OPERATING PRINCIPLES
OF THE
UNIVAC® FILE-COMPUTER

ISSUED JANUARY 1, 1956

REMINGTON RAND UNIVAC
DIVISION OF SPERRY RAND CORPORATION

SPTM 4287 REV. 2

OPERATING PRINCIPLES

UNIVAC FILE-COMPUTER

INDEX

ILLUSTRATIONS

SECTION I

INTRODUCTION

INTRODUCTION TO THE UNIVAC FILE-COMPUTER.

The Univac File-Computer System is a medium sized electronic system designed to combine efficiently electronic computing with large capacity internal magnetic drum storage for random access processing of unsorted data. It possesses common language versatility and many types of input-output may be used simultaneously. Input-output units and storage units can be put together as building block units to produce a system satisfying individual requirements. The five basic components of any computer system are input, storage, arithmetic, control and output. The Univac File-Computer has these components but the type, number and capacity of these units in any grouping of equipment is determined by the individual application requirements.

Expanding requirements for better control of the basic business functions have dictated the need for computers with larger and larger internal storage systems. After carefully studying many business problems it was determined that this storage should possess a format for storing data similar to the form of the business transaction. This basic typical form is the individual unit record which is an extension of that common to punched card systems. In payroll we are concerned with the employee. Inventory control problems require the handling of single items successively. Sales analysis, material control, billing and invoicing and virtually every other business problem deal with individual unit records. This format then, has been adapted as the layout for storing data within the system. This layout, together with the user's ability to specify the desired length of this record allows maximum utility of the large storage capacity available in the Univac File-Computer.

An evaluation of the Univac File-Computer's potential in solving problems involving high volume processing of unsorted data can be best made by examining its versatility.

STORAGE VERSATILITY

The large-scale random-access internal memory permits instant reference to stored unit records which include all the control figures and master data required for the complete processing of each input item.

This random-access storage feature permits entry of the input data to the system in the random sequence of its arrival. It can thus eliminate manual steps such as pulling cards from a tub or looking up other types of records. It can also eliminate the need for periodic batch processing, saving the usual delays as well as the machine steps of sorting and merging input data, then re-sorting output data.

On the other hand, the Univac File-Computer will readily handle the complete task of batch processing when this is indicated. By means of magnetic-tape, magnetic-drum or punched-card methods, input data and stored unit records can be sequenced and merged as needed for fast, efficient reference.

INPUT-OUTPUT VERSATILITY.

Another trail-blazing feature is the simultaneous operation of up to 31 universal language input-output units -- any combination of punched-card, magnetic tape, perforated tape, line printer, electric typewriter, ten-key tape printer, and key punch units.

This unusual flexibility opens the way for practically unlimited innovations of procedures to solve particular problems of data processing and reference to stored data. In particular, it permits a direct tie-in with automated recording of the transaction items to be used as input data, and with automated methods for utilizing output data from the system.

PROGRAMMING VERSATILITY.

Processing of each input item can include many computations and logical decisions based on simultaneous reference to both the input data and the stored data (balances, to-date totals, rates or prices, etc.). Output of results can be made immediately -- including stored master descriptions, etc.

As part of the same processing, all stored balances and totals can be brought forward to reflect the item processed -- maintaining complete control figures on a truly current basis.

Several applications can be combined into a single high-speed processing. For example, billing can be combined with inventory control and sales statistics -- requiring only a single processing run of each customer order. Similarly, a production control system that includes both machine loads and operation schedules requires only a single processing for each work order and each job ticket. Complex mathematical problems can be completed in a single pass of a punched-card deck.

The combination of multiple input and expandable programming capacity permits the simultaneous processing of many different types of data -- with each input item processed selectively according to its own requirements.

REPORTING VERSATILITY.

All the data held in random-access storage is instantly available by keyboard inquiry when needed, eliminating any need to search records or wait until records are available.

Condensed reports may be read out selectively from any type of storage, with calculations to extract the most significant figures of current position for management and operating reports. This method eliminates the mental arithmetic and individual judgments or the many extra machine steps usually required for such selective reports.

Another capability is the immediate reporting of any condition requiring supervisory attention, at the instant such a condition develops during normal data-processing runs. Such reports can be printed out directly on an input-output unit located in the supervisory office concerned, and keyboard inquiry can also be made directly from that office for any stored information required.

BUILDING-BLOCK VERSATILITY.

The wide choice of input-output and storage units permits the designing of a system which is truly specific to the particular data-processing needs. Additional units may be added to the system at any time to meet expanded needs.

The flexibility of the Univac File-Computer System also will permit the introduction of future developments in data-processing equipment and techniques without the danger of obsolescence for the complete system.

The following six sections of this manual are designed to combine a detailed description of the computer with the tools and guidance to program effectively problems within the scope of the computer's capacity. Section II describes, in general terms, the components of the Univac File-Computer. Little attention is paid to the details concerning these components as this is covered in later sections. The purpose of this section is to give an overall picture of the system and illustrate how the various elements are tied together. Sections III, IV and V are designed to present a detailed description of the specifications of the Input-output equipment, Storage system, and Arithmetic and control system respectively. Section V concerns itself primarily with the tools for programming. In Section VI illustrative problems are presented together with their solutions. In addition, guidance is given in the preparation of problems for programming. These problems are included to illustrate the various functions of the computer and do not necessarily illustrate the most realistic approach to these simple problems. Section VII however has as its major objective the presentation of realistic approaches to various problems and the most effective use of the computer's commands and logic in arriving at the desired goal.

SECTION II

GENERAL COMPONENTS OF THE UNIVAC FILE-COMPUTER

GENERAL COMPONENTS OF THE UNIVAC FILE-COMPUTER.

INDEX - SECTION II

I. GENERAL

The Univac File-Computer, as all computers, can be analyzed by function or components. These computer functions or requirements are:

A. Input

B. Output

C. Storage

D. Arithmetic

E. Control

All computers require components of one form or another to accomplish these five functions. As an example of this we might analyze a 10-key adding machine in light of these requirements and we would find that each of the functions is fulfilled as follows:

A. Input -- Keyboard digit keys 0 -- 9

B. Output -- Printed tape of input data and totals

C. Storage - The accumulating register which remembers and carries the total forward.

D. Arithmetic -- The accumulating register which adds or subtracts the input data to arrive at a total

E. Control - The motor bar, repeat key, sub-total key and subtract key.

As can be seen by the above analogy all computers contain these five functions in one form or another and, of course, in varying degrees of capacity.

The form these functions take in the Univac File-Computer are listed in this section along with a general explanation of how data is processed through the system. Subsequent sections describe the detail operating characteristics and use of each of the components mentioned.

2. INPUT - OUTPUT (See section III for detail operating characteristics)

The types of input - output devices available with the Univac File-Computer are:

A. 90 Column card sensing punching unit - 150 CPM

B. 90 Column card sensing punching unit - 200 CPM

C. 80 Column card sensing punching unit - 200 CPM

D. Inquiry Keyboard

E. Typewriter with paper tape and plugboard format control

1. Paper tape reader -- 20 characters per second

2. Paper tape punch -- 20 characters per second

F. Paper tape reader -- 200 characters per second

G. Paper tape punch -- 60 characters per second

H. Key Punch - 90 column

I. Magnetic Tape -- Plastic

J. Magnetic Tape -- Metallic - Compatible with Univac

K. High-Speed Printer

Any combination of these devices in any total number from one (1) to thirty one (31) may be included in any one Univac File-Computer System. Since each device is equipped with its own buffer storage, multiple device systems can share the computer through direction of the multiplex function (see section V for detail operation).

3. STORAGE (See Section IV for detailed characteristics and operation)

There are five basic storage sections within the Univac File-Computer, of which one is composed of magnetic cores, while the remaining four types are magnetic drum storages. The five storage sections and their respective access speeds and capacities are:

A. Input-Output Buffer Storages -- Magnetic Cores

These storages act as a translator medium between the input-output device and the computer input-output drum track for the device. The storage may be, dependent on the device, of either 12 or 120 digit capacity. However, even if the buffer is of 12 digit capacity it can handle up to 120 digits by repeated transfers.

B. Input Drum Storages -- Magnetic Drum 12,000 RPM

These storages are used by the computer for obtaining input data from and delivering output data to the input-output buffer storages. There are ten (10) storages of twelve digit capacity each for every input-output device included in the system. The average access time to any input storage is 2.5 milliseconds.

C. Intermediate Storages -- Magnetic Drum 12,000 RPM

This storage section is used to retain constant or intermediate computing results. Either thirty (30) or fifty (50) of these twelve digit storages may be included in a system. Average access to any storage is 2.5 milliseconds.

D. Program Storage -- Magnetic Drum 12,000 RPM

For additional program capacity (over and above the plugboard) additional storages may be included in a system. These storages may be used to store either three address program instructions, results, or constant data. Average access time to any of these storages is 2.5 milliseconds.

E. Large Capacity Storage -- Magnetic Drums 1,750 RPM

Large capacity drum storage is used for storing file reference data, summarized results or volatile file data. Each unit or drum can store up to 180,000 characters of data and a total of ten (10) drums can be included in one system to obtain a total of 1,800,000 characters of storage. The average access time to any set of characters in this storage sections is 17 milliseconds.

External storage in the form of magnetic tapes or cards is also available.

4. ARITHMETIC AND CONTROL (See Section V for detailed operating instructions)

The arithmetic section of the machine provides the necessary circuitry to perform the data processing functions upon the stored information. The functions provided for in the Univac File-Computer are:

A. Addition

B. Subtraction

C. Multiplication

D. Division

E. Comparison - alphabetic and/or numeric

F. Transfer of data - alphabetic and/or numeric

G. Left Zero Elimination - an editing function

H. Channel Search -- an automatic sequential search of the large capacity drums for desired data.

I. Result sign determination and branching (+,-,o)

The control section of the machine provides the means of linking and directing these processes of the arithmetic section to operate in a logical, pre-determined manner upon the stored data.

A. External Programming: Control Panel of 48 non-sequential reusable three address steps.

B. Internal Programming: Stored instructions from hi-speed drum to provide additional three address steps.

Both of these means of control can be utilized in any one system.

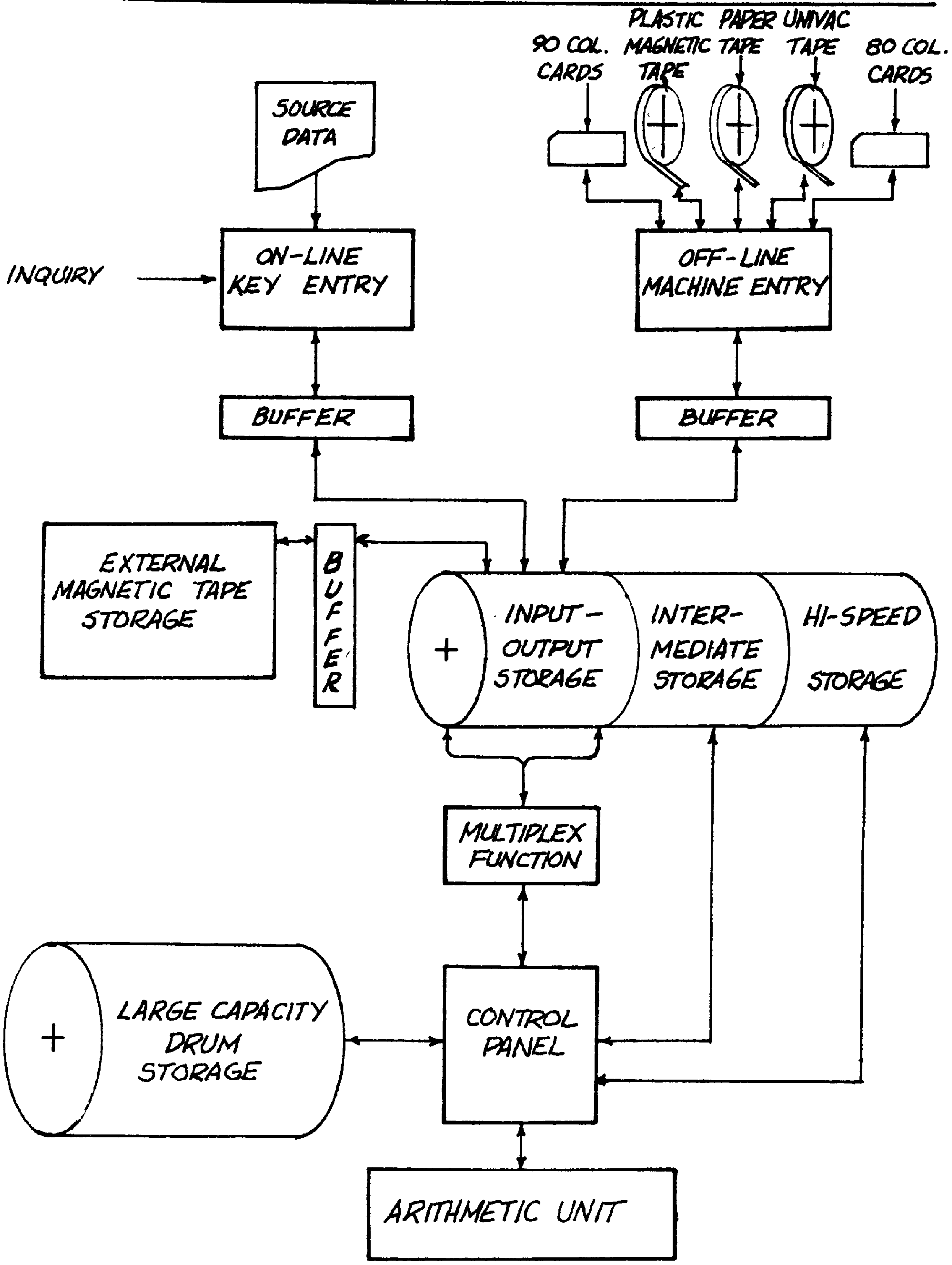
5. GENERAL SCHEMATIC OF ENTIRE SYSTEM

Figure 1 illustrates the general flow of data within a complete system utilizing all major components.

As mentioned before the subsequent sections describe, in detail, the functions, use, and applications of all of these components.

FIGURE 1

GENERAL SCHEMATIC of UNIVAC FILE-COMPUTER



SECTION III

INPUT - OUTPUT EQUIPMENT

1. GENERAL

This section deals with the detailed operational characteristics of all the various input-output devices which may be included in a Univac File-Computer system. A section has also been included to deal with the multiplex adapter control necessary for multiple device systems.

The general format of data pertaining to each of the devices is as follows:

1) Description of media utilized by the device

2) Operation and function of the manual operating controls and indicators

3) Diagrams of the feeding sequence and functions

4) Diagram and explanation of the plugboard related to the device

5) Timing chart of the control functions

6) Explanation of verification checks related to the device

7) Program planning sheet for the device

8) Physical measurements and installation requirements

In addition to the above data, the section dealing with the multiplex adapters shows the necessary wiring related to each device when it is to be used in either a scan or demand mode of operation.

2. 90 COLUMN CARD SENSING PUNCHING UNIT - 150 CPM

1) Input-Output Media: Ninety column punched cards, each column containing six punching positions i.e. (0,1,3,5,7,9) The punching codes are:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H
0	x												x	x	x			
1		x	x								x	x				x		
3				x	x									x	x			x
5						x	x				x	x		x			x	
7								x	x				x			x	x	x
9			x		x		x		x	x	x					x		

	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	%
0				x	x	x							x	x	x	x			x
1		x					x	x		x	x						x		x
3	x	x	x				x	x	x			x		x	x		x		
5	x	x	x		x	x			x		x		x					x	
7								x	x	x	x	x	x		x	x		x	
9			x	x		x						x		x		x	x	x	

2) Operation & Function of the Manual Operating Controls & Indicators

POWER (Circuit Breaker) SWITCH is located near the front of the machine on the left side just below the Control Panel. This switch serves to turn On and Off all current to the 150 CPM Card Unit and 150 CPM Card Adapter Unit from the main power supply.

MOTOR (Toggle Switch) - This switch serves to turn On and Off the current to the Motor of the 150 CPM Card Unit. The Power Switch must be On as well as the Motor Switch before the current will flow to the Motor.

AUTO. FEED (Toggle Switch) - This switch is set On or Off to obtain one of the following two card feeding operations:

On - (Continuous) - After card feeding has been started, it will continue automatically. The last card from the Card Feeding Magazine will feed completely through the machine to the Card Receivers.

Off - (Single Cycle) - After card feeding has been started, only one card will be fed or one machine cycle taken.

CARD FEED (Throw Switch) - This switch is moved momentarily either up or down to one of the following two positions. The switch returns to the center position when released.

START - With cards in the Card Feeding Magazine, this switch is moved UP momentarily to this position to start the card feeding operation. Whether the card feeding operation continues automatically or whether only one card is fed depends on the setting of the Auto. Switch.

When the machine is set for automatic card feeding, only one movement to the Start position is necessary. The card feeding will continue automatically following that initial impulse.

When the machine is set for single cycle operation, this switch is moved to the Start for each card feeding cycle desired. To feed one card completely through the machine, three machine cycles are required.

STOP - To obtain optional stopping during an automatic run, this switch is moved momentarily to the DOWN position. Card feeding will stop at the conclusion of the current Program.

UNIT (Throw Switch) - This is a safety switch used in conjunction with either the Clear Switch to its right, or the Card Release Switch to its left (see below).

This switch is moved momentarily either up to its PUNCH position or down to its CALC. (Calculate) position. The switch returns to the center position when released.

CLEAR (Throw Switch) - This switch when moved simultaneously with the Unit Switch is used to reset the selectors by disconnecting B+.

CARD RELEASE (Throw Switch) - This switch is moved momentarily to its UP position simultaneously with an UP movement of the Unit Switch. This switch movement causes a card to be ejected from the machine into the Rear Card Receiver without punching. Any information in the Punching Setup Section will, however, be cleared.

One Up movement of both the Card Release and Unit Switches will feed a card from the Punching Section to the Rear Card Receiver, (despite the fact that the Sort may have been impulsed during the Program for that card), feed a card from the Sensing Section into the Punching Section (this card will be sensed), feed a card from the Card Feeding Magazine into the Sensing Section unless the cards have been removed from the Magazine before moving the switch.

Before using the Card Release Switch, any card in the Card Feeding Magazine would usually be removed or prevented from feeding by means of the Card Lifting Lever.

READY (White Indicator) - This indicator will light when the Card Unit is ready for operation. This means that the Card Unit is being supplied with the B+ voltage and the Motor is turned On.

VOLTAGE (Red Indicator) - Not used in the File-Computer.

CALCULATE (White Indicator) - While the Computer is connected to the Input/Output, this indicator will be lit. When this indicator lights and as long as it stays lit, no cards can be fed by means of the Card Feed Switch.

Reference to this indicator applies especially when the Computer is performing lengthy or iterative Programs to assure the operator that the machine is functioning.

INPUT CHECK (Red Indicator) - Not used in the File-Computer.

CARD FEED (Red Indicator) - The machine will stop at the conclusion of a Program with this indicator lit to detect a mis-fed card between the Punching and Sensing Sections or between the Punching Section and the Card Receivers.

All such mis-fed cards must be removed from the machine before resuming the card feeding operation. When the card feeding channel is clear, the operation is resumed with the Card Feed Switch.

RECEIVER (White Indicator) - The machine will stop at the conclusion of a Program with this indicator lit in the event of: a full Card Receiver; a full Chip Pan.

After removing the cards from the full Card Receiver or after emptying the Chip Pan, the light will go out. The card feeding is resumed by moving the Card Feed Switch to the Start position.

When the indicator lights and as long as it stays lit, no automatic card feeding can be obtained.

MAGAZINE (White Indicator) - The machine will stop with this indicator lit when the last card leaves the Card Feeding Magazine.

Should the Card Feeding Magazine become empty during a run, the stoppage will occur with this light lit. When the new supply is placed in the magazine, the indicator will go out. The card feeding is resumed by moving the Card Feeding Switch to the Start position.

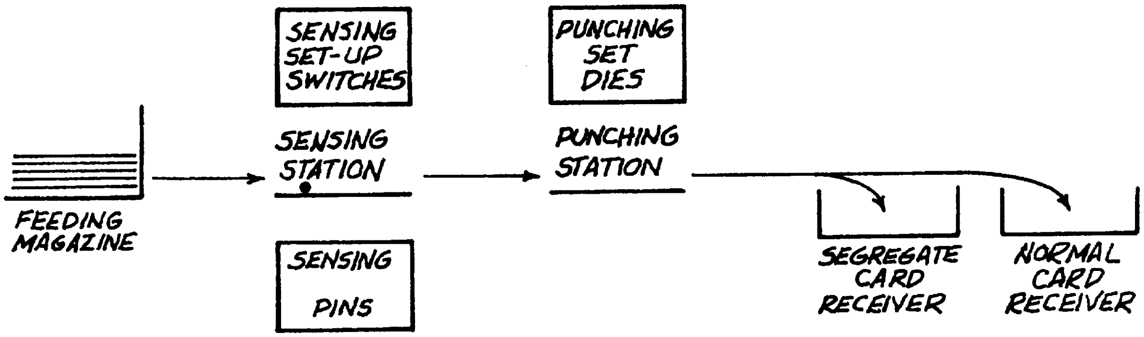
0 ÷ 0 (White Indicator) - Not used in the File-Computer.

N ÷ 0 (White Indicator) - Not used in the File-Computer.

TEMPERATURE (Red Indicator) - Not used in the File-Computer.

3) Diagram and Explanation of Feeding Sequence

FIGURE 2



Sequence of Feeding:

After the cards have been placed in the feeding magazine, the "Card Feed" Switch is raised and the following cycle is performed.

1st cycle

1. A card is fed to the sensing station.

2nd cycle

1. The sensing pins rise and set-up the sensing switches related to the positions punched in the card.

2. The sensing pins drop back to normal.

3. a. The Computer is notified that the input device is ready to be processed.

   b. The card at the sensing station moves to the punching station.

   c. The next card from the feeding magazine moves to the sensing station.

All subsequent cycles are controlled by the Computer program. Each time the device receives a "TRIP" signal the following cycle takes place:

1. a. The punching dies descend to punch the card with the data delivered to output storage for punching.

   b. The sensing set-up switches are reset to normal (cleared).

2. a. The punching dies rise and are reset to normal (cleared).

   b. The sensing pins rise and set-up the sensing switches related to the positions punched in the card at the sensing station.

3. The sensing pins drop back to normal.

4. a. The card at the punching station is delivered to either the "segregate" or "normal" receiving magazine dependent on whether a "segregate" control pulse was received from the computer during the last computing cycle.

   b. The computer is notified that the input device is ready to be processed.

   c. The card at the sensing station moves to the punching station.

d. The next card from the feeding magazine enters the sensing station.

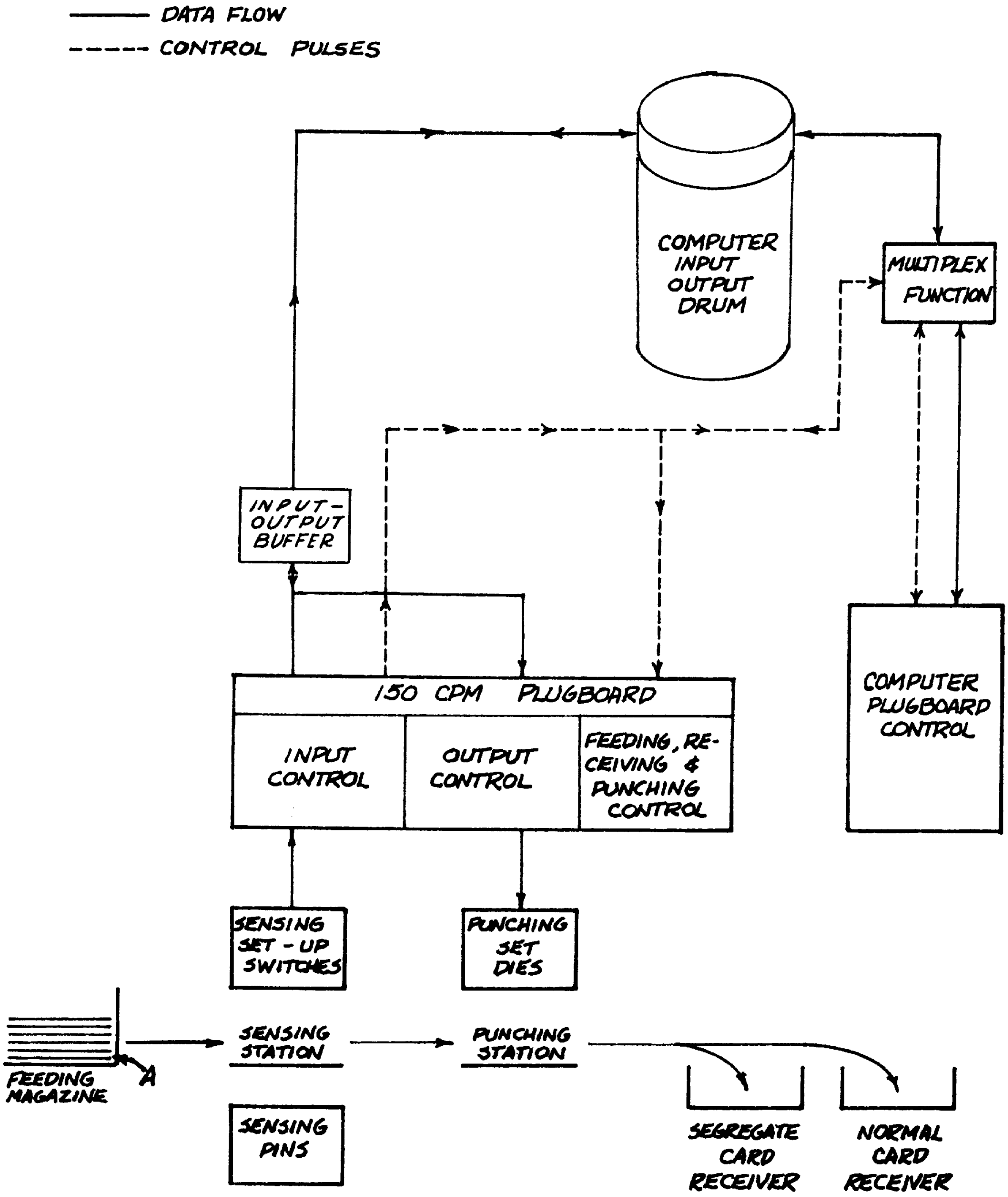
When the feeding magazine becomes empty the last card to enter will complete its calculations, be punched and delivered to a receiving magazine. At this point the card feeding will stop and the computer will not be notified that the device is ready to be processed.

If, at any time, during a cycle, a card fails to feed from the punching station to the receiver; or from the sensing station to the punching station, the card feeding will stop and the Computer will not be notified that the device is ready to be processed.

4) Diagram And Explanation Of The Plugboard

Before discussing the 150 CPM plugboard in detail the general flow of data and control between the device, its plugboard, and the computer will be explained. A schematic of this data flow is shown in figure 3.

FIGURE 3



——— DATA FLOW
- - - - - CONTROL PULSES
COMPUTER INPUT OUTPUT DRUM
MULTIPLEX FUNCTION
INPUT-OUTPUT BUFFER
150 CPM PLUGBOARD
INPUT CONTROL
OUTPUT CONTROL
FEEDING, RECEIVING & PUNCHING CONTROL
COMPUTER PLUGBOARD CONTROL
SENSING SET-UP SWITCHES
PUNCHING SET DIES
FEEDING MAGAZINE
A
SENSING STATION
PUNCHING STATION
SENSING PINS
SEGREGATE CARD RECEIVER
NORMAL CARD RECEIVER

The preceding diagram shows the detail data and control flow between the 150 CPM unit and the Computer. To illustrate this matter more clearly, trace the data flow and card feeding operations of Card A.

Raise the manual Card Feed switch and:

a. Card A enters the card sensing station.

b. The sensing pins rise and set up the sensing switches related to the positions punched in Card A.

c. Information punched in Card A is now available on the input plugboard and through input wiring may begin to be transferred into the buffer core storage and thence into the input drum storages.

d. Sensing pins drop.

e. Card A moves to the punch station.

f. The device signals the Computer that Card A's input is ready to be processed.

g. The next card in the feeding magazine moves to the sensing station.

h. The Computer has been calculating Card A's data and delivering the output results to the output drum storages. During this time Card A remains at the punching station.

i. On completion of the program a "TRIP" signal will be delivered to the input-output plugboard thru an output control line from the Computer. This trip signal activates the following events in sequence.

   1) All drum output storages to be punched in the card are delivered (thru the input-output plugboards control) to the desired set dies and the dies are set for punching.

   2) The punching set dies descend and the dies which were set in (1) perforate their related positions in Card A.

   3) The set dies rise and clear.

   4) The sensing set-up station clears.

   5) The sensing pins rise and set up the sensing switches for the card in the sensing station.

   6) The sensing pins return to normal.

7) Card A moves from the punching station to the normal receiver (note: if the Computer had delivered a segregate pulse to the input-output plugboard via an output control line during Card A's program, Card A would enter the segregate receiver).

8) The card in the sensing station repeats the cycle beginning at (e) above.

As can be seen from the above example the functions available on the input-output control board are utilized during the following steps:

c. Determination of what card columns go to which input storage units.

i. Wiring of the trip signal from an output control line.

i. (1) Determination of what card columns are to be punched from which output storage units.

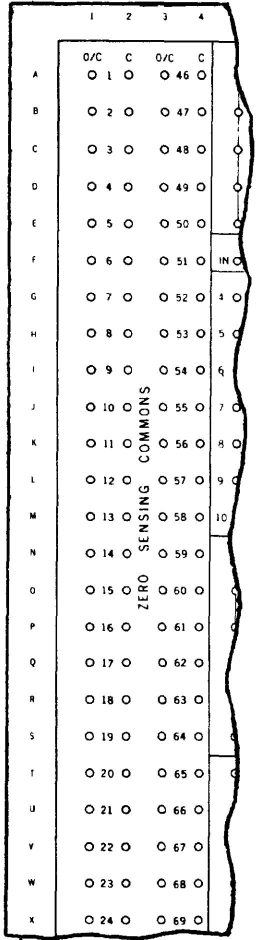
These three points are the major considerations in wiring the plugboard and are treated individually in the following detailed discussion of the input-output plugboard.

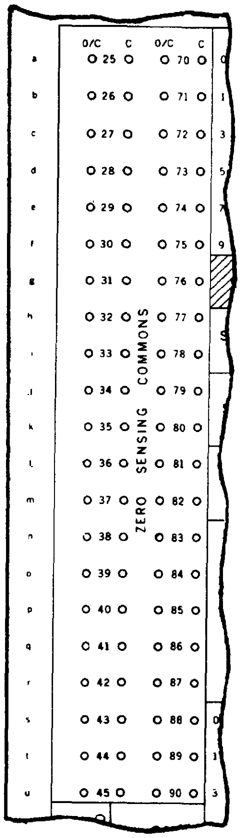
The major areas of the plugboard are explained from a functional standpoint and examples of wiring are included for the various uses of the plugboard hubs. (Note complete plugboard diagram on page III - 38.)

Card Sensing

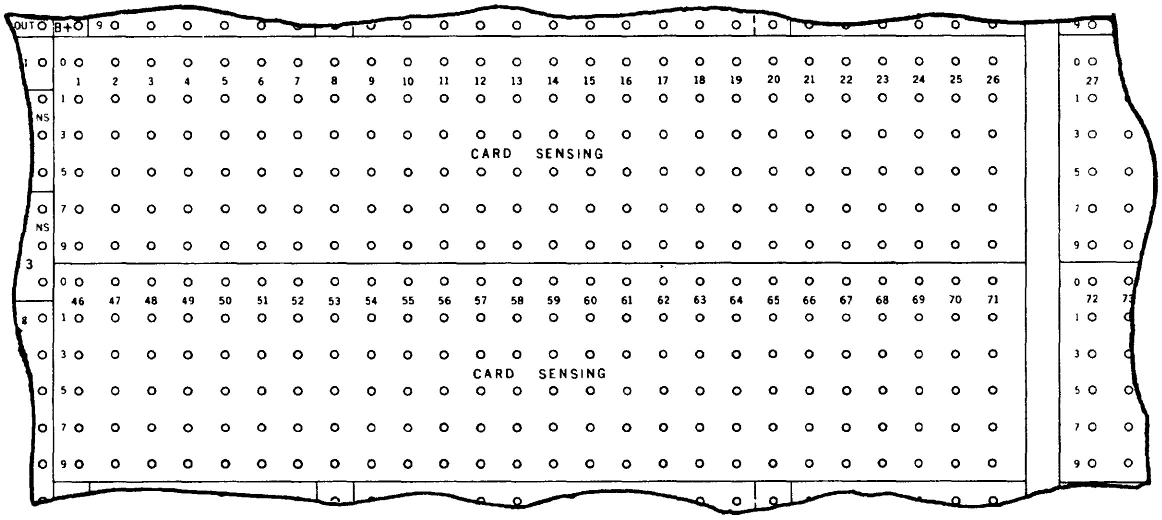
The plugboard hubs associated with the sensing of data from the card are:

(1) Zero sensing commons (A-X a-u) (1-4)

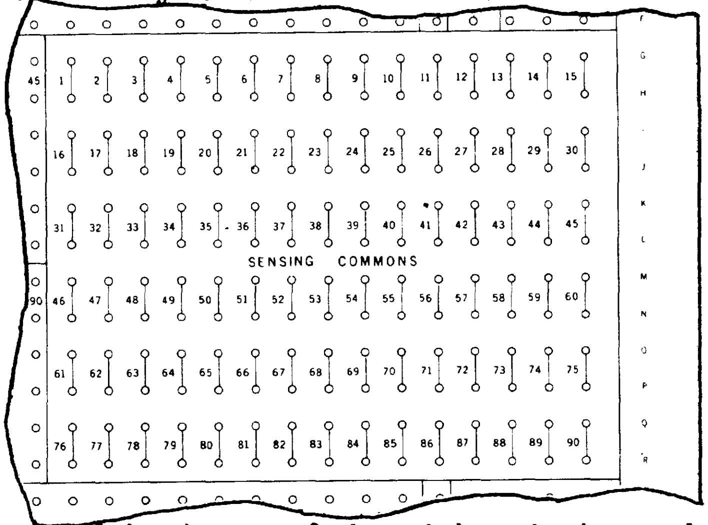




(2) Card Sensing (G-R) (9-53)



(3) Sensing Commons (G-R) (54-68)



In order to explain the use of these hubs, the internal machine connections between these three parts of the card sensing switches are illustrated. Since all of these hubs are plugboard outlets of the 90 card sensing switches we can best understand their function by illustrating a sensing switch. For this purpose, let us examine the sensing switch connections for card column 46.

SCHEMATIC of 150 CPM CARD SENSING CIRCUITRY

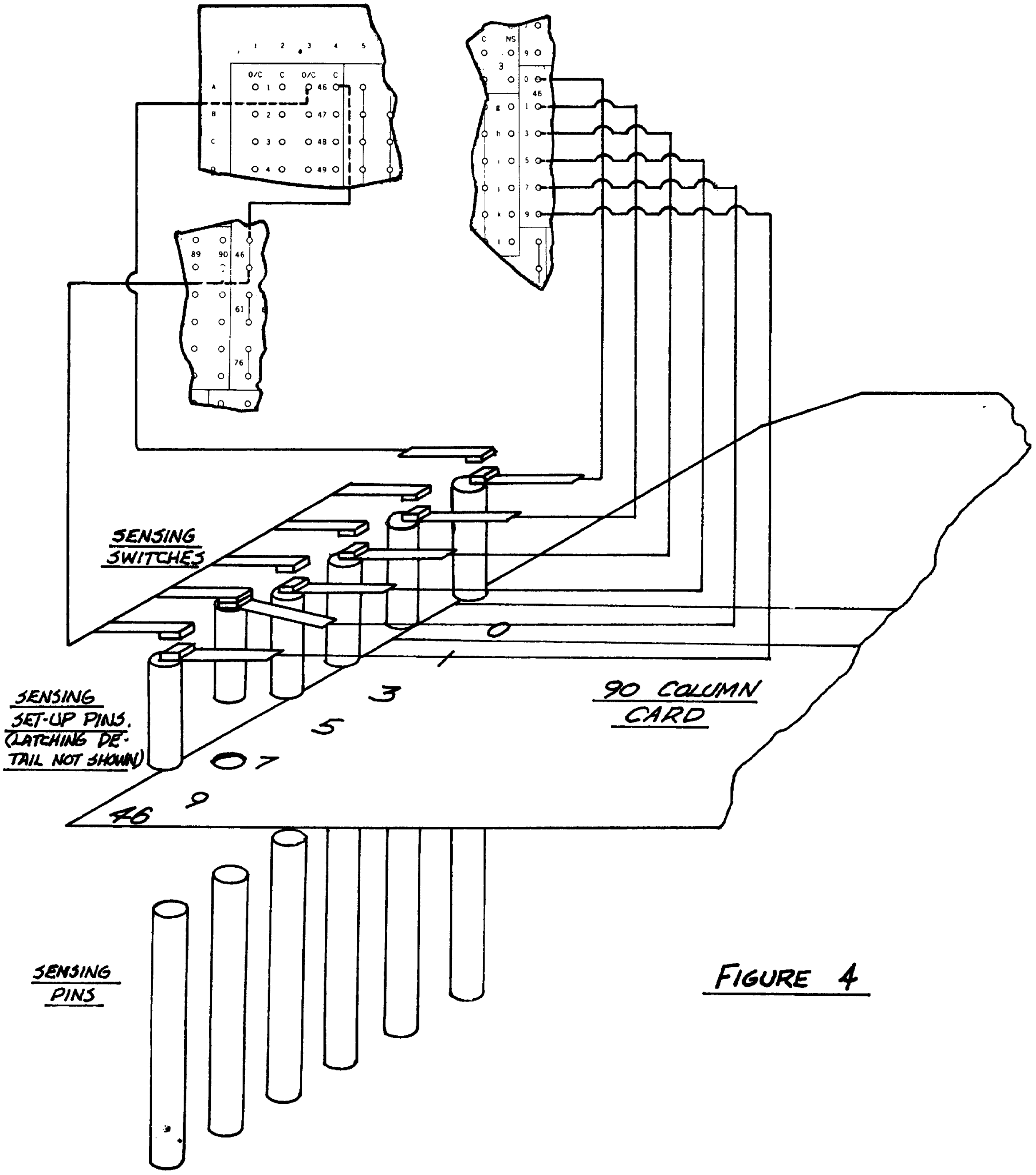


FIGURE 4

In the case shown in figure 4 the following things have occurred:

1) The card entered the sensing station

2) The sensing pins rose, the seven (7) pin passed through the hole in the card, pushed the sensing set-up pin seven (7) into a latched position. This also closed the seven position sensing switch. All other sensing pins (0,1,3, 5 & 9) were stopped by the card and therefore did not activate their respective sensing set-up pins.

3) The sensing pins dropped (returned to normal), however the seven (7) sensing set-up pin remains latched (latching mechanism not shown in diagram) thus keeping the seven (7) closed. The seven (7) sensing switch will remain closed until the device receives a "TRIP" signal which un-latches the set-up pin allowing it to drop. This breaks the seven (7) sensing switch connection.

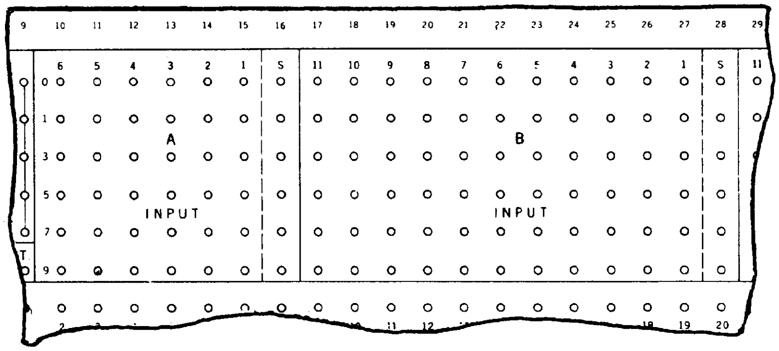
As can be seen from figure 4, any current entering the card sensing common will be received at the card sensing positions related to the positions punched in the card. Note that if any current is to be received at the zero (0) positi.on the originating current must enter the O/C hub of the zero sensing commons. This separation of the zero switch from the 1,3,5,7, & 9 switches allows us to use zero's over numeric fields as control positions while still allowing the use of the other positions in the column as normal input data. Of course if the column being wired contains alphabetic data, then the "O/C hub" should be wired to the "C hub" thus connecting the zero position to the common in the same manner as the other positions.

The assignments of card columns to specific input storage locations are shown in the following paragraphs which deal with the input storage sections of the plugboard.

Input Storage

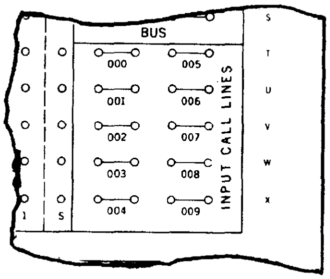
The plugboard hubs associated with delivering data sensed from the card to the computers input storage are:

1) Input Field Entry hubs (A-F) (10-64) & (S-X)(10-64)



These hubs represent direct entries to the input buffer core storage. Thus if card sensing columns are connected to these input entry hubs and current is run into the card sensing common then the buffer core storage will receive the data punched in the card columns and transfer this data to the input drum storage. The current used for this purpose is obtained from the:

2) Input Call lines (T-X) (65-68)



There is a set of two common hubs for each of the ten possible input storage locations (000-009). These hubs emit current at the appropriate time to allow entry of the data punched in the card into the buffer storage in time to be entered from the buffer into the related drum storage location (000-009). It should be noted that although there are ten, twelve digit, drum input storage locations there is only one, twelve digit, buffer storage; thus the data from the card is entered into the buffer and thence into the drum on a timed sequential basis. The sequence of entry is:

000,002,004,006,008,001,003,005,007 and 009

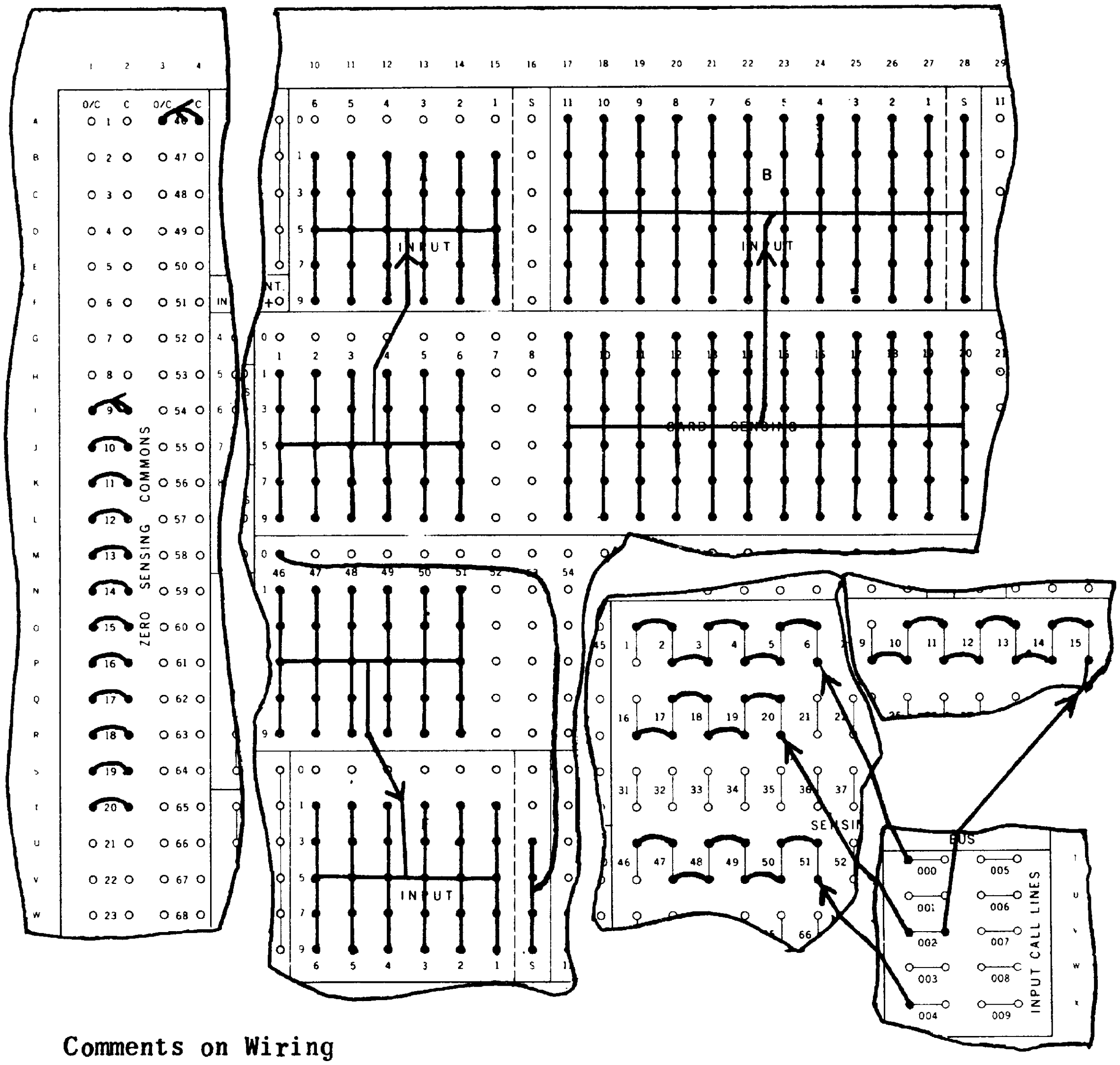
An example of the necessary wiring will show these input functions more clearly.

Assume that card columns are assigned to input storages as follows:

Card Columns	Input Storage	Type of Data
1-6	000	Numeric-always positive (+)
9-20	002	Alphabetic Description
46-51	004	Numeric data-negative if a 0 in Col. 46

The wiring for this input data would be:

Note: The arrows shown on all following diagrams are drawn as an aid in tracing the wiring.



Comments on Wiring

1. Since columns 1-6 were known to be numeric it was not necessary to wire the zero positions since the absence of an input code will generate a zero in the Computer arithmetic operations. However, the zero positions could be wired without changing the result in any way.

2. Since columns 1-6 were known to always represent a positive number there was no need to wire anything to the sign position. The lack of a specific minus code will cause the Computer to accept the data as a positive number.

3. Since columns 9-20 contain an alphabetic description and therefore will not be used in any arithmetic (+ - ÷ X) operations, it is possible to wire all twelve columns to one storage (002), the twelfth column being entered in the sign position. Since the minus code is a (3,5,7,9) and no alphabetic character consists of these positions the Computer would consider this a positive value.

4. Since columns 46-51 are considered negative if there is a zero (0) in column 46, the zero position of column 46 has been wired to the sign position of the input entry field. Note that a special wire is used here to convert the single zero position into the four position minus code.

The above wiring provides for the following sequence of events to take place.

1. Input call line 000 emits and probes card sensing switches 1-6.

2. Those positions of the columns 1-6 card sensing switches representing the holes punched in the card emit 000 call line current and send it into the input field entry hubs and thence into the buffer storage.

3. The input buffer storage reads out and enters the data it contains from card columns 1-6 into the 000 location of the input storage drum. This also clears the buffer.

4. Input call line 002 emits and probes card sensing switches 9-20 and also the O/C hubs of columns 9-20.

5. Those positions of the columns 9-20 card sensing switches representing the holes punched in the card emit 002 call line current and direct it into the input field entry hubs and thence into the buffer core storage.

6. The input buffer storage reads out and enters the data it contains from card columns 9-20 into the 002 location of the input storage drum. This operation also clears the buffer of data.

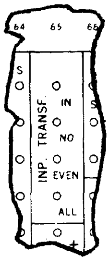
7. Input call line 004 emits and probes card sensing switches 46-51 and also the O/C hub of column 46.

8. Those positions of columns 46-51 representing the holes punched in the cards, emit 004 call line current and direct it to the input buffer cores. In addition a O in column 46 creates four position entries in the sign position of the buffer (3,5,7,9) creating the negative indication.

9. The input buffer reads out and enters the data it contains from card columns 46-51 and the O in column 46 into the 004 location of the input drum. This "read out" operation also clears the buffer of information.

The next matter to be considered in planning input storage is: what storages should be used? The answer to this question lies in the next subject of:

(3) Input Transfer (Control) (A-D (65)



These hubs control how many input transfer cycles from buffer to drum will be accomplished. In order to further clarify this we should realize that the transfer of input data from the card columns to the buffer to the drum occurs in the following sequence:

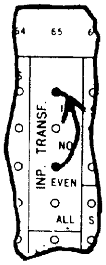
1. 000 call line to card columns to input buffer to the input drum location for the call line.
2. 002 - same as 1
3. 004 - " " "
4. 006 - " " "
5. 008 - " " "
6. 001 - " " "
7. 003 - " " "
8. 005 - " " "
9. 007 - " " "
10. 009 - " " "

This complete sequence of operation requires 2.5 ms average to find the first drum location (000) and then two complete drum revolutions or 10 ms to complete the cycle for a total of 12.5 ms. This time can be reduced through use of the input transfer hubs to 7.5 ms. if only five storages are used or zero ms. if no inout is reauired. These three cases are:

1. Assume that only 5 fields of data are required as input:

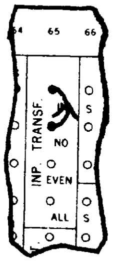
Card Cols.	Input Storage
1-6	000
7-12	002
13-18	004
19-26	006
27-33	008

The card field columns are assigned to the even numbered input storages so that the input data is transferred on the first drum revolution, thus giving an input transfer time of 7.5 ms (2.5 to locate 000 plus 5 ms for one drum revolution). The input transfer control is wired:

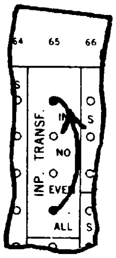


This technique may be used in all cases where 5 storages or less are required.

