODUCTS $2_1 + 3_1$	+	S6	PRODUCT $4_1 + \Sigma 2_1 + 3_1 + 4_2$	+	S5					⑩								−	11

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1		PS1	11/10	DEC. LOC. - N3	6/5
1-2			6/5	DEC. LOC. - S6	1/0
1-3					
1-4					

SYM	CARD FIELD TITLE	NEG. COL	NEG. POS	OUTPUT PUNCHING 10	9	8	7	6	5	4	3	2	1	DEC LOC	SET 1	SET 2	CLEAR	SYM
S1	PRODUCT DIGITS 16-20								61	62	63	64	65	11/10				S1
S2	PRODUCT DIGITS 11-15								56	57	58	59	60	6/5				S2
S3														6/5				S3
S4														1/0				S4
S5	PRODUCT DIGITS 1-10			46	47	48	49	50	51	52	53	54	55	6/5				S5
S6														COM T1-2				S6
S7																		S7
S8																		S8
S9																		S9
S10																		S10
S11																		S11
S12																		S12

S1-1369 RRBH 12456

FROM	TO
START	ST. 1
SET 1	
SET 2	
SORT 1	
SORT 2	
CLEAR	
P.S. I	ST. 9
P.S. II	
P.S. III	
P.S. IV	

REPRODUCE CONTROLS	COL.	POS.
REP		
SKIP		
SET HOLD		
SEC REP		

	0 ÷ 0	N ÷ 0
STOP		
SORT		

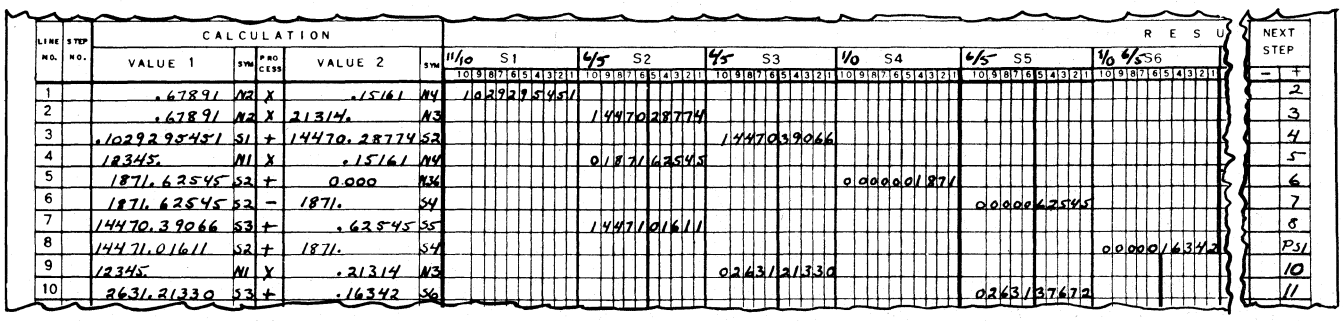
RESTART	
STEP	
PROGRAM	

The planning chart and values which will be used in it are:

Amount 1: N1 N2 → 1234567891

Amount 2: N3 N4 → 2131415161

Result: S5 S2 S1 → 026313767201611954 51



LINE NO.	STEP NO.	CALCULATION VALUE 1	SYM	PROCESS	VALUE 2	SYM	RESULT 11/10 S1	4/5 S2	4/5 S3	1/0 S4	4/5 S5	1/0 4/5 S6	NEXT STEP −	NEXT STEP +
1		.67891	N2	X	.15161	N4	1029295451							2
2		.67891	N2	X	21314.	N3		1447028774						3
3		.1029295451	S1	+	14470. 28774	S2			1447039066					4
4		18345.	N1	X	.15161	N4		0187162545						5
5		1871. 62545	S2	+	0.000	N6				0000001871				6
6		1871. 62545	S2	−	1871.	S4					0000062545			7
7		14470. 39066	S3	+	.62545	S5		1447101611						8
8		14471. 01611	S2	+	1871.	S4						0000016342		PS1
9		12345.	N1	X	.21314	N3			0263121330					10
10		2631. 21330	S3	+	.16342	S6					0263137672			11

The step by step explanation of the program is:

Step 1: The last 5 digits of amount 1 are multiplied by the last 5 digits of amount 2. Since both of these elements have 5 decimal positions, the 10-digit product is placed in a storage with an 11/10 decimal location. The last 5 digits of this product are also the last 5 digits of the final 20-digit product, and are wired for punching in columns 61-65. The first 5 digits of this product are to be added to products which will be computed in later steps.

Step 2: The last 5 digits of amount 1, consisting of 5 decimal positions, are multiplied by the first 5 digits of amount 2, consisting of 5 whole numbers. The 10-digit product is therefore placed in a 6/5 storage.

Step 3: Product 1 is added to product 2. Since product 1 has an 11/10 decimal location while product 2 has a 6/5 decimal location, the first 5 digits of product 1 are added to the last 5 digits of product 2. Any carry that occurs will be added into the sixth digit of product 2. At the end of this step the original product 2 is no longer needed.

Step 4: The first 5 digits of amount 1, consisting of 5 whole numbers, are multiplied by the last 5 digits of amount 2, consisting of 5 decimal positions. The 10-digit product is therefore placed in a 6/5 storage.

Step 5: Product 3 is transferred to a 1/0 storage to isolate the first 5 digits. Note that product 3 cannot be added directly to the value computed in step 3 since both of these storages contain 10-digit numbers. It is therefore possible to develop a carry into the 11th column of the A section of the accumulator. By separating product 3 into the two 5-digit numbers, this problem will be avoided.

Step 6: The first 5 digits of product 3 are subtracted from the entire value of product 3, isolating the last 5 digits. This value has a 6/5 decimal location. The original product 3 is no longer needed.

Step 7: The last 5 digits of product 3, isolated in step 6, are added to the value computed in step 3. The last 5 digits of this value are the sum of the first 5 digits of product 1 plus the last 5 digits of product 2. Any carry that develops in step 7 will be added to the first 5 digits of product 2, which are in positions 6-10 of the result storage. Positions 1-5 of the result storage contain the sixth through tenth digits of the final 20-digit product, and are wired to punch in columns 56-60.

Step 8: The value computed in step 7 is added to the first 5 digits of product 3. The purpose of the addition is to add the first 5 digits of product 2, including any carry from the sixth through tenth digits of the final 20-digit product, to the first 5 digits of product 3. The result is placed in a 1/0 storage, which provides for any carry which may occur as a result of this addition.

PS 1: PS 1 picks up selector T1. In pole T1-1 the decimal location of element N3 is changed so that it consists of 5 decimal positions instead of 5 whole numbers. The purpose of this change is to obtain a 6/5 decimal location for the product of the final multiplication. The decimal location of S6 is changed so that the decimal location of the sum of the first 5 digits of products 2 and 3 will be changed to 6/5 and align with the last 5 digits of multiplication 4.

Step 9: The first 5 digits of amount 1, consisting of 5 whole numbers, are multiplied by the first 5 digits of amount 2, consisting of 5 decimal positions. The result of this multiplication is placed in a 6/5 storage.

Step 10: Product 4 is added to the sum of the first 5 digits of the products of multiplications 2 and 3. Because of the decimal location, this sum is added to the last 5 digits of product 4. Any carry that occurs will be added to the first 5 digits of product 4. This storage now contains product digits 11-20 of the final product, and is wired to punch in columns 46-55.

22. Fractional twelfths

If fractional twelfths are punched in a card field, it becomes necessary to change the twelfths to units in order to use the field in a computer program. It may also be necessary to convert units back to twelfths for punching into a summary card. The following examples show a method for handling each of these problems.

A. CONVERTING DOZENS TO UNITS

Assume that quantity is punched in columns 27-31, with columns 27-30 punched as whole dozens and column 31 the fraction of a dozen. Column 31 is punched with standard twelfth punching - that is, 1/12 to 9/12 are punched as 1 to 9; 10/12 is punched as 0 and 5; 11/12 is punched as 3 and 7. The quantity is negative if a zero is punched in column 27.

The most simple method of converting quantity from dozens to units is to read it as two elements. In the following example, the whole dozens, columns 27-30, will be element N2, while the fraction of a dozen will be element N3. Element N2 is sensed with normal input wiring. Due to the coding of fractional dozens, however, column 31 is used to pick up selectors through which constant values are created to represent the fraction of a dozen.

There is only one credit position for the two elements - a zero in column 27. Since one credit position cannot be used to make two elements negative without a selector, the zero in column 27 is wired to pick up selector T9. Through the select of T9-1 and T9-2, N2 and N3 are wired to become minus.

The element chart, selector chart, and program for converting dozens to units follow.

ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD-F1, F2, ETC: CONTROL - C1, C2, ETC.) DECIMAL LOCATION, NEGATIVE CONTROL AND TRANSFER LINE. INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N1															
N2	F2	27	0	Pv.	1/0							27	28	29	30
N3	COM T22			T9	1/0									X	X
N4															

ELEM. DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															
N22															

ACCUMULATOR INPUTS

TOP LINE - TRANSFER LINE (F1, F2, ETC.)
INDICATE:
BOTTOM LINE - CARD COLUMNS

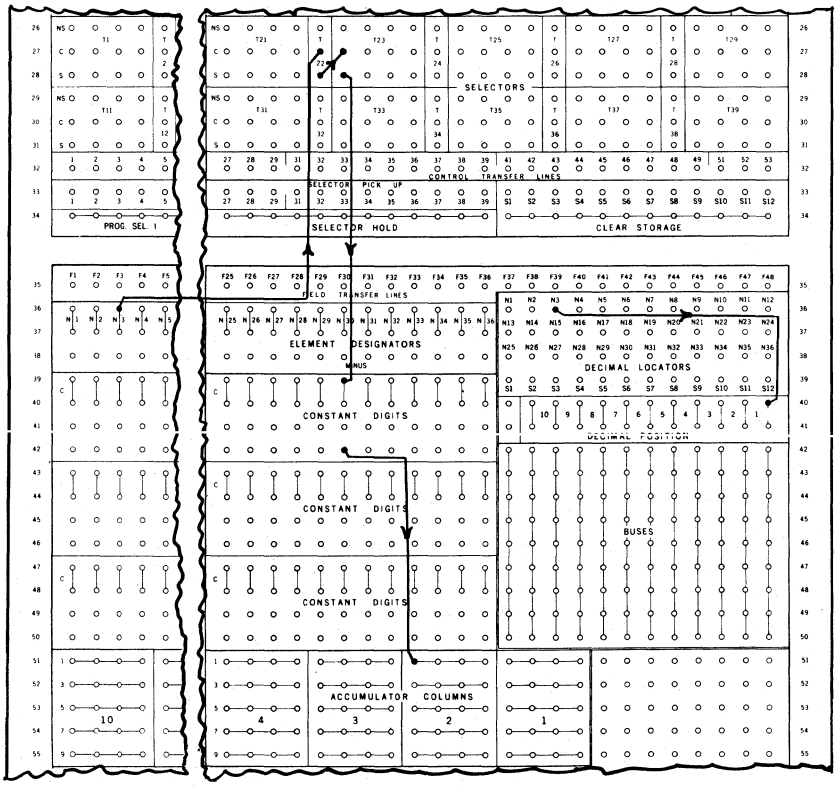
SYM.	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
A1										
A2							F2	F2	F2	F2
							27	28	29	30
A3										

	SELECTOR	TFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
9.1	C9		0/27	MINUS	ELEM. DES. N2	
9.2				MINUS	ELEM. DES. N3	
9.3						
9.4						
21.1						
21.2						
21.3						
21.4						
22	C22		0/31	COM. T 23-1	ELEM. DES. N3	COM T24
23.1	C23		5/31	CONSTANT 10	SEL. T 22	- (ZERO)
23.2				COM. T27-3	NS T 26	COM. T25-2
23.3						
23.4						
24	C24		1/31	COM. T27-1	NS T 22	COM. T26
25.1	C25		7/31	CONSTANT 11	SEL. T26	COM. T27-2
25.2				COM T27-4	NS. T23-2	COM. T28
25.3						
25.4						
26	C26		3/31	COM. T25-1	NS T24	COM. T23-2
27.1	C27		9/31	CONSTANT 2	SEL. T24	CONSTANT 1
27.2				CONSTANT 4	NS. T25-1	CONSTANT 3
27.3				CONSTANT 6	SEL T23-2	CONSTANT 5
27.4				CONSTANT 8	SEL. T25-2	CONSTANT 7
28	C27		9/31	CONSTANT 9	N.S. T25-2	—
29.1						
29.2						
29.3						
29.4						

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP —	NEXT STEP +
												1/0	1/0												
6		WHOLE DOZ.	+ −	N2	X	12	+	N4	DOZ. CONVERTED	+ −	S1	6												7	7
7		DOZ. CONVERTED	+ −	S1	+	FRACTIONAL DOZ.	+ −	N3	TOTAL UNITS	+ −	S2		7											8	8

Step 6: Whole dozens are multiplied by 12 to convert them to units.

Step 7: The units are added to a constant value which represents the fractional part of a dozen. The constant value is obtained by wiring the element designator power of N3 to the common of T22. From the appropriate selector the power is routed to a constant digit, and from there to the accumulator column. The following example shows the method of wiring the constant value of 10 from the select of T23-1.



Total units may be used in the program for any necessary computation without the difficulties which arise when twelfths are used. Since twelfths carry at 12 rather than 10 as the computer does, addition and subtraction become almost impossible without this conversion.

The use of decimal equivalents is unsatisfactory in accumulating, since several of the fractions do not come out evenly and therefore the total may not be a true total when reconverted to twelfths.

It may be necessary to multiply the number of dozen times a price per dozen as well as accumulate the quantity. In this case the price should also be converted to a unit basis. This may be done by dividing the price by 12, and carrying the result to three or four places following the decimal.

For example, assume that the price per dozen is $2.00, and 1 1/2 dozen have been ordered. The quantity of 1 1/2 dozen is punched as 1.6. When converted to units, this becomes 18. The result of dividing $2.00 by 12 is 1.666. The amount would be computed as follows:

$$\begin{array}{r} 1.666 \\ \times\ 18 \\ \hline 13328 \\ 1666 \\ \hline 29988 \end{array} \qquad \begin{array}{r} 2.9988 \\ +\ .005 \\ \hline 3.0038 \end{array} \quad \text{or} \quad 3.00$$

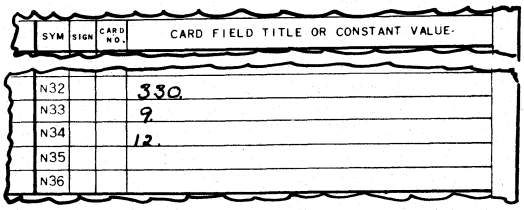
Note that if accumulating is unnecessary, it would be possible to use the decimal equivalent of the twelfth fraction instead of converting the quantity to units. To do this, the quantity would be wired as one element instead of two as when changing it to units. Using the preceding example, element designator N2 would be wired both to columns 27-30 and the common of T22. The constant values would be the decimal equivalents instead of those shown on the selector chart. For example, instead of a constant value of 1, the constant would be .0833. Card columns 27-30 would be wired to accumulator columns 8-5, while the constant digits would be placed in accumulator columns 4-1. Element N2 would be assigned a 5/4 decimal location.

B. CONVERTING UNITS TO DOZENS

It is frequently necessary to convert units back to dozens for punching in a summary card. If the problem involves accumulation from several cards, the accumulating total should be kept in the form of units until the summary card, when it can be reconverted to dozens.

Three steps are required to change total units to dozens and fraction of a dozen. If the fraction of a dozen is to be punched in the standard twelfth code, two additional steps are required for this.

The constant values and program are:



STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PRO-CESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 1/0	S2 1/0	S3 1/0	S4 1/0	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP —	NEXT STEP +
15		TOTAL IN UNITS	±	S3	÷	12	+	N34	WHOLE DOZ.	±	S2		(15)											16	16
16		WHOLE DOZ.	±	S2	X	12	+	N34	WHOLE DOZ. IN UNITS	±	S1	16												17	17
17		TOTAL IN UNITS	±	S3	−	WHOLE DOZ. IN UNITS	±	S1	FRACT. DOZ.	±	S4				(17)									PS1	18
18		9	+	N33	COM T8	FRACT. DOZ.	±	S4	TEST + 9 OR LESS − 10 OR 11	±	S3			18										19	SET 1
19		REMAINDER {-1 IF 10/12, -2 IF 11/12	−	S3	X	330	+	N33	CODE FOR PCH 10/12, 11/12	−	S4				(19)									SET 1	−

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
7-4					
8		PS1	+	PR. ST. 18	−
9-1		PS1	C12 (0/27)	C11 (NEG. SIGN-S4)	

Step 15: Total units are divided by 12 and placed in a storage with a 1/0 decimal location which drops off any remainder. The result is the number of whole dozens in the total.

Step 16: The number of whole dozens is multiplied by 12 to convert them to units.

Step 17: The number of whole dozens given in units is subtracted from total units. The result is the fraction of a dozen remaining. The minus branching of step 17 impulses PS 1 so that the process of step 18, which is a testing step, may be changed.

Step 18: The fraction of a dozen is subtracted from 9 to determine whether the value is 9 or less, or 10 or 11. Note that if the fraction of a dozen is negative, it must be added rather than subtracted to obtain the same result. If the value is 9 or less, it is ready to be set in the punching dies. Column 1 of storage S4 is wired normally, 1-9, to column 31.

If the fraction is 10 or 11, one additional step is required, and the program is routed to step 19 rather than to set.

Step 19: If the fraction is 10/12, a remainder of -1 is located in S3 after the subtraction in step 18; if the fraction is 11/12, a remainder of -2 is located in S3. This remainder is multiplied by 330, resulting in -330 if the fraction is 10/12, and -660 if the fraction is 11/12. The result is placed in S4, where the fraction of a dozen was located. This clears the 10 or 11 from S4, leaving column 1 of S4 blank. The 3 positions in storage columns 2 and 3 are wired to punch 0 and 5, the code for 10/12; the 5 positions in storage columns 2 and 3 are wired to punch 3 and 7, the code for 11/12. Note that although the 3, 5, and 7 positions of column 31 are Y-wired, it is known that when the setting occurs from storage column 1, columns 2 and 3 are blank; when setting occurs from columns 2 and 3 column 1 is blank. Therefore this Y-wiring is permissable. The program chart and wiring for this are:

SYM	CARD FIELD TITLE	NEG. COL	NEG. POS	OUTPUT PUNCHING 10	9	8	7	6	5	4	3	2	1	DEC LOC	SET 1	SET 2	CLEAR	SYM
S1	WHOLE DOZ.	27	0							27	28	29	30	1/0	X			S1
S2														1/0				S2
S3														1/0				S3
S4	STG. COL. 2: POS. 3→5/31, POS. 5→7/31 FRACT. DOZ. STG. COL 2: POS. 3→9/31, POS. 5→3/31	C11									SEE NOTE 31			1/0	X			S4
S5																		S5
S6																		S6
S7																		S7
S8																		S8
S9																		S9
S10																		S10
S11																		S11
S12																		S12

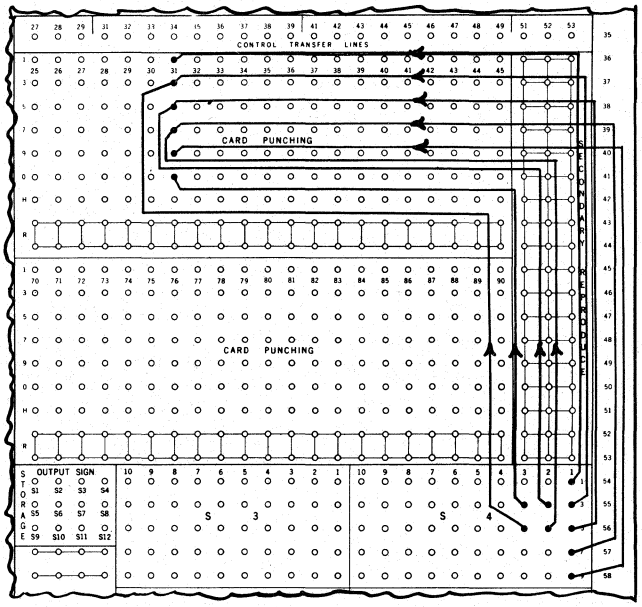
S1-1369 RRBH 12456

FROM	TO
START	
SET 1	
SET 2	
SORT 1	
SORT 2	
CLEAR	
P.S. I	ST. 18
P.S. II	
P.S. III	
P.S. IV	

REPRODUCE CONTROLS	COL.	POS.
REP		
SKIP		
SET HOLD		
SEC REP		

	0 ÷ 0	N ÷ 0
STOP		
SORT		

RESTART	
STEP	
PROGRAM	



23. Program select loop

A. GENERAL

A program select loop is a means of reusing the same step or series of steps several times during the same card without being limited by the four program selects available. With four program selects it is possible to go through the same series of steps a maximum of five times, using different values each time. The case may arise, however, where more than five values should be substituted, or where one or two of the program selects are used for other purposes. In such cases, two program selects may be impulsed, dropped, and impulsed again an almost unlimited number of times. This reuse of program selects is called a program select loop.

Two of the methods used in setting up program select loops are discussed below. One requires three poles of a four-pole selector for each loop through the same step. This leaves one pole for the portion of the step which is to be changed. It is possible to associate more than one selector with each loop, so that more than one choice may be made. The other method requires only two poles of a four-pole selector, but also requires a single-pole selector as a controlling selector.

In connection with both of these methods, it is well to have a brief review of certain features of selectors. If a program select has been impulsed, a connection to ground is made from the pick-up of the selectors associated with the program select. This is through the program select ground hubs (constant-program panel, line 34, A-X). This connection, which holds the selectors on the select side, normally remains until trip time.

Control commons (input-output panel lines 20-26, v-x) are also means of allowing current from the pick-up of a selector to reach ground, thereby holding a selector on the select side. Control commons are usually used in wiring control positions in a card to pick up selectors. However, if the pick-up of a selector reaches a control common by any means, the selector will remain select until the connection is broken or until trip time.

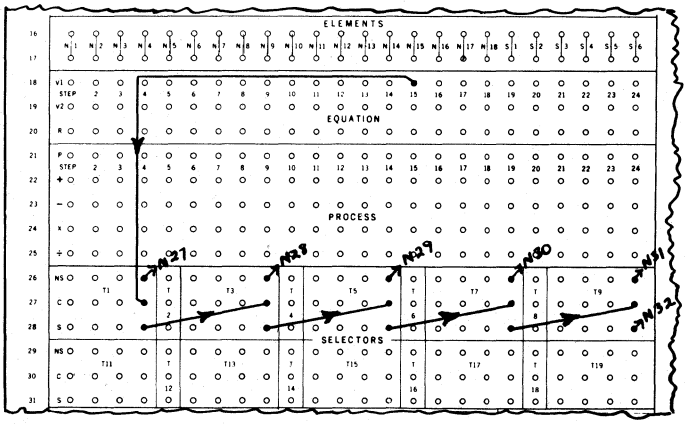
If a program select has been impulsed and the program select ground hubs are connected to control common hubs, the program select will be "dropped out" immediately. The reason is that the power from the pick-up of the selector uses the way of least resistance in going to ground, which happens to be control common rather than program select. Referring to the diagram on page 67, coil H is no longer magnetized, so the connection at E is broken. The program select is thereby deactivated, and, through selectors, may be used to pick up a different selector the next time it is impulsed in the same card.

B. USING THREE POLES OF A SELECTOR FOR EACH LOOP

Assume that step 15 is to be repeated six times. As value 1 of step 15, six different elements are to be used.

1st time through:	N 27
2nd time through:	N 28
3rd time through:	N 29
4th time through:	N 30
5th time through:	N 31
6th time through:	N 32

Using the 4th pole of four-pole selectors to make the choice, the wiring would be standard for the reuse of program steps.

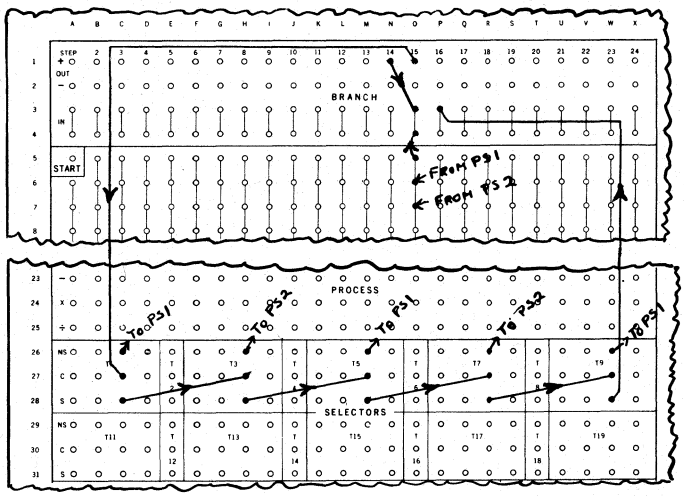


The sequence by which the selectors are used is as follows:

Loop #	Selectors		V1 of Step 15 is wired through		To	Picked up by PS at end of step
	Select	Non-Select	Select	N-S		
1	---	T1,3,5,7,9	---	T1-4	N27	T1
2	T1	T3,5,7,9	T1-4	T3-4	N28	T3
3	T1,3	T5,7,9	T1-4,3-4	T5-4	N29	T5
4	T1,3,5	T7,9	T1-4,3-4,5-4	T7-4	N30	T7
5	T1,3,5,7	T9	T1-4,3-4,5-4,7-4	T9-4	N31	T9
6	T1,3,5,7,9	---	T1-4,3-4,5-4,7-4,9-4	---	N32	

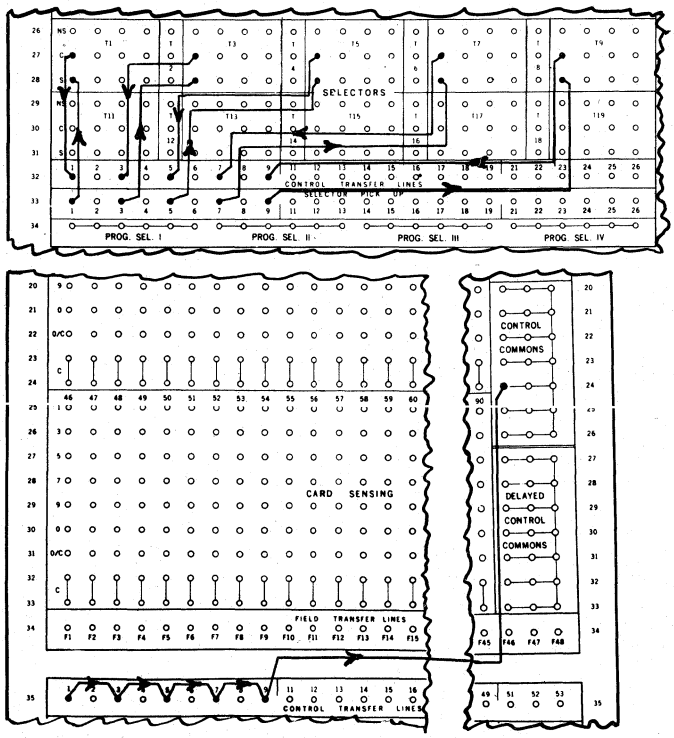
The pick-up of the selectors must be done alternately by two program selects. If PS 1 is used to pick up selector T1, PS 2 would be used to pick up T3. PS 1 can be reused for T5, PS 2 for T7, and PS 1 for T9.

The determination of which program select will be used may be made through these same selectors. The exit of step 14 is wired to step 15. The branching of step 15 is wired to the common of T1-3. Depending on which selectors have become select, the exit of step 15 will be routed to PS 1, PS 2, or in the case of the sixth loop, through the select side of T9-3 to step 16. Both PS 1 and PS 2 are wired to enter step 15.



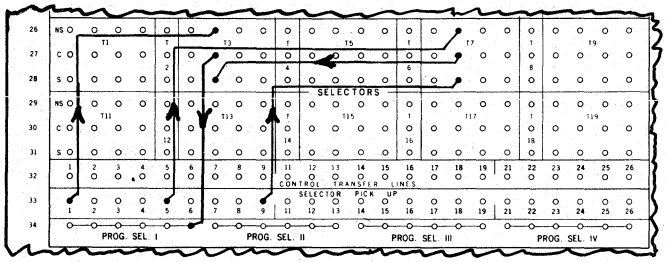
Since it is necessary to pick up each selector by a program select, and keep the selector picked up by a control common, the pick up of each selector is wired to both. If the pick-up of a selector is wired through the select side of one of the poles of the same selector to a control common, as soon as the selector becomes select the current from the pick-up can reach control common ground. The selector will remain select until trip time when the connection is automatically broken.

This wiring would be as follows.

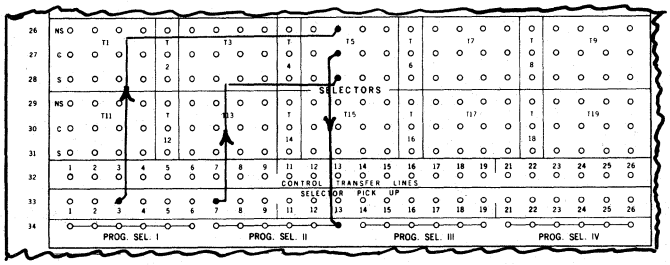


Program select ground is wired through the non-select side of the selector which will be picked up on the following loop. From there it is wired to the pick-up of the selector which is to be picked up on the current loop. That is, PS 1 ground is wired through the non-select of T3 to the pick-up of T1; it is also wired through the select of T3 and non-select of T7 to pick up T5. PS 2 ground is wired through the non-select of T5 to pick up T3, and through the non-select of T9 to pick up T7. The final selector to be picked up by each program select may be wired through the select side of the selector which was picked up on the preceding loop - that is, T9 may be picked up through the select side of T7. This may be shown as follows:

PS 1



PS 2



In the above example, since T7 is the last selector to be picked up by PS 2, the pick-up could be wired either through the non-select side of T9 or the select side of T5.

To correlate all of the above wiring:

First loop:

1. N27 is called upon through the non-select of T1-4.

2. The exit of step 15 is wired to PS 1 through the non-select of T1-3.

3. T1 is picked up by PS 1 through the non-select of T3-2.

4. The pick-up of T1 reaches control common ground and is held select through the select of T1-1.

5. Because control common ground and PS 1 ground are wired together through the pick-up of T1, PS 1 drops out.

6. From PS 1 the program returns to step 15.

Second loop:

1. N28 is called upon through the select of T1-4 and the non-select of T3-4.

2. The exit of step 15 is wired to PS 2 through the select of T1-3 and the non-select of T3-3.

3. T3 is picked up by PS 2 through the non-select of T5-2.

4. T3 is held select through the select of T3-1.

5. PS 2 drops out.

6. From PS 2 the program returns to step 15.

Note: If a control common is not wired to hold T1, during this loop T1 will become non-select. During the time that T3 is changing from non-select to select, the connection from PS 1 ground to the pick-up of T1 through T3-2 is broken, and this break is sufficient to return T1 to non-select. In order for the program select loop to operate correctly, a selector which has been picked up must remain select until the final loop has been completed.

Third loop:

1. N29 is called upon through the select of T1-4, T3-4, and the non-select of T5-4.

2. The exit of step 15 is wired to PS 1 through the select of T1-3, T3-3, and the non-select of T5-3.

3. T5 is picked up by PS 1 through the select of T3-2 and the non-select of T7-2.

4. T5 is held select through the select of T5-1.

5. PS 1 drops out.

6. From PS 1 the program returns to step 15.

Fourth loop:

1. N30 is called upon through the select of T1-4, T3-4, T5-4, and the non-select of T7-4.

2. The exit of step 15 is wired to PS 2 through the select of T1-3, T3-3, T5-3, and the non-select of T7-3.

3. T7 is picked up by PS 2 through the select of T5-2. If PS 2 were to be used again, the wiring to the pick-up of T7 would be through the select of T5-2 and the non-select of T9-2.

4. T7 is held select through the select of T7-1.

5. PS 2 drops out.

6. From PS 2 the program returns to step 15.

Fifth loop:

1. N31 is called upon through the select of T1-4, T3-4, T5-4, T7-4, and the non-select of T9-4.

2. The exit of step 15 is wired to PS 1 through the select of T1-3, T3-3, T5-3, T7-3, and the non-select of T9-3.

3. T9 is picked up by PS 1 through the select of T3-2 and T7-2. If PS 1 were to be used again, the wiring to the pick-up of T9 would be through the select of T3-2, T7-2, and the non-select of T11-2.

4. T9 is held select through the select of T9-1.

5. PS 1 drops out.

6. From PS 1 the program returns to step 15.

Sixth loop:

1. N32 is called upon through the select of T1-4, T3-4, T5-4, T7-4, and T9-4.

2. The exit of step 15 is wired to step 16 through the select of T1-3, T3-3, T5-3, T7-3, and T9-3. The program select loop is complete.

If more than one type of choice must be made by means of the program select loop, additional selectors may be wired for each loop. The pick-up of each selector used above will probably be wired to a bus, since each pick-up is wired both to PS ground and control common. Any additional selectors needed would be wired to the same bus. For example, to pick up both T1 and T11 at the end of the first loop, the only additional wiring necessary is the pick-up of T11 to the bus to which the pick-up of T1 is wired.

On the selector chart, the wiring examples used above would be entered as follows:

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1		SEL. T1-1 (CTL. COM.)	P.U. T1	C1 (CTL. COM)	-
1-2					
1-3		N.S. T3-2 (PS1)	COM. T3-3	+ BR. STEP 15	PS1
1-4			COM. T3-4	V1, STEP 15	N27
2					
3-1		SEL. T3-1 (CTL. COM)	P.U. T3	C3 (CTL. COM)	-
3-2			COM. T7-2	PS1 GROUND	P.U. T1
3-3		N.S. T5-2 (PS2)	COM. T5-3	SEL. T1-3	PS2
3-4			COM. T5-4	SEL. T1-4	N28
4					
5-1		SEL. T5-1 (CTL. COM)	P.U. T5	C5 (CTL. COM.)	-
5-2			P.U. T7	PS2 GROUND	P.U. T3
5-3		N.S. T7-2 (PS1)	COM. T7-3	SEL. T3-3	PS1
5-4			COM. T7-4	SEL. T3-4	N29
6					
7-1		SEL. T7-1 (CTL. COM)	P.U. T7	C7 (CTL. COM.)	-
7-2			P.U. T9	SEL. T3-2	P.U. T5
7-3		SEL. T5-2 (PS2)	COM. T9-3	SEL. T5-3	PS2
7-4			COM. T9-4	SEL. T5-4	N30
8					
9-1		SEL. T9-1 (CTL. COM.)	P.U. T9	C9 (CTL. COM)	-
9-2					
9-3		SEL. T7-2 (PS1)	STEP 16	SEL. T7-3	PS1
9-4			N32	SEL. T7-4	N31

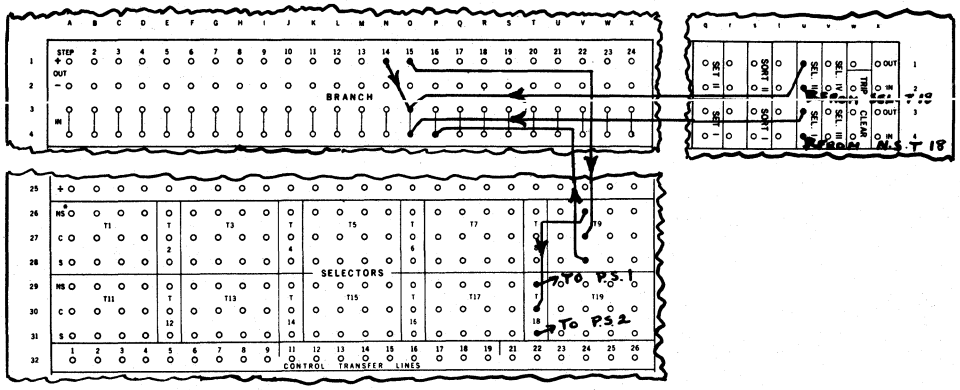
C. USING TWO POLES OF A SELECTOR FOR EACH LOOP

A second method of wiring a program select loop is through use of a control selector instead of the third pole of each selector, making the third pole available for another choice. If two types of choice were to be made in the previous example instead of one, this method would save four selectors. Although it requires one controlling selector, the other method would require five additional selectors - one additional for each that is used in the original problem.

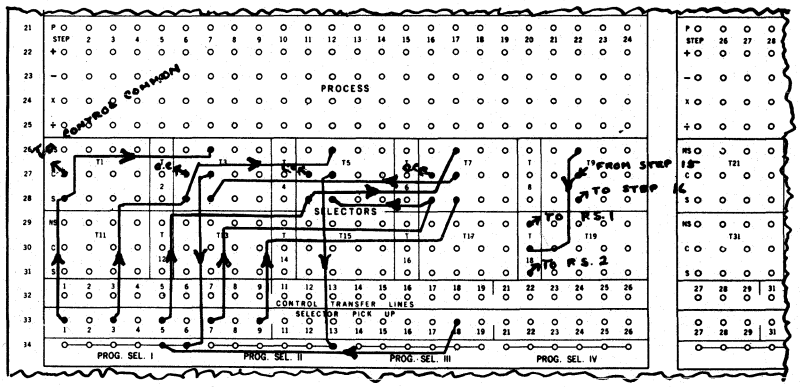
With this second method the choices to be made would be wired in the same way as before - that is, like the fourth poles in the above example. Both the third and fourth poles may be wired in this fashion. The first poles, containing the control common wiring to keep a selector on the select side once it has been picked up, are also identical. With the exception of an additional wire, the second poles, which indicate which selector is to be picked up, are the same. The difference lies in the use of T18, in this case, as a controlling selector. T18 will determine whether PS 1 or PS 2 is to be impulsed next.

The exit of step 15 is wired to the common of the second pole in T9. T9 is the last selector which is picked up, the pick-up occurring at the end of the fifth loop. The non-select side of T9-2 is wired to the common of T18, while the select side is wired to the entry of step 16. This serves the purpose of getting out of the loop in the same way that the select side of T9-3 did in the first example. The non-select of T18 is wired to impulse PS 1, while the select side impulses PS 2. A "flip-flop" will be set up so that T18 will alternate between non-select and select.

The wiring discussed in the preceding paragraph would be:



To control the flip-flop operation of T18, the only wire needed is one from PS 1 ground to the pick-up of T18. To explain how it operates, the wires which are concerned are combined on the following diagram.



First loop: The first time through the loop, all selectors are non-select. Through the non-select of T9-2 and T18, PS 1 is impulsed. PS 1 picks up T18, and through the non-select of T3-2 picks up T1. As soon as T1 is picked up, the control common wired into T1-1 acts both to hold T1 select and to drop out PS 1. However, T18 is also held select because the current emitted from its pick-up flows through PS 1 ground and the non-select of T3-2 to the select of T1-1 and control common ground.

Second loop: At the end of the second loop, T18 is select, and the exit of step 15 is routed through the non-select of T9-2 and the select of T18 to PS 2. PS 2 ground is wired through the non-select of T5-2 to the pick-up of T3. The control common wired into T3-1 holds T3 select and drops out PS 2. When T3 becomes select, it is impossible for the current emitting from T18 pick-up to reach control common ground through the non-select of T3-2. T18 therefore becomes non-select.