2. Assume that no input is required in the problem. In this case the input transfer control is wired:

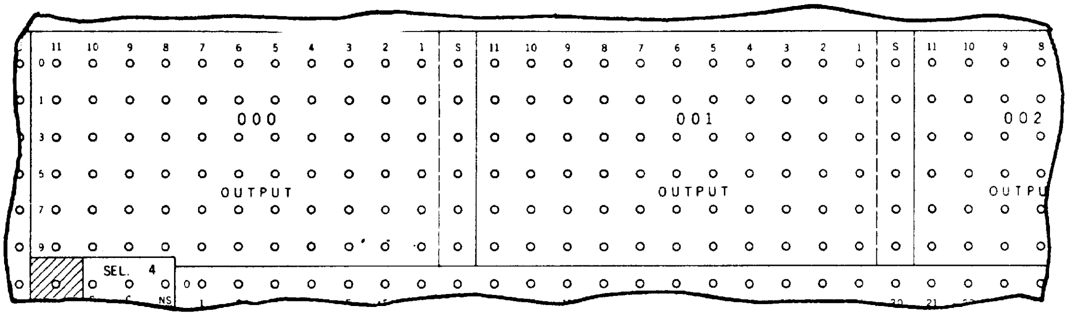


3. Of course if more than five storages are required the input transfer control must be wired:



It should be noted that all three of these methods of input transfer control can be combined in one program and controlled by positions punched in the card. This control usually will be accomplished through selectors (discussed in following paragraphs) since the input transfer control must always be wired to one of the three positions.

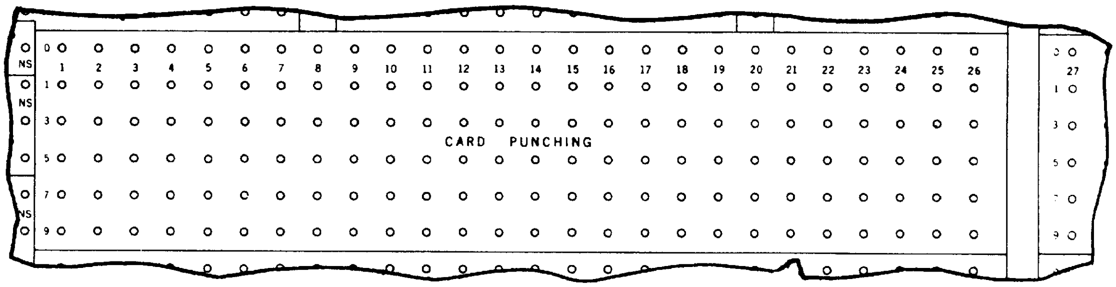
Output Storage (a-f) (5-64) & (s-x) (5-64)



These storage output hubs represent the output storage buffer exits and are used to deliver the output data developed by the Computer to the card punching section.

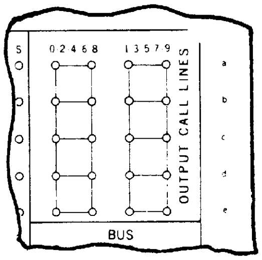
It is to be noted that an output storage position cannot be wired to punch to more than one place. However, wiring from one of two storages to punch into one place can be done if one of the storages is odd and the other even.

Card Punching (g-r) (9-53)



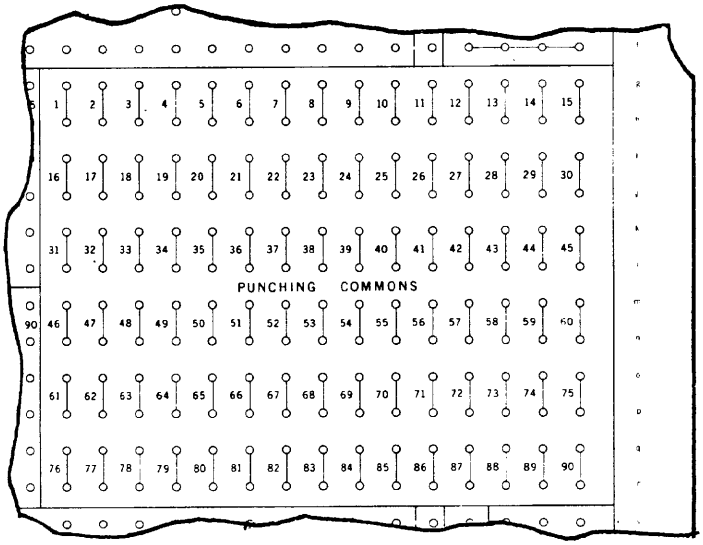
These hubs represent the entries to the punching dies which are used to punch the output results into the card. These hubs are wired directly from the output storage hubs in order to receive the output data from the Computer. The controlling or signalling current which effects this setting of punch dies from the storage output is from the output call lines.

Output Call Lines (a-e) (65-68)



These output call lines emit a timed pulse controlled by the "TRIP" signal received from the Computer. As can be seen, there are in effect only two call lines; one for even storages (the first input-output drum revolution) and one for odd numbered storages (the second input-output drum revolution). Dependent on whether the storage delivering the result has an odd or even number, one of the two call lines is selected and wired to the punching commons of the card columns into which the stored data is to be punched.

Punching Commons (g-r)(54-68)

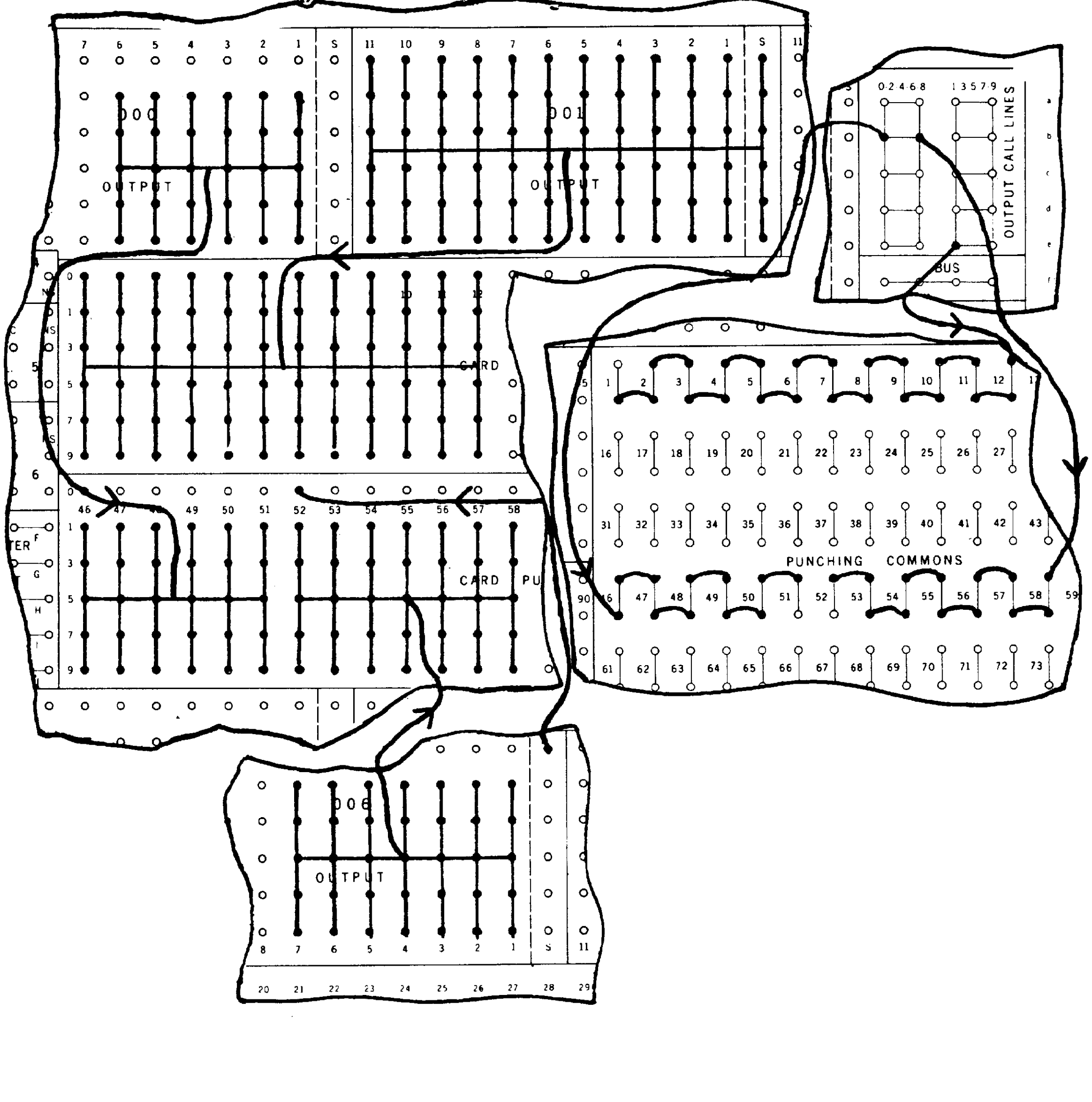


These hubs receive the signal from the output call lines which instructs them to set the punch dies in their column with the data received at their card punching positions. The card punching positions would of course, be wired to the output storage related to the call line pulse received at the punching common.

In order to integrate the information concerning Output Storage, Card Punching, Output Call lines, and Card Punching Commons let us take an example where:

Size of Field	Sign	Storage	Numeric or Alpha	Punch in Cols.
6	+	000	Numeric	46-51
12	none	001	Alpha	1-12
7	+ or -	006	Numeric	52-58 punch a zero in 52 if negative

The wiring for this situation would be:



Comments on Wiring

1. The zeros in columns 46-51 and 53-58 need not be wired if zero punching is not desired.

2. Since output storage emits a zero from the sign position for negative control, this position in 006 is wired to the desired negative control punching position (0/52). Note that this could have been wired to punch any other position desired (1,3,5,7 or 9).

3. Note that in the case of 001 the zeros must be wired if we are to get the correct alphabetic punching.

4. Note that there is no sign indication of a positive result although the zero in the sign position signifies a negative result.

The remaining subjects to be covered can be considered under one heading.

CONTROL FUNCTIONS

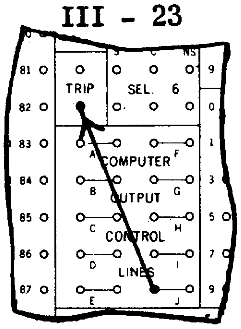
The control functions and features cover those plugboard hubs which are used to vary the input or output format, energize certain computer functions on the basis of data punched in the card, receive signals from the computer to energize output control functions, etc. All of these functions are considered individually in the following paragraphs.

Feeding and Punching Control

As noted earlier in this section, the 150 CPM device feeds two cards upon raising the start switch and then feeds only under control of the computer. This control is provided by pulses received from the computer (by programming) through the output control lines (n-r) (5-8). These control lines are then wired to the feeding and punching functions of:

Trip (l-m) (5)

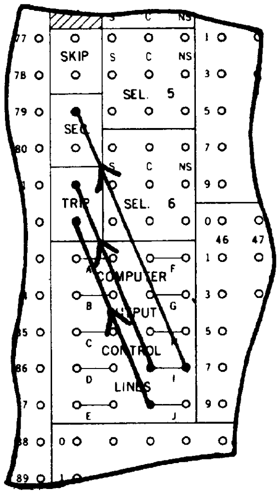
Let us take an example where the computer delivers a program end or complete pulse to output control line J and this pulse is to instruct the 150 CPM device to punch the output data, feed the card into the normal receiver, sense the next card and feed it to the punching station, and feed the following card to the sensing station. The wiring would be as follows on the 150 CPM plugboard.



The two trip hubs perform the same functions, however they are diode protected so that current going into one cannot backfeed thru a wire going into the other.

Segregate (j-k) (5)

Assume in this case that the program sometimes desires to trip the card and segregate it into the segregate receiver, and at other times the program wishes to merely trip the card. For purposes of illustration let us assume that the program delivers a pulse to output control line I when it wishes to segregate and Trip and to output control line J when it wishes to merely trip the card and deliver it to the normal receiver. The wiring for these functions would be:



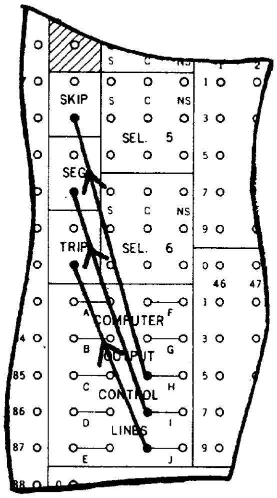
Since the Trip and Segregate hubs are all diode protected there will not be a back current to the Segregate hub when output control line J is activated.

Skip (h-i) (5)

This function prevents the punching of all output storage data (even though wired) when it is impulsed prior to or simultaneously with the trip function. It should be noted that the skip function's two hubs are also diode protected to prevent back circuits. To illustrate this function let us assume the computer will impulse the indicated control lines when it wants the following output function:

Output Control	Function Desired
J	Trip
I	Segregate
H	Skip

The wiring would be:



Note that neither segregate or skip need to be combined with a trip function, but may actually come from the computer many steps before the trip signal is received. Thus in the above case the computer could, by impulsing the correct control lines during the program, accomplish the following combinations of functions.

1. Trip
2. Segregate and Trip
3. Segregate, Skip and Trip
4. Skip and Trip

Selector Control (see section V for explanation of selectors)

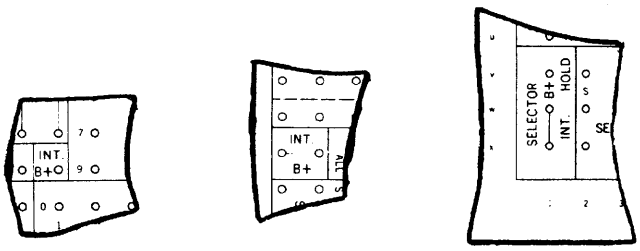
Use of the selectors on the 150 CPM plugboard can be sub-classified into the following areas of interest:

1) Use of selectors to vāry the input-output plugboard wiring on the basis of card controls.

2) Use of selectors to vary the input-output plugboard wiring on the basis of computer control.

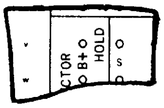
Both of these areas require use of the 150 CPM selectors, B + current, and in the second use, program selects. Before discussing the applications we will cover these functions of the plugboard.

B+ Int (F)(9), (E-F)(65), (w-x)(1)



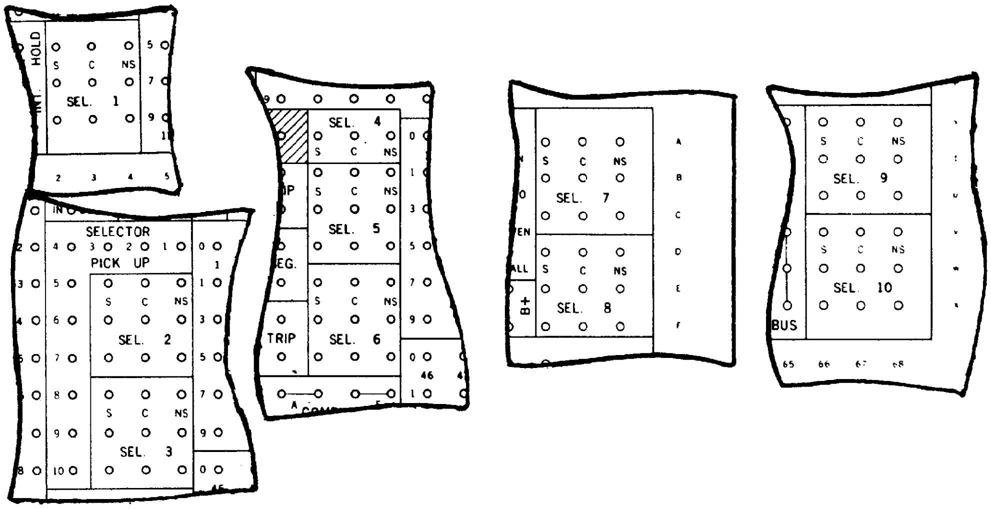
The B + int (interrupted) hubs emit current capable of energizing the 150 CPM selectors during the calculating time of any card. This current stops emitting after output punching time and starts emitting again before Input call line time for the next card.

B + Hold (v)(1)



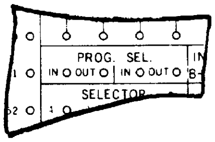
This B + power starts emitting as soon as the input device is turned on and continues to emit, without interruption until the device is turned off or the Clear Switch is used (See III-3).

Selectors (H-M)(6-8), (g-m)(6-8), (v-x)(2-4),
(A-F)(66-68), (s-x)(66-68)



These selectors act as switches in that any current entering the C hub (common) will emit from either the S hub (select) or the NS hub (non-select) dependent upon whether or not the related selector pick-up (G)(5-8), (H-M)(5) has been energized by B + or program select current. If one of these two types of current has been received at the pickup then any pulse entering into the C hub will be received out of the S hub. Conversely, if current is not received at the pickup hub then any pulse entering into the C hub will emit from the NS hub. Any time the pickup is energized and then the current stops being received at the pickup, the selector will resume a non-select state (current into C will exit from NS).

Program Selects (F) (5-8)



The two program selects are used to convert current received from the computer, via output control lines, into the equivalent of B + int for use in energizing the selectors. The current from the output control line is wired to the "in" of the program select and when the current is received the "out" of the program select delivers B + interrupted current and may be used to energize selectors.

As stated previously, these selectors are used primarily in two ways. Examples of these applications are:

1) To vary input-output plugboard wiring on the basis of card controls.

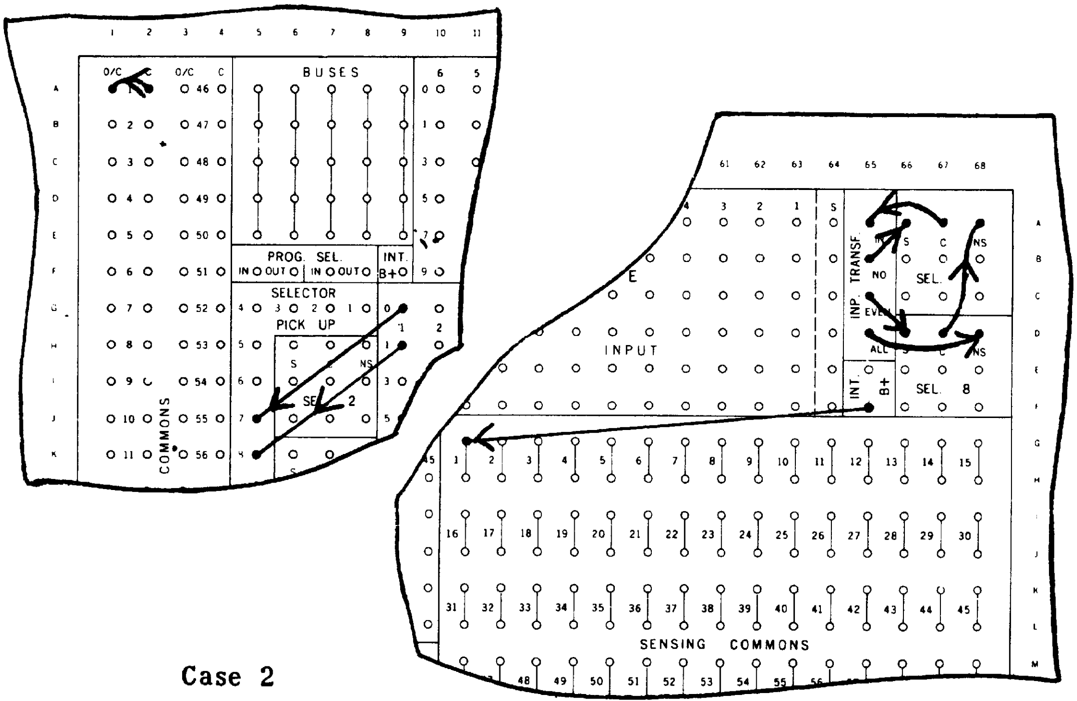
Case 1
Assume that:

a. if there is a 0 in column 1 no input information is needed. Therefore, the "no" hub (B-65) should be wired to the "in" hub (A-65) to prevent the input transfer cycle.

b. if there is a 1 in Column 1 the input data in 000,002 and 004 should be transferred to storage. Therefore the "even" hub (C-65) should be wired to the "in" hub.

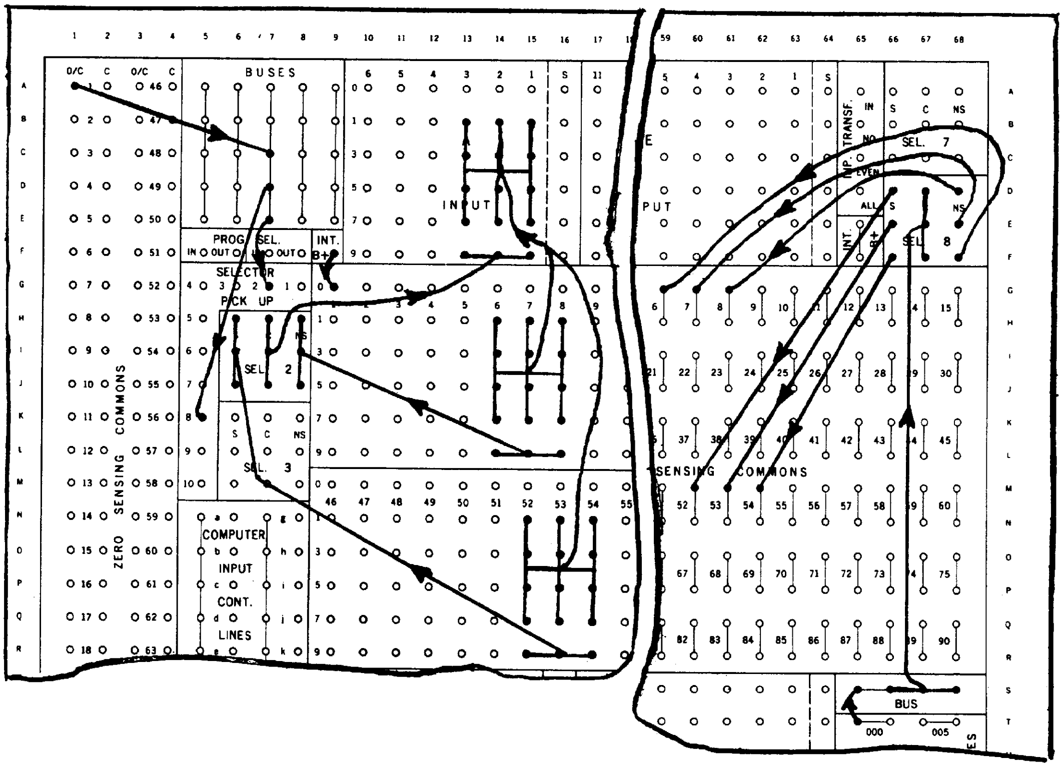
c. if there is neither a 0 nor a 1 in column 1 all input data should be transferred to storage. Therefore, the "all" hub (D-65) should be wired to the "in" hub.

The wiring for this case is:



Case 2

Assume that card columns 6-8 should be wired to the three least significant positions of input entry A if there is not a 0 in column 1. If there is a 0 in column 1, columns 52-54, rather than 6-8, should be wired to the three least significant positions of input entry A. In either case the field should be called for by 000. Both fields are numeric.

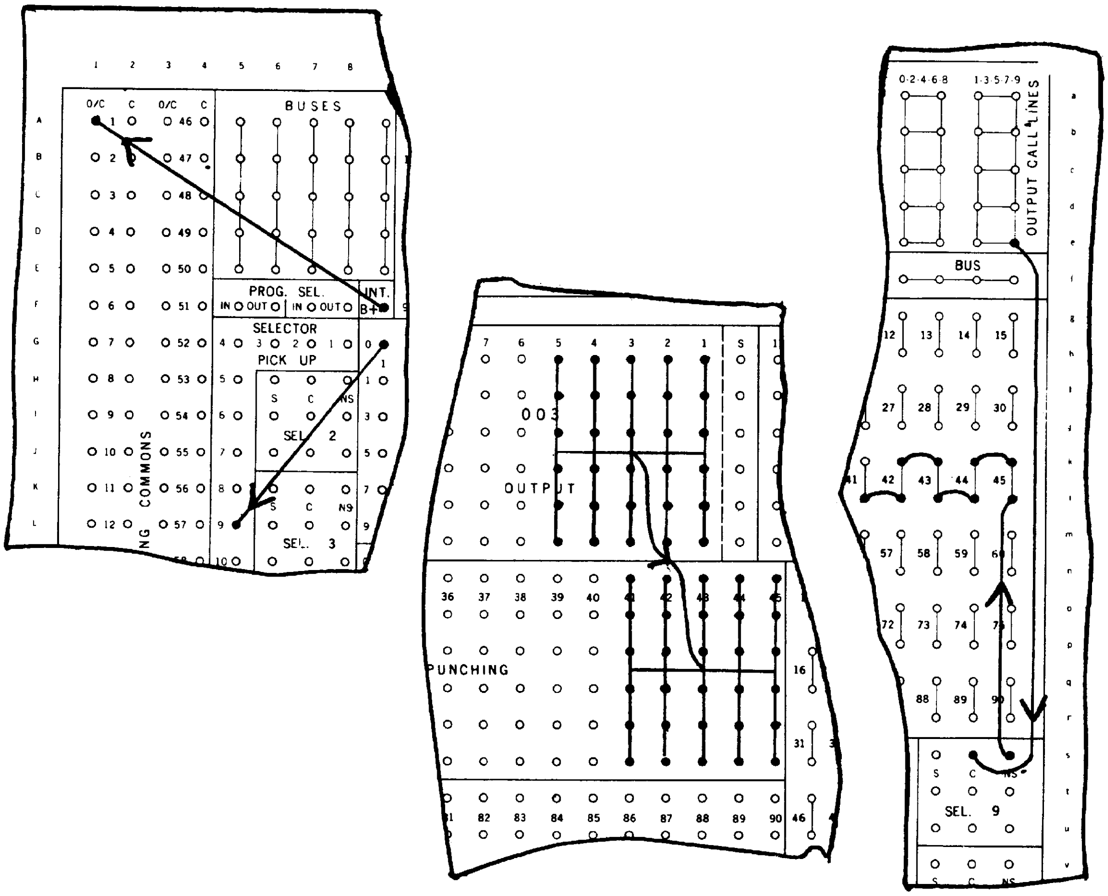


Note that, not only must the input call line for 000 be selected, but also the 9 position of every entry column. This is necessary because of the back circuit condition that would occur when the column selected to be sensed is punched with an odd code (1,3,5,7,9), and the unselected column is punched with the related even code (2,4,6,8). For example, in the preceding case assume that the card being sensed is punched with a 1 in column 52, a 2 (position 1 and 9) in column 6 and a 0 in column 1. Since there is a 0 in column 1, input call line 000 current enters the sensing common of column 52, through the sensing switch and out the position 1 wire to the 1 input entry position---from there to the 1 position in column 6, through column 6 sensing switch and out the 9 position of column 6. Thus, if the 9 position was not selected, it would impulse input entry position 9, entering a 1 and a 9 (2) in the input position.

If the fields contained alphabetic data, all positions woula have to be selected.

Case 3

If there is no 0 in column 1 output storage 003 should be punched into card columns 41-45. If there is a 0 in column 1, columns 41-45 should not be punched. The wiring is as follows:

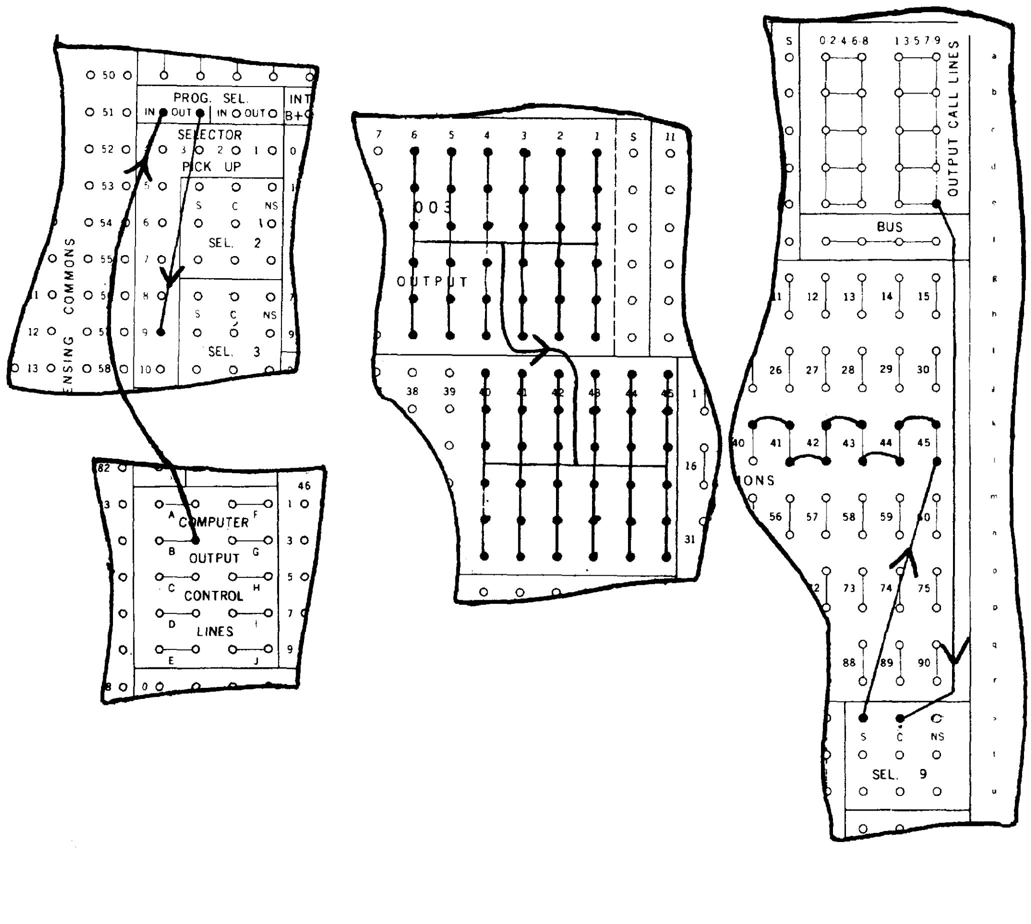


Many other types of selection which fall into this class are possible; however, the foregoing examples should be sufficient to realize the possibilities of selector usage.

2) Use of Selectors to vary the input-output plugboard wiring on the basis of computer control.

In these cases the only plugboard positions which will be considered are those of an output nature. This is because the input functions take place before the computer has an opportunity to control them. The one exception to this rule is where a condition arising on one card is to effect a subsequent card or cards.

Case 1
If the computer delivers a pulse to output control line B we want to punch the six least significant positions of 003 into card columns 40-45. If output control line B is not energized by the computer we do not want to punch 003 into columns 40-45. The wiring for this case is:



The program select was used to energize the selector rather than the output control line for two reasons, these are:

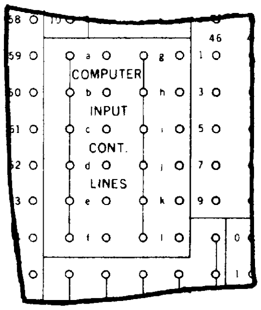
1. The output control pulse is momentary and would not be emitting at punching time, thus the selector wouldn't be in a select position at the correct time. Whereas if the output control activates the program select, even momentarily, the "out" hub of the program select emits, and thus energizes the selector, until after punching time.

2. The output control pulse is not of the correct characteristics of power to activate a selector directly.

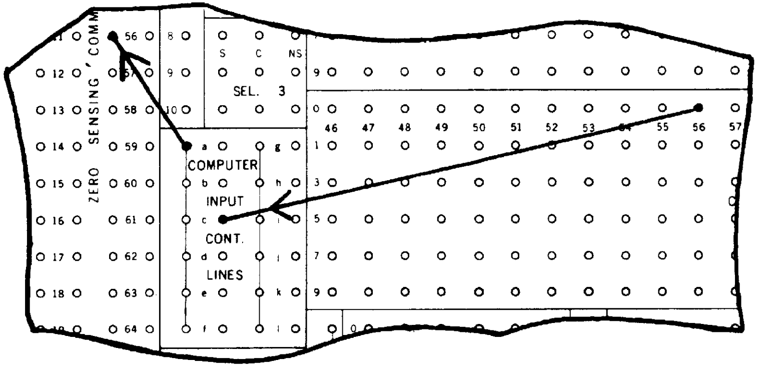
The above cases illustrate the use of the selectors on the input-output device. The next matter is the discussion of how to activate computer selectors on the basis of card controls.

Input Control Lines (N-S) (5-8)

The input control lines are direct tie lines between the input-output device and the computer (under control of the multiplex function). They consist of two hubs per line and appear as:

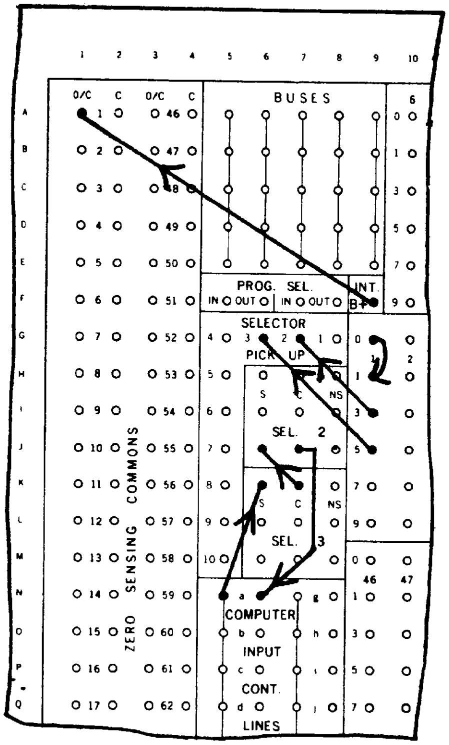


The common hub on the left is the equivalent of computer selector pick-up power and it is used to probe card columns for the presence of a position. If the position is present this current is allowed to get to the computer thru the right hand hub. Wiring on the computer plugboard (see section V) completes the selector pick-up wiring. For example, assume that if there is a 0 in column 56 a computer selector should be energized thru input control line "c". The wiring would be:



Wiring from input control line c on the computer plugboard to the selector pick-up would pick up the selector, provided the circuit was completed by wiring the computer selector ground.

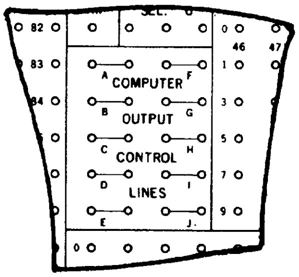
These controls can also be used in conjunction with other input-output plugboard selectors in order to reduce the number of control lines used. For instance we might want to pick-up a computer selector only if there was a 0, a 1, a 3, and a 5 in column 1. This could be wired as follows:



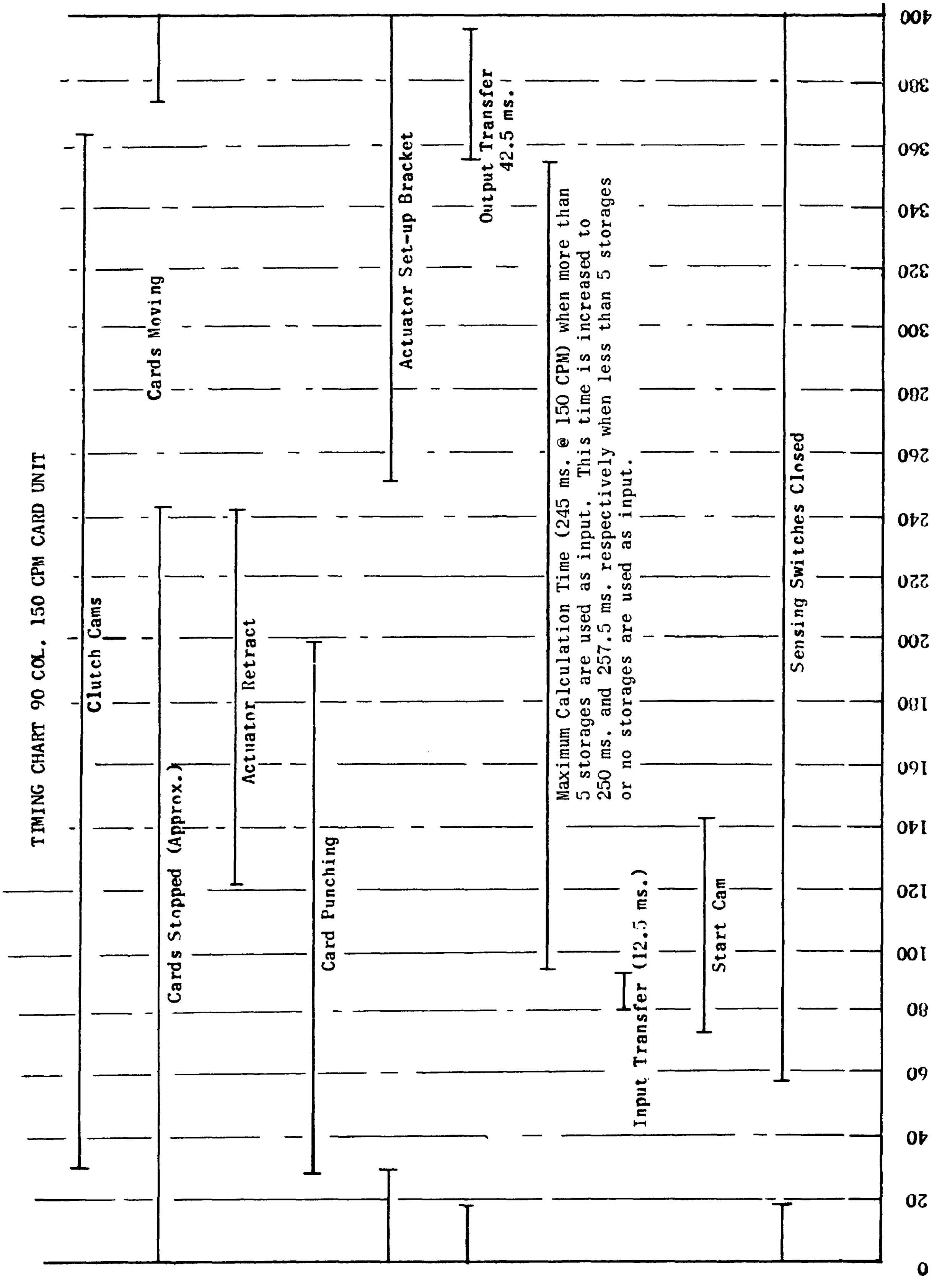
It should be remembered that the input control lines are diode protected to accept current in only one direction; that is from the input device to the computer.

Output Control Lines (n-r) (5-8)

Use of these control lines has been demonstrated from the input-output plugboard standpoint, and their use from the computer standpoint is shown in Section V; however, it should be noted here that computer output control lines will pass current in only one direction, that is from the computer to the output device.



5) Timing Chart



TIMING CHART 90 COL. 150 CPM CARD UNIT

Maximum Calculation Time (245 ms. @ 150 CPM) when more than 5 storages are used as input. This time is increased to 250 ms. and 257.5 ms. respectively when less than 5 storages or no storages are used as input.

The timing chart indicated that maximum calculation time for operation at 150 CPM is 245 ms. The following table indicates card production for programs requiring more than 245 ms.

Total Program Time in Milliseconds *	Approximate Card per Minute Production
245	150
250	140
260	136
270	132
280	130
290	126
300	124
320	119
340	112
360	109
400	103
450	93
500	86
550	80
600	75
650	68
700	65
750	61
800	58
850	55
900	53
950	51
1000	50

* These times are increased by:

5 ms if less than 5 input storages are used
12.5 ms if no input storages are used

8) Physical Measurements and Installation Requirements

Length	33"
Width	30"
Height	65"
Weight	1020 lbs.
Voltage	230/220/208 Volts Single Phase 60 cycles
Power	1.0 Kilowatt
Cooling Air In Cubic Feet Per Minute	None
Heat Dissipation	57 BTU's per minute

MULTIPLEX ADAPTER

OPERATING CONTROLS - 150 CPM CARD ADAPTER UNIT

START (Push Button Switch) - Turns on power to Adapter Unit. Light above start button is lit while power is on. Circuit breaker on 150 CPM Card Unit must be closed to furnish power to the Adapter Unit.

STOP (Push Button Switch) - Turns off power to Adapter Unit.

OPERATE (Indicator) - This lamp when lit indicates the Adapter Unit is ready to function with the Computer and Card Unit.

OVER TEMPERATURE (Indicator) - This lamp when lit indicates the Adapter Unit is running at an excessive temperature. Should the temperature continue to rise, the computer will shut down automatically.

REDUCED FILAMENT (Indicator) - This lamp when lit indicates the unit is operating with reduced filament voltage for marginal checking purposes. The controls for reducing filament voltage are located in the Main Control Unit.

DISABLE MPX (Toggle Switch) - When this switch is set to the On position, the Multiplexer station will be immediately disabled if the Computer is not connected (by the Multiplexer) to the Input/Output track. If the Computer is connected to the Input/Output track when the switch is set to the On position the Multiplexer station will not be disabled until the Computer releases the Input/Output track. As long as the Computer is connected to the Input/Output track the Calculate light on the Card Unit is on.

PUNCH TEST (Toggle Switch) - When this switch is set to the On position, the Adapter Unit control relays are disconnected from the Card Unit signal cams. This allows the Card Unit to be operated for test purposes without generating Adapter Unit control signals.

UNIVAC FILE-COMPUTER SYSTEM

150 CPM INPUT-OUTPUT CHART

APPLICATION: ______________ PROGRAM NO.: ______________

INPUT-OUTPUT STORAGE FIELD ASSIGNMENT

DEMAND UNIT NO. ________ OR SCAN UNIT NO. ________

INPUT FIELD ASSIGNMENT

SYM.	DESCRIPTION	CARD COLUMNS AND DECIMALS												SIGN CTL.
		11	10	9	8	7	6	5	4	3	2	1	SN.	
000														
001														
002														
003														
004														
005														
006														
007														
008														
009														

ENTER DESCRIPTION OF INPUT DATA

ENTER CARD COLUMNS TO SENSED AND INDICATE DECIMAL (▲).

OUTPUT FIELD ASSIGNMENT

SYM.	DESCRIPTION	CARD COLUMNS AND DECIMALS												SIGN CTL.
		11	10	9	8	7	6	5	4	3	2	1	SN.	
000														
001														
002														
003														
004														
005														
006														
007														
008														
009														

ENTER DESCRIPTION OF OUTPUT DATA

ENTER CARD COLUMNS TO BE PUNCHED AND INDICATE DECIMAL (▲).

INPUT CONTROL LINES

SYM.	FROM
a	
b	
c	
d	
e	
f	
g	
h	
i	
j	
k	
l	

ENTER CONDITION WHEN THE INPUT CONTROL LINE SHOULD BE IMPULSED, SUCH AS 0 COLUMN 45, ETC.

OUTPUT CONTROL LINES

SYM.	TO
A	
B	
C	
D	
E	
F	
G	
H	
I	
J	

ENTER FUNCTION TO BE IMPULSED, SUCH AS TRIP, IN PROGRAM SELECT 1, ETC.

INPUT-OUTPUT SELECTORS

PICK UP	S#	SELECT	COMMON	NON-SELECT
	1a			
	4			
	5a			
	5b			
	5c			
	6a			
	9a			
	9b			
	9c			
	10a			
	10b			
	10c			

ENTER CARD POSITION AND COLUMN WHICH INDICATES THE VALUE IS NEGATIVE.

ENTER POSSIBLE SIGN OF VALUE

Enter description of function or hub to be energized by or directed to the common hub if the selector is not picked up

Enter the function(s) which pick up selector, such as IPCa, 1 in col. 46 or combinations as B+ thru 1/46

ENTER POSITION TO BE PUNCHED IF VALUE IS NEGATIVE.

ENTER POSSIBLE SIGN OF VALUE: PLUS (+), MINUS (-) OR BOTH (±)

Enter description of function or hub to be directed in one of two ways

Enter description of function or hub to be energized by or directed to the common hub if the selector is picked up

ENTER THE NUMBER OF THE SELECTOR(S) TO BE PICKED UP, SUCH AS PU 8, ETC.

INPUT TRANSFER

IN (from)	
NO (to)	
EVEN (to)	
ALL (to)	

INDICATE CONTROL POSITION (IF ANY) WHICH CONNECTS NO, EVEN OR ALL TO THE IN HUB:
NO - NO INPUT DATA IS READ
EVEN - ONLY EVEN STORAGES ARE READ
ALL - ALL STORAGES ARE READ

PROGRAM SELECTS

NO.	IN FROM	POWER OUT TO
1		
2		

ENTER THE OUTPUT CONTROL LINE WHICH SHOULD IMPULSE THE ENTRY.

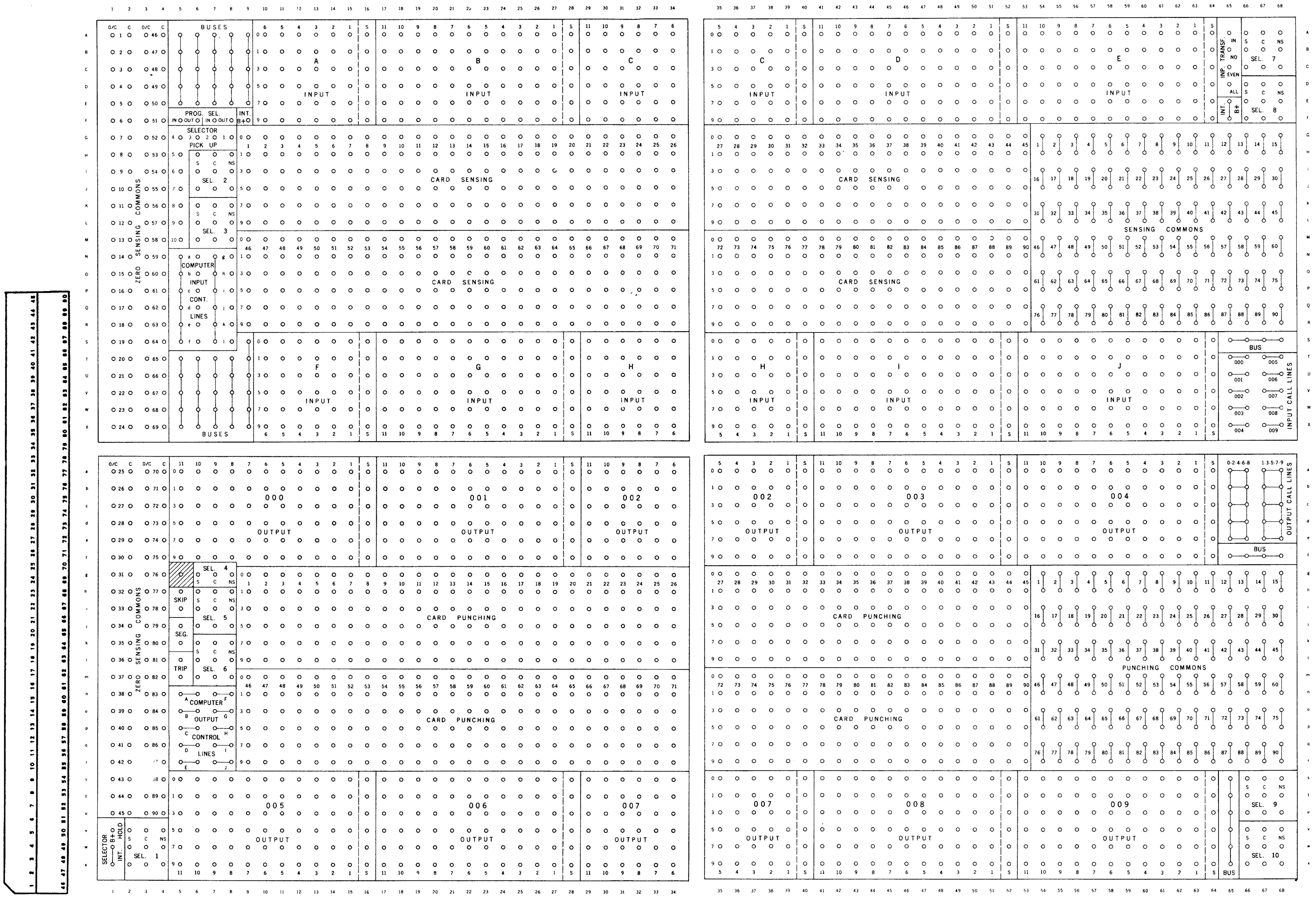
REMARKS:

UNIVAC FILE - COMPUTER

SINGLE CARD FEED
INPUT - OUTPUT PANEL

Remington Rand
DIVISION OF SPERRY RAND CORPORATION
TABULATING MACHINES
III - 38

APPLICATION:—



SECTION IV

STORAGE COMPONENTS

STORAGE COMPONENTS

INDEX - SECTION IV

PAGE NO.

STORAGE COMPONENTS

There are four main storage components of the Univac File-Computer: input-output storage, intermediate storage, high speed storage, and large capacity storage. Each of these types of storage is magnetic drum storage.

A magnetic drum is a cylinder, coated with a magnetic material, which is mounted on a shaft and rotated at a constant speed.

Data...numeric, alphabetic, or a combination...is recorded (stored) on the surface of the drum in the form of small magnetic spots. Magnetic drum storage is permanent storage; that is, once information is recorded on the drum it is always available. Turning off the power to the system, or a power failure, does not remove the magnetic spots from the drum. Stored data can be changed or removed from the drum by recording new data in the same location of the drum. Each class of the new data, such as alpha, numeric, space, etc., replaces the original data stored in this space.

The method of recording and reading information on the drum can be compared with the method of recording and reading information in a punched card. With a punched card, information is recorded and stored by holes punched in the card. Information from the card is available by "reading" the holes punched. With the magnetic drum, information is recorded and stored on the surface of the drum by small magnetic spots. Information on the drum is available by "reading" these magnetic spots. There are three important differences between punched card storage and magnetic drum storage:

1. Capacity- The punched card can store 90 characters or less; the large capacity magnetic drum can store 180,000 characters.

2. Reuse - Data stored on the magnetic drum can be readily altered.

3. Access - Any storage area on the drum is readily available.

The surface of a magnetic drum is divided into narrow bands called "channels" or "tracks". It is on these channels that data is recorded by the small magnetic spots. Associated with each channel of the magnetic drum are stationary read-write heads. As the drum revolves, these heads can "read" or "write" magnetic spots on a channel of the drum.

The internal machine code, or "language", of the Univac File-Computer is the seven level Univac code...that is a specific combination of seven possible pulses or magnetic spots represent a specific character, numeric or alphabetic, just as a specific combination of the six possible positions in a 90 column punched card represents a specific character. The Univac code has three major parts:

UNIVAC 7 LEVEL CODE

0 — Check Pulse: The verification circuits of the Univac File-Computer recognize only odd number codes as legitimate. The check pulse is automatically added to an even code character to make the code odd. If a pulse is dropped or added during transmission, the verification causes a parity error signal.