Third loop: At the end of the third loop, T18 is non-select, and the exit of step 15 is routed through the non-select of T9-2 and the non-select of T18 to PS 1. PS 1 ground is wired through the select of T3-2 and the non-select of T7-2 to the pick-up of T5. It is also wired to the pick-up of T18. The control common in T5-1 holds T5 select and drops out PS 1. Current emitting from the pick-up of T18 flows through PS 1 ground to the select of T3-2, the non-select of T7-2, and the select of T5-1 to control common ground, thereby keeping T18 select.

Fourth loop: At the end of the fourth loop, T18 is select, and the exit of step 15 is routed through the non-select of T9-2 and the select of T18 to PS 2. PS 2 ground is wired through the select of T5-2 to the pick-up of T7. The control common wired into T7-1 holds T7 select, and drops out PS 2. Since T7 has become select, it is impossible for the current emitting from T18 pick-up to reach control common ground in T5-1, so T18 becomes non-select.

Fifth loop: At the end of the fifth loop, T18 is non-select, and the exit of step 15 is routed through the non-select of T9-2 and the non-select of T18 to PS 1. PS 1 ground is wired through the select of T3-2 and the select of T7-2 to the pick-up of T9. Although PS 1 is also wired to the pick-up of T18, this is of no significance at this point, since T9 has become select, and on the next loop the exit of step 15 will not be routed to T18.

Sixth loop: At the end of the sixth loop, T9 is select. Therefore the exit of step 15 is routed through the select of T9-2 to the entry of step 16, and the loop is completed.

The wiring for the above loop would be entered on the selector chart as follows:

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
11-1					
11-2					
11-3					
11-4					
12					
13-1					
13-2					
13-3					
13-4					
14					
15-1					
15-2					
15-3					
15-4					
16					
17-1					
17-2					
17-3					
17-4					
18		P.S.1	To P.S.2	N.S. T 9-2	To P.S.1
19-1					
19-2					
19-3					
19-4					

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1		SEL. T1-1 (CTL. COM.) N.S. T 3-2 (P.S.1)	P.U. T1	C1 (CTL. COM.)	
1-2					
1-3					
1-4					
2					
3-1		SEL. T 3-1 (CTL. COM.) N.S. T5-2 (P.S.2)	P.U. T3	C3 (CTL. COM.)	
3-2			COM. T 7-2	P.S.1 GROUND	P.U. T1
3-3					
3-4					
4					
5-1		SEL. T5-1 (CTL. COM.) N.S. T 7-2 (P.S.1)	P.U. T5	C5 (CTL. COM.)	
5-2			P.U. T7	P.S.2 GROUND	P.U. T3
5-3					
5-4					
6					
7-1		SEL. T 7-1 (CTL. COM.) SEL T5-2 (P.S.2)	P.U. T7	C7 (CTL. COM.)	
7-2			P.U. T9	SEL. T 3-2	P.U. T5
7-3					
7-4					
8					
9-1					
9-2			STEP 16	+ BR. STEP 15	COM. T18
9-3		SEL. T 7-2 (P.S.1)			
9-4					

24. Punching zeros from storage

Storage outputs on the 60 and 120 do not have zero positions which will punch directly into the card. Normally this is unimportant in fields which are not designating, since the printing of zeros by the tabulator does not depend on their being punched in the card. Techniques have been developed, however, to punch preceding zeros, which may be of particular use in a payroll where net pay amount may need preceding astericks for check protection, trailing zeros, and both preceding and trailing zeros.

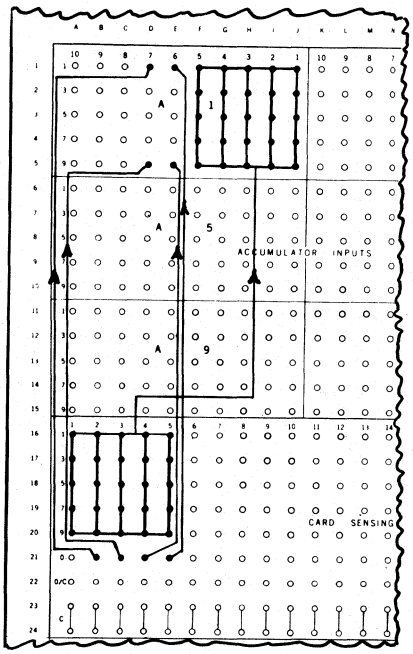
The punching of zeros in designating fields is a more common requirement, since frequently summary cards are created which must be sorted with detail cards in which zeros are punched. The absence of zeros in such cases would cause the tabulator to break control where it should not do so.

A. ZEROS FROM DESIGNATING FIELDS

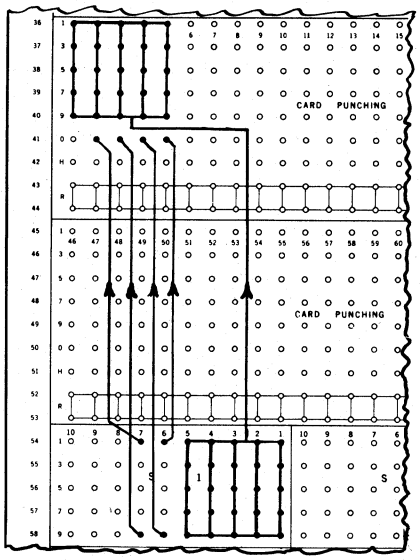
In those cases in which zeros must be punched in designating fields, the card from which the designating information is stored usually is punched with the necessary zeros. If this is not the case, technique B, page 199, technique C, page 203, or technique D, page 207, must be used. With the necessary zeros punched in the first card of a group, however, the problem is relatively simple.

If the number to be stored is not larger than seven digits, wiring similar to that used for alpha will permit the punching of zeros from the same storage in which the designating information is stored for testing. If another storage is available, ten columns of designating information can be handled in this way. The zeros are changed to numeric combinations for storage, and wired from the numeric combinations to punch as zeros. One storage column is required for punching zeros in two card columns. A five-digit field, which could have four trailing zeros, would require two additional storage columns; a seven-digit field could have six trailing zeros, which would require three additional columns of storage. This of course would use all ten available columns in the storage unit.

The basic wiring for this is as follows. A zero in one card column is wired to the 1 position in an accumulator input column; the zero in another card column is wired to the 9 in the same accumulator input column. Assuming a five-digit number punched in card columns 1-5, the wiring on the input-output panel for placing the value in storage would be:



To punch the value from storage S1 into columns 1-5, the wiring would be:



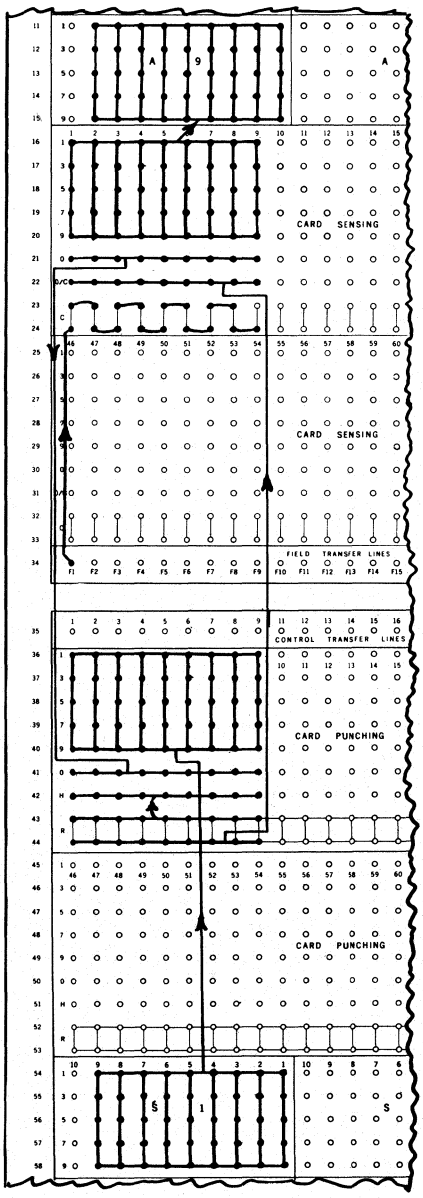
Note that the only way in which this will affect the testing of designating information from card to card is that the number as read by the computer will not necessarily be in ascending sequence, due to the creation of numeric values from zeros. Therefore, as when alpha is wired, the first card of a group is recognized at more than one branching of the first step if the designating information is not cleared on a preceding summary card (see page 120.)

Note, however, that if the zero wiring follows the numeric wiring instead of preceding it, this problem does not occur. In the above example, the 1-9 wiring for card columns 5-1 would be placed in storage columns 7-3, while the zero wiring would be placed in storage columns 2-1. Some sample values would enter storage as follows:

Value as punched	Value as stored
50000	5000022
50001	5000121
50010	5001029
50100	5010012

If the designating field contains more than seven digits, one solution is the use of an additional storage which would contain the zero wiring. If this is not available, however, it is possible to reproduce the zero positions while reading the significant digits as an element, using them for testing, and punching them from storage into the card. This is possible because zero has a separate common from the rest of the positions in a column. The zero and zero common are wired as for normal reproducing, while the remaining positions make use of standard element wiring. In addition to this wiring, it is necessary to have a reproduce control hole in the card from which the zeros are to be reproduced.

The wiring for this is as follows:



B. PUNCHING PRECEDING ZEROS

As mentioned on page 196, preceding zeros are sometimes required on a payroll application. In writing checks on the tabulator, it is usually necessary to fill in the spaces preceding the first significant digit of net amount. If zeros are punched in the card in these spaces, the zeros may be wired on the tabulator to the 37th position, which could contain asterisks. In the example which follows, zeros following the first significant digit are unnecessary, since the tabulator will print them automatically.

To punch the preceding zeros, it is necessary to set up a program select loop. Two poles of a selector are required for each possible preceding zero, so the

loop explained on page 193 would be the most satisfactory. Assuming that net pay is a six-digit field to be punched in columns 63-68, it will be necessary to test the net pay storage a maximum of six times, testing each digit in turn to determine whether it is blank. If it is, a zero is to be punched in the corresponding card column and the adjacent column to the right is to be tested. If the digit is significant, no zero is to be punched and the testing is completed. The program select loop would be set up exactly as shown on page 193, leaving poles 3 and 4 of each selector available for the zero punching routine. Assume that net pay is in storage S2, with a 3/2 decimal location. Step 15 involves the subtraction of a constant value from S2 to determine the size of the net pay amount. If net pay is $90.50, the loops will be:

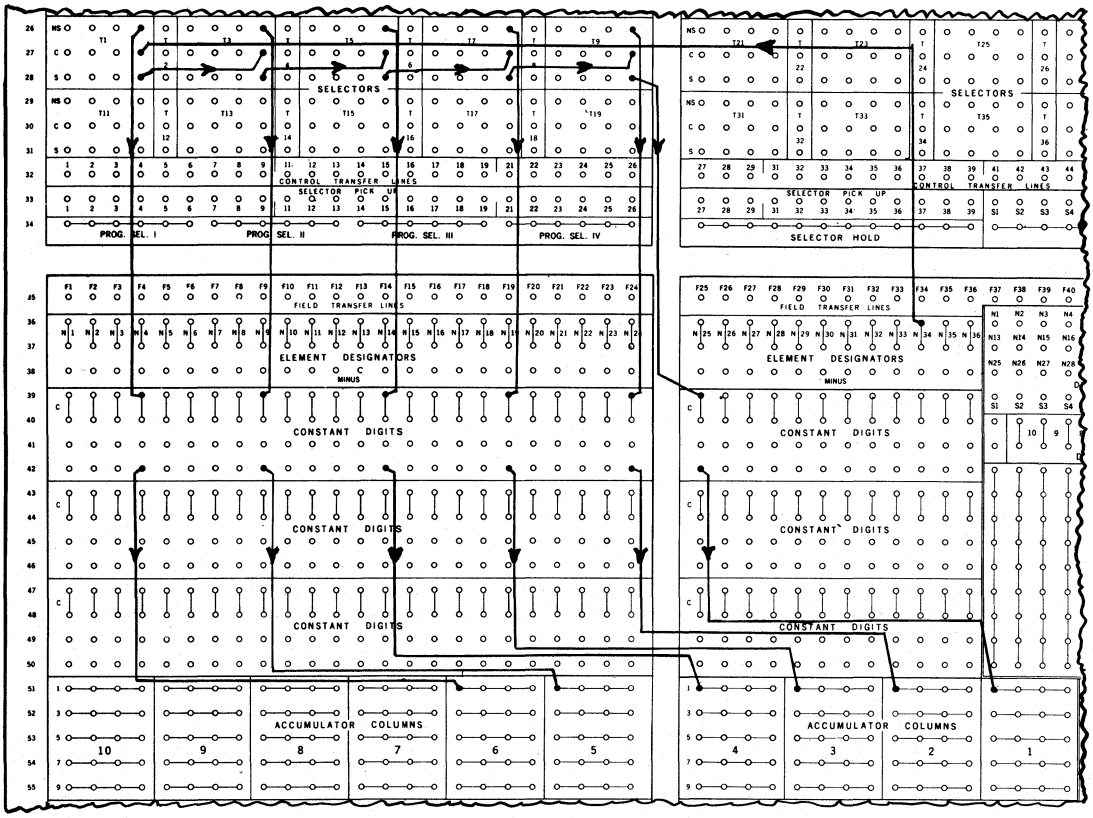
1st loop:	90.50	
	-1000.00	
	- 909.50	Punch zero in column 63, return to step 15
2nd loop:	90.50	
	- 100.00	
	- 9.50	Punch zero in column 64, return to step 15
3rd loop:	90.50	
	- 10.00	
	+ 80.50	Do not punch zero in column 65, go to step 16

As long as the result of the subtraction is minus, a zero is to be punched and the loop is to return to step 15. When the result is plus, the loop has been completed.

If the constant value is N34, which will successively be made 1000, 100, 10, 1, .1, and .01 as shown below, step 15 would be:

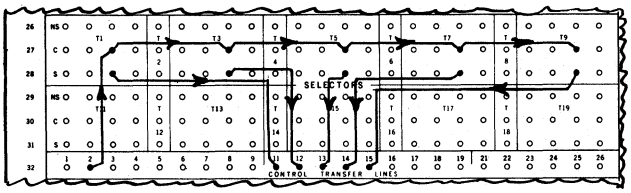
STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
												4/3	3/2												
15		NET PAY	+	S2	−	1000/100/10/1/.1/.01	+	N34	TEST + DO NOT PUNCH ZERO − PUNCH ZERO	+ −	S1	15												COM. T9-2	16

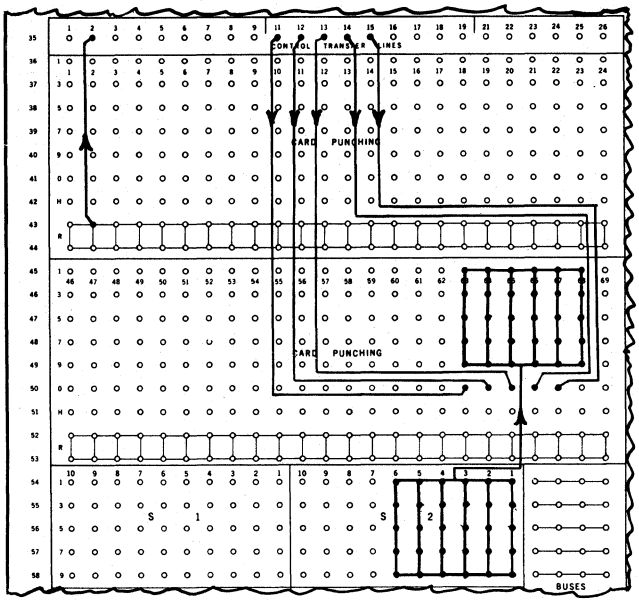
The changing of element N34 into the six values listed above is done through one pole of each selector used in the program select loop. N34 is assigned a 3/2 decimal location. The element designator is wired through the selectors to the constant digit required on each loop, as follows:



Note that it would also be possible to assign each value (1000, 100, 10, etc.) a separate element number, and select V2 of step 15 instead of the value of N34. It would also be possible to wire N34 as a value of 1 in accumulator column 6, and change the decimal location. On the first loop the decimal would be 3/2, the second 4/3, the third 5/4, etc.

The card into which net pay will punch is wired to reproduce, either by means of a control hole or program select. The third pole of each selector is used for the punching of zeros, using the reproduce function to do so. Reproduce power, which may come from any of the hubs in lines 43-44 and 52-53, A-u, is wired to the common of each of the poles. The power is transferred to the constant-program panel by a control or field transfer line, and bused on this panel. The select side of each selector is wired through a control transfer line to punch the zero. If the first result of step 15 is plus, the program continues with step 16 immediately, no selectors become select, and no zeros are punched. If the first result of step 15 is minus, selector T1 becomes select because PS 1 is impulsed; therefore a zero is punched in column 63. The wiring for the third selector pole is as follows:





On the selector chart, the third and fourth selector poles would be entered as follows:

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1					
1-2					
1-3			C11 (0/63)	C2 (REP. POWER)	—
1-4			COM. T 3-4	ELEM. DEC. N3H	1000
2					
3-1					
3-2					
3-3			C12 (0/64)	C2	—
3-4			COM. T5-4	SEL. T 1-4	100
4					
5-1					
5-2					
5-3			C13 (0/65)	C2	—
5-4			COM. T 7-4	SEL. T 3-4	10
6					
7-1					
7-2					
7-3			C14 (0/66)	C2	—
7-4			COM. T9-4	SEL. T5-4	1
8					
9-1					
9-2					
9-3			C15 (0/67)	C2	—
9-4			.01	SEL. T 7-4	.1

C. PUNCHING PRECEDING AND TRAILING ZEROS

The following program is one for punching both preceding and trailing zeros. It uses a program select loop, as is the case with the program for preceding zeros. Since this also requires two poles of a four-pole selector for each zero to be punched, the same loop will be used as in the previous example. However, one additional loop is required, a four-pole controlling selector (T23) is used instead of T18, and there is a change in the exit of the loop due to the fact that this method requires more program steps.

To set up the routine, one selector is required for each zero to be punched, plus a four-pole controlling selector. Two program steps are necessary before the loop, plus two iterative steps for the loop. Four storages are needed in addition to the one from which the value will be punched. One of the storages is made negative, and the output sign is used for setting the zeros to be punched. The other three storages are working storages used during the loops. Two constants are needed for the program:

N36 - Zero 4/3 decimal location
N34 - 10,000 1/0 decimal location

For purposes of comparison with the other method, the value in which the zeros are to be punched will be net pay, located in S2 with a 3/2 decimal location. The card columns in which it will be punched are 63-68.

One of the two preliminary steps before entering the loop is the adjusting of the decimal location of net pay. The decimal must be placed so that it will precede the first zero to be punched. In this case the maximum size of the field is six digits, so a 7/6 decimal is needed.

To change a 3/2 decimal location to a 7/6 decimal, the value can be divided by 10,000. For example:

S2 ÷ N34 = S4 90.50 ÷ 10,000 = .009050

The other preliminary step is making a storage minus so that the negative sign may be used for setting the zeros. This may be accomplished by subtracting a number from zero. Since a constant of 10,000 has been set up, this may be used.

N36 - N34 = S1 Zero - 10,000 = -10,000

Assuming that as of the end of step 14 net pay has been computed, the two preparatory steps would be entered on the program chart as follows:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 1/0	S2 3/2	S3	S4 7/6	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP −	NEXT STEP +
15		NET PAY (XXXX.XX)	+	S2	÷	10,000	+	N34	NET PAY (.XXXXXX)	+	S4				15									−	16
16		ZERO	+	N36	−	10,000	+	N34	MAKE S1 NEGATIVE	−	S1	16												17	−

The first step in the looping routine is the subtraction of net pay from zero. The result is first placed in a storage with a 2/1 decimal location, so only the first digit is entered. If the first digit is zero, the result is plus; while if it is significant, the result is minus. From a plus branching the routine goes to a set instruction, with the negative sign of S1 used to punch a zero in column 63. With each loop, whenever the result is plus this same negative sign is used to punch a zero in the proper column. If the result is minus, the program continues with the next step. For example:

Step 17: N36 - S4 = S5	Next instruction
a) 0 - .009050 = +.0	Set 2
b) 0 - .119050 = -.1	Step 18

During the following step the above result (+.0 or -.1) is added to net pay. The effect is to change the digit which has just been tested to zero.

Step 18: S4 + S5 = S6

a) .009050 + .0 = .009050
b) .119050 + (-.1) = .019050

The program then returns to the preceding step, using S6 instead of S4. This time the result of the subtraction is placed in a storage with a 3/2 decimal.

Step 17: N36 - S6 = S5

a) 0 - .009050 = +.00
b) 0 - .019050 = -.01

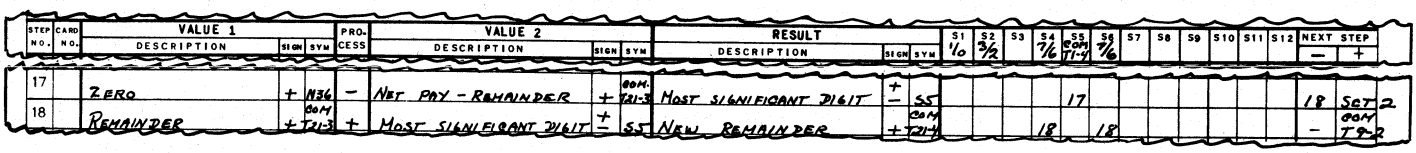
Actually the same storage is always used for the result of this step, with the decimal location changed by the same selector which determines the column in which the zero is punched. On the following step, the net pay amount is again adjusted.

Step 18: S6 + S5 = S4

a) .009050 + .00 = .009050
b) .019050 + (-.01) = .009050

Note that the two steps alternate in their use of S4 and S6. This is controlled through the same selector which governs whether the end of the loop is to go to PS 1 or PS 2.

The two program steps used in the loop would be entered on the program chart as follows:



SYM	CARD FIELD TITLE	NEG. COL	NEG. POS	OUTPUT PUNCHING 10	9	8	7	6	5	4	3	2	1	DEC LOC	SET 1	SET 2	CLEAR	SYM
s1	"0" FOR PUNCHING	COM. T1-3												1/0		X		s1
s2	NET PAY							63	64	65	66	67	68	3/2	X			s2
s3																		s3
s4														7/6				s4
s5														COM T1-4				s5
s6														7/6				s6
s7																		s7
s8																		s8
s9																		s9
s10																		s10
s11																		s11
s12																		s12

FROM	TO
START	
SET 1	
SET 2	STEP 18
SORT 1	
SORT 2	
CLEAR	
P.S. I	STEP 17
P.S. II	STEP 17
P.S. III	
P.S. IV	

REPRODUCE CONTROLS	COL.	POS.
REP		
SKIP		
SET HOLD		
SEC REP		

	0 ÷ 0	N ÷ 0
STOP		
SORT		

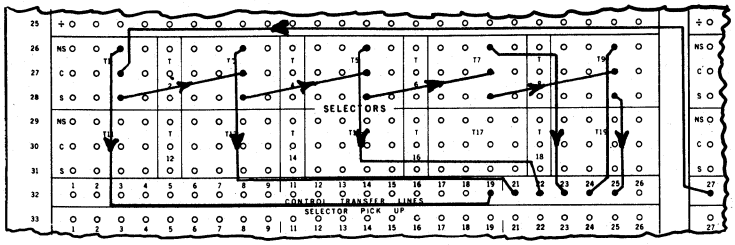
RESTART	
STEP	
PROGRAM	

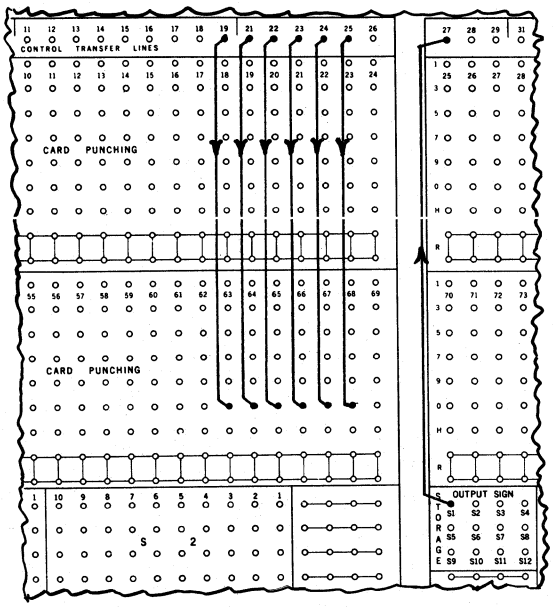
S1-1369 RRBH 12456

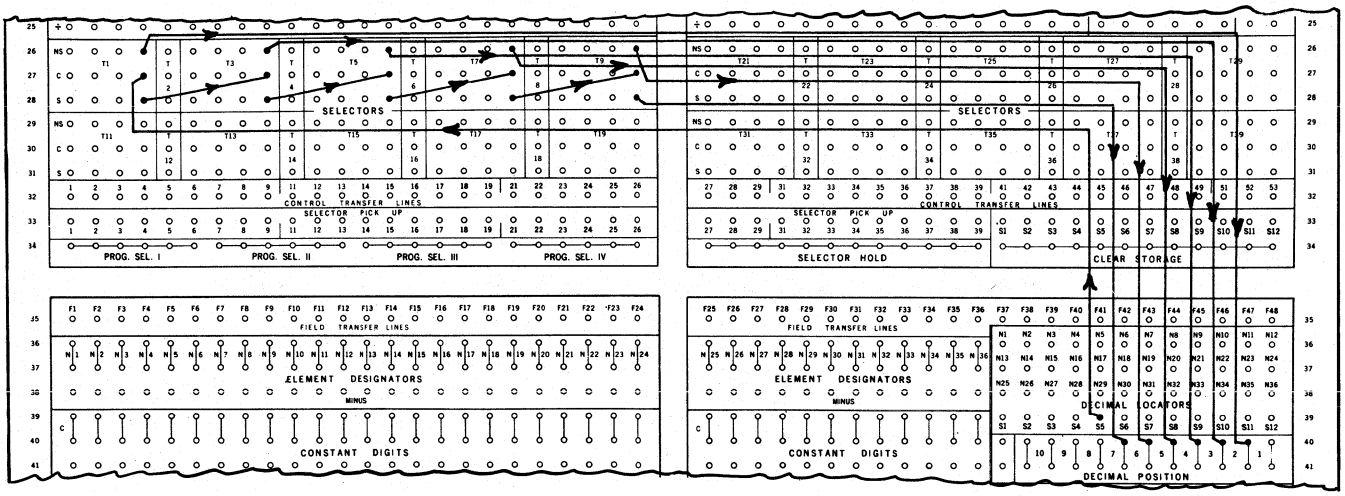
The selector chart, which includes the wiring for setting up the loop, as well as the choices made during the loop, is filled in as follows:

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1		SEL. T1-1	P.U. T1	C1 (CTL. COM.)	
1-2		(CTL. COM.)			
1-3		N.S. T3-2	COM. T 3-3	C27(S1-OUTPUT SW)	C19 (0/63)
1-4		(P.S. 1)	COM. T 3-4	S5 - DEC. LOC.	2/1
2					
3-1		SEL. T 3-1	P.U. T 3	C3 (CTL. COM.)	
3-2		(CTL. COM.)	COM. T 7-2	P.S.1 GROUND	P.U. T1
3-3		N.S. T5-2	COM. T5-3	SEL. T1-3	C21 (0/64)
3-4		(P.S. 2)	COM. T5-4	SEL. T1-4	3/2
4					
5-1		SEL. T5-1	P.U. T5	C5 (CTL. COM)	
5-2		(CTL. COM)	P.U. T7	P.S.2 GROUND	P.U. T 3
5-3		N.S. T 7-2	COM. T 7-3	SEL. T3-3	C22 (0/65)
5-4		(P.S.1)	COM T 7-4	SEL. T 3-4	4/3
6					
7-1		SEL. T 7-1	P.U. T 7	C7 (CTL. COM.)	
7-2		(CTL. COM.)	P.U. T9	SEL. T 3-2	P.U. T5
7-3		SEL. T 5-2	COM. T9-3	SEL. T5-3	C23 (0/66)
7-4		(P.S. 2)	COM. T9-4	SEL. T5-4	5/4
8					
9-1		SEL. T9-1	P.U. T9	C9 (CTL. COM)	
9-2		(CTL. COM.)	STEP 19	+ BR. STEP 18	COM. T21-2
9-3		SEL. T 7-2	C25 (0/68)	SEL. T 7-3	C24 (0/67)
9-4		(P.S. 1)	7/6	SEL. T 7-4	6/5
21-1					
21-2		P.S. 1	TO P.S. 2	N.S. T9-2	TO P.S. 1
21-3			S6	V2 · STEP 17 V1 · STEP 18	S4
21-4			S4	R. STEP 18	S6

The third and fourth poles of the selectors would be wired as follows:







The computer sequence of steps through which a net pay amount of $90.50 would be calculated using a planning chart is shown below:

Line No.	Step No.	Calculation: Value 1	Sw	Process	Value 2	Sw	Result: S1 1/0	S2	S3	S4 7/6	S5	S6 7/6	S7	S8	S9	S10	S11	S12	Next Step −	Next Step +
1	15	90.50	S2	÷	10,000	[illegible]				.009050										16
2	16	0.000	[illegible]	−	10,000	[illegible]	− 10000												17	
3	17	0.000	[illegible]	−	.009050	S4					.0									SET 2
4	18	.009050	S4	+	.0	S5						.009050								PS1
5	17	0.000	[illegible]	−	.009050	S6					.00									SET 2
6	18	.009050	S6	+	.00	S5				.009050										PS2
7	17	0.000	[illegible]	−	.009050	S4					− .009								18	
8	18	.009050	S4	+	−.009	S5						.000050								PS1
9	17	0.000	[illegible]	−	.000050	S6					.0000									SET 2
10	18	.000050	S6	+	.0000	S5				.000050										PS2
11	17	0.000	[illegible]	−	.000050	S4					− .00005								18	
12	18	.000050	S4	+	.00005	S5						.000000								PS1
13	17	0.000	[illegible]	−	.000000	S6					.000000									SET 2
14	18	.000000	S6	+	.000000	S5				.000000										19
15																				

D. PUNCHING TRAILING ZEROS

The program given below for punching only trailing zeros and those following the decimal point is quite different from those previously described. It does not involve a program select loop, and therefore does not require as many selectors. It does, however, require a large number of storages and steps on the computer. Usually there is little room for much additional calculation in this program. Note, however, that a program select loop could be set up for this program to reduce the number of steps.

This program is similar to the program for punching both preceding and trailing zeros in that an artificial decimal location is created for the value. In this case the decimal location is immediately following the first digit of the value. In the example that has been used, the six-digit field would be given a 6/5 decimal location. Each digit is isolated in turn by adding the value to zero, and placing the result in storages with successively higher decimal locations. The isolated digit is subtracted from the original value so that the column from which it is isolated will become zero. The next digit to the right is then isolated.

Assume that at the end of step 14 the net pay amount is in S2 with a 3/2 decimal location. The program and planning chart are as follows:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/5	S2 3/2	S3 1/0	S4 2/1	S5 3/2	S6 4/3	S7 5/4	S8 6/5	S9 4/5	S10	S11 4/5	S12 4/5	NEXT STEP —	NEXT STEP +
15		NET PAY (XXXX.XX)	+	S2	÷	1000	+	N34	NET PAY (XXXXXX)	+	S1	15												-	16
16		X.XXXXX	+	S1	+	ZERO	+	N36	1ST DIGIT	+	S3			16										-	17
17		X.XXXXX	+	S1	-	1ST DIGIT	+	S3	.XXXXX	+	S11											17		-	18
18		.XXXXX	+	S11	+	ZERO	+	N36	2ND DIGIT	+	S4				18									-	19
19		.XXXXX	+	S11	-	2ND DIGIT	+	S4	.0XXXX	+	S12												19	-	20
20		.0XXXX	+	S12	+	ZERO	+	N36	3RD DIGIT	+	S5					20								-	21
21		.0XXXX	+	S12	-	3RD DIGIT	+	S5	.00XXX	+	S11											21		-	22
22		.00XXX	+	S11	+	ZERO	+	N36	4TH DIGIT	+	S6						22							-	23
23		.00XXX	+	S11	-	4TH DIGIT	+	S6	.000XX	+	S12												23	-	24
24		.000XX	+	S12	+	ZERO	+	N36	5TH DIGIT	+	S7							24						-	25
25		.000XX	+	S12	-	5TH DIGIT	+	S7	6TH DIGIT	+	S8								25					-	26
26		6TH DIGIT	+	S8	-	.00001	+	N33	TEST + NOT ZERO − ZERO	+ −	S9									(26)				27	27
27		5TH DIGIT	+	S7	-	.0001	+	N32	TEST + NOT ZERO − ZERO	+ −	S8								(27)					28	28
28		1ST DIGIT	+	S3	-	1	+	N31	TEST + NOT ZERO − ZERO	+ −	S11											28		29	31
29		2ND DIGIT	+	S4	-	.1	+	N30	TEST + NOT ZERO − ZERO	+ −	S11											29		30	PS1
30		3RD DIGIT	+	S5	-	.01	+	N29	TEST + NOT ZERO − ZERO	+ −	S11											30		34	PS2
31		4TH DIGIT	+	S6	-	.001	+	N28	TEST + NOT ZERO − ZERO	+ −	S7							(31)						COM T2	COM T2
32		3RD DIGIT	+	S5	-	.01	+	N29	TEST + NOT ZERO − ZERO	+ −	S6						(32)							COM T4	COM T4
33		2ND DIGIT	+	S4	-	.1	+	N30	TEST + NOT ZERO − ZERO	+ −	S5					(33)								34	34

SYM	CARD FIELD TITLE	NEG. COL	NEG. POS	OUTPUT PUNCHING 10	9	8	7	6	5	4	3	2	1	DEC LOC	SET 1	SET 2	CLEAR	SYM
S1														4/5				S1
S2	NET PAY							63	64	65	66	67	68	3/2				S2
S3														1/0				S3
S4														2/1				S4
S5		64	0											3/2	X			S5
S6		65	0											4/3	X			S6
S7		66	0											5/4	X			S7
S8		67	0											6/5	X			S8
S9		68	0											4/5	X			S9
S10																		S10
S11														4/5				S11
S12														4/5				S12

S1-1369 RRBH 12456

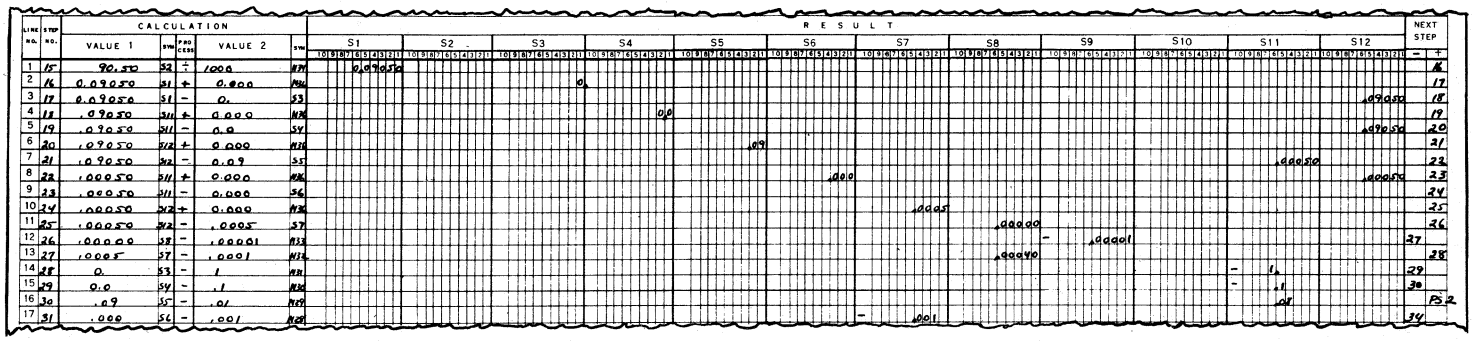
FROM	TO
START	
SET 1	
SET 2	
SORT 1	
SORT 2	
CLEAR	
P.S. I	STEP 31
P.S. II	STEP 31
P.S. III	
P.S. IV	

REPRODUCE CONTROLS	COL.	POS.
REP		
SKIP		
SET HOLD		
SEC REP		

	0 ÷ 0	N ÷ 0
STOP		
SORT		

RESTART	
STEP	
PROGRAM	

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
2		PS2	STEP 34	± BR. STEP 31	STEP 32
3-1					
3-2					
3-3					
3-4					
4		PS1	STEP 34	± BR. STEP 32	STEP 33



LINE NO.	STEP NO.	CALCULATION VALUE 1	SW	PROCESS	VALUE 2	SW	RESULT S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP
1	15	90.50	S2	÷	1000	H1	.09050												16
2	16	0.09050	S1	+	0.000	H2			0										17
3	17	0.09050	S1	−	0.	S3												.09050	18
4	18	.09050	S11	+	0.000	H3				00									19
5	19	.09050	S11	−	0.0	S4												.09050	20
6	20	.09050	S12	+	0.000	H3					09								21
7	21	.09050	S2	−	0.09	S5											.00050		22
8	22	.00050	S11	+	0.000	H4						.000						.00050	23
9	23	.00050	S11	−	0.000	S6													24
10	24	.00050	S12	+	0.000	H3							.0005						25
11	25	.00050	S2	−	.0005	S7								.00000					26
12	26	.00000	S8	−	.00001	H13								−	.00001				27
13	27	.0005	S7	−	.0001	H13								.00040					28
14	28	0.	S3	−	1	H8											− 1.		29
15	29	0.0	S4	−	.1	H8											− .1		30
16	30	.09	S5	−	.01	H9											.08		PS2
17	31	.000	S6	−	.001	H9							− .001						34

During step 15, the value is divided by 1,000 to change the decimal to 6/5. Steps 16-25 are used to isolate the digits in the value. A pair of steps is needed for each digit except the last. In each case the digit is to be stored for later testing.

The first step of the pair involves the transfer of the value to a storage which will drop off all digits to the right of the one being isolated. In the next step this digit is subtracted from the original value to bring the corresponding column in the value to zero. The next pair of steps isolates the next digit to the right. The planning chart, lines 2-11, follows the value of $90.50 through this procedure. Note that for a value containing six digits, only five pairs of testing steps are needed since the last digit is isolated when the fifth digit is subtracted from the value.

The means of punching zeros in this routine is the wiring of the negative indication of storages to punch zeros. If a 1 is subtracted from the isolated digit, the sign of the storage will be plus if the digit is significant and minus if the digit is zero.

.09	.1	0.
− .01	− .1	− 1.
+.08	+.0	-1.

The subtraction takes place into a different storage for each digit.

Steps 26 and 27 (lines 12 and 13 on the planning chart) show the subtraction when the digit follows the actual decimal - in this case, the 5th and 6th digits. Since zero is always to be punched in these columns if there is no significant digit present, only one step is needed for each digit.

For the digit preceding the actual decimal, a zero is to be punched only if there is no significant digit in the card column but there is a significant digit preceding this column in the value. Therefore the value must first be tested to determine the location of the first significant digit, and the digits following this tested for the presence of zeros.

Steps 28-30 (lines 14-16 on the planning chart) are the steps to locate the first significant digit. Only three testing steps are necessary with four digits preceding the decimal, since any zero following the fourth digit also follows the decimal and has already been taken care of.

Program selects are impulsed at the plus branching of the test for the second and third digits. They are used to remember the location of the first significant digit and stop the testing for trailing zeros at this point.

Steps 31-33 (line 17 on the planning chart) are the steps which test for trailing zeros. All cards except those in which none of the first three digits are significant begin on step 31. Each card goes through as many steps as necessary to test for zeros following the first significant digit, branching to step 34, the first step following the zero punching routine, when the testing is complete.

Upon the set 1 instruction, the significant digits in net pay are punched from storage S2. Zeros are punched from the negative sign in those storages which became negative on the testing steps.

25. Techniques for conserving functions

The following list contains suggestions for conserving the existing components of the 60 or 120. Many of them are quite specialized in their application, and therefore cannot be used in all programs. However, by checking this list the programmer may find a way of handling a problem which appears beyond the capacity of the computer.

A. ELEMENTS

1) Wire two card-read fields as one element, page 125.

2) Wire constants through input, page 134.

3) Convert codes to constants, page 140.

4) Make the same element more than one constant value through selectors, pages 184 and 200.

B. STORAGES

1) Set information in punching dies during the program, use these storages as intermediate storages, page 53.

2) Change the decimal location of storage, page 81.

3) Use an element as the result of a step, page 116.

4) Reuse designation storage, page 122.

5) Place more than one value in a storage, pages 125 and 166.

6) Reproduce zeros in designating fields of over seven digits, page 198.

C. STEPS

1) Reuse steps through program selects, page 79.

2) Change the decimal location of storage, page 81.

3) Use an element as the result of a step, page 116.

4) Accumulate by reading two card-read fields as one element, page 125.

5) Crossfooting, page 130.

6) Program select loop, page 187.

D. SELECTORS

1) Pick up a selector by an even number, page 76.

2) Wire more than one choice to the common of a selector, page 81.

3) Card position selection, page 137.

4) Make an element zero or a significant value, page 138.

E. PROGRAM SELECTS

1) Program select loop, page 187.

26. Approaching a program

Many times the question is asked, "Where do you start when working on a program?" There is no simple, single answer to this question. As a programmer becomes more experienced, he will develop certain techniques and habits which he finds are easiest for him. However, the suggestions outlined below may be of assistance in developing a method of approach.

A. GENERAL UNDERSTANDING OF PROBLEM

The first step in programming is to understand the problem as it is handled at present. Whether it is done manually or by tabulating, it is well to draw a rough flow chart of the operations. This will frequently point out portions of the problem which have been omitted in the explanation. The programmer may also discover certain related operations which can easily be incorporated in the program but which were not originally contemplated as being part of it. By continually asking "Why?", it may also develop that certain techniques used with the previous method are actually unnecessary in the computing routine. They may have been included over a long period of time to correct conditions which no longer exist or which will be eliminated by the computer, or there may be an easier method on the computer which would be too cumbersome with the other method.

It is well to extend the general statement of the problem in both directions, including the original source of the data to be processed as well as the use made of the results of the program. This not only helps in understanding the problem, but also may point out what has been omitted and would be desirable, as well as unnecessary computations.

B. DETAILED ANALYSIS OF PROBLEM

In conjunction with preparation for actual programming, the following points should be considered.

B-1 Card forms

A complete set of all card forms entering the routine should be obtained. Preferably they should be prepunched, but if this is impossible, sample entries should be filled in on the cards. This portion of the analysis may be done in conjunction with the step-by-step analysis which follows, obtaining the needed information about each card form as it appears in the routine. The following points should be considered:

1) Differentiation between fields punched prior to the computing run and those which the computer can or should compute.

2) Maximum size of each field. Frequently the size of a field on the card form is larger than any input data on an individual card, or the output data which must be punched in a card.

3) Presence of over-capacity punching.

4) Presence of alpha in the designating field.

5) The decimal location of each field in the card, with particular note of whether a price may be price each, per hundred, per thousand, etc.

6) All card codes which are punched, regardless of whether they have an

apparent bearing on the problem. This should include both the reason for or meaning of the hole, as well as all cards in which it is present.

7) Blank columns or blank control positions which may be used for punching additional controls if necessary.

8) The possibility of relocating overlapping card fields.

B-2 Method of computation

As the programmer makes the detailed analysis of the problem, he should at all times be considering how it will be handled on the computer. At this point it may be possible to do a rough flow chart of the program itself, or at least to make notes of certain features which should be included in the program. This step-by-step analysis is probably the most important feature of preliminary program planning. It is a complete analysis of the method of computation of all final results. If the problem is at all complicated, it is well to carry sample figures through the entire problem to avoid any misunderstanding or omission. The step-by-step analysis should reduce any problem to a point where the basic arithmetic steps through which the computer will solve the problem become apparent.

The following lists may help to serve as a check list of points to be considered as the problem is outlined. The first list refers to all programs, while the second list is particularly applicable to multiple card routines.

All programs

1) What percent of the total volume are exceptions to a general rule? Should these be separated for manual handling, or can the procedure for the exceptions be incorporated in the regular routine?

2) What elements, intermediate results, and final results can be negative? If a value is usually positive, is there any condition under which it might be negative?

3) What elements, intermediate results, and final results might be zero? What effect will this have on the program?

4) What intermediate results and final results should be rounded? When should they be rounded, and in what storage position? Should some type of rounding besides half-cent be used?

5) By what formulae have rates and rate tables been developed? Would it be possible to compute certain rates instead of prepunching all of them?

6) To how many places should each result be carried?

Multiple card routines

1) Does each card type have an identifying code punched? If not, can a code be assigned so that there will be a positive method of identification for each card type? Can the same code be in more than one type?

2) What should the sequence of the cards be?

3) Of which card types will there be only one for a group? Which may be multiples?

4) Will there always be at least one card of every type in each group? If not, what will be the effect on the program?

5) Should any of the cards be separated, either by control hole or because of a factor discovered during the routine?

6) A check of the designating information in each group should usually be made. If it is not desired, why not?

7) Can the first card of a group be identified by control hole for the designation check?

8) Are summary cards included in the routine? If not, would they be of any value?

9) Is progressive rounding a desirable feature?

10) Is there any information which should be reproduced from card to card? If so, what control holes can be used?

11) In which cards are punched the fields to be used as elements? Can the information be obtained from more than one card?

C. PRELIMINARY PROGRAM FLOW CHART

When the programmer has completed the above analysis, he should be ready to do a rough program. Before programming, it is well to do a rough flow chart of the problem as it will be solved by the computer. This flow chart differs from the one previously mentioned in that it outlines all the computer steps which will be used. Each step should be shown separately.

If the program is a multiple card routine, the first steps will almost invariably be a check of the designating information. When this has been completed, the cards should branch through selectors to their individual routines. The program for each card should be handled in the sequence in which the cards enter the routine. The preceding step-by-step analysis should indicate the program for each type of card.

D. FINAL PROGRAMMING

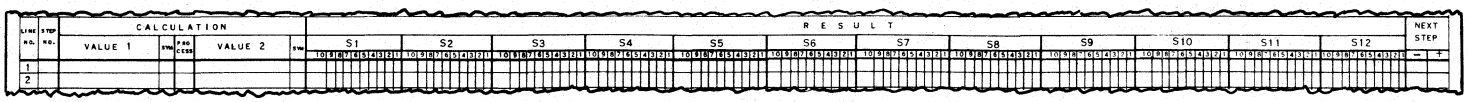
When the rough flow chart has been completed, the programmer is ready to use the program charts. The first step would be to enter the elements and constants at the top of both programming sheets. Then, beginning at

"Start", the program for each card should be entered in sequence. At each place where a selector is used, as much information as possible should be entered on the selector chart. When an interstep function is used, the next step should always be entered in the appropriate box. As each storage is used for the first time, the decimal location should be entered both in the heading at the top of the program step section and opposite the storage in the punching section.

Sometimes a programmer will discover that a step has been omitted from the routine for a card. It is usually easier to add the step at the bottom of the form, being sure that both the step branchings show clearly what has been done. Ideas for short cuts, reuse of steps, etc., may also occur. It is often better to complete the first program regardless of the sequence of the steps on the chart, and to incorporate all such changes in a final program. Although the first program usually should be one that would work if wired exactly according to the chart, the program from which the panels are wired should preferably have incorporated all changes in the proper location.

E. PLANNING CHART

When the program charts have been completed, it is well to use a planning chart with sample figures to test the program. At least one card of every type should be included in the sample; if certain cards may be missing, it is well to follow a group such as this through its routine to see what effect this has. In fact, the more possibilities that are tested, the more successful will be the use of the chart.



Step No.: In this box is written the step number as indicated on the program chart. Only the steps through which a particular card is routed are entered. If the first card goes through only steps 1, 4, 5, 6, and 7, these will be the first five steps on this chart. If an operational function is used, this may be treated as a step, although no values would be entered.

Value 1, Value 2: The actual numeric values of the sample are entered. For an operational function, these are left blank.

Sym.: The symbol (N1-N36 or S1-S12) of the element or storage as indicated on the program chart. No entry is make for an operational function.

Process: This will be +, -, ×, or ÷. It is left blank for an operational function.

Result: It is advisable to draw a line with a colored pencil indicating the decimal location of each storage. The numeric value of the result is

then aligned around the decimal location. When the operational function of clear is used, "0" should be written across all storages which are cleared. For a set instruction, "S" should be written across the storages which are set.

Next step: In this box the next step is written under either + or -, depending on the sign of the result, never under both. If the branching is routed through a selector, the step to which the card is sent is written, rather than the selector number as is done on the program chart.

An example of the use of this chart may be found on page 181.

When the program charts have been completed and checked, the programmer should wire the two panels and check out the boards as outlined under operating instructions.

F. EXAMPLE

On the following pages is an example of an approach to programming. The problem is stated more or less as the information might be given to a programmer. The first rough flow chart follows, with a few notes about the source of information if the entire program is done on the 120. Next are notes on how the program would be handled by the computer. Then come the card forms with pertinent notes, followed by a step-by-step analysis. The final portion is the program flow chart, from which the program itself will be worked out.

The problem is an annual physical inventory. At present tabulating cards are punched for each item at each inventory location. One item may be stored in more than one location, so an item may have multiple item cards. The cards are keypunched with part number, quantity, and descriptive information such as tag number, bin location, etc. The cards are sorted by part number, and price is introduced on the MCRP. On the PMP, quantity is multiplied by price for the extension. The cards are then listed, with totals by part number, and the listing is sent to the Cost Department. The Cost Department checks its ledgers to determine whether the part has been ordered during the year. If it has been ordered, nothing further is done with it. If not, the most recent stock status summary report is checked to determine the quantity used during the preceding year. The stock status summary is a report which is tabulated monthly, containing among other things the total quantity of each item used since the preceding physical inventory. A mental or calculating machine computation is made dividing the total quantity on hand by the usage for the year. The result of this is the approximate number of years supply on hand. For example, with 15 on hand and 3 used this year, the available supply should last for 5 years. If the supply is three years or more, the value of the inventory must be written down. On a tabulating card is written the part number and the dollar amount of the write-down, which is a percentage of the total value of the part according to the following table:

3 years	25%
4 years	50%
5 or more years	75%

These cards are punched and interfiled with the inventory cards. They are then used to tabulate the final inventory reports.

The first rough flow chart might look like the sample on page 218.

As a result of analyzing the rough flow chart, it appears that the entire inventory from the sorting of the inventory cards to the final tabulation can be handled on the 120. To do so would necessitate the following changes in the procedure.

1) Interfile master price cards with the inventory cards so that the inventory cards can be priced by the 120.

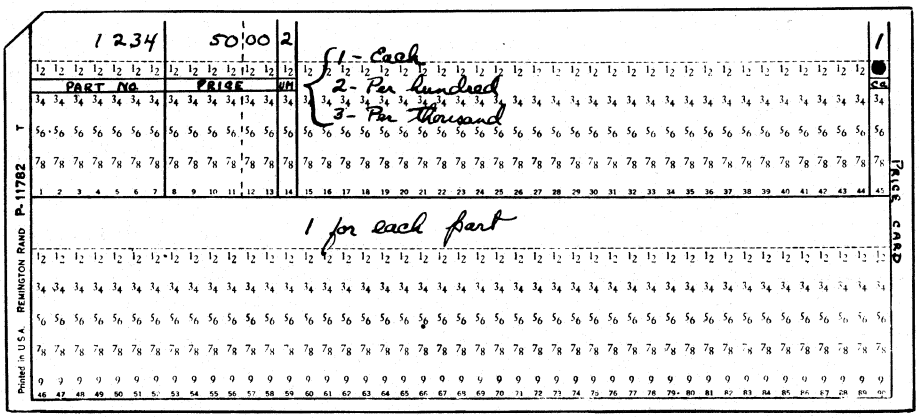
2) Interfile blank summary cards into which total quantity, net amount, and write-down amount will be punched.

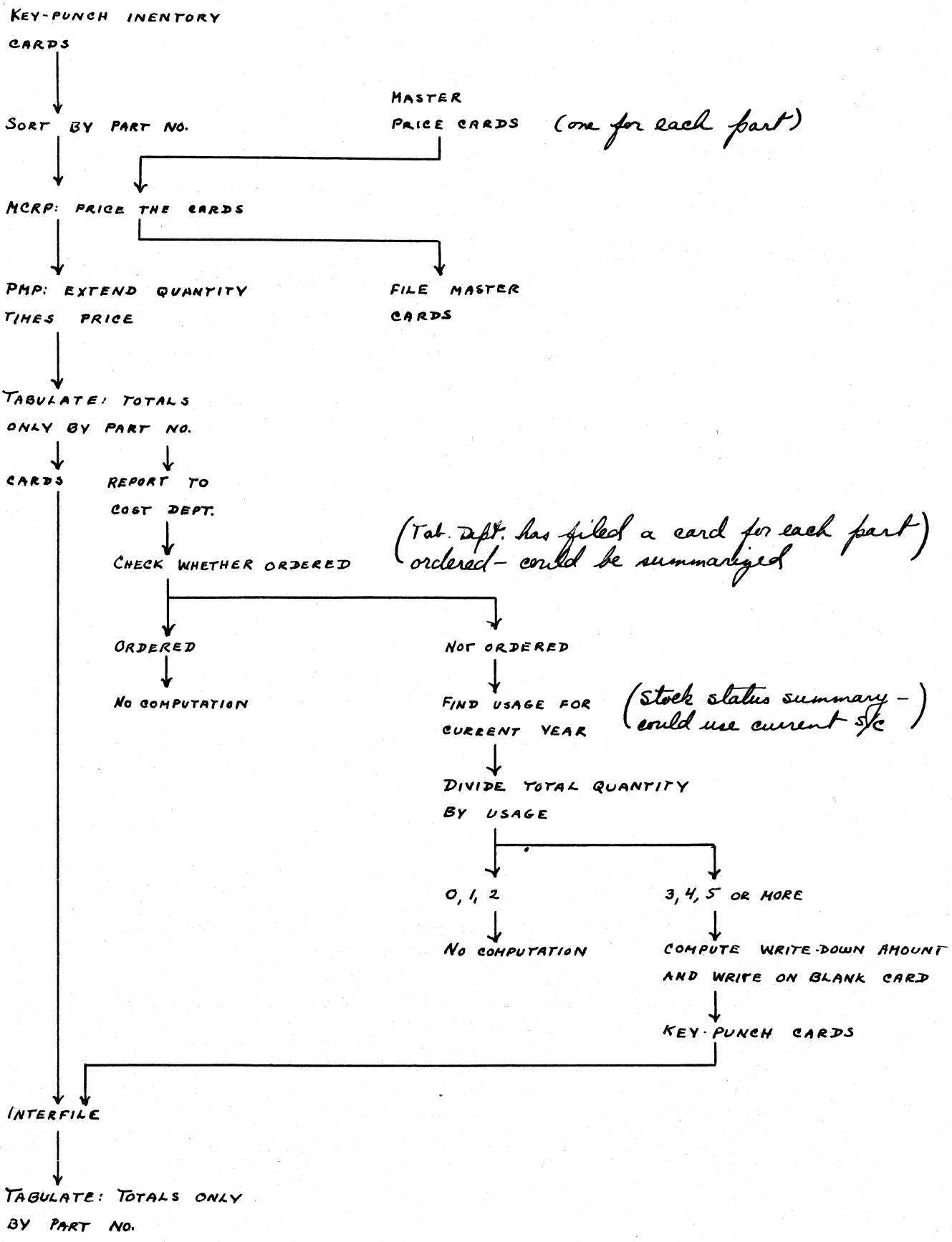
3) Have an indication of whether the part has been ordered during the year. Since an order card for each part has been filed following completion of monthly reports, these could be summarized into one card for each part. All this card would need is part number.

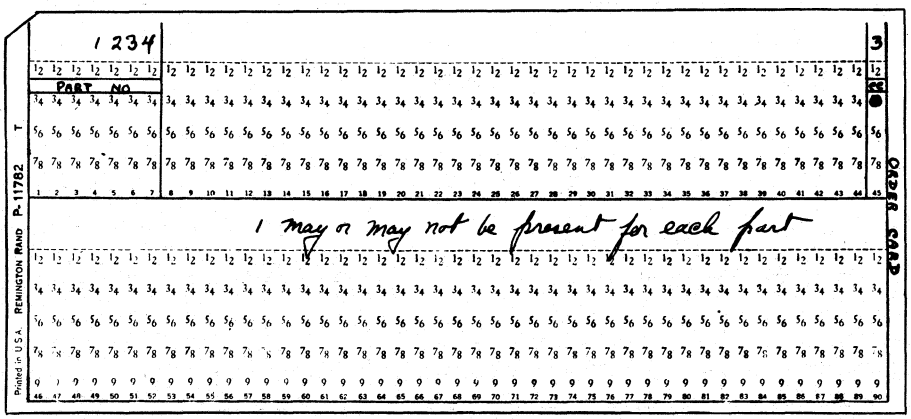
4) Sort in the stock status summary cards for the last period of the year, which contain the total usage for the year.

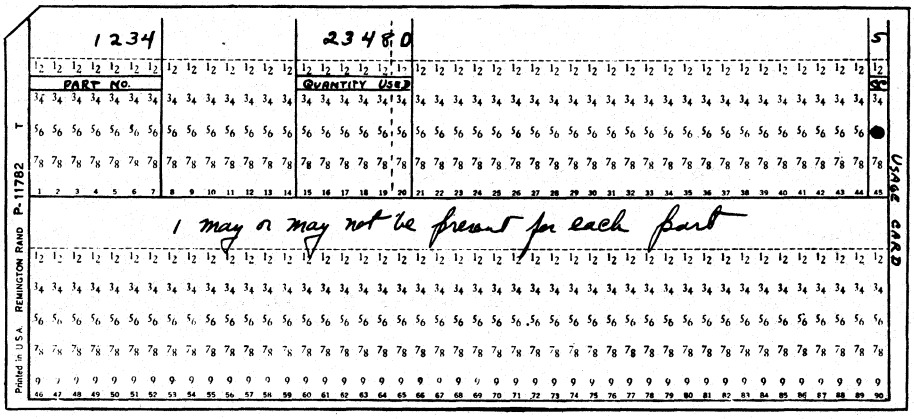
5) Use the summary cards for the final inventory report. (Note: there are other inventory reports on which the detail information is shown.)

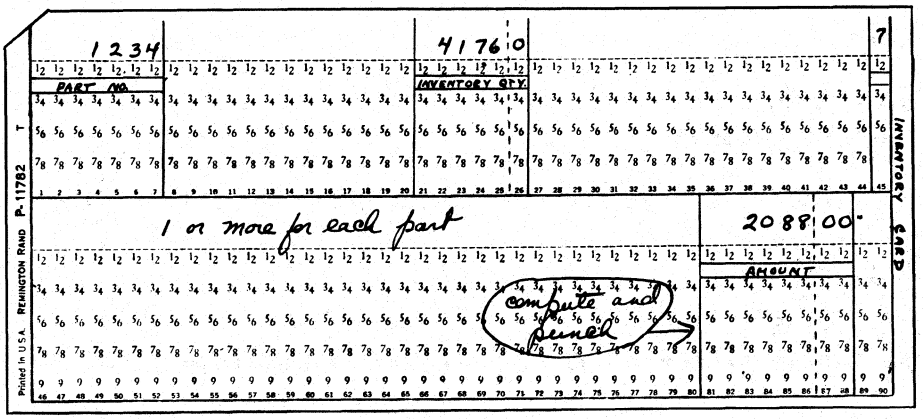
At this point a set of cards should be obtained. The examples shown below contain the portion of each type of card which has a bearing on the computation.

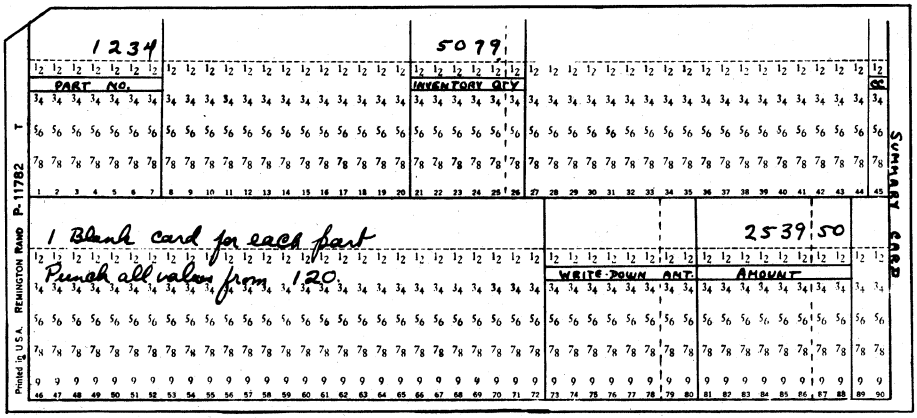



KEY-PUNCH INENTORY CARDS
SORT BY PART NO.
MASTER PRICE CARDS
(one for each part)
MCRP: PRICE THE CARDS
PMP: EXTEND QUANTITY TIMES PRICE
FILE MASTER CARDS
TABULATE: TOTALS ONLY BY PART NO.
CARDS
REPORT TO COST DEPT.
CHECK WHETHER ORDERED
(Tab. Dept. has filed a card for each part ordered - could be summarized
ORDERED
NO COMPUTATION
NOT ORDERED
FIND USAGE FOR CURRENT YEAR
(Stock status summary - could use current s/c)
DIVIDE TOTAL QUANTITY BY USAGE
0, 1, 2
NO COMPUTATION
3, 4, 5 OR MORE
COMPUTE WRITE-DOWN AMOUNT AND WRITE ON BLANK CARD
KEY-PUNCH CARDS
INTERFILE
TABULATE: TOTALS ONLY BY PART NO.


1234
3
PART NO.
1 may or may not be present for each part
ORDER CARD
Printed in U.S.A. REMINGTON RAND P-11782 T


1234
23480
5
PART NO.
QUANTITY USED
1 may or may not be present for each part
USAGE CARD
Printed in U.S.A. REMINGTON RAND P-11782 T


1234
41760
7
PART NO.
INVENTORY QTY.
1 or more for each part
20880
AMOUNT
compute and punch
INVENTORY CARD
Printed in U.S.A. REMINGTON RAND P-11782 T



The step-by-step analysis may be handled in many ways. The following is a narrative method, containing notes on steps to include in the program, as well as arithmetic computations of certain portions of the problem.

1) Price card

Control hole - 1/45
Always one card for each part number, always first of group

Store part number

Store price (adjusting price to "each" in this card would save a step on each inventory card)

2) Order card

Control hole - 3/45
May or may not be present

Store a 1 to remember that it was present? Use selector hold? 0/44 is an available control hole for selector hold.

3) Usage card

Control hole - 5/45
May or may not be present

Store quantity used. If not present, quantity used will be zero

4) Inventory card

Control hole - 7/45
At least one always present. May be singles or multiples

Inventory quantity × price (must be adjusted) = amount. Round to 3/2 decimal. Can never be negative.

Accumulate quantity and amount for summary card.

5) Blank summary card

No control holes
Always one for each part

Test whether the part has been ordered. If so, no further calculation. If not, go to next step.

Divide total inventory by usage to equal the number of years supply. Drop off decimals.

15 ÷ 25 = 0
15 ÷ 15 = 1
15 ÷ 4 = 3
15 ÷ 2 = 7
15 ÷ 0 = 0 This must be changed - test for usage before this step.

Test to determine number of years supply.

Eliminate under 3
2 - 2 = +0 No further calculation
2 - 3 = -1 Continue testing

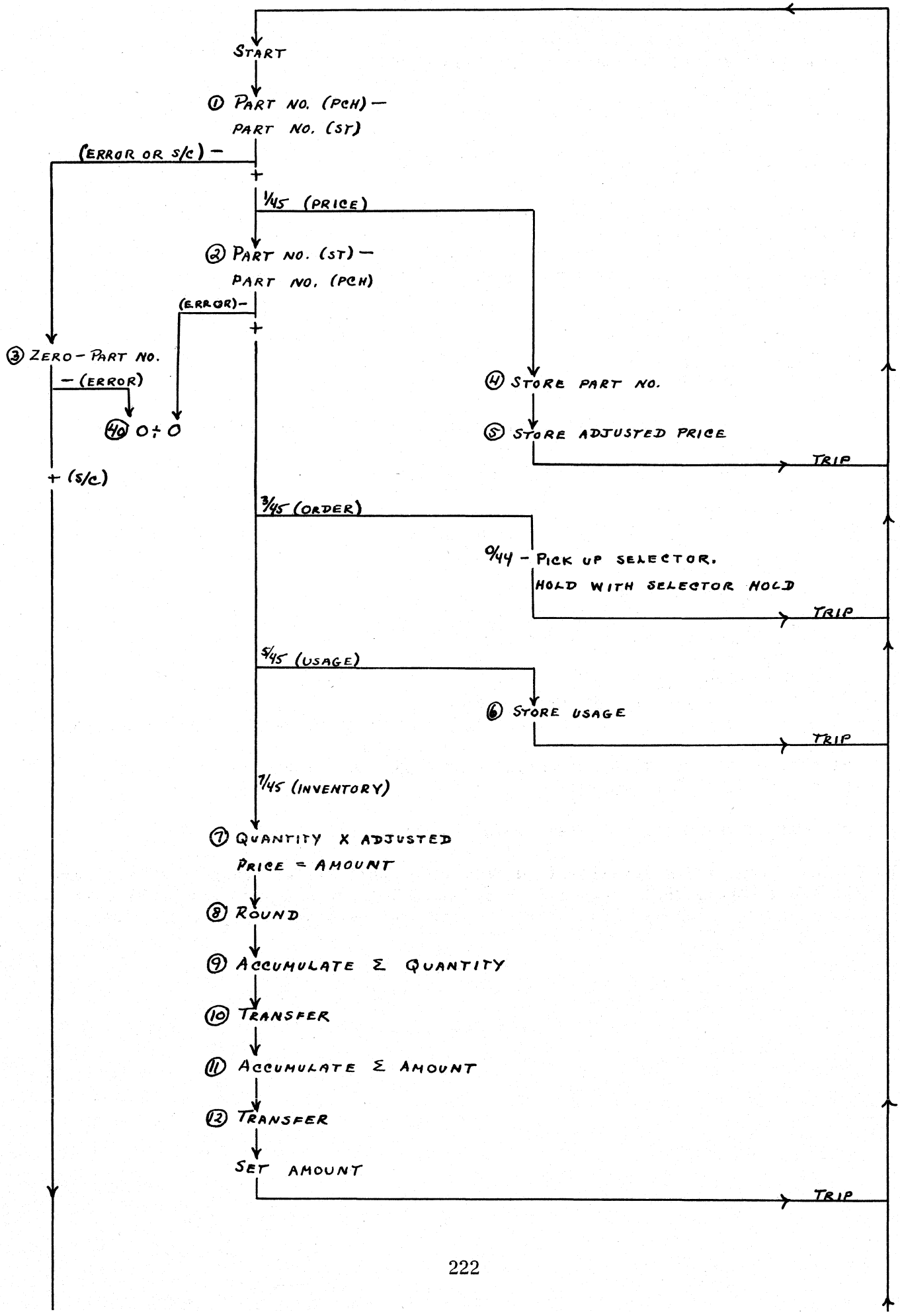
Three years supply
3 - 3 = +0 25% write-down
3 - 4 = -1 Continue testing

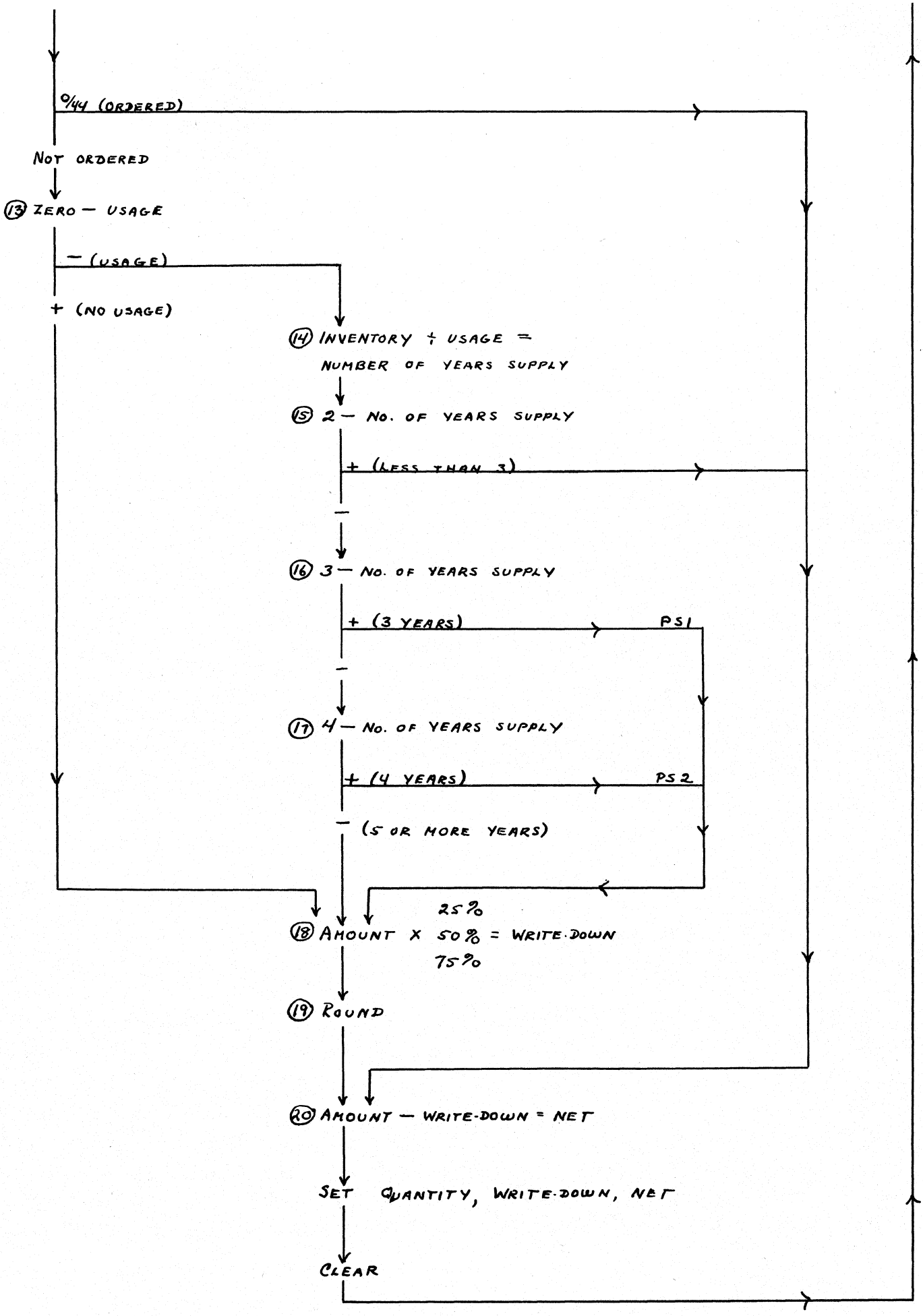
Four years supply
4 - 4 = +0 50% write-down
4 - 5 = -1 75% write-down (direct no usage cards to this step)

Multiply total amount by percentage - use program selects above to save steps - round to 3/2 decimal.

Subtract write-down from total amount to equal net.

At this point it is possible to draw a flow chart of the final program. From the chart, the actual programming is done. The program for this problem is found on pages 272 and 273. The flow chart appears on pages 222 and 223.


START
① PART NO. (PCH) — PART NO. (ST)
(ERROR OR S/C) —
+
1/45 (PRICE)
② PART NO. (ST) — PART NO. (PCH)
(ERROR) —
+
③ ZERO — PART NO. — (ERROR)
40 0 ÷ 0
+ (S/C)
④ STORE PART NO.
⑤ STORE ADJUSTED PRICE
TRIP
3/45 (ORDER)
0/44 — PICK UP SELECTOR. HOLD WITH SELECTOR HOLD
TRIP
5/45 (USAGE)
⑥ STORE USAGE
TRIP
7/45 (INVENTORY)
⑦ QUANTITY X ADJUSTED PRICE = AMOUNT
⑧ ROUND
⑨ ACCUMULATE Σ QUANTITY
⑩ TRANSFER
⑪ ACCUMULATE Σ AMOUNT
⑫ TRANSFER
SET AMOUNT
TRIP


0/44 (ORDERED)
NOT ORDERED
(13) ZERO — USAGE
— (USAGE)
+ (NO USAGE)
(14) INVENTORY ÷ USAGE =
NUMBER OF YEARS SUPPLY
(15) 2 — NO. OF YEARS SUPPLY
+ (LESS THAN 3)
—
(16) 3 — NO. OF YEARS SUPPLY
+ (3 YEARS)
PS1
—
(17) 4 — NO. OF YEARS SUPPLY
+ (4 YEARS)
PS2
— (5 OR MORE YEARS)
25%
(18) AMOUNT X 50% = WRITE-DOWN
75%
(19) ROUND
(20) AMOUNT — WRITE-DOWN = NET
SET QUANTITY, WRITE-DOWN, NET
CLEAR

Section IV Operating Instructions

1. Wiring connection panels

The actual connections to be made have been explained in previous sections. This section deals with suggestions for simplifying the physical wiring of the panels. The suggestions on the sequence of wiring the two panels are not to be considered fixed rules, but they may be of assistance when a programmer first faces the problem of wiring panels. It is important to develop some pattern of wiring so that entire sections of the board will not be omitted.

A. INPUT-OUTPUT PANEL

A-1 Card sensing to accumulator input

Wire each card-read field in turn as it appears on the program chart, using 5 prong wires. Be certain that all corner cuts on the wires are facing the same direction in both sensing and input. Wire the field transfer lines first since the column numbers are covered by 5 prong wires, and it is difficult to locate the column numbers after the 5 prong wires have been placed. If there is a control hole to make an element minus, wire it at the same time that the element is wired. If there is any special wiring, such as alpha or over-capacity wiring, be certain that a complete explanation of the way in which it was wired is attached to the program chart.

A-2 Selector pick-ups

Using the selector chart, wire all control holes to the indicated control transfer lines, wiring a control common to the common of the column when each column is wired.

A-3 Storage outputs to card punching

Wire in turn each storage which is to punch, beginning with S1 and progressing through S12. Use 5 prong wires, being certain that all corner cuts are facing the same direction. If a negative control hole is to be punched, wire it when the corresponding storage is wired.

A-4 Reproducing

Wire the card sensing columns to the card punching columns, using 5 prong wires and adding a single one for the zero positions if necessary. Wire the reproduce or secondary reproduce hubs, and the hold hubs if necessary. Wire whatever controls are indicated on the program chart--reproduce, secondar reproduce, skip, and set hold.

B. CONSTANT-PROGRAM PANEL

B-1 General controls

Wire restart to step or program. Wire N ÷ 0 and 0 ÷ 0 if indicated.

B-2 Element designators - card-read fields

Wire the F-lines to the proper element designators. Wire the control transfer lines which make the elements minus.

B-3 Element designators - constants

Wire each constant value completely, from element designator to accumulator columns, before continuing to the next. When it becomes necessary to use a bus for positions within an accumulator column it is easier to use the bus with the same relative location within the group of buses as the location of the position in the accumulator column section. If any constants are minus, be certain to wire the minus control.

B-4 Decimal location

Wire the decimal location of each element and storage used during the program. For the more frequently used decimal locations, such as 3/2 and 4/3, it is well to jack immediately to the bus beneath the location. This will keep the numbers on the connection panel in view, making it easier to find the correct one.

B-5 Set storage, clear storage

Wire both of these sections.

B-6 Branching and operational functions

Wire the branching of the entire program, both steps and operational functions, in sequence, without attempting to do the equations for the steps.

B-7 Equation

In wiring the equations, it is easier to wire all value 1's, then all value 2's and finally the results. It is also easier if the bottom row of hubs under "Elements" (line 17) is never wired. Placing a wire here hides the element or storage number, making it more difficult to identify the number if used again.

B-8 Process

Wire all processes in sequence.

B-9 Selectors

Check all selector poles with the selector chart to be certain that all hubs have been wired. Wire the pick-up of each selector from its control transfer line or program select.

C. GENERAL

All connection panels should be clearly labeled with the program or programs which are wired on the panel. If more than one program is wired on the same panel and it is necessary to change any wires between programs, it is well to wrap the wires in masking tape so that they may be easily identified.

2. Operation

A. INSTALLATION OF CONNECTION PANELS

The two connection panels are inserted behind a panel on the left front of the computing unit. The covering panel is raised by lifting the bar at the base of the panel. Both connection panels are placed with the printing facing the center of the computing unit.

The constant-program panel is placed in the upper position; the input-output panel is placed below it. The input-output panel is locked in place by a bar which is located in the computer just beneath the panel. This bar must be raised to change the input-output panel, and while it is up the constant-program panel cannot be changed.