0
0 — Two character indicators: Two pulses utilized for alphabetic and special character codes.

0
0
0
0 — Four Binary Bits: In the binary system, all numbers are represented by a series of two symbols only... "0" and "1". Letters and special characters may also be represented by combinations of these two symbols. The Univac seven level code is based on the binary system. The "1" and "0" symbols correspond to the presence of a magnetic spot or pulse, and the absence of the pulse.

In binary notation, the decimal digits "0" through "9" are expressed:

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

The digit "1" in the binary system doubles in value each time it moves to the left. This system requires four binary "bits" to represent each decimal digit from "0" to "9".

The internal Univac code is based on the four bit binary notation of 1-2-4-8 to represent decimal digits. The Univac code, however, is an excess three binary code, i.e., the Univac code for 1 is binary code 4, for 2...binary 5, for 3...binary 6, etc.:

Decimal System	Pure Binary 8421	Univac Excess 3 Binary
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

This system is extended by the use of two more binary bits to handle the alphabet and special characters.

The Univac 7 level Code:

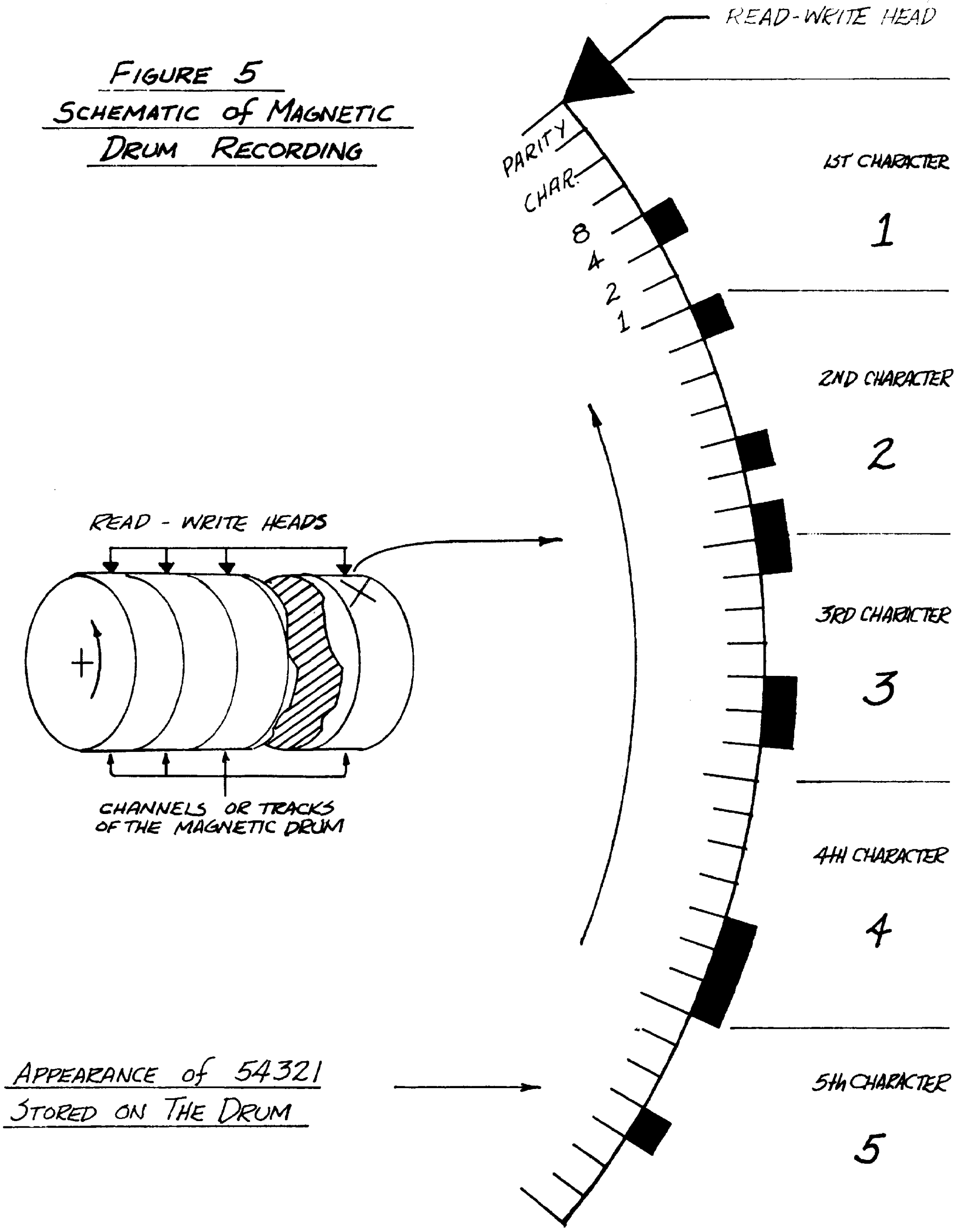
	CHECK PULSE	CHARACTER INDICATOR		8	4	2	1
0	x					x	x
1					x		
2	x				x		x
3	x				x	x	
4					x	x	x
5				x			
6	x			x			x
7	x			x		x	
8				x		x	x
9	x			x	x		
A	x		x		x		
B			x		x		x
C			x		x	x	
D	x		x		x	x	x
E	x		x	x			
F			x	x			x
G			x	x		x	
H	x		x	x		x	x
I			x	x	x		
J	x	x			x		

	CHECK PULSE	CHARACTER INDICATOR		8	4	2	1
K		x			x		x
L		x			x	x	
M	x	x			x	x	x
N	x	x		x			
O		x		x			x
P		x		x		x	
Q	x	x		x		x	x
R		x		x	x		
S	x	x	x		x		x
T	x	x	x		x	x	
U		x	x		x	x	x
V		x	x	x			
W	x	x	x	x			x
X	x	x	x	x		x	
Y		x	x	x		x	x
Z	x	x	x	x	x		
SPACE							x
IGNORE	x						
MINUS						x	
PLUS	x	x	x			x	x

Unassigned combinations of the Univac 7 level code are used for special characters.

Data is stored serially around the circumference of the channel of the magnetic drum in the Univac 7 level code. The least significant digits are read or stored first; the most significant digits last. This is shown in Figure 5.

FIGURE 5
SCHEMATIC of MAGNETIC DRUM RECORDING

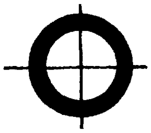


In addition to the four main storage components of the Univac File-Computer, which are all magnetic drum storage, there is a second type of storage, related to the input-output devices called buffer storage. Buffer storage is magnetic core storage and is used exclusively to transfer data from an input-output device to input-output drum storage or to transfer data from input-output drum storage to an input-output device.

A magnetic core is a small, doughnut shaped ring, which can be magnetized. Each core stores one bit of information. Bits of information are read into the cores by sending current through wires passing through the centers of the cores.

This current may, or may not, change the information in the core, depending on its polarity. That is, if the core, before receiving any pulse was positively magnetized in the "1" state, only a negative current pulse could change this core to the "0" state. Similarly, a core in the "0" state can only be changed to a "1" by applying a positive pulse. Information read into, stored in, or read out of the cores, is directly related to the bit code of the computer.

Magnetic core



The "1" or "0" state of a magnetic core is comparable to the presence or absence of a magnetic spot in a certain position on a magnetic drum. A number of cores are grouped together to store a character in the Univac 7 level code. The combination of "1" and "0" state of the cores in this group determines the character stored. Twelve of these groups of cores are in turn linked together to store input-output fields of twelve characters including sign.

1. THE SMALL OR HIGH-SPEED DRUM

The high-speed storage drum is 15" long and 4-3/8" in diameter. It revolves at a speed of 12,000 revolutions per minute. Any desired location on the drum can be found within an average of 2.5 milliseconds. Each track on the surface of the high-speed drum is divided into 10 units or words of 12 characters including sign. Each one of these units is identified by its "address"; that is, its specific location on the drum.

The high-speed drum is divided into 3 major portions:

1. Input-Output Storage
2. Intermediate Storage
3. High-Speed Storage

FIGURE 6

THE SMALL OR HI-SPEED DRUM

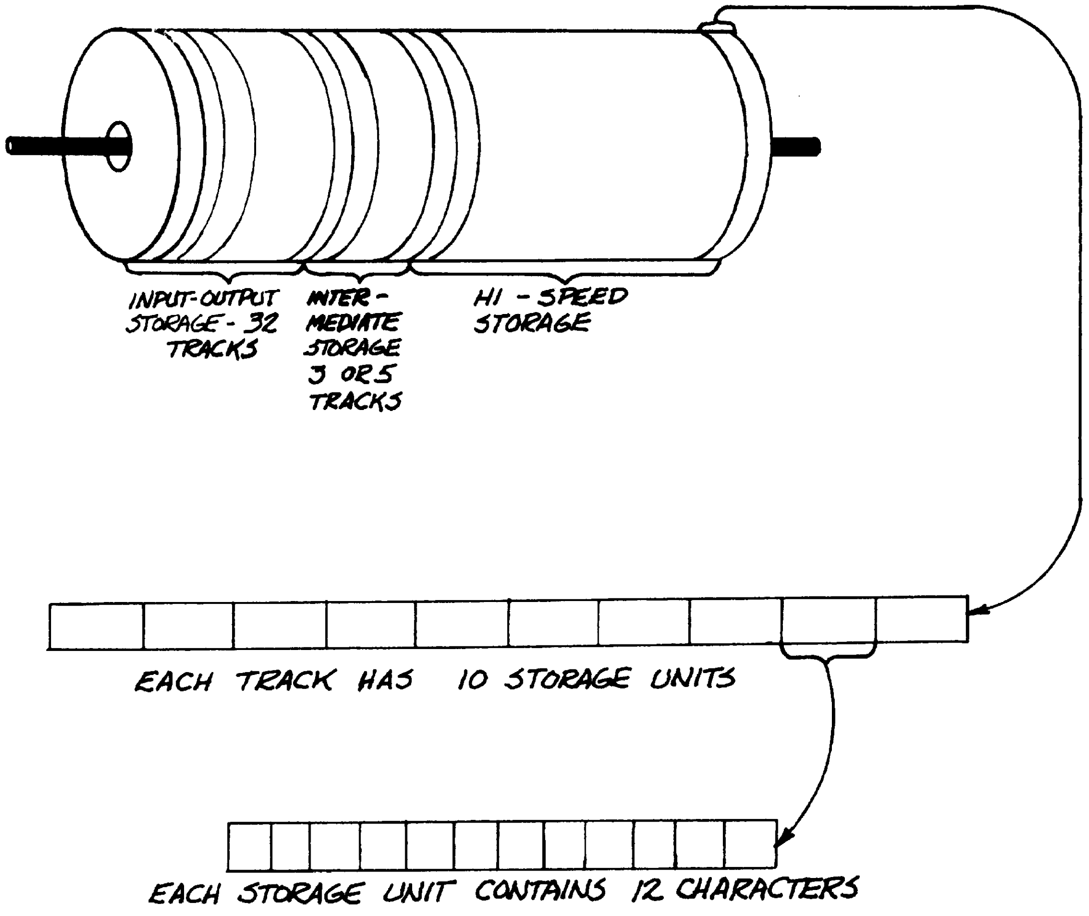
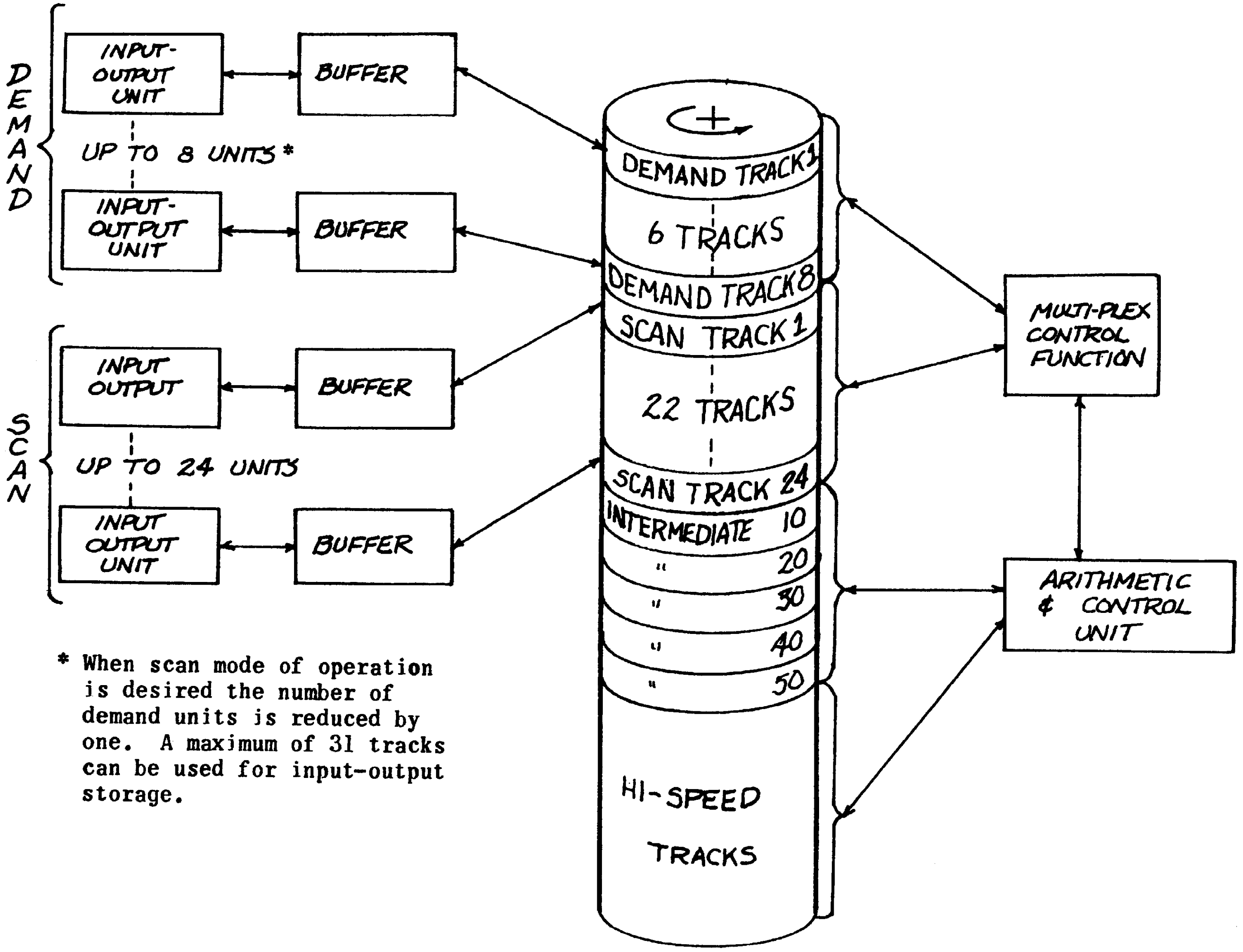

INPUT-OUTPUT STORAGE - 32 TRACKS
INTER-MEDIATE STORAGE 3 OR 5 TRACKS
HI - SPEED STORAGE
EACH TRACK HAS 10 STORAGE UNITS
EACH STORAGE UNIT CONTAINS 12 CHARACTERS

FIGURE 7
FUNCTIONAL DIAGRAM of the Hi-SPEED STORAGE DRUM



* When scan mode of operation is desired the number of demand units is reduced by one. A maximum of 31 tracks can be used for input-output storage.

(1) INPUT-OUTPUT STORAGES

A. Use of input-output storage
Input-output storage units have three basic uses:

1. The major function of the input-output storage units is to make available to the Univac File-Computer system data from input-output devices and to make available to the input-output devices results and data from the Univac File-Computer system.

2. Input-output storage units may be used to store intermediate results.

3. Input-output storage units are addressable, both through external (plugboard) and internal (stored) instructions. These units may also be used for internal instructions.

B. Capacities and Designations of Input-Output Storage

The input-output storage tracks are included as part of the basic Univac File-Computer.

There are 32 tracks of input-output storage on the high-speed drum...1 track for each of the 8 possible demand input-output devices and 1 track for each of the 24 possible scan input-output devices. If the computer is operating with scan units, only 7 demand input-output tracks are available. Each track has a capacity of 10 storage units...12 characters including sign. The addresses of these storage units on each track are:

000,001,002,003,004,005,006,007,008, and 009.

Thus, there are actually 32 storage units on the high-speed drum, identified as 000, 32 identified as 001, etc. However, at any one time, only one of these input-output storage tracks is available to the arithmetic and control section of the system. (See preceding functional diagram of the high-speed storage drum). The multiplex function controls which input-output track is to be connected with the main system at one specific time.

When an input-output storage unit is called for, the related storage unit from the input-output track, which is connected with the system (through the multiplex function) at that time will be delivered to the arithmetic and control section of the system. For example: If, while demand input-output unit 4 is connected to the system, storage unit 007 is called for, the data from storage unit 007 on demand track 4 of input-output storage, will be delivered to the arithmetic and control section of the system. If a result is delivered to storage unit 007 while demand input-output unit 4 is connected with the system, that result will be delivered to storage unit 007 on demand input-output track 4. (Refer to figure 15 page V - 73.)

Since each input-output device has a separate track of input-output storage, it is possible to:

1. Process the information from the input-output track which is connected to the system and to deliver results back to that track. At the same time:

2. Read input data from some of the input-output devices to their respective input-output storage tracks, and:

3. Deliver results and data from other input-output storage tracks to their respective input-output devices.

C. Buffer Storage

Associated with each input-output unit included in a Univac File-Computer system, is a temporary storage... "buffer storage". This buffer storage is magnetic core storage and is used exclusively to transfer data from an input-output device to its related input-output storage track, and to transfer data from an input-output storage track to its related input-output device.

Included in the buffer storage unit, associated with each input-output device, is a "decoding" mechanism. The internal language of the Univac File-Computer is Univac 7 level code. As data is read into the buffer from an input-output device, it is decoded from its code (such as the 6 position punched card code) to the Univac 7 level code. As data is read from the buffer to the input-output device, the Univac 7 level code is decoded to the code of the input-output unit.

Buffer storage is necessary because of the variable speed between the input-output device and the high-speed drum. The high-speed drum revolves at a constant rate of 12,000 revolutions per minute. The speed of the input-output devices is variable. It would be impossible to match the speed of input-output devices (such as on on-line key punches) with the constant speed of the high-speed drum.

(2) INTERMEDIATE STORAGE

A. Use

Intermediate storage units are addressable through plugboard programming. As the name implies, intermediate storage fields are used to store intermediate data and results. These storage fields are also used for storing constant values.

B. Capacities and Designations

Three intermediate storage tracks are included as part of the basic Univac File-Computer.

The designations or "addresses" of these three tracks are 10, 20 and 30. Each track has a capacity of ten storage units...12 characters including sign. The addresses of these units are:

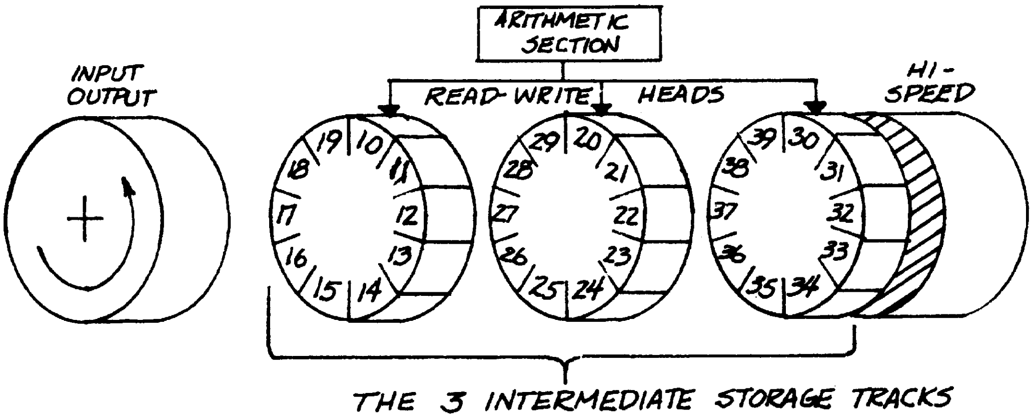
Track 10: 10,11,12,13,14,15,16,17,18 and 19.

Track 20: 20,21,22,23,24,25,26,27,28 and 29.

Track 30: 30,31,32,33,34,35,36,37,38 and 39.

FIGURE 8

INTERMEDIATE STORAGE



Two additional tracks of intermediate storage are available for expansion. The addresses of these units are:

Track 40: 40,41,42,43,44,45,46,47,48 and 49.

Track 50: 50,51,52,53,54,55,56,57,58 and 59.

(3) HIGH-SPEED STORAGE

A. Use

High-speed storage units are addressable only through internal (stored instructions) programming. High-speed storage units may be used to:

1. Store internal instructions.
2. Store intermediate results.
3. Store reference tables, rates, etc.
4. Store constant values.
5. Store any combination of the above four items.

High-speed storage will not be included in the model of the machine which this manual describes. Therefore, a detailed discussion of its capacity and usage has been omitted.

2. THE LARGE CAPACITY DRUM

The large capacity storage drum is 17" long and 17" in diameter. It revolves at a speed of 1750 revolutions per minute. A specific location on the drum can be located within an average of 17 milliseconds.

The large capacity drum has a capacity of 180,000 characters. From 1 to 10 large capacity drums may be included in a Univac File-Computer system.

A. Use

The large capacity, random access, storage drum is an exclusive feature of the Univac File-Computer system.

The primary reason for engineering the Univac File-Computer is to enable one to keep a current balance for a large number of items with high volume activity. The system is well suited to handle inventory and distribution types of problems...problems where balances must be kept on a current basis, where amounts must be allocated to various accounts or classifications and balances brought forward.

The large capacity storage drum is used to store large quantities of information, which are repeatedly referred to or changed. Some of the uses of the large capacity drum are:

1. Store a continuing record, such as inventory balance and post entries affecting inventory.
2. Store a volatile record, such as a schedule and post changes and progress to it.
3. Classify and accumulate large masses of unsorted data...data reduction.
4. Store tables and rates for reference in calculations.
5. Serve as a storage media for distribution and summarization of results, so that reference, calculation, and summarization can be accomplished in a single operation.
6. Serve in combination of items 1 through 5

B. Capacities and Designations

The large capacity storage drum has a capacity of 180,000 characters...numeric, alphabetic, or a combination of numeric and alphabetic characters. From 1 to 10 large capacity storage drums may be included in a Univac File-Computer system. Ten drums would provide 1,800,000 characters of storage. If fewer than 10 large capacity drums are included in an initial Univac File-Computer system, additional drums can be added, as required, to a maximum of 10.

The surface of the large capacity drum is divided into three major "drum sections". Each section is divided into 100 tracks, or channels. Each channel has a capacity of 600 characters -- 3 drum sections x 100 channels per section x 600 characters per channel = 180,000 characters per drum.

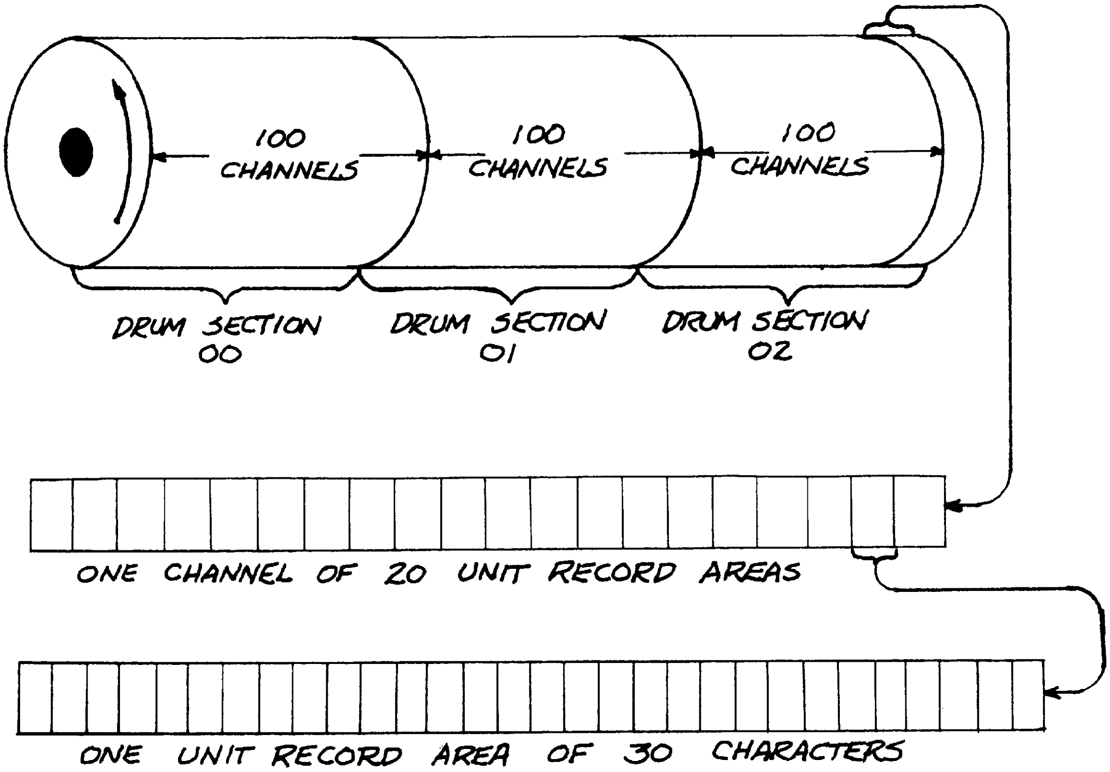
At the time a large capacity storage drum is manufactured for a Univac File-Computer system, the 600 characters of each channel on the drum are divided into specific "unit record areas". There are 11 unit record area lengths available:

12,15,20,24,30,40,50,60,75,100, or 120 characters.

The length of the unit record area is fixed for all large capacity drums included in one Univac File-Computer system. The data for the applications to be processed by a Univac File-Computer system determines which length unit record area is required.

The number of unit record areas available on one drum depends upon the character length of the unit record area. For example, with a unit record length of 30 characters, each channel of the drum contains 20 unit record areas: 600 characters per channel ÷ 30 characters, per unit record = 20 unit record areas per channel. This drum then contains 6,000 unit record areas 20 unit record areas per channel x 300 channels per drum = 6,000 unit record areas per drum.

FIGURE 9
LARGE CAPACITY DRUM



This figure illustrates a large capacity drum with a 30 character unit record area. The upper portion represents the drum and the 300 channels on the drum; next, a representation of the 20 unit record areas within one channel; and lastly, the 30 characters within one unit record area.

The following table lists the 11 possible unit record area lengths and the related number of unit records available with 1 through 10 large capacity drums.

UNIT RECORD LENGTH	NUMBER OF UNIT RECORDS ON:										
	EACH CHANNEL	1 Drum	2 Drums	3 Drums	4 Drums	5 Drums	6 Drums	7 Drums	8 Drums	9 Drums	10 Drums
12	50	15000	30000	45000	60000	75000	90000	105000	120000	135000	150000
15	40	12000	24000	36000	48000	60000	72000	84000	96000	108000	120000
20	30	9000	18000	27000	36000	45000	54000	63000	72000	81000	90000
24	25	7500	15000	22500	30000	37500	45000	52500	60000	67500	75000
30	20	6000	12000	18000	24000	30000	36000	42000	48000	54000	60000
40	15	4500	9000	13500	18000	22500	27000	31500	36000	40500	45000
50	12	3600	7200	10800	14400	18000	21600	25200	28800	32400	36000
60	10	3000	6000	9000	12000	15000	18000	21000	24000	27000	30000
75	8	2400	4800	7200	9600	12000	14400	16800	19200	21600	24000
100	6	1800	3600	5400	7200	9000	10800	12600	14400	16200	18000
120	5	1500	3000	4500	6000	7500	9000	10500	12000	13500	15000

Each unit record of a large capacity drum has a specific location on the drum...it is within one of the three drum sections, on one of the 100 channels within the drum section, and in one of the unit record areas within the channel.

The specific location of a unit record area on the drum is the "address" of that unit record.

The function of the address for each unit record is the same as the function of a street address...to identify one specific location out of many locations.

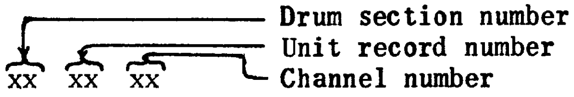
The address numbering system of the large capacity drum is a six digit address composed of:

1. Drum section number................2 digits

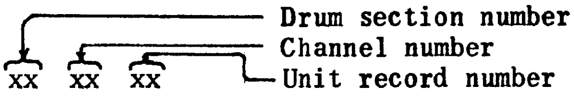
2. Unit record area number............2 digits

3. Channel number.....................2 digits

The structure of the address can be specified for a system as:



or.

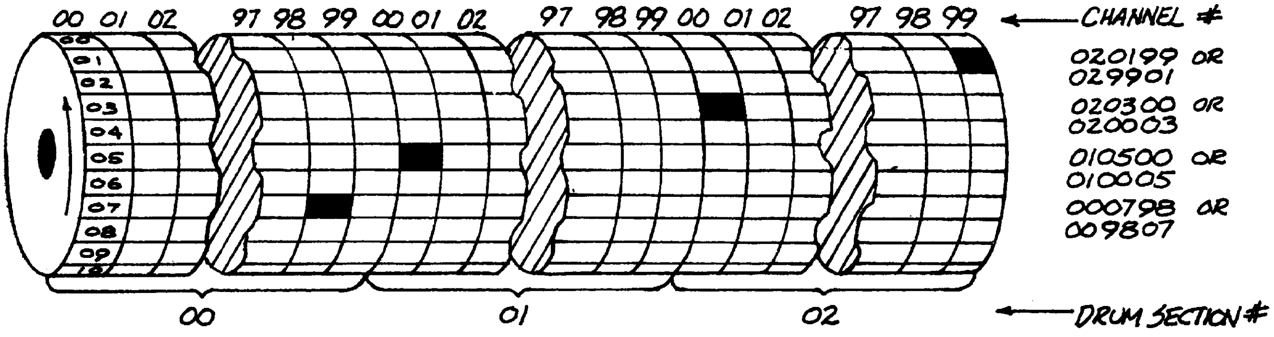


Drum Section Number: There are three sections on each large capacity storage drum; and up to ten drums may be included in a Univac File-Computer system. The drum section numbers of the address range from 00 through 29. The drum section numbers of the first large capacity drum are 00, 01, and 02; the second drum 03, 04, 05;.... the tenth drum are 27, 28, 29.

Unit Record Area Number: The unit record area address numbering depends on the unit record area length. (See preceding unit record area table.) The unit record area length determines the number of unit record areas per channel. The address numbering system is based on the number of unit record areas per channel. If the unit record area length of the large capacity drums included in a Univac File-Computer system is 12 characters, there are 50 unit record areas per channel. The range of the unit record area address numbers would be 00 through 49. If the unit record area length is 120 characters, the unit record area address numbers would be 00 through 04, since each channel would contain 5 unit record areas.

Channel Number: Each drum section contains 100 channels. The channel numbers of the address range from 00 through 99.

FIGURE 10
ILLUSTRATION of the LARGE CAPACITY DRUM
ADDRESS NUMBERING SYSTEM



The standard unit record length of the large capacity drum can be divided into a maximum of 20 fields by external wiring of a drum connection panel. Each of the 20 fields may have a capacity of from 1 to 12 characters including sign. The identification, or address, of these 20 fields are:

F0, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16, F17, F18, and F19.

C. How is data stored on, or read from, the large capacity drum?

Associated with each of the 100 channels of the 3 drum sections of each large capacity drum, which is constantly rotating, is a stationary read-write head. As the drum revolves, the unit record areas on each channel are continuously passing under the related read-write head.

Data is stored on the drum, or read from the drum though the channel read-write heads in conjunction with an address register switch. This switch connects any one of the channel read-write heads to the revolver through its associated circuitry.

(a) The address register and the staticizer: Associated with the large capacity drums is a 6 digit register called the "address register". The function of the address register is to receive the 6 digit address of a specific unit record area on the large capacity drums. The address is automatically transferred from the address register to the staticizer. The staticizer is a static register which controls the connection of the address register switch to a specific channel read-write head.

To perform the actual reading or writing of a specific unit record area, a "read unit record" or "write unit record" instruction is necessary. First, the address of the desired unit record must be placed in the address register. From there it is automatically transferred to the staticizer and the address register switch then connects the proper read-write circuits to the channel within the drum section specified by the address in the staticizer. Then a read unit record or write unit record instruction is given to the machine and the unit record number portion of the address in the staticizer controls the location of the specific unit record within the desired channel.

In many applications, the identification numbering system of the data to be stored on the drum will not be compatible with the address numbering system of the large capacity drums; or it may not be possible to code the input data with the complete address of the related unit record area. In these instances, a "channel search" instruction is used to locate and read the desired unit record area. (See section E following.)

(b) The revolver: The revolver is the means of rapid access to the large capacity storage drums. At any one time during a program, one unit record of the large capacity drums is available to the arithmetic and control unit of the system from an additional channel on the drum called the "revolver". The revolver is divided into the same unit record length as the other 300 storage channels on the drum. The twenty possible large capacity drum storage fields, F0 through F19, are directly associated with the revolver.

With a read unit record instruction, the data within the unit record, specified by the address delivered to the address register, is read from the drum and written on the revolver. That data can then be called upon by the arithmetic and control section of the system through the twenty storage field, FO through F19. These fields are commonly called the "revolver fields". New data,changed balances, etc., can be placed in that unit record through these revolver fields.

With a write unit record instruction, the data is read from the revolver and written in the unit record area, specified by the address delivered to the address register.

D. Read Unit Record and Write Unit Record - Diagram of Circuitry.

Read unit record is the instruction which reads the data from a specified unit record area and writes that data on the revolver. The data in that unit record area is then available to the arithmetic and control unit of the system.

Write unit record is the instruction which reads the data from the revolver and writes that data in a specified unit record area of the large capacity drums. It provides the means of storing data and results on the large capacity drums.

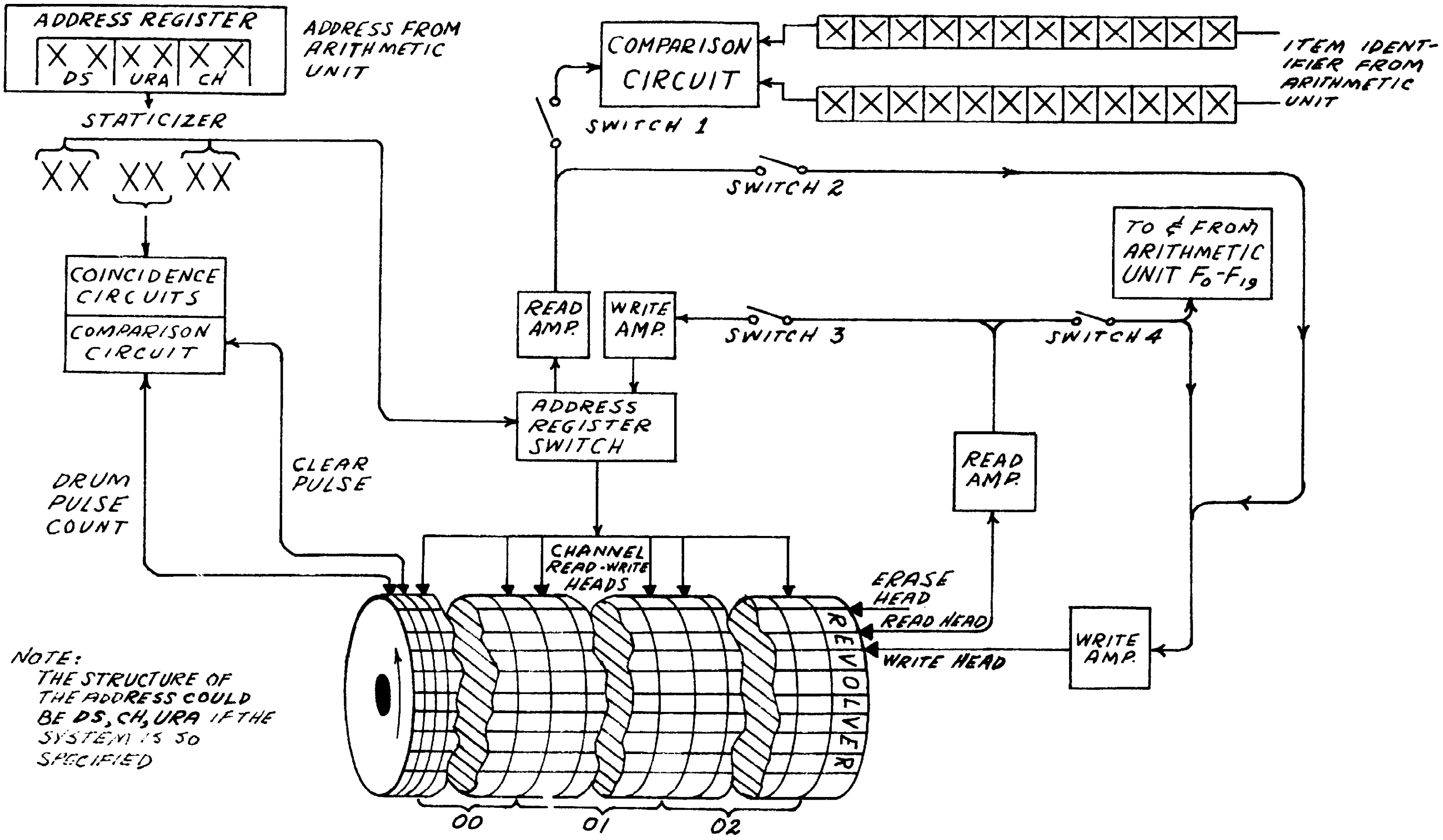
Before either the read or write unit record instruction is given, the address of the desired unit record must be given to the address register. From there it is automatically transferred to the staticizer and the address register switch connects the drum section and channel specified by the address in the staticizer to the associated read circuitry.

If a prior read or write unit record instruction is in process when a new address is delivered to the address register, the automatic transfer of the address from the address register to the staticizer is delayed until the prior instruction is completed. When the prior instruction is completed the address is then automatically transferred from the address register to the staticizer and the indicated switching will immediately begin.

The unit record number portion of the address in the staticizer controls switches which locate the specific unit record within the channel.

FIGURE 11

BLOCK DIAGRAM OF THE LARGE CAPACITY DRUM CIRCUITRY



NOTE: THE STRUCTURE OF THE ADDRESS COULD BE DS, CH, URA IF THE SYSTEM IS SO SPECIFIED

Operation of Switches in Figure 11

Operation	Switch 1	Switch 2	Switch 3	Switch 4
Read URA.	Open	Closed—Opens when Switch 4 closes.	Open	Open—Closes when desired URA is located.
Write URA	Open	Open	Open—Closes when desired URA is located on the drum	Closed
Channel Search	Open—Closes at identifier time only	Closed—Opens when Switch 4 closes	Open	Open—Closes when identifiers match & last character of URA has been read
	REVOLVER FIELD ACCESS			
Use Field (FO-F19)	Open	Open	Open	Closed
Deliver result to Field (FO-F19)	Open	Open	Open	Closed—Opens only during specified Field (FO-F19) time

This block diagram of the large capacity drum circuitry illustrates the relation of the storage channels on the drum, the revolver, and the arithmetic section of the system.

Note that the revolver has one read head and one write head. The write head is spaced one unit record length before the read head. In addition, an erase head is located approximately 12 characters after the read head. This head is a permanent magnet.

The address of the desired unit record area is placed in the address register and then automatically transferred to the staticizer. The switching immediately begins to the read-write head of the channel specified by the drum section and channel number portion of the address in the staticizer. The drum is constantly revolving under these heads. When a read unit record instruction is given the unit record number portion of the address in the staticizer is compared with timed drum pulses from the drum. Switch 2 and switch 4 are controlled by a coincidence pulse which is emitted when a match occurs. While the specified unit record is passing under the read-write head, switch 2 is closed. The data from that unit record area can then pass through the read-write head and the address register switch, across the closed switch 2, to the write head of the revolver. When the drum has revolved so that the last character from the specified unit record is written on the revolver, switch 2 is opened and switch 4 is closed. Since the drum is rotating, the desired data is now in the unit record area of the revolver that is passing under the revolver read head. That data passes through the revolver read head, across the closed switch 4, back to the write head of the revolver. Switch 4 remains closed, and the data is continuously recirculated in this loop.

The moment switch 4 closes after a read unit record instruction, that is, the moment the desired data is written once on the revolver, it is available to the arithmetic and control section of the system; and since that data is constantly recirculated it is continually available at the revolver read head.

New data, changed balances, etc., that are to be stored on the drum are first written on the revolver through one of the 20 revolver fields, F0 through F19. This writes the new or changed information on the unit record area passing under the revolver write head at that time. As the drum continues to revolve, the new or changed data is written in the unit record area of the revolver and is available to the arithmetic and control section of the system.

When a write unit record instruction is given Switch 3 is controlled by the comparison between the unit record number portion of the address and the timed drum pulses. When the desired unit record area of the drum is passing under the read-write head, switch 3 is closed. The data from the revolver can then pass through the revolver read head, across closed switch 3, to the channel read-write head, and is written in the specified unit record area.

The read and write unit record instructions are equipped with interlocks, making it possible to perform calculations while the read or write function is taking place. If one of the revolver fields F0 through F19 is addressed before, the read or write function is completed, the instruction is initiated after the completion of the read or write function.

E. Channel Search

In previous references to the address delivered to the address register, it was assumed that the complete address of the specific unit record area desired was available; that is, that it was possible to code the input data with the address of its related unit record area on the large capacity drums, or that it was possible to correlate the identification numbers, such as item number, of the input data with the address numbering system of the large capacity drums.

Very often, it will not be possible to do either one of these things. How then, can a specific unit record area be located when the complete address of that area is not known?

Two types of channel search instructions provide the means of accomplishing this: channel search equal and channel search unequal. With either of these instructions a partial address of the desired item, preferably drum section and channel number, is developed from the designation of the input data. The identification of each item is stored in its unit record area. It is then possible to deliver the partial address of drum section and channel, plus a starting unit record number to the address register, and with a channel search function, compare the item identification of each unit record area in that channel with the identification of the item desired until the corresponding unit record area is found.

When the desired unit record area is found, the data from that area is "trapped" on the revolver, making it constantly available by continuously recirculating the data on the revolver.

With either channel search instruction a complete six digit address of drum section, unit record, and channel (xx xx xx) must first be delivered to the address register. A partial address of drum section and channel alone (xx xx or xxxx), depending on the address structure specified for the system, would cause a parity error signal. With channel search, therefore, it is necessary that a unit record number be added to the partial address of drum section and channel and the result delivered to the address register before the channel search function is initiated. Note: The address register will treat zeros and space codes alike. A partial address of drum section and channel with space codes in the URA portion of the address will be treated as 00 by the address register and the staticizer.

The unit record area within a channel, at which either type of channel search is started, is determined by the unit record number portion of the address in the address register. The end of channel search is determined by the drum index pulse, that is at the end of the last unit record area within a channel. Normally zeros or space codes will be added to the partial address of drum section and channel and delivered to the address register, so that the channel search will start with the first unit record area on the channel. For example, if with a Univac File-Computer system in which the fixed URA length is 30 characters...if the unit record number portion of the address were 05, the channel search would start with unit record area 05; and if matching designation were not found, would end with unit record area 19. Unit record areas 00 through 04 would not be searched.

The two channel search instructions are channel search equal and channel search unequal.

(a) Channel search equal: With the channel search equal instruction it is possible to search a selected channel for a unit record area with stored indentification which matches the desired indentification and to "trap" that entire unit record on the revolver. If a matching identification is not found the last unit record area in the channel is "trapped" on the revolver. With this instruction it is also possible to search a selected channel for a unit record area with stored identification which matches either of 2 identifiers and to trap the entire unit record area with matching identification on the revolver. If neither identifier is found the last unit record area in the channel is trapped.

The channel search equal instruction controls switches, which will open and close under control of the unit record number portion of the address in the staticizer and the comparison of the identifiers. Refer to schematic drawing of the large capacity drum circuitry (Figure 11).

Before a channel search equal instruction is given, the partial address of drum section and channel plus a starting unit record number is delivered to the address register. From there it is automatically transferred to the staticizer and the address register switch connects the read-write head of the specified channel to the associated read circuitry. Then, when the channel search equal instruction is given, switch 1,controlled by the field identification on the revolver, is closed, and switch 2, controlled by the starting address, is also closed.

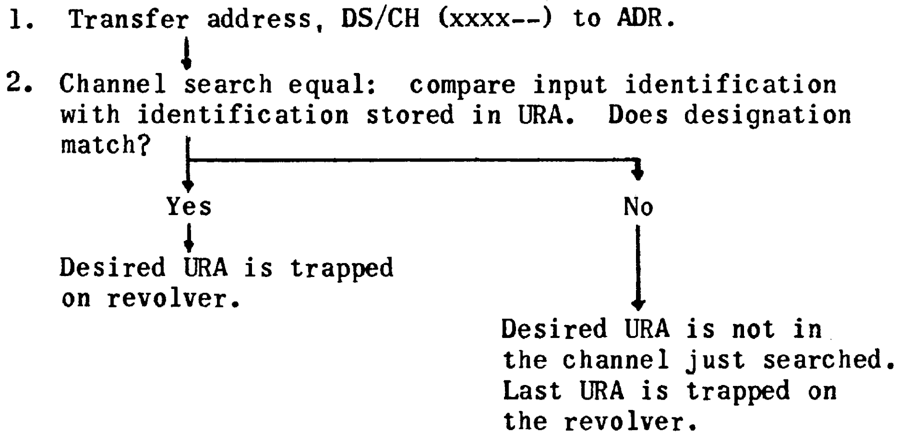
When the unit record area, specified in the address, starts to pass under the read-write head switch 4 is opened. The opening of switch 4 prevents the continuous recirculation of the data on the revolver. The closing of switch 1 permits the identification data in each unit record area to pass to the comparison circuits to be compared with the identification of the desired item. The closing of switch 2 permits the data from the unit record area of the drum to pass to the write head of the revolver...thus, as the drum revolves, the data from each successive unit record area of the channel being searched is written on the revolver. When the identification from a unit record area on the channel being searched matches the desired identification in the arithmetic unit comparator, switches 1 and 2 are opened at the end of that unit record area and switch 4 is closed. With the closing of switch 4, the data of the desired unit record is "trapped" on the revolver.

When a matching unit record is found, its location on the channel is "remembered" until a new address is entered in the address register. Therefore, it is possible to find a unit record by channel search, change some of the data, such as balances, etc., and return the changed information to the same location on the drum by a write unit record instruction.

If there is no matching identification in the channel searched, switches 1 and 2 are opened at the drum index pulse--the end of the last unit record area on the channel--and switch 4 is closed. Thus, the last unit record area is "trapped" on the revolver.

If a matching unit record is found during a channel search equal instruction with one identifier, the machine will emit a signal which is used to direct the machine to the next operation. If a matching record is found during a channel search equal instruction with two identifiers, the machine will emit one signal if the trapped URA matches the first identifier or another signal if the trapped URA matches the second identifier. The machine can, therefore, be directed to two different operations depending on which identifier is found. If no matching unit record area is found during a channel search equal instruction, a third signal will be emitted by the machine. This signal is used to direct the machine to the next operation desired if no matching URA is found.
Note: In all following examples, the address numbering structure is shown as DS/CH/URA. This could, of course, be DS/URA/CH, depending on the structure specified for the particular system.

Following is a flow chart of a channel search equal operation with one identifier: (Actual programming is discussed in Section V.)



1. Transfer address, DS/CH (xxxx--) to ADR.
2. Channel search equal: compare input identification with identification stored in URA. Does designation match?

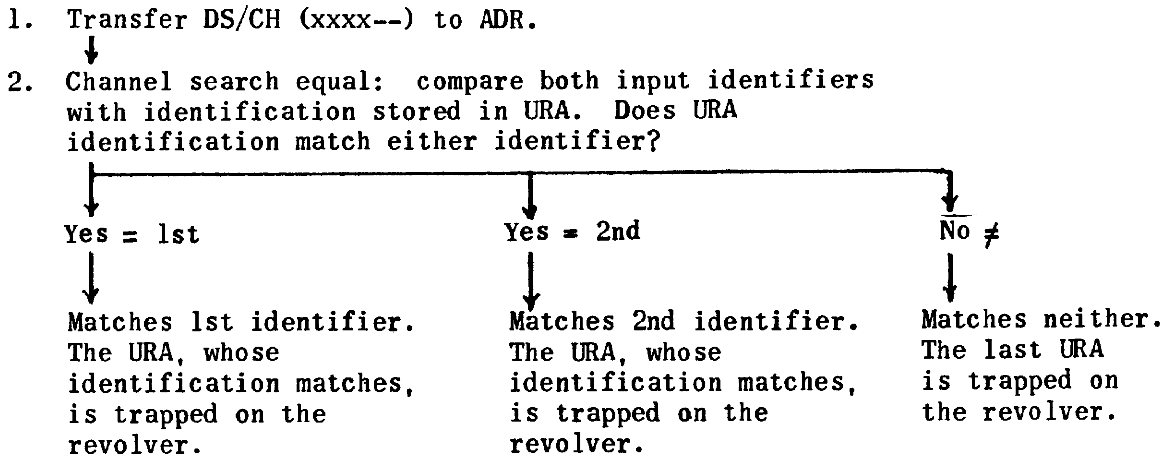
Yes

Desired URA is trapped on revolver.

No

Desired URA is not in the channel just searched. Last URA is trapped on the revolver.

Following is a flow chart of a channel search equal operation with two identifiers:



1. Transfer DS/CH (xxxx--) to ADR.

2. Channel search equal: compare both input identifiers with identification stored in URA. Does URA identification match either identifier?

Yes = 1st

Matches 1st identifier. The URA, whose identification matches, is trapped on the revolver.

Yes = 2nd

Matches 2nd identifier. The URA, whose identification matches, is trapped on the revolver.

No ≠

Matches neither. The last URA is trapped on the revolver.

The maximum number of characters (excluding sign) that can be compared during a channel search operation is 11. Occasionally, an application will require that the identification of the items to be located by channel search is more than 11 characters. When this occurs, it is impossible to compare the full identification of the desired item during a channel search equal operation. However, it is possible to locate the desired item through a channel search equal operation in conjunction with programmed comparing.