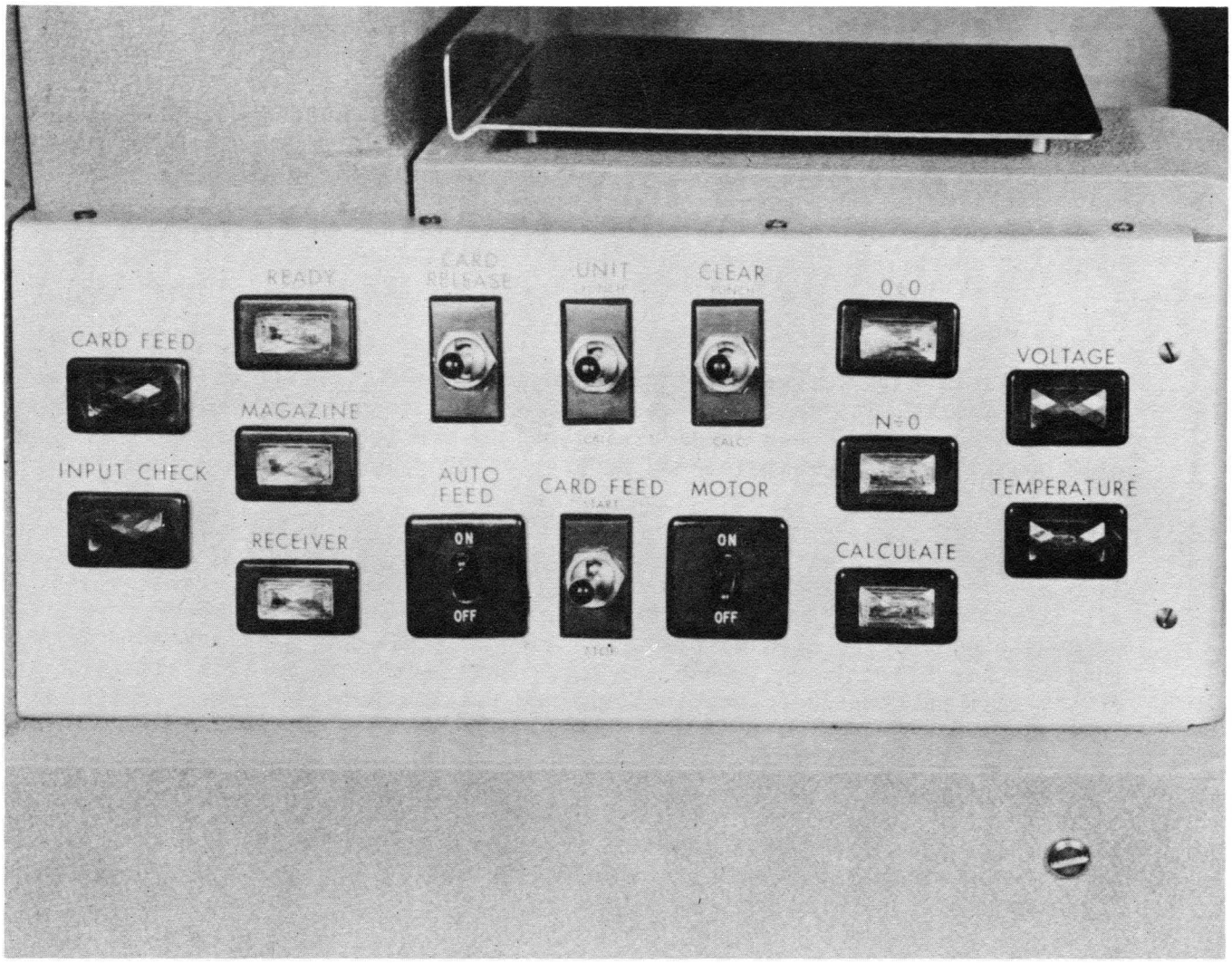
To remove a connection panel lift it slightly by the handles, then pull outward. To insert a connection panel place the studs which are in the upper corners of the panel in the hooks which are in the upper corners of the housing. Press inward and downward until it is in place.

If the connection panel cover is lifted while the computer is on, the red voltage light in the sensing-punching unit will light. It will remain lit until the cover has been closed and the clear switch is moved to the punch position.

When a change is being made to a panel, the panel should be removed from the computer.

B. CONTROL PANEL

Located on the control panel are the lights indicating to the operator the condition of the computer, as well as the switches used to obtain the desired operation. Each of these is discussed briefly below.



B-1 Card feed

When a misfeed occurs in the sensing and punching sections or between the punching section and the card receivers, the computer will stop with this light lit. The program which is being calculated when the misfeed occurs will be completed. The misfed card or cards must be removed from the computer before continuing the card feeding. When this has been done, the the program may be restarted by lifting the card feed switch.

B-2 Input check

If an element containing punching which is not a numeric code is called upon as part of a program, the computer will stop with the input check light lit. Since the program cannot continue past the point at which the element was called upon, it is usual to eject the card by simultaneously raising the unit switch and the card release switch. The computer will cycle once and stop. The card which has just been ejected may be removed and the program may be continued by raising the card feed switch to "start."

B-3 Ready

The ready light will go on when all three "on" switches have been turned on and the computer is being supplied with proper operating voltage. Note that although the ready light indicates that the proper voltage is being supplied, the voltage light will remain on until the clear switch is turned to "punch". A warm-up period of almost two minutes is required before the computer is ready for operation.

B-4 Magazine

When the last card leaves the feeding magazine, the computer will stop with this light lit. The last card, however, will be computed and ejected into a receiving magazine. When cards are placed in the feeding magazine, the light will go out, and the computer may be started by raising the card feed switch.

B-5 Receiver

When either a receiving magazine or the chip pan becomes full, the computer will stop. The program for the card which was in the sensing chamber will be completed and the card ejected. No automatic feeding will take place as long as this light is on. When the cards have been removed from the receiver or the chip pan emptied, the light will go out and automatic feeding may be resumed.

B-6 Card release

The card release lever is a means of ejecting a card from the punching section without punching, despite the fact that information has been set in the punching dies. All punching dies will be cleared.

The unit lever must be raised simultaneously with the card release lever. The card in the punching section is always ejected into the rear receiving magazine, even though a sort signal may have been given to the card during its program. As this card moves into the receiving magazine, the following card moves from the sensing section to the punching section and is computed. Unless the card lifting lever is depressed, a card will be fed from the feeding magazine to the sensing section. However, automatic feeding does not take over.

B-7 Unit

The unit switch is a safety switch which will move either up or down. It is raised with the card release switch to release the card. It is also raised

with the clear switch to clear the punching dies. It is lowered in conjunction with the clear switch to clear all storages and return all selectors to non-select.

B-8 Clear

The clear switch is used to clear either the punching dies or the storages and to drop out selectors. It will also clear the voltage light if proper voltage is being received.

When moved upward in conjunction with the unit switch it will clear the punching dies of all information which has been set up, either reproduced or from storage. The clearing occurs on the next feeding cycle. If the computer has been stopped during the program for a card, the card should be ejected by use of the card release switch. Since this switch ejects a card without punching, no punching will occur in this card even though values are in the punching dies.

When moved downward in conjunction with the unit switch, the clear switch will immediately clear all storages and restore all selectors to non-select.

The clear switch is moved upward without the unit switch to clear the voltage light. The voltage light will go out, provided correct operating voltage is being received.

B-9 Automatic feed

The automatic feed has two positions, "on" and "off". When set at "on", continuous card feeding will occur upon the lifting of the card feed switch. If it is set at "off", the card feed switch must be lifted for each cycle.

B-10 Card feed

The card feed switch is lifted to start the computer, and lowered to stop it. Whether the computer will feed continuously or single cycle when the card feed switch is raised is dependent upon the setting of the automatic feed switch. To stop the computer during a run, the switch is moved to the down position. The computer will stop when the program which is being followed has reached "trip." The card being computed will not be ejected, nor will additional cards be fed from the feeding magazine.

B-11 Motor

The motor switch when on allows current to flow to the sensing-punching unit. The power switch must also be on to obtain the current. This switch need not be turned on during the warm-up period of the computer, but must be turned on before the ready switch will light.

B-12 $0 \div 0$

The $0 \div 0$ light will light when the computer detects a $0 \div 0$. If the computer is wired to stop on this condition, the light will remain lit. The card will be in the punching section when the computer stops, and may be ejected into the rear receiver by use of the card release switch. The card release switch

will not cause the resumption of automatic feeding. After the card has been removed from the receiving magazine, the card feed switch may be lifted to resume the feeding.

B-13 N ÷ 0

The N ÷ 0 light will light when the computer detects a number divided by zero. If the computer is wired to stop on this condition, the light will remain lit. The card will be in the punching section when the computer stops, and may be ejected into the rear receiver by use of the card release switch. The card release switch will not cause the resumption of automatic feeding. After the card has been removed from the receiving magazine, the card feed switch may be lifted to resume the feeding.

B-14 Calculate

The calculate light will be on during calculation, but go out during card feeding. If a program step fails to prove, this light will remain lit when the computer stops. It is a convenient check on a long or iterative program to prove that the computer is still operating.

B-15 Voltage

The voltage light will light at the start of a run, when the power switch, motor switch, and the switch on the computing unit have been turned on. It will also light at any time during a run when the proper voltage is not being supplied, and card feeding will stop. Unless the panel covering the connection panels is securely closed, the voltage light will stay on. Automatic feeding cannot be obtained when the light is on.

To turn off the voltage light, lift the clear switch without the unit switch. If proper voltage is being supplied, the light will go out and the card feed switch may be used to start feeding. If proper voltage is not being received, the light will remain lit.

B-16 Temperature

If the temperature of the computing unit becomes too high, the computer will stop with this light lit. The computing voltage will automatically shut off and the storages will clear. When the temperature is sufficiently cool the light will go out, the computing voltage will turn on automatically, and the computer may be started by raising the card feed switch.

C. STARTING THE COMPUTER

A brief summary of the steps required to start the computer is:

1. Insert the two connection panels.
2. Turn on the power switch, computer on switch, and motor switch.
3. Wait until the ready light lights.

4. Lift the clear switch to clear the voltage light.
5. Clear the computer.
6. Insert cards in the feeding magazine.
7. Check the lights to see that all except "ready" are out.
8. Lift the card feed switch.

Three switches are required to start the computer: the main power switch located beneath the control panel on the sensing-punching unit, the computer "on" switch located beneath the test control panel on the computing unit, and the motor "on" switch located on the control panel on the sensing punching unit. Although the motor switch may be turned on at the same time as the other two, it is not necessary to turn it on until the end of the warm-up period.

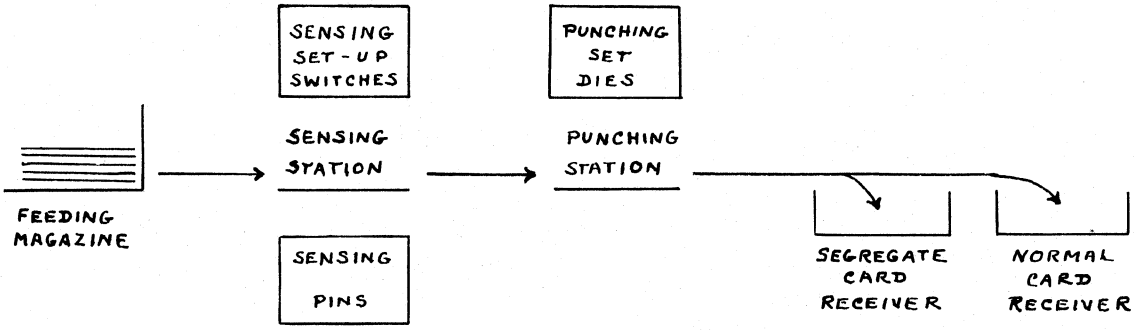
When the power switch and computer "on" switch have been turned on, a warm-up period of approximately two minutes is required before the computer is ready to operate. During this period the voltage light will be on. If the motor switch has not already been turned on, this should be done at the end of the two minute warm-up period. As soon as the computer is ready to operate after the motor switch has been turned on, the white "ready" light will light. The clear switch should then be moved to the punch position to turn off the voltage light. As long as the voltage light is on, the computer will not feed automatically.

Before starting a run, the computer should always be cleared. The punching section is cleared of all information, including reproduce information, by raising the clear switch to the punch position at the same time that the unit switch is raised. When this is done a buzzing noise may occasionally occur, which may be stopped by lifting the unit switch and the card release switch simultaneously. Since the card release switch will feed a card, if any cards are in the feeding magazine the card lifting lever should be used to prevent the feeding. To clear the storages and return all selectors to non-select, the clear switch is moved down to the calculate position at the same time that the unit switch is moved down.

When cards are placed in the feeding magazine, the white "magazine" light will go out. The operator should check to see that all lights on the control panel are out except the ready light. Automatic feeding will not occur if the red voltage light is on. The automatic feed should also be checked, since automatic feeding will not occur unless the switch is on. If automatic feeding does not occur even though the lights are correct and automatic feed is set on, the operator should check the program test panel to see the setting of the step-read switch. Unless it is in the center position, automatic feeding will not occur.

To start a program, the card feed switch is raised.

D. CARD CYCLES



On the first card cycle a card is fed from the feeding magazine to the sensing section. However, no sensing or calculation occurs.

On the second card cycle the entire card is sensed simultaneously, and the set up is locked in the sensing switches. A start signal is immediately given to begin the calculation. At the same time the card moves from the sensing section to the punching section, and the following card moves to the sensing section.

Calculation of the first card should be completed by the time it reaches the punching section. If not, however, the computer will wait until calculation has been completed before beginning the next cycle.

At the beginning of the third cycle, the punching section moves downward. This downward movement clears the set-up which was locked in the sensing switches. The first card is punched with the values which were set in the punching dies while calculation was taking place. The punching dies move upward, and are cleared of the values which were just punched. At the same time the second card is sensed and its set-up locked in the sensing switches. Tne conclusion of this cycle is the ejection of the first card into one of the two receiving magazines, while the following card is being calculated and moving into the punching section.

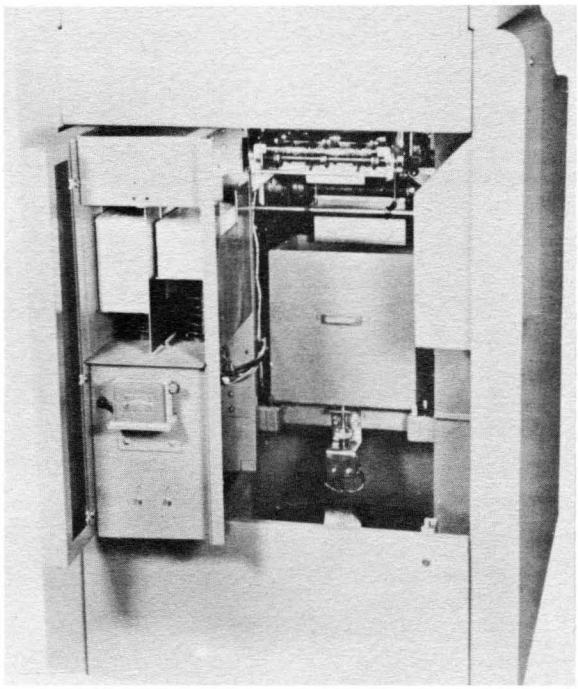
When the computer stops because the last card has left the feeding magazine, this last card will be completely calculated and ejected. A "no card sensing switch", located in the sensing section, prevents sensing and calculation if a card is not in the sensing section when sensing normally occurs. Therefore, during the cycles in which the last card is feeding out, the computer does not go through the program using zeros as values.

E. GENERAL INSTRUCTIONS

Instructions should be provided for the operator as to the correct procedure to follow in regard to the various lights. The card handling procedure for certain lights such as voltage, temperature, card feed, and input check may well be the same for all programs. However, O ≠ O and N ≠ O are often intentional stops, the reason for which will vary from program to program, and for which different procedures are required. It is important that instructions be available for each case, either attached to the program charts or in a separate list for the operator.

Note that the chip pan on the sensing-punching unit must be emptied occasionally. If it becomes full, the magazine light on the control panel will light, and automatic feeding will not occur until the pan has been emptied.

The chip pan may be removed by opening the hinged panel on the lower rear of the sensing-punching unit.



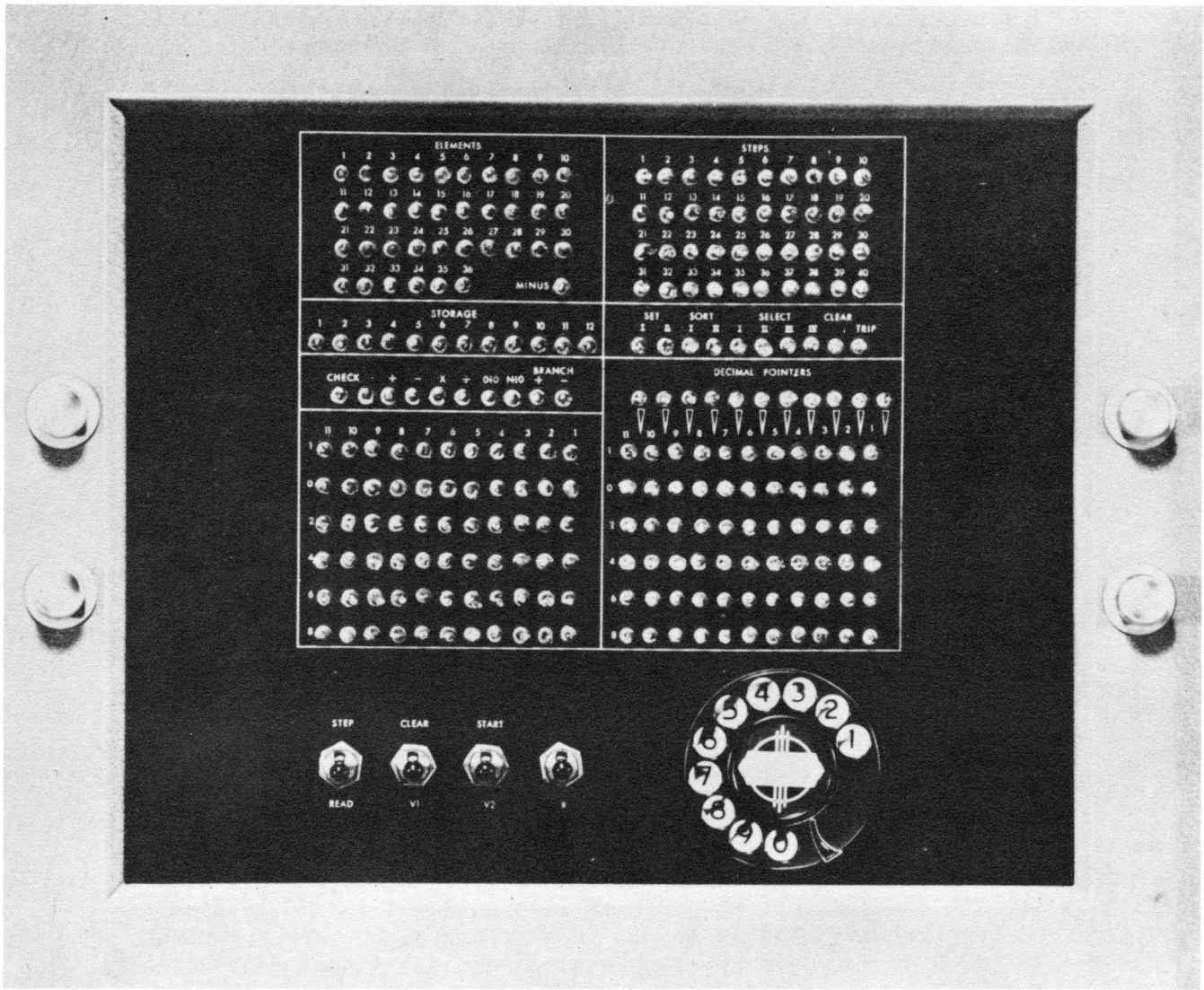
It is important to test a program thoroughly before using it on regular work, both to make certain that the problem is programmed correctly and that it is wired correctly. To do so, it is very useful to have a finished planning chart with as complete an example as possible. A test deck using the same values should be punched. The test deck can then be run through the computer and balanced to the predetermined results on the planning chart.

If the results are not the same, or the computer hangs up on a card, it becomes necessary to use the program test panel, which is located on the left end of the computing unit. By means of this panel, each program step and operational function can be checked.

3. Testing a program

A. PROGRAM TEST PANEL

A-1 Description



On the panel are located neon indicator lights which are used to show the operation of the computer. The switches beneath the panel and the step-advance dial are used to advance through the program. The panel itself will swing either to the front or to the rear of the computing unit. It is swung

to the front for ease in testing a program. This is done by pushing together the two knobs on the left side of the panel and pulling out. The knobs on the right side are used to swing the panel to the rear.

A-2 Control switches

1) Step-read switch

Except when the panel is being used for testing, this switch should remain in the center position. Automatic feeding of the computer cannot be obtained if it is in the up position.

This switch must be moved up to "step" each time before the step advance dial is used. If this is not done, the "check" indicator light will go out, and the computer will hang up. If this happens, the test must be begun again from "start". Note that particularly in a multiple-card routine, this may cause a distortion of some values, such as accumulating fields. In this case, it is necessary to trip out the card in the computer, clear, and start from the first card.

This switch is moved down to the "read" position during the time that the test panel is being read.

2) Clear-V1 switch

This switch automatically returns to the center position when it is not being held up or down.

When this switch is moved up to "clear" it clears all storages, S1-S12, regardless of whether they are wired to clear.

This switch is moved down to V1 and held down while reading value 1 of the program step.

3) Start-V2 switch

This switch automatically returns to the center position when it is not being held up or down.

The switch must be moved up to "start" to begin the program, before the step advance dial can be used. It will bring the program through the first step. It may be lifted at any time during a program to return to the first step. However, note that the repetition of steps may in some programs cause a distortion of certain values.

The switch is moved down to V2 and held down while reading value 2 of the program step.

4) R switch

This switch automatically returns to the center position when it is not being held down.

The switch is moved down to R and held down while reading the result of the program step.

A-3 Step advance dial

This dial, containing the numbers 0-9 is used to advance through the program steps. The number "0" is equivalent to 10. The numbers do not refer to program step numbers, but rather to the number of steps and operational functions through which the program is to advance.

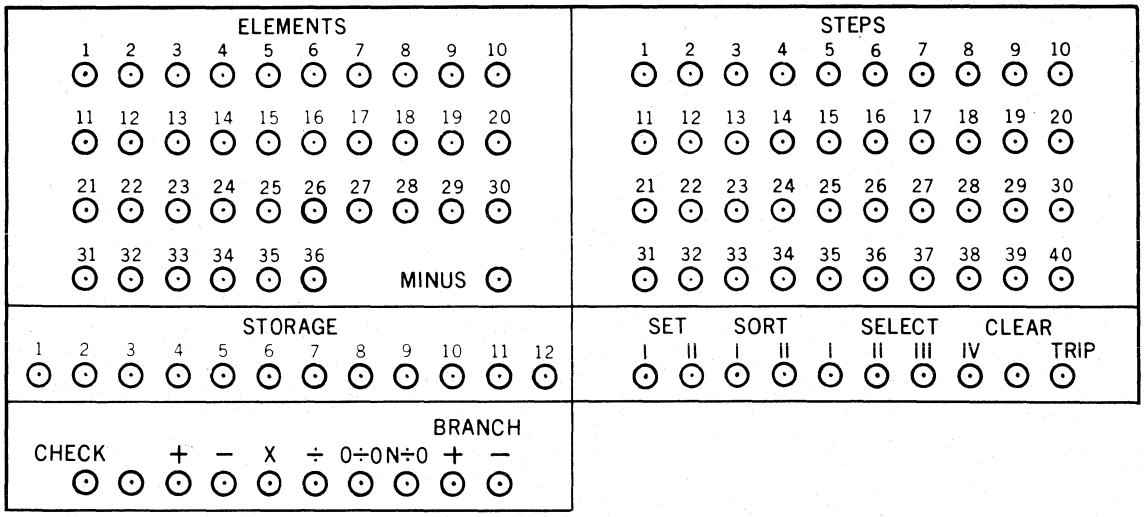
If the operator desires to advance one step at a time, "1" is always dialed following the checking of the current step and the lifting of the step-read switch.

To progress through more than one step at a time, dial the difference between the step number at which the computer is stopped and the desired step. For example, to reach the fifth step of a program from start, dial 4. The lifting of the start switch brings the program to step 1, and the difference between 1 and 5 is 4. Note that if the particular card does not go through a step, step 2 for example, the number to dial would be 3. If the number to be dialed is over 10, it is necessary to dial more than once. Zero would be dialed first to progress through the first 10 steps, followed by whatever number is necessary to make up the remainder of the difference.

All steps through which the computer passes are calculated.

A-4 Indicator lights

The following is a diagram of the indicator lights, which will be explained in turn in the following sections.



A-5 Steps

There is one light for each of the 40 program steps of the computer. As a step is being performed, the light corresponding to the step number is lit. If the computer stops during a program, the light corresponding to the step number on which it has stopped is lit.

A-6 Operational functions

There is one light for each of the operational functions:

Set 1 and set 2
Sort 1 and sort 2
Program selects 1, 2, 3, and 4
Clear
Trip

As the computer performs one of these functions, the corresponding light is lit. If the computer stops during an operational function, the light remains lit. All lights except the program select lights go out as the program progresses to the next step or operational function. Program select lights remain lit until trip time, unless the program select is dropped out during the program.

A-7 Elements

There is one light for each of the 36 elements. These lights will light as the corresponding element is called on as V1 or V2 of a program step. When the computer stops none of them will be lit, since it stops at the end of a step.

When a program step is being read, the clear-V1 switch and start-V2 switch are moved down in turn to the V1 and V2 positions. If an element is being used as V1 or V2, the corresponding light will light. The minus light, in the lower right corner of the section, will light if the element is minus; otherwise the element is plus.

A-8 Storage

There is one light for each of the twelve storage units. These lights will light as the corresponding storage is called on as V1, V2 or the result of a program step. When the computer stops none of them will be lit, since it stops at the end of a step.

When a program step is being read, the clear-V1 switch, start-V2 switch, and R switch are moved down in turn to the V1, V2, and R positions. If a storage is being used as V1 or V2, the corresponding light will light. The corresponding light will also light for the storage which is being used as the result. If the storage is minus when called upon, the minus light in the lower right corner of the element section will light; otherwise the storage is plus.

A-9 Process

1) Check

This light will light following each program step provided the step proves. If the computer stops at the end of a step which has proved, the light remains lit. Failure to light indicates that the step did not prove.

Even though a step has proved, the check light will go out if the step advance dial is moved without the previous lifting of the step-read switch. If this happens it is necessary to restart the program.

2) +, -, ×, ÷

There is one light for each of the four processes. The light indicating which of the processes is being used on a step will remain lit if the computer stops at the conclusion of a program step.

3) 0 ÷ 0, N ÷ 0

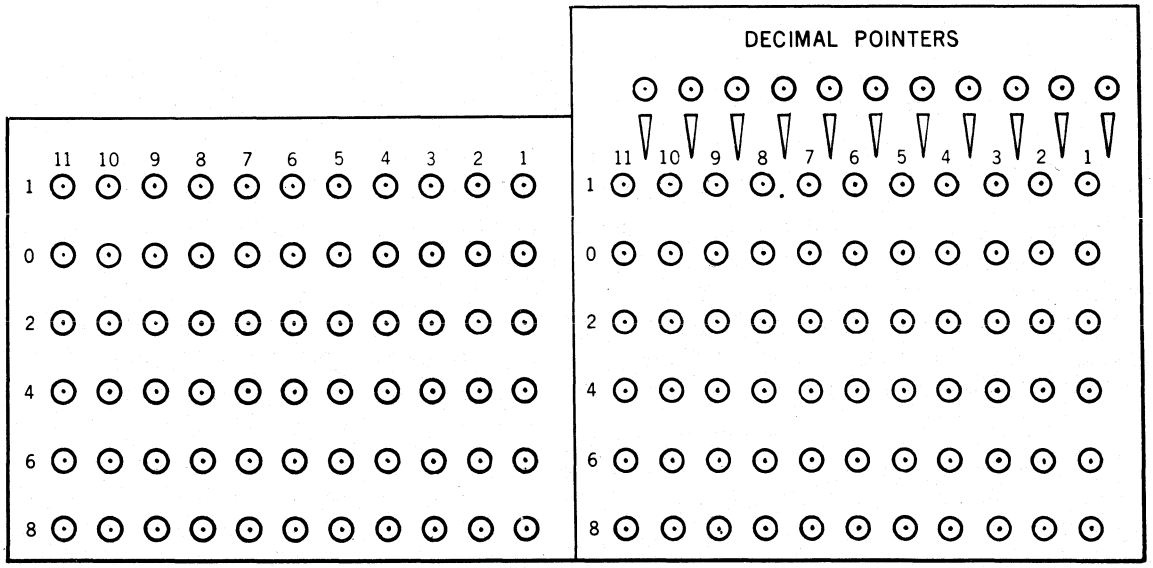
If the computer attempts to divide zero by zero or a number by zero, the corresponding light will light. It will remain lit if the computer stops at the end of a program step.

4) Branch +, -

Either the plus or minus branching light will light at the end of each program step, indicating whether the result is positive or negative. It lights regardless of whether the corresponding branching is wired on the connection panel.

A-10 Accumulator

In this section are located columns representing the 22 columns of the accumulator. The 11 columns on the right represent the A section, while the 11 columns on the left represent the M section.



There are six positions in each column. If the value is positive, the even digits (zero is considered even for this purpose) 0, 2, 4, 6 and 8 are formed by the lighting of the single corresponding light. The odd digits 1, 3, 5, 7, and 9 are formed by the lighting of the preceding even digit plus the 1 light.

Value	Lights
0	0
1	0,1
2	2
3	1,2
4	4
5	1,4
6	6
7	1,6
8	8
9	1,8

If the value is negative, the digits are indicated as tens complements.

Value	Lights
0	0
1	1,8
2	8
3	1,6
4	6
5	1,4
6	4
7	1,2
8	2
9	0,1

The accumulator light will light as V1, V2, and R enter the accumulator. They may be read when the clear-V1 switch is lowered to read V1, when the start-V2 switch is lowered to read value 2, and when the R switch is lowered to read the result.

A-11 Decimal Pointers

There is a light for each decimal location, 1/0 to 11/10. The light corresponding to the decimal location of each element or storage will light when the accumulator lights light for value 1, value 2, and the result.

B. TESTING A PROGRAM

To test a program, move the step-read switch on the program test panel up to the step position before feeding any cards. This will cause the computer to

single cycle instead of feed automatically. To place the first card in a position so that its program can be followed, lift the card feed switch on the sensing-punching unit once. The computer will cycle twice. On the first cycle the card will move to the sensing section. On the second cycle it is sensed, the sensed values are locked in the sensing section, and the card moves to the punching section. The program test panel may now be used for the first card. If the program for the first card has already been checked, lifting the card feed switch on the sensing-punching unit once more will bring the second card into a reading position.

The first step for the reading of any card is the lifting of the start-V2 switch on the program test panel. This will bring the computer to the first step of the program, and must always be done whether or not the first step is to be read. It is possible to lift the start switch at any time during a program to return the first step, though if the program calls for accumulating, the accumulated total would be inflated each time unless the clear switch is raised. The routine for the reading of each step is:

1) Check the step light to verify the step number.

2) Move the step-read switch to the read position.

3) Move the clear-V1 switch to V1, and hold it down.

4) Check V1. This includes the following:

Element or storage number
Minus light
Decimal location
Value as entered into the accumulator

5) Release the V1 switch.

6) Check the process.

7) Move the start-V2 switch to V2 and check V2 in the same way as V1. Release the V2 switch.

8) Move the R switch to R and check R in the same way as V1 and V2. Release the R switch.

9) Check whether the branching is + or -.

10) Be certain the check light is on.

11) Raise the step-read switch to step.

12) Using the step-advance dial, dial the next step. Since the number which is dialed determines how many steps the program will advance, for a complete testing always dial 1.

As mentioned above, the use of a work sheet will simplify the determination of what the result of each step should be. If a light such as decimal location,

element or storage is not lit, it usually means that the wire was omitted when wiring the connection panel. If an element appears to be incorrect when read into storage, always check the punching of the card.

When a program has been followed through completely and the trip signal is given, the card will be punched and tripped out. At the same time the next card will be fed from the sensing section to the punching section. When the start switch has been raised, the program for this card may be followed.

Section V
Course Outline and Sample Problems

1. Course outline

The following outline is a suggested method of studying the 60 and 120, involving both reading of the text and the programming of sample problems. The problems, although based to a certain extent on actual programs, are designed mainly to give practice in various programming techniques.

The first problem is to be worked in sections during the study of the fundamental computer components. The following problems are to be programmed completely at specified points in the course.

In programming a problem, the first step is to read the statement of the problem thoroughly. Next a flow chart should be drawn, outlining the steps of the problem as completely as possible. From this flow chart the actual programming may be done.

Actual solutions to the problems are included at the end of each problem. It should be noted that there may be many correct solutions to one problem. In the "Program Comment" section following each program, some of the other possibilities are discussed, as well as some of the reasons for doing the program as shown.

The following is a suggested method of studying the 120.

1) Read: Introduction, pages 4-6
Accumulator, pages 14-20
Elements, pages 20-37

2) On program charts enter the table of factors, element and accumulator input sections of problem 1.

3) Read: Storage, pages 37-43

4) On the program charts enter the card field titles, output punching, and decimal locations of storage in problem 1.

5) Read: Program step, pages 43-50
Half-cent rounding of multiplication, pages 104-105

6) On the program charts enter each step required for problem 1. Leave the plus branching of the last step blank.

7) Read: Operational functions, pages 50-61

8) On the program charts enter the operational functions of problem 1. Study the solution of problem 1 and the program comments.

9) Read: Transfer into storage, page 103
Accumulation, pages 103-104
Range testing (except section C), pages 111-113

10) Program problem 2. When completed, study the solution and program comments.

11) Read: Selectors (to program selects), pages 62-66
Testing identifying information (A-C), pages 117-122

12) Program problem 3. When completed, study the solution and program comments.

13) Read: Selectors, pages 66-86

14) Program problem 4. When completed, study the solution and program comments.

15) Read: Approaching a program, pages 211-223
Rounding, pages 106-111
Determining amounts at various ranges, pages 113-116
Use of an element as the result of a step, pages 116-117
Testing designating information (D-G), pages 122-125
Placing more than one value in a storage, pages 125-129
Wiring alpha into the accumulator, pages 132-134

16) Program problem 5. When completed, study the solution and program comments.

17) Read: Reproduce, pages 87-100
Crossfooting, pages 130-132

18) Program problem 6. When completed, study the solution and program comments.

19) Read: Wiring constants through input, pages 134-135
Wiring a card column as an element and to pick up selectors, pages 135-137
Card position selection, pages 137-138
Making an element zero or a significant value, pages 138-140
Conversion of codes to constants, pages 140-142
Overlapping card fields, pages 142-146
Wiring more than one program on a panel, pages 146-149
Verification, pages 149-151
Fractional twelfths, pages 182-187
Summary of programming techniques, pages 210-211

20) Program problem 7. When completed, study the solution and program comments.

21) Advanced programming techniques.
Card code checking, pages 151-165
Accumulating positive and negative values in one storage, pages 166-175
Square root, pages 176-177
Obtaining a product of more than ten digits, pages 178-182
Program select loop, pages 187-196
Punching zeros from storage, pages 196-210

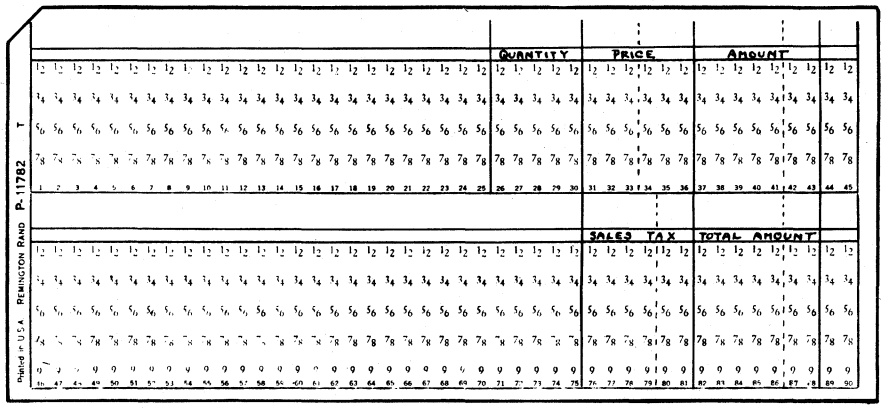
2. Sample problems

A. PROBLEM 1 - INVOICING

A-1 Statement of problem: Item cards for writing invoices have been pulled from a tub file. The prepunched information includes quantity and price. After pertinent information such as customer number has been punched in the cards, they are sent to the computer for extension. Only item cards are put through the computer in this run. In each card the computer is to:

1) Multiply quantity times price for the amount. Round the amount.
2) Multiply the rounded amount by 3% for the sales tax. Round the sales tax.
3) Add the amount and the sales tax to equal the total amount.

A-2 Card form.



A-3 Elements:

Element Number	Card Columns	Description
N1	26-30	Quantity
N2	31-36	Price
N34		.03
N35		.005

A-4 Storages:

Number	Description	Decimal
S1	Working	4/3
S2	Amount	3/2
S3	Sales Tax	3/2
S4	Total Amount	3/2

A-5 Punch:

Columns	Description
37-43	Amount
76-81	Sales Tax
82-88	Total Amount

PROGRAM CHART NO. 1.

APPLICATION: INVOICING

UNIVAC 60 & 120

PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand

TABULATING MACHINES

TABLE OF FACTORS

SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE
N1	+		QUANTITY (COLS. 26-30)	N13				N25			
N2	+		PRICE (COLS. 31-36)	N14				N26			
N3				N15				N27			
N4				N16				N28			
N5				N17				N29			
N6				N18				N30			
N7				N19				N31			
N8				N20				N32			
N9				N21				N33			
N10				N22				N34	+		.03
N11				N23				N35	+		.005
N12				N24				N36			

CARD DESCRIPTION

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2 3/2	S3 3/2	S4 3/2	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP —	NEXT STEP +
1		QUANTITY	+	N1	X	PRICE	+	N2	AMOUNT (NR)	+	S1	1												-	2
2		AMOUNT	+	S1	+	.005	+	N35	AMOUNT (RD)	+	S2		(2)											-	3
3		.03	+	N34	X	AMOUNT	+	S2	TAX (NR)	+	S1	3												-	4
4		TAX (NR)	+	S1	+	.005	+	N35	SALES TAX (RD)	+	S3			(4)										-	5
5		AMOUNT	+	S2	+	SALES TAX	+	S3	TOTAL AMOUNT	+	S4				(5)									-	SET 1
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																									
32																									
33																									
34																									
35																									
36																									
37																									
38																									
39																									
40																									

SYM	CARD FIELD TITLE	NEG. COL	NEG. POS	OUTPUT PUNCHING 10	9	8	7	6	5	4	3	2	1	DEC LOC	SET 1	SET 2	CLEAR	SYM
S1														4/3				S1
S2	AMOUNT						37	38	39	40	41	42	43	3/2	X			S2
S3	SALES TAX							76	77	78	79	80	81	3/2	X			S3
S4	TOTAL AMOUNT						82	83	84	85	86	87	88	3/2	X			S4
S5																		S5
S6																		S6
S7																		S7
S8																		S8
S9																		S9
S10																		S10
S11																		S11
S12																		S12

FROM	TO
START	ST. 1
SET 1	TRIP
SET 2	
SORT 1	
SORT 2	
CLEAR	
P.S. I	
P.S. II	
P.S. III	
P.S. IV	

REPRODUCE CONTROLS	COL.	POS.
REP		
SKIP		
SET HOLD		
SEC REP		

	0 ÷ 0	N ÷ 0
STOP		
SORT		

RESTART	
STEP	X
PROGRAM	

S1-1369 RRBH 12456

SELECTOR CHART NO.: *1*

APPLICATION: *INVOICING*

UNIVAC 60 & 120

PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand

TABULATING MACHINES

ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD-F1, F2, ETC: CONTROL - C1, C2, ETC.) DECIMAL LOCATION, NEGATIVE CONTROL AND TRANSFER LINE. INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N1	*F1*				*1/0*						*26*	*27*	*28*	*29*	*30* ▲
N2	*F2*				*4/3*					*31*	*32*	*33* ▲	*34*	*35*	*36*
N3															
N4															
N5															
N6															
N7															
N8															
N9															
N10															
N11															
N12															
N13															
N14															
N15															
N16															
N17															
N18															

ELEM. DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															
N22															
N23															
N24															
N25															
N26															
N27															
N28															
N29															
N30															
N31															
N32															
N33															
N34					*3/2*									▲ *0*	*3*
N35					*4/3*								▲ *0*	*0*	*5*
N36															

ACCUMULATOR INPUTS

INDICATE:
TOP LINE - TRANSFER LINE (F1, F2, ETC.)
BOTTOM LINE - CARD COLUMNS

SYM.	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
A1						*F1*	*F1*	*F1*	*F1*	*F1*
						26	*27*	*28*	*29*	*30* ▲
A2					*F2*	*F2*	*F2*	*F2*	*F2*	*F2*
					31	*32*	*33* ▲	*34*	*35*	*36*
A3										
A4										
A5										
A6										
A7										
A8										
A9										
A10										
A11										
A12										

REPRODUCE

FROM COLUMN																																																											
TO COLUMN																																																											
CONTROL																																																											

SELECTOR CONTROL CHART

SYMBOLS: SELECTOR . . . T1-1, T1-2, T1-3, T1-4, T2, ETC. COMMON . . . COM NON SELECT . . N.S.
SELECT . . . SEL. BRANCH . . . BR. PROCESS . . PR.
VALUE 1 . . . V1 VALUE 2 . . . V2 RESULT . . . RES.
ELEMENTS . . . N1, N2, N3, ETC. STORAGE . . . S1, S2, S3, ETC. PROGRAM SELECTOR . . . P.S.1, P.S.2, ETC.

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT	SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
11-1						1-1					
11-2						1-2					
11-3						1-3					
11-4						1-4					
12						2					
13-1						3-1					
13-2						3-2					
13-3						3-3					
13-4						3-4					
14						4					
15-1						5-1					
15-2						5-2					
15-3						5-3					
15-4						5-4					
16						6					
17-1						7-1					
17-2						7-2					
17-3						7-3					
17-4						7-4					
18						8					
19-1						9-1					
19-2						9-2					
19-3						9-3					
19-4						9-4					
31-1						21-1					
31-2						21-2					
31-3						21-3					
31-4						21-4					
32						22					
33-1						23-1					
33-2						23-2					
33-3						23-3					
33-4						23-4					
34						24					
35-1						25-1					
35-2						25-2					
35-3						25-3					
35-4						25-4					
36						26					
37-1						27-1					
37-2						27-2					
37-3						27-3					
37-4						27-4					
38						28					
39-1						29-1					
39-2						29-2					
39-3						29-3					
39-4						29-4					

S1-1370 RRBH 12456

A-6 Program comments

1. Elements

Table of factors
a) No card number is needed since only one type of card is involved.
b) Both card-read fields and constant values are entered.

Accumulator input
a) No constants are entered in this section.
b) Show the decimal location of each card-read field.

2. Storages

Since S1 is a working storage, no description is entered. However, its decimal location must be entered.

3. Program

Step 2: Although a final value, amount, has been computed, it is preferable not to set the value until the end of the program when all values have been computed. Therefore the next step is step 3 rather than set 1.

Step 3: Since the non-rounded amount which is in S1 at the beginning of this step is no longer needed, the result of step 3 may be placed in S1. This makes it unnecessary to assign an additional storage with a 4/3 decimal location.

Note that if .03 were V2 instead of V1, the multiplication time on this step would be longer. However, since the program is so short, it will operate at 150 cards a minute either way.

Step 5: At the completion of the program it is necessary to set the values which are to punch in the punching dies. Therefore the next step is "Set 1."

Set 1: The next instruction is trip. The computer could have been routed to clear, but this is actually unnecessary since all storages are cleared on every card by the entry of new information.

B. PROBLEM 2 - INVOICING

B-1 Statement of Problem:

Item cards for writing invoices have been pulled from a tub file. The pre-punched information includes quantity and price. Customer number is then punched in the cards, and they are sorted by customer number. On the collator a blank card is inserted following each customer which will become a summary card. The total amount of the invoice is to be punched in the summary card, as well as a discount based on the total amount, and a net amount. The following calculations are to be performed:

All cards

Determine whether the card is an item card or a summary card. This may be done by subtracting a field which is punched in the item card from zero. Note that normally the sequence of customer number would be checked; however, the check will be omitted for this problem.

Item cards

1. Store customer number so that it can be punched in the summary card.

2. Multiply quantity times price for the amount. Round the amount.

3. Accumulate the amount from card to card for the gross amount of the invoice.

Summary cards

1. Test gross amount to determine in which of the following brackets it falls.

Under	$ 500.00
500.00	- 999.99
1,000.00	- 1,999.99
2,000.00	- and over

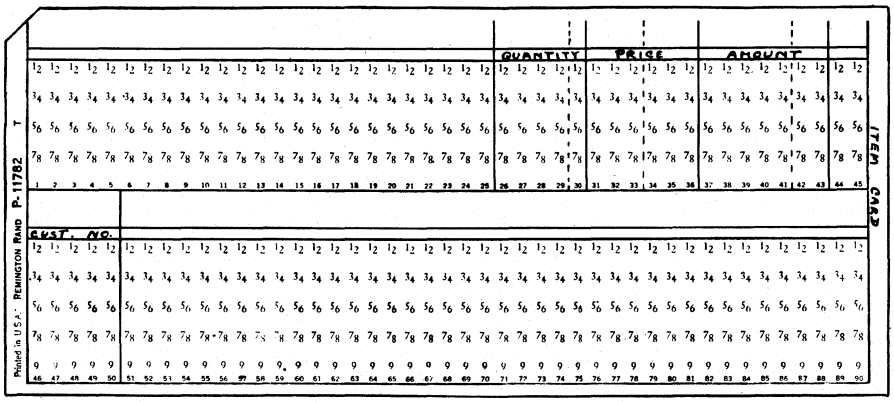
2. The total amount of the invoice is to be discounted at the following rates:

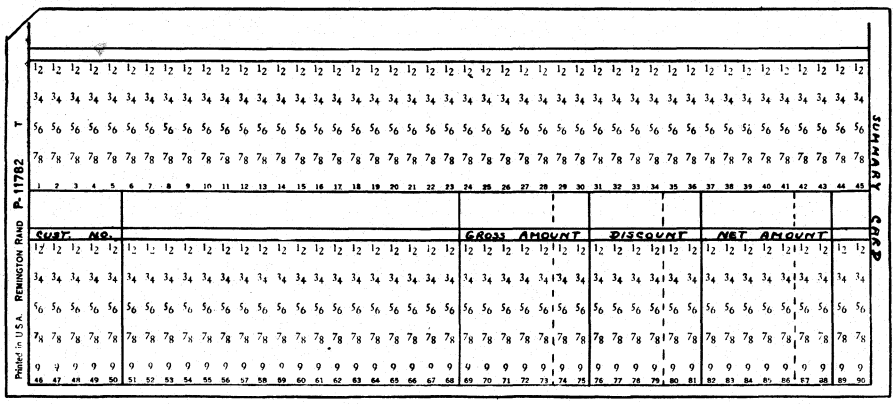
Amount	Discount
Under $500.00	.01
500.00 - 999.99	.015
1,000.00 - 1,999.99	.02
$2,000.00 - and over	.03

For example, an invoice of $1,500.00 would be computed as $1500 × .02 = $30.00. Round the amount of the discount.

3. Subtract the discount from the gross amount to equal the net amount.

B-2 Card forms:





B-3 Sense:

	Columns	Description
Item Card:	46-50	Customer Number
	26-30	Quantity
	31-36	Price
Summary card:	None	

B-4 Punch:

	Columns	Description
Item Card:	37-43	Amount
Summary card:	46-50	Customer Number
	69-75	Gross Amount
	76-81	Discount
	82-88	Net Amount

PROGRAM CHART NO. 2

APPLICATION: INVOICING

UNIVAC 60 & 120

PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand

TABULATING MACHINES

TABLE OF FACTORS

SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE
N1	+	1	CUSTOMER NO. (COLS. 46-50)	N13				N25			
N2	+	1	QUANTITY (COLS. 26-30)	N14				N26			
N3	+	1	PRICE (COLS. 31-36)	N15				N27			
N4				N16				N28	+		.03
N5				N17				N29	+		.02
N6				N18				N30	+		.015
N7				N19				N31	+		.01
N8				N20				N32	+		2000.00
N9				N21				N33	+		1000.00
N10				N22				N34	+		500.00
N11				N23				N35	+		.005
N12				N24				N36	+		0.000

CARD DESCRIPTION 1 = ITEM CARD S = SUMMARY CARD

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2 3/2	S3 3/2	S4 3/2	S5 3/2	S6	S7	S8	S9	S10	S11	S12 1/0	NEXT −	STEP +
1	1,S	ZERO	+	N36	−	CUSTOMER NO.	+	N1	TEST − ITEM CARD (+ S/C)	−	S1	1												2	7
2	1	CUSTOMER NO.	+	N1	+	ZERO	+	N36	STORE CUST. NO.	+	S12												(2)	−	3
3	1	QUANTITY	+	N2	X	PRICE	+	N3	AMOUNT (NR)	+	S1	3												−	4
4	1	AMOUNT (NR)	+	S1	+	.005	+	N35	AMOUNT (RD)	+	S2		(4)											−	5
5	1	Σ AMOUNT	+	S3	+	AMOUNT	+	S2	NEW Σ AMOUNT	+	S1	5												−	6
6	1	NEW Σ AMOUNT	+	S1	+	ZERO	+	N36	NEW Σ AMOUNT	+	S3			(6)										−	SET 1
7	S	GROSS	+	S3	−	500.00	+	N34	TEST − UNDER 500.00 (+ 500.00 OR OVER)	−	S1	7												10	8
8	S	GROSS	+	S3	−	1000.00	+	N33	TEST − 500.00 TO 999.99 (+1000.00 OR OVER)	−	S1	8												11	9
9	S	GROSS	+	S3	−	2000.00	+	N32	TEST − 1000.00 TO 1999.99 (+ 2000.00 OR OVER)	−	S1	9												12	13
10	S	.01	+	N31	X	GROSS	+	S3	DISCOUNT (NR)	+	S1	10												−	14
11	S	.015	+	N30	X	GROSS	+	S3	DISCOUNT (NR)	+	S1	11												−	14
12	S	.02	+	N29	X	GROSS	+	S3	DISCOUNT (NR)	+	S1	12												−	14
13	S	.03	+	N28	X	GROSS	+	S3	DISCOUNT (NR)	+	S1	13												−	14
14	S	DISCOUNT (NR)	+	S1	+	.005	+	N35	DISCOUNT (RD)	+	S4				(14)									−	15
15	S	GROSS	+	S3	−	DISCOUNT	+	S4	NET AMOUNT	+	S5					(15)								−	SET 2
16																									
17																									
18																									
19																									
20																									
21																									
22																									
23																									
24																									
25																									
26																									
27																									
28																									
29																									
30																									
31																									
32																									
33																									
34																									
35																									
36																									
37																									
38																									
39																									
40																									

SYM	CARD FIELD TITLE	NEG. COL	NEG. POS	OUTPUT PUNCHING 10	9	8	7	6	5	4	3	2	1	DEC LOC	SET 1	SET 2	CLEAR	SYM
S1														4/3			X	S1
S2	AMOUNT						37	38	39	40	41	42	43	3/2	X		X	S2
S3	GROSS AMOUNT						69	70	71	72	73	74	75	3/2		X	X	S3
S4	DISCOUNT							76	77	78	79	80	81	3/2		X	X	S4
S5	NET AMOUNT						82	83	84	85	86	87	88	3/2		X	X	S5
S6																		S6
S7																		S7
S8																		S8
S9																		S9
S10																		S10
S11																		S11
S12	CUSTOMER NO.								46	47	48	49	50	1/0		X	X	S12

FROM	TO
START	ST. 1
SET 1	TRIP
SET 2	CLEAR
SORT 1	
SORT 2	
CLEAR	TRIP
P.S. I	
P.S. II	
P.S. III	
P.S. IV	

REPRODUCE CONTROLS	COL.	POS.
REP		
SKIP		
SET HOLD		
SEC REP		

	O ÷ O	N ÷ O
STOP		
SORT		

RESTART	
STEP	X
PROGRAM	

81-1389 RRDH 12456

SELECTOR CHART NO.: 2.

APPLICATION: INVOICING

UNIVAC 60 & 120
PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand
TABULATING MACHINES

ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD-F1, F2, ETC: CONTROL - C1, C2, ETC.) DECIMAL LOCATION, NEGATIVE CONTROL AND TRANSFER LINE. INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N1	F1				1/0						46	47	48	49	50
N2	F2				2/1						26	27	28	29	30
N3	F3				1/3					31	32	33	34	35	36
N4															
N5															
N6															
N7															
N8															
N9															
N10															
N11															
N12															
N13															
N14															
N15															
N16															
N17															
N18															

ELEM. DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															
N22															
N23															
N24															
N25															
N26															
N27															
N28					3/2									0	3
N29					3/2									0	2
N30					1/3								0	1	5
N31					3/2									0	1
N32					3/2					2	0	0	0	0	0
N33					3/2					1	0	0	0	0	0
N34					3/2						5	0	0	0	0
N35					1/3								0	0	5
N36					1/3							0	0	0	0

ACCUMULATOR INPUTS

TOP LINE - TRANSFER LINE (F1, F2, ETC.) INDICATE:
BOTTOM LINE - CARD COLUMNS

SYM.	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
A1						F1	F1	F1	F1	F1
						46	47	48	49	50
A2						F2	F2	F2	F2	F2
						26	27	28	29	30
A3						F3	F3	F3	F3	F3
						31	32	33	34	35	36
A4										
A5										
A6										
A7										
A8										
A9										
A10										
A11										
A12										

REPRODUCE

FROM COLUMN	
TO COLUMN	
CONTROL	

SELECTOR CONTROL CHART

SYMBOLS: SELECTOR . . . T1-1, T1-2, T1-3, T1-4, T2, ETC.
COMMON . . . COM
NON SELECT . . N.S.
SELECT . . . SEL.
BRANCH . . . BR.
PROCESS . . PR.
VALUE 1 . . . V1
VALUE 2 . . . V2
RESULT . . . RES.
ELEMENTS . . . N1, N2, N3, ETC.
STORAGE . . . S1, S2, S3, ETC.
PROGRAM SELECTOR . . . P.S.1, P.S.2, ETC.

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
11-1					
11-2					
11-3					
11-4					
12					
13-1					
13-2					
13-3					
13-4					
14					
15-1					
15-2					
15-3					
15-4					
16					
17-1					
17-2					
17-3					
17-4					
18					
19-1					
19-2					
19-3					
19-4					
31-1					
31-2					
31-3					
31-4					
32					
33-1					
33-2					
33-3					
33-4					
34					
35-1					
35-2					
35-3					
35-4					
36					
37-1					
37-2					
37-3					
37-4					
38					
39-1					
39-2					
39-3					
39-4					

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1					
1-2					
1-3					
1-4					
2					
3-1					
3-2					
3-3					
3-4					
4					
5-1					
5-2					
5-3					
5-4					
6					
7-1					
7-2					
7-3					
7-4					
8					
9-1					
9-2					
9-3					
9-4					
21-1					
21-2					
21-3					
21-4					
22					
23-1					
23-2					
23-3					
23-4					
24					
25-1					
25-2					
25-3					
25-4					
26					
27-1					
27-2					
27-3					
27-4					
28					
29-1					
29-2					
29-3					
29-4					

B-5 Program comments

Step 1: Any of the three fields punched in the item card may be used as value 2. If the card being tested is a summary card, that field will be blank and the result will be a plus zero. If any number is punched, the result will be negative, indicating that the card is an item card. Since it is preferable to program the cards in the sequence in which they enter the computer, a negative result goes to step 2, which is the first step of the item card routine. A positive result goes to the first step of the summary card routine.

Step 2: Customer number is stored in S12 merely because that is the usual storage for designating information.

Step 3: Although quantity (with one decimal place) times price (three decimal places) could result in four digits following the decimal, only three places are significant when the result is to be rounded to two places. Therefore the non-rounded result can be placed in a 4/3 storage.

Set 1: The computer should not be directed to clear following set 1. Since it must be directed to clear gross amount in a summary card, a clear instruction at this point would clear the accumulating gross in each card.

Step 7: To obtain the desired results, this step could be stated one of two ways:

Gross - $500.00 or
$499.99 - Gross

The first method is preferable since it requires fewer wires to wire the constant. Since no indication is given of the volume of invoices in each group, it does not matter whether the first test is for $500 or $2000. Any card with an amount under $500 is directed immediately to its multiplication step, while cards of $500 and over must be tested further.

Step 8: It does not matter whether this step is a continuation of the testing or the discount computation for the first group.

Steps 10-13: Each card will go through only one of these steps. By always placing the result in the same storage, only one rounding step is needed. The exits of all of these steps are directed to the common rounding step.

Note that if the discount rates, which are smaller values, are V2 instead of V1, the time for multiplication will be greater. However, the summary card program is so short that it will operate at 150 cards a minute either way.

Step 15: Following the completion of the computation, the values must be set. Since set 1 was used for the items card, set 2 is used for the summary card.

Set 2: The set 2 instruction must be directed to clear.

Clear: It is essential to clear gross amount since the storage is used for accumulating values in the item cards. If it is not cleared, the gross amount of the previous invoice will be added to the amounts in the following cards. It takes no longer to clear the remaining storages as well, but this is not necessary.

C. PROBLEM 3 - LABOR EXTENSION

C-1 Statement of problem

This problem involves the extension of a day rate payroll. An employee may work on several different jobs during the day, with a separate labor ticket prepared for each one, which must be extended. Regardless of the job on which he is working, he receives the same rate. Since he may change shifts during the week, the shift on which he works is punched in the cards. Shift premium is added to the man's day rate before the card is extended.

At the end of the week an attendance card is punched from clock cards at the gate with the total number of hours worked during the week. This total is to be balanced to the total of the hours on the labor tickets.

There is a master rate file containing one card for each employee, with the employee's day rate punched.

The rate file, labor tickets, and attendance cards are sorted together in that sequence by employee number, then placed in the computer.

The following calculations are to be performed:

All cards

Test employee number to be certain that all cards within a group are the same. Stop the computer on an out-of-sequence card.

Rate cards

1. Store employee number.

2. Store rate.

Labor tickets

1. Add shift premium to day rate. Shift is punched 1, 2, or 3 for 1st, 2nd, or 3rd shift. The rates are:

1st shift:	.00
2nd shift:	.05
3rd shift:	.11

2. Multiply rate times hours to equal labor.

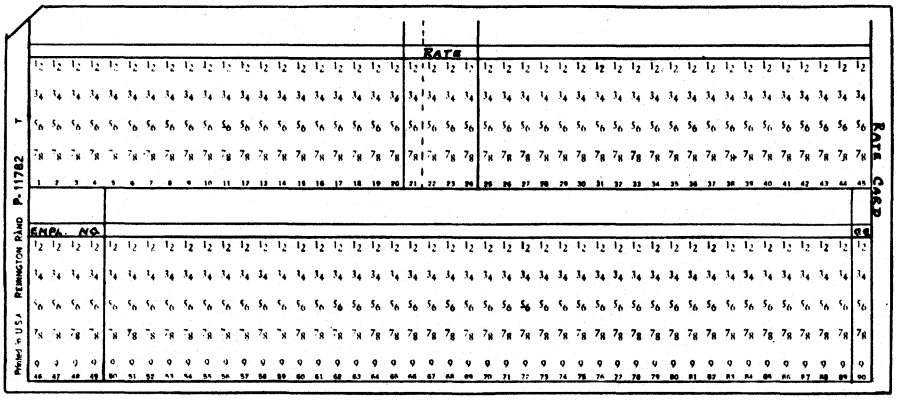
3. Accumulate labor.

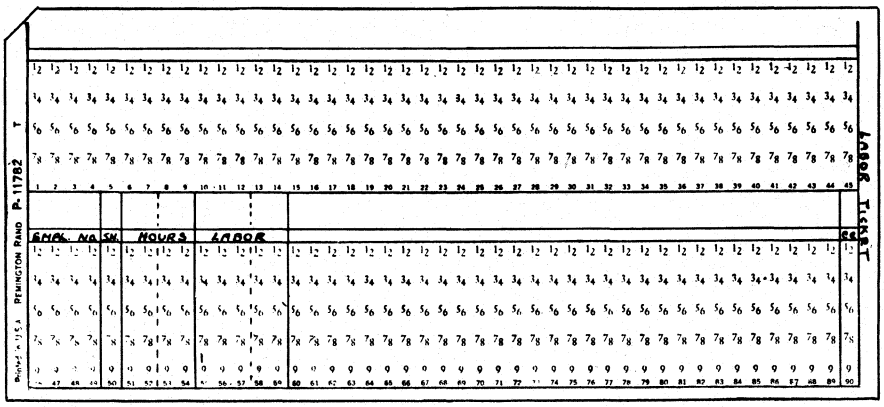
4. Accumulate hours.

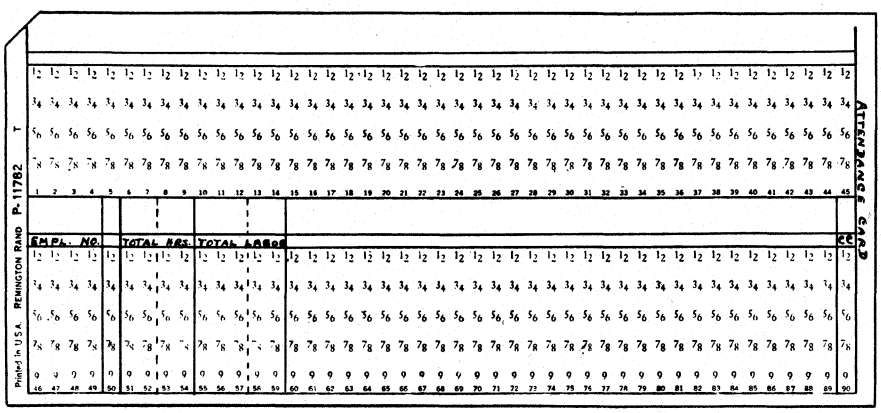
Attendance cards

Balance total hours with accumulated hours from the labor tickets. If the hours are not equal, sort the card.

C-2 Card forms







C-3 Sense:

	Columns	Description
Rate card:	90, pos. 1	Card code
	46-49	Employee number
	21-24	Rate
Labor ticket:	90, pos. 3	Card code
	46-49	Employee number
	50	Shift
	51-54	Hours
Attendance card:	90, pos. 5	Card code
	46-49	Employee number
	51-54	Total hours

C-4 Punch:

	Columns	Description
Labor ticket:	55-59	Labor
Attendance card:	55-59	Total labor

PROGRAM CHART NO. 3

APPLICATION: LABOR EXTENSION

UNIVAC 60 & 120

PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand

TABULATING MACHINES

TABLE OF FACTORS

SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE
N1	+	1,3,5	EMPLOYEE NO. (COLS. 46-49)	N13				N25			
N2	+	1	RATE (COLS. 21-24)	N14				N26			
N3	+	3/5	HOURS / TOTAL HOURS (COLS. 51-54)	N15				N27			
N4				N16				N28			
N5				N17				N29			
N6				N18				N30			
N7				N19				N31			
N8				N20				N32			
N9				N21				N33	+		.11
N10				N22				N34	+		.05
N11				N23				N35	+		.005
N12				N24				N36	+		0.000

CARD DESCRIPTION 1- RATE CARD 3- LABOR TICKET 5- ATTENDANCE CARD

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2 4/3	S3 3/2	S4 3/2	S5 3/2	S6	S7	S8	S9	S10	S11 4/3	S12 1/0	NEXT STEP −	NEXT STEP +
1	1,3,5	EMP. NO. (PCH)	+	N1	−	EMP. NO (ST.)	+	S12	TEST + MAY BE OK − ERROR	±	S1	1												40	OOM T2
2	3,5	EMP. NO. (ST.)	+	S12	−	EMP. NO (PCH)	+	N1	TEST + OK − ERROR	±	S1	2												40	OOM T4
3	1	EMP. NO. (PCH)	+	N1	+	ZERO	+	N36	STORE EMP. NO.	+	S12												(3)	−	4
4	1	RATE	+	N2	+	ZERO	+	N36	STORE RATE	+	S11											(4)		−	TRIP
5	3	RATE	+	S11	+	SHIFT PREM. RATE	+	COM T8	ADJUSTED RATE	+	S2		5											−	6
6	3	ADJ. RATE	+	S2	X	HOURS	+	N3	LABOR (NR)	+	S1	6	—											−	7
7	3	LABOR (NR)	+	S1	+	.005	+	N35	LABOR (RD)	+	S3	—		(7)										−	8
8	3	Σ LABOR	+	S4	+	LABOR	+	S3	NEW Σ LABOR	+	S1	8												−	9
9	3	NEW Σ LABOR	+	S1	+	ZERO	+	N36	NEW Σ LABOR	+	S4	—			(9)									−	10
10	3	Σ HOURS	+	S5	+	HOURS	+	N3	NEW Σ HOURS	+	S1	10												−	11
11	3	NEW Σ HOURS	+	S1	+	ZERO	+	N36	NEW Σ HOURS	+	S5	—				(11)								−	SET 1
12	5	Σ HOURS	+	S5	−	TOTAL HOURS	+	N3	TEST + MAY BE OK − ERROR	±	S1	12												SORT 1	13
13	5	TOTAL HOURS	+	N3	−	Σ HOURS	+	S5	TEST + OK − ERROR	±	S1	13												SORT 1	14
14	5	Σ LABOR	+	S4	+	ZERO	+	N36	TRANSFER Σ LABOR	+	S3			(14)										−	SET 2
15																									
16																									
17																									
18																									
19																									
20																									
21																									
22																									
23																									
24																									
25																									
26																									
27																									
28																									
29																									
30																									
31																									
32																									
33																									
34																									
35																									
36																									
37																									
38																									
39																									
40	ER	ZERO	+	N36	÷	ZERO	+	N36	ERROR STEP	+	S1	40													

SYM	CARD FIELD TITLE	OUTPUT PUNCHING	DEC LOC	SET 1	SET 2	CLEAR	SYM
s1			4/3			X	s1
s2			4/3			X	s2
s3	LABOR / TOTAL LABOR	55 56 57 58 59	3/2	X	X	X	s3
s4			3/2			X	s4
s5			3/2			X	s5
s6							s6
s7							s7
s8							s8
s9							s9
s10							s10
s11			4/3			X	s11
s12			1/0			X	s12

FROM	TO
START	ST. 1
SET 1	TRIP
SET 2	CLEAR
SORT 1	ST. 14
SORT 2	
CLEAR	TRIP
P.S. I	
P.S. II	
P.S. III	
P.S. IV	

REPRODUCE CONTROLS

	COL.	POS.
REP		
SKIP		
SET HOLD		
SEC REP		

	0 ÷ 0	N ÷ 0
STOP	X	X
SORT		

RESTART

STEP	X
PROGRAM	

81-1369 RRBH 12454

SELECTOR CHART NO.: 3

APPLICATION: LABOR EXTENSION

UNIVAC 60 & 120

PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand INC.

TABULATING MACHINES

ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD-F1, F2, ETC: CONTROL - C1, C2, ETC.) DECIMAL LOCATION, NEGATIVE CONTROL AND TRANSFER LINE. INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM DESIG	TO	NEG. CONT. COL	POS	TRF LINE	DEC LOC	10	9	8	7	6	5	4	3	2	1
N1	F1				1/0							46	47	48	49
N2	F2				4/3							21	22	23	24
N3	F3				3/2							51	52	53	54
N4															
N5															
N6															
N7															
N8															
N9															
N10															
N11															
N12															
N13															
N14															
N15															
N16															
N17															
N18															

ELEM. DESIG	TO	NEG. CONT. COL	POS	TRF LINE	DEC LOC	10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															
N22															
N23															
N24															
N25															
N26															
N27															
N28															
N29															
N30															
N31															
N32															
N33					3/2									1	1
N34					3/2									0	5
N35					4/3								0	0	5
N36					4/3							0	0	0	0

ACCUMULATOR INPUTS

TOP LINE - TRANSFER LINE (F1, F2, ETC.)
INDICATE:
BOTTOM LINE - CARD COLUMNS

SYM.	10	9	8	7	6	5	4	3	2	1
A1							F1	F1	F1	F1
							46	47	48	49
A2							F2	F2	F2	F2
							21	22	23	24
A3							F3	F3	F3	F3
							51	52	53	54
A4										
A5										
A6										
A7										
A8										
A9										
A10										
A11										
A12										

REPRODUCE

FROM COLUMN	
TO COLUMN	
CONTROL	

SELECTOR CONTROL CHART

SYMBOLS: SELECTOR . . . T1-1, T1-2, T1-3, T1-4, T2, ETC. — COMMON . . . COM — NON SELECT . . N.S.
SELECT . . . SEL. — BRANCH . . . BR. — PROCESS . . PR.
VALUE 1 . . . V1 — VALUE 2 . . . V2 — RESULT . . . RES.
ELEMENTS . . . N1, N2, N3, ETC. — STORAGE . . . S1, S2, S3, ETC. — PROGRAM SELECTOR . . . P.S.1, P.S.2, ETC.

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
11-1					
11-2					
11-3					
11-4					
12	C12	9/50 (2 SH.)	N 34	SEL. T 8	N 36
13-1					
13-2					
13-3					
13-4					
14	C14	3/50 (3 SH.)	N33	N.S. T8	—
15-1					
15-2					
15-3					
15-4					
16					
17-1					
17-2					
17-3					
17-4					
18					
19-1					
19-2					
19-3					
19-4					
31-1					
31-2					
31-3					
31-4					
32					
33-1					
33-2					
33-3					
33-4					
34					
35-1					
35-2					
35-3					
35-4					
36					
37-1					
37-2					
37-3					
37-4					
38					
39-1					
39-2					
39-3					
39-4					

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1					
1-2					
1-3					
1-4					
2	C2	1/90 (RATE)	ST. 3	+ BR. ST. 1	ST. 2
3-1					
3-2					
3-3					
3-4					
4	C4	3/90 (LABOR)	ST. 5	+ BR. ST. 2	COM. T 6
5-1					
5-2					
5-3					
5-4					
6	C6	5/90 (ATT.)	ST. 12	N.S. T4	—
7-1					
7-2					
7-3					
7-4					
8	C8	1/50 (1 & 2 SH.)	COM. T 12	V2 - ST. 5	COM. T 14
9-1					
9-2					
9-3					
9-4					
21-1					
21-2					
21-3					
21-4					
22					
23-1					
23-2					
23-3					
23-4					
24					
25-1					
25-2					
25-3					
25-4					
26					
27-1					
27-2					
27-3					
27-4					
28					
29-1					
29-2					
29-3					
29-4					

51 1370 RRBH 12456

C-5 Program comments

1. Elements

Although hours and total hours are punched in different types of card, the card columns are identical. Therefore only one element is needed.

Since shift is used to pick up selectors, it is not read in as an element.

2. Storages

The storages in which rate is stored from the rate card and in which the adjusted rate is placed must both have 4/3 decimal locations, or the last digit of rate will be dropped off.

Labor and total labor cannot be Y-wired into the same card columns for punching. Therefore they are both punched from the same storage.

3. Program

Step 1: A rate card will always branch on the plus side of step 1. The plus branching is routed to selector T2, picked up by a 1 in column 90. If the selector is select, the card is a rate card and is directed to step 3, the first step of the rate card routine. If the selector is non-select, the card is either a labor card or an attendance card. In either case the sequence check must be completed, so the cards are all sent to step 2. Any card branching on the minus side of step 1 must be an error.

Step 2: Any card branching on the minus side of step 2 must be an error. Any card branching on the plus side must be correct. The plus branching is sent to selector T4, picked up by a 3 in column 90. If the selector is select, the card is a labor card, and is directed to the first step of the labor card routine. If the selector is non-select, the card should be an attendance card. However, for positive checking it is preferable to send the non-select of T4 to selector T6, picked up by a 5 in column 90. If the selector is select, the card is sent to the first step of the attendance card routine. If the selector is non-select, the card does not have a 1, 3, or 5 punched, and is therefore an error. The computer will hang up, since there is no instruction as to the next step.

Step 4: No punching is to occur in a rate card. Therefore the card does not go to a set instruction following the completion of its program, but directly to trip.

Step 5: During this step the shift premium rate is added to day rate, with selectors used to determine which constant value will be used. Value 2 is wired to the common of T8, which is picked up by a 1 in column 50. T8 will be select if either a 1 or a 2 is punched in column 50, since the punched code for a 2 is a combination of 1 and 9. It is therefore necessary to wire the select side to T12, which is picked up by a 9. If T12 is select, the card is 2nd shift, and N34, .05, is to be used as the shift rate. If T12 is non-select, only a 1 is punched in the card and N36, zero, is to be used as shift premium rate. Note that an element must be wired on the non-select side of T12, or

the computer will hang up because there is no V2.

If T8 is non-select, the shift should be 3rd. However, for positive checking it is preferable to wire the non-select side of T8 to the common of T14. If there is a 3 punched in column 50, T14 will be select and N33, .11, will be used as V2. If there is not a 3 punched in column 50, the card is incorrect, and the computer will hang up because there is not a 1, 2, or 3 punched.

Set 1: When the program for a labor card has been completed, the computer is instructed to set S3 on set 1. At this time the labor amount for the card in the computer is stored in S3. When this has been set, the next instruction is to trip. Clear should not be used, or the accumulating totals will be cleared in each labor card.

Steps 12 and 13: These two steps are the pair used to determine whether one number is equal to another. A minus result on either step indicates that the two numbers are not equal, and the computer is directed to sort the card. Since the step following the sort instruction will be the same regardless of whether the card is sorted on step 12 or 13, sort 1 may be used for the minus branching of both steps.

Sort 1: Sort 1 is wired to step 14 to complete the program. This is the same step to which the cards which balance are wired.

Step 14: Step 14 is the transfer of the total labor from S4 in which it has been accumulating to S3. It is necessary because S4 and S3 cannot both be wired to columns 55-59, or a backfeed would develop.

Set 2: Since only S3 is to be set both times a set instruction is given to the computer, either set 1 or set 2 could be used for the second set instruction. If set 1 is used here, however, the next instruction must be controlled through a selector. In a labor card the next instruction is trip, whereas in an attendance card the next instruction must be clear. Therefore it is easier to use set 2 as the second set instruction.

Clear: The attendance card must be directed to clear both S4 and S5, in which labor and hours are accumulating. Though not necessary, it does no harm to clear the other storages which are used.

D. PROBLEM 4 - ANNUAL PHYSICAL INVENTORY

D-1 Statement of problem:

This problem is the extension of a physical inventory taken once each year. One part may be stored in several locations, so there may be several inventory cards for one part number. The inventory is taken on pre-numbered stub cards, with the stub left at the location of the part. When the cards are received by the tabulating department, they are punched with part number, quantity, and certain designating information which has no bearing on this problem.

A master file of cards is maintained which is punched with part number, price, and unit of measure. The price may be price each, price per hundred, or price per thousand, as indicated by the unit of measure. The problem involves the write-down or reduction of the inventory valuation of any part which has not been ordered this year and of which there is three or more years supply on hand. A file of cards containing only part number is maintained during the year, with one card for each part which has been ordered, either from an outside supplier or on an order to manufacture within the company. The summary cards from the final stock status report for the year contain the total quantity used during this year. By dividing this quantity used into the total inventory quantity, the number of years supply will be obtained. For example, if 5 have been used this year and there are 10 on hand, there is a two year supply.

The price cards, inventory cards, order cards, and quantity used cards are sorted together by part number. A blank card, which will be used as a summary card, is collated following each part number.

The calculations are:

All cards

Test part number to be certain that all cards within a group contain the same number.

Price card (one for each part)

1. Store part number.

2. Store price.

Order card (one may or may not be present for each part)

Remember whether this card was present. A 0 in column 44, which is not present in any other card, may be used as a selector pick-up.

Usage card (one may or may not be present for each part)

Store quantity used during the year.

Inventory card (one or several for each part)

1. Multiply quantity times price per unit for the amount.

2. Accumulate inventory quantity.

3. Accumulate amount.

Blank summary card (one for each part)

1. If the part has been ordered during the year, no write-down is to be calculated. The net amount is the same as the total amount.

2. If the part has not been ordered, test to determine whether it has been used during the year. If it has not been used, the division of inventory quantity by quantity used will result in a number divided zero, to which the computer will give a zero answer. This will place it in the category of no write-down, when actually it should be written down the maximum amount.

3. Divide the total inventory quantity by the quantity used during the year to determine the number of years supply. Drop off all decimals in the quotient. If the number of years supply is zero, 1, or 2, no write-down is to be calculated, and the net amount is the same as the total amount.

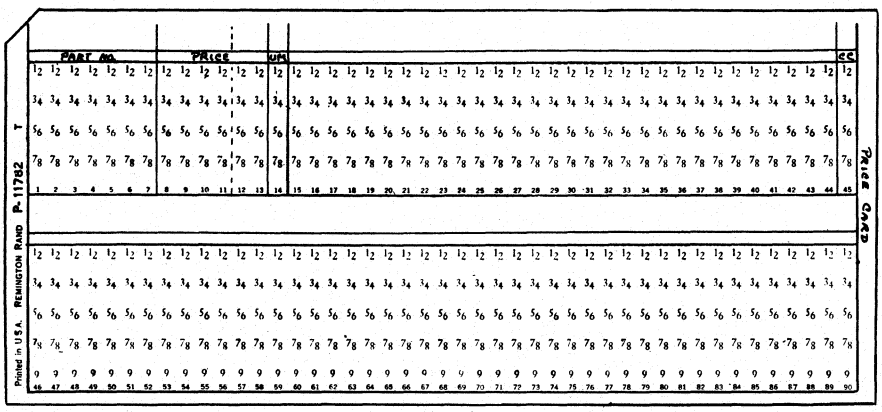
4. If the number of years supply is 3 or more, the total amount (dollar value) is to be multiplied by the following percentages to obtain the write-down amount.

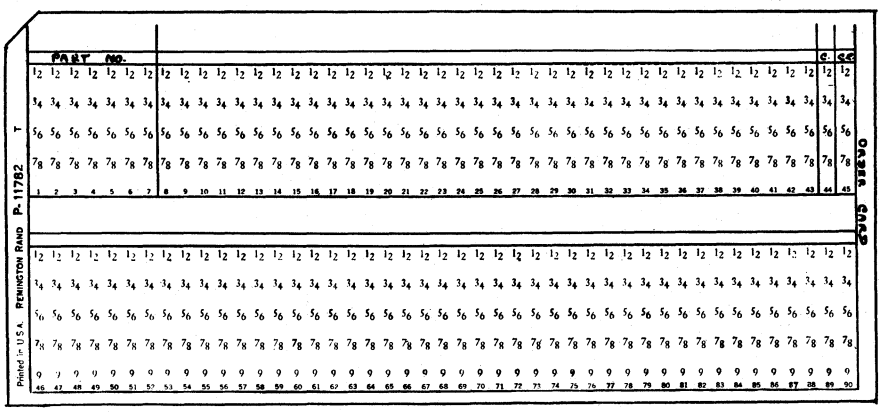
3 years - 25%
4 years - 50%
5 or more years - 75%

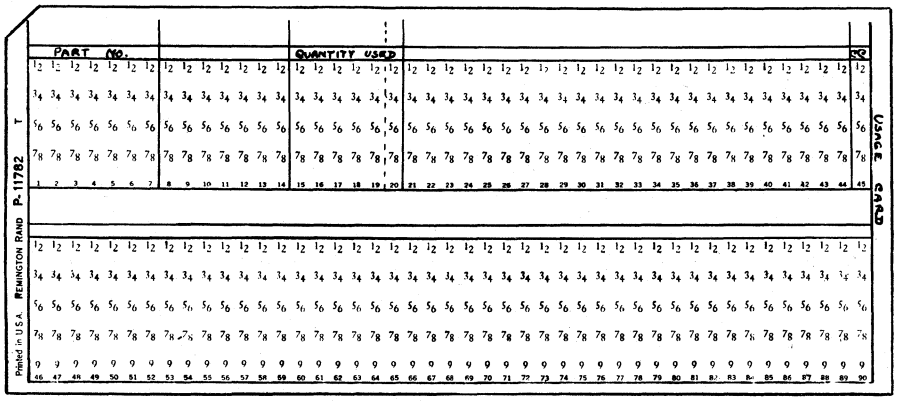
The write-down amount is to be rounded.