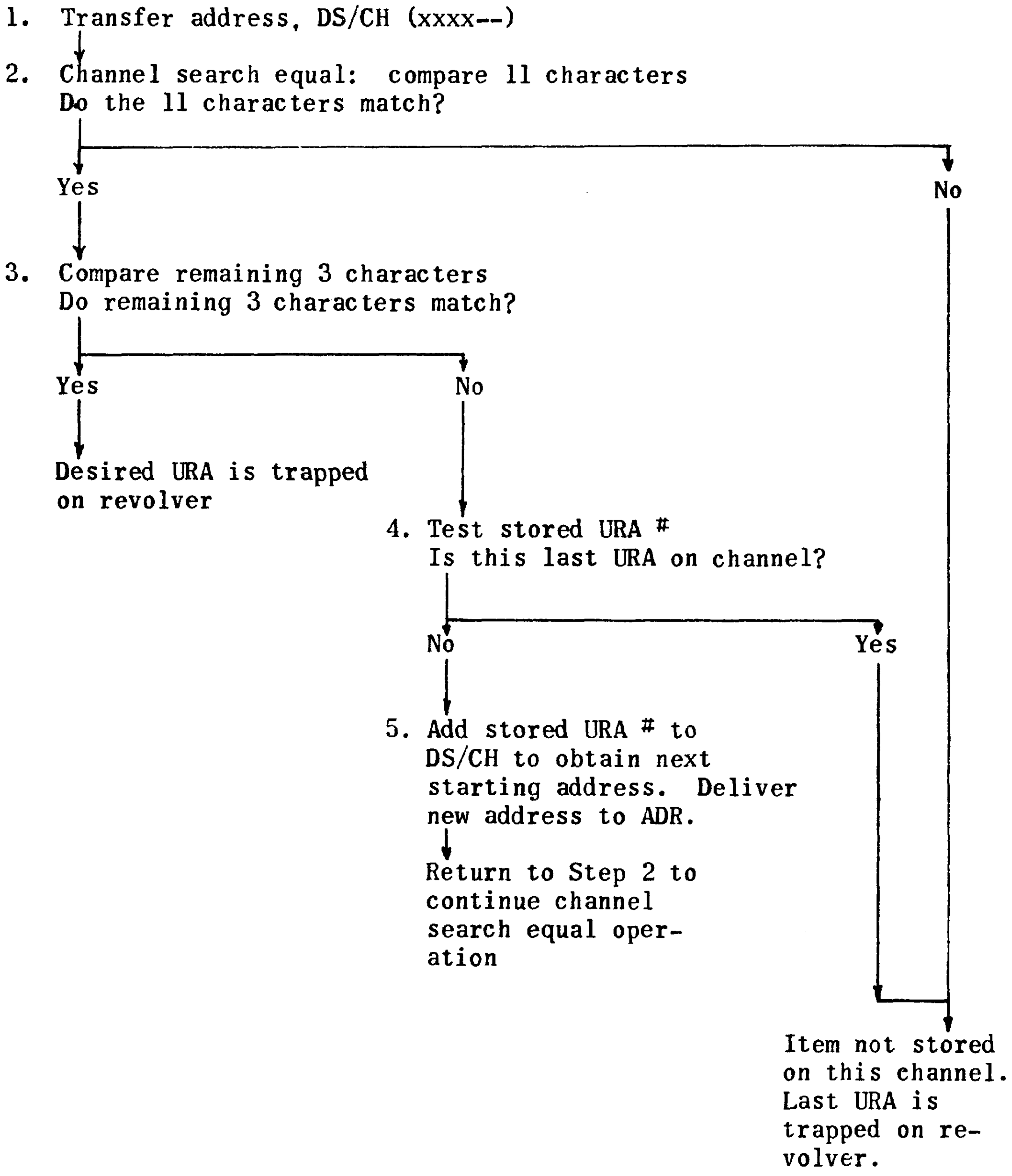
Eleven characters of the item designation are compared during a channel search equal operation. If a match is found, the remaining characters of the designation are then compared by programmed comparison. If the remaining characters also match, the unit record area "trapped" on the revolver is the desired one. If they do not match, the channel search equal operation is resumed at the next unit record area in that channel.

Following is a flow chart of a channel search equal operation for an item with a 14 character identification.

(Actual programming is discussed in Section V.) The full identification of the item is stored in its unit record area. In addition, the unit record number of the following unit record area on that channel is also stored. Then, to perform the channel search, the partial address of drum section and channel number is transferred to the address register. The channel search equal is then started, comparing the first 11 characters of the desired item with the first 11 characters of the identification stored in the URA. Since zeros or spaces are in the URA position the channel search will start with the first unit record area on the channel.

If a match of these 11 characters of the identification is found, the machine will emit a signal and that unit record area will be "trapped" on the revolver. The remaining three characters of the identification are then compared. If they also match, the desired unit record area is available on the revolver. If they do not match, the unit record area trapped on the revolver is not the desired one. The search of the channel must, therefore, be continued. To do this, the unit record area number stored in the unit record trapped on the revolver is tested. If this is not the last unit record on the channel, the stored unit record number (the number of the next unit record area in that channel) is added to the partial address and entered in the address register. The channel search equal instruction is again called for and the search resumes at the next unit record area. This sequence continues until the desired item is found in the channel, or until the entire channel has been searched without locating the item.

FLOW CHART OF CHANNEL SEARCH FOR ITEMS WITH 14 CHARACTER DESIGNATIONS



1. Transfer address, DS/CH (xxxx--)

2. Channel search equal: compare 11 characters
Do the 11 characters match?

Yes

No

3. Compare remaining 3 characters
Do remaining 3 characters match?

Yes

No

Desired URA is trapped on revolver

4. Test stored URA #
Is this last URA on channel?

No

Yes

5. Add stored URA # to DS/CH to obtain next starting address. Deliver new address to ADR.

Return to Step 2 to continue channel search equal operation

Item not stored on this channel. Last URA is trapped on revolver.

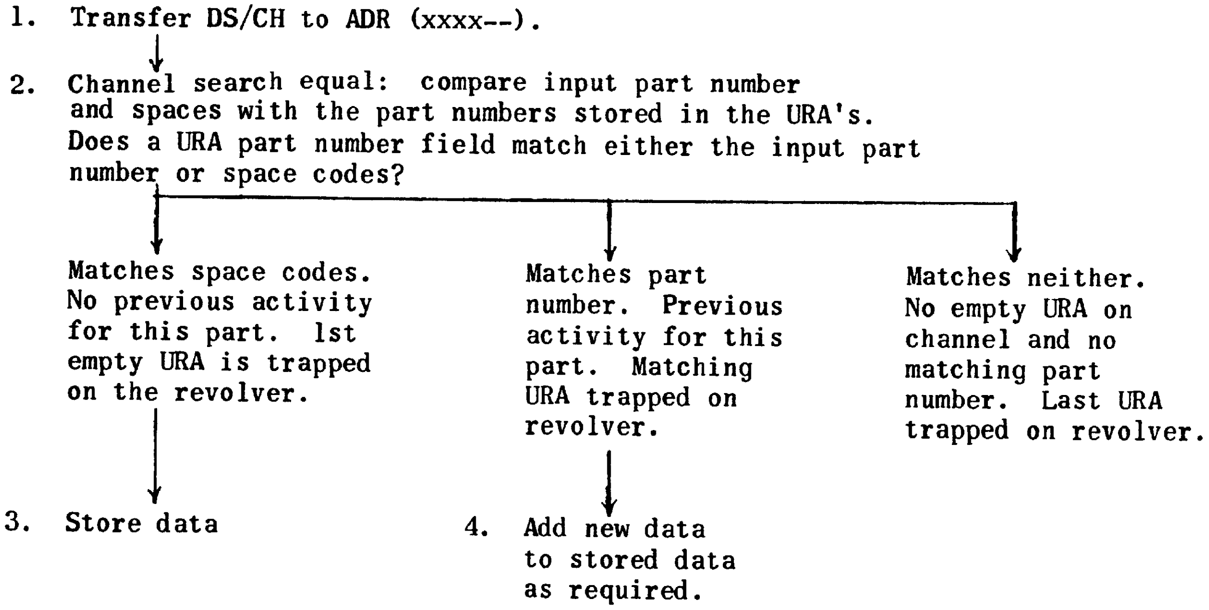
Note: Complete designation stored in URA. URA number of following URA is also stored.

Note: The channel search equal instruction with two identifiers is usually used when periodically loading and/or making additions to a volatile record on the large capacity drums. The identification of the input data is used as one identifier and space codes or zeros as the second.

As the items are processed they are successively written in the URA's --00-01,02, etc.--of the channel designated by their partial address. The channel search equal instruction is initiated for each item, starting with URA 00 of its designated channel, looking either for a matching designation or the first empty URA on the channel. The first item to be processed for a specific channel will produce a signal indicating a match has been found and trapped on the revolver for the second identifier--the space codes. URA 00, the first empty URA on the channel, is trapped on the revolver. The data for that item is then stored in URA 00. The next item for that specific channel will be stored in URA 01 since it will be trapped as the next empty URA when processing the item; the third item will be stored in URA 02, etc.

When an item is processed that has had previous activity--one or more documents for the same item have already been processed and stored-- the channel search operation will produce a signal indicating that a match has been found and trapped on the revolver for the first identifier--the designation of the item. The data for the item being processed is then added to the data for the item already stored on the drum.

Following is a flow chart of this channel search equal operation.



1. Transfer DS/CH to ADR (xxxx--).

2. Channel search equal: compare input part number and spaces with the part numbers stored in the URA's. Does a URA part number field match either the input part number or space codes?

Matches space codes. No previous activity for this part. 1st empty URA is trapped on the revolver.

Matches part number. Previous activity for this part. Matching URA trapped on revolver.

Matches neither. No empty URA on channel and no matching part number. Last URA trapped on revolver.

3. Store data

4. Add new data to stored data as required.

(b) Channel search unequal: With the channel search unequal instruction it is possible to search a selected channel for a unit record area with stored identification which does not match the identifier and to trap the entire unit record area with an unequal identification. If the identification of all the unit record areas on the channel match the identifier, the last unit record area on the channel is trapped on the revolver.

The channel search unequal instruction governs switches (similar to the channel search equal instruction) which open and close under control of the unit record number portion of the address in the staticizer and the comparison of the identifier. Refer to the schematic drawing of the large capacity drum circuitry (Figure 11).

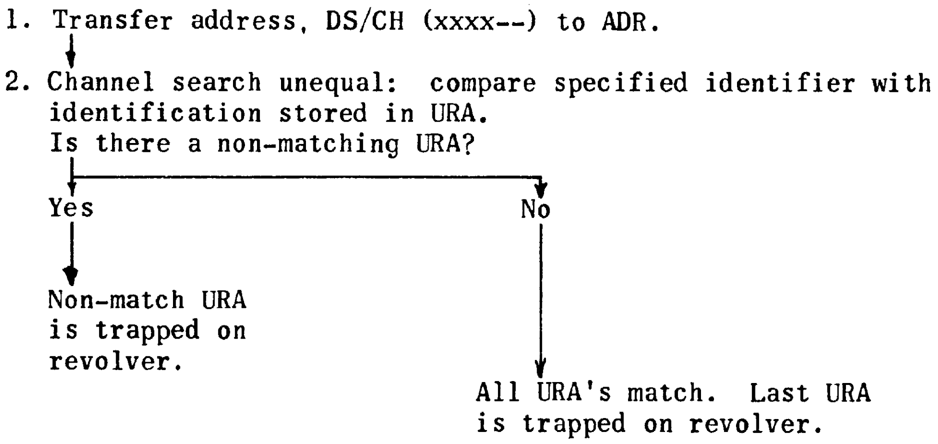
Before the channel search unequal instruction is given, the partial address of drum section and channel plus a starting unit record number is delivered to the address register. From there it is automatically transferred to the staticizer and the address register switch connects the read-write head of the specified channel to the associated read circuitry. Then, when the channel search unequal instruction is given, switch 2, controlled by the starting address, closes when this starting URA starts to pass under the read-write head. Then switch 1, controlled by the field identification on the revolver, closes when the particular field is passing under the read-write head. At the same moment, switch' 4 is opened. The identification of each URA is then compared with the identifier in the arithmetic unit and each successive unit record area on the channel being searched is written on the revolver. When the identification of a unit record area on the channel being searched does not match the desired identifier, switches 1 and 2 are opened at the end of that unit record area and switch 4 is closed. With the closing of switch 4 the data of the non-matching unit record area is "trapped" on the revolver.

When a non-matching unit record area is trapped on the revolver its location on the channel is "remembered" until a new address is entered in the address register.

If the identification of every unit record area on the channel searched matched the specified identifier switches 1 and 2 are opened at the drum index pulse--the end of the last unit record on the channel--and switch 4 is closed. Thus, the last unit record area on the revolver is trapped.

Two separate signals will emit; one if a non-matching unit record was found and trapped on the revolver or a second if all the unit record areas matched the specified identifier and the last unit record is trâpped on the revolver.

Following is a flow chart of a channel search unequal operation:



1. Transfer address, DS/CH (xxxx--) to ADR.

2. Channel search unequal: compare specified identifier with identification stored in URA.
   Is there a non-matching URA?

Yes

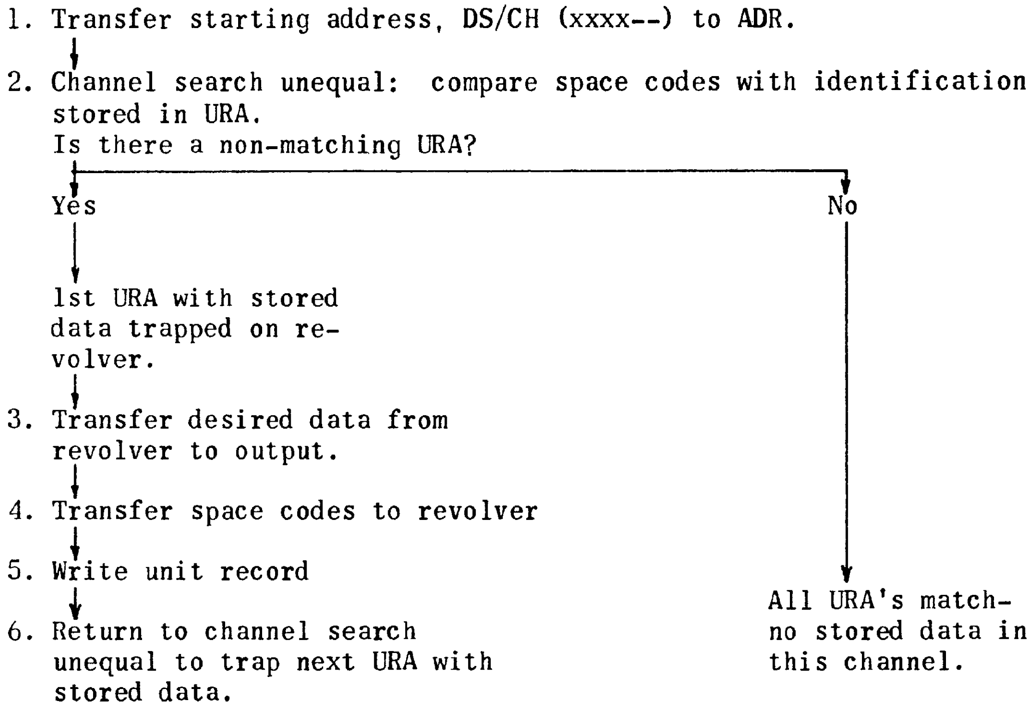
Non-match URA
is trapped on
revolver.

No

All URA's match. Last URA
is trapped on revolver.

The channel search unequal instruction is usually used in an unloading routine for the large capacity drums. Space codes or zeros are used as the identifier.

The starting drum section and channel number plus unit record number 00 is delivered to the address register and the channel search unequal instruction is given, comparing the space codes as the identifier with the identification field of the URA's. The first URA in the channel with stored data is trapped on the revolver. The desired information from that URA is transferred to output. Then space codes are transferred to the revolver field and a write unit record instruction is given--clearing that URA on the drum. The channel search unequal instruction is then given again, looking for the next URA with stored data, etc.



1. Transfer starting address, DS/CH (xxxx--) to ADR.

2. Channel search unequal: compare space codes with identification stored in URA.
   Is there a non-matching URA?

Yes

1st URA with stored data trapped on revolver.

3. Transfer desired data from revolver to output.

4. Transfer space codes to revolver

5. Write unit record

6. Return to channel search unequal to trap next URA with stored data.

No

All URA's match- no stored data in this channel.

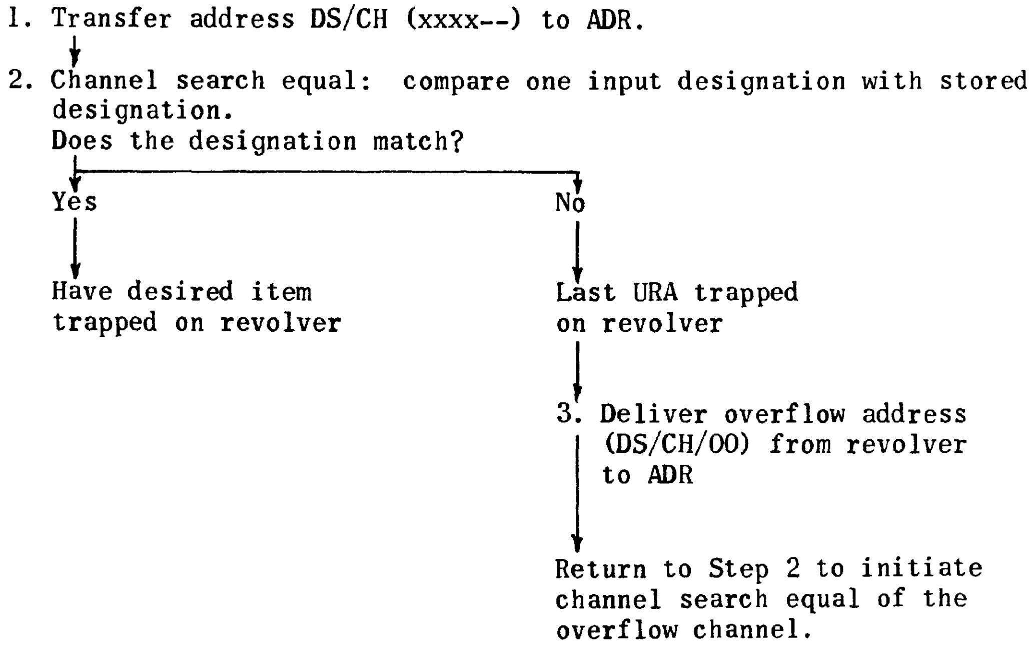
F. Channel Overflow

In some instances, the partial addresses developed from the designation of input data will indicate that the number of items to be stored in one channel exceed the number of unit record areas on that channel. When this occurs, a programming technique of channel overflow is used in conjunction with channel search. Then, if the desired item is not found on the channel searched, it is possible to direct the machine to the next channel that should be searched to locate the item.

When channel overflow occurs, one specific unit record area (the same one on each channel that overflows) is used to store the overflow address; that is, the address of the next channel to search, if the item is not located in this channel. Usually, the last unit record area in a channel is used to store the overflow address since, with either the channel search equal or the channel search unequal instruction, the last unit record area is trapped on the revolver if the desired URA was not located in the channel. Then, if after searching a channel the desired item is not found, the overflow address stored in the last URA (which is available on the revolver) is entered in the address register and the channel search is resumed in the overflow channel.

The following is a flow chart of a channel search equal operation when there is the possibility of channel overflow. (Actual programming is discussed in Section V.)

FLOW CHART OF CHANNEL SEARCH EQUAL WITH OVERFLOW POSSIBILITY



1. Transfer address DS/CH (xxxx--) to ADR.

2. Channel search equal: compare one input designation with stored designation.
   Does the designation match?

Yes

Have desired item trapped on revolver

No

Last URA trapped on revolver

3. Deliver overflow address (DS/CH/00) from revolver to ADR

Return to Step 2 to initiate channel search equal of the overflow channel.

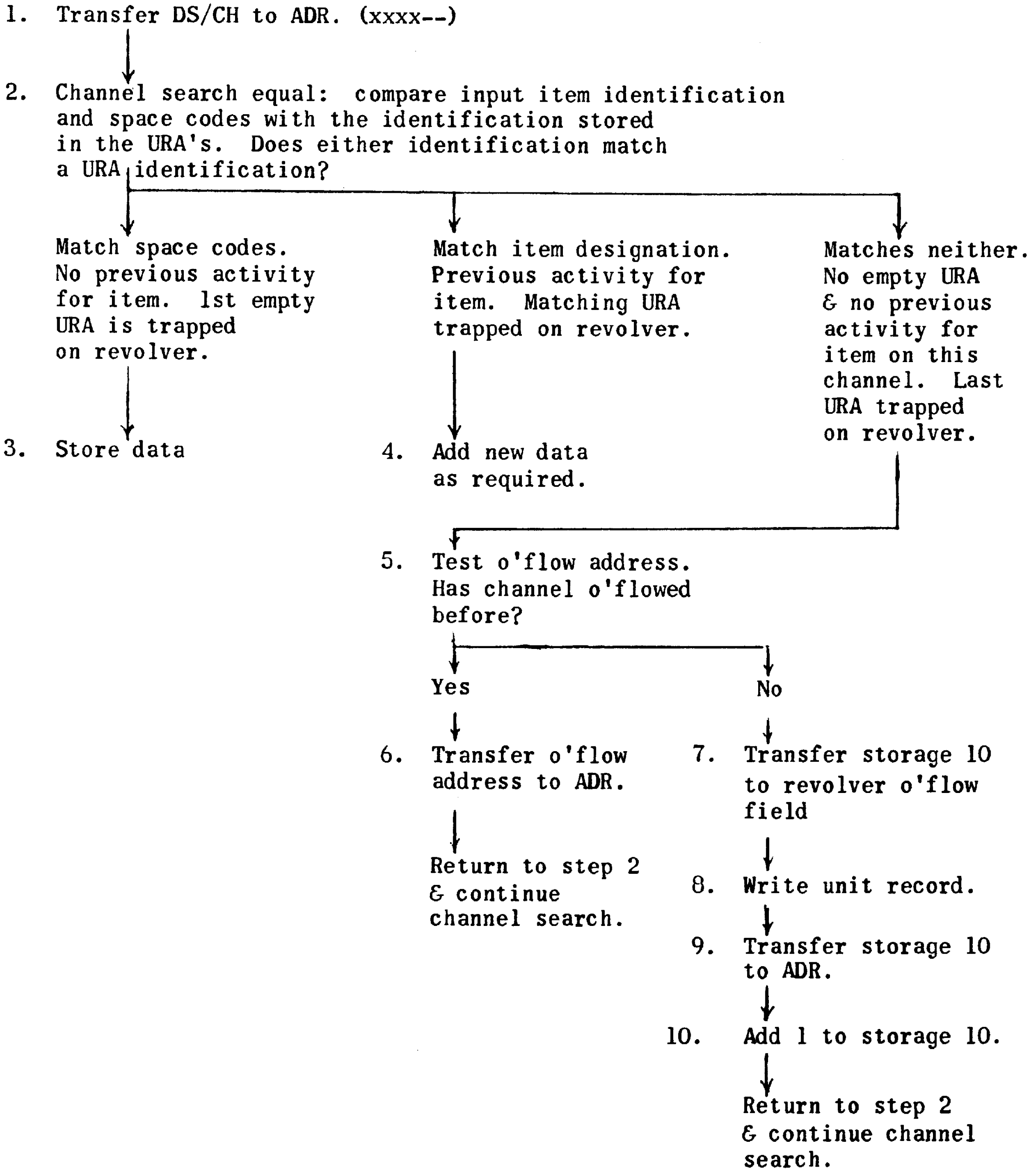
In this chart of a channel search operation, with the possibility of channel overflow, the channel overflow address is always stored in the last unit record area of any channel that may overflow. The stored overflow address consists of drum section (xx), channel (xx), and unit record area 00.

First the partial address of drum section,channel and space codes is transferred from input to the address register. The channel search equal therefore, starts with the first unit record on that channel. If the desired item is located during the search, the machine will emit a signal and the desired unit record area is trapped on the revolver. If the item is not located, the machine will emit another signal and the last unit record area...the one which contains the overflow address...is written on the revolver. The stored overflow address is then called for from the revolver and entered in the address register. Channel search is again initiated and the overflow channel is searched. This sequence can be continued until the desired item is located.

The following are flow charts of the programming technique for a channel search equal instruction with two identifiers as used for a loading routine and channel search unequal as used for an unloading routine. The possibility of overflow is considered.

In each case the last URA in a channel is used for the overflow address.

Loading routine: Channel search equal with two identifiers and automatic assigning of overflow channels:



1. Transfer DS/CH to ADR. (xxxx--)

2. Channel search equal: compare input item identification and space codes with the identification stored in the URA's. Does either identification match a URA identification?

Match space codes. No previous activity for item. 1st empty URA is trapped on revolver.

3. Store data

Match item designation. Previous activity for item. Matching URA trapped on revolver.

4. Add new data as required.

Matches neither. No empty URA & no previous activity for item on this channel. Last URA trapped on revolver.

5. Test o'flow address. Has channel o'flowed before?

Yes

6. Transfer o'flow address to ADR.

Return to step 2 & continue channel search.

No

7. Transfer storage 10 to revolver o'flow field

8. Write unit record.

9. Transfer storage 10 to ADR.

10. Add 1 to storage 10.

Return to step 2 & continue channel search.

NOTE: In the preceding example a portion of the channels are set aside for overflow. The address of the first overflow channel is stored in intermediate storage 10 before the run. During the run the overflow channels are automatically assigned as a channel is filled.

Unloading routine: Channel search unequal

1. Transfer starting DS/CH (xxxx--) to ADR.
2. Channel search unequal: Compare space codes with identification stored in URA.
   Is there a non-matching URA?

Yes
1st URA with stored data is trapped on revolver.

3. Transfer desired data from revolver to output.
4. Transfer space codes to revolver
5. Write Unit Record

Return to channel search to trap next URA with stored data.

No
All URA's match--no stored data in this channel
Last. URA in channel is trapped on revolver.

6. Add 1 to storage 10
7. Transfer storage 10 to ADR

Return to channel search to unload the next channel.

Note: In this example the channels are sequentially unloaded. The address of the starting channel is stored in intermediate storage 10 before the run.

G. The Large Capacity Drum Connection Panel

Once a unit record of the large capacity storage drums is written on the revolver, by a read unit record or channel search operation, the data within that unit record is available to the arithmetic and control section of the system through the 20 revolver fields, F0 through F19. Data from the system is placed on the revolver through the revolver fields.

The standard unit record area of the large capacity drums can be divided into these 20 possible revolver fields by wiring the drum connection panel. One connection panel governs all the large capacity drums in a system. That is, the field assignment, wired on the connection panel is effective for every unit record area called upon during a run. The field assignment can be varied during a run by the use of "selectors". The wiring of the connection panel can, of course, be changed between runs to produce a different field assignment arrangement for another run.

A tabulator wiring unit divides the standard unit record area of punched cards into fields for a tabulation run. The function of the drum connection panel is similar. It is the means of dividing the standard unit record character lenghth of the large capacity drums into the 20 possible revolver fields. The size of the revolver fields may vary from 1 through 12 characters including sign.

Remington Rand

DIVISION OF SPERRY RAND CORPORATION

TABULATING MACHINES

UNIVAC FILE - COMPUTER

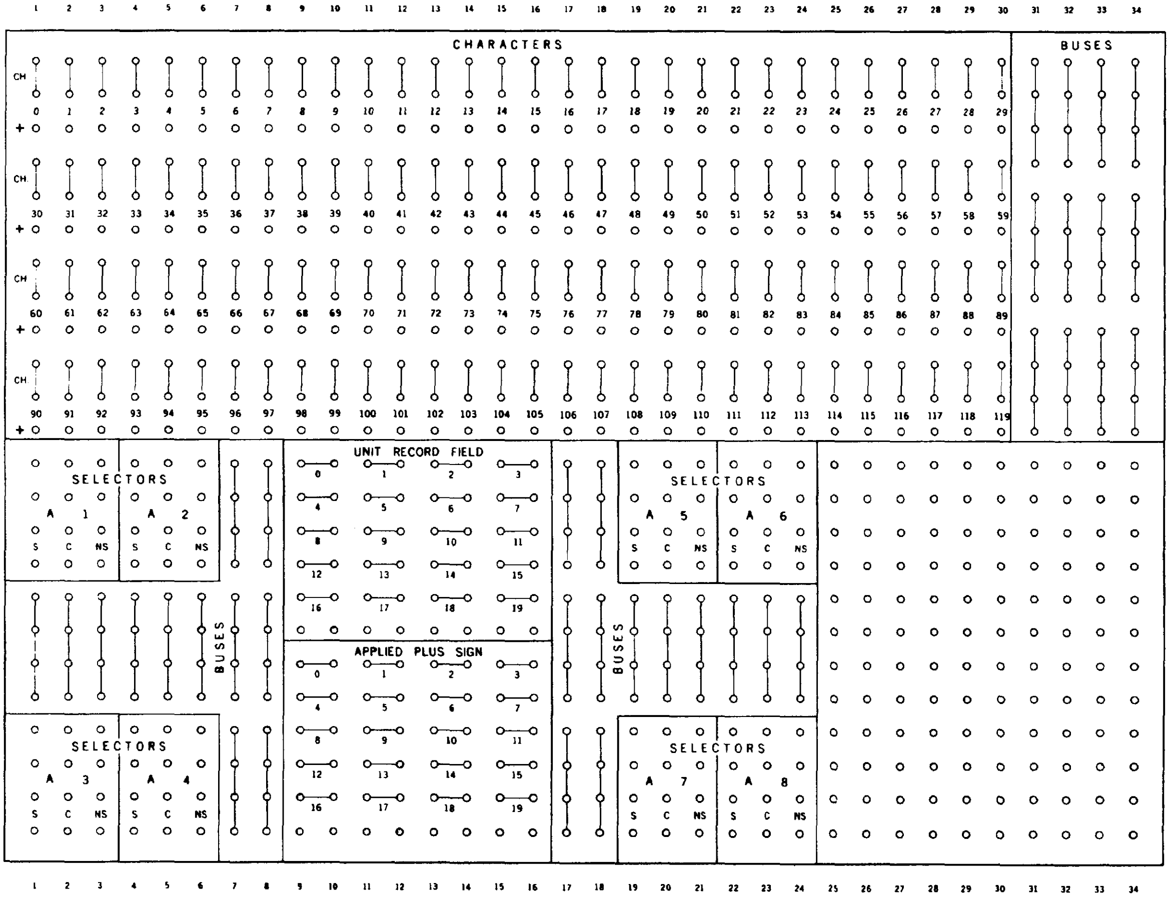
UNIT RECORD FIELD ASSIGNMENT PANEL

APPLICATION:— ______________________________

____

____

____



Characters: (A-L; 1-30) This section of the connection panel includes 120 double hubs, identified as CH and numbered 0 through 119. These represent the characters in the unit record area, up to the maximum of 120. (Note, double hubs... hubs that are joined together by a straight line on the connection panel are common; that is, they are internally joined together.) Data in the unit record area, which is to be treated as a field of information, must be in adjacent unit record area characters. Adjacent characters are grouped into fields by coupling (with external wires) the related character assignment hubs. The least significant character of a field is always represented by the left hand character within a coupled group of character assignment hubs, the most significant character by the right hand character.

Unit Record Field: (M-Q; 9-16) This section of the panel includes 20 double hubs, identified as 0 through 19. These represent the 20 possible revolver fields, F0 through F19. To give a group of coupled character hubs field identity, any character of the group is wired to a unit record field assignment hub.

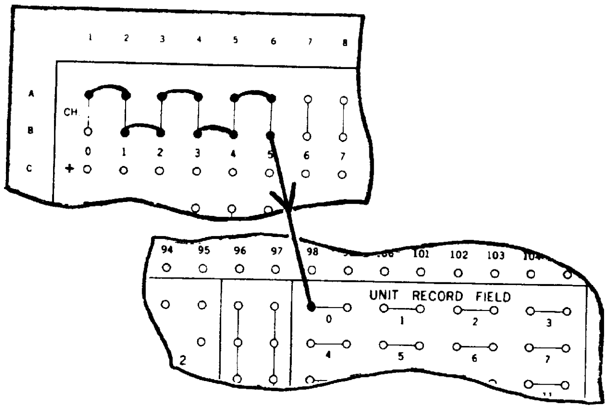
Positive Sign Hubs: (C; 1-30), (F; 1-30), (I; 1-30) and (L; 1-30) Immediately below each of the character hubs, in the character section of the control panel, is a hub identified with a +. These are the positive sign hubs. Normally, the least significant character location of a field stores the sign code. When the sign is not stored, a positive sign must be generated. This is done by wiring the positive sign hub directly under the least significant character of the field to a hub in the "applied plus sign" section.

Applied Plus Sign: (S-W; 9-16) This section of the connection panel includes 20 double hubs, identified as 0 through 19. These are related to the 20 possible revolver fields, F0 through F19. Whenever a sign is not stored for a field, a + sign is generated by the wiring from the positive sign hub of the least significant character of that field, to the applied plus sign hub which corresponds to the revolver field number to which the group is connected.

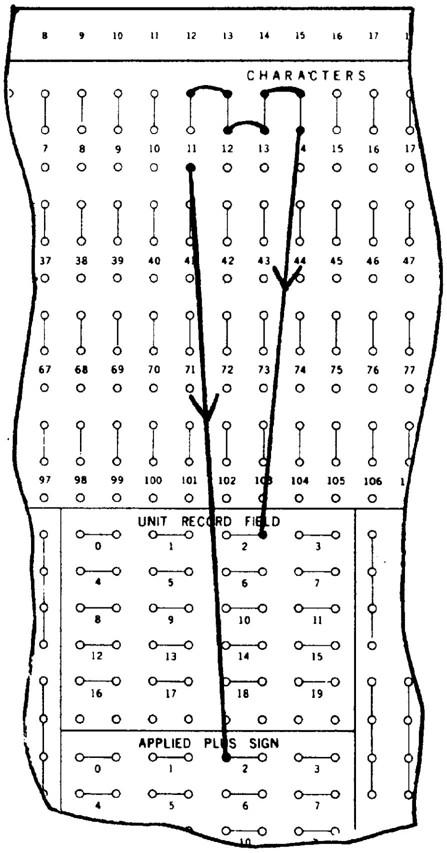
Selectors: (M-P; 1-6), (U-X; 1-6), (M-P; 19-24), and (U-X; 19-24). Eight 4 pole selectors, identified as A1 through A8, are included on the large capacity drum connection panel. The pickups of these selectors are on the computer connection panel. These selectors are used to vary the field assignment wiring. (Note: Selectors are explained in Section V.)

Examples:

1. Characters 0 through 5 are assigned to unit record field F0. The sign of the field is stored in character 0.



2. Characters 11 through 14 are assigned to unit record field F2. No sign is stored so the positive sign hub of character 11 (the least significant character) is wired to the applied plus sign of field F2 to generate a positive sign for the field.



3. Selectors on the drum connection panel are used to vary the character and field assignment when two or more types of records are stored on the large capacity drum(s) for one application and/or when two or more applications are being processed simultaneously by the system.

   In this example assume that two applications are being processed simultaneously. The field assignments desired are:

First Application	Second Application
Field F0 - Characters 0-3 Sign - Applied plus	Field F0 - Characters 0-6 Sign - Character 0
Field F6 - Characters 13-15 Sign - Applied plus	Field F6 - Characters 9-12 Sign - Character 9

   Selector A1 is non-select when a revolver field for the first application is called for; it is select when a revolver field for the second application is called for.

Therefore, characters 0 through 3 are connected to field F0, the plus sign of character 0 is connected to F0 applied plus sign, and character 13 through 15 are connected to field F6 with the plus sign of character 13 connected to F6 applied plus sign for the first application. For the second application, characters 0 through 6 are connected to field F0 and no applied plus sign for F0 is connected; and characters 9 through 12 are connected to field F6 with no connection between these characters and the applied plus sign of F6.

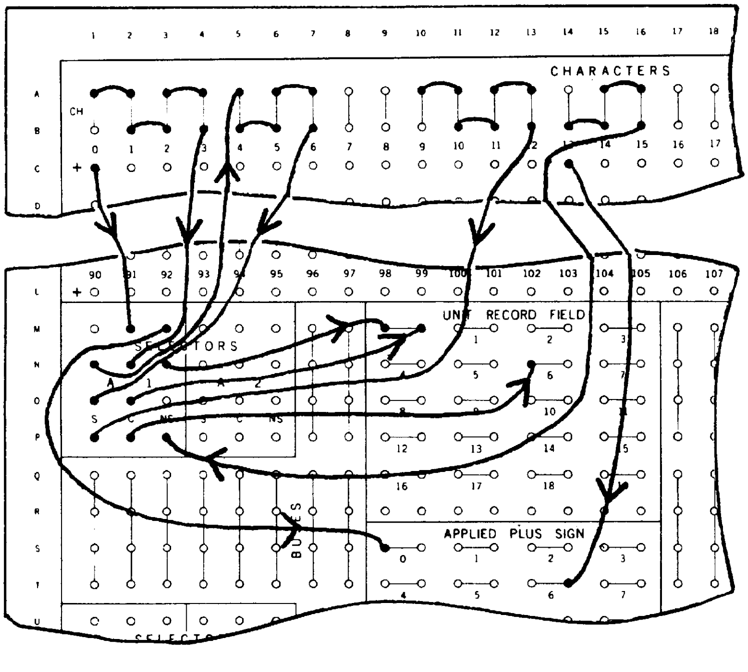
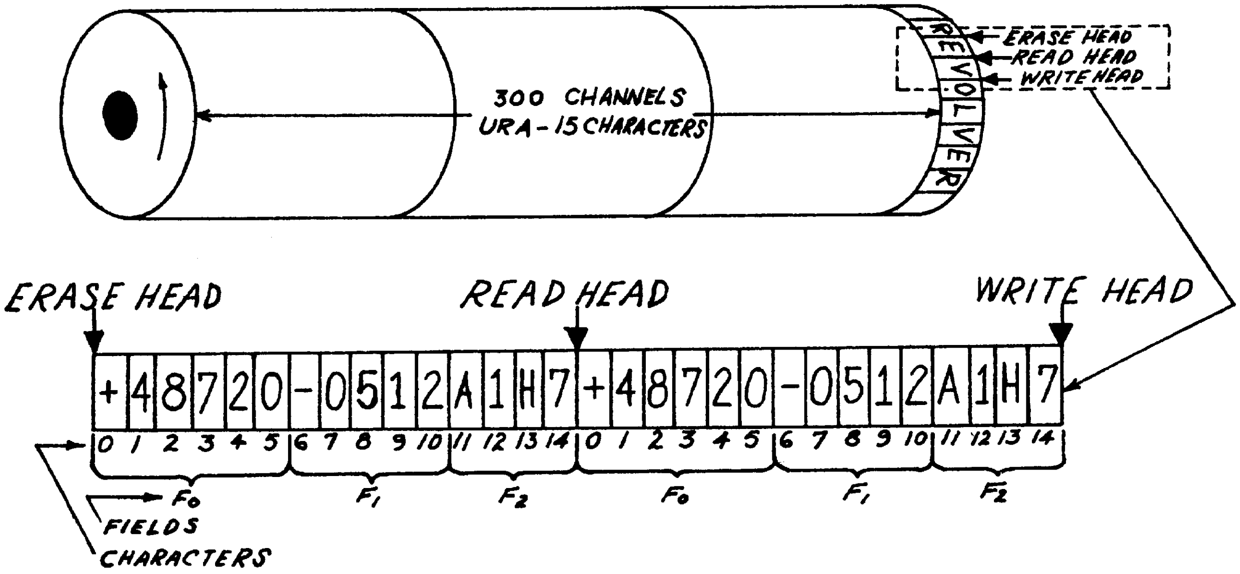
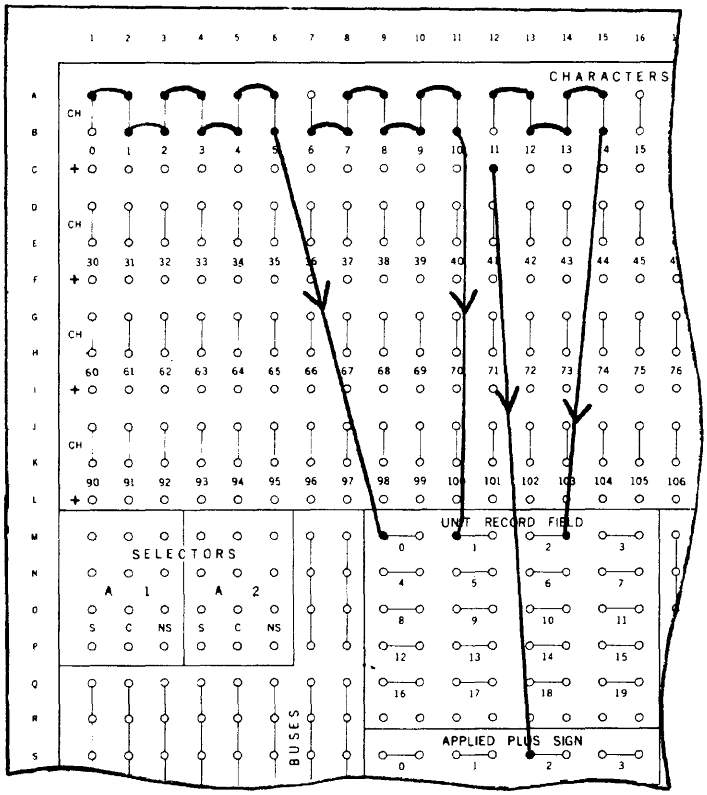


FIGURE 12
DIAGRAM OF REVOLVER UNIT
RECORD AREAS AND FIELDS



The preceding schematic drawing illustrates two revolver unit record areas, the data written on the revolver and the break-down of the revolver unit record area of 15 characters into revolver fields. Below is an illustration of the drum connection panel wiring for the field assignment of the characters.

Illustration of Connection Panel Wiring for Above Break-Down of Revolver URA Into Fields



Note: In accordance with the data written on the revolver in the above schematic, these values are obtained when the revolver fields are called:

	Value	Sign
F0 =	02784	+
F1 =	2150	-
F2 =	7H1A	+

Since field F2 contains alpha information it is obvious that it would not be used in an arithmetic process but in the functional process of transfer. Therefore, it would not be necessary to wire the applied + sign for F2 if the value F2 never required shifting, since data may be stored in the sign position of a storage unit. If the applied + sign were not wired, the value obtained when field F2 was called for would be:

	Value	Sign
F2 =	7H1	A

(See Section V for discussion of shifting and use of sign position to store data.)

Points to be observed in wiring large capacity drum connection panel:

1. One connection panel controls the field assignment of all large capacity storage drums in the system.

2. Only those characters in the character section on the panel which correspond to the fixed unit record area length of the system are wired. For example, if the unit record area length of the system is 50 characters, only character hubs 0 through 49 are used.

3. A desired field of information must be in adjacent characters of the unit record area.

4. The first character to the left in an adjacent group of characters is the least significant character; the last character to the right, the most significant. If no applied plus sign is generated, the least significant character in a field holds the sign; if an applied plus sign is generated, the least significant character is the least significant digit of the value.

5. Any character of the coupled character hubs may be wired to the unit record field assignment hub.

6. If the sign of a field is not stored, the positive sign hub of the least significant character must be wired to the related field applied plus sign to generate a plus sign unless the sign position is to be used to store alpha/numeric data which is to be used in a transfer operation.

7. If the number of characters in a field of data that is to be written on the revolver, through one of the revolver fields, exceeds the number of characters wired to that field, the most significant characters of that data are lost. For example, if a result of 11 characters, 1234567890 +, were delivered to a revolver field wired to 9 character hubs, the two most significant characters would be lost. 34567890 + would be written on the revolver.

8. The least significant character of a field and the applied plus sign establish the decimal relationship of the coupled characters.

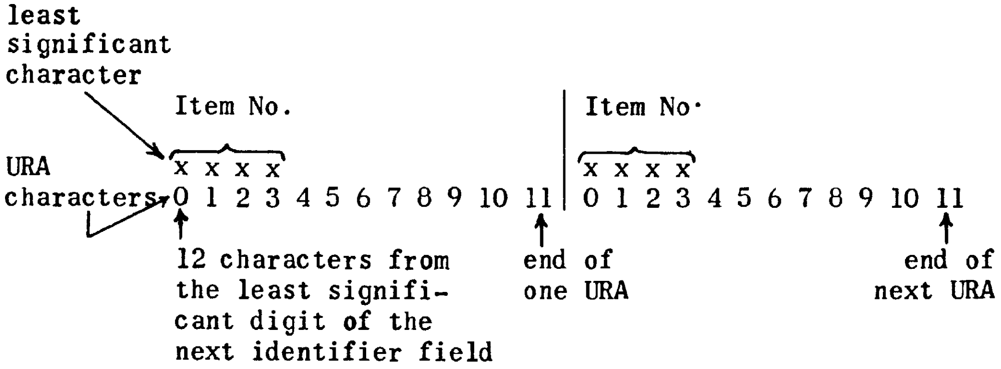
   a) If an applied plus sign is generated, the least significant character holds the units position of the value; the next character the tens position, etc.

b) If an applied plus sign is not generated and if the sign of the value is stored, the least significant character holds the sign; the next character, the units position of the value; the next, the tens position, etc.

c) If an applied plus sign is not generated and if the sign of the value is not stored, the least significant character holds the units position of the value (this character is in the sign position); the next character the tens position, etc.

9. In a revolver field that is to be compared during a channel search operation, the least significant character of that field must be located at least 12 characters from the next identifier field. For example:

Assume a unit record area of 12 characters. The least significant character of the field to be compared during the channel search operation must be stored in character 0:



10. If two revolver fields are to be compared during a channel search operation, the least significant characters of the fields must be at least 12 characters apart--that is, separated by 11 characters. (Note, if more than two fields are to be compared see following section H.) For example:

Assume a unit record area of 30 characters. Two fields of 11 characters plus sign each are to be compared during the channel search operation. They could be stored in the unit record area in this fashion:

```
        1st Item No.                         2nd Item No.
x x x x x x x x x x x x     x  x  x  x  x  x  x  x  x  x  x  ..
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23..29

the least significant       least significant characters are 0 and 12
characters are separated
by 11 characters
```

H. Unit Record Length

The unit record area length is the same for all large capacity drums included in a Univac File-Computer system. In analyzing the applications to be handled by the system, it may be found that the unit record area requirements for the large capacity drums vary from one application to another. When this occurs, the unit record length of the application which requires the most frequent reference to the large capacity drums, is generally chosen for the system. Programming techniques can then be used to divide the unit record length into controllable areas for the applications which require a shorter unit record length, and/or to couple unit record areas for the applications which require a longer unit record length.

In addition, when two or more applications are being processed simultaneously by the system, the field assignment of the unit record area characters may vary according to the application. When this occurs, the characters to be assigned to a specific field can be selected through the large capacity drum connection panel selectors.

a. Selection of Fields: The selectors on the drum connection panel are the means of altering the revolver character and field assignment. This selection was illustrated in Example 3 in the preceding section.

b. Multiple Items Stored in One Unit Record Area: When the unit record area requirements of an application are smaller than the fixed unit record length, two of the items can be stored in one unit record by dividing the unit record length into controllable areas by programming. The unit record in which the item is stored can be found by a channel search operation (if the specific address is not available from input) and then, the specific item within that unit record area by programmed comparing of the designation.

For example, assume that the fixed unit record length of a system is 40 characters, and that a payroll application to be processed by the system required only 20 characters of storage:

Employee number........	4 characters
Earnings-to-date.......	6 characters
FICA-to-date...........	4 characters
FWT-to-date............	6 characters

Two of these payroll unit records can be stored in the fixed unit record area of 40 characters:

	Description	URA Characters			Field Assignment
1.	Emp. number	0	through	3	F0
	Earn-to-date	4	"	9	F1
	FICA-to-date	10	"	13	F2
	FWT-to-date	14	"	19	F3
2.	Emp. number	20	"	23	F4
	Earn-to-date	24	"	29	F5
	FICA-to-date	30	"	33	F6
	FWT-to-date	34	"	39	F7

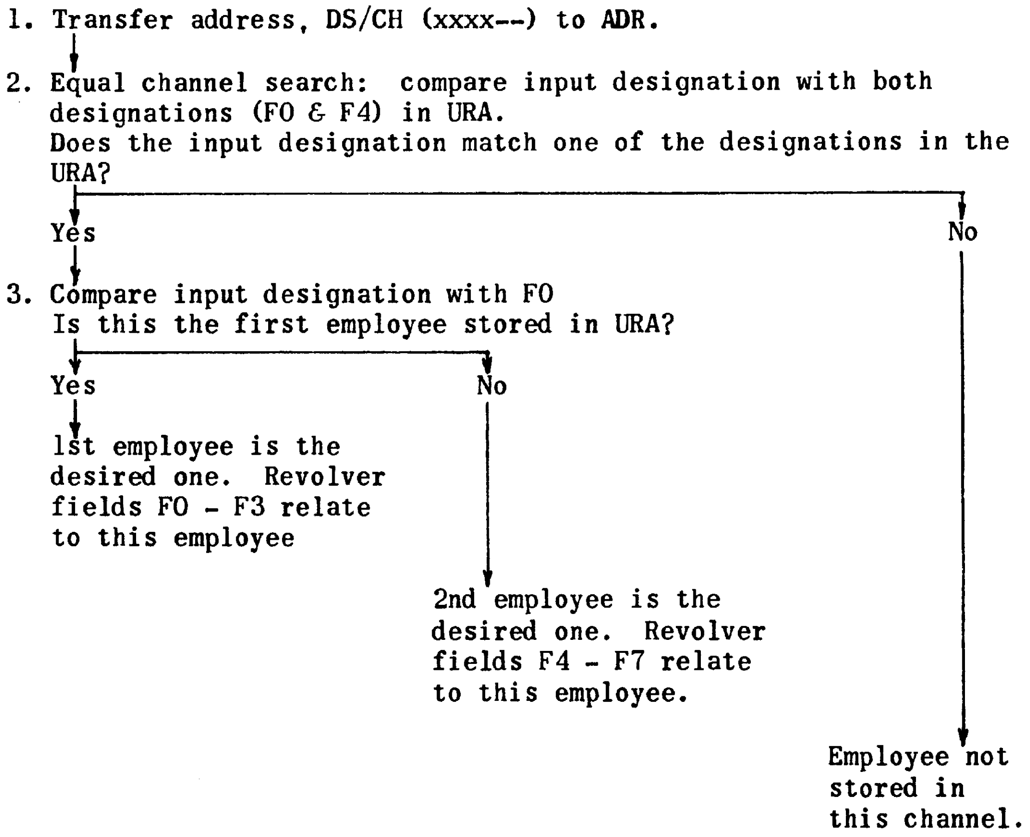
In order to locate the stored data for a specific employee the employee number from the input data is compared, during an equal channel search operation, with the 2 stored employee numbers in each unit record area. When one of the stored employee numbers matches the desired employee number, that complete unit record area is trapped on the revolver.