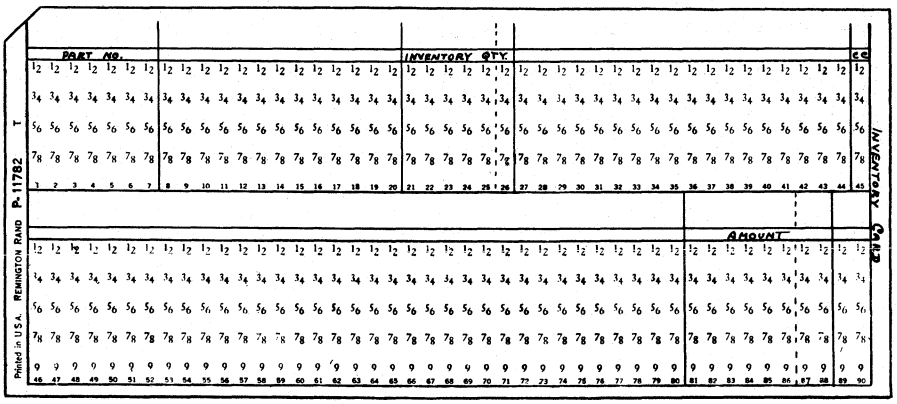
5. Subtract the write-down amount from the gross amount to equal the net amount.

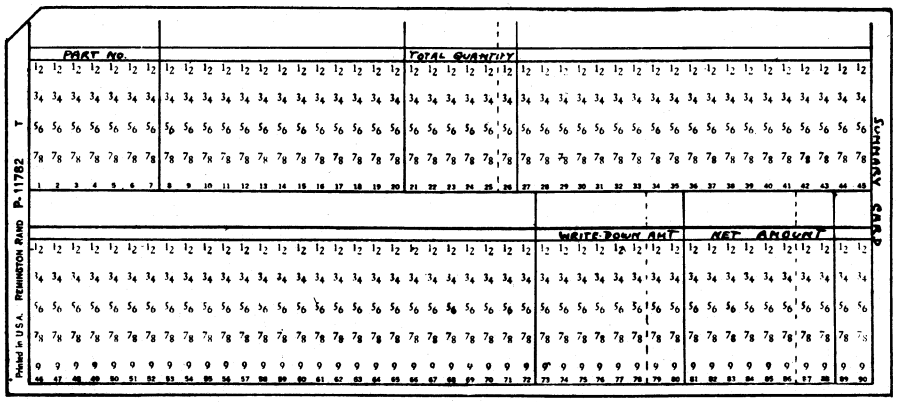
D-2 Card forms:




PART NO.
C.
CC
ORDER CARD
P-11782
REMINGTON RAND
Printed in U.S.A.


PART NO.
QUANTITY USED
CC
USAGE CARD
P-11782
REMINGTON RAND
Printed in U.S.A.


PART NO.
INVENTORY QTY.
CC
AMOUNT
INVENTORY CARD
P-11782
REMINGTON RAND
Printed in U.S.A.



D-3 Sense:

	Columns	Description
Price card:	45, pos. 1	Card code
	1- 7	Part number
	8-13	Price
	14	Unit of measure { 1 = each; 2 = per hundred; 3 = per thousand
Order card:	45, pos. 3	Card code
	44, pos. 0	Selector pick-up
	1- 7	Part number
Usage card:	45, pos. 5	Card code
	1- 7	Part number
	15-20	Quantity used (each, not in unit of measure)
Inventory card:	45, pos. 7	Card code
	1- 7	Part number
	21-26	Inventory quantity (each, not in unit of measure)
Summary card:	None	

D-4 Punch:

	Columns	Description
Inventory card:	81-88	Amount
Summary card:	1- 7	Part number
	21-26	Total quantity
	73-80	Write-down amount (if any)
	81-88	Net amount

PROGRAM CHART NO. 4

APPLICATION: ANNUAL PHYSICAL INVENTORY

UNIVAC 60 & 120
PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand
TABULATING MACHINES

TABLE OF FACTORS

SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE
N1	+	1,3,5,7	PART NO. (COLS. 1-7)	N13				N25			
N2	+	1	PRICE (COLS. 8-13)	N14				N26			
N3	+	5	QUANTITY USED (COLS. 15-20)	N15				N27			
N4	+	7	INVENTORY QUANTITY (COLS. 21-26)	N16				N28			
N5				N17				N29	+		.75
N6				N18				N30	+		.50
N7				N19				N31	+		.25
N8				N20				N32	+		5
N9				N21				N33	+		4
N10				N22				N34	+		3
N11				N23				N35	+		.005
N12				N24				N36	+		0.000

CARD DESCRIPTION 1= PRICE CARD 3= ORDER CARD 5= USAGE CARD 7= INVENTORY CARD 5= SUMMARY CARD

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2 1/0	S3	S4	S5	S6 3/2	S7 3/2	S8 2/1	S9 3/2	S10 2/1	S11 6/5	S12 1/0	NEXT -	STEP +
1	ALL	PART NO. (PCH)	+	N1	-	PART NO (ST)	+	S12	TEST - 3/C, ERROR + MAY BE OR, 1ST CARD	+	S1	1												3	COM T1-1
2	3, 7	PART NO. (ST)	+	S12	-	PART NO (PCH)	+	N1	TEST - ERROR + OR	+	S1	2												40	COM T2
3	5 ERR	ZERO	+	N36	-	PART NO. (PCH)	+	N1	TEST - ERROR + 3/C	+	S1	3												40	COM T5-2
4	1	PART NO (PCH)	+	N1	+	ZERO	+	N36	STORE PART NO.	+	S12											(4)		-	5
5	1	PRICE	+	N2	+	ZERO	+	N36	STORE ADJUSTED PRICE	+	S11										(5)			-	TRIP
6	5	QUANTITY USED	+	N3	+	ZERO	+	N36	STORE QTY. USED	+	S10									(6)				-	TRIP
7	7	QUANTITY	+	N4	X	ADJUSTED PRICE	+	S11	AMOUNT (NR)	+	S1	7												-	8
8	7	AMOUNT (NR)	+	S1	+	.005	+	N35	AMOUNT (RD)	+	S9								(8)					-	9
9	7	Σ QUANTITY	+	S8	+	QUANTITY	+	N4	NEW Σ QUANTITY	+	S1	9												-	10
10	7	NEW Σ QUANTITY	+	S1	+	ZERO	+	N36	NEW Σ QUANTITY	+	S8								(10)					-	11
11	7	Σ AMOUNT	+	S7	+	AMOUNT	+	S9	NEW Σ AMOUNT	+	S1	11												-	12
12	7	NEW Σ AMOUNT	+	S1	+	ZERO	+	N36	NEW Σ AMOUNT	+	S7							(13)						-	SET 1
13	5	ZERO	+	N36	-	QUANTITY USED	+	S10	TEST - USED + NOT USED	+	S1	13												14	18
14	5	Σ QUANTITY	+	S8	÷	QUANTITY USED	+	S10	NO. YEARS SUPPLY	+	S2		14											-	15
15	5	NO. YEARS SUPPLY	+	S2	-	3	+	N34	TEST - 0, 1, 2 + 3 OR MORE	+	S1	15												20	16
16	5	NO. YEARS SUPPLY	+	S2	-	4	+	N33	TEST - 3 + 4 OR MORE	+	S1	16												P31	17
17	5	NO. YEARS SUPPLY	+	S2	-	5	+	N32	TEST - 4 + 5 OR MORE	+	S1	17												P32	18
18	5	WRITE-DOWN %	+	COM T14	X	Σ AMOUNT	+	S7	WRITE-DOWN AMT (NR)	+	S1	18												-	19
19	5	WRITE DOWN AMT (NR)	+	S1	+	.005	+	N35	WRITE DOWN AMT (RD)	+	S6							(19)						-	20
20	5	GROSS AMT	+	S7	-	WRITE-DOWN AMT	+	S6	NET AMOUNT	+	S9									(20)				-	SET 2
21																									
22																									
23																									
24																									
25																									
26																									
27																									
28																									
29																									
30																									
31																									
32																									
33																									
34																									
35																									
36																									
37																									
38																									
39																									
40		ZERO	+	N36	÷	ZERO	+	N36	ERROR STEP	+	S1	40													

SYM	CARD FIELD TITLE	NEG. COL	NEG. POS	OUTPUT PUNCHING 10	9	8	7	6	5	4	3	2	1	DEC LOC	SET 1	SET 2	CLEAR	SYM
S1											▲			4/3			X	S1
S2													▲	1/0			X	S2
S3																		S3
S4																		S4
S5																		S5
S6	WRITE-DOWN AMT					73	74	75	76	77	78	▲ 79	80	3/2		X	X	S6
S7												▲		3/2			X	S7
S8	Σ QUANTITY							21	22	23	24	25	▲ 26	2/1		X	X	S8
S9	AMOUNT / NET AMOUNT				81	82	83	84	85	86	▲ 87	88		3/2	T3-2	X	X	S9
S10												▲		2/1			X	S10
S11								▲						6/5			X	S11
S12	PART NO.						▲ 1	2	3	4	5	6	7	1/0		X	X	S12

FROM	TO
START	ST. 1
SET 1	TRIP
SET 2	CLEAR
SORT 1	
SORT 2	
CLEAR	TRIP
P.S. I	ST. 18
P.S. II	ST. 18
P.S. III	
P.S. IV	

REPRODUCE CONTROLS	COL.	POS.
REP		
SKIP		
SET HOLD		
SEC REP		

	0 ÷ 0	N ÷ 0
STOP	X	X
SORT		

RESTART	
STEP	X
PROGRAM	

81-1369 RRBH 12456

SELECTOR CHART NO.: 4

APPLICATION: ANNUAL PHYSICAL INVENTORY

UNIVAC 60 & 120
PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand
TABULATING MACHINES

ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD-F1, F2, ETC: CONTROL - C1, C2, ETC.) DECIMAL LOCATION, NEGATIVE CONTROL AND TRANSFER LINE. INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N1	F1				1/0				1	2	3	4	5	6	7
N2	F2				COM T6				8	9	10	11	12	13	
N3	F3				2/1				15	16	17	18	19	20	
N4	F4				2/1				21	22	23	24	25	26	
N5															
N6															
N7															
N8															
N9															
N10															
N11															
N12															
N13															
N14															
N15															
N16															
N17															
N18															

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															
N22															
N23															
N24															
N25															
N26															
N27															
N28															
N29					3/2									7	5
N30					3/2									5	0
N31					3/2									2	5
N32					1/0										5
N33					1/0										4
N34					1/0										3
N35					4/3								0	0	5
N36					4/3								0	0	0

ACCUMULATOR INPUTS

TOP LINE - TRANSFER LINE (F1, F2, ETC.)
INDICATE:
BOTTOM LINE - CARD COLUMNS

SYM.	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
A1				F1	F1	F1	F1	F1	F1	F1
				1	2	3	4	5	6	7
A2				F2	F2	F2	F2	F2	F2	F2
				8	9	10	11	12	13	
A3				F3	F3	F3	F3	F3	F3	F3
				15	16	17	18	19	20	
A4				F4	F4	F4	F4	F4	F4	F4
				21	22	23	24	25	26	
A5										
A6										
A7										
A8										
A9										
A10										
A11										
A12										

REPRODUCE

FROM COLUMN	
TO COLUMN	
CONTROL	

SELECTOR CONTROL CHART

SYMBOLS: SELECTOR . . . T1-1, T1-2, T1-3, T1-4, T2, ETC.
COMMON . . . COM
NON SELECT . . N.S.
SELECT . . . SEL.
BRANCH . . . BR.
PROCESS . . PR.
VALUE 1 . . . V1
VALUE 2 . . . V2
RESULT . . . RES.
ELEMENTS . . . N1, N2, N3, ETC.
STORAGE . . . S1, S2, S3, ETC.
PROGRAM SELECTOR . . . P.S.1, P.S.2, ETC.

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
11-1					
11-2					
11-3					
11-4					
12	C12	3/14 (PER 1000)	6/5	NS T6	—
13-1					
13-2					
13-3					
13-4					
14		PS1 (25%)	N31	V1 - ST. 18	COM T16
15-1					
15-2					
15-3					
15-4					
16		PS2 (50%)	N30	NS T14	N29
17-1					
17-2					
17-3					
17-4					
18					
19-1					
19-2					
19-3					
19-4					
31-1					
31-2					
31-3					
31-4					
32					
33-1					
33-2					
33-3					
33-4					
34					
35-1					
35-2					
35-3					
35-4					
36					
37-1					
37-2					
37-3					
37-4					
38					
39-1					
39-2					
39-3					
39-4					

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1	C1	1/45 (PRICE CARD)	ST. 4	+ BR. ST. 1	ST. 2
1-2			—	COM T 5-1	SELECTOR HOLD
1-3					
1-4					
2	C2	3/45 (ORDER)	TRIP	+ BR. ST. 2	COM. T 4
3-1	C3	7/45 (INVENTORY)	ST. 7	N.S. T 4	—
3-2			SET 1	SET 59	SET 2
3-3					
3-4					
4	C4	5/45 (USAGE)	ST. 6	NS T2	COM T 3-1
5-1	C5	0/44 SEL. T5-1 (SEL. HOLD)	P.U. T5	COM. T 1-2	—
5-2			ST. 20	+ BR. ST. 3	ST. 13
5-3					
5-4					
6	C6	1/14 (EACH, PER)	COM. T8	DEC. LOC. N2	COM. T12
7-1					
7-2					
7-3					
7-4					
8	C8	9/14 (PER 100)	5/4	SEL. T6	3/2
9-1					
9-2					
9-3					
9-4					
21-1					
21-2					
21-3					
21-4					
22					
23-1					
23-2					
23-3					
23-4					
24					
25-1					
25-2					
25-3					
25-4					
26					
27-1					
27-2					
27-3					
27-4					
28					
29-1					
29-2					
29-3					
29-4					

D-5 Program comments:

1. Elements

The decimal location of price is shown as the common of selector T6. Price is given as price each, per hundred, or per thousand, while quantity is always in units. Therefore an adjustment must be made so that the correct result of multiplication will be obtained. Quantity should not be changed, since a total quantity in units is required.

Price may be adjusted in at least two ways. The preferable method is that of changing the decimal location, depending on the unit of measure. The decimal location is wired to the common of selector T6, picked up by a 1 in column 14. If the selector is select, either a 1 or 2 is punched. The select side is wired to the common of T8, picked up by a 9 in column 14. If the 9 is not present, only a 1 is punched, the unit of measure is each, and the desired decimal location is 3/2. If T8 is select, a 2 is punched, and the unit of measure is per hundred. By moving the decimal location two places to the left of that shown on the card, the same effect will be obtained as dividing the price by 100; therefore the decimal location should be 5/4.

If selector T6 is non-select, the coding in the card should be 3. For positive proof, however, the non-select of T6 is wired to the common of T12. If T12 is select, the price is per thousand. By moving the decimal location three places to the left of that in the card, the same effect will be obtained as dividing the price by 1000; therefore the decimal location should be 6/5. If T12 is non-select, there is not a 1, 2, or 3 punched and the computer will hang up the first time that price is used, because N2 will have no decimal location.

A second method of adjusting price is the dividing of price by 1, 100, or 1000, or multiplying it by 1, .01, or .001 when it is stored. To do this, the values of 1, 100, and 1000 or 1, .01, and .001 would be set up as constants. The same selectors as above would be used to select the proper constant. The step would be:

Price (N2) ÷ 1/100/1000 (Com. T6) = Adjusted price (S11) or
1/.01/.001 (Com. T6) × Price (N2) = Adjusted price (S11)

The constant values would be wired on the select and non-select sides of T8 and the select of T12 instead of the decimal locations shown above.

Note that a 1 must be used as the divisor for those parts for which the price is stated as each. If zero were used as a divisor, the quotient would be zero (N ÷ 0 = 0.) 1 would also be used as the multiplier for a price stated as each, since zero multiplied times the price would also result in zero.

On a long program, the method of using a division step or multiplication step is not as satisfactory as adjusting the decimal location. They require more calculating time than the addition step which can be used for storing price if the decimal is selected.

2. Program

Step 1: All cards go through the first step of the testing routine. If the

result is minus, the card could be an error. However, it could also be a summary card, since part number is not punched and the subtraction of the stored part number from zero would result in a minus. The minus branching is directed to step 3 to determine whether the card is an error or a summary card.

If the branching is plus, the card may be correct. Since there is a control hole in the first card of a group, these cards should be separated at the end of step 1. Therefore the plus branching is routed to the common of T1-1. If the selector is select, the card is a price card and is directed to step 4, the first step of the price card routine. All other cards are sent to step 2 for the completion of the testing routine.

Step 2: Any card branching on the minus side is an error, while all cards branching on the plus side are in sequence. From the plus branching the cards go to the first step of their routines. The selector wiring will be discussed in connection with each type of card, beginning with the order card.

Step 3: Errors and summary cards are directed to this step from the minus branching of step 1. If the result of this step is plus, there is no punched part number, and the card is a summary card. The selector used at the plus branching will be discussed under the routine for the summary card. If the result is minus there is a punched part number and the card is in error. It is therefore directed to step 40.

Price Card

If selector T1, picked up by a 1 in column 45, is select, the card is a price card. The plus branching of step 1 is wired to the common of T1-1, and all price cards are immediately sent to step 4.

Step 4: Part number is stored. The price card is always the first of the group, and the part number in all succeeding cards is compared against this part number.

Step 5: The adjusted price is stored. As discussed on page 274 this is simply an addition step if the decimal location of price is selected. If a fixed decimal location of 3/2 is given to price, this step would be the division of price by 1, 100, or 1000, or the multiplication of price by 1, .01, or .001 depending on the unit of measure. Since no punching is to occur in the price card, the plus branching is directed to trip.

Order Card

From the plus branching of step 2, all cards are directed to the common of T2, picked up by a 3 in column 45. If T2 is non-select the card is not an order card, and is routed to T4. If T2 is select, the card is an order card.

The purpose of including the order card is so that during the routine for the summary card the computer will know whether or not the part has been ordered during the year. This may be done in two ways. One method is to use a program step to store any one of the constant values which have been wired. Then during the routine for the summary card the storage in which this has been placed is

subtracted from zero. If the result is plus, nothing has been stored, and therefore there was no order card. If the result is minus, a value has been placed in storage, and the part was ordered during the year. Note that with this method the storage in which the indication is stored must be cleared at the end of each summary card. This method also requires the use of two steps and a storage, which may be avoided by using a selector hold.

It is stated as part of the problem that a zero in column 44 is available in this card, and may be used as selector pick-up. The selector, once it has been picked up, must be held by selector hold, since otherwise it will return to non-select as soon as the order card is tripped out.

The zero in 44 must pick up a four-pole selector, since one pole is used for selector hold wiring, while a second is used for the wiring of branching in the summary card. T5 is used in this example. T5 must be dropped out following its use in the summary card and before another order card is sensed. The logical place to do this is when the price card is in the computer, although a program select could be used for this purpose in the summary card.

The wiring of selector hold is as follows: Selector hold is wired to the non-select of T1-2, which is picked up on a price card. The common of T1-2 is wired to the common of T5-1. The select of T5-1 is wired to pick up T5. The zero in column 44 is also wired to pick up T5.

Until an order card enters the computer, T5 is non-select. As soon as a card with zero in column 44 is sensed, T5 becomes select. The power emitting from the pick up of T5 passes through the select of T5 and non-select of T1 and reaches selector hold ground. T5 therefore remains select, even though the order card leaves the computer. It remains select until the next price card enters, at which time T1 is picked up and the connection to ground is broken.

Note that a 3 in column 45, the card code for an order card, cannot be used with selector hold. If the selector picked up by a 3 in 45 were to remain select until the next price card, all cards entering the computer from the order card to the price card would actually be sensed as order cards.

Since the pick-up of T5 is the means of remembering that the order card was present, no further steps are required for an order card once it has completed step 2. The select of T2, picked up by a 3 in column 45, is therefore wired to trip.

Usage Card

From the plus branching of step 2 all cards are directed to the common of T2, which is non-select unless an order card is sensed. The non-select of T2 is wired to the common of T4, picked up by a 5 in column 45, which is the code for a usage card. The select of T4 is wired to step 6, the first step of the usage card.

Step 6: Quantity used is stored. This is the only step required on a usage card, so the plus branching is wired to trip.

Inventory Card

If a card which branches plus from step 2 is neither an order card nor a usage card, it should be an inventory card. For positive checking, however, the non-select of T4 is wired to the common of T3-1, picked up by a 7 in column 45. If the selector is select, the card is an inventory card and is wired to step 7, the first step of the inventory card routine. If the selector is non-select, there was not a 3, 5, or 7 punched. The computer will hang up for lack of a next instruction.

Steps 7, 8: These are normal steps for computing and rounding the amount.

Steps 9-12: These are normal steps for accumulating quantity and amount.

Set 1: The only value to be punched in the inventory card is amount from S9. However, S9 must also be set on the summary card, since amount is to be punched in the same columns as in the inventory card. To prevent the setting of the additional storages which must be set in the summary card both on the summary card and the inventory card, S9 must be wired through a selector. T3 is convenient for this purpose, since it is picked up on an inventory card. The setting of S9 is wired to the common of T3-2, and on the select side to set 1. Following set 1 the inventory card is tripped out.

Summary Card

The summary card is identified at the plus branching of step 3. The problem states that any card which has been ordered during the year does not have any write-down calculation. Since this is the first calculation to be performed, cards for parts which were ordered may be routed directly to step 20, which transfers total amount to the storage from which it will be punched. Selector T5 is select if the part has been ordered. The plus branching of step 3 is wired to the common of T5-2, and on the select side to step 20. If the part has not been ordered, T5 is non-select and the program continues with step 13.

Step 13: The quantity used storage must be tested to determine whether any parts have been used during the year. If none have been used, the part is to be written down a full 75%, which is the write-down amount for 5 or more years supply. If the branching is plus, meaning that none have been used, the card can be directed immediately to step 18, which multiplies the total amount by the write-down percent. Another possible way of handling the problem is to store a quantity used of .01 if the quantity used storage is zero. These cards can then follow the routine for all the other cards which were not ordered. When total quantity is divided by .01, the result will be at least 5, so the card will fall in the proper write-down class. For example:

$$2 \div .01 = 200$$

The cards on the minus branching of step 13, indicating that some were used during the year, are sent to step 14.

Step 14: Total quantity is divided by quantity used to determine the number of years supply. Note that the result is placed in a storage with a 1/0 decimal to drop off any decimals that might result. This is not necessary with steps 15-17 stated in the form as shown; however, if the alternate form of the testing

steps is used, as mentioned below, it becomes essential. The plus branching is routed to the first testing step to determine the number of years supply.

Step 15: The first testing step may be stated in one of two ways:

1. Number of years supply minus 3
2. 2 minus number of years supply

In either case, the cards with less than 3 years supply are separated, in the first example on the minus branching and in the second example on the plus branching. The same principles apply in both cases, so only the first example will be explained.

The cards with 0, 1 or 2 years supply, which branch minus on step 15, have no write-down computed. They are therefore directed to step 20, in which total amount is transferred to the storage from which it is to be punched. The cards which branch plus on step 15, which have 3 or more years supply, are routed to step 16.

Step 16: The cards with 3 or more years supply are tested. A minus branching indicates that there is a three years supply. PS 1 is impulsed to "remember" this fact, and the cards are sent from PS 1 to step 18, in which write-down is computed. A plus branching on step 16 indicates 4 or more years supply, and the cards are routed to step 17.

Step 17: The cards with 4 or more years supply are tested. A minus branching indicates a four years supply. PS 2 is impulsed to "remember" this fact, and the cards are sent from PS 2 to step 18, in which write-down is computed. A plus branching indicates 5 or more years supply, and these cards are sent directly to step 18.

Step 18: In this step write-down amount is computed for all three percents - 25, 50 and 75. Note that it is equally correct to use one multiplication step for each of the three multiplications. This approach saves the 20 milliseconds required to impulse a program select for those parts with three or four years supply. However, it does require two additional program steps, which might be important on a longer routine. In the example PS 1, which is impulsed if there is a three year supply, picks up T14. PS 2, which is impulsed if there is a four year supply, picks up T16. If the supply is 5 or more years, no selector is picked up. V2 of step 18 is wired to the common of T14. If T14 is select N31, which is 25%, is used as V2. The non-select of T14 is wired to the common of T16. If T16 is select, N30, which is 50%, is used as V2. If neither T14 nor T16 is select, there must be 5 or more years supply. Therefore N29, which is 75%, is wired as V2. Note that the cards from step 13 which had no quantity used go directly to step 18. Neither T14 nor T16 has been picked up, so 75% will be used as V2.

Step 19: A normal rounding step for write-down amount.

Step 20: All summary cards going through the computer are directed to step 20. Net amount must be placed in S9, since S9 is wired to the card columns in which net amount is to punch. If a write-down amount has been computed, this step will determine the net amount. If no write-down has been computed, as in the

case of parts which were ordered or of which there is less than three years supply, this step becomes a step transferring gross amount to S9 for punching. Since S6, the storage in which write-down amount would have been placed, is clear, the step becomes gross amount minus zero equals net amount. This has the same effect as the more usual step of gross amount plus zero equals net amount. Note that since the step is available, it would be correct to direct cards without a write-down amount to such a step instead of to step 20. From step 20, all cards are directed to set 2.

Set 2: S9 must be set on both set 1 and set 2, as explained under set 1. It is wired through selector T3-2, picked up when an inventory card is in the computer. On the non-select side of T3-2, therefore, the setting of S9 is wired to set 2. Part number, write-down amount, and total quantity are also set on set 2. From set 2, the cards must be directed to clear.

Clear: Certain storages must be cleared following set 2. They are:

S6 - Write-down amount. This is to prevent a write-down amount from a previous card from being punched in a card in which no write-down is computed.

S7 - Gross amount. An accumulating storage.

S8 - Total quantity. An accumulating storage.

The other storages may be cleared at the same time, although this is not necessary.

E. PROBLEM 5 - DAILY LABOR EXTENSION AND PAYROLL SUMMARY

E-1 Statement of problem

The problem involves the extension of a standard hour payroll. An employee may work on several jobs during a day, any one of which may be either day work or piece work. A separate tabulating card job ticket is prepared for each job. These tickets contain information in addition to that discussed below which is pertinent to labor distributions and job costs, but is not mentioned in the following discussion since it is unnecessary in the computing routine.

On all cards home department, labor class (day work or piece work), and hours worked are entered. On a piece work card the timekeeper also enters pieces produced and standard hours per 100 pieces. If an employee worked overtime, even on a job which he began on his regular shift, a separate card is prepared for the overtime. This card contains actual hours worked on overtime.

All cards for the day are punched and verified. They are then sorted by department charged (this department is not used in the computing run). The purpose of this is to punch burden percent in the cards on the MCRP, since burden percent varies with the department charged. Burden percent is then punched in each card from a master file.

The cards are sorted by clock number and home department. On the MCRP the employee rate is punched in each card. This rate is used for extending both day work and piece work. A blank card which will be used as a summary card is then inserted by the collator following each employee. The cards are now ready for the 120. The computation is as follows:

1. The cards must be checked to be certain that all cards in a group contain the same department and clock number. Note that department contains one digit of alpha.

2. If an employee is working on day work, the computation of his labor is merely hours worked times rate, rounded. This computation must also be done when he is working on piece work, since he is guaranteed at least the amount of his hours worked times rate on each job.

3. On each job ticket involving piece work, pieces produced are multiplied by standard hours to determine the standard hours produced. This figure, multiplied by rate and rounded, is the piece work pay.

4. On all piece work cards, piece work pay must be compared with day work pay. The greater of the two is the labor amount to be punched in the card. If day work is greater, the amount of the difference, called make-up pay, as well as "0" in column 90 is to be punched in the card. This is a sorting control used to separate cards with make-up so that they may be analyzed.

5. Labor amount is multiplied by burden percent to obtain the burden. Burden is rounded.

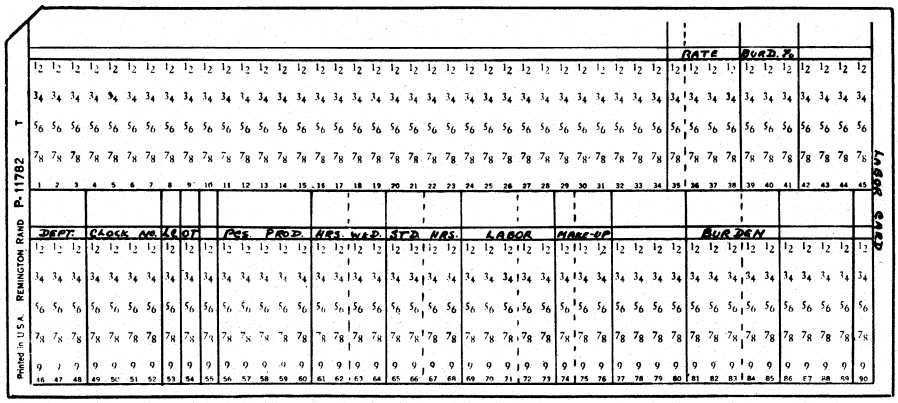
6. Accumulate total hours worked. This includes hours from both regular job tickets and overtime job tickets.

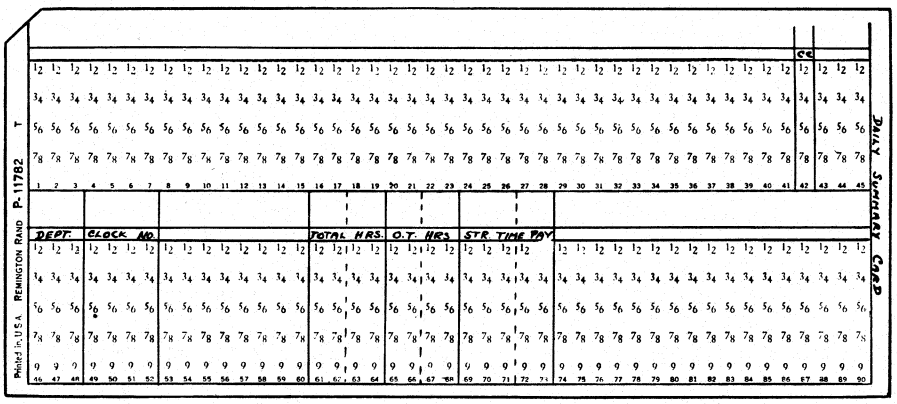
7. Accumulate overtime hours. This is a total of hours from overtime cards, not overtime premium hours (1/2 of overtime hours).

8. Accumulate total labor (straight time pay)

When a summary card enters the computer, all values which are to be punched in it have been calculated. The following information, which will be used in computing net pay, (problem 7), is to be punched in each card: department, clock number, total hours, overtime hours, straight time pay. The summary cards are to be sorted by the computer.

E-2 Card Forms:





E-3 Sense

	Columns	Description
Labor Cards:	46-48	Home department (column 48 is alpha)
	49-52	Clock number
	35-38	Rate
	39-41	Burden percent
	53	Labor Class 5 - day work 6 - piece work
	54	"Zero" if overtime card
	56-60	Pieces produced (not hundred pieces)
	61-64	Hours worked
	65-68	Standard hours per 100 pieces
Summary card:	None	

E-4 Punch

	Columns	Description
Labor Cards:	69-73	Labor
	74-76	Make-up pay
	81-85	Burden
	90	"Zero" if day work is greater than piece work
Summary Card:	46-48	Department
	49-52	Clock number
	42	"3" in all cards
	61-64	Total hours
	65-68	Overtime hours
	69-73	Straight time pay

PROGRAM CHART NO. 5

APPLICATION: LABOR EXTENSION & PAYROLL SUMMARY

UNIVAC 60 & 120 PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand TABULATING MACHINES

TABLE OF FACTORS

SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE
N1	+	L	DEPT., CLOCK NO.
N2	+	L	RATE
N3	+	L	BURDEN %
N4	+	L	PIECES PRODUCED
N5	+	L	HOURS WORKED
N6	+	L	STANDARD HOURS PER PIECE
N7			
N8			
N9			
N10			
N11			
N12			
N13			
N14			
N15			
N16			
N17			
N18			
N19			
N20			
N21			
N22			
N23			
N24			
N25			
N26			
N27			
N28			
N29			
N30			
N31			
N32			
N33			
N34			
N35	+		.005
N36	+		0.000

CARD DESCRIPTION L = LABOR CARD S = SUMMARY CARD

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 5/4	S2 4/3	S3 3/2	S4	S5	S6 3/2	S7 3/2	S8 3/2	S9 3/2	S10 3/2	S11 3/2	S12 4/3	NEXT —	STEP +
1	L,S	DEPT., CLOCK NO. (PCH)	+	N1	−	DEPT., CLOCK NO (ST.)	+	S12	TEST − S/C, ERROR (+ MAY BE OR, 1ST DET.)	+ −	S1	1												3	2
2	L	DEPT., CLOCK NO. (ST.)	+	S12	−	DEPT., CLOCK NO (PCH)	+	N1	TEST − 1ST DETAIL, ERROR (+ OR S/C)	+ −	S1	2												4	6
3	S ERR.	ZERO	+	N36	-	DEPT., CLOCK NO (PCH)	+	N1	TEST − ERROR (+ S/C)	+ −	S1	3												40	21
4	1ST DET.	ZERO	+	N36	-	DEPT., CLOCK NO (ST)	+	S12	TEST − ERROR (+ 1ST DETAIL)	+ −	S1	4												40	5
5	1ST DET.	DEPT., CLOCK NO. (PCH)	+	N1	+	ZERO	+	N36	STORE DEPT., CLOCK NO	+	S12											(5)		-	6
6	L	RATE	+	N2	X	HOURS WORKED	+	N5	DAY WORK PAY (NR)	+	S1	6												-	7
7	L	DAY WORK PAY (NR)	+	S1	+	.005	+	N35	DAY WORK PAY (RD)	+	S11										(7)			-	COM T1-1
8	L	STANDARD HOURS PER PIECE	+	N6	X	PIECES PRODUCED	+	N4	STANDARD HOURS PROD	+	S1	8												-	9
9	L	STANDARD HOURS PROD	+	S1	X	RATE	+	N2	PIECE WORK PAY (NR)	+	S2		9											-	10
10	L	PIECE WORK PAY (NR)	+	S2	+	.005	+	N35	PIECE WORK PAY (RD)	+	S3			10										-	11
11	L	PIECE WORK PAY	+	S3	−	DAY WORK PAY	+	S11	PIECE WORK GREATER (+) − MAKE UP PAY	+ −	S10									(10)				PS1	12
12	L	PIECE WORK PAY	+	S3	+	ZERO	+	N36	TRANSFER TO LABOR	+	S11											(12)		-	13
13	L	BURDEN %	+	N3	X	LABOR	+	S11	BURDEN (NR)	+	S1	13												-	14
14	L	BURDEN (NR)	+	S1	+	.005	+	N35	BURDEN (RD)	+	S9								(14)					-	15
15	L	Σ HOURS WORKED	+	S8	+	HOURS WORKED	+	N5	NEW Σ HOURS WORKED	+	S1	15												-	16
16	L	NEW Σ HOURS WORKED	+	S1	+	ZERO	+	N36	NEW Σ HOURS WORKED	+	S8							(16)						-	COM T6
17	L	Σ OVERTIME HOURS	+	S7	+	OVERTIME HOURS	+	N5	NEW Σ OVERTIME HOURS	+	S1	17												-	18
18	L	NEW Σ OVERTIME HOURS	+	S1	+	ZERO	+	N36	NEW Σ OVERTIME HOURS	+	S7							(18)						-	19
19	L	Σ LABOR	+	S6	+	LABOR	+	S11	NEW Σ LABOR	+	S1	19												-	20
20	L	NEW Σ LABOR	+	S1	+	ZERO	+	N36	NEW Σ LABOR	+	S6						(20)							-	SET 1
21	S	ZERO	+	N36	-	Σ LABOR	+	S6	STRAIGHT TIME PAY	−	S11											(21)		SET 2	-
22																									
23																									
24																									
25																									
26																									
27																									
28																									
29																									
30																									
31																									
32																									
33																									
34																									
35																									
36																									
37																									
38																									
39																									
40	ERR	ZERO	+	N36	÷	ZERO	+	N36	ERROR STEP	+	S1	40													

SYM	CARD FIELD TITLE	NEG. COL	NEG. POS	OUTPUT PUNCHING 10	9	8	7	6	5	4	3	2	1	DEC LOC	SET 1	SET 2	CLEAR	SYM
S1														5/4			X	S1
S2														4/3			X	S2
S3														3/2			X	S3
S4																		S4
S5																		S5
S6														3/2			X	S6
S7	OVERTIME HOURS									65	66	▲67	68	3/2		X	X	S7
S8	TOTAL HOURS									61	62	▲63	64	3/2		X	X	S8
S9	BURDEN								81	82	83	▲84	85	3/2	X		X	S9
S10	MAKE-UP PAY	90	0								74	▲75	76	3/2	CUM T4		X	S10
S11	LABOR / STRAIGHT TIME PAY	42	3							69	70	71	▲72	73	3/2	COM T1-2	X	S11
S12	DEPT. CLOCK NO							46	47	ALPHA 48	48	48	49	▲50	51	52	4/3	X	X	S12

FROM	TO
START	ST. 1
SET 1	TRIP
SET 2	SORT 1
SORT 1	CLEAR
SORT 2	
CLEAR	TRIP
P.S. I	ST. 13
P.S. II	
P.S. III	
P.S. IV	

REPRODUCE CONTROLS	COL.	POS.
REP		
SKIP		
SET HOLD		
SEC REP		

	0 ÷ 0	N ÷ 0
STOP	X	X
SORT		

RESTART	
STEP	X
PROGRAM	

S1-1369 RRBH 12456

SELECTOR CHART NO.: 5

APPLICATION: LABOR EXTENSION + PAYROLL SUMMARY

UNIVAC 60 & 120

PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand

TABULATING MACHINES

ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD-F1, F2, ETC: CONTROL - C1, C2, ETC.) DECIMAL LOCATION, NEGATIVE CONTROL AND TRANSFER LINE. INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	10	9	8	7	6	5	4	3	2	1
N1	F1				4/3		46	47	48	48	48	49	50	51	52
N2	F2				4/3							35	36	37	38
N3	F3				3/2								39	40	41
N4	F4				1/0						56	57	58	59	60
N5	F5				3/2							61	62	63	64
N6	F6				5/4							65	66	67	68
N7															
N8															
N9															
N10															
N11															
N12															
N13															
N14															
N15															
N16															
N17															
N18															

ELEM. DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															
N22															
N23															
N24															
N25															
N26															
N27															
N28															
N29															
N30															
N31															
N32															
N33															
N34															
N35					4/3								0	0	5
N36					4/3							0	0	0	0

ACCUMULATOR INPUTS

INDICATE: TOP LINE - TRANSFER LINE (F1, F2, ETC.) BOTTOM LINE - CARD COLUMNS

SYM.	10	9	8	7	6	5	4	3	2	1
A1		F1	F1	F1	F1	F1	F1	F1	F1	F1
		46	47	48	48	48	49	50	51	52
A2							F2	F2	F2	F2
							35	36	37	38
A3								F3	F3	F3
								39	40	41
A4						F4	F4	F4	F4	F4
						56	57	58	59	60
A5							F5	F5	F5	F5
							61	62	63	64
A6							F6	F6	F6	F6
							65	66	67	68
A7										
A8										
A9										
A10										
A11										
A12										

REPRODUCE

FROM COLUMN	
TO COLUMN	
CONTROL	

SELECTOR CONTROL CHART

SYMBOLS: SELECTOR . . . T1-1, T1-2, T1-3, T1-4, T2, ETC. COMMON . . . COM NON SELECT . . N.S.
SELECT . . . SEL. BRANCH . . . BR. PROCESS . . . PR.
VALUE 1 . . . V1 VALUE 2 . . . V2 RESULT . . . RES.
ELEMENTS . . . N1, N2, N3, ETC. STORAGE . . . S1, S2, S3, ETC. PROGRAM SELECTOR . . . P.S.1, P.S.2, ETC.

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
11-1					
11-2					
11-3					
11-4					
12					
13-1					
13-2					
13-3					
13-4					
14					
15-1					
15-2					
15-3					
15-4					
16					
17-1					
17-2					
17-3					
17-4					
18					
19-1					
19-2					
19-3					
19-4					
31-1					
31-2					
31-3					
31-4					
32					
33-1					
33-2					
33-3					
33-4					
34					
35-1					
35-2					
35-3					
35-4					
36					
37-1					
37-2					
37-3					
37-4					
38					
39-1					
39-2					
39-3					
39-4					

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1	C1	5/53	COM. T2	+ BR. ST. 7	—
1-2			SET 1	SET S 11	SET 2
1-3					
1-4					
2	C2	9/53	ST. 8	SEL. T1-1	ST. 13
3-1					
3-2					
3-3					
3-4					
4		PS 1	SET 1	SET S 10	—
5-1					
5-2					
5-3					
5-4					
6	C6	0/54 (O.T.)	ST. 17	+ BR. ST. 16	ST. 19
7-1					
7-2					
7-3					
7-4					
8					
9-1					
9-2					
9-3					
9-4					
21-1					
21-2					
21-3					
21-4					
22					
23-1					
23-2					
23-3					
23-4					
24					
25-1					
25-2					
25-3					
25-4					
26					
27-1					
27-2					
27-3					
27-4					
28					
29-1					
29-2					
29-3					
29-4					

S1-1370 RRBH 12456

E-5. PROGRAM COMMENTS:

1. Elements:

Department and clock number should be treated as one element. Department has one column of alpha, and this column must be wired into the accumulator in three accumulator columns to prevent the stopping of the computer by an input check. This results in nine columns of identification. Since the designation is to be tested in S1, which has a 5/4 decimal location, department and clock number must be given a 4/3 or 5/4 decimal location so that any possible result will align itself in S1 without hanging up the computer.

Although no decimal location is shown on the card form for burden percent, it must be assigned one at input to obtain the correct result of the multiplication. For example, to multiply by 75%, .75 is actually used.

Standard hours are given per hundred pieces, while pieces produced are an actual count, not the number of hundreds of pieces. The most simple way of adjusting the values is to assign either pieces produced a decimal location of 3/2, or standard hours a 5/4 decimal location.

2. Storage:

Storage S12 is assigned a 4/3 decimal location so that department and clock number, with a 4/3 decimal location, may be stored in it.

3. Program:

Step 1: All cards are directed to the first step of the program. Although one column of department is alpha, this does not affect the program since S12 is always cleared by a preceding summary card. A summary card will always branch minus, along with errors. All labor cards which are correct and first detail cards will branch plus, but must be tested further with the values reversed.

Step 2: When the values in step 1 are reversed, correct labor cards will branch on the plus side and are directed to the labor card routine. Both first detail cards and errors will branch on the minus side. These must be tested further to determine which they are, and are directed to step 4.

Step 3: Cards which branched on the minus side of step 1 must be errors or summary cards. The summary cards are separated on the plus branching of the step, while the errors are routed to step 40.

Step 4: This step determines whether the cards from the minus branching of step 2 are errors or first detail cards. Error cards are routed to step 40 to stop the computer, while first detail cards are sent to step 5 for the storing of the new department and clock number.

Step 5: Department and clock number are stored. The first detail cards are then routed to step 6, which is same step to which all other correct detail cards were sent.

Steps 6 and 7: Rate is multiplied by hours worked to compute day work pay, which is then rounded. This must be done for piece work jobs as well as day work jobs, since the employee is guaranteed at least this amount for each job. The plus

branching step of 7 is wired to the common of T1-1, which is picked up by a 5 in column 53. The 5 should be present in all cards, while in addition a 9 is present in piece work cards. The non-select of T1-1 is not wired, so the computer will stop if neither a 5 nor a 6 is punched. The select of T1-1 is wired to the common of T2, picked up by a 9 in column 53. If the selector is non-select only a 5 is punched and the card is a day work card. The routine for computing piece work is therefore by-passed, and the program continues with step 13. If the selector is select, the card is a piece work card and the card is routed to step 8.

Step 8: Pieces produced are multiplied by the standard hours per piece to obtain standard hours produced. S1 is given a 5/4 decimal location because of this step. The result could have four places following the decimal, and since it is to be used as a value in a further multiplication, all four places are needed to prevent the loss or gain of a penny in the multiplication.

Steps 9 and 10: Standard hours produced are multiplied by rate to obtain piece work pay, which is rounded.

Step 11: Day work pay is subtracted from piece work pay to determine which is the greater. If the two are equal, the card will branch on the plus side with the piece work cards. This is the correct branching for an equal condition, since a zero in column 90 is to be punched only if there is make-up pay. The plus branching is directed to step 12, while the minus branching is wired to PS 1.

A program select should be impulsed to "remember" that day work is greater than piece work. PS 1 picks up T4. T4 is used to determine whether or not to set make-up pay in the punching dies. If the result of step 11 is minus, the amount of the make-up pay is in S10 as a minus value. The negative sign is wired to punch the zero in column 90. If T4 is select, S10 will be wired to set make-up pay, while on the non-select side nothing will be wired. S10 and the zero in column 90 will therefore be set only when day work is greater than piece work.

Note that it is possible to add another step to punch the zero in 90. A constant value could be transferred into a storage, and wired to punch the zero. In such a case, however, all other piece work cards as well as day work cards would have to be routed to this step, adding zero instead of a constant value to zero. This would clear the storage if the employee made out on the job which is being computed while he did not make out on the preceding job. Otherwise the constant from the preceding card would punch in the current card, unless the setting of the storage is controlled by the selector controlling the setting of make-up pay.

If the day work pay is greater than piece work pay, the amount of the labor is already in S11. The program therefore continues with step 13.

Step 12: If piece work pay is greater than day work pay, piece work pay must be transferred as labor to S11.

Steps 13 and 14: Burden is computed by multiplying labor times burden percent and rounding.

Steps 15 and 16: Hours worked are accumulated. All cards, including overtime cards, are routed through these steps since the accumulation is to be total hours worked. The plus branching of step 16 is routed to the common of T6. T6 is picked up by a zero in column 54, punched in an overtime card. If the selector is select the card is an overtime card and must go to steps 17 and 18 for the accumulating of overtime hours. If it is non-select, these steps may be skipped and the program continues with step 19.

Steps 17 and 18: Overtime hours are accumulated. The same element is used as in accumulating total hours, but the accumulation is done only on overtime cards.

Steps 19 and 20: Labor is accumulated from all cards. The total labor for the day becomes the employee's straight time pay.

Set 1: Labor, burden and make-up pay, if any, are set in the punching dies. The setting of labor, S11, is wired to the common of T1-2, picked up by a 5 in column 53 which is present in all labor cards. The select is wired to set 1. Since labor and straight time pay are to be punched in the same card columns, the setting of S11 must be controlled through a selector.

Make-up pay, S10, is wired to the common of T4, which is picked up by PS 1. PS 1 was impulsed when day work pay was greater than piece work pay. Although there is always a value in S10, it is set only when day work is greater than piece work. The negative sign of S10 is always present when day work is greater, and is used to punch the necessary control hole.

Following set 1, the labor cards are tripped out.

Step 21: A summary card is identified at the plus branching of step 3, and is routed to step 21. The sum of the labor is transferred to S11, which is wired to the columns in which straight time pay is punched. This is done by subtracting the sum of the labor from zero to make S11 negative. The negative sign of S11 is used to punch the 3 in column 42 which is to be punched in all summary cards. Note that it would be possible to add another step transferring a constant value of 3 to an available storage for punching. It would also be possible to transfer the .005 to a storage, and wire the 5 to punch a 3 in column 42.

Set 2: Department and clock number, straight time pay, total hours, and overtime hours, are wired to set. Straight time pay is wired to the common of T1-2, and on the non-select side (no 5 is present in column 53) to set 2. It would also have been possible to impulse PS 2 during the summary card routine prior to the set 2 instruction, and use a selector controlled by PS 2 to govern the setting of S11 (select to set 2, non-select to set 1.)

Sort 1: The sort instruction may be given any time during the summary card routine prior to trip.

Clear: A clear instruction must be given to the summary card. During the clear instruction, the following storages must be cleared:

S6 - Accumulation of labor.

S7 - Accumulation of overtime hours

S8 - Accumulation of hours

S12- Department and clock number (so that S12 will be clear to identify the first card of the next group)

The other storages may as well be cleared.

F. PROBLEM 6 - BILLING (CROSSFOOTING)

F-1 Statement of Problem

This problem involves a crossfooting of a large number of small items, which is frequently found in a billing application in the clothing or shoe industries. Any one item may be ordered in several sizes, and the size spread must be shown on the invoice. The example given is that of a shoe manufacturer.

Detail cards punched with style number, width, and two prices are kept in a tub file. A card is pulled for each style and width ordered. The price at which the card is to be extended depends upon the customer who orders it.

Master name, address, and shipping instruction cards are also pulled from tub files. There may be four or five cards in each heading set. The first card contains a control hole indicating the price at which the customers' item cards are extended.

The master cards and detail cards are sent to a keypunch operator. An order card is punched with information about the order, including customer number and order number. This will be the first card of a group. The name and address cards follow the order card, with no additional punching. Into each detail card the punch operator punches customer number and the size spread - that is, in the appropriate columns she punches the quantity of that size ordered. If any shoes of size 12 or over are ordered, all those of size 12 or over are punched in a second card for the same style and width. This is because these sizes sell at a different price from the smaller sizes. In actual practice, order number would be repeat-punched into the detail cards; however, for the purposes of this problem the order number will be reproduced on the computer into the detail cards from the order cards. The cards are now ready for the computer. The calculations are:

All Cards

Test customer number.

Order Card

Order number is to be reproduced from the order card into the detail cards.

Name and Address Cards

The first name card indicates the price at which the detail cards are to be extended. It also contains salesman number, which is to be reproduced into the detail cards. The remaining name and address cards are to be tripped through with no calculation or punching.

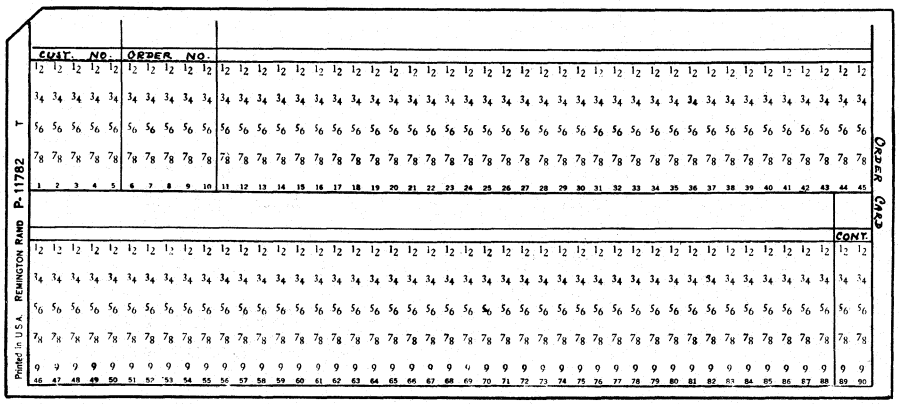
Detail Cards

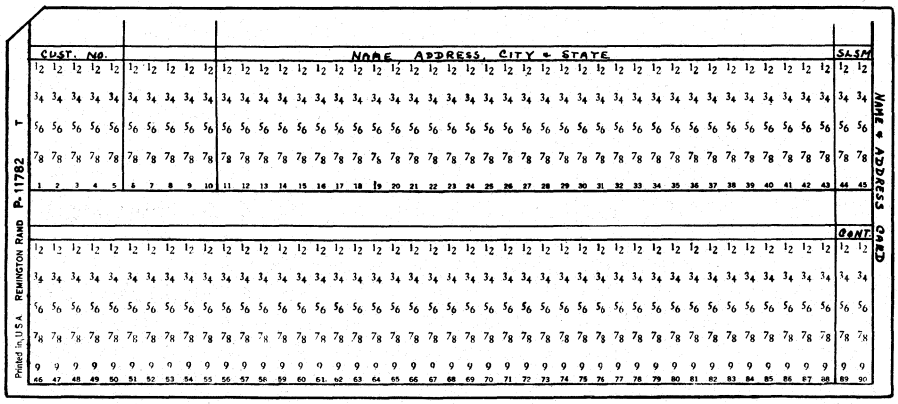
Accumulate (crossfoot) the sizes punched in the card to determine total quantity ordered. If the shoes are size 12 or over, $1.00 is added to the price, A or B, which the customer is to pay. Quantity is then multiplied by price to determine the total amount on the cards.

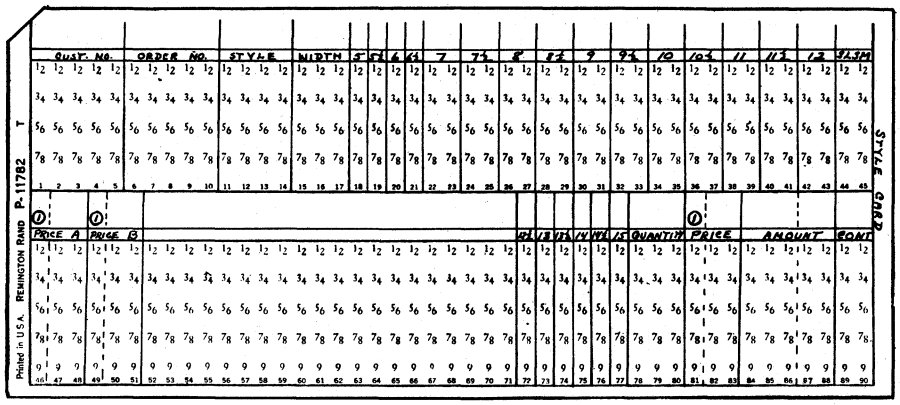
Note:

Although it is possible to store order number and salesman for punching, one of the purposes of this problem is to include the programming of the reproduce function. Therefore they are to be reproduced, not stored.

F-2 Card Forms







F-3 Sense

	Columns	Description
Order Card:	89, pos. 0	
	90, pos. 0	
	1- 5	Customer number
	6-10	Order number
First name card:	90, pos. 0	
	90, pos. 3	
	90, pos. 7	
	89, pos. 3	Use price A; if not present, use price B
	1- 5	Customer number
	44-45	Salesman number
Remaining masters:	90, pos. 0	
	90, pos. 5	
	1- 5	Customer number
Style cards:	90, pos. 1	
	89, pos. 9	If card contains size 12 and over
	18-43, 72-77	Size distribution
	46-48	Price A (0 in 46 is over-capacity)
	49-51	Price B (0 in 49 is over-capacity)

F-4 Punch

	Columns	Description
Style Card:	6-10	Order number
	44-45	Salesman number
	78-80	Total quantity
	81-83	Price (0 in 81 is over-capacity)
	84-88	Amount

PROGRAM CHART NO. 6

APPLICATION: BILLING (CROSSFOOTING)

UNIVAC 60 & 120

PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand

TABULATING MACHINES

TABLE OF FACTORS

SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE
N1	+	1,2,3	CUSTOMER NO.
N2	+	3	SIZES 5, 5½, 6
N3	+	3	SIZES 6½, 7, 7½
N4	+	3	SIZES 8, 8½, 9
N5	+	3	SIZES 9½, 10, 10½
N6	+	3	SIZES 11, 11½, 12
N7	+	3	SIZES 12½, 13, 13½
N8	+	3	SIZES 14, 14½, 15
N9	+	3	PRICE A
N10	+	3	PRICE B
N11			
N12			
N13			
N14			
N15			
N16			
N17			
N18			
N19			
N20			
N21			
N22			
N23			
N24			
N25			
N26			
N27			
N28			
N29			
N30			
N31			
N32			
N33	+		1.00
N34	+		.001001001
N35			
N36	+		0.000

CARD DESCRIPTION 1 = ORDER CARD 2 = NAME & ADDRESS CARD 3 = STYLE CARD

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 1/0	S2 1/0	S3 4/3	S4 4/3	S5 6/5	S6	S7	S8	S9	S10	S11	S12 1/0	NEXT STEP −	NEXT STEP +
1	1,2,3	CUST. NO. (PCH)	+	N1	−	CUST. NO (ST)	+	S12	TEST − ERROR + MAY BE OK, ORDER	+ −	S1	1													COM T1-1
2	2,3	CUST. NO. (ST)	+	S12	−	CUST NO (PCH)	+	N1	TEST − ERROR + OK	+ −	S1	2												40	COM T2
3	1	CUST. NO. (PCH)	+	N1	+	ZERO	+	N36	STORE CUST. NO.	+	S12												(3)	−	TRIP
4	3	SIZES 5, 5½, 6	+	N2	+	SIZES 6½, 7, 7½	+	N3	Σ SIZES (1)	+	S1	4												−	5
5	3	Σ SIZES (1)	+	S1	+	SIZES 8, 8½, 9	+	N4	Σ SIZES (2)	+	S2		5											−	6
6	3	Σ SIZES (2)	+	S2	+	SIZES 9½, 10, 10½	+	N5	Σ SIZES (3)	+	S1	6												−	7
7	3	Σ SIZES (3)	+	S1	+	SIZES 11, 11½, 12	+	N6	Σ SIZES (4)	+	S2		7											−	COM. T3-2
8	3	Σ SIZES (4)	+	S2	+	SIZES 12½, 13, 13½	+	N7	Σ SIZES (5)	+	S1	8												−	9
9	3	Σ SIZES (5)	+	S1	+	SIZES 14, 14½, 15	+	N8	Σ SIZES (6)	+	S2		9											−	10
10	3	Σ SIZES (6)	+	S2	X	.001001001	+	N34	TOTAL (XXXXXX.XXX)	+	S3			10										−	11
11	3	TOTAL (XXXXXX.XXX)	+	S3	+	ZERO	+	N36	FIRST 6 DIGITS (XXXXXX)	+	S1	11												−	12
12	3	TOTAL (XXXXXX.XXX)	+	S3	−	FIRST 6 DIGITS (XXXXXX)	+	S1	QUANTITY (XXX)	+	S4				(12)									−	13
13	3	PRICE	+	COM T2-2	+	1.00/ZERO	+	COM T3-3	STORE ADJ. PRICE	+	S3			(13)										−	14
14	3	ADJ. PRICE	+	S3	X	QUANTITY (XXX)	+	S4	AMOUNT	+	S5					(14)								−	SET
40	ERR	ZERO	+	N36	÷	ZERO	+	N36	ERROR STEP	+	S1	40													

SYM	CARD FIELD TITLE	NEG. COL	NEG. POS	OUTPUT PUNCHING 10	9	8	7	6	5	4	3	2	1	DEC LOC	SET 1	SET 2	CLEAR	SYM
S1														1/0			X	S1
S2														1/0			X	S2
S3	PRICE (COL 81 - OVER CAPACITY)								81	81	82	83		4/3	X		X	S3
S4	QUANTITY											78	79 80	4/3	X		X	S4
S5	AMOUNT								84	85	86	87	88	6/5	X		X	S5
S6																		S6
S7																		S7
S8																		S8
S9																		S9
S10																		S10
S11																		S11
S12														1/0				S12

FROM	TO
START	ST. 1
SET 1	CLEAR
SET 2	
SORT 1	
SORT 2	
CLEAR	TRIP
P.S. I	
P.S. II	
P.S. III	
P.S. IV	

REPRODUCE CONTROLS	COL.	POS.
REP	C1, C8	
SKIP	90	0
SET HOLD		
SEC REP	90	7

	0 ÷ 0	N ÷ 0
STOP	X	X
SORT		

RESTART	
STEP	X
PROGRAM	

S1-1369 RRBH 12456

SELECTOR CHART NO.: 6

APPLICATION: BILLING (CROSSFOOTING)

UNIVAC 60 & 120
PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand
TABULATING MACHINES

ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION. NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD-F1, F2, ETC: CONTROL - C1, C2, ETC.) DECIMAL LOCATION. NEGATIVE CONTROL AND TRANSFER LINE. INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N1	F1				1/0						1	2	3	4	5
N2	F2				1/0				18			19			20
N3	F3				1/0				21		22	23		24	25
N4	F4				1/0			26	27		28	29		30	31
N5	F5				1/0			32	33		34	35		36	37
N6	F6				1/0		38		39		40	41		42	43
N7	F7				1/0				72			73			74
N8	F8				1/0				75			76			77
N9	F9				3/2							46 o/c	46	47	48
N10	F10				3/2							49 o/c	49	50	51
N11															
N12															
N13															
N14															
N15															
N16															
N17															
N18															

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															
N22															
N23															
N24															
N25															
N26															
N27															
N28															
N29															
N30															
N31															
N32															
N33					3/2								1	0	0
N34					9/8		0	0	1	0	0	1	0	0	1
N35															
N36					4/3							0	0	0	0

ACCUMULATOR INPUTS

TOP LINE - TRANSFER LINE (F1, F2, ETC.) INDICATE:
BOTTOM LINE - CARD COLUMNS

SYM.	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
A1						F1	F1	F1	F1	F1
						1	2	3	4	5
A2				F2			F2			F2
				18			19			20
A3				F3		F3	F3		F3	F3
				21		22	23		24	25
A4			F4	F4		F4	F4		F4	F4
			26	27		28	29		30	31
A5			F5	F5		F5	F5		F5	F5
			32	33		34	35		36	37
A6			F6	F6		F6	F6		F6	F6
			38	39		40	41		42	43
A7				F7			F7			F7
				72			73			74
A8				F8			F8			F8
				75			76			77
A9							F9	F9	F9	F9
							46	46	47	48
A10							F10	F10	F10	F10
							49	49	50	51
A11										
A12										

REPRODUCE

FROM COLUMN	6	7	8	9	10	44	45
TO COLUMN	6	7	8	9	10	44	45
CONTROL	RH	RH	RH	RH	RH	SH	SH

SELECTOR CONTROL CHART

SYMBOLS: SELECTOR . . . T1-1, T1-2, T1-3, T1-4, T2, ETC.
COMMON . . . COM
NON SELECT . . N.S.
SELECT . . . SEL.
BRANCH . . . BR.
PROCESS . . . PR.
VALUE 1 . . . V1
VALUE 2 . . . V2
RESULT . . . RES.
ELEMENTS . . . N1, N2, N3, ETC.
STORAGE . . . S1, S2, S3, ETC.
PROGRAM SELECTOR . . . P.S.1, P.S.2, ETC.

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1	C1	0/89	ST. 3	+ BR. ST. 1	ST. 2
1-2			C7 (REPRODUCE)	C8 (REPRODUCE)	—
1-3			—	COM. T 5-1	SELECTOR HOLD
1-4		(ORDER)			
2	C2	3/90 5/90 (MASTER)	TRIP	+ BR. ST. 2	COM. T 4
3-1	C3	9/89	ST. 7	SEL. T 4	ST. 4
3-2			ST. 8	+ BR. ST. 7	ST. 10
3-3			N33	V2 ST. 13	N36
3-4		(SIZE 12 +)			
4	C4	1/90 (STYLE)	COM. T 3-1	N.S. T2	—
5-1	C5	3/89	P.U. T5	COM. T 1-3	—
5-2		SEL. T5-1	N9	V1 ST. 13	N 10
5-3					
5-4		(PRICE A)			
6					
7-1					
7-2					
7-3					
7-4					
8					
9-1					
9-2					
9-3					
9-4					
21-1					
21-2					
21-3					
21-4					
22					
23-1					
23-2					
23-3					
23-4					
24					
25-1					
25-2					
25-3					
25-4					
26					
27-1					
27-2					
27-3					
27-4					
28					
29-1					
29-2					
29-3					
29-4					

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
11-1					
11-2					
11-3					
11-4					
12					
13-1					
13-2					
13-3					
13-4					
14					
15-1					
15-2					
15-3					
15-4					
16					
17-1					
17-2					
17-3					
17-4					
18					
19-1					
19-2					
19-3					
19-4					
31-1					
31-2					
31-3					
31-4					
32					
33-1					
33-2					
33-3					
33-4					
34					
35-1					
35-2					
35-3					
35-4					
36					
37-1					
37-2					
37-3					
37-4					
38					
39-1					
39-2					
39-3					
39-4					

F-5 Program Comments

1. Elements

The maximum total quantity is three digits, as indicated on the card form. Since three digits must therefore be allowed for each size when it is placed in accumulator input, a maximum of three sizes may be combined as one element.

2. Reproduce

The columns which are to be reproduced are indicated beneath the element section on the selector chart. Order number is to be reproduced from the first card, so the control is indicated as RH - reproduce, hold. Salesman number is to be reproduced from the second card, so it is indicated as SH - secondary reproduce, hold.

The only control hole peculiar to the order card is a zero in column 89, which therefore must be used as a reproduce control. However, since it should also identify the card, the zero is wired to pick up a selector, T1, rather than directly to reproduce. The two reproduce hubs are then wired by means of control transfer lines to the common and select sides of T1-2. Whenever a zero in column 89 is present, the two hubs will be connected through the select of T1-2; when the control hole is not present, no connection will be made.

A zero in column 90 is present in all cards except style cards. This, therefore, is the control hole which should be used as a skip control. The reproduce hole, zero in column 89, cannot be used since skipping is to occur in cards other than the order card. A skip control must be wired in the name and address cards to prevent the punching of the current order number in cards which are reused for many orders.

Salesman number is to be picked up from the first master card, while the reproduce control is punched in the order card. Therefore, secondary reproduce is used. Either a 3 or 7 in column 90, both of which are peculiar to the first name card, may be used as a secondary reproduce control.

3. Program

Step 1: All cards are routed to step 1. Since there are no blank summary cards, all cards which branch minus are errors. Cards which branch plus are routed to the common of T1-1, which is picked up by a zero in column 90, present in an order card. Order cards, which are the first of a group, are directed to step 3, while all others are wired to step 2 for the completion of the testing routine.

Step 2: The testing of customer number is completed. Any card branching minus is an error. The plus branching is wired to the common of T2. T2 is picked up by either a 3 in column 90 (first name card) or a 5 in column 90 (all other name cards). Note that it would be possible to use two selectors, one picked up by a 3 and the other picked up by a 5. Nothing would be gained, however, since the select of both of them would be wired to trip, and if neither is select the wiring is to the common of a selector picked up by a 1 in column 90 (style card).

Although a first master card is tripped out, selector T5 will be picked up if a 3 in column 89 is present, indicating that price A is to be used in the detail cards. It will be held select by the selector hold wiring (non-select of T1-3, through commons of T1-3 and T5-1, and the select of T5-1.) Selector T5 will always be dropped out when an order card enters the computer, since T1 becomes select and the power emitting from the pick-up of T5 can no longer reach selector hold ground. The non-select of selector T2, picked up by a master card, is wired to the common of T4, picked up by a 1 in column 90, present in a style card. If T4 is non-select, the computer will hang up for lack of a correct code. If the selector is select, the card is a style card.

The first steps in a style card involve the accumulation of total quantity. The problem states that size 12 and over are punched in a separate card. Therefore, the select of T4 is wired to T3-1, which is picked up by a 9 in column 89, present in those cards punched with size 12 and over. If the selector is non-select, the program continues with step 4, which accumulates the sizes beginning with size 5. The select of T3-1 is wired to step 7, which accumulates size 12. It would, of course, be possible to route all cards through all the accumulating steps.

Step 3: Customer number is stored from the order card. The card is then tripped out.

Steps 4-7: (Sizes 5-11 1/2) From the non-select of T3-1, the cards which are punched with sizes 5-11 1/2 are routed to step 4. In steps 4-7 the sizes are accumulated. Although size 12 is wired as part of element N6, which is V2 of step 7, in these cards size 12 is never punched. At the end of step 7, S2 contains three totals, as follows:

Sizes	Storage columns
1) 5, 6 1/2, 8, 9 1/2, 11	9-7
2) 5 1/2, 7, 8 1/2, 10, 11 1/2	6-4
3) 6, 7 1/2, 9, 10 1/2	3-1

The plus branching of step 7 is wired to the common of T3-2, which is select when a card containing sizes 12-15 is in the computer. Since step 7 will be used for sizes 12-15 as well as 5-11 1/2, the plus branching should be controlled so that cards containing sizes 5-11 1/2 do not go through the remaining accumulating steps unnecessarily. The non-select of T3-2 is therefore wired to step 10.

Step 7-9: (Sizes 12-15) From the select of T3-1 the card containing sizes 12-15 are routed to step 7. Since the storages used in accumulating are to be cleared on each card, storage S1 is zero. Step 7 therefore becomes zero + size 12. The exit of step 7 is wired to the common of T3-2, and on the select side to step 8, which continues the accumulation of sizes 12 1/2 - 15. At the end of step 9, S2 contains three totals as follows:

	Sizes	Storage columns
1)	12 1/2, 14	9-7
2)	13, 14 1/2	6-4
3)	12, 13 1/2, 15	3-1

Step 9 is wired to step 10, as were the cards containing sizes 5 - 11 1/2.

Step 10: The value in S2, which is either the sum (in three totals) of sizes 5 - 11 1/2 or 12 - 15 is multiplied by .001001001. This crossfoots the three totals into one, with the total as the first three digits following the decimal point. It is therefore placed in a 4/3 storage, which will drop off all unnecessary places to the right.

Step 11: This value is transferred to a 1/0 storage to obtain the digits preceding the decimal point.

Step 12: The digits preceding the decimal point are subtracted from the full value, leaving only the total quantity. This must be placed in a 4/3 storage. Since the decimal location should actually follow the quantity rather than precede it, an additional step in which quantity is multiplied by 1000 would correctly adjust the decimal location. However, by making allowance for the unadjusted decimal point when wiring output punching, the multiplication can be avoided.

Step 13: Price must be stored, since it is to be punched in each card. V1 of step 13 is wired to the common of T5-2. T5 is select if the name card contained in 3 in column 89, indicating that price A is to be used. The select of T5-2 is wired to N9, price A, while the non-select, is wired to N10, price B.

V2 of step 13 is wired to the common of T3-3. T3 is select if the card contains sizes 12-15, and non-select if the sizes are 5 - 11 1/2. The select of T3-3 is wired to N33, which will add 1.00 to the price of the shoes. The non-select is wired to zero, so that the price will be stored as it is punched in the card.

Step 14: The adjusted price is multiplied by quantity. Quantity as stored is a decimal value (.xxx) although it should be a whole number (xxx.). Since price has two places following the decimal, the result must be placed in a 6/5 storage. From the card form it is determined that there will be no more than five significant digits. Therefore the amount is placed in storage as .xxxxx. No rounding is necessary, since if the true value of quantity were used, there would be only two places following the decimal.

Set 1: Price, quantity, and amount are set in the punching dies. Quantity and amount are wired to card columns making allowance for the decimal value of quantity.

Clear: If all cards go through all accumulating steps, it is not necessary to use clear. However, with the program as shown it is necessary. Cards for sizes 12 - 15 are routed from step 2 to step 7. If S1 is not cleared on every card, it might have the total of sizes 5 - 10 1/2 from a preceding card.

Although storages S1-S5 are wired to clear on every card, S12 cannot be wired to clear. To do so would clear the designation which has been stored from card to card.

G. PROBLEM 7 - GROSS TO NET PAYROLL

G-1 Statement of Problem

This problem is a continuation of problem 5, involving the weekly calculation of net pay using the daily summary cards created in problem 5.

The daily summary cards for each employee are sorted with the following cards: a previous year-to-date card, a card containing total hours worked which will be used as a gross pay card, a detail card for each deduction, a card into which deductions are summarized, and a net pay card. As a result of the computer run, two cards will be created to write the payroll register and payroll checks. One card will contain earning information, while the other will contain deduction information. The new to date values will also be punched in the card containing earning information.

From the week's attendance card on which an employee punches in and out, a card is key-punched containing department, clock number, shift, and total hours for the week. This card becomes the week's gross pay card. A deck of cards containing department and clock number is reproduced from this deck, creating a deck of cards to be used as deduction summary cards. A second deck of cards is reproduced containing department and clock number which will be used as net pay cards.

A file of cards is maintained for each standard deduction - charity, insurance, and credit union. Each week corrections and adjustments are made to the file or files which are to be deducted in that payroll. Cards for miscellaneous deductions such as purchases are key-punched. The detail deduction cards are then sorted together by department and clock number.

The gross pay cards are used in two runs on the collator to separate previous year-to-date cards and the detail deduction cards for employees who did not work during the week. Since the previous year-to-date card contains name and tax class, these may be reproduced into the gross pay card during the run.

All cards to be used in computing net pay are then sorted together by department and clock number in the following sequence:

Previous year-to-date
Daily summary
Gross pay
Detail deduction
Deduction summary
Net pay

The computation is as follows:

All cards

Test department and clock number.

Programming note: Use only three steps, including the storing of department and clock number, for the test.

Previous year-to-date card

This card is last week's gross pay card into which were punched gross pay to date, FICA to date, and withholding tax to date, including the values from last week's payroll. After the completion of the payroll the card code of "1", indicating gross pay card, was changed to "2", indicating year-to-date card.

1. Store the following to date figures: FICA, withholding tax, gross pay.

2. Store department and clock number.

Daily summary cards

1. Accumulate the following values: hours, overtime hours, straight time pay.

Programming note: To have sufficient steps, reuse the accumulating steps so that only two of the forty steps are required.

Gross pay card

1. Balance attendance hours with the total of hours from the daily summary cards. Sort the gross pay cards if the hours are not equal. Use total hours from the summary cards for calculations in succeeding steps.

2. Compute shift premium (hours × shift premium rate) and round. Second shift premium is .07, third shift is .13.

3. Add straight time pay and shift pay to obtain a preliminary gross.

4. Compute overtime for those employees who worked overtime. To do this, divide preliminary gross by total hours to obtain the average rate. Multiply the average rate by the overtime premium hours (1/2 the overtime hours) to obtain overtime pay. Do not round either division, but carry average rate to four decimal points and overtime premium hours to three decimal points. Round overtime pay.

5. Add overtime pay to preliminary gross to obtain gross pay.

6. Compute withholding tax. To do this, multiply tax class by $13.00 to obtain the exempt income. Subtract exempt income from gross pay, and multiply the resulting taxable income by 18%. Round withholding tax. Note that for some employees the exempt income will exceed gross income.

7. Compute FICA. To do this, subtract FICA to date from $84.00, the maximum FICA to be deducted in one year, to determine the remaining FICA which must be paid. Gross pay is then multiplied by 2% and rounded, to determine the tentative FICA (the FICA to be deducted if the maximum has not been reached). The remaining FICA is compared to the tentative FICA to determine which is greater. If the tentative FICA is greater, the remaining FICA is used as the FICA for the week.

8. Accumulate the following to date values: gross, FICA, withholding tax.

9. Add FICA and withholding tax to obtain the preliminary total deduction amount.

10. Subtract preliminary total deduction amount from gross pay to obtain preliminary net pay.

11. Punch to date values, hours and earnings information. Note that total hours which are punched in columns 56-60 are punched again in columns 61-64. This is for ease of designing the wiring units for the payroll register and payroll checks.

Detail deduction cards

1. Subtract the amount of the deduction from net pay to obtain the new net pay. If there is insufficient pay to cover the amount of the deduction, sort and trip the card immediately.

2. Add the deduction amount to the total deduction amount to obtain the new total deduction amount.

3. Since there may be more than one deduction of the same type (several miscellaneous deductions, or insurance from a preceding week which was not deducted as well as this week's insurance), the types of deductions must be accumulated. Each type of deduction must be placed in the storage from which it will be punched.

Deduction summary card

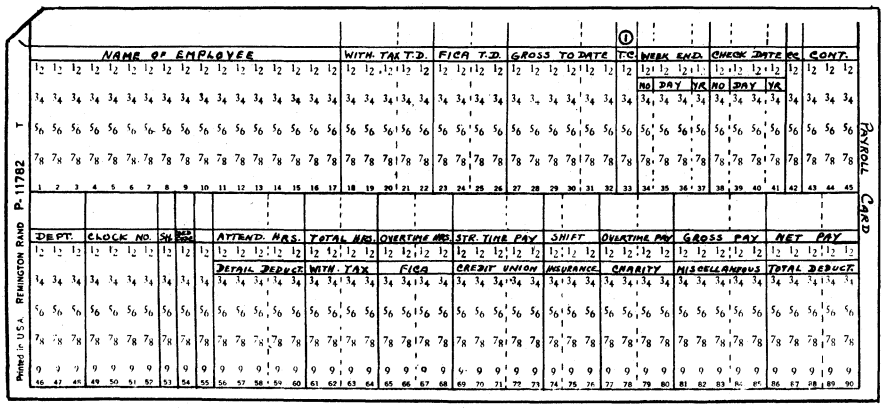
Punch deductions, taxes, and total deductions.

Net pay card

Punch net pay in the same columns in which total deduction amount was punched.

After the cards have been computed on the 120, they are sorted by card code. If any adjustments must be made because attendance hours did not balance with accumulated hours, these are taken care of. The net pay cards are then compared with the gross pay cards on the MCRP, reproducing net pay amount into the gross pay cards. The gross pay cards and deduction summary cards are then sorted together for tabulating the payroll register and payroll checks.