Note that there are 19 characters between the least significant characters of the two stored employee numbers (characters 1 through 19), and that the least significant character of the last employee number in the unit record area is 20 characters (characters 20 through 39), from the end of the unit record area. Therefore, for the channel search operation, the least significant characters of the employee numbers are separated by at least 11 characters. The least significant character of the last stored employee number is at least 12 characters from the end of the unit record area.

When the unit record area is "trapped" on the revolver, programmed comparing is used to determine which of the 2 groups of stored employee information is the desired one. The first stored employee number (field F0) is compared with the desired employee number. If they are equal, revolver fields F0 through F3 relate to the desired employee; if they are not equal, fields F4 through F7 relate to the desired employee.

Following is a flow chart of a channel search operation for this example...2 items stored in one unit record area. (Actual programming is discussed in Section V).

FLOW CHART OF CHANNEL SEARCH WHEN DATA FOR 2 ITEMS ARE STORED IN ONE UNIT RECORD AREA: NO OVERFLOW



1. Transfer address, DS/CH (xxxx--) to ADR.

2. Equal channel search: compare input designation with both designations (FO & F4) in URA.
Does the input designation match one of the designations in the URA?

Yes

3. Compare input designation with FO
Is this the first employee stored in URA?

Yes

1st employee is the desired one. Revolver fields FO - F3 relate to this employee

No

2nd employee is the desired one. Revolver fields F4 - F7 relate to this employee.

No

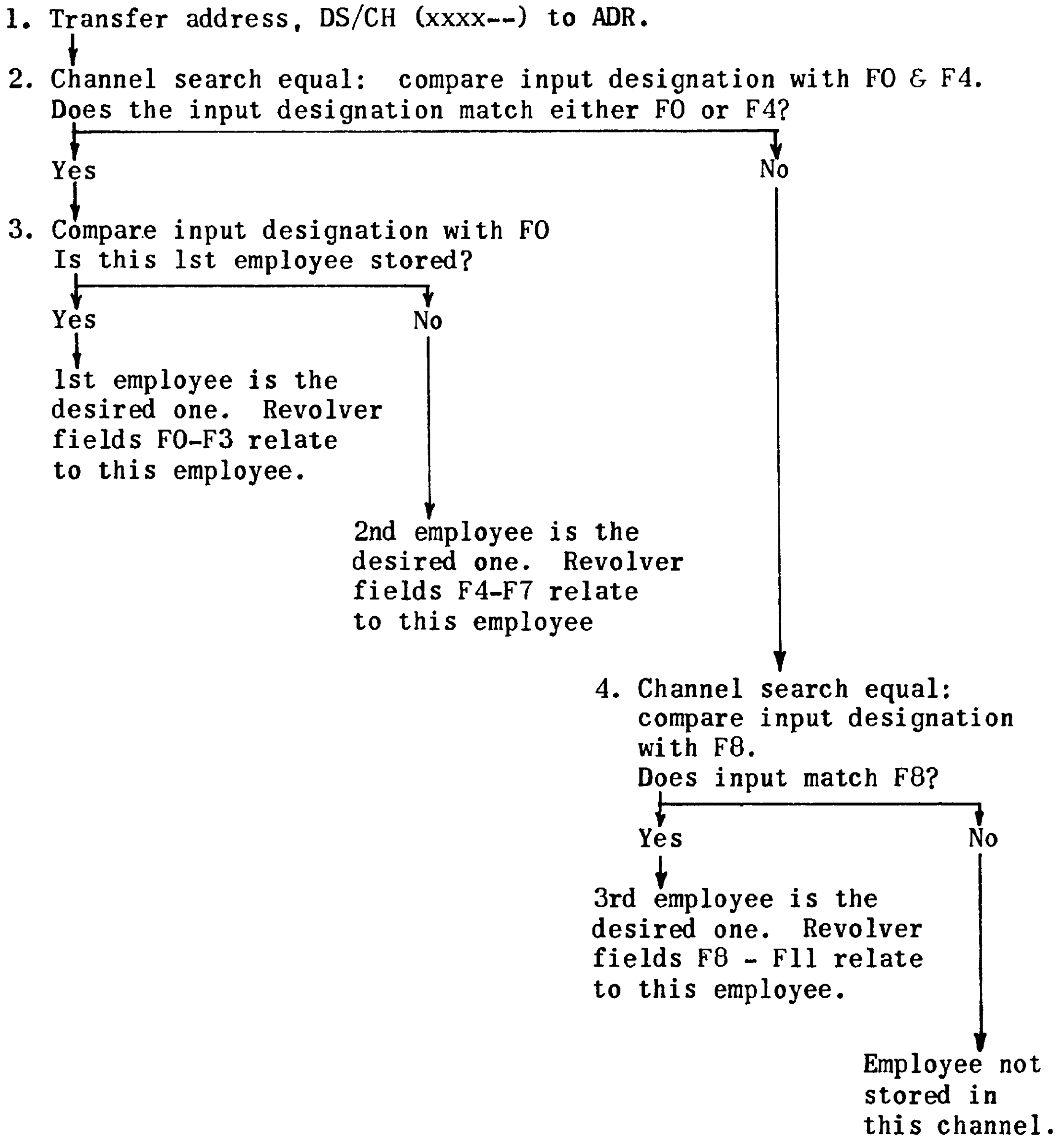
Employee not stored in this channel.

It is possible to store the information for more than 2 items in one URA and to locate the desired item by a channel search equal instruction and programmed comparison. Following are three methods of accomplishing this:

Method 1. Perform 2 or, if necessary, more channel search operations of the channel. For example, the preceding illustration showed the data for 2 items, requiring 20 characters each, stored in one 40 character URA. Assume that the fixed URA length is 60 characters. It is possible to store the data for 3 items in one URA and locate the desired item by channel searching the channel twice.

	Description	URA Characters	Field Assignment	
1.	Emp. number	0 through 3	F0	1
	Earn.-to-date	4 " 9	F1	
	FICA-to-date	10 " 13	F2	
	FWT-to-date	14 " 19	F3	
2.	Emp. number	20 " 23	F4	2
	Earn.-to-date	24 " 29	F5	
	FICA-to-date	20 " 33	F6	
	FWT-to-date	34 " 39	F7	
3.	Emp. number	40 " 43	F8	3
	Earn.-to-date	44 " 49	F9	
	FICA-to-date	50 " 53	F10	
	FWT-to-date	54 " 59	F11	

In order to locate the stored data for the specific employee, the employee number from the input data is first compared, during a channel search equal operation, with F0 & F4. If a match is found then the specific employee can then be determined by comparison (see preceding example). If a match is not found, the channel search equal operation is again called for, this time comparing the input designation with F8.



1. Transfer address, DS/CH (xxxx--) to ADR.

2. Channel search equal: compare input designation with F0 & F4.
Does the input designation match either F0 or F4?

Yes

3. Compare input designation with F0
Is this 1st employee stored?

Yes

1st employee is the desired one. Revolver fields F0-F3 relate to this employee.

No

2nd employee is the desired one. Revolver fields F4-F7 relate to this employee

No

4. Channel search equal: compare input designation with F8.
Does input match F8?

Yes

3rd employee is the desired one. Revolver fields F8 - F11 relate to this employee.

No

Employee not stored in this channel.

The next two methods can be used if the URA length of the system is divisible by 12, ie, 12, 24,60 & 120. Both of these methods are most practical with the longer URA lengths -- 60 and 120.

Both methods make use of a special channel search address, "all". This "all" address tells the machine to ignore the revolver field assignment, wired on the drum plugboard, and to consider each successive group of 12 characters as a field. That is, with a URA length of 120 characters, when a channel search instruction is given, comparing the desired designation with "all" URA fields, the machine ignores the wired field assignment and compares the specified designation first with characters 0 through 11, then with 12 through 23, then 24 through 35, etc. through 108 through 119. These comparisons are made for each URA on the channel. If any one of the 10 groups of 12 characters in a URA match the desired designation, the machine will emit a signal and that URA will be trapped on the revolver.

Since each successive group of 12 characters are treated as a field when using the channel search "all" address, 2 precautions should be observed:

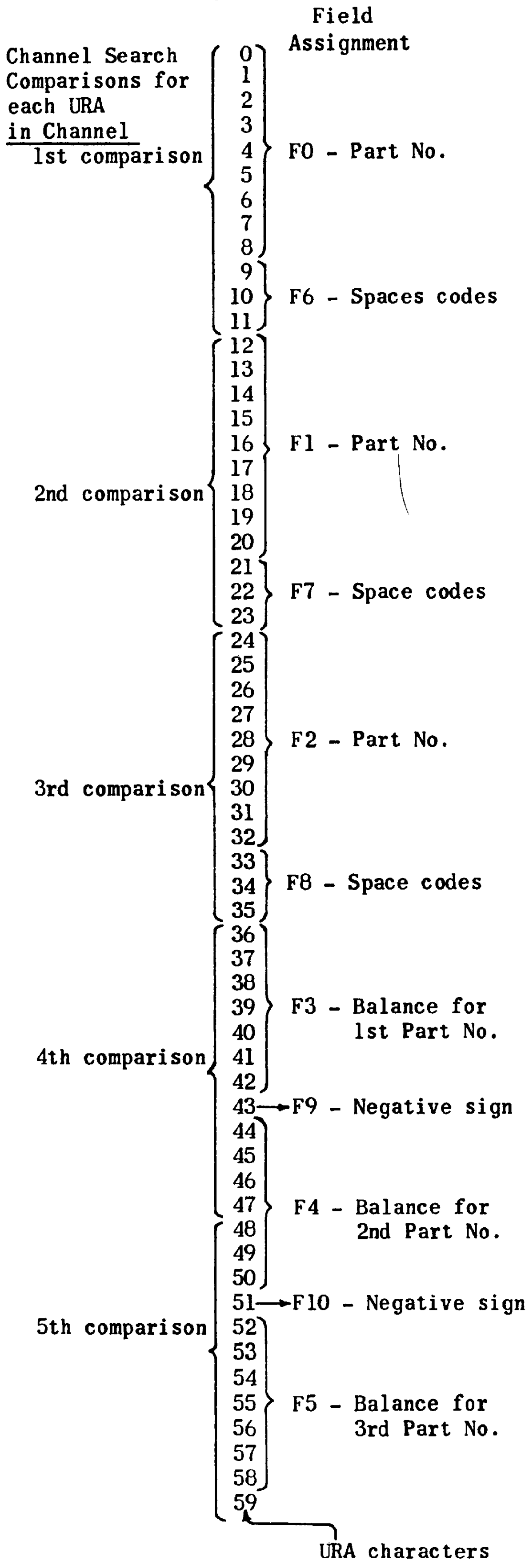
1. 12 characters must be allowed for each identification stored;

2. Other values, such as balances, amounts, etc. in a group of 12 characters could match the specified designation.

The following two methods illustrate the means of locating a specific item, when more than 2 items are stored in one URA, with the channel search "all" address.

Method 2. Store an "impossible code" within each group of 12 characters which do not contain an identification. This "dummy" or "impossible" code should be one that cannot possibly be part of the designation of the desired item. A negative code for example, would never be part of an employee number or part number. If a match is then found during channel search equal, it is certain to be a valid identification.

Method 2 - URA Layout.

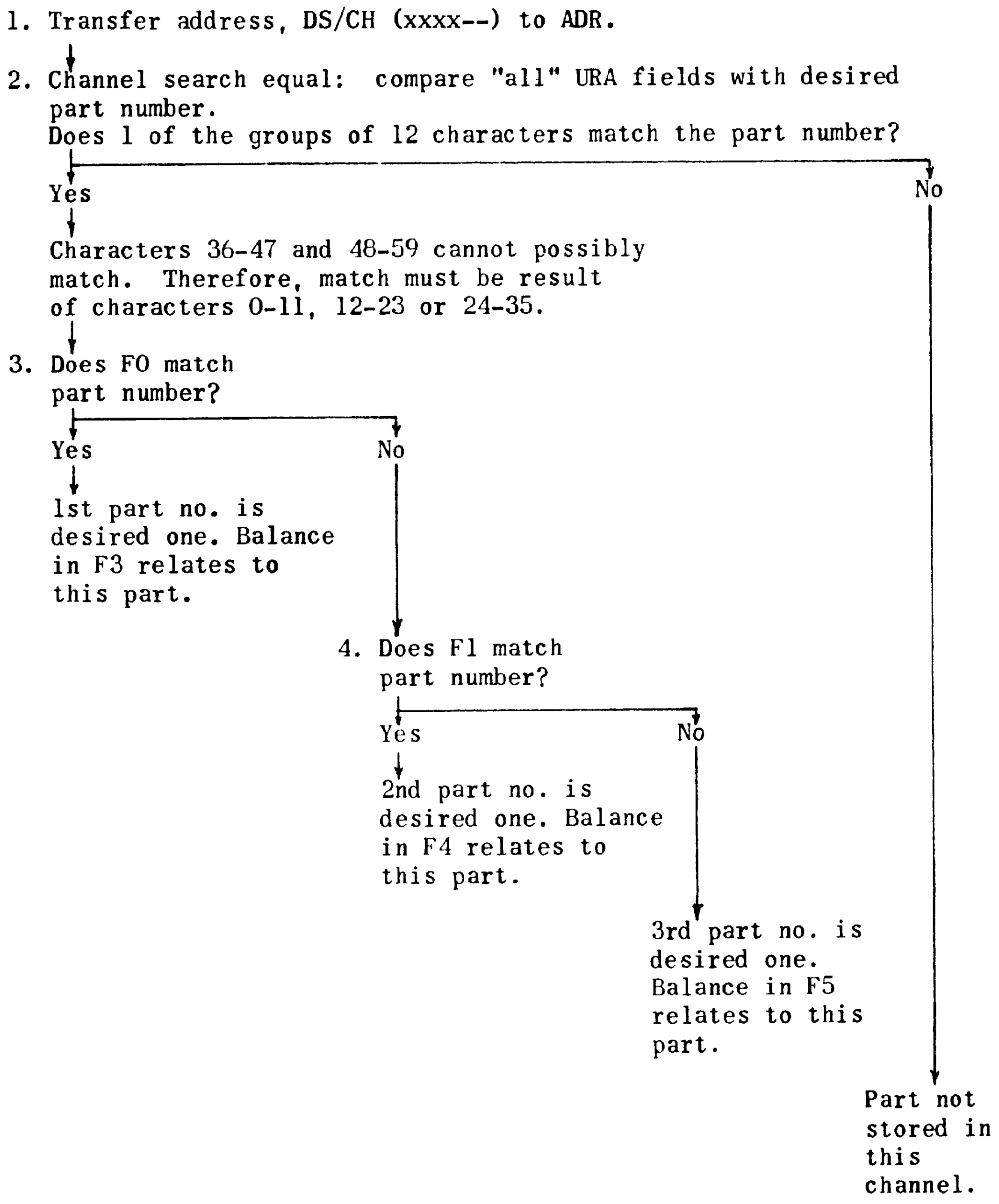


In this example, with the channel search equal operation and the channel search "all" address, the wired field assignment of the characters is ignored and characters 0-11, 12-23, 24-35, 36-47, and 48-59 of each URA will be compared with the desired part number.

Since a negative code will never be in a part number, the negative signs stored in characters 43 and 51 will prevent any possibility of an equal comparison between the desired part number and the values stored in characters 36 - 47 or 48 - 59.

Therefore, if an equal comparison is obtained it must have been from characters 0-11, 12-23 or 24-35.---the 3 stored part numbers.

Flow chart of Method 2:



1. Transfer address, DS/CH (xxxx--) to ADR.

2. Channel search equal: compare "all" URA fields with desired part number.
   Does 1 of the groups of 12 characters match the part number?

Yes

Characters 36-47 and 48-59 cannot possibly match. Therefore, match must be result of characters 0-11, 12-23 or 24-35.

3. Does F0 match part number?

Yes

1st part no. is desired one. Balance in F3 relates to this part.

No

4. Does F1 match part number?

Yes

2nd part no. is desired one. Balance in F4 relates to this part.

No

3rd part no. is desired one. Balance in F5 relates to this part.

No

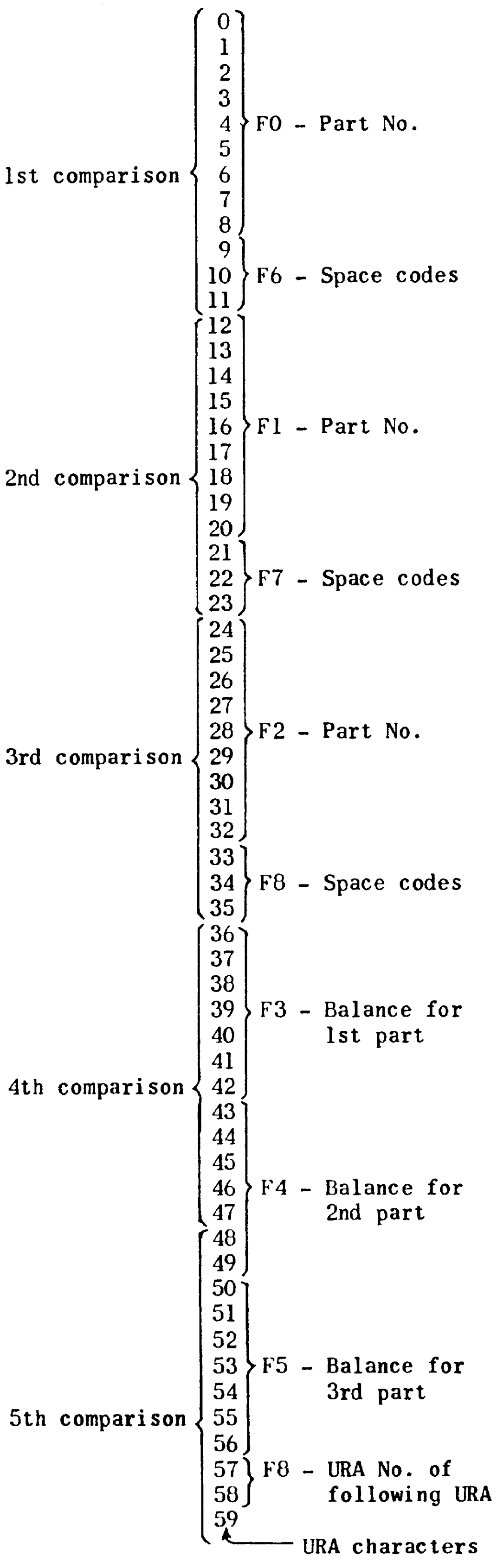
Part not stored in this channel.

Method 3:

Following is the third method of storing data for more than two items in one URA and locating the desired item by channel search.

Method 3: URA Layout

In each URA, store the URA number of the following URA. For example:

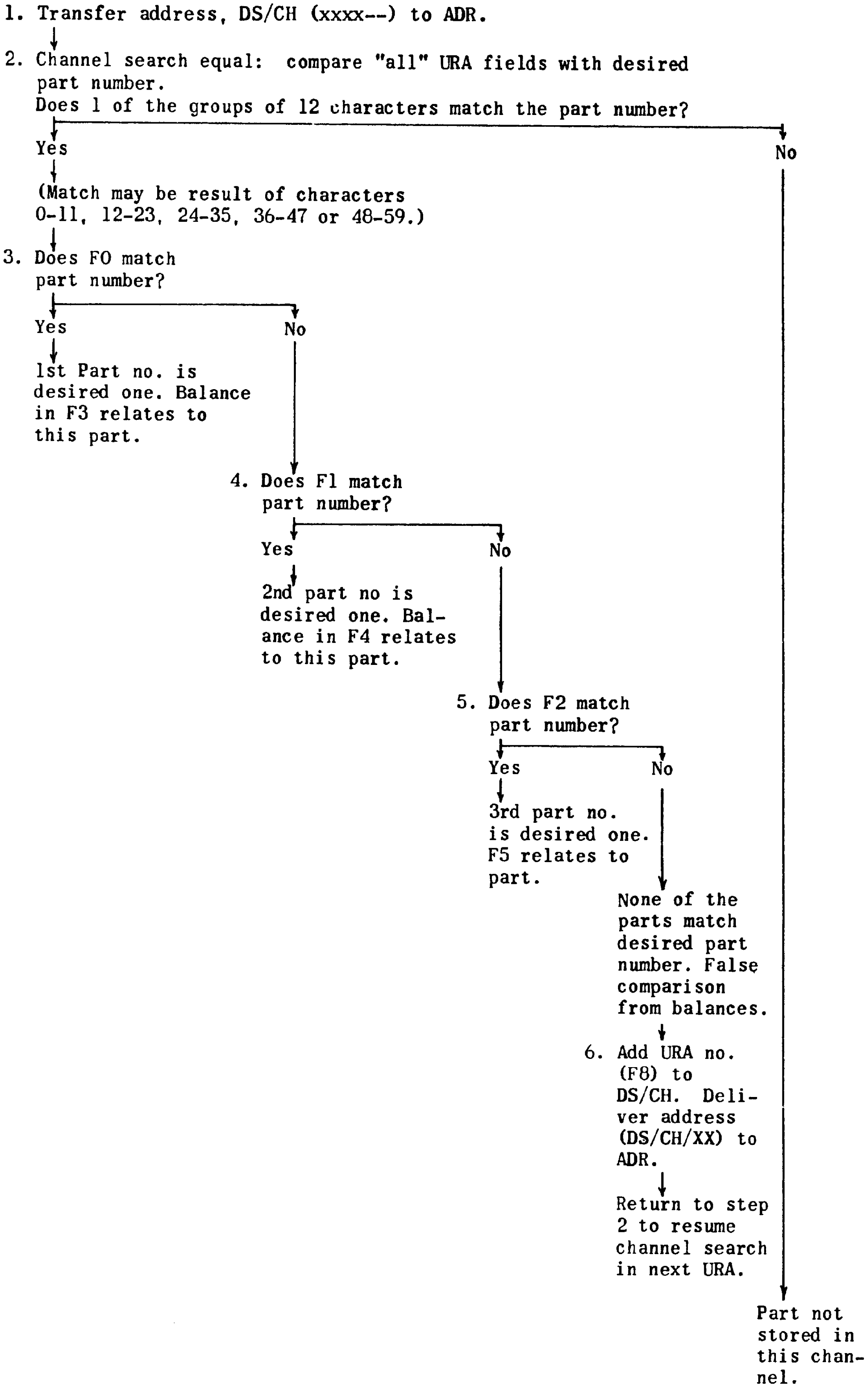


In this example, with the channel search equal instruction and the channel search "all" address, characters 0-11, 12-23, 24-35, 36-47 and 48-59 will be compared with the desired part number.

An equal comparison could result from any one of the 5 groups of characters; from the first 3 groups if the part number matched or from the last 2 groups if the balances stored happened to match the desired part number.

Therefore, if a match is obtained, the 3 stored part numbers are tested against the specified part number. If one matches, the desired URA is available on the revolver. If none of them match, the stored URA number (of the following URA) is added to the partial address of DS and CH and the channel search is resumed at the next URA in that channel.

Flow chart of Method 3:



1. Transfer address, DS/CH (xxxx--) to ADR.

2. Channel search equal: compare "all" URA fields with desired part number.
Does 1 of the groups of 12 characters match the part number?

Yes

(Match may be result of characters 0-11, 12-23, 24-35, 36-47 or 48-59.)

3. Does F0 match part number?

Yes

1st Part no. is desired one. Balance in F3 relates to this part.

No

4. Does F1 match part number?

Yes

2nd part no is desired one. Balance in F4 relates to this part.

No

5. Does F2 match part number?

Yes

3rd part no. is desired one. F5 relates to part.

No

None of the parts match desired part number. False comparison from balances.

6. Add URA no. (F8) to DS/CH. Deliver address (DS/CH/XX) to ADR.

Return to step 2 to resume channel search in next URA.

No

Part not stored in this channel.

c. Multiple Unit Record Areas for the Storage of One Item: When the unit record area requirements for an application are greater than the fixed unit record area length, two (or more) unit records can be used for the storage of data related to one item. The item designation and part of the data related to that item are stored in one unit record area. In addition, the address of the unit record area which contains the balance of the data related to that item is also stored.

The unit record area which contains the first portion of data can be located by an equal channel search operation and the references to and/or calculations based on that portion of the data can be performed. Then, the stored address of the unit record area which contains the balance of the data related to that item can be entered in the address register and the second portion of the data made available on the revolver by a read unit record instruction.

If possible, the unit record area chosen to hold the balance of the data for an item, is located in the same channel as the unit record area which contains the first portion of the data. Then, it is only necessary to store 2 characters (the unit record number) for the address of the second unit record area. If this is not possible, the full 6 digit address of the unit record area which contains the balance of the data must be stored.

The item identification must be stored in the unit record area which contains the first portion of data related to that item, so that the unit record area can be located by a channel search equal operation (if the specific address is not available from input). It does not have to be stored in the unit record area which contains the balance of data related to that item since the stored address can be used to locate the specific unit record area.

Since different data is stored in the various unit record areas, the revolver field alignment must either be arranged to be consistent with both types of unit record areas (the URA which contains the 1st portion of data, and the URA which contains the balance of data related to one item) or the fields must be selected according to the portion of the data being referred to.

For example, assume that the fixed unit record length of a system is 20 characters and that a payroll application to be processed by the system requires 34 characters of storage:

Employee number	5 characters
Day Work Rate............	5 characters
Average Incentive rate...	5 characters
Earnings-to-date.........	6 characters
FWT-to-date..............	6 characters
FICA-to-date.............	4 characters
State Tax-to-date........	3 characters
	34 characters

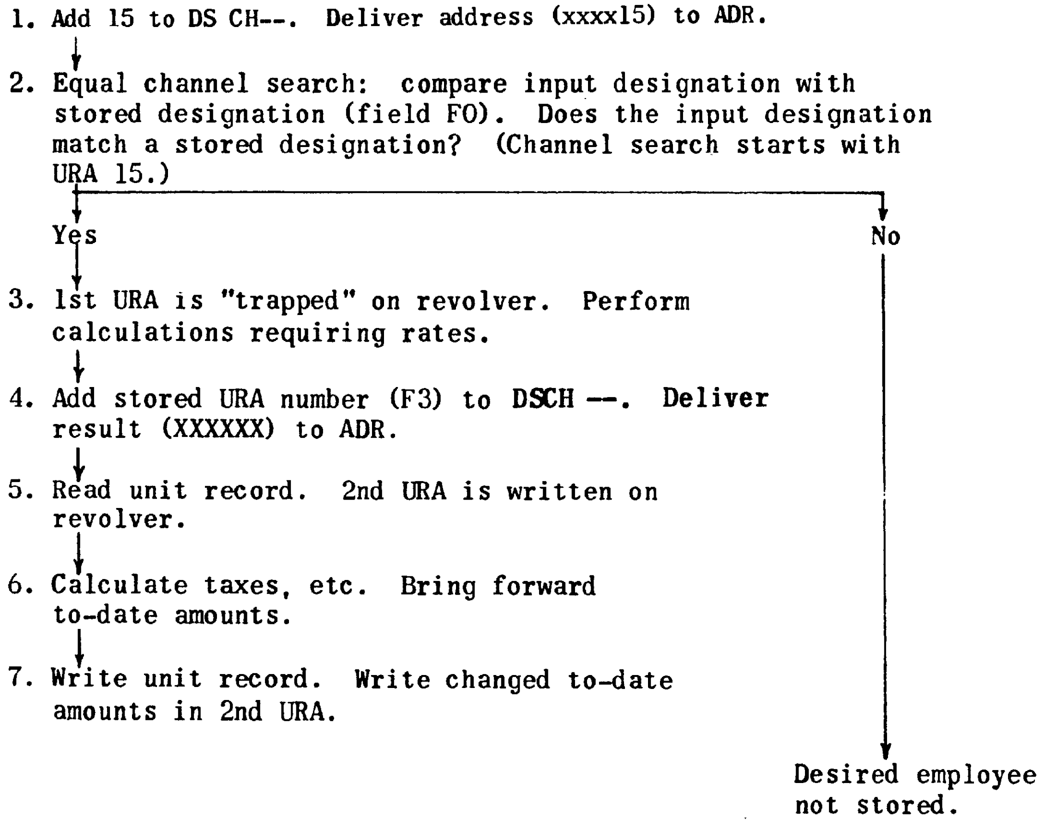
Two unit record areas are required to store the data for one item. In this example, assume that it is possible to locate both unit record areas required for an item in the same channel. Therefore, only 2 characters are required to store the address of the next unit record area. The data could be stored:

UNIT RECORD AREA WHICH CONTAINS 1ST PORTION OF DATA			UNIT RECORD AREA WHICH CONTAINS BALANCE OF DATA		
Description	Field	URA Character	URA Character	Field	Description
Employee No.	F0	X0	0X	F0	Earn-to-date
		X1	1X		
		X2	2X		
		X3	3X		
		X4	4X		
		5	5X		
Day Rate	F1	X6	6X	F1	FWT-to-date
		X7	7X		
		X8	8X		
		X9	9X		
		X10	10X		
		11	11X		
Average Incentive Rate	F2	X12	12X	F2	FICA-to-date
		X13	13X		
		X14	14X		
		X15	15X		
		X16	16		
Address	F3	X17	17X	F3	State-to-date
		X18	18X		
		19	19X		

The revolver field alignment is consistent for both types of unit record areas. The last 15 unit record areas on each channel (15 through 29) are used to store the lst portion of data; the first unit record areas (00 through 14) are used to store the 2nd portion of data for the related employees. The channel search is started with URA 15, comparing field FO--thus, only the stored employee number is compared with the desired one; earnings-to-date, in the same relative characters in the lst 15 unit record areas is not allowed to compare with the desired employee number.

Following is a flow chart of a channel search operation for this example...1 item stored in 2 unit record areas. (Actual programming is discussed in Section V.)

FLOW CHART OF CHANNEL SEARCH WHEN DATA FOR 1 ITEM IS STORED IN 2 URA'S: NO OVERFLOW



1. Add 15 to DS CH--. Deliver address (xxxxl5) to ADR.

2. Equal channel search: compare input designation with stored designation (field FO). Does the input designation match a stored designation? (Channel search starts with URA 15.)

Yes

3. lst URA is "trapped" on revolver. Perform calculations requiring rates.

4. Add stored URA number (F3) to DSCH —. Deliver result (XXXXXX) to ADR.

5. Read unit record. 2nd URA is written on revolver.

6. Calculate taxes, etc. Bring forward to-date amounts.

7. Write unit record. Write changed to-date amounts in 2nd URA.

No

Desired employee not stored.

SECTION V

ARITHMETIC AND CONTROL SECTION PROGRAMMING

ARITHMETIC AND CONTROL PROGRAMMING

INDEX - SECTION V

1. INTRODUCTION

A program is a series of instructions which, when followed by the machine will produce the desired computational results. These instructions are given to the machine in a form known as a program step. A program step will perform one complete arithmetic or operational function.

Each program step consists of five basic elements:

1. The arithmetic or operational process to be performed: addition, subtraction, multiplication, division, transfer, compare, left zero eliminaton, or channel search.

2. The storage location of the first value (V_1) upon which the process is to operate.

3. The storage location of the second value (V_2) upon which the process is to operate.

4. The storage location to which the result of the process (R) is to be delivered.

5. The next step. This function permits the joining together of steps to form a program.

As can be seen, a program is merely a combination or series of individual steps which direct the machine to perform the operations required to obtain a desired result or results.

There are two means of setting up a program for the Univac File-Computer; and they may be used in combination with each other:

1. External Program: This is the type of programming used with the Univac 60-120. All instructions are wired on a removable connection panel.

2. Internal Program: With this type of program, the instructions are stored on the high-speed storage drum. The connection panel is used with internal programming to determine the process and to initiate an internal instruction; that is, to tell the control section to look on the drum, rather than the connection panel, for its next instruction.

   Combination of internal and external program: With this type of program, part of the instructions are wired on the connection panel and part are stored on the high-speed storage drum.

The basic form of a program step consists of the program elements arranged in the following form:

Value 1	Process	Value 2	=	Result	Next
(V_1)	(Pr)	(V_2)		(R)	Step

There are nine processes available on the Univac File-Computer, classified as follows:

Arithmetic:

Addition
Subtraction
Multiplication
Division

Operational:

Transfer
Compare
Equal Channel Search
Unequal Channel Search
Left Zero Elimination

The following sections deal with each element of a program step in relation to its use with external programming only. At the end of Section V the Control and Program Panel is shown with a numbered guide indicating where the particular hubs are described in the section.

2. VALUE 1, VALUE 2, AND RESULT

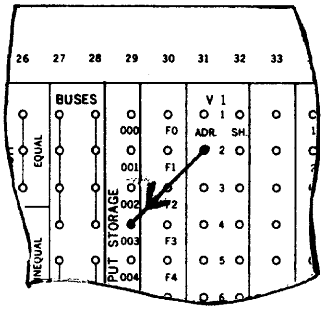
Value 1 and Value 2 are always wired to a storage location. In external programming, these storages may be of any of the following types: input-output, intermediate, or large capacity drum-revolver field. The designation used for a value is the numeric address of the storage field:

1.	Input-output	000 through 009
2.	Intermediate	10 through 39 (To 59 if additional are ordered)
3.	Large Capacity Drum	F0 through F19

Input-output storage address hubs are located on the connection panel in lines 29, A-J; 48, A-J; and 59, A-J. Intermediate storage address hubs are located in lines 29, K-x and 30, m-x; 48, K-x and 49 m-x; 59, K-x and 60, m-x. Large capacity drum storage field address hubs are located in lines 30, A-T; 49, A-T; and 60, A-T. Although each hub is a single hub, it may be reused without any danger of back feed. Although V_1, V_2, and R each has its own set of storage hubs, the storage hubs are actually common and may be used interchangeably.

The result of a step may be placed in one of the three types of storage units indicated for V_1 and V_2. The result of a step may also be placed in the ADR (Address register), or the CD (code distributor register). In this case, the designation would be ADR or CD, instead of a numeric storage designation.

The V_1 address hubs are located in line 31, A-x; V_2 address hubs are located in line 50, A-x; R address hubs are located in line 61, A-x. To use a particular storage as part of a step, it is merely necessary to wire from the value or result of the step to the storage desired. For example, to call on storage 003 as V_1 of step 2, the wiring would be:



The shift hubs associated with each value and result are explained in the section on Shifting.

For all processes, except transfer, the maximum size of the two values to be operated upon is 11 digits, plus the sign of the value. In transfer, the sign position may be used for a sign or for a numeric or alphabetic character, making the maximum size 12 digits. In multiplication and division all digits must be numeric; with addition and subtraction all digits should normally be numeric, although there is a feature called "alpha add"; in all other processes the values may be either numeric or alphabetic.

The maximum size of the result of an arithmetic process delivered to storage on one step is 11 digits, plus the sign of the result, although a larger answer may be developed during certain processes. The sign of the result is delivered automatically by the arithmetic unit. In transfer, the result may be either 11 characters plus sign or 12 characters.

Storage fields do not have a fixed, or predetermined, decimal location. All numbers are acted upon as if they were whole numbers. Both values and the result of each step can be independently decimally aligned by a feature called "shift". Any value or result can be shifted either to the right or to the left a maximum of 11 digits (except in certain cases of multiplication and division). This feature gives the programmer complete control over the decimal alignment of all factors.

There must be a V_1 wired on every step programmed. There must be a V_2 wired on every step, except where the process is transfer, left zero elimination, or channel search unequal.

In multiplication, V_1 is the multiplicand and V_2 is the multiplier. The program is speeded up if the multiplier is the value with the least number of digits.

In division, V_1 is the dividend, V_2 is the divisor.

In any arithmetic process and in the compare process, V_1 and V_2 may have the same storage location. In addition, subtraction, transfer, and left zero elimination the result may have the same storage location as V_1 or V_2.

The placing of a value into a storage unit as the result of a step will completely replace all information previously located in that storage unit, even though the new value is smaller than the old value, or contains only spaces or zeros. The new value will remain in the storage unit until replaced by other information.

If the result of an arithmetic process does not have 11 significant digits, the remaining digits will be delivered to the result storage as zeros. The only exception to this is a right shift in the result (see section on shifting).

For example, a result of: |-|-|-|-|-|-|1|2|3|4|5| will be delivered to storage as |0|0|0|0|0|0|1|2|3|4|5| . The arithmetic section "packs" zeros in the result.

In an operational process, spaces will be delivered to the result storage as spaces; zeros.will be delivered as zeros. The only exception to this rule is left zero elimination (see process section on left zero elimination).

3. SHIFTING

As mentioned above, the Univac File-Computer treats all values as if they were whole numbers. Through the feature called shift, the programmer can give the proper decimal alignment to V_1, V_2, or R.

All values are handled in storage fields of 11 digits plus the character for the sign of the value. The positions in a storage unit are numbered from right to left as follows:

|11|10|9|8|7|6|5|4|3|2|1|SN|

The registers in the arithmetic section of the computer in which a process occurs are also numbered in the same fashion. Without shifting, the 11 digits of a storage field will enter their respective positions in the register. The shift feature causes the value in the storage field to enter the register in a shifted position.

The sign position, however, cannot be shifted. If the sign position is used for a character other than the sign of the value, this fact must be considered if any shifting is attempted.

Shifting is used unconsciously when adding or subtracting with a pencil and paper...the values are physically aligned. For example, to add 12.3671 and 16.8, the values are not written:

```
1 2.3 6 7 1
      1 6.8
-----------
1 2 3 8 3 9
```

Rather, one of the values is written down and the other value shifted to the same decimal alignment. In this case, the second value could be shifted 3 places to the left in order to decimally align both values:

```
1 2.3 6 7 1
1 6.8
-----------
2 9.1 6 7 1
```

Now assume that 1 2.3 6 7 1 is V_1 in intermediate storage unit 22 in these positions: |0|0|0|0|0|1|2▲3|6|7|1|+| ; and that 16.8 is V_2 in input storage unit 001 in these positions: |0|0|0|0|0|0|0|0|1|6▲8|+| ; and the problem is to place the result in intermediate storage 23. If the program step is wired to add the two values, shifting neither, an incorrect answer is obtained ...just as with pencil and paper.

V_1	Pr	V_2		R
22	+	001	=	23

```
        Storage                                          Register

22  |0|0|0|0|0|1|2▲3|6|7|1|  enters register as  |0|0|0|0|0|1|2▲3|6|7|1|
+
001 |0|0|0|0|0|0|0|0|1|6▲8|  enters register as  |0|0|0|0|0|0|0|0|1|6▲8|

                                       Result      |0|0|0|0|0|1|2|3|8|3|9|

                           enters storage 23 as    |0|0|0|0|0|1|2|3|8|3|9|
```

However, if we arrange to shift V_2 three places to the left, as with pencil and paper, the correct result is obtained:

V_1	Pr	V_2		R
22	+	001(3L)	=	23

Note the use of (3 L) to indicate "3 shift left"

```
        Storage                                          Register

22  |0|0|0|0|0|1|2▲3|6|7|1|  enters register as  |0|0|0|0|0|1|2▲3|6|7|1|
+
001 |0|0|0|0|0|0|0|0|1|6▲8|  enters register as  |0|0|0|0|0|1|6▲8|0|0|0|

                                       Result      |0|0|0|0|0|2|9▲1|6|7|1|

                           enters storage 23 as    |0|0|0|0|0|2|9▲1|6|7|1|
```

The shift operation does not change the position of the value in the storage unit. It does change the position in which the value is received by the register in the arithmetic section. At the end of the program step in the last illustration, storage 001 still contains its value as: |0|0|0|0|0|0|0|0|1|6▲8|

Either value of a program step or the result of a program step may be shifted to the right or to the left a maximum of 11 digits. For instance, in the above example, assume that only one decimal place must be retained in the result. Two ways in which this can be accomplished are:

1. Shift V_2 three places to the left, then shift the result three places to the right.

2. Shift V_1 three places to the right.

```
(1)  V1    Pr    V2            R
     --    --    --           --
     22    +     001(3L)  =   23(3R)
```

```
        Storage                                         Register

22  |0|0|0|0|0|1|2▲3|6|7|1|  enters register as   |0|0|0|0|0|1|2▲3|6|7|1|
+
001 |0|0|0|0|0|0|0|0|1|6▲8|  enters register as   |0|0|0|0|0|1|6▲8|0|0|0|

                                         Result   |0|0|0|0|0|2|9▲1|6|7|1|

                           enters storage 23 as   |-|-|-|0|0|0|0|0|2|9▲1|
```

```
(2)  V1          Pr    V2         R
     --          --    ---        --
     22(3R)      +     001   =    23
```

```
    |0|0|0|0|0|1|2▲3|6|7|1|  enters register as   |-|-|-|0|0|0|0|0|1|2▲3|

    |0|0|0|0|0|0|0|0|1|6▲8|  enters register as   |0|0|0|0|0|0|0|0|1|6▲8|

                                         Result   |0|0|0|0|0|0|0|0|2|9▲1|

                           enters storage 23 as   |0|0|0|0|0|0|0|0|2|9▲1|
```

Note in the first illustration above, a result is shifted as it is received by the storage unit, not as the process is occuring in the register.

On a left shift (V_2 in the first example), zeros are "packed" to the right of the value. On a right shift, however, (R in the first example, V_1 in the second example), spaces are entered to the left of the number.

When digits in V_1 or V_2 are shifted to the right or to the left beyond the 11 positions of the register, they are dropped off. They do not affect the result in any way.

Each value and result of a program step has a shift hub which can be wired. V_1 shift hubs are located on the connection panel in line 32, A-x; V_2 shift hubs in line 51, A-x; and R shift hubs in line 62, A-x. Close to these hubs are the ones indicating whether the shift is to be left or right, and the number of places that the value is to be shifted. The left shift hubs are located in lines 34, A-K and a-k; 53, A-K and a-k; 64 A-K and a-k. The right shift hubs are located in lines 34, M-X and m-x; 53, M-X and m-x; and 64 M-X and m-x. The shift hubs for the value or result are wired to the desired left or right shift. The shift hubs may be bussed if necessary.

The particular rules for shifting connected with each process are explained in the process section.

4. PROCESSES

Each program step has available nine arithmetic or operational processes, any one of which may be used on any program step. One of them must be used in every step called upon by the machine.

The four arithmetic processes are:

Addition
Subtraction
Multiplication
Division

These processes always involve two values (V_1 and V_2), each of which may be a maximum of 11 digits plus sign. The values are usually numeric (see addition for the only use of alphabetic information in an arithmetic process.)

The values and their signs are handled according to algebraic rules; and the result of the process is delivered to storage with the sign of the result. Since the maximum size of a storage unit is 11 digits plus sign, this is also the maximum size of the result which may be delivered on one arithmetic step.

The five operational processes are:

Transfer
Compare
Equal Channel Search
Unequal Channel Search
Left Zero Elimination

These processes always involve a V_1. Compare and channel search equal always involve a V_2. The values may be alphabetic or numeric. There will be a result of all operations except compare. In these operational processes a value or a result may contain 11 digits plus the sign of the value. Transfer may contain 12 digits of alphabetic or numeric information.

The process hub for each step is located in line 21, A-x. This hub is wired to the desired process, the location of which will be discussed in detail below for each process.

A. ADDITION

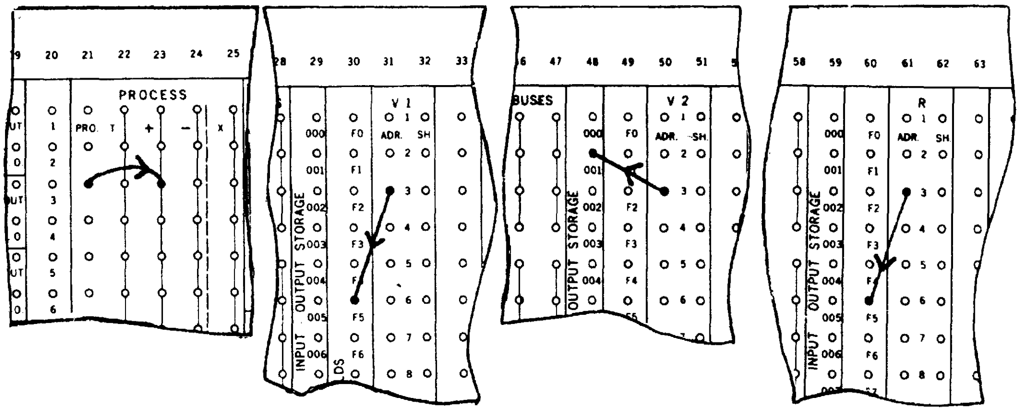
This process is used when adding V_1 and V_2 together to obtain a sum, or for combining alphabetic and numeric data for output storage or similar purposes. The symbol used in programming for this process is +. The addition process hubs are located in line 23, A-x. As a sample step, assume that the problem is to accumulate sales-to-date. The values are:

Input-output storage 001:	Sales today
Drum revolver field F5:	Sales-to-date

Step 3)	V_1	Pr	V_2	=	R
	F5		001	=	F5

(Sales-to-date) (Sales today) = (New sales-to-date)

The wiring on the connection panel for the above step would be:



The special rules for addition are as follows:

(1) For normal addition, both V_1 and V_2 must be numeric. When the corresponding positions in both V_1 and V_2 are numeric, the quantities are always added.

V_1 |0|0|0|0|0|0|0|1|3|8|9▲|+|

+

V_2 |-|-|-|-|-|-|-|-|-|2|3▲|+|

=

R |0|0|0|0|0|0|0|1|4|1|2▲|+|

(2) If in the corresponding positions, either V_1, V_2, or both, are alphabetic, the character in V_1 will always be transmitted to the result. If in the same values, certain corresponding positions are both numeric, addition will occur in these positions

V_1	-	R	M	X	-	-	-	-	-	-	-	+
+												
V_2	-	-	-	-	-	3	7	2	9	-	-	+
=												
R	0	R	M	X	0	3	7	2	9	0	0	+

(3) If an ignore code occurs in V_1, the corresponding character of V_2 will be transmitted to the result. Ignore codes occurring in V_2 are considered alphabetic and Rule (2) applies.

V_1	-	-	-	2	7	-	i	i	-	-	-	+
+												
V_2	-	-	-	-	-	-	R	X	7	4	2	+
=												
R	0	0	0	2	7	0	R	X	7	4	2	+

(4) Whenever a character is transmitted to the result by rule 2 or 3, any carry into this position caused by addition in the position to the right, will be destroyed.

V_1	-	-	8	B	X	i	i	2	4	0	0	+
+												
V_2	-	-	i	-	-	2	4	7	8	R	M	+
=												
R	0	0	8	B	X	2	4	0	2	0	0	+

(5) Spaces are considered numeric.

(6) The left hand digit of a result must fall within the 11 digit register. If it does not, the machine will signal an overflow condition. For example:

		11	10	9	8	7	6	5	4	3	2	1	SN
V_1		8	7	4	0	9	2	-	-	-	-	-	+
+													
V_2		6	0	2	1	3	9	-	-	-	-	-	+
=													
R	1	4	7	6	2	3	1	0	0	0	0	0	+

The above addition developed a 1 in the twelfth position. In this case, the step exit will be obtained from the +/- overflow hub rather than the normal step exit hub, thus indicating an overflow. By shifting V_1 and V_2 to the right, this could be avoided. It is not possible to shift R to avoid this condition, since the overflow is signalled at the time the carry occurs in the register.

V_1	(1 R)	8	7	4	2	9	2▲	-	-	-	-	-	+
+													
V_2	(1 R)	6	0	2	1	3	9▲	-	-	-	-	-	+
=													
R		1	4	7	6	2	3	1▲	0	0	0	0	+

(7) As mentioned earlier, it is essential to align decimal points before doing addition. The following examples show some of the possibilities.

(a)

V_1	(1 R)	-	-	-	-	6	7	6	5▲	4	2	-	+
+													
V_2	(3 R)	-	-	7	6	3	4▲	2	5	-	-	-	+
=													
R		0	0	0	0	1	4	3	9	9▲	6	7	+

In the above example, it would also have been possible to shift V_2 two to the right and R one to the right. In this case, the result would have been:

-	0	0	0	1	4	3	9	9▲	6	7	+

(b)

V_1		-	-	-	6	4	3▲	2	9	4	-	-	+
+													
V_2	(I R)	-	7	6	2	3▲	7	0	6	6	5	-	+
=													
R	(3 R)	-	-	-	0	0	8	2	6	7▲	0	0	+

If in this case, V_1 were shifted 3 right, and V_2 4 right, the result would be:

0	0	0	0	0	0	8	2	6	6▲	9	9	+

The difference is due to the fact that .004 would be dropped off in V_1 and .00665 in V_2. The sum of the values dropped is .01065, which affects the result to the extent of .01. The programmer must remember not to shift digits in either V_1 or V_2 outside of the register if they might have significance in the result.

(8) The verification of an addition step by repeating the step cannot be eliminated for addition. Therefore, even if "no check" is wired, checking will occur.

(9) The storage location of V_1 or V_2 of an addition step may be used to store the result.

A summary of the rules for alpha add is given below:

V_1 V_2	Numeric	Alpha	Ignore
Numeric	Sum	Alpha *	Numeric *
Alpha	Numeric*	Alpha (V_1)	Alpha
Ignore	Numeric	Alpha	Ignore

* A carry (borrow) generated by the addition of numbers in preceding digit position is destroyed.

B. SUBTRACTION

This process is used when subtracting the storage wired as V_2 from the storage wired as V_1. The symbol used in programming for this process is -. The subtraction process hubs are located in line 24, A-x. As a sample step, assume that the problem is to subtract quantity ordered from the on-hand balance to determine the new on-hand balance.

The values are:

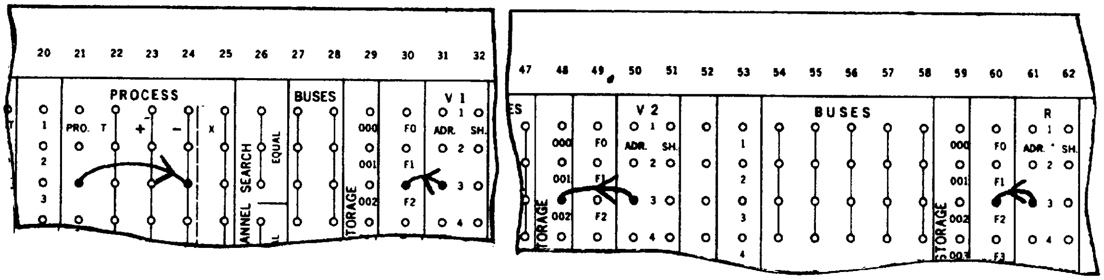
Input-output storage 002:	Quantity ordered
Drum revolver field F2:	On-hand balance

The step would be:

Step 3) $\frac{V_1}{F2}$ $\frac{Pr}{-}$ $\frac{V_2}{002}$ = $\frac{R}{F2}$

(On-hand bal.)-(Quan. ordered) = (New on-hand bal.)

The wiring on the connection panel for the above step would be:



An overflow condition might develop for subtraction, as well as for addition, because of the rules of algebra.

V_1 |6|3|8|4|5|3▲|-|-|-|-|-|+|

-

V_2 |5|0|0|2|1|1▲|-|-|-|-|-|-|

=

R |1|1|3|8|6|6|4▲|0|0|0|0|0|+|

As in the case of addition, the step exit will come from the +/- overflow hub rather than the normal step exit.

The same special rules apply to subtraction as were listed above for addition, with the two following exceptions:

(1) Whenever a character is transmitted to the result due to alphabetic information or an ignore code (see addition, rules 2 and 3), the borrow which would normally occur from that position is destroyed.

(2) If alphabetic characters or ignore codes appear in either V_1 or V_2, the sign of V_1 will be transmitted as the sign of the result.

C. MULTIPLICATION

This process is used when multiplying V_1 by V_2. The symbol used for this process is X. The multiplication process hubs are located in line 25, A-L and a-1. As a sample step, assume that the problem is to multiply hours times rate to determine straight time pay. The values are:

Input-output storage 001: Hours

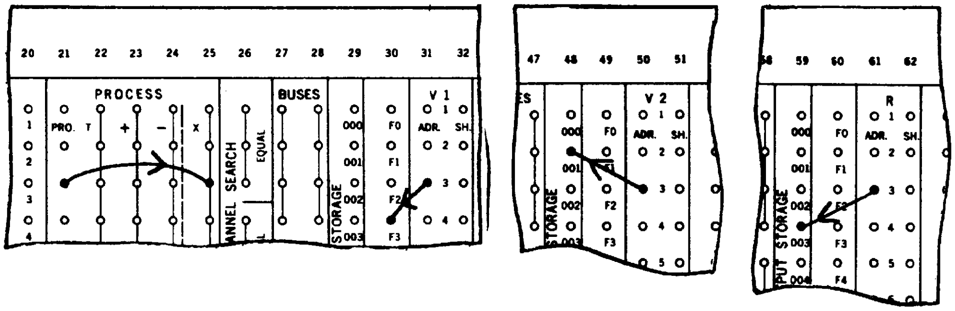
Drum revolver field F3: Hourly rate

Place straight time pay in input-output storage 003.

The step would be:

	V_1	Pr	V_2	=	R
Step 3)	F3	X	001		003
	(Rate)	X	(Hours)	=	(Straight time pay)

The wiring on the connection panel for the above step would be:



To round the result of the multiplication requires an additional step. This is usually done by adding .005 to the result. If a man earned $15.2845 (unrounded), the addition of .005 would make the rounded earnings $15,2895; if his earnings were $15.2878. the rounded earnings would be $15.2928. If the result is to be used in further calculation, it is important to shift the result so as to drop off the position in which the rounding 5 has been added.

The step to accomplish this might be:

V_1	Pr	V_2	=	R
003 (1R)	+	29		009 (1R)
(S.T.Pay)	+	(5)	=	(Rounded S.T.Pay)

(1R) V1 |0|0|0|0|0|1|5▲2|8|7|8|+|

+

V_2 |-|-|-|-|-|-|-|-▲-|-|5|+| This would have been entered previously in intermediate storage 29.

=

(1R) R |-|-|0|0|0|0|0|1|5▲2|9|+|

The special rules for multiplication are:

(1) Both values in multiplication must be numeric.

(2) Unless the verification of each multiplication step is prevented by the wiring of "no-check", it is impossible to use a normal result shift in multiplication. Because of the method of multiplication, the File-Computer would attempt to compare the entire result as recomputed with the shifted result; therefore the step would never check. In most cases, however, a rounding step follows multiplication. As long as the total number of digits in the product does not exceed 11, the insignificant digits can be shifted off to the right during the rounding step. However, if the total number of digits in the result exceeds 11, a normal right shift does not move the digits to the left of the 11th position; therefore, the value would lose significant digits on the left. This is more fully explained below.

If verification of each step is prevented by the wiring of "no-check", result shifting can take place, subject to the limitations in the following paragraphs.

With the possibility of an 11 digit multiplier and an 11 digit multiplicand, a 22 digit product can be developed. The arithmetic section of the File-Computer contains 4 registers, as follows:

A

11	10	9	8	7	6	5	4	3	2	1

B

11	10	9	8	7	6	5	4	3	2	1

C

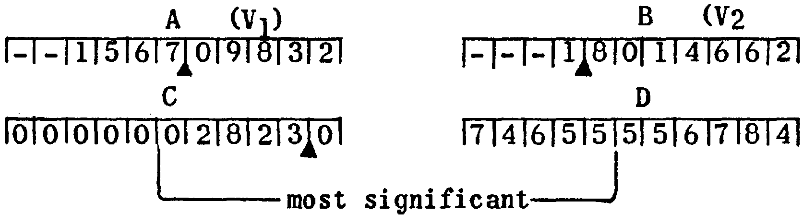
11	10	9	8	7	6	5	4	3	2	1

D

11	10	9	8	7	6	5	4	3	2	1

On a multiplication step, V_1 (multiplicand) is placed in the A register; V_2 (multiplier) is placed in the B register; and the product is developed in the C and D register, with the least significant digit of the product in the 1 position of the D register, and the other digits developing to the left. The product is always transferred from the D register into the result storage. With a product of more than 11 digits, the additional digits will develop in the C register. A normal result shift will shift only the D register; therefore, any attempt to shift significant digits from the C register into the D register would actually result in the entry of spaces in the left of the D register, rather than the digits from the C register.

Since the maximum size of a storage field is 11 digits, no more than 11 digits can be retained as a product. In most cases of a product of over 11 digits, it is more important to retain the 11 digits to the left, rather than the 11 digits to the right.



When the product is more than 11 digits, it is possible to retain either the 11 most significant digits of a product or the 11 least significant digits on one step. It is also possible to drop off insignificant digits to the right, thus retaining less than 11 digits in the product.

The principle involved is the shifting of V_1 and V_2 to the left...so that all of the desired product digits fall within the C register. Then, by wiring the result shift to a special shift hub known as "EX. SH.", all the digits in the C register are shifted to their same relative position in the D register. The EX. SH. hubs are located in line 64, X and x.

To determine the total number of shifts required in V_1 and V_2.

a. Determine the maximum size of the product (the sum of the number of digits in V_1 and V_2). A convenient way of doing this is making an X for each digit in V_1 and V_2, and placing the decimal point in its proper location. The product in the example above could be:

X X X X X▲X X X X X X X X X X X X

b. Determine how many digits are insignificant--how many are to be dropped off to the right. The formula for doing this is:

V_1 decimals + V_2 decimals - Desired result decimals = drop off digits.

Decimals are the number of storage positions to the right of the decimal point.

In the example shown above, assume that it is desirable to retain 6 digits following the decimal point.

V_1 dec.	+	V_2 dec.	-	R dec.	=	Drop off digits
5	+	7	-	6	=	6

c. The formula for computing total left shifts of V_1 and V_2 is:

11 - drop off digits = V_1 left shift + V_2 left shift.

In assigning the left shifts, shift V_1 as many places to the left as there are spaces to the left of the first significant digit; then shift V_2 the remaining number required. For example:

Assume V_1 =

-	-	1	5	6	7	0	9	8	3	2

and V_2 =

-	-	-	1	8	0	1	4	6	6	2

(a) Retaining the 11 most significant digits

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

Product Drop off

Drop off V_1 + V_2
11 - 6 = 5

V_1 : 2 left shift
V_2 : 3 left shift
R : EX.S

Would be computed as:

0	2	8	2	3	0	7	4	6	5	5	5	5	6	7	8	4	0	0	0	0	0

Product

(b) Retaining less than 11 digits in product

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

Product Drop Off

Drop off V_1 + V_2
11 - 9 = 2

V_1 : 2 left shift
V_2 : 0 left shift
R : EX.S

Would be computed as:

0	0	0	0	2	8	2	3	0	7	4	6	5	5	5	5	6	7	8	4	0	0

Product

(c) Dropping off the maximum of 11 digits

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

Product Drop Off

Drop off V_1 + V_2
11-11 = 0

V_1 : 0 left shift
V_2 : 0 left shift
R : EX.S

This would be computed just as it stands

```
Assume V1 = |-|-|-|-|-|-|-|5|1|2|5| and V2 = |-|-|-|-|-|-|-|6|0|5|5|
                               ▲                           ▲
```

Retain 6 most significant digits of product

```
|11|10|9|8|7|6|5|4|3|2|1|| 11|10|9|8|7|6|5|4|3|2|1|  Drop off V1 + V2
| 0| 0|0|0|0|0|0|0|0|0|0||  0| 0|0|3|1|0|3|1|8|7|5|   11-2 = 9
                                  |     ▲     |   |
                                  |           |   |  V1 : 7 left shift
                                  └─Product───┘Drop┘
                                               off   V2 : 2 left shift
                                                     R: EX.S
```

Would be computed as:

```
|0|0|0|0|0|3|1|0|3|1|8||7|5|0|0|0|0|0|0|0|0|0|
|              ▲      |
└──────Product────────┘
```

Note that this method can be used for dropping off digits to the right when the product is less than 11 digits, although it would be a faster machine operation to drop off insignificant digits on the next step when rounding.

3. If "no check" is wired, the location of the result may be the same as the location of V_1 or V_2. However, if it is not wired the location of the result must be different from both V_1 and V_2, because the check operation reaccesses the operands from V_1 and V_2.

4. The program is speeded up if V2 is the value whose sum of the digits is least.

5. The sign of the product is delivered to the result storage according to the laws of algebra.

D. DIVISION

This process is used when dividing V_1 by V_2, in order to obtain the quotient or remainder. The division process hubs are located in line 25, M-X and m-x. The programming symbol for this process is ÷ . As a sample step, assume that the problem is to compute an average hourly rate. The values are:

Input-output storage 003 = Total hours 1 decimal

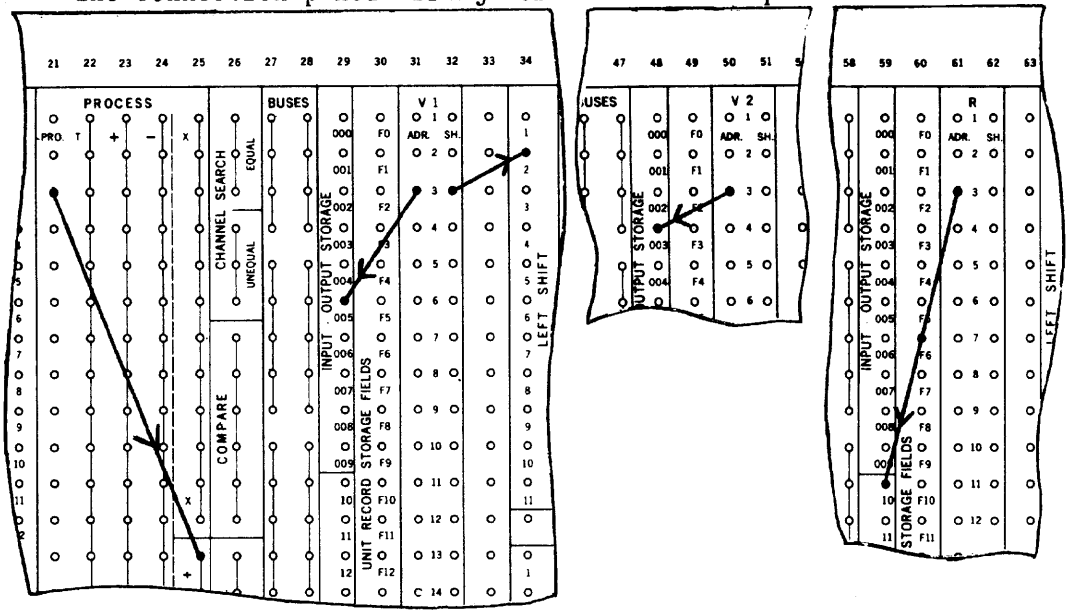
Input-output storage 005 = Gross pay 2 decimals

Place average rate in intermediate storage 10 3 decimals

This step would be:

	V_1	Pr	V_2	=	R
Step 3)	005 (2 L)	÷	003		10
	(Gross Pay)		(Total Hours)		(Average Rate)

The connection panel wiring for the above step would be:



The special rules for division are as follows:

(1) Both values in division must be numeric.

(2) To determine the number of shifts necessary to obtain a quotient with the desired number of decimals, the following rule is used:

If Rd - (V_{1d} - V_{2d}) =

+ and < 12 : V_1 Left Shift

+ and $\geqslant 12$: V_1 (11 Left Shift) + V_2 right shift to = total Shifts required

- : V_2 Left Shift or R right Shift (only if "no-check" is wired)

Note 1: "d" in above formula means decimals.

Note 2: V_1 left shift on the division process does not drop off any digits to the left in V_1, but merely acts to increase the number of quotient digits. Decimals refer to the number of storage positions to the right of the decimal point.

In the sample step above, the average rate might be carried to 2, 3 or 4 decimals. Using the formula, the shifts in each case would be computed as follows:

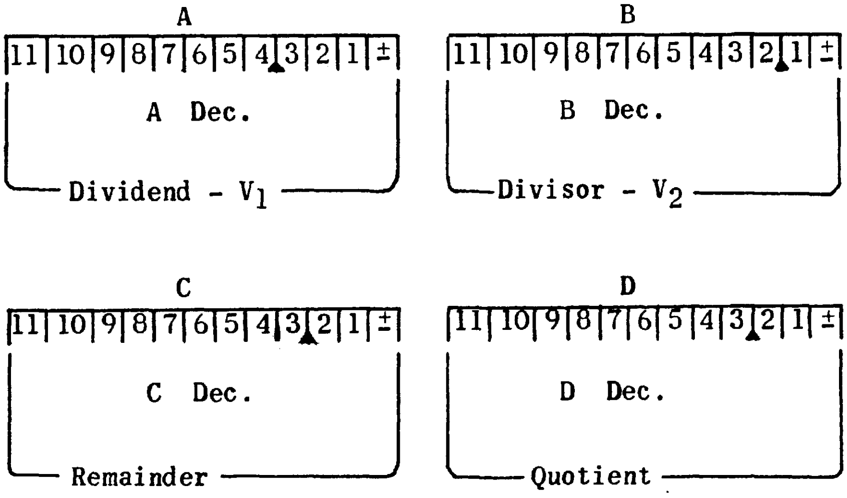
R dec.	-	V_1 dec.	+	V_2 dec.	=	Shift
2	-	2	+	1	=	+1. V_1 left shift of 1
3	-	2	+	1	=	+2. V_1 left shift of 2
4	-	2	+	1	=	+3. V_1 left shift of 3

A V_2 left shift might occur as follows:

R dec.	-	V_1 dec.	+	V_2 dec.	=	Shift
1	-	3	+	1	=	-1. V_2 left shift of 1

(3) Either an 11 digit quotient or an 11 digit remainder can be obtained on any one step. Any attempt to obtain a quotient of greater than eleven digits will result in the step exit emitting from the divide overflow hub, rather than the normal step exit.

The values and results of the division process appear in the computer's registers as follows:



Quotient decimal = V_1 dec. - V_2 dec. + V_1 left shift + V_2 right shift
- V_1 right shift - V_2 left shift

Remainder decimal = V_1 dec. + V_1 left shift
- V_1 right shift

As can be seen, the quotient is normally delivered as the result. If it is desired to deliver the remainder as the result rather than the quotient, it is necessary to give the result a special shift instruction (Ex.SH).

This shift transfers the remainder into the D register and the quotient into the C register.

A sample problem illustrating the use of the remainder is:

Drum Revolver Field F3 = Sales-to-date (in units)

Intermediate Storage 39 = Constant value of 12

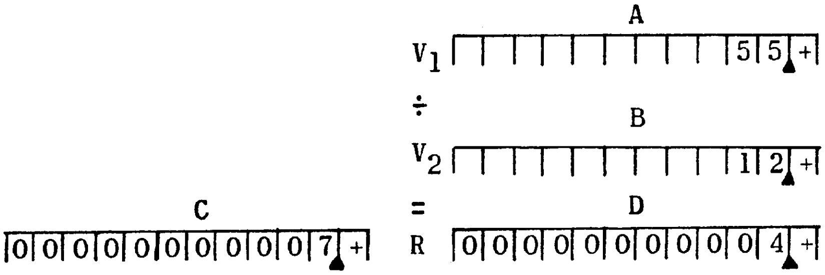
Place sales-to-date (in dozens and fractions of dozens) in output storage 005

The program would be:

Step #	V1	SH	Pr	V2	SH	R	SH	Next Step #
15	F3 (Sales)		÷	39 (12)		10 (Dozens)		16
16	F3 (Sales)		÷	39 (12)		11 (Fractions)	EX SH	17
17	10 (Dozens)	2L	+	11 (Fractions)		005 (Dozens & Fractions)		18

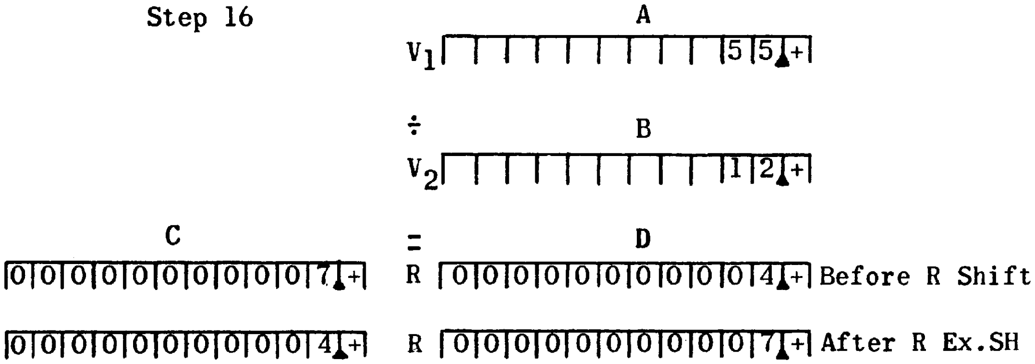
The registers would appear as:

Step 15



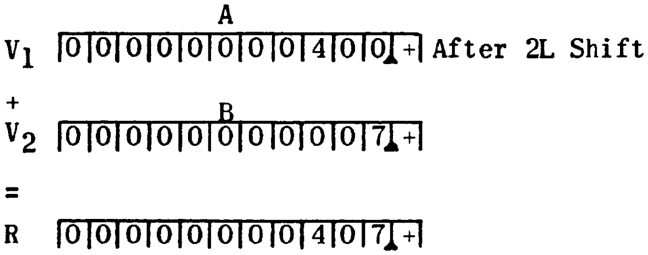
To 10

Step 16


A
V_1 5 5 +
÷
B
V_2 1 2 +
C
=
D
0 0 0 0 0 0 0 0 0 0 7 +
R 0 0 0 0 0 0 0 0 0 0 4 + Before R Shift
0 0 0 0 0 0 0 0 0 0 4 +
R 0 0 0 0 0 0 0 0 0 0 7 + After R Ex.SH

To 11

Step 17


A
V_1 0 0 0 0 0 0 0 0 4 0 0 + After 2L Shift
+
B
V_2 0 0 0 0 0 0 0 0 0 0 7 +
=
R 0 0 0 0 0 0 0 0 4 0 7 +

To 005

Of course, in one step the quotient could be delivered as:

0	0	0	0	0	0	4▲	5	8	3	3	+

which would be the decimal equivalent of 4 7/12 dozen.

The program step would be:

V_1	SH	Pr	V_2	SH	R	SH
F3	4L	÷	39		005	
(Sales)			(12)		(Sales dozens and decimal fractions)	

The four left shifts of V_1 are required to obtain the four decimals in the result as determined from rule 2.

(4) If division is made with a zero divisor (V_2), the quotient will always be 0.

(5) Unless "no-check function" is wired, the storage location of the result should not be the same as the location of V_1 or V_2. If "no-check" is wired, this limitation does not apply.

(6) The sign of the quotient is determined according to the laws of algebra. The sign of the remainder is the sign of the dividend.

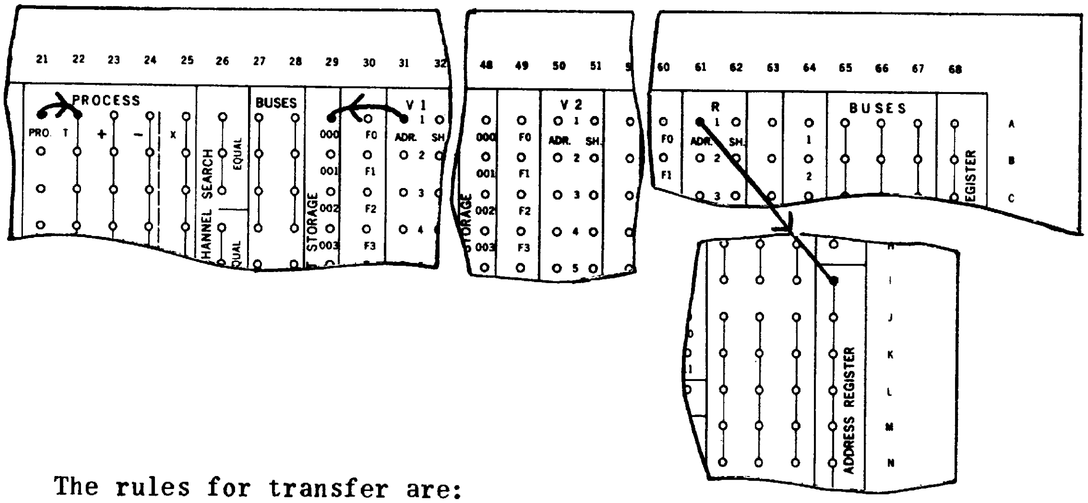
E. TRANSFER

This process is used to transfer information from one storage to another, or from a storage into the ADR or CD. The transfer hubs are located in line 22, A-x. The programming symbol for this process is T. As a sample step, assume that drum section and channel are located in input-output storage 000, and it is desired to transfer them to the ADR.

The program step would be:

Step 1) $\dfrac{V_1}{000}$ $\dfrac{PR}{T}$ $\dfrac{V_2}{-}$ = $\dfrac{R}{ADR}$
(drum sect. & channel) (address register)

The connection panel wiring for this step would be:



The rules for transfer are:

(1) The value to be transferred may be numeric, alphabetic, or a combination of alphabetic and numeric.

(2) The sign of the value is automatically delivered as the sign of the result.

(3) With a transfer operation, there is only one value, unless the transfer is a masking transfer as covered in rule 6. The value to be transferred is always wired as V_1.

(4) During transfer the value to be transferred may be shifted, giving it a different decimal alignment or dropping off digits to the right or to the left.

With a value in input-output storage 005 to be transferred to intermediate storage 10, some of the results which can be obtained are:

V_1 storage 005 = |-|-|1|2|3|4|5|6|7|-|-|+|

	V_1	PR	V_2	= R	
(a)	005	T	-	10	|-|-|1|2|3|4|5|6|7|-|-|+|
(b)	005 (2R)	T	-	10	|-|-|-|-|1|2|3|4|5|6|7|+|
(c)	005 (5R)	T	-	10	|-|-|-|-|-|-|-|1|2|3|4|+|
(d)	005 (4L)	T	-	10	|3|4|5|6|7|-|-|0|0|0|0|+|
(e)	005 (4L)	T	-	10 (8R)	|-|-|-|-|-|-|-|-|3|4|5|+|

(5) The transfer process does not pack zeros. A value such as: |-|-|-|-|-|A|B|-|1|0|5| will be transferred as: |-|-|-|-|-| |A|B|-|1|0|5| Of course, if a value is shifted to the left during a transfer process, the rule for a left shift applies; that is zeros will be placed in the positions to the right (See example d above).

(6) A "masking" transfer permits the transfer of a value, while, at the same time, any desired digits are suppressed and replaced by ignore codes. The programming symbol for this process is T. As a sample problem, assume that a portion of a part number is to be transferred to the ADR. The values are:

Input-output storage 000:	Part number
Intermediate storage 20 :	Word to be used in masking part number
Intermediate storage 11 :	Blank

The program could be:

(1)	V_1	Pr	V_2	=	R
	000 (part number)	T	20 (masking word)		10 (2 R)

```
                  ┌────────────────────► Drum Section
                ┌─┴─┐     ┌──────────► Channel
                │   │   ┌─┴─┐
V1   |A|B|1|1|4|M|2|9|9|1|X|
T
V2   |-|-|-|-|-|9|9|0|0|0|0|
=
R    |-|-|A|B|1|1|4|i|i|9|9|
```

(2)	$\underline{V_1}$	$\underline{Pr}$	$\underline{V_2}$	=	$\underline{R}$
	10	+	11		ADR
	(drum sect. channel)		(blank storage)		

V_1	-	-	A	B	1	1	4	i	i	9	9
+											
V_2	-	-	-	-	-	-	-	-	-	-	-
=											
R	0	0	A	B	1	1	4	0	0	9	9

Note that two steps were required to accomplish this, because ignore codes will not be accepted by the ADR. Since the ADR will accept only the last six digits of a value, it was not necessary to mask the five digits to the left of the storage.

The rules for masking transfer are:

(a) V_1 is the value in which certain characters are to be suppressed. It may be numeric, alphabetic, or a combination of the two.

(b) V_2 is the value which causes the characters in V_1 to be suppressed and replaced by ignore codes. The masking code is determined by a zero or any other digit in V_2. A zero in V_2 will transfer the character in V_1 to the result. Any other digit in V_2 will transfer an ignore code to the result. V_2, the "masking word", will in most cases be stored in intermediate storage as a constant.

(c) A masking transfer does not pack zeros.

F. COMPARE

This process is used when V_1 is to be compared with V_2 to determine whether V_1 is equal to, greater than, or less than V_2. By wiring to an interstep function known as branching, the results of the comparison are made available. The programming symbol for the process is C. The compare hubs are located in line 26, G-L and g-l.

For example, assume an invoicing problem with a tabulating card input-output device, where the cards represent the items on a customer order grouped together by customer number. The problem is to determine when the first card of a new order enters the machine, so that the dollar total of the preceding order can be accumulated with orders to date in the customer's URA. The values are:

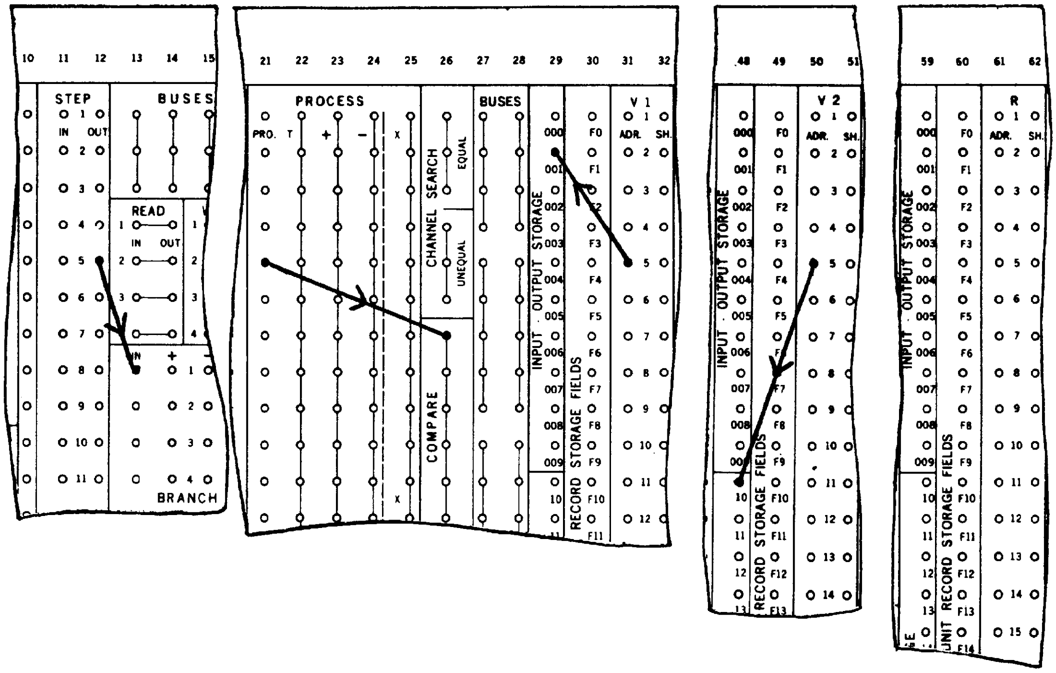
Input-output storage 001: Customer number

Intermediate storage 10 : Customer number (stored from the first card of the order)

The program step would be:

	V_1	Pr	V_2	=	R	Next Step
Step 5)	001	C	10		-	BR 1

The connection panel wiring for the above is as follows:



The rules governing compare are:

(1) V_1 and V_2 may be numeric, alphabetic, or a combination thereof. Comparisons are made in accordance with the following sequence.

Minus $<$ space = 0 $<$ 1 $<$ ---9 $<$ A--- $<$ R $<$ + $<$ S--- $<$ Z

(a)	V_1	-	-	-	-	-	-	1	2	3	4	5	+	$V_1 > V_2$
	V_2	-	-	-	-	-	-	-	1	8	9	9	+	

(b)	V_1	-	-	-	A	B	D	9	8	7	6	5	+	$V_2 > V_1$
	V_2	-	-	-	A	C	C	1	2	3	4	4	+	

(2) Zeros and spaces are considered equal by the compare process. An ignore code in V_1 or V_2 will suppress comparison of that digit.

V_1	-	-	-	1	4	i	5	6	-	4	0	+	$V_1 = V_2$
V_2	0	0	0	1	4	A	5	6	0	i	0	+	

(3) The signs of the two values are considered according to the laws of algebra during the compare process. This sign comparison is made first by the computer followed by the highest order digit comparison and proceeding to the least significant digit (not including sign position). Any code in the sign position is considered positive unless a negative code appears in this position. A sign code in any other position will follow the sequence of Rule (1).

(a)	V_1	-	-	-	-	-	1	2	3	4	5	6	+	$V_1 > V_2$
	V_2	0	0	0	0	0	1	2	3	4	5	6	-	

(b)	V_1	-	-	-	-	-	-	-	2	5	3	4	-	$V_2 > V_1$
	V_2	-	-	-	-	-	-	-	-	-	4	6	-	

(4) No Result (R) is wired on a compare step.

(5) The "next step" of a compare step is wired into a branch which has three "out" hubs: +, -, and 0. The result of the comparison is indicated as follows:

+ V_1 is greater than V_2

- V_1 is less than V_2

0 V_1 is equal to V_2

Each of these hubs is wired to the next step or interstep function desired.

(6) V_1 and V_2 may be shifted during the compare process to align the decimals or to drop off digits to the right or to the left. Comparison takes place after the shifts are completed.

(a) V_1 C V_2 (2R)

V_1	-	-	-	-	-	-	1	2	3	4	5	+
V_2	-	-	-	-	1	2	3	4	5	0	0	+

V_1 is equal to V_2

Without the shift, V_1 is less than V_2

(b) V_1 (6L) C V_2 (6L)

V_1	-	A	B	C	D	E	1	2	3	4	5	+
V_2	F	G	H	I	J	K	1	2	3	4	5	+

V_1 is equal to V_2

Without the shift, V_1 is less than V_2

G. LEFT ZERO ELIMINATION

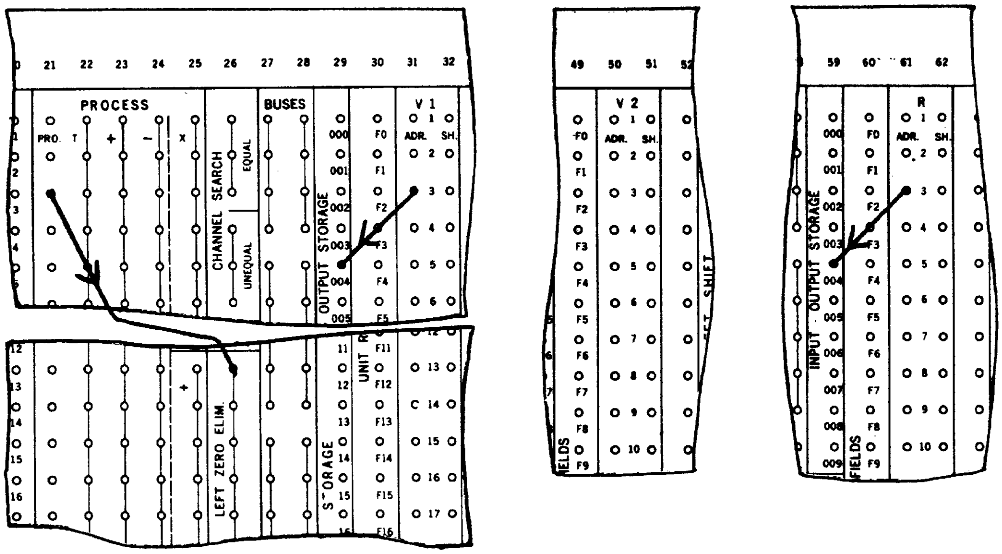
This process is used to replace zeros to the left of the first significant digit in a value with space codes. The programming symbol for the process is LZE. The hubs are located in line 26, M-R and m-r.

With a tabulating card input-output device, it may be desirable not to punch preceding zeros in a value. With an amount in input-output storage 004, the program step to accomplish this would be:

Step 3)	$\frac{V_1}{004}$ (amount)	$\frac{Pr}{LZE}$	$\frac{V_2}{-}$	=	$\frac{R}{004}$ (amount)

V_1	0	0	0	0	0	0	1	2	5	5	0	+
R	-	-	-	-	-	-	1	2	5	5	0	+

The connection panel wiring for the above step would be:



The rules governing left zero elimination are:

(1) V_1 may be numeric, alphabetic, or a combination thereof.

(2) The process operates by shifting V_1 to the left until the first significant digit is positioned in the 11th character position. It then shifts the value to its original position, replacing any zeros to the left with space codes. The storage location of the result may be any storage location desired, thus affecting a transfer at the same time as suppressing zeros.

(3) No V_2 is wired on a left zero elimination step.

(4) Both V_1 and R may have an external shift wired.

V_1	Pr	V_2	-	R	
004	LZE	-		004	(3 R)

V_1	0	0	0	0	8	3	6	5	2	9	4	+
R	-	-	-	-	-	-	-	8	3	6	5	+

H. CHANNEL SEARCH

There are two channel search processes - channel search equal and channel search unequal. Both are used to search the channel of the large capacity storage drum whose drum section and channel number are in the ADR.

A channel search process is used only when a full address (drum section, channel and URA) is not available; it is a means of locating the URA when only the drum section and channel are known. The equal process hubs are located in line 26, A-C and a-c; the unequal process hubs are located in line 26, D-F and d-f.

EQUAL CHANNEL SEARCH

Channel search equal compares two identifiers with the field in each URA in which the identifier has been stored. Either of the two identifiers may be at input or intermediate storage, however, they do not have to be wired to the same storage. If either of the two identifiers match that of a URA, that URA is written on the revolver. If neither of the two identifiers match, the last URA on the channel is written on the revolver. The programming symbol for this process is ECS.

The purpose of having two possible identifiers in a channel search-equal step is the simplification of certain loading routines. In these routines the second identifier will usually contain only spaces or zeros. This use of channel search is covered in section IV. In almost all other cases, only the identifier from input will actually be used. However, because of the provision for two identifiers, if only one is required it must be wired as both V_1 and V_2 of a program step. This wiring is normal wiring of V_1 and V_2 to the same storage.

The result of a channel search-equal step is wired to the channel search identifier field in which the identifier has been stored. Note that these hubs, located in line 58, O-k, are used only during the channel search process. The unit record area storage fields, line 60, A-T, are used at all other times when referring to a field on the drum revolver.

As an example of the programming required to complete an equal channel search operation, assume the following:

Input-output storage 000:	Drum section, channel
Input-output storage 001:	Complete part number
Drum field FO :	Complete part number

The two program steps required to locate the URA and write it on the revolver (assuming there is no channel overflow) would be:

	V_1	PR	V_2	=	R	Next step
(1)	000	T			ADR	2

(drum section, channel)

(2)	001 (part no.)	ECS	001 (part no.)		CSFO (part no.)

ECS Out

V_1 =	V_2 =	≠
Step 3	Step 3	Step 6

Substituting values in the above steps:

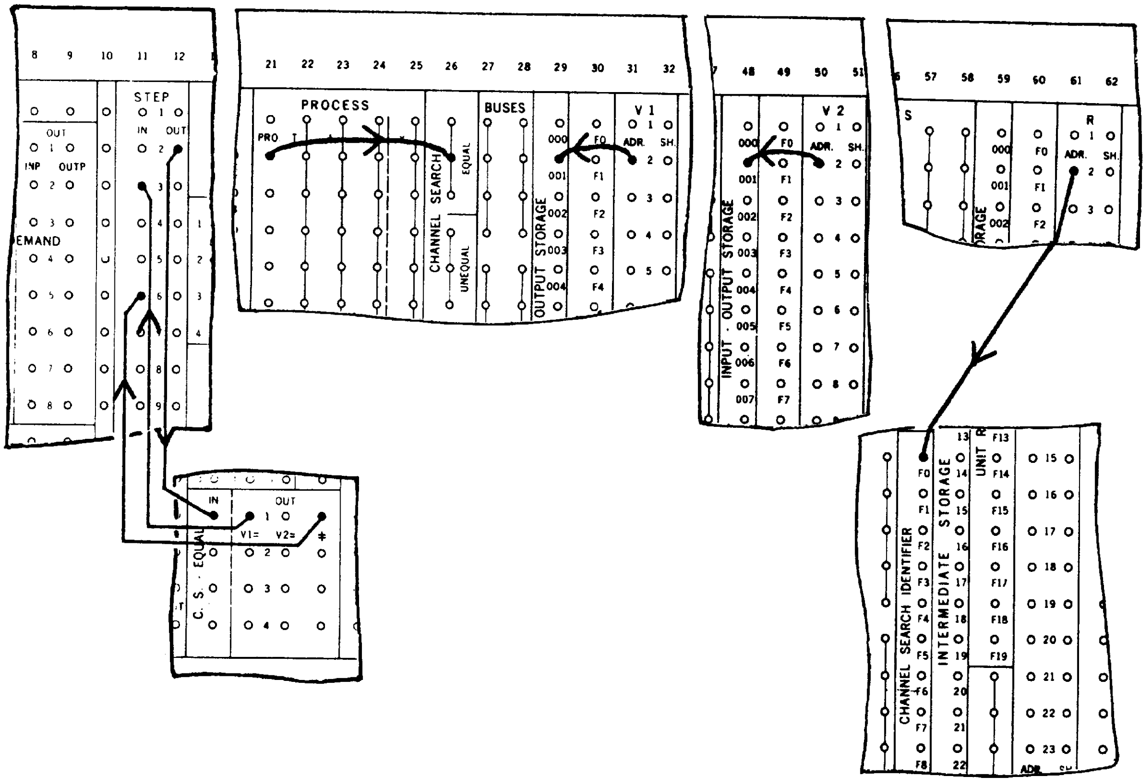
(1) V_1 (000) |-|-|-|-|-|1|6|3|3|-|-|+|
T
=
R (ADR) $$\left|-|-|-|-|-|\underbrace{1|6}_{\text{(drum sect.)}}|\underbrace{3|3}_{\text{(channel)}}|\underbrace{-|-}_{\text{(URA)}}|+\right|$$

(2) V_1 (001) |-|-|-|A|B|1|0|9|6|3|3|+|
ECS
V_2 (001) |-|-|-|A|B|1|0|9|6|3|3|+|

R (CSFO) |-|-|-|A|B|1|0|9|6|3|3|+|

As of the beginning of step 3, all fields in the unit record area of part no. AB109633 are available for use.

The connection panel wiring for step 2 would be:



The rules governing the equal channel search process are:

(1) V_1 and V_2 may be numeric, alphabetic, or a combination. V_1 and V_2 are the locations of the input or intermediate storage. R is the drum field to be searched. Note that with an equal channel search process, the full designation must be stored in the URA. With a complete address to place in the ADR and the use of "read", the designation need not be stored in the URA in order to locate the URA.

(2) The drum field being used during channel search is identified as "Channel Search Identifier". During the channel search process, the actual fields on the drum are being compared with the input and/or intermediate storage. Once a URA has been located and written on the revolver, the drum revolver fields are used. There are two types of indication: CSFO through CSF19, used only during channel search and FO through F19, used at all other times. If two fields in a record are to be searched, wire one channel search field from the R address hub and the other from the result shift hub.

(3) The process itself is wired into an equal channel search hub. There are 6 of these hubs on the connection panel.

(4) The exit of the equal channel search step is wired to an in hub of the CS equal branches (U-X) (13-16). If a URA whose identifier equals the V_1 or V_2 value is found, it is written on the revolver and the next step branch is obtained from the CS Equal Out hubs $V_1 =$ or $V_2 =$ (U-X) (14-15) respectively. If no URA is found in the channel whose identifier equals V_1 or V_2 then the last URA in the channel is written on the revolver and the next step exit is obtained from the equal CS ≠ hub (U-X) (16). This last exit will normally be wired to the channel overflow routine (see rule 11).

(5) V_1, V_2 and R may be shifted to give them the same decimal alignment, as in the compare process.

(6) The channel search process does not consider zeros and spaces as equals.

(7) An ignore code in V_1 or V_2 will not prevent comparison of that position with the corresponding character in R.

(a) V_1 & V_2	-	-	-	-	-	A	i	i	1	2	3	
ECS												V_1 & V_2 compare with R
R	-	-	-	-	-	A	i	i	1	2	3	

(b) V_1 & V_2	-	-	-	-	-	A	i	i	1	2	3	
ECS												V_1 & V_2 do not compare with R
R	-	-	-	-	-	A	B	C	1	2	3	

(8) A channel search is started at the URA designation in the ADR. It will end at the last URA in the channel. In a system with URA numbers from 00 to 19, if the number 13 were entered in the ADR in the URA positions, channel search would take place only on URA's 13 through 19. Therefore, to search a channel completely, 00 should be entered into the ADR in the URA position.

If significant digits might be present in the URA positions, ignore codes should be substituted for them before adding to a blank storage and placing in the ADR. Shifting of V_1 and R might also be used to accomplish this suppression.

(9) When a URA is written on the revolver, the address of the URA from which it was taken is remembered until a new address is entered in the ADR. However, it is not possible to take this address from the ADR and place it in storage, thereby learning the full address of the URA.

(10) If a complete drum address (drum section, channel, unit record area) is available, channel search is not used; the interstep function of "read" is used to place the information on the revolver instead. When channel search is used, "read" is not used.

(11) Channel Overflow...In many File-Computer systems, it will be impossible to develop a full six digit address (drum section, URA, channel) from the existing code number. However, in most cases it will be possible to develop a drum section and channel number. Channel number should be two digits with fairly even distribution; that is, as many 00's as 01's, as 02's, as 99's. This will minimize the chances of channel overflow.

Obviously, when using this technique, there will be cases where the number of items to be placed on a channel is greater than the number of URA's available on that channel. At the same time, there may be channels where the number of items to be placed is considerably less than the URA's on the channel. Through the programming technique of channel overflow, the excess items from one channel can be placed on a channel where space is available.

Assume a Univac File-Computer system with a URA of 24 digits. There are, therefore, 25 URA's in each channel, and each channel can contain a maximum of 25 items. Analysis of the digits of the item number which have been selected as drum section and channel reveals that in drum section 02, channel 76, there would be 30 items. However, in drum section 00, channel 55, there are only 16 items. All other channels are within the 25 item capacity of each channel.

Channel 0276-- has five items for which there is no space; channel 0055-- has room for nine additional items. The excess items from 0276-- will, therefore, be stored on 0055--. In order to do this, one URA from channel 0276-- must be used as an "overflow address"...that is, where to look if the item number is not found on channel 0276--. This means that only 24 items can be stored on 0276--, and the extra six will be stored on 0055--.

The URA in which the overflow address is stored is the last URA on the channel, in this case URA # 24.

The reason for placing the overflow address in the last URA on the channel is obvious when we realize that the equal channel search process automatically places the last URA in the channel on the revolver if a URA matching either V_1 or V_2 is not found during the search. In addition the equal channel search branch hubs notify us of the overflow condition by emitting the branch exit from the ≠ hub.

Any field of a URA may be used to store the overflow address. It should, of course, be at least four digits in length.

The principle of overflow programming is as follows. If the item number is not found during an equal channel search operation, the last URA on the channel is written on the revolver. This URA will contain the overflow address, if any. On the next step, the overflow address is placed in the ADR. The program is then routed back to the channel search step, and the new channel is searched. This type of program is known as a "loop"; the machine will search as many channels as necessary until the item is found, always going to the new channel whose address has been found on the channel just searched.

Assume the following:

Input storage 001:	Drum section and channel (0276--)
Input storage 002:	Complete item number (A502376)
Drum storage FO:	Complete item number or overflow address...(A604839, etc.) or (005500)

Program:

Step:	V_1	Pr	V_2	=	R	Next Step
1.	001 (0276--)	T			ADR (0276--)	2
2.	002 (A502376)	ECS	002 (A502376)		CSFO (A604839 etc.)	ECS Branch

V_1 =
(item no. is found)
To step 4

V_2 =
(item no. is found)
To step 4

≠
(item no. is not found)
To step 3

Step:	V_1	Pr	V_2	=	R	Next Step
3.	FO (005500)	T	------		ADR (005500)	2

On step 1, the drum section and channel are placed in the ADR. On step 2, the channel is searched for the item number; if it is found, the program continues with step 4. If it is not found, the last URA in the channel is written on the revolver. On step 3, the overflow address, located in field FO of the overflow URA is transferred to the ADR. The program reverts to step 2. As before, if the item is found on the new channel, the program continues at step 4. If it is not found, the new overflow address is transferred to the ADR and the loop is continued.

In some cases an incorrect item number is being searched for in a channel which does not contain overflow data in the last URA. Because of this, it is necessary to make a test of the last URA to determine if it is a legitimate overflow address URA, before transferring the overflow address to the ADR. This can be done in many ways; however, the most usual technique is to place the overflow address in some URA field other than the item number field. If this is done, then the item number field will be blank if it is an overflow URA. This field can then be tested for a blank to determine valid overflow URA's.

Unequal Channel Search

The unequal channel search process is the exact opposite of the equal channel search operation in that it is used to locate all unit record areas which are unlike V_1 (Note: V_2 is never wired on the unequal channel search process) rather than URA's which are like V_1 as is the case in the equal channel search process. The programming symbol for this process is UCS.

In principle the unequal channel search allows us to locate any URA within a channel whose identifier is not equal to the V_1 value. If all the URA's within the channel have identifiers which are equal to the V_1 value, then the last URA within the channel is captured on the revolver.

In addition to being able to locate URA's whose identifier field does not equal V_1 we can also locate URA's in which all fields do not equal'V_1 (only in URA's which are divisible by 12). This is done by wiring the result of the step to the CS ALL hub (k-58).

The primary use of the unequal channel search process and its related branching function is in clearing or unloading drums, for it allows us to locate all those URA's which are not equal to a given storage (V_1). If we wire V_1 to a blank storage then this process will locate, within the channel specified by the ADR, the URA's whose identifier field or all fields are not blank and write the URA which is not blank on the revolver. An example will serve to illustrate the use of this process.

Assume that we wish to locate all URA's, within the system, which are not blank in FO and clear them so that they are blank. Further assume that by either an input entry for each channel, containing the DS and CH number of every channel in the system or by a progressive address total we can obtain the drum section and channel number from 000. The values dealt with are:

000 =	-	-	-	-	-	0	1	9	6	0	0	+	
						DS		CH		URA			(URA # always equals 00)
19 =	-	-	-	-	-	-	-	-	-	-	-	+	
CSFO=	-	-	-	-	-	X	X	X	X	X	X	+	Item no. or blank (space)

The program for this operation would be:

	V_1	PR	V_2	R	To
1.	000 (019600)	T	-	ADR (019600)	2
2.	19 (--------)	UCS	-	CSFO (--XXXXXX)	UCS Branch

$V_1 \neq$	$V_1 =$
clear all revolver fields write to step 2	take next address and then back to step 1.

In step number 1 the address, (obtained from a card or from a computer progressive total), is transferred to the ADR and the program proceeds to step number 2. Step 2 compares FO of every URA in the channel against a blank storage (19) for inequality. If a URA whose FO is not blank is found, then that URA is written on the revolver, and the exit of the step, being wired to the UCS branch, emits from the UCS $V_1 \neq$ hub. This exit would then be guided to a series of steps which clear the revolver fields with spaces (19) and then writes the revolver fields back to the drum URA, clearing the URA which caused the inequality signal. The routine is then directed back to step 2 in order to capture any more URA's in the channel which contain data in FO. Eventually all URA's within the channel will be cleared and then, since all FO's in the channel are blank (or equal to V_1), the exit of the step will emit from the UCS $V_1 =$ hubs, thus instructing us to place a new address in the ADR by either feeding the next address card or increasing the address progressive total by 1 in the 3rd position.

The rules governing channel search-unequal are in general the same as those given for the equal channel search process except for the points mentioned above.