G-2 Card form:



G-3 Sense:

	Columns	Description
Previous year to date	42, pos. 2	Card code
	46-48	Department (col. 48 is alpha.)
	49-52	Clock number
	18-22	Withholding tax to date
	23-26	FICA to date
	27-32	Gross to date
Daily summary	42, pos. 3	Card code
	46-48	Department (col. 48 is alpha.)
	49-52	Clock number
	61-64	Total hours
	65-68	Overtime hours
	69-73	Straight time pay
Gross pay	42, pos. 1	Card code
	46-48	Department (col. 48 is alpha.)
	49-52	Clock number
	33	Tax class (0 = overcapacity)
	53	Shift { 1 = 1st shift; 2 = 2nd shift; 3 = 3rd shift }
	57-60	Attendance hours
Detail deductions	42, pos. 5	Card code
	46-48	Department (col. 48 is alpha.)
	49-52	Clock number
	54	Deduction code 1 = Miscellaneous 3 = Insurance 5 = Credit union 7 = Charity
	56-60	Deduction amount

	Columns	Description
Deduction summary	42, pos. 6	Card code
	46-48	Department (col. 48 is alpha.)
	49-52	Clock number
Net pay	42, pos. 7	Card code
	46-48	Department (col. 48 is alpha.)
	49-52	Clock number

G-4 Punch:

	Columns	Description
Previous year to date	None	
Daily summary	None	
Gross pay	18-22	Withholding tax to date
	23-26	FICA to date
	27-32	Gross to date
	61-64	Total hours
	65-68	Overtime hours
	69-73	Straight time pay
	74-76	Shift premium
	77-80	Overtime pay
	81-85	Gross pay
Detail deduction	None	
Deduction summary	61-64	Withholding tax
	65-68	FICA
	69-73	Credit union
	74-76	Insurance
	77-80	Charity
	81-85	Miscellaneous
	86-90	Total deductions
Net pay	86-90	Net pay

PROGRAM CHART NO. 7

APPLICATION: GROSS TO NET PAYROLL

UNIVAC 60 & 120

PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand

TABULATING MACHINES

TABLE OF FACTORS

SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE	SYM	SIGN	CARD NO.	CARD FIELD TITLE OR CONSTANT VALUE
N1	+	1,2,3 5,6,7	DEPT., CLOCK NO. (46-52)	N13				N25			
N2	+	1	TAX CLASS (33)	N14				N26			
N3	+	1/5	ATTEND. HRS./DETAIL DEDUCTION (56-60)	N15				N27			
N4	+	2	WITHHOLDING TAX TO DATE (18-22)	N16				N28	+		.02
N5	+	2	FICA TO DATE (23-26)	N17				N29	+		84.00
N6	+	2	GROSS TO DATE (27-32)	N18				N30	+		.18
N7	+	3	TOTAL HOURS (61-64)	N19				N31	+		13.00
N8	+	3	OVERTIME HOURS (65-68)	N20				N32	+		2.
N9	+	3	STRAIGHT TIME PAY (69-73)	N21				N33	+		.13
N10				N22				N34	+		.07
N11				N23				N35	+		.005
N12				N24				N36	+		0

CARD DESCRIPTION 2= PREVIOUS YEAR TO DATE 1= GROSS PAY 3= DAILY SUMMARY 5= DETAIL DEDUCTION 6= DEDUCTION SUMMARY 7= NET PAY

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1 4/3	S2 4/3	S3 5/4	S4 3/2	S5 3/2	S6 3/2	S7 3/2	S8 3/2	S9 3/2	S10 3/2	S11 3/2	S12 4/3	NEXT STEP −	NEXT STEP +
1	1,2,3 5,6,7	DEPT., CLOCK NO. (PCH)	+	N1	−	DEPT., CLOCK NO. (ST)	+	S12	DIFFERENCE	±	S1	1												COM T1-1	COM T1-1
2	1,3, 5,6,7	DIFFERENCE	±	S1	÷	ZERO	+	N36	TEST 0 ÷ 0 OK; N ÷ 0 ERROR	±	S2		2											−	COM T1-2
3	2	WITH. TAX TO DATE	+	N4	+	ZERO	+	N36	STORE WITH. TAX TO DATE	+	S11											(3)		−	4
4	2	FICA TO DATE	+	N5	+	ZERO	+	N36	STORE FICA TO DATE	+	S10										(4)			−	5
5	2	GROSS TO DATE	+	N6	+	ZERO	+	N36	STORE GROSS TO DATE	+	S9									(5)				−	6
6	2,1	DEPT, CLOCK NO.	+	N1	+	ZERO	+	N36	STORE DEPT., CLOCK NO.	+	S12												(6)	−	TRIP
7	3	Σ HOURS/O.T. HRS./STP	+	COM T7-1	+	HOURS/O.T. HRS/STP	+	COM T7-2	NEW Σ HOURS/OT HRS/STP	+	S1	7												−	8
8	3	NEW Σ HOURS/O.T. HRS/STP	+	S1	+	ZERO	+	N36	NEW Σ HOURS/OT HRS/STP	+	COM T7-1						(8)	(8)	(8)					−	COM T7-3
9	1	ATTEND. HOURS	+	N3	−	Σ HOURS	+	S8	TEST + MAY BE OK − ERROR	±	S1	9												SORT 1	10
10	1	Σ HOURS	+	S8	−	ATTEND. HOURS	+	N3	TEST + OK − ERROR	±	S1	10												SORT 1	COM. T12
11	1	SHIFT PREM. RATE	+	COM T13-2	X	HOURS	+	S8	SHIFT PREM. (NR)	+	S1	11												−	12
12	1	SHIFT PREM (NR)	+	S1	+	.005	+	N35	SHIFT PREM (RD)	+	S5					(12)								−	13
13	1	STRAIGHT TIME PAY	+	S6	+	SHIFT PREM.	+	S5	PRELIMINARY GROSS	+	S1	13												−	14
14	1	ZERO	+	N36	−	OVERTIME HOURS	+	S7	TEST + NO OVERTIME − OVERTIME	±	S2		14											15	19
15	1	PRELIMINARY GROSS	+	S1	÷	HOURS	+	S8	AVERAGE RATE	+	S3			15										−	16
16	1	OVERTIME HOURS	+	S7	÷	2	+	N32	OVERTIME PREM. HOURS	+	S12												16	−	17
17	1	AVERAGE RATE	+	S3	X	OVERTIME PREM. HOURS	+	S12	OVERTIME PAY (NR)	+	S2		17											−	18
18	1	OVERTIME PAY (NR)	+	S2	+	.005	+	N35	OVERTIME PAY (RD)	+	S4				(18)									−	19
19	1	PRELIMINARY GROSS	+	S1	+	OVERTIME PAY	+	S4	GROSS PAY	+	S3			(19)										−	SET 1
20	1	TAX CLASS	+	N2	X	13.00	+	N31	EXEMPT INCOME	+	S1	20												−	21
21	1	GROSS PAY	+	S3	−	EXEMPT INCOME	+	S1	TAXABLE INCOME	±	S2		21											PS3	22
22	1	ZERO/18%	+	COM T16	X	TAXABLE INCOME	±	S2	WITHHOLDING TAX (NR)	+	S1	22												−	23
23	1	WITHHOLDING TAX (NR)	+	S1	+	.005	+	N35	WITHHOLDING TAX (RD)	+	S8								(23)					−	24
24	1	84.00	+	N29	−	FICA TO DATE	+	S10	REMAINING FICA TO PAY	+	S1	24												−	25
25	1	.02	+	N28	X	GROSS PAY	+	S3	TENT. FICA (NR)	+	S2		25											−	26
26	1	TENT. FICA (NR)	+	S2	+	.005	+	N35	TENT. FICA (RD)	+	S7							(26)						−	27
27	1	REMAINING FICA	+	S1	−	TENT. FICA	+	S7	TEST + PAY TENT. FICA − PAY REMAIN. FICA	±	S2		27											28	29
28	1	REMAINING FICA	+	S1	+	ZERO	+	N36	FICA	+	S7							(28)						−	29
29	1	GROSS TO DATE	+	S9	+	GROSS PAY	+	S3	NEW GROSS TO DATE	+	S2		(29)											−	30
30	1	FICA TO DATE	+	S10	+	FICA	+	S7	NEW FICA TO DATE	+	S9									(30)				−	31
31	1	WITHHOLDING TAX TO DATE	+	S11	+	WITHHOLDING TAX	+	S8	NEW WITH. TAX TO DATE	+	S10										(31)			−	SET 2
32	1	FICA	+	S7	+	WITHHOLDING TAX	+	S8	PRELIM. TOTAL DEDUCTS	+	S11											(32)		−	33
33	1	GROSS PAY	+	S3	−	PRELIM. TOTAL DEDUCTS	+	S11	PRELIM. NET PAY	+	S10										(33)			−	CLEAR
34	5	NET PAY	+	S10	−	DEDUCTION AMOUNT	+	N3	NEW NET PAY	±	S1	34												SORT 2	35
35	5	NEW NET PAY	+	S1	+	ZERO	+	N36	NEW NET PAY	+	S10										(35)			−	36
36	5	TOTAL DEDUCTS	+	S11	+	DEDUCTION AMOUNT	+	N3	NEW TOTAL DEDUCTS.	+	S1	36												−	37
37	5	NEW TOTAL DEDUCTS.	+	S1	+	ZERO	+	N36	NEW TOTAL DEDUCTS	+	S11											(37)		−	38
38	5	Σ DETAIL DEDUCTS	+	COM T11-1	+	DEDUCTION AMOUNT	+	N3	NEW Σ DETAIL DEDUCTS	+	S1	38												−	39
39	5	NEW Σ DETAIL DEDUCTS	+	S1	+	ZERO	+	N36	NEW Σ DETAIL DEDUCTS.	+	COM T11-1			(39)	(39)	(39)	(39)							−	TRIP
40	7	NET PAY	+	S10	+	ZERO	+	N36	TRANSFER NET PAY	+	S11											(40)		−	SET 1

SYM	CARD FIELD TITLE	NEG. COL	NEG. POS	10	9	8	7	6	5	4	3	2	1	DEC LOC	SET 1	SET 2	CLEAR	SYM
s1														4/3				s1
s2	GROSS TO DATE						27	28	29	30	31	32		4/3		X		s2
s3	GROSS PAY / MISCELLANEOUS						81	82	83	84	85			5/4	X		X	s3
s4	OVERTIME PAY / CHARITY									77	78	79	80	3/2	X		X	s4
s5	SHIFT PREMIUM / INSURANCE										74	75	76	3/2	X		X	s5
s6	STRAIGHT TIME PAY / CREDIT UNION								69	70	71	72	73	3/2	X		X	s6
s7	OVERTIME HOURS / FICA									65	66	67	68	3/2	X		COM T3-1	s7
s8	TOTAL HOURS / WITHHOLDING TAX									61	62	63	64	3/2	X		COM T3-1	s8
s9	FICA TO DATE									23	24	25	26	3/2		X		s9
s10	WITHHOLDING TAX TO DATE								18	19	20	21	22	3/2		X		s10
s11	TOTAL DEDUCTIONS / NET PAY								86	87	88	89	90	3/2	COM T1-3			s11
s12														4/3				s12

FROM	TO
START	ST. 1
SET 1	COM. T1-4
SET 2	ST. 32
SORT 1	COM. T12
SORT 2	TRIP
CLEAR	COM T3-2
P.S. I	ST. 7
P.S. II	ST. 7
P.S. III	ST. 22
P.S. IV	

REPRODUCE CONTROLS	COL.	POS.
REP		
SKIP		
SET HOLD		
SEC REP		

	0 ÷ 0	N ÷ 0
STOP		X
SORT		

RESTART	
STEP	X
PROGRAM	

S1-1369 RRBH 12456

SELECTOR CHART NO.: 7

APPLICATION: GROSS TO NET PAYROLL

UNIVAC 60 & 120
PUNCHED-CARD ELECTRONIC COMPUTERS

Remington Rand
TABULATING MACHINES

ELEMENTS

CONSTANT FACTORS: INDICATE DECIMAL LOCATION, NEGATIVE CONTROL AND VALUE IN DESIRED ACCUMULATOR COLUMNS.
CARD FACTORS: INDICATE TRANSFER LINE (FIELD-F1, F2, ETC: CONTROL - C1, C2, ETC.) DECIMAL LOCATION, NEGATIVE CONTROL AND TRANSFER LINE. INDICATE CARD COLUMNS IN ACCUMULATOR INPUTS ON RIGHT.

ELEM DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATOR COLUMNS 10	9	8	7	6	5	4	3	2	1
N1	F1				4/3		46	47	48	48	48	49	50	51	52
N2	F2				1/0									9c 33	33
N3	F3				3/2						56	57	58	59	60
N4	F4				3/2						18	19	20	21	22
N5	F5				3/2							23	24	25	26
N6	F6				3/2					27	28	29	30	31	32
N7	F7				3/2							61	62	63	64
N8	F8				3/2							65	66	67	68
N9	F9				3/2						69	70	71	72	73
N10															
N11															
N12															
N13															
N14															
N15															
N16															
N17															
N18															

ELEM. DESIG	TO	NEG. CONT. COL	NEG. CONT. POS	NEG. CONT. TRF LINE	DEC LOC	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
N19															
N20															
N21															
N22															
N23															
N24															
N25															
N26															
N27															
N28					3/2									0	2
N29					3/2							8	4	0	0
N30					3/2									1	8
N31					3/2							1	3	0	0
N32					1/0										2
N33					3/2									1	3
N34					3/2									0	7
N35					4/3								0	0	5
N36					4/3							0	0	0	0

ACCUMULATOR INPUTS

TOP LINE - TRANSFER LINE (F1, F2, ETC.) INDICATE;
BOTTOM LINE - CARD COLUMNS

SYM.	ACCUMULATED COLUMNS 10	9	8	7	6	5	4	3	2	1
A1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1
	46	47	48	48	48	49	50	51	52	
A2								F2	F2	
								33	33	
A3						F3	F3	F3	F3	F3
						56	57	58	59	60
A4						F4	F4	F4	F4	F4
						18	19	20	21	22
A5						F5	F5	F5	F5	F5
						23	24	25	26	
A6					F6	F6	F6	F6	F6	F6
					27	28	29	30	31	32
A7						F7	F7	F7	F7	
						61	62	63	64	
A8						F8	F8	F8	F8	
						65	66	67	68	
A9						F9	F9	F9	F9	F9
						69	70	71	72	73
A10										
A11										
A12										

REPRODUCE

FROM COLUMN	
TO COLUMN	
CONTROL	

SELECTOR CONTROL CHART

SYMBOLS: SELECTOR . . . T1-1, T1-2, T1-3, T1-4, T2, ETC. COMMON . . . COM NON SELECT . . N.S.
SELECT . . . SEL. BRANCH . . . BR. PROCESS . . PR.
VALUE 1 . . . V1 VALUE 2 . . . V2 RESULT . . . RES.
ELEMENTS . . . N1, N2, N3, ETC. STORAGE . . . S1, S2, S3, ETC. PROGRAM SELECTOR . . . P.S.1, P.S.2, ETC.

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
11-1	C11	1/54	S 3	V1 ST. 38; R ST. 39	COM. T 15-1
11-2					
11-3					
11-4		(MISC.)			
12	C12	1/53 (1ST & 2ND SHIFT)	COM. T 13-1	SORT 1; + BR. ST. 10	COM. T 14
13-1	C13	9/53	ST. 11	SEL. T 12	ST. 13
13-2			N34	V1, ST. 11	N33
13-3					
13-4		(2ND SHIFT)			
14	C14	3/53 (3RD SHIFT)	ST. 11	N.S. T 12	—
15-1	C15	3/54	S 5	N.S. T 11-1	COM. T 17-1
15-2					
15-3					
15-4		(INSURANCE)			
16		PS 3	N36	V1, ST. 22	N30
17-1	C17	5/54	S 6	N.S. T 15-1	COM. T 18
17-2					
17-3					
17-4		(CREDIT UNION)			
18	C18	7/54 (CHARITY)	S 4	N.S. T 17-1	—
19-1					
19-2					
19-3					
19-4					
31-1					
31-2					
31-3					
31-4					
32					
33-1					
33-2					
33-3					
33-4					
34					
35-1					
35-2					
35-3					
35-4					
36					
37-1					
37-2					
37-3					
37-4					
38					
39-1					
39-2					
39-3					
39-4					

SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
1-1	C1	1/42 (GROSS) (PREV. Y.T.D.)	COM. T 5-1	+, - BR. ST. 1	ST. 2
1-2			ST. 9	+ BR. ST. 2	COM. T 2
1-3			—	SET S11	SET 1 PWR.
1-4			ST. 20	FROM SET 1	CLEAR
2	C2	3/42 (DAILY 5%)	ST. 7	N.S. T 1-2	COM. T 4
3-1	C1	1/42 (GROSS) (PREV. Y.T.D.)	—	CLEAR S7, S8	CLEAR PWR.
3-2			ST. 6	FROM CLEAR	TRIP
3-3					
3-4					
4	C4	5/42 (DEDUCT)	COM. T 5-2	N.S. T 2	COM. T 6
5-1	C5	9/42 (PREV. Y.T.D.) (DEDUCT 5%)	ST. 3	SEL. T 1-1	ST. 2
5-2			SET 1	SEL. T 4	ST. 34
5-3					
5-4					
6	C6	7/42 (NET PAY)	ST. 40	N.S. T 4	—
7-1		PS1	COM. T 9-1	V1 ST. 7; R ST. 8	S8 (Σ HRS)
7-2			COM. T 9-2	V2 ST. 7	N7 (HRS)
7-3			COM. T 9-3	+ BR. ST. 8	PS1
7-4					
8					
9-1		PS2	S6 (Σ S.T.P.)	SEL. T 7-1	S7 (Σ O.T. HRS.)
9-2			N9 (S.T.P.)	SEL. T 7-2	N8 (O.T. HRS.)
9-3			TRIP	SEL. T 7-3	PS 2
9-4					
21-1					
21-2					
21-3					
21-4					
22					
23-1					
23-2					
23-3					
23-4					
24					
25-1					
25-2					
25-3					
25-4					
26					
27-1					
27-2					
27-3					
27-4					
28					
29-1					
29-2					
29-3					
29-4					

G-5 Program Comments

Step 1: Since this is a long program and all 40 steps may be used easily, without a zero ÷ zero step, this is an excellent program in which to use a two step test for department and clock number. The test shown as steps 1 and 2 of the problem may be used since a card code is present in each card. Both the plus and minus branchings of step 1 are directed to the common of T1-1, since with an alpha column in the department, the first card of a new group may branch either plus or minus.

If selector 1 is select, the card is either a gross pay card or a year-to-date card, so the select of T1-1 is wired to the common of T5-1 to determine whether a 9 is present as well as a 1. If T5 is select, the card is a previous year-to-date card and therefore the first of a group; the select of T5-1 is wired to step 3, which is the first step of the year-to-date card. The non-select of both T1-1 and T5-1 are wired to step 2 for the completion of the test. Note that the point of this test is to determine whether the difference between N1 and S12 is zero or a number, either positive or negative; therefore both the minus and plus branchings are wired to step 2.

Another two step test which might be used in this problem is to follow a zero plus zero selector delay step with a step using N36, zero, as the result of a subtraction. The disadvantage is that a light will not light to show the cause of the computer's hanging up. The same disadvantage occurs if a normal testing routine of reversing the values is used, with the minus branching not wired.

It would be possible to include a zero divided by zero step by reusing a program step. For example, two storing steps may be combined into one by use of a program select, or two similar steps used in different cards may be combined into one by card control. It would also be possible to pick up a program select from the minus branching of the testing steps and use the selector which it picks up to substitute zero for value 2 in a division step normally used for another purpose. In this last case both 0 ÷ 0 and N ÷ 0 would be wired to stop, since there might or might not be a value in the storage used as value 1.

Step 2: All cards except the previous year-to-date cards enter this step. If the difference between N1 and S12 is zero, this step becomes zero divided by zero. The card is correct, and since zero divided by zero is not wired to stop, it is routed to the common of T1-2. The further wiring of T1-2 will be taken up with each type of card. If the difference between N1 and S12 is any digital value, positive or negative, the punched department and clock number are not the same as those stored; this step becomes a number divided by zero. N ÷ 0 is wired to stop the computer, and the N ÷ 0 light will light indicating an error.

Previous year to date card

Steps 3-6: Previous year-to-date cards are routed to step 3. During steps 3-6 department, clock number, and the three to date values are stored. Since it becomes necessary to use S12 as a working storage during the program for the gross pay card, department and clock number are stored as the last step of the routine. Department and clock number must be stored again as the last

step of the gross pay routine, and if this is also the last step of the year-to-date card, use of a selector at the branching of the step can be avoided.

Daily summary cards

The plus branching of step 2 is wired to the common of T1-2. The non-select of T1-2 is wired to the common of T2. If T2, picked up by a 3 in column 42, is select, the card is a daily summary card. The select of T2 is wired to step 7.

Steps 7-8: Steps 7 and 8 are basic accumulating steps. Since the program actually exceeds 40 steps, the two steps are reused twice by means of program selects. This in effect increases the capacity of the computer to 44 steps instead of 40.

Gross pay card

The plus branching of step 2 is wired to the common of T1-2. If T1-2 is select, the card must be a gross pay card, since the previous year-to-date cards, the only other cards with a 1 in column 42, have already been separated. The select of T1-2 is wired to step 9.

Steps 9 and 10: Attendance hours, which are punched in the gross pay card, are compared with hours accumulated from the daily summary cards. If either step 9 or step 10 branches minus, the hours are not the same and the card is wired to sort. Sort 1 is used in both cases, since the next step will be the same whether the card is sorted on step 9 or step 10. From both sort 1 and the plus branching of step 10 the cards are routed to the common of T12 to determine whether the employee is entitled to any shift premium. If T12, picked up by a 1 in column 53, is select, the employee is either first or second shift. The select of T12 is wired to the common of T13-1 to determine whether a 9 is also punched. If T13 is select, the card is second shift and is directed to step 11 for the computation of shift premium. If T13 is non-select, the employee is first shift and is not entitled to shift premium. Since the storage in which shift premium is to be placed has not been used since it was cleared at the end of the preceding employee, and there is thus no danger of a carry-over of information, the steps for the computation of shift premium may be completely skipped. The non-select of T13-1 is therefore wired to step 13.

The non-select of T12, indicating that there is not a 1 punched in column 53, is wired to the common of T14, picked up by a 3 in column 53. If T14 is select, the employee is third shift and is routed to step 11. If T14 is non-select, neither a 1 nor a 3 is punched, and the computer will hang up for lack of an instruction.

Step 11: Hours are multiplied by shift premium rate. Only cards punched with a 2 or 3 in columns 53 enter this step, and they may be differentiated by either a 1, 9, or 3. In this example a 9 is used. If T13-2 is select, the card is second shift and N34, .07, is used as the shift premium rate; If T13-2 is non-select the card is third shift and N33, .13, is used as the shift premium rate.

Step 12: Shift premium is rounded.

Step 13: Shift premium is added to straight time pay to obtain preliminary gross. Cards which do not have shift premium are routed to this step so that straight time pay will be placed in storage as preliminary gross. Since no shift premium has been computed in these cards, straight time pay is added to zero in this step.

Step 14: Overtime hours are subtracted from zero to determine whether the employee worked overtime. Since four steps are involved in the computation of overtime pay, it is well to omit these steps if possible. The minus branching, indicating that there is overtime, is wired to step 15; the plus branching is wired to step 19.

Step 15: Preliminary gross is divided by total hours to determine average rate. Note that all pay except overtime must be included in the preliminary gross. Average rate is carried to four places, and need not be rounded.

Step 16: Overtime hours are divided by 2 to obtain overtime premium hours. Note that it would be possible to multiply them by .5. By placing the result in a 4/3 storage, it becomes unnecessary to round it. Since there are insufficient working storages available with 4/3 decimal locations, it becomes necessary to use S12, in which department and clock number are stored. At the conclusion of the routine for the gross pay cards, they will be routed back to the step which stores department and clock number.

Steps 17 and 18: Average rate is multiplied by overtime premium hours to obtain overtime premium pay, and rounded.

Step 19: Overtime pay is added to preliminary gross to obtain gross pay. Cards which contain no overtime are routed to this step so that in all cases gross pay will be placed in S3.

Set 1: At the end of step 19 gross pay, overtime pay, shift premium, straight time pay, overtime hours, and total hours are all computed and ready to punch. The only one of these values needed in further computation is gross pay, and the storages in which overtime hours and total hours are stored are needed for taxes. By going to set 1 following step 19 these values can be set for punching, and the storages from which they were set be freed for further values. All six of the storages are wired to set on set 1. The storage from which total deductions and net pay will be punched is to be set on a later set 1 instruction; since at this time it contains withholding tax to date, it must be prevented from setting. Therefore the setting of S11 is wired to the common of T1-3, which is select in the gross pay card. The select of T1-3 is not wired, so no setting will occur.

From set 1 in the gross pay card it is necessary to return to the program, step 20, yet in succeeding cards set 1 must be wired to another instruction. Therefore the exit of set 1 is wired to the common of T1-4. The select of T1-4 is wired to step 20.

Step 20: There is more than one method of computing witholding tax. The method shown here requires four steps, but very few elements and selectors.

A second method, requiring only two or three steps but 11 constants and 6 selectors, is shown following the program comments for this problem. Step 20 is the computation of exempt income. Tax class (the number of dependents) is multiplied by $13.00 (the amount of tax exempt income for each dependent each week). It is unnecessary to round this since there will be only two significant places following the decimal point.

Step 21: Exempt income is subtracted from gross pay to determine taxable income. The result of this subtraction can be either plus or minus, since it is possible for a man's exempt income to exceed his gross pay. If the result is negative, PS 3 is impulsed so that a selector may be used in the following step. Both PS 3 and step 21 are wired to step 22.

Step 22: Taxable income, if a plus value, is multiplied by 18% to obtain the non-rounded withholding tax. If taxable income is negative, it must be multiplied by zero to clear the storage in which withholding tax is placed. Since total hours was previously located in the same storage, unless it is cleared the total hours figure will be used as withholding tax in later steps. V1 of step 22 is therefore wired to the common of T16, picked up by PS 3. If T16 is select, N36, zero, is used as V1; if it is non-select N30, .18, is used as V1.

Step 23: Withholding tax is rounded. If there is no withholding tax, the result of this step will still be zero. Withholding tax must be placed in the storage in which total hours were previously located, since they are to punch in the same columns.

Step 24: The computation of FICA on the basis of $4200 gross to date requires more steps than using $84.00 FICA to date. In this problem, therefore, $84.00 is used.

Step 24 is the subtraction of FICA to date from $84.00. The result of this step is always plus, since FICA to date never exceeds $84.00. The value placed in storage is the amount of FICA required to reach $84.00. If an employee has already reached $84.00, the value placed in storage is zero.

Step 25: Gross pay is multiplied by 2% to obtain the tentative FICA - that is, the FICA which will be deducted provided the employee has not reached $84.00 including the current deduction.

Step 26: FICA is rounded into the storage from which it will be punched.

Step 27: The tentative FICA is subtracted from the remaining FICA to pay. If the result is plus, the remaining FICA is greater than the tentative FICA to be deducted this week. The tentative FICA is actual FICA this week, and the program continues with step 29. If the result is minus, the remaining FICA is less than the tentative FICA; therefore the remaining FICA must be transferred to the FICA storage. These cards are directed to step 28.

Step 28: Remaining FICA is transferred to the FICA storage. If an employee has already reached $84.00, zero will be transferred into the storage, clearing it. The program continues with step 29.

Steps 29-31: Gross to date, FICA to date, and withholding tax to date are up-dated with the values from the current week's payroll.

Set 2: The last values to be punched in the gross pay card have now been computed. Since the storages in which the to date figures are located are needed for the computation of total deductions and net pay, the to date values may be set at this time. Set 1 has already been used in this card, so set 2 should be used. From set 2 the program is continued with step 32.

Step 32: FICA and withholding tax are added together to obtain preliminary total deductions.

Step 33: The preliminary total deductions are subtracted from gross pay to obtain a preliminary net.

Clear: Following the completion of the gross pay card, the program must be routed to clear. From the following cards deductions are placed in the same storages that were used for earnings information. If these storages are not cleared before the deduction cards, the earnings information will be accumulated as deduction amounts when the deductions are accumulated and placed in their punching storages. Taxes, which are to punch in the deduction summary card, cannot be cleared at this time, yet they should be cleared on a later clear instruction. Therefore the clearing of S7 and S8 are wired to the common of T3-1, which is select in the gross pay card. The select of T3-1 is not wired, so S7 and S8 will not clear at this time.

The exit of clear is wired to the common of T3-2. In a gross pay card department and clock number must be replaced in S12, while this is unnecessary in other cards. The select of T3-2 is wired to step 6, which stores department and clock number. From step 6 the card is tripped out. Note that if step 6 is not the last step of the previous year-to-date routine, the exit of the step can be wired to a selector to determine whether to trip the gross pay card or continue with the previous year-to-date routine.

Detail deduction cards

The non-select of T2 is wired to the common of T4, picked up by a 5 in column 42. If the selector is select, the card is either a detail deduction or deduction summary card. The select of T4 is wired to the common of T5-2, to determine whether there is a 9 punched. If T5 is non-select, the card is a detail deduction, and is wired to step 34.

Step 34: Deduction amount is punched in columns 56-60, no matter what the type of deduction may be. This amount is subtracted from net pay to obtain the new net pay. If the result of this step is negative, the employee had insufficient pay to cover the amount of the deduction. The card is therefore sorted, indicating that it has not been deducted, and tripped immediately. The net pay amount in S10 has not been changed, and this card enters no further calculations. Note that the detail deduction cards should be in sequence with the one which it is most important to deduct first, followed by the others in the order of their importance.

Step 35: The new net pay must be transferred back to S10 so that it will be in S10 when the next deduction card enters the computer.

Step 36: Deduction amount is added to total deductions to obtain the new total deduction amount.

Step 37: The new total deduction amount is transferred back to S11, so that it will be in S11 when the next deduction card enters the computer.

Steps 38 and 39: Since there may be more than one deduction of the same type (miscellaneous, charity, insurance, or credit union), the detail deductions must be accumulated. The storages in which they are to be accumulated were cleared on the gross pay card. Steps 38 and 39 are normal accumulating steps, except that the storages in which the accumulation takes place are selected. V1 of step 38 and the result of step 39 are wired to the common of T11-1. If T11 is select, the card is a miscellaneous deduction. Since miscellaneous deductions are to be punched in the same columns as gross pay, the select of T11-1 is wired to S3. The non-select of T11-1 is wired to the common of T15-1.

If T15 is select, the card is an insurance card. Since insurance is to be punched in the same columns as shift premium, the select of T15-1 is wired to S5. The non-select of T15-1 is wired to the common of T17-1.

If T17 is select, the card is a credit union card. Credit union is to be punched in the same columns as straight time pay, so the select of T17-1 is wired to S6. The non-select of T17-1 is wired to the common of T18.

If T18 is select, the card is a charity card. Charity is to be punched in the same columns as overtime pay, so the select of T18 is wired to S4. If T18 is non-select, it means that none of the four codes have been punched; the non-select is not wired, so the computer will hang up.

From the plus branching of step 39, the detail deduction card is tripped out of the computer.

Deduction Summary Card

If T4 and T5 are both select, the card in the computer is a deduction summary card. Since no further calculation is required in this card, the select of T5-2 is wired to set 1.

Set 1: Set 1 is used in a deduction summary card because the same storages are to be set as were set on set 1 in the gross pay card, with the addition of S11, total deductions. The setting of S11 is wired to the common of T1-3. T1 is non-select in a deduction summary card, so S11 is set in addition to the other storages.

From set 1 it is advisable to go to clear. Set 1 can be used in the net pay card for setting net pay, but if this is done the other storages which set on set 1 should be clear so that the deduction amounts will not be punched in the net pay cards. It is also essential to clear these storages before the cards for the next employee enter the computer.

Storages S7 and S8 should be cleared although they were not cleared when the clear instruction was given in the gross pay card. The clearing of S7 and S8 is wired to the common of T3-1, which is non-select on a deduction summary card. On the non-select side both storages are wired to clear.

The exit of clear is wired to the common of T3-2. The non-select of T3-2 is wired to trip.

Net Pay Card

The non-select of T4 is wired to the common of T6, which is select when a net pay card is in the computer. The select of T6 is wired to step 40. If T6 is non-select, the card was not punched with a 1, 3, 5, or 7, and is therefore incorrect. The non-select is not wired, so the computer will hang up.

Step 40: Net pay is transferred to the storage from which it will punch.

Set 1: All storages except S11 which are wired to set on set 1 are clear. The setting of S11 is wired to the common of T1-3, which is non-select. The non-select of T1-3 is wired to set 1. Net pay is therefore the only value which will be set in a net pay card. Although it is not necessary to clear any storages following the net pay card, the exit of set 1 is wired to the common of T1-4, and through the non-select side to clear. Since the program for this card is very short, the clear instruction will not reduce the speed of the computer and can do no harm. Therefore instead of wiring the exit of set 1 through another selector so that it will go directly to trip on a net pay card, the net pay card is allowed to clear. This completes the routine for gross to net payroll.

Optional Method of Computing Withholding Tax

As mentioned earlier, it is possible to compute withholding tax in two or three steps rather than four by the use of 11 constants and six selectors.

The first of the steps is the multiplication of the total gross pay, regardless of exemptions, by 18%. This results in a non-rounded withholding tax, with no allowance made for exemptions.

A constant value is created for each tax class. This value is computed by multiplying tax class by $13.00 to obtain the total amount of exempt income. This value is multiplied by 18% to determine the tax allowance for exemptions. .005 is then subtracted from the result. The minus .005 included in the constant value will actually round the withholding tax. Since this constant is subtracted in the second step of the computation from the non-rounded withholding tax, the subtraction of the minus .005 actually adds .005 to the withholding tax.

The computation of the constant value for tax class 3 would be:

$$\begin{aligned} 3 \times 13.00 &= 39.00 \\ .18 \times 39.00 &= 7.02 \\ 7.02 - .005 &= 7.015 \end{aligned}$$

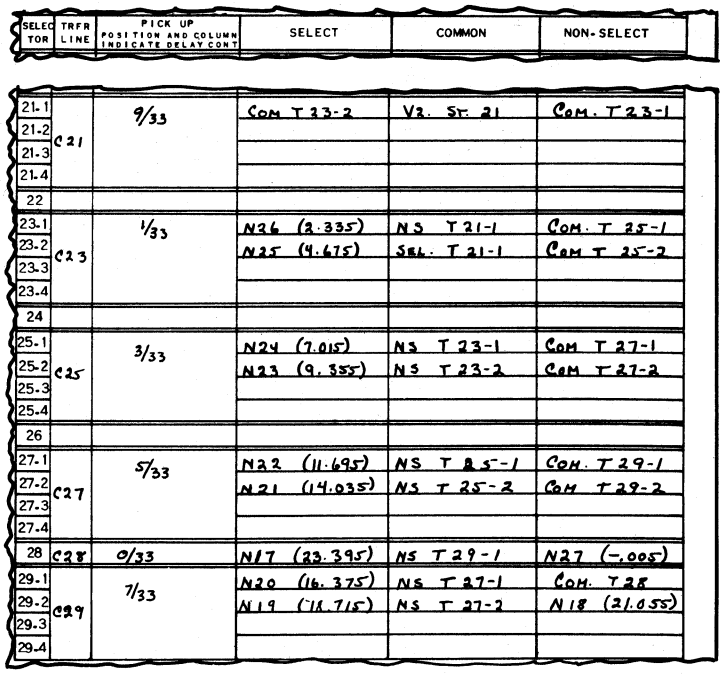
A list of the constant values for the first 10 exemptions is:

Tax Class	Constant
0	-.005
1	2.335
2	4.675
3	7.015
4	9.355
5	11.695
6	14.035
7	16.375
8	18.715
9	21.055
10	23.395

In the step in which the constant value is subtracted from the non-rounded withholding tax, the constant value which will be used is determined by selectors picked up by tax class.

The program for computation of withholding tax in the above problem by this method would be:

STEP NO.	CARD NO.	VALUE 1 DESCRIPTION	SIGN	SYM	PROCESS	VALUE 2 DESCRIPTION	SIGN	SYM	RESULT DESCRIPTION	SIGN	SYM	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	NEXT STEP —	NEXT STEP +
20	1	GROSS PAY	+	S3	X	.18	+	N30	WITHHOLDING TAX (NR)	+	S1	20												-	21
21	1	WITHHOLDING TAX (NR)	+	S1	-	TAX ALLOWANCE FOR EXEMP	±	COM TRAN	WITHHOLDING TAX (RD)	±	S8								(21)					22	23
22	1	ZERO	+	N36	+	ZERO	+	N36	CLEAR S8	+	S8								(23)					-	23



SELECTOR	TRFR LINE	PICK UP POSITION AND COLUMN INDICATE DELAY CONT	SELECT	COMMON	NON-SELECT
21.1	C21	9/33	COM T 23-2	V2. ST. 21	COM. T 23-1
21.2					
21.3					
21.4					
22					
23.1	C23	1/33	N26 (2.335)	NS T 21-1	COM. T 25-1
23.2			N25 (4.675)	SEL. T 21-1	COM T 25-2
23.3					
23.4					
24					
25.1	C25	3/33	N24 (7.015)	NS T 23-1	COM T 27-1
25.2			N23 (9.355)	NS T 23-2	COM T 27-2
25.3					
25.4					
26					
27.1	C27	5/33	N22 (11.695)	NS T 25-1	COM. T 29-1
27.2			N21 (14.035)	NS T 25-2	COM T 29-2
27.3					
27.4					
28	C28	0/33	N17 (23.395)	NS T 29-1	N27 (-.005)
29.1	C29	7/33	N20 (16.375)	NS T 27-1	COM. T 28
29.2			N19 (18.715)	NS T 27-2	N18 (21.055)
29.3					
29.4					

Note the inclusion of a step to clear S8 if the result of step 21 is minus, meaning that an employee's tax exemption exceeds his total tax. This step could be avoided by impulsing a program select on the minus branching of step 21, and using the selector it picks up to prevent the setting of S8 and to substitute zero whenever withholding tax is called upon.

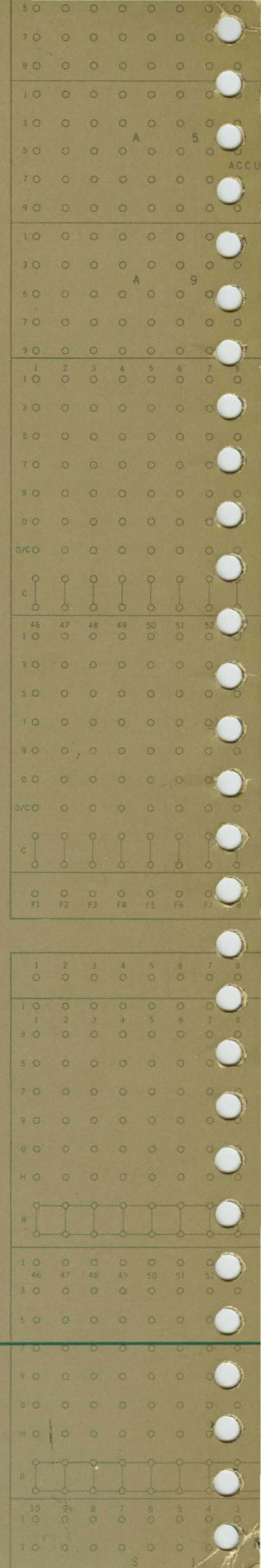
Withholding tax on an income of $90.00 for a man with three exemptions would be computed by the two methods as follows:

4 Step

20) 3 × 13.00 = 39.00
21) 90.00 - 39.00 = 51.00
22) 51.00 × .18 = 9.180
23) 9.180 + .005 = 9.18

2-3 Step

20) 90.00 × .18 = 16.20
21) 16.20 - 7.015 = 9.18



10265953 4

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION
315 FOURTH AVENUE, NEW YORK 10, N. Y.