5. EXIT OR OUT OF A STEP

The exit, or out of a step, is the means whereby the individual steps are tied together to form a program. The step out hubs are located at line 12, A-x. When the process called for has been completed, the result entered into storage, and the step completely checked (if "no-check" is not wired), a pulse will be emitted at the exit hub. This pulse can be wired to the entry of the following functions:

1. The next step of the program
2. Any interstep function
3. Demand assignment
4. Output function transfer line

When the Univac File-Computer, through its self-checking features, detects certain types of errors, the exit of the step in which the error was detected is not emitted at the out hub of the step, but rather at a hub indicating the type of error. (See B, following)

A discussion of the places to which an out of a step may be wired, as well as the error hubs, follows:

A. WIRING FROM "OUT" OF A STEP

1. The next step of the program

In many cases, the "out" of a step will be wired to the "in" of the step immediately beneath it on the plugboard. The step in hubs are located at line 11, A-x. That is, the out of step 3 may be wired to the in of step 4. However, the Univac File-Computer is completely non-sequential in operation. This means that the out of a step can be wired to the in of any of the other 47 steps. A program could be set up to go from step 1 to 5 to 3 to 4 to 10 to 6 to 48, etc. It would even be possible to start at step 48 and work backwards to step 1.

As discussed under channel search, it is possible to set up a "loop" of steps; that is to go back through a series of steps until a certain desired result is obtained. Note that in all cases where a loop is set up, some means must be found of getting out of the loop, for example, the branching of a channel search or +, -, or 0 branching. If this is not done, the program will never continue past the loop. Another example of a loop is the calculation of square root, which is done by approximating the square root, testing, and making another approximation until the root is found. To accomplish this, the same series of steps is used several times. If step numbers 3 through 6 were used 5 times, this would be the equivalent of 20 steps; yet only 4 are wired on the plugboard.

The exits of several different steps can be wired to the entry of the same step. If one program step is common to several different routines, the exit of the last step of each routine could be wired to the entry of the common step, saving the necessity of repeating it for each routine.

2. An interstep function

An interstep function is an operation that occurs between program steps. It has an in and an out in the same way that a program step does. The in is wired from the out of a program step or another interstep function, and the out is wired to the entry of any of the functions to which a step out is wired. A pulse will be emitted from the out hub when the function has been completed. (See interstep functions for a more complete discussion.)

The interstep functions are:

a. Step clear
b. Read unit record
c. Write unit record
d. Branching
e. Program selector
f. Channel search branching

3. Demand assignment

The out of a step may be wired to one of two types of hubs in the demand assignment section..."test" and "in".

If it is wired to a test hub, (line 3-4, B-I) the corresponding "yes" ready hub (line 6, B-I) will emit if the associated demand input-output device is fully loaded. If the corresponding demand unit is not fully loaded, the corresponding "no" ready hub (line 5, B-I) will emit.

If the exit of a step is wired to "in" of demand (line 7, B-I) it will cause the multiplex function to shift to the unit demanded. When the connection has been completed, one of the two out hubs will emit. If the last operation by this demand unit was to transfer information into the output medium the "output" hub (line 9, B-I) will emit. If the last operation by this demand unit was to transfer information from the input medium into storage the "input" hub (line 8, B-I) will emit.

4. Output control lines

The out of a step may be wired to any one of the 10 output control lines (line 2-7, e-n) which are numbered A - J. There are 6 hubs for each line, and wiring may be to any of the 6 without danger of a "backfeed". Each of the 10 lines may perform a special function at the input-output device to which the multiplex function is connected at the moment. For example, function A, when connected to the card sensing-punching unit may sort a card; when connected to a tape unit, function A may feed tape. Function B on the card sensing-punching unit might trip a card, and so forth.

B. ERROR HUBS

For the following reasons, the Univac File-Computer will signal an error condition:

1. Invalid address
2. Parity check
3. Arithmetic error
4. Addition or subtraction overflow
5. Division overflow

When one of these conditions exists, the following will occur:

(1) The step "out" will not emit a pulse. Instead, the corresponding error hub will be energized. This hub may be wired to program selectors, output transfer lines (to operate a function in an input-output device), and similar functions. However, it should not be used to start another program step.

(2) The computer will hang up at the point in the program in which the error was detected. Before another program step can be started, this step must be cleared. This is done by wiring to "step clear". It may be done by wiring directly from the error hub to step clear, or by wiring first to a program selector or similar function and then to step clear. Since there are two hubs for each error, one hub may be wired to step clear while the other is wired to a program selector or other function.

A more detailed explanation of each of the error hubs follows:

1. Invalid address (line 3-4, a)

Invalid addresses include:

a. Alphabetic information (including ignore codes)
b. Drum section and/or URA numbers larger than the maximum range of the system. For example, in a one drum system, any drum section number of 03 or more would be invalid.

If an invalid address is entered into the ADR and a read, write, or channel search operation is called for, the instruction will not be performed. No data will be read out or altered. The invalid address error signal will emit on the first following step with a value or result wired to one of the revolver fields, F0 through F19.

2. Parity error (line 7-8, X)

The Univac File-Computer checks each character as it is transmitted into, or out of a register, to be certain that it has not changed during transmission. If it does not check, the step will not be completed, and the two "parity" error hubs will emit a pulse.

3. Arithmetic error (line 3-4, X)

The File-Computer will automatically check each process. If a process does not prove the "arith" error hubs will be energized. One of these hubs can be wired to "step repeat", causing the machine to repeat the step until the correct answer is computed, provided the error is a computer error and not a program error (i.e. R shift in multiplication when check is made).

The machine hangs up on the step on which the error was made until a correct answer is computed by means of step repeat (if it was wired). In addition and subtraction, no result is delivered to storage until a correct answer has been computed. In multiplication and division, an incorrect answer could be placed in storage, though it will be replaced by the correct result computed on the step repeat.

If the "no-check" hub is wired, the machine will not verify the results of a multiplication or division.

4. Addition or subtraction overflow (line 5-6, X)

This error signal will occur when a carry takes place into the 12th position.

5. Division overflow (line 5-6, a)

If the left shifts wired on a division step would develop more than 11 quotient digits, the division process is not begun; and the computer hangs up. This occurs whether or not step repeat is wired.

INTERSTEP FUNCTIONS

An interstep function is a function which occurs between steps of a program. For each function used, there is an "in" hub, which is usually wired from the out of a step or another interstep function. When the operation has been completed, the corresponding out hub will emit. This can be wired to another interstep function or to a program step.

The interstep functions are:

a. Step clear
b. Read unit record
c. Write unit record
d. Branching
e. Program select
f. Channel search branching

Each of these is considered in detail.

A. STEP CLEAR (line 3-8, b)

Step clear is used to clear the step on which the File-Computer hangs up because one of the errors listed in Section 5-B has occurred. When one of these errors occurs the program cannot be continued until the step has been cleared by means of step clear.

The "in" hub is wired from the error hub. The "out" hub may be wired to another step, a program select, or any other place to which a step out is wired.

Note that when a pulse is emitted from step clear, there is no indication as to which step has just been cleared. When an error occurs, it is probably advisable not to continue the program, but rather to stop the computer. The cause of the error can then be determined, and the program continued.

B. READ UNIT RECORD (line 13-14, D-G and 13-14, e-h)

This interstep function is used to read a specific unit record area on the large capacity drum, and write that URA in every unit record area on the revolver. In order to use this function, a complete address (drum section, unit record area, and channel) must have previously been placed in the address register.

Note: Switching of the address register to the specified drum section and channel is initiated as soon as the address in the address register is transferred to the staticizer. This transfer takes place automatically when the address is delivered to the address register, unless a prior read or write unit record or channel search operation is in process. In that case, the automatic transfer is delayed until the prior instruction is completed---then the transfer takes place and the switching immediately commences. It is possible, therefore, to "overlap" other instructions with the switching time. If the switching is not completed when a read, write or channel search instruction is received, the instruction will be "remembered" until switching is completed and then will be initiated.

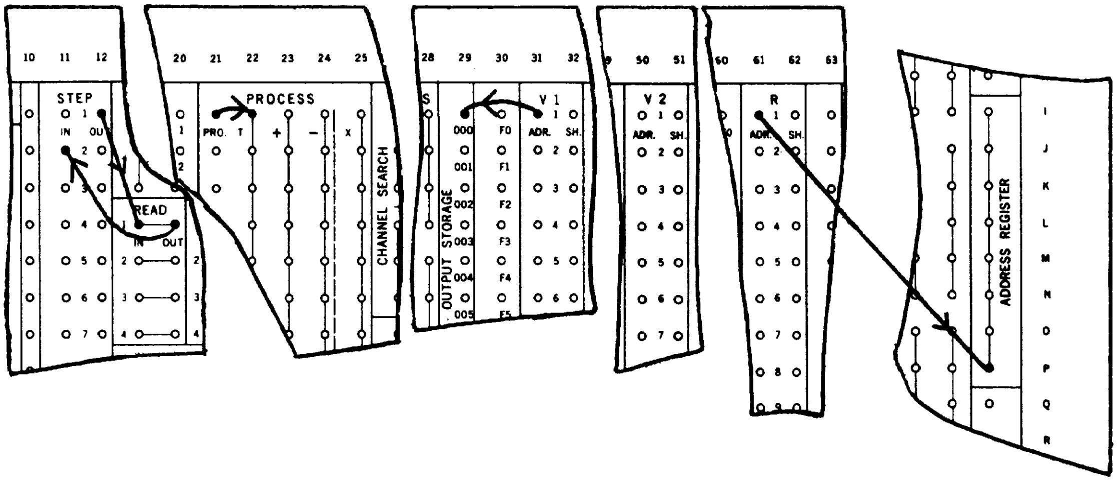
The "out" hub of a read function emits a pulse immediately following the receipt of an "in" pulse. Calculations not involving a revolver field can take place simultaneously with reading. If a revolver field is called upon before the URA is written on the revolver (and, therefore, available for use), an automatic interlock will delay the program until the read function has been completed.

There are 8 read functions available, indicated as R1 through R8. This makes it possible to go to 8 different places from a read function without the use of selectors.

Assuming that a complete address is placed in the address register from input storage 000 on step 1...the programming would be:

Step	V_1	Pr	V_2	=	R	Next Step	R1	From	To
1	000	T			ADR	R1		Step 1	Step 2

The connection panel wiring for this step would be:



Note that both channel search and read accomplish the same purpose...the writing of a URA on the revolver so that the fields in the URA are available for use. Unless one of the two operations is used, it is impossible to call on the storage fields of the large capacity drum. Read is used when a complete address is available, channel search when an incomplete address is to be used. Note also that a read instruction does not remove the fields from the large capacity drum; the URA remains on the drum unchanged until a write instruction is given.

C. WRITE UNIT RECORD (line 15-16, D-G and 15-16, e-h)

This interstep function is used to read the fields in the revolver and write them in the URA whose address is in the ADR. Whenever one of the large capacity drum fields (FO through F19) has been changed during a program, a write instruction should be given to the File-Computer so that the changed information can be placed in its URA. During the program, changes are made only on the revolver; therefore, unless a write instruction is given to the File-Computer, the new information will not be placed on the large capacity drum. If no change has been made, as for example, in referring to a list of rates, a write instruction is unnecessary.

When the "in" of a write instruction receives a pulse, the fields on the revolver will be written in the URA specified by the address in the staticizer, usually the URA which was read last. If two URA's are read before a write instruction is given to the File-Computer, the write instruction will always place the information in the last URA which was read. If, at the time the "in" of a write instruction is impulsed, a channel search operation had been performed and no new address transferred to the ADR, the data from the revolver will be written in the URA located by channel search.

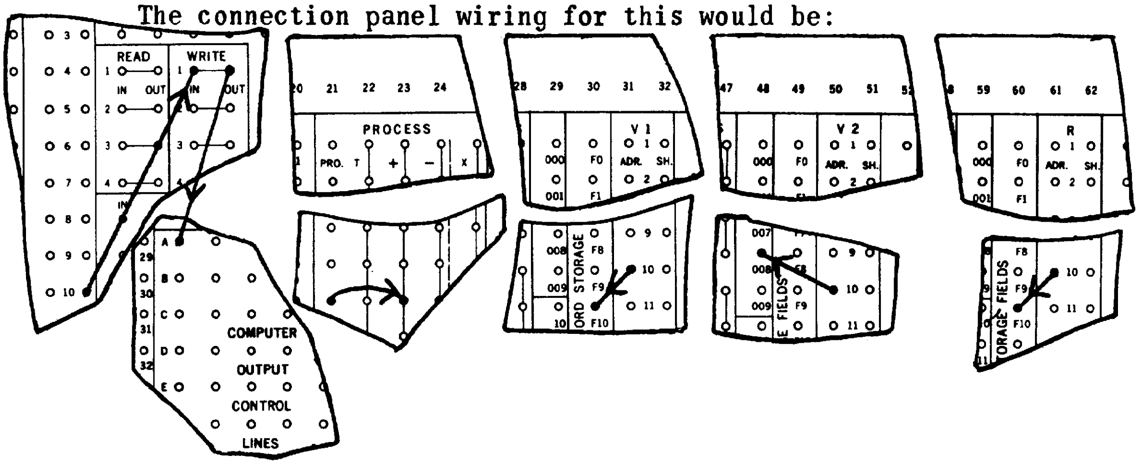
The "out" of a write function emits a pulse immediately following the receipt of an "in" pulse. Calculations not involving a revolver field can take place simultaneously with writing. If a drum field is called upon before writing is complete, an automatic interlock will delay the step until the completion of the writing operation.

There are 8 write instructions available, indicated as Wl through W8. This makes it possible to go to 8 different places from a write instruction without the use of selectors. If it is desired to go into a write instruction from several different steps because of varying routines, yet always go to the same place following the write instruction, only one write need be used. The "in" would be wired from all the steps desired, while the out would go to the one common destination. It would be equally correct to wire each step to a separate write instruction.

Assume that the last step of a problem is to accumulate sales-to-date, then write the information from the revolver back on the drum, and finally inform the output device that the program has been completed. With the detail sales information in input storage 008, and sales-to-date in drum field F10, the program step might be:

Step	V_1	Pr	V_2	=	R	Next Step	W4	From	To
10	F10	+	008		F10	W4		Step 10	Transfer line A

The connection panel wiring for this would be:



There is no relationship between the read hubs and the write hubs. In other words, when R1 is used, it does not matter whether W1 or W8 is used for writing the information back on the drum.

Whether read or channel search is used, write is the only method of placing the information back on the drum. A new address may be transferred to the ADR while a read, write or channel search operation is in process... its automatic transfer to the staticizer will be delayed until the completion of the previous instruction.

D. BRANCHING (line 13-16, H-P and 13-16, i-q)

Branching is the means of determining the next operation, depending on whether the result of a step is +, -, or 0. It is also used in connection with the compare process to determine whether V_1 is larger than, equal to, or less than V_2 (see "compare").

For example, one step in a payroll problem might be to subtract an employee's non-taxable income from his gross pay. If the remainder is plus, it should be multiplied by the tax rate; if it is minus, no tax is to be deducted. Note that compare is not the best process to use, since the result itself may be used in further computation, and there is no result with a compare process.

Assume the following:

Input storage	007:	Gross pay
Intermediate storage	12:	Non-taxable income
Intermediate storage	20:	18% (Tax rate)

Place the taxable income in storage 10, withholding tax in 004.

The steps to accomplish this might be:

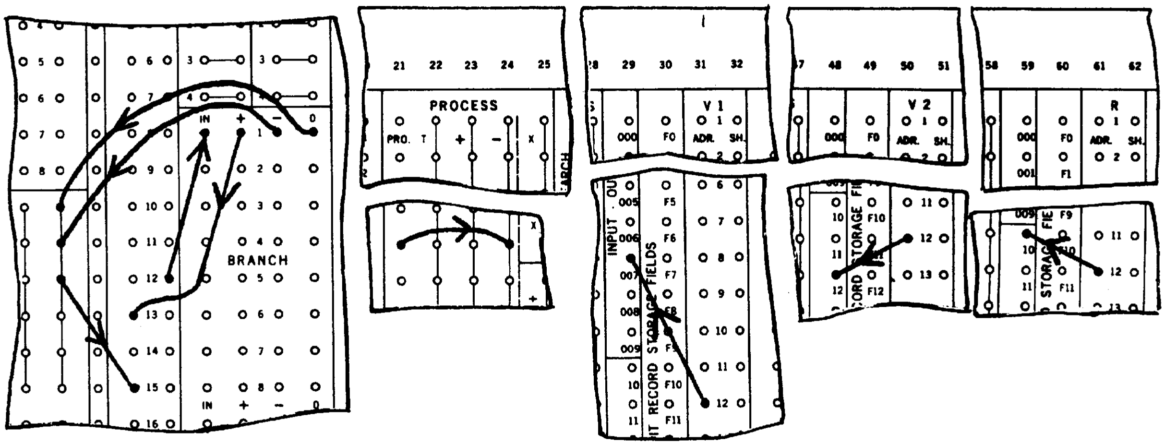
Step	V_1	Pr	V_2	=	R	Next Step	From	+	-	0
12	007	-	12		10	BR1	12	13	15	15
	(gross Pay)		(non-tax inc)		(tax inc)					
13.	10	X	20		004	14				
	(tax inc)		(.18)		(with.tax)					

14. Rounding step

15. Continue program

Note that the zero hub is wired. If the result should happen to be zero, and the zero hub were not wired, the machine would hang up... in this case, if the answer were zero, the same steps would be skipped, as if the answer were minus.

The connection panel wiring for step 12 would be:



There are 18 branches, referred to in programming as BR1 through BR18. This means that 18 determinations of + -, or 0 (including comparisons) can be made without the use of selectors.

E. <u>SELECTORS</u>

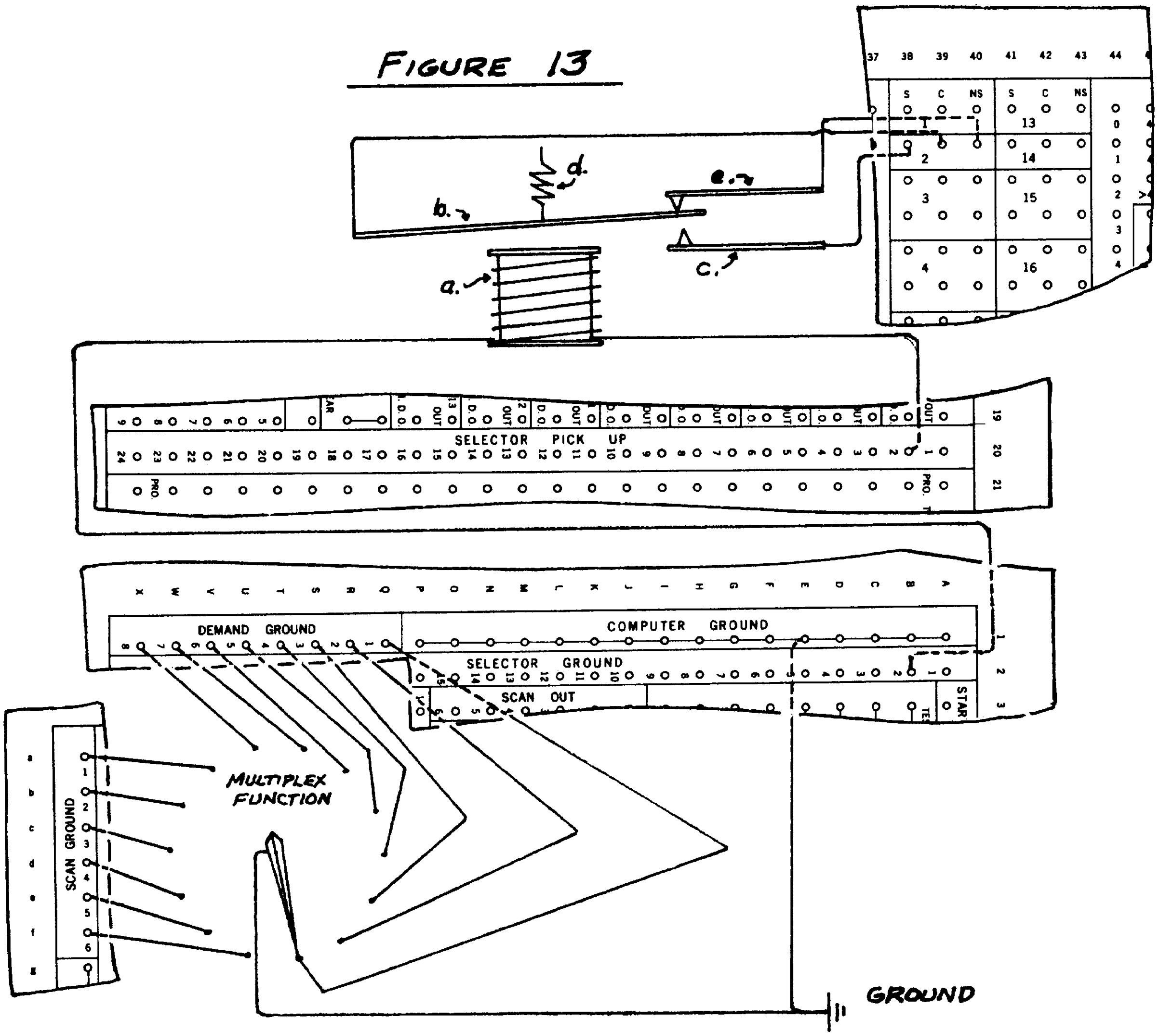
(1) Operation of selectors

Although a selector itself is not an interstep function, it is important to explain what a selector is before explaining the interstep function of Program Select.

A selector is a two-way, electrically operated switch which allows the programmer to route machine functions in one of two directions, based upon the presence or absence of control positions, negative results, or any other controllable machine function.

Basically, a selector functions as a relay and as such is shown in Fig. 13.

FIGURE 13



The iron core (a) around which the wire is coiled becomes magnetized when current is placed in the pickup hub, and the ground hub is wired to either computer ground, demand ground or scan ground. Thus, when current enters the pickup, the metal bar (b) over the iron core is drawn down to make contact with the "select" bar (c), and any current entering into the common hub will emit or come out of the select hub. Conversely, if current does not enter the pickup hub, the iron core will not be magnetized and the spring (d) will keep the common bar (b) in contact with non-select bar (e); and any current entering the common hub will be received at the non-select hub.

In order to complete the circuit, and, therefore, make the selector effective, it is necessary to wire the ground hub to one of three places: computer ground, demand or scan ground. Demand or scan ground is used when the selector has reference to a particular input-output device. There are 8 hubs labeled "demand ground", each of which belongs to its corresponding demand input-output device. There are 6 hubs labeled "scan ground". Each of these is related to the 6 corresponding scan outs. (See following section V-8 for discussion of multiplex control).

When the ground of a selector is wired to a demand ground hub, the circuit will be completed only when that particular demand input-output device is connected to the computer through the multiplex function. For example, suppose that selector 1 is to be energized whenever a minus condition occurs on step 29 if demand input-output device 3 is connected to the computer. The ground of selector 1 would be wired to demand ground 3. No matter how many times step 29 has a minus result, the selector would be energized only when demand input-output device 3 is connected to the computer.

Scan ground operates in the same fashion. It is effective only when a scan input-output device, related to the specific scan out, is connected to the system through the multiplex function.

If a selector is to operate regardless of the input-output device connection to the computer, the ground of the selector is wired to "computer ground". When this is done, whenever a pulse enters the pickup, the selector will be energized.

The File-Computer has 48 selectors (lines 38-43, A-z). These are of two types...single pole and multi-pole. The difference is that when the pickup of a multi-pole selector is energized, more than one common bar (b) is affected. Thus, more than one choice may be made based upon the condition causing the selector to be energized. The 48 selectors are divided as follows:

16 single-pole selectors

24 two-pole selectors

8 four-pole selectors

A selector may be diagrammed as follows:

	PU	GD	CG	DG	SCAN	S	C	NS
Selector 1	o	o	o	o	o	o	o	o

PU = Pickup - wired from the pulse causing the selector to become select. (line 20, A-z)

GD = Ground - wired to either computer ground, demand or scan ground. (line 2, A-z)

CG = Computer ground. (line 1, A-P & 1, g-z)

DG = Demand ground. (line 1, Q-X)

SCAN = Scan ground. (line 1, a-f)

S = Select - wired to hub desired when selector is on the select side. (line 38, A-z and 41, A-z)

C = Common - wired from hub which is to change, dependent upon the position of the selector. (line 39, A-z and 42, A-z)

NS = Non-select - wired to hub desired when selector has not been changed to the select side. (line 40, A-z and 43A-z)

2. Use of selectors

Selectors may be used to alter any factor, or combination of factors, in a program: value 1, process, value 2, result, next step, etc. There are two basic ways in which a selector operates:

a. Altering a program dependent upon a condition recognized at the input device.

b. Altering a program dependent upon a condition recognized by the computer as a result of its calculations.

In the first case, the exception is pre-coded and placed in the input device; in the second case, it is recognized that the exception might occur as a result of the program, but it is impossible to pre-code the input media causing the exception. An example of each type follows:

(a) Use of selector precoded at input device.

Assume a payroll problem with the following information in input storage units:

Storage	Description
000	Day rate
001	Hours

wired through input transfer lines

1 - 1st shift
2 - 2nd shift } Shift code
3 - 3rd shift

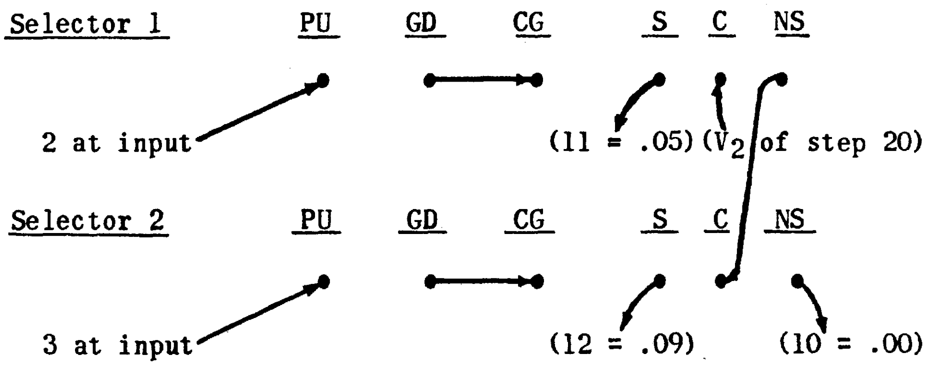
Shift	Shift rate	Stored as constant in storage
1	.00	10
2	.05	11
3	.09	12

By using selectors, only one step would be required to add shift rate to day rate to arrive at total pay rate. The step would be:

Step	V_1	Pr	V_2	=	R
20.	000 (day rate)	+	10, 11 or 12 (shift rate)		20 (total rate)

In this case, value 2 is to be put through selectors to determine whether to add the constant from storage 10, 11, or 12 to the day rate.

Assume that selector 1 will be on the select side for 2nd shift employees and that selector 2 will be on the select side for 3rd shift employees. If neither selector 1 nor selector 2 is select, the employee must be on the 1st shift. This may be diagrammed as follows:



V_2 of step 20, which is to be selected, is wired to the common of selector 1. If selector 1 is on the select side, the employee is 2nd shift. Storage 11 will be called upon, and .05 will be added to the day rate. If selector 1 is not on the select side, the employee is either 3rd shift or 1st shift. The wiring is, therefore, to the common of selector 2 to determine his shift. If selector 2 is on the select side, the man is 3rd shift. Storage 12 is called upon and .09 will be added to the day rate. If a man is neither 2nd shift nor 3rd shift, neither selector will be on the select side, and the employee must be 1st shift. On the non-select side of selector 2, storage 10 will be called upon, and .00 will be added to the man's day rate.

(b) Use of selector determined by computer.

Assume a problem where the entire 48 step capacity of the machine has been used, but one more step is needed when there is a minus result on step 48. The problem is to reuse one of the existing steps to accomplish the additional operation. The extra step appears as:

V_1	Pr	V_2	=	R	To
002	X	003		004	OPC A

In order to make the most economical use of selectors, search through the program for another similar step. Assume the step most similar is:

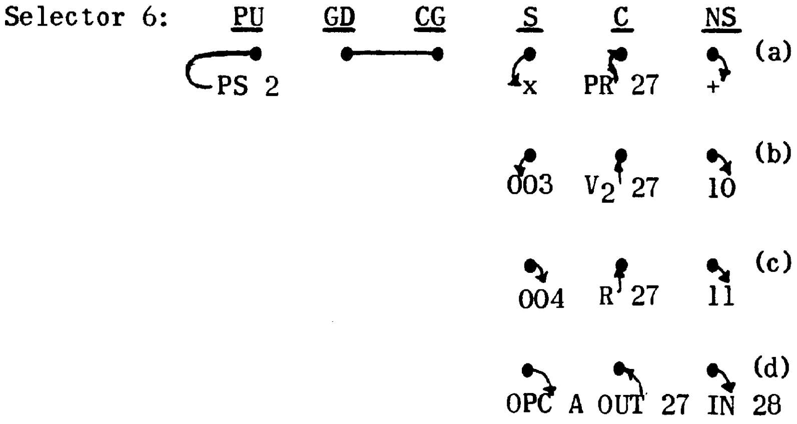
STEP 27:

V_1	Pr	V_2	=	R	To
002	+	10		11	28

As can be seen by comparing the two steps, the problem is to go through step 27 once and do the operation once; however, when there is a minus result on step 48, repeat step 27 changing the process from plus to multiply, V_2 from 10 to 003, result from 11 to 004, and step exit from 28 to output control line A. In order to do this, the machine must be told that it has been through step 27 once, and that the values should now be changed. The means of doing this is a program select (explained more fully below). A program select is an interstep function which changes the selectors associated with it from the non-select side to the select side, and keeps them on the select side for a predetermined length of time.

In this case, a 4-pole selector should be used, since there are four choices to be made. Assume that immediately upon a minus branching of step 48, the program is wired to program select 2, which is associated with selector 6, a 4-pole selector. (Note that a selector remains on the select side only for the duration of the pulse at the pickup; the minus pulse itself could not be used as a pickup since it would not last into the next step, and the selector would immediately revert to the non-select side.)

A diagram of selector 6 would be:



When selector 6 is on the non-select side, it means that we have not gone through step 48; therefore, we must be using step 27 for the first time. However, if selector 6 is on the select side, it means that we have gone through step 48, have received a minus branch, and are using step 27 for the second time. The first time through step 27, the process will be + , V_2 will be storage 10, R will be storage 11, and out of step 27 will be step 28. The second time through, the process will be X, V_2 will be 003, R will be 004, and out of step 27 will be to output control line A. It was not necessary to put V_1 through the selector since the storage location will be 002 in both cases.

A brief summary of the points to remember about selectors:

1. A selector is merely a switch to redirect a pulse; it does not emit a pulse of its own.

2. A selector will remain on the select side only as long as a pulse is entering its pickup; when that pulse stops, the selector will return to the non-select side.

3. Any function of the machine can be changed through use of selectors.

4. If only one factor is to be changed because of a certain condition, use a single pole selector; if more than one is to be changed, use a multi-pole selector. The same pulse can be used to operate several selectors if necessary.

5. The factor which is to be changed is wired to the common. The variables are wired to the select and non-select sides.

F. PROGRAM SELECTS (lines 18-19, A-p)

Program selects are used to pick up selectors at a specific time, or upon detection of a certain condition during the program. It is a means of lengthening or "remembering" a pulse. In the second example of the use of selectors, the minus sign of step 48 was a momentary pulse which was no longer in existence during the time that selector 6 should be on the select side; by wiring to a program select, a pulse was created which would last the required length of time.

There are 16 program selects provided, referred in programming as PS1 through PS16. Each program select has 4 hubs:

1. In - When a pulse is received at the "in" hub, no matter how long in duration, the out (power) hub begins to emit current. The "in" hub is usually wired from the out of a step, the +, -, or 0 of branching, or some other similar function. It resembles the "in" of a step.

2. Delayed out or immediate out - 10 of the program selects are provided with delayed out hubs, PS 1 through PS 10, and 6 with immediate out, PS 11 through PS 16. The out hub is the means of continuing the program, resembling the out of a step. A delayed out hub will not emit until the selector or selectors associated with the program select have changed from the non-select to the select side, a process requiring 15 milliseconds. A delayed out hub should be used when the associated selector is to be used immediately. An immediate out hub will emit immediately following the pulsing of "in". This may be used when the selector is not to have an effect until at least 15 milliseconds have passed (check step process times). Unless the "in" hub of the program select is split wired or bussed, the delayed or immediate out hub must be wired. The delayed or immediate out hub is wired to the in of a program step, interstep function, or any other place to which the out of a step is wired.

   The delayed out of program selects 2 through 10 will emit 15 milliseconds after the related "in" has received an impulse. The delayed out of program select 1 will emit 20 milliseconds after its "in" has received an impulse. Selectors to be picked-up from an input code require 20 milliseconds to be changed from the non-select to the select side.

If calculations or operations, selected through the selectors picked up by the input codes, are to be made early in the program, this program select, in conjunction with a function delay, can be used to insure that 20 milliseconds have elapsed before the selectors picked-up by the input codes are used. (See function delay, part 7-C following.)

Note: It is not necessary to wire the delayed or immediate outs of program selects. The pulse which impulses the "in" of the program select can be bussed to the next operation desired. (See out expanders 7-B, following for illustration.)

3. Out (Power) - The out (power) hub of a program select is the hub that emits a steady pulse to hold a selector on the select side. As soon as a pulse goes into the "in" hub, the power hub will start to emit. It requires 15 milliseconds for the associated selector to change to the select side: the selector will remain select until a pulse is received by the drop-out hub (note that this hub bears no relationship to the immediate or delayed out). The power hub is wired to the pickup of the selector or selectors to be associated with the program select.

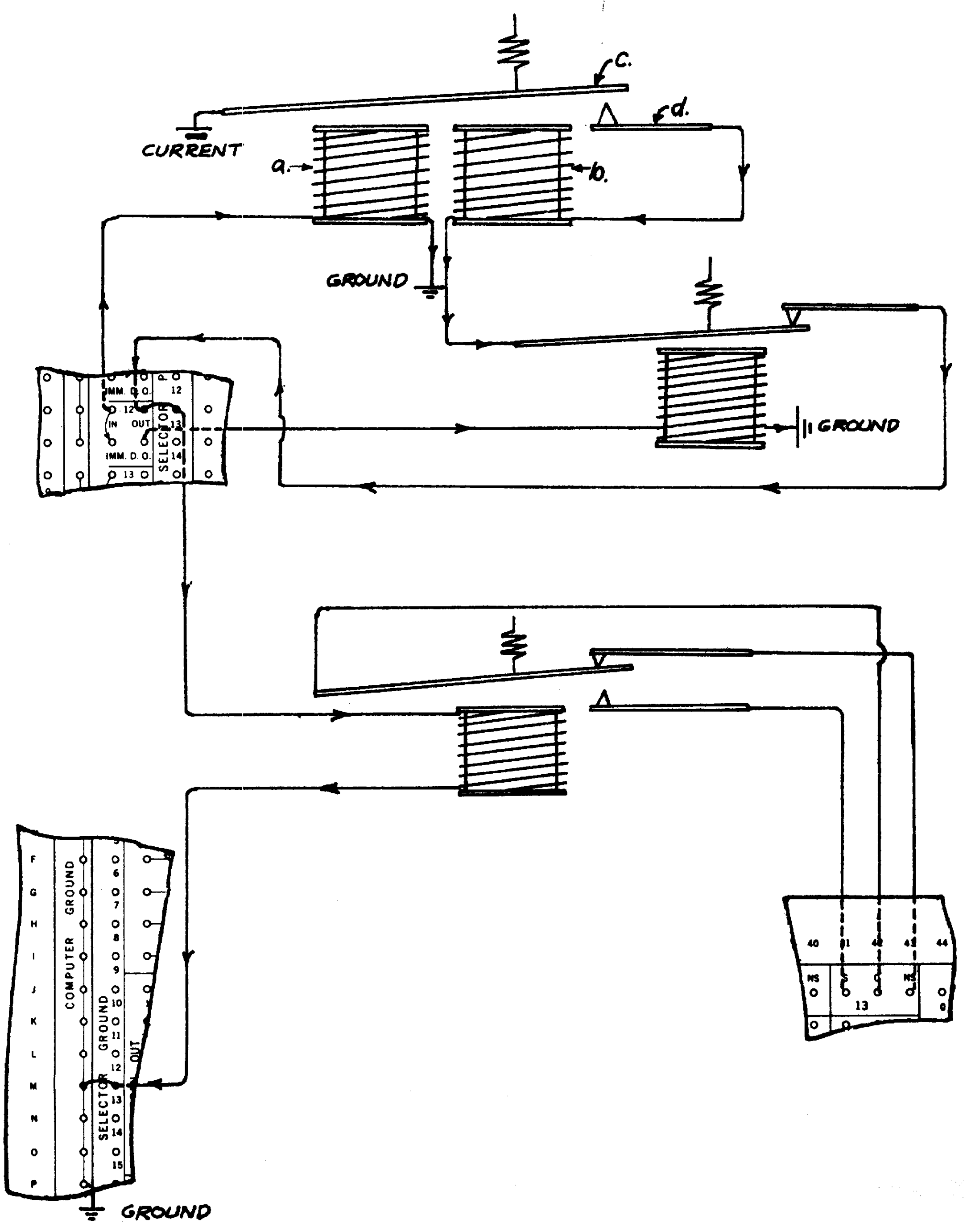
4. Drop out - The purpose of the drop-out hub is to end the pulse going to the selector, and thereby allow the selector to return to the nonselect side. As in the case of changing a selector to the select side, it requires 15 milliseconds to drop out the selector. The pulse used to do this may be any pulse, though frequently it will be the out of a step or some similar function.

No "out" is provided for the drop-out of a program select. If the out of a step or an interstep function is used to drop out the program select, some means must be found to continue the program. This is usually done by bussing the drop-out pulse through an out-expander or normal bus hubs.

No automatic drop-out is provided for program selects. However, it is possible to drop out all program selects which are emitting by wiring to "clear". When a pulse is wired to "clear", it has the same effect as wiring to the "drop-out" of all active program selects. This would most frequently be done at the end of a program, so that all selectors would be on the non-select side at the start of a new program. "Clear" (line 18-19, Q-R and 18-19, q-r) has no "out" hub, so it is usually bussed in the same way as the drop-out of a program select.

Figure 14 is a schematic diagram of the operation of a program select. Note that electronic tubes, not relays, actually accomplished this function: relays are used for ease of illustration only.

FIGURE 14
SCHEMATIC of PROGRAM SELECT AND SELECTOR OPERATION



When relay "a" is energized, bar "c" comes in contact with bar "d", allowing current to reach the selector pick-up. This keeps relay "b" energized, and holds the selector on the select side until a drop-out pulse breaks the connection. Once this is done, bar "c" will not come into contact with "d" again until a pulse is received at the "in".

G. <u>SELECTOR HOLD (B+)</u> line 9, e-h

Selector hold is a constant source of power which never stops emitting as long as the machine is on. It may be used to keep a selector on the select side even though the pulse which was entering the pick-up has ceased. The two most important uses would be:

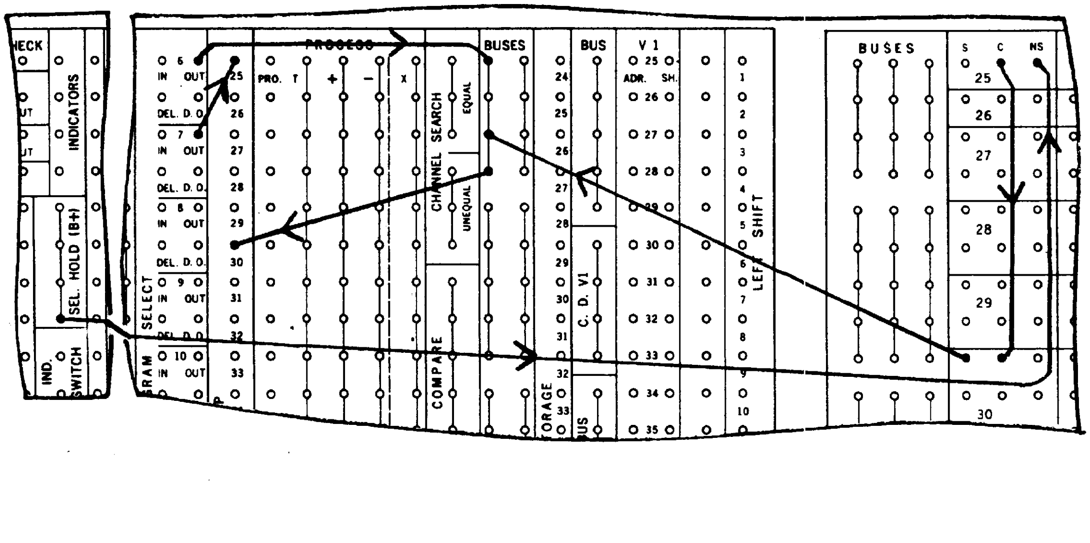
1. To "remember" a control from an input device, even though the control is no longer present.

2. To keep selectors associated with a program select on the select side even though a clear signal, which would normally drop out the program select, is given.

Selector hold has no drop out of its own. Once a selector is picked up through a selector hold connection, it will remain on the select side until the machine is turned off, unless some means is found of breaking the connection. This is usually done through another selector, as well as one pole of the selector which is to be controlled.

Assume that selector 30, a 4-pole selector, is to be held on the select side even though a clear signal is given. Pole "a" will be used to control the selector hold. Selector 25, a single pole selector, will also be used as a control of selector hold. Poles b, c, and d of selector 30 may be used for their normal purpose of re-routing functions.

The connection panel wiring for this would be:



Until selector 30 is picked up by PS 6, selector hold power cannot flow to the pick-up of selector 30. It can only go from the non-select side of selector 25 through the common of 25 to the common of selector 30a. With selector 30 on the non-select side, the current can go no further. However, as soon as PS 6 changes selector 30 to the select side, selector hold power will flow through the select side of 30a to the PU of selector 30. Then, even though PS 6 is dropped out by a clear pulse, selector hold power will keep the selector on the select side. It will remain select until PS 7 is impulsed and picks up selector 25. Since selector hold power will no longer be connected to the common of selector 25 and cannot reach the pick-up of selector 30, selector 30 will revert to the non-select side. It will remain there until PS 6 pulses the pick-up again. Note that PS 7, which is controlling selector 25, must be dropped out before selector hold power can be used again to hold selector 30.

H. Channel Search Branching (See discussion of the channel search processes, section V)

When either of the channel search processes are used, the "out" of the related program step is wired to "in" of the channel search equal or unequal branching, as dictated by the function used (lines 13-16, U-X and 13-15, a-d). If the out of the step is wired to a channel search equal "in", the related $=V_1$ or $=V_2$ or ≠ hub will emit, depending on the comparison received from the channel search. If the out of the step is wired to a channel search unequal "in", the related = or ≠ hub will emit, depending on the comparison.

7. OTHER FUNCTIONS

This section explains various features of the Univac File-Computer which have not been covered in other sections. These include:

1. Alternate switches
2. Out expanders
3. Function delays
4. Function sequence
5. Code distributor
6. Input control lines
7. Output control lines
8. Indicator lights
9. Unit record field assignment selector pick-up

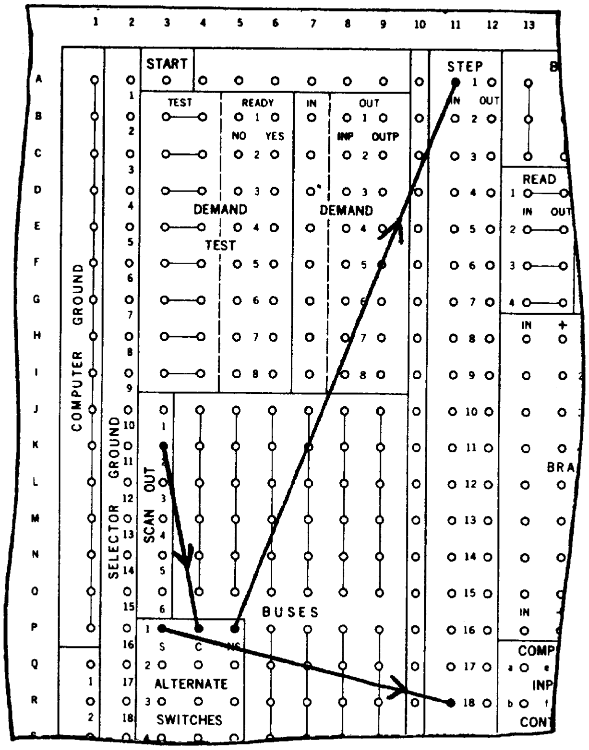
A. ALTERNATE SWITCHES (line 3-5, P-U)

An alternate switch is basically a type of selector under the control of the operator of the machine. Six of these switches are provided. Externally mounted on the control cabinet are six switches, each with an open and closed position. On the connection panel are three hubs for each switch...select, common, and non-select.

The function which is to be changed is wired into the common of the switch, the variables to the select and non-select sides, just as in a selector. The main difference between a selector and an alternate switch is that the operator manually determines whether the switch is to be select or non-select; therefore, the setting is constant for a particular program.

If there are two programs on one board which are not being handled simultaneously, an alternate switch could be used to determine which program is to be used at a particular time. Assume that one program starts on step 1, the other on step 18. The input device demand or scan out would be wired to the common of the alternate switch, step 1 "in" could be wired to the non-select side, step 18 "in" to the select side (step 1 could have been wired to the select side and step 18 to the non-select side). The step on which a program will start is, therefore, dependent on the setting which the operator makes of the alternate switch.

The connection panel wiring for this would be:



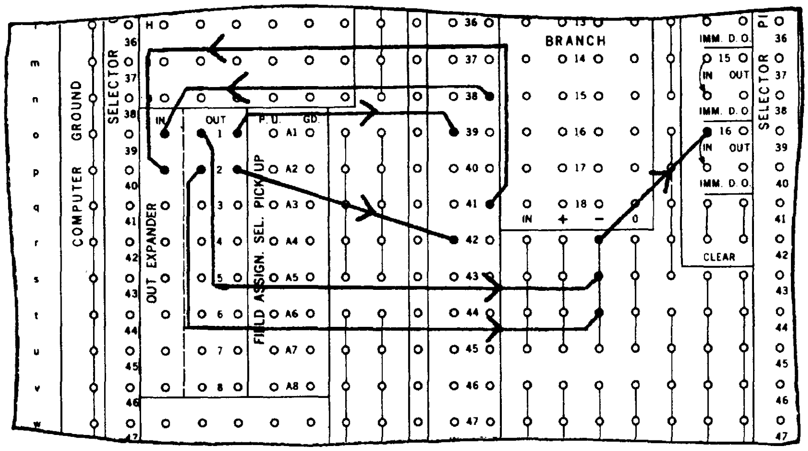
B. <u>OUT EXPANDERS</u> (lines 3-5, o-v)

There are 8 out expanders on the computer plugboard. Each out expander has 1 in hub and 2 related out hubs. When the "in" hub is impulsed the two related "out" hubs emit. These hubs are diode protected to prevent back circuits.

The out expanders are used when the exit of a step or function is to be wired to 4 or more places and/or to prevent back circuits.

As an example of the use of out expanders to prevent back circuits and also to illustrate when the delayed or immediate out of a program select would not be wired, assume that 2 distinct conditions will occur for one program and that both conditions must impulse program select 16. However, the first condition must impulse PS 16 from the out of program step 38 and then go to program step 39, while the second condition must impulse PS 16 from program step 41 and then go to program step 42. The "out" of program step 38 would be wired to the "in" of one out expander; one of the related "outs" would be wired to the "in" of program select 16 and the other to the "in" of program step 39. The "out" of program step 41 would be wired to the "in" of another out expander; one of the related "outs" would be wired to the "in" of program select 16 and the other to the "in" of program step 42. Thus, although the "outs" of both program steps 38 and 41 are wired to the "in" of PS 16 there will be no back circuit as to the choice of the next step since the out expanders are diode protected.

Illustration of wiring:

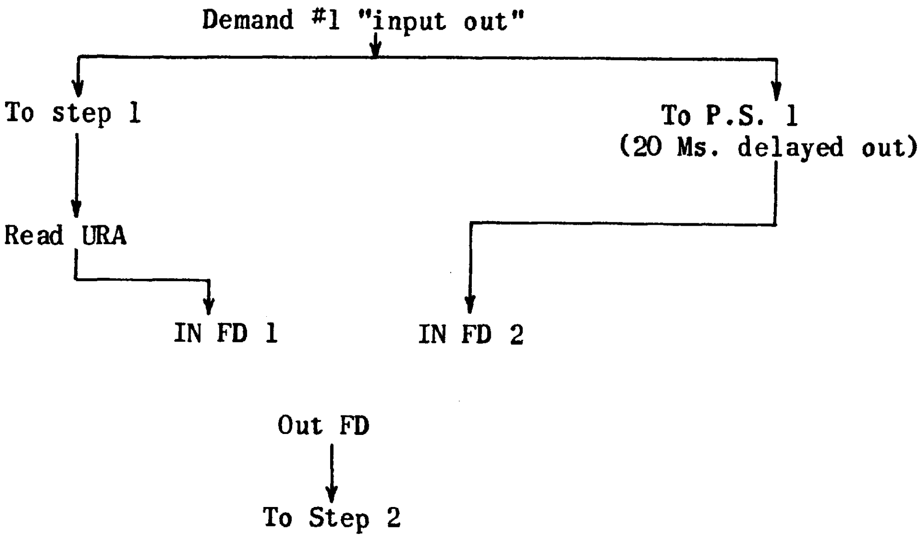


C. FUNCTION DELAY (line 3-8, c-d)

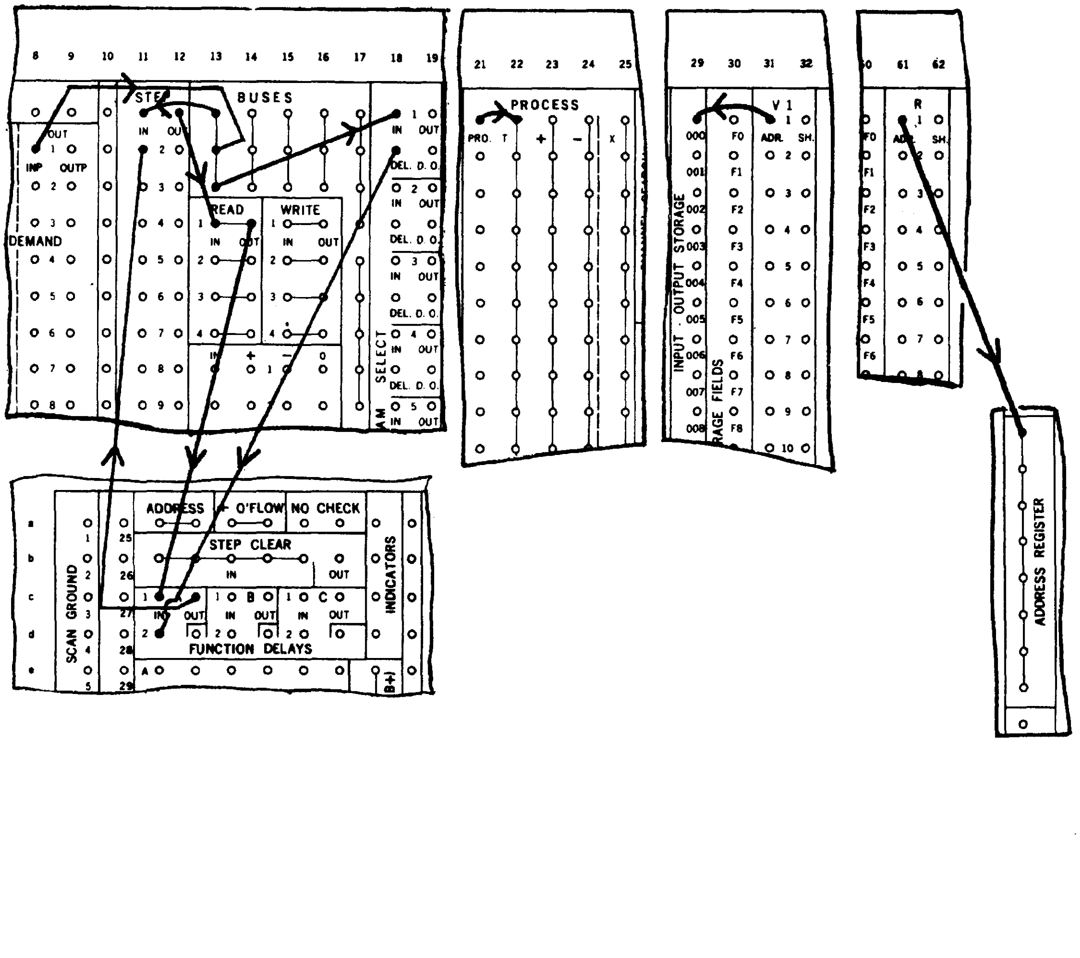
Function delay is for the purpose of delaying the start of an operation until two other operations have been completed. Three function delays are provided in the File-Computer.

Each function delay has two input hubs identified as In 1 and In 2, and one output hub. The output hub will emit only when a pulse has been delivered to each of the input hubs. These two pulses need not arrive simultaneously. Once a pulse has been received at one of the input hubs, the only way of clearing it out is by receipt of the pulse at the other input hub and the resultant emission from the output. The output may be wired to any place to which the exit of a step is wired.

Assume that the "input out" of demand input-output unit #1 is to start program step 1. Program step 1 is used to transfer an address from input storage to the ADR, and then a read unit record instruction is to be given. The immediate out of the read unit record is to be wired to program step 2 where value 1 and 2 are to be selected, based on input codes. There must be at least 20 elapsed milliseconds between the start of this program and the calling of V_1 and V_2 of step 2 since they are selected through selectors picked up by input controls. The function delay, in conjunction with program selector 1 (20 millisecond delayed out) can be used to insure that 20 milliseconds have elapsed:



The "input out" of demand unit 1 is bussed to the in of program step 1 and also to the in of program select 1. The out of step 1 is wired to read URA and the out of read URA to in FD 1. The out of program select 1 is wired to the other in of the function delay, in FD 2. The out of the function delay is wired to step 2. Thus the in of step 2 will not be impulsed until both the out of read URA and the 20 millisecond delayed out of program select 1 have emitted. Assuming the address is in input storage 000 the wiring for this example would be:



D. FUNCTION SEQUENCE (line 6-8, U-V)

There are 2 function sequence units on the plugboard. Each unit consists of these 3 hubs:

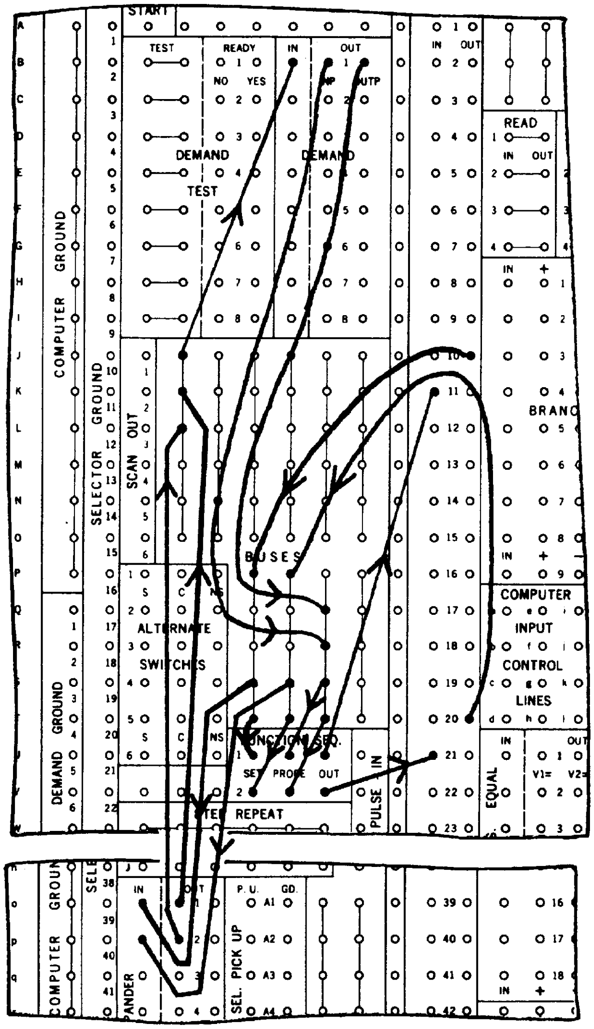
Set - An input hub to which a step out or similar pulse may be wired. A pulse delivered to this hub will be "remembered" by the device.

Probe - An input hub to which a step out or similar pulse may be wired. A pulse delivered to this hub will cause a pulse to be emitted from the out hub if a pulse was previously received by the set hub. A pulse delivered to the probe hub is not "remembered" by the device.

Out - This hub will emit a pulse whenever the probe hub is impulsed if the set hub was previously impulsed. Note, these units are similar to the function delay. However, the pulse to set must be received before the pulse to probe. Pulses delivered in the reverse order would have no effect on the device.

The function sequence units can be used effectively to route a program to different program steps or functions when one demand unit is referred to twice within one program. For example, assume that demand unit 1 is to be called for from the out of program step 10 and should start program step 11. It must also be called for from the out of program step 20 and should then start program step 21.

Illustration of the wiring: Note, out expanders are used to prevent back circuits.



E. CODE DISTRIBUTOR

The code distributor is a means whereby a two digit numeric code, ranging from 00 to 49, may be used to select the storage location to be used as V_1, V_2, or R in a program step. A one digit code, ranging from 0 to 9, may be used to select the next program step or function.

The one digit number (0-9) or two digit number (00-49), which is to do the selecting, will be located in a storage unit. In many cases it will be at an input storage location, though it may be in an intermediate or drum storage location. This number is transferred into the code distributor register, (line 68, A-H and 68, a-h). Spaces will not be accepted by the CD register. The CD register will recognize only the last two digits of a storage field. However, the number to be placed in the CD register must be located in the last two digits of the field, either originally or by shifting. Once a value has been placed in the CD register, it will remain there until replaced by a new value.

The first step in the use of the code distributor will always be the transfer (or addition, if necessary to provide zeros instead of spaces) of the storage in which the code is located into the CD register. Assuming that the code is in input storage 009, the step would appear as:

V_1	Pr	V_2	=	R
009	T	-		CD

The connection panel wiring would be the normal wiring for a transfer step.

Located on the connection panel are 61 hubs related to the code distributor. The hubs are numbered as follows:

00 through>49 - Codes for selection of V_1, V_2, and R. (line 44, A-z and 45, A-C).

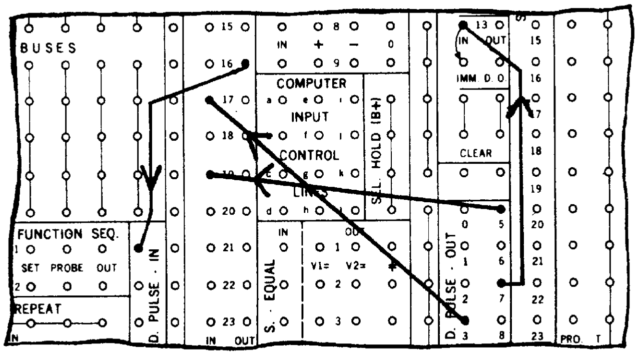
0 through 9 - codes for selection of next step or function (line 18-19, T-X).

The 00-49 codes are wired to storage locations. If, for example, it is desired to call on intermediate storage 25 if code 10 is in the CD register, the 10 CD hub would be wired to storage 25. As many of these hubs as desired can be wired to storage locations.

If any code greater than 49 is placed in the CD register, whatever storage location is wired to the>49 hub will be called upon.

The 0-9 hubs are wired to the entry of steps, interstep functions, or any other place to which step exits can be wired.

For example, if from the out of step 16, step 17 is desired when code 3 is present, step 19 when code 5 is present, and PS 13 when code 7 is present, the wiring would be:



Also associated with the code distributor are 4 hubs labeled CDV1 (line 30, f-i), 4 hubs labeled CDV2, (line 49, f-i), 4 hubs labeled CDR, (line 60, f-i), and 4 hubs labeled CD pulse IN, (line 9, U-X). These are timing hubs used to call upon the storage desired as the value or result of a step or the next function desired. For example, suppose that a quantity in a card is to be added to one of several intermediate storages, dependent upon which code is present. Assume the folowing:

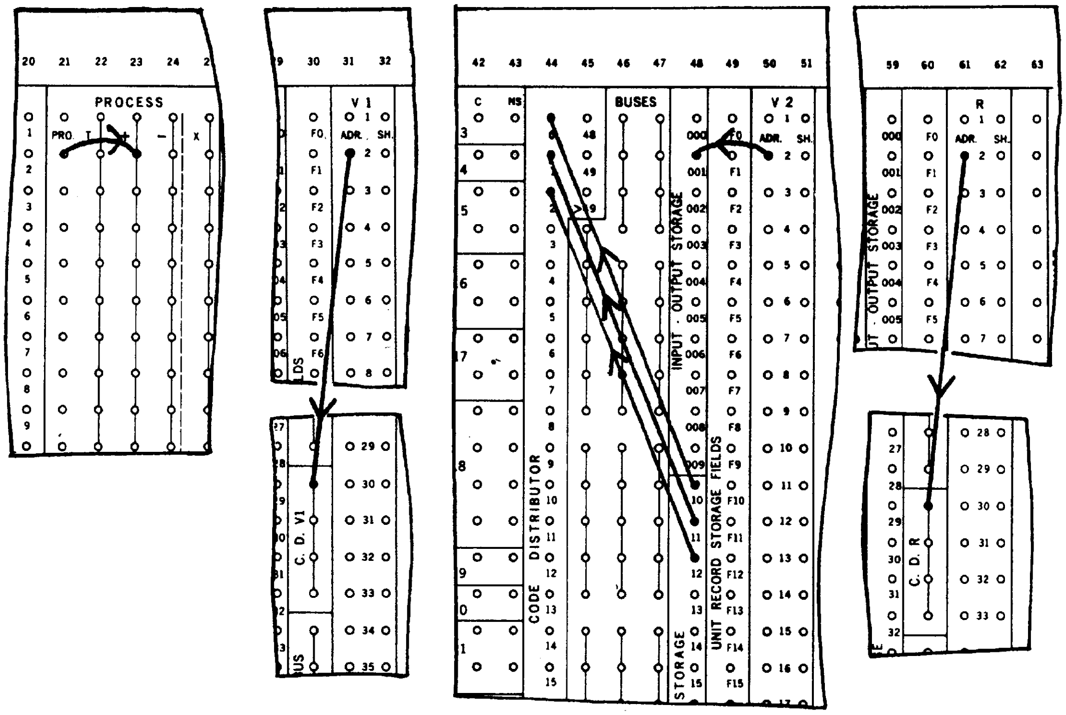
Quantity ordered: Input storage 001
Code: Input storage 000
If 00 - add to intermediate storage 10
If 01 - " " " " 11
If 02 - " " " " 12

The two program steps required would be:

Step No.	V_1	Pr	V_2	-	R	Next Step
1.	000 (code)	T			CD (CD register)	2
2.	CDV1 (prev. quan.)	+	001 (quan.)		CDR (new quan.)	3

The CDV1 means that at V_1 time of step 2, the storage location connected to the code in the CD register will be called upon. V_1 of step 2 is wired to one of the CDV1 hubs. If code 00 is in the CD register, storage 10 will be called upon; if code 01 is in the CD register, storage 11 will be called upon; if code 02 is in the CD register, storage 12 will be called upon. The same will hold true of the result.

The connection panel wiring for step 2 would be:



If a code is called upon which is not wired to a storage location, the machine will hang up.

F. <u>INPUT CONTROL LINES</u> (line 13-14, Q-T)

An input control line is a means of transferring a pulse from the input-output device, connected to the system through the multiplex function, to the computer connection panel.

There are 12 input control lines, identified as "a" through "1". A pulse delivered to one of these on an input-output connection panel will emit from the related input control line on the computer connection panel when that input-output device is connected to the system.

Normally, these control lines are used to pick-up a selector from a condition present in the input data, such as a control position in a punched card. The pulse to activate the pick-up of a selector comes from the input-output connection panel, through an input control line. The input control line on the computer connection panel is wired to the pick-up of the desired selector. (See section III, Input-Output Equipment).

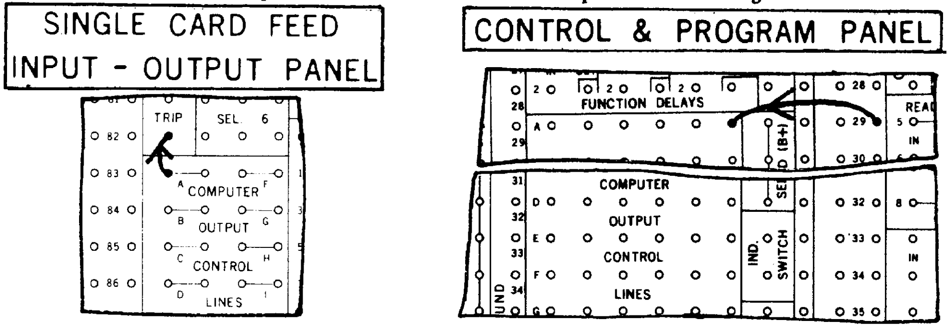
A selector picked up in this manner will remain on the select side as long as the input-output device which caused the pick-up to be impulsed, is connected to the system. 20 milliseconds should be allowed for the selector to change from the non-select to select if the desired condition is present in the input data. The selector will drop out when the multiplex function switches to another device.

G. OUTPUT CONTROL LINES (lines 3-8, e-n)

An output control line is a means of transferring a pulse from the computer connection panel to the particular input-output unit connected to the system through the multiplex function. There are 10 output control lines provided, indicated as A through J. There is no internal connection or relationship between input control lines and output control lines.

Each output control line has 6 entry hubs on the computer connection panel. This hubs are diode protected to prevent back circuits.

The pulse wired into an output control line is generally the exit of a step or some similar function. On the input-output device it is wired to some control function. For example, on a card sensing-punching unit, it might be used to segregate the card, to skip punching in the card, or to trip the card. It would be wired directly to the function desired. For example, to trip a card following step 29, the connection panel wiring would be:



If it is desired to continue the program on the computer, as well as perform the output function, the exit of the step should be bussed to the output control line and to the next step desired.

Note, both the input and output control lines are diode protected. The input control lines can only be used to deliver a pulse from the input output device to the computer connection panel; the output control lines can only be used to deliver a pulse from the computer to the input-output device.

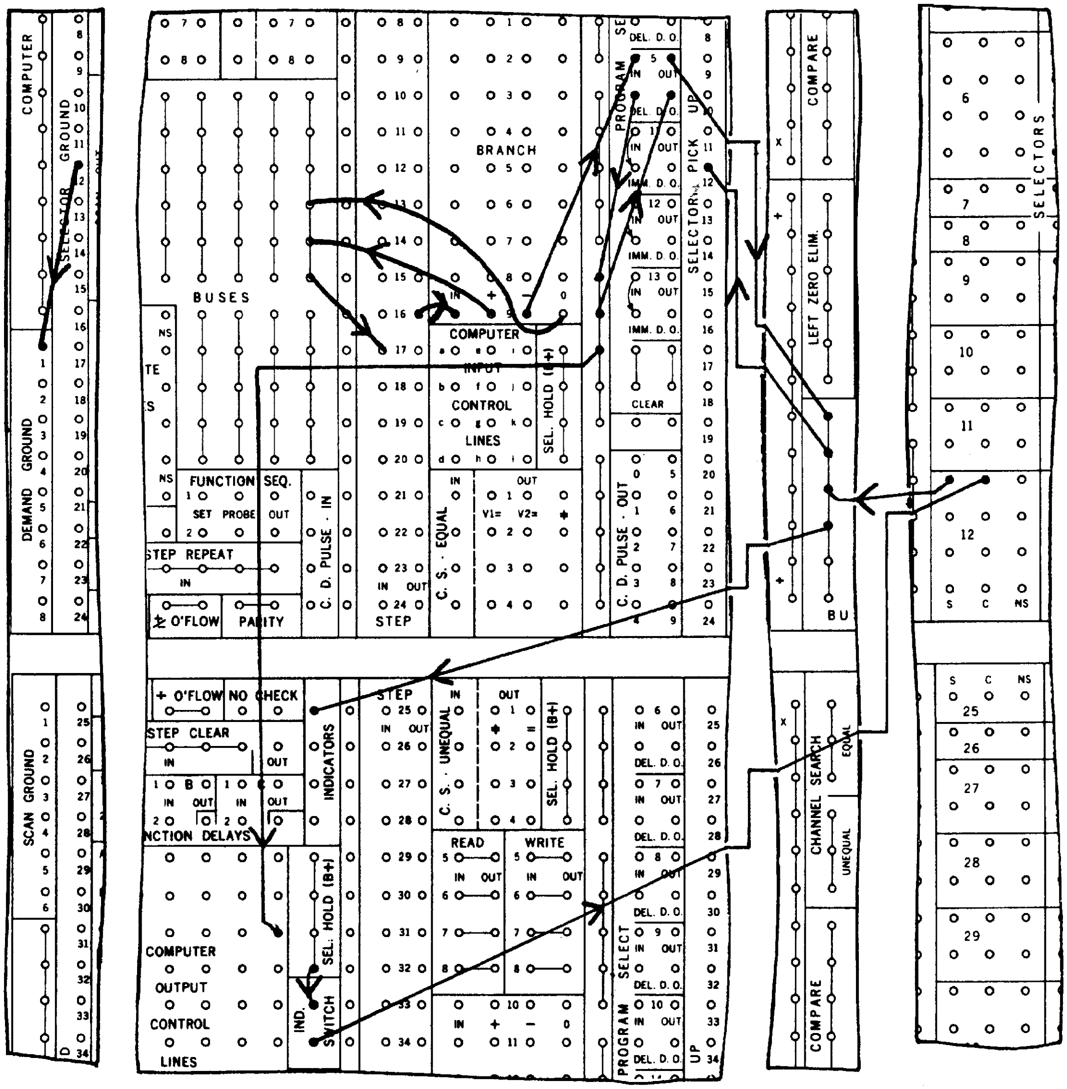
H. INDICATOR LIGHTS

There are four indicator lights on the computer display panel which can be lit from the computer connection panel, (line 9, a-d). An indicator light will be lit on the display panel when a pulse is delivered to the corresponding hub on the computer panel. It will stay lit as long as the pulse is present. It will drop out when the pulse is no longer impulsing the related hub.

Associated with these four lights is an indicator light switch (line 9, i-j). This switch is normally closed, allowing a pulse that enters one of the hubs to emit from the other. The switch is opened by a manual button mounted on the computer control cabinet. When this button is manually depressed the indicator switch is opened, breaking the connection between the two hubs on the computer control panel.

Normally, these lights are utilized, in conjunction with the indicator switch, B+ and a selector to indicate that a specific condition has been detected by the computer. For example, assume that demand input-output unit 1 is a card sensing-punching unit, and it is the only unit that goes through step 16. If a - result is obtained on program step 16 the machine should be stopped and the card that caused the - condition should be immediately investigated. Indicator light 1 should also be lit so the operator will know the cause of machine stoppage.

Illustration of wiring:



In the above illustration the out of step 16 is wired to the in of Branching #9. The - out hub is wired to the in of PS 5. (The + and o hubs would be wired to continue the normal program). The power out of PS 5 is wired to the pick up of selector 12. The ground of selector 12 is wired to demand ground #1. Thus, selector 12 will be select only if a - result was obtained on step 16.

The B+ is used to light indicator light 1 since it is a constantly emitting pulse. It is first routed through the indicator switch, so that the light can be manually dropped out when the card has been investigated. The B+ is then routed through the common and select of selector 12a to the indicator light 1 hub. The B+ pulse, then, can only get to the indicator light when selector 12 is select--- when a - result occurred on step 16.

The delayed out of PS 5 is wired to output transfer line C. On the input-output connection panel, output transfer line C is wired to skip the card out and to stop the machine. The delayed out is also wired to drop out PS 5. Selector 12, however, is maintained in the select position because B+, through the select of selector 12a, is impulsing the pick-up.

When the machine stops, the operator will instantly know that a - result occurred on step 16 since the indicator light 1 is lighted. To resume operation the manual indicator switch is manually depressed, thus dropping out the indicator light and also selector 12, and the start button depressed.

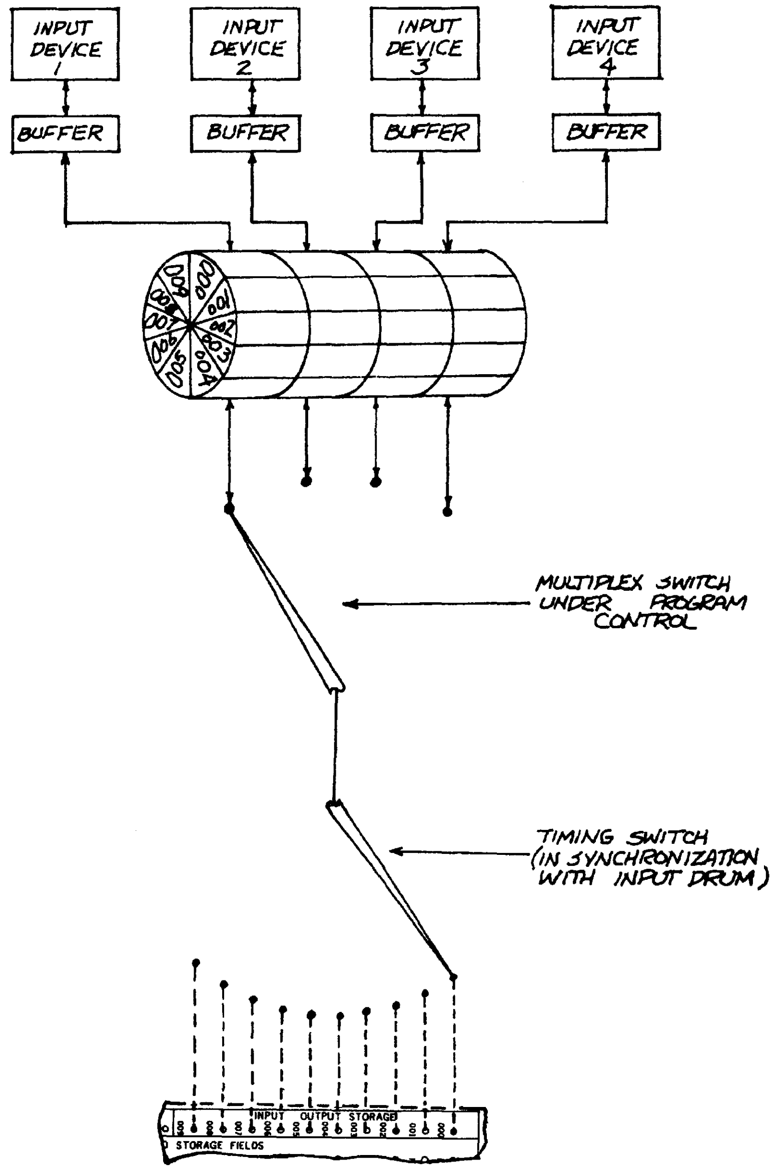
I. UNIT RECORD FIELD ASSIGNMENT SELECTOR PICK-UPS

On the connection panel for the large capacity drums there are 8 selectors. These selectors are identical in operation to the selectors previously mentioned. The pick-up and ground hubs, which are also identical to the pick-up and ground hubs for other selectors, are located on the computer connection panel. The wiring of these pick-ups and grounds is the same as that discussed previously in this section (6 E).

8. MULTIPLEX CONTROL

Since the Univac File-Computer can service more than one input device at a time it is essential that we have some means of controlling the connection of the input devices to the computer. This control is provided by the function called multiplexing. As will be recalled from Section IV (Storage) each input device enters its data into the buffer related to the device, and then the buffer transfers this data to its input drum track. It is from this input drum track that the computer obtains the input data and it is to this track that the output data is delivered. It may then be said that the multiplex function occurs between the input drum and the computer. Thus in a four (4) device system the schematic of input control is:

FIGURE 15
SCHEMATIC of INPUT CONTROL



In the diagram, the computer is connected through the multiplex switch in the storages related to input device #1, and any calculations performed at this time will be based on device #1's input information. In summary we can say that the multiplex function is a switching device which connects the computer to one input device's storage information at one time.

The multiplex operation may be controlled in one of two ways; these techniques are:

a. Scan Multiplexing
b. Demand Multiplexing

A. SCAN MULTIPLEXING

Scan multiplexing means that the machine will successively examine each scan input device to determine which device has information ready to be processed (i.e.-it has finished a complete input transfer cycle). The scan is sequential in nature in that it first looks at the storages for scan device 1 then scan device 2 etc. The multiplexer stops and connects any device's drum track to the computer if the device is ready. The multiplexer remains connected to the track until the calculations for the device are completed, and then (under program control) it moves on to test the next device. Thus many devices may be loading data into the input drum at one time, while one device is being processed by the computer. The program control for a scan multiplex operation is illustrated by the following example:

1. There are 4 input devices to be operated on a scan mode of operation and these devices are numbered 1,2,3 & 4.

2. Each device, when processed by the computer, is to go through a separate series of program steps as follows:

Device	In At Step	Out After Step
1	1	10
2	11	15
3	16	25
4	26	48

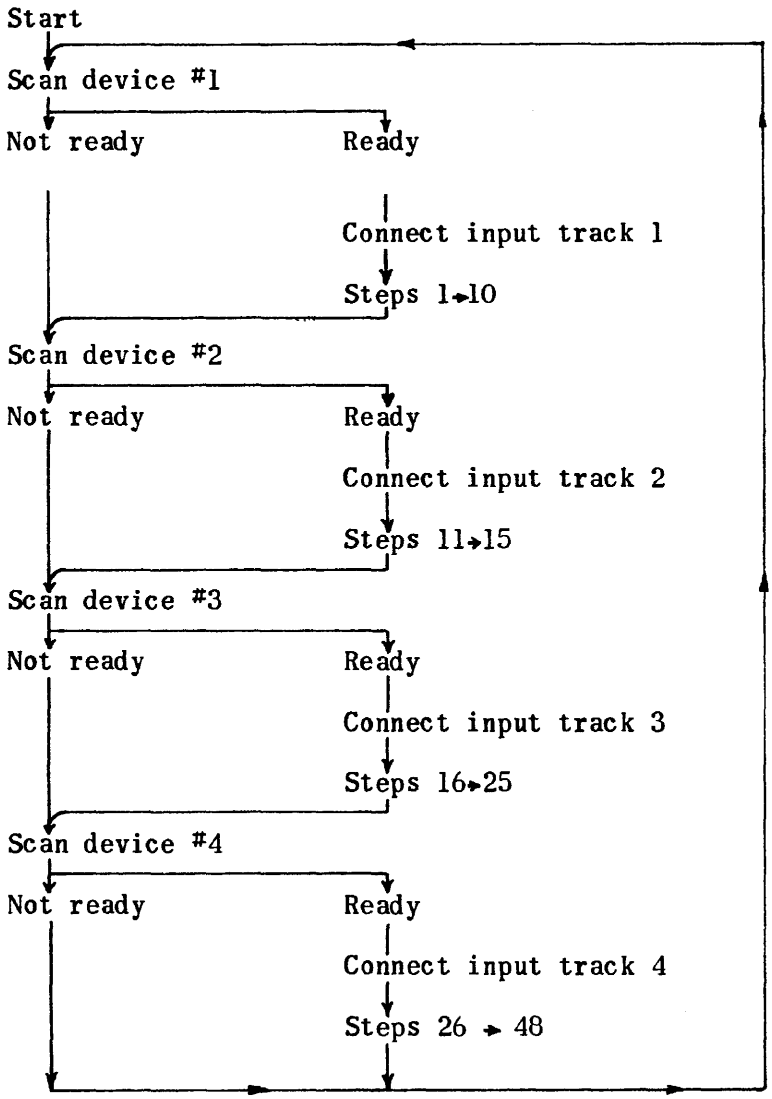
3. In order to have input devices operate in a scan mode we must sacrifice one "Demand Input Unit". This unit is used as the multiplex control for the scan devices. In this case we will assume Demand unit eight (8) controls input devices 1,2,3 & 4.

The following functions must be controlled by our wiring of the various input and computer plugboards:

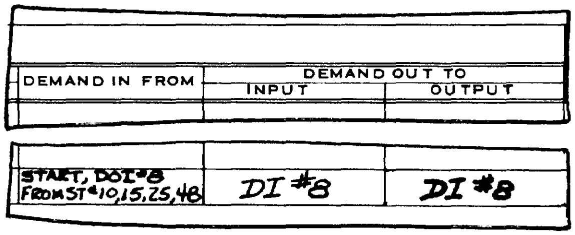
a. The adapter must scan the input devices in the following sequence: 1,2,3,4, 1,2,3,4, 1......4, 1......4, etc.

b. The computer must connect to the first device which is ready, process the data through the required steps, and then proceed scanning the next device in sequence.

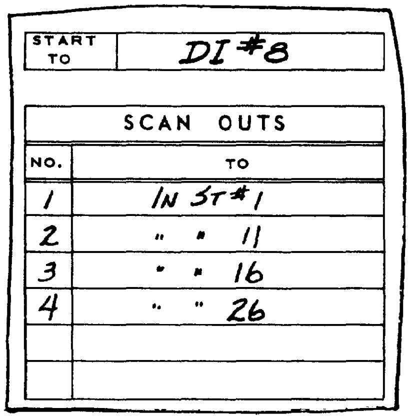
For example:



The program necessary to accomplish the above is:

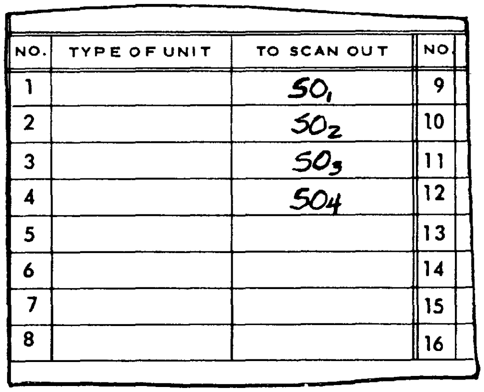


DEMAND IN FROM	DEMAND OUT TO	
	INPUT	OUTPUT
START, DOI#8 FROM ST#10,15,25,48	DI #8	DI #8



START TO	DI #8

SCAN OUTS	
NO.	TO
1	IN ST #1
2	" " 11
3	" " 16
4	" " 26

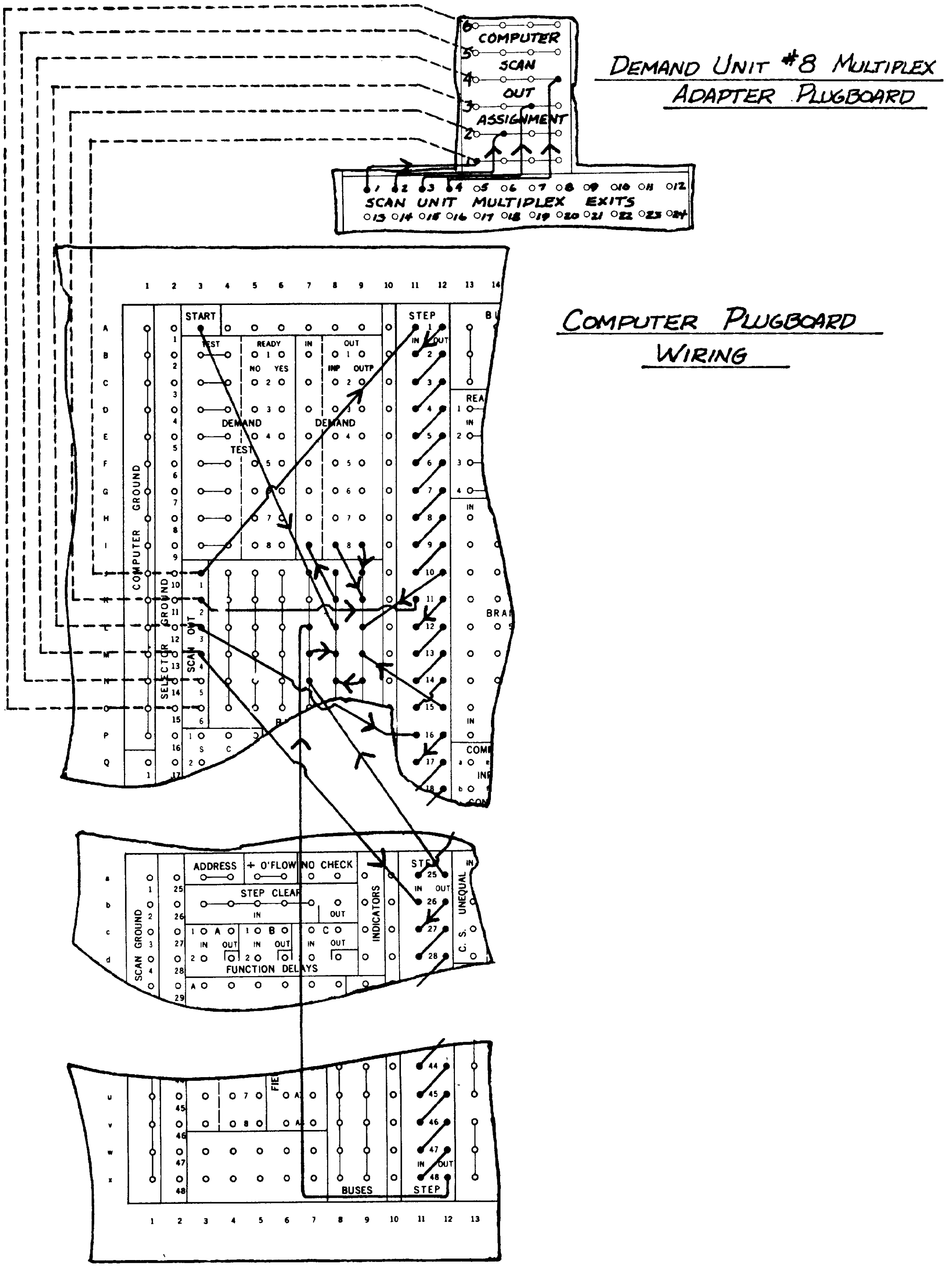


NO.	TYPE OF UNIT	TO SCAN OUT	NO.	
1		SO1	9	
2		SO2	10	
3		SO3	11	
4		SO4	12	
5			13	
6			14	
7			15	
8			16	

This program accomplishes the following:

When the start button is pressed we Demand Unit #8 which, since it is acting as a multiplexer for units 1 - 4, scans unit 1. If unit 1 is ready we proceed through steps 1 thru 10. At the end of step 10 the exit of the step redemands unit #8 and unit 2 is scanned. If unit 1 was not ready the next unit (unit 2) would automatically be scanned. After unit 4 is scanned and found not ready we obtain a pulse from the "demand out" which is wired back to the "demand in" to repeat the cycle beginning with unit # 1.

The plugboard wiring to accomplish the above would be:



As can be seen by the dotted lines in the diagram, the computer Scan-Out hubs on the multiplex adapter plugboard are internally connected with the "Scan Out" hubs on the computer plugboard. Thus the scan unit multiplex exits emit current when the particular device is ready to be processed and the computer is connected to the related input track. This current comes out of the computer plugboard routine scan out assignment hubs and is directed through external wires to the appropriate step. The completion of a devices routine re-demands the Multiplex adapter and tells it to test the next unit and so on. Of course if any unit in the sequence is not ready to be processed the multiplexer automatically moves on and tests the next device in sequence.

As can be seen there are only six (6) routine assignment hubs for each multiplex adapter although each adapter can handle up to 24 devices. This should not present a problem since it is rare that more than 6 routines would be required in any one processing run. However, should such a case occur, selector techniques could be used to effectively expand the routine possibilities.

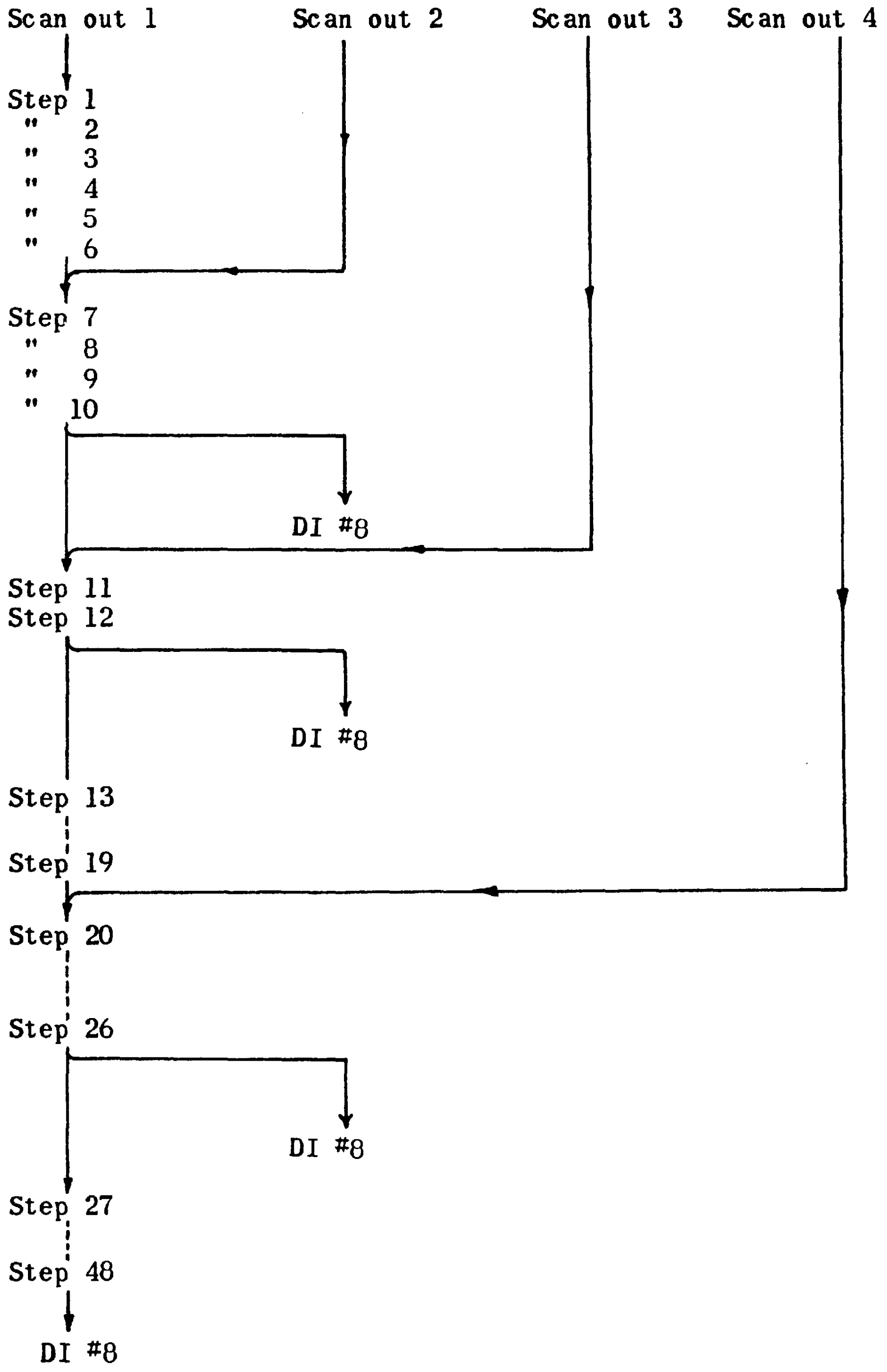
SCAN GROUND

It may often be true that the input device routines overlap in certain steps of the program. Thus some means of branching step exits on the basis of what routine is effective at the time is needed. Scan ground provides a means of accomplishing this in a very direct manner. Scan ground is merely a ground function which is controlled by the routine used. Thus, on the computer plugboard, there is a scan ground exit hub for each scan out hub. Their use might be best shown by an example:

Assume that in the last case we want:

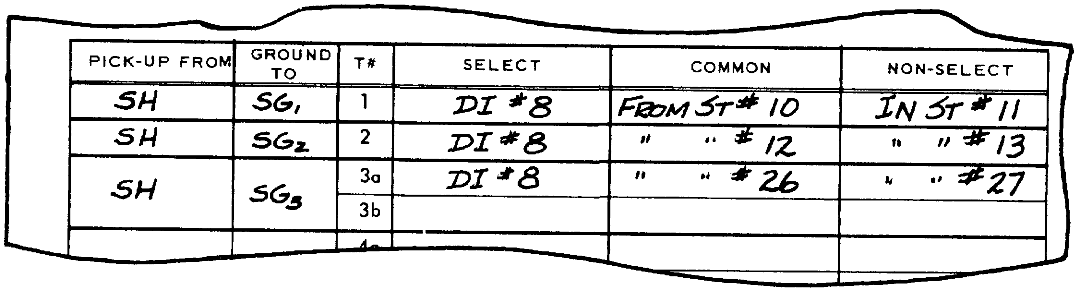
Device No.	In At Step	Out At Step
1	1	10
2	7	12
3	11	26
4	20	48

As can be seen there are step overlaps between each devices routine, these may be flow charted as:



The flow chart clearly points out that we must select the exits of steps 10, 12, and 26 on the basis of which routine is being processed.

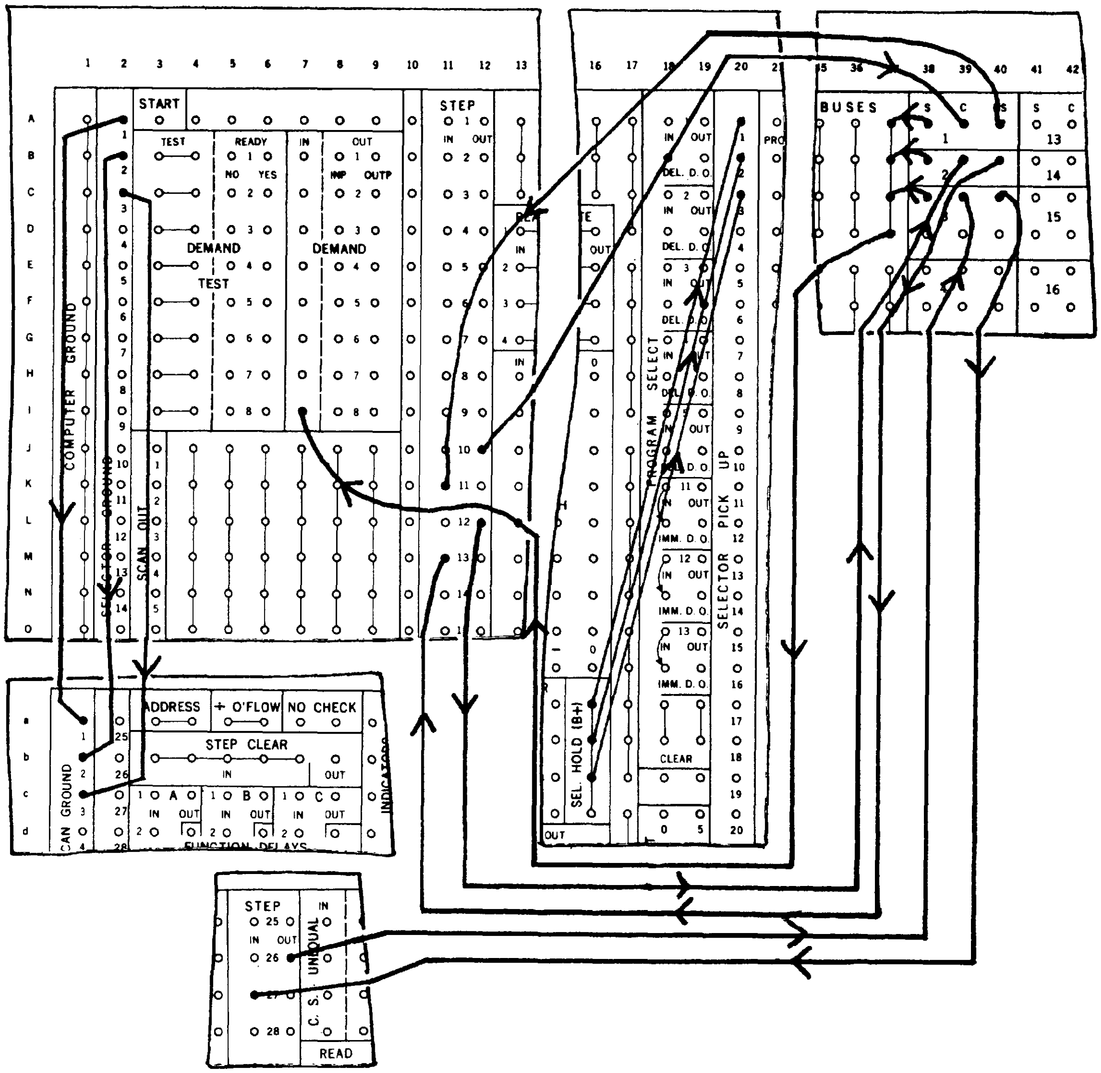
The selector programming for this is:



PICK-UP FROM	GROUND TO	T#	SELECT	COMMON	NON-SELECT
SH	SG_1	1	DI #8	FROM ST # 10	IN ST # 11
SH	SG_2	2	DI #8	" " # 12	" " # 13
SH	SG_3	3a	DI #8	" " # 26	" " # 27
		3b			

The fact that selector hold is wired to the pickup of the three selectors indicates that current is entering each selector pickup at all times during the program. However, since the selector grounds are wired to scan ground, the circuit will be completed (and the selector transferred) only when the appropriate device is connected to the computer through the multiplex adapter, thus allowing the selector hold pulse to get to ground.

The wiring for these selectors would be:



In summary, note the following facts concerning the "Scan Multiplex" mode of operation.

1) One demand unit is required to act as a multiplex adapter for the scan units.

2) Only six scan out hubs are available for all scan units included in a system.

3) The demand unit appropriated to act as a multiplex adapter for the scan units serves the following purposes:

a) The Demand "In" hub starts the sequential scan of the input devices when it receives a pulse. The unit the scan begins on is always the unit following the last one processed by the computer.

b) The Demand "Out" hub emits when the last scan unit has been tested and found not ready. This pulse is usually wired back to the Demand "In" hub to start the scanning cycle again. It also emits when the last scan unit has been processed.

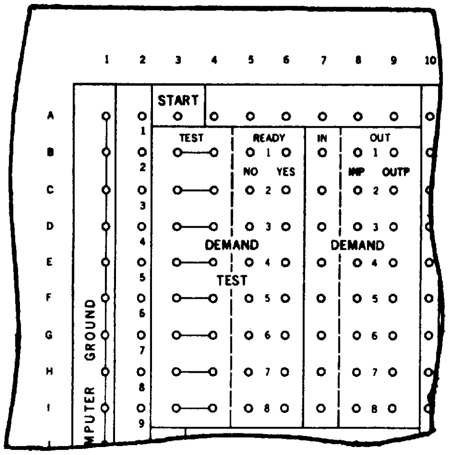
4) Through use of "Scan Ground", selectors can be activated to identify which routine or unit is being processed by the computer at any time.

B. DEMAND PROGRAMMING

In the preceding section the "Scan Multiplex" mode of operation was discussed. In this portion the other mode of input-output device control available with the Univac File-Computer is covered. It should be noted that either or both modes of control can be utilized in one program.

Scan Control is a technique whereby the input device informs the computer that it is ready to be processed. In other words the input device controls the computer. Obviously such an arrangement is not satisfactory for programs which require activating an input-output device on the basis of a condition arising in the program, nor would "scan multiplexing" be of any use in handling a pure output device such as an on-line printer. It is for these reasons that the Univac File-Computer has the capacity to include up to 8 input-output units operating on what is known as a "Demand" basis. The term demand means that the unit can be directly interrogated and activated by the program itself, rather than by automatic multiplex scanning.

Before illustrating the use of Demand controlled input-output units, the functions of the plugboard hubs related to each demand unit are described.



As can be seen, there is provision for eight input-output demand units. This is of course reduced by one for each scan adapter included in the system.

This true since a Demand unit's controls must be utilized to perform the multiplex function for the scan devices. The meaning and use of the "demand" unit hubs are as follows:

1) Test: These two common hubs are probed by a program pulse (start, step exit, etc.) to determine if the unit's input storage is full. This is, in effect, the same as a scan multiplex test except that here the test is under control of the program; in a scan multiplex operation the test is automatic. It should be noted that entering current in these hubs does not connect the computer to the device. Instead it merely tests the device's input storages to determine if the device has completed its input transfer cycle.

2) Ready: This hub emits current after a pulse has been delivered to the test hubs if the device has completed the input transfer. This current is usually wired to the "In" hub in order to connect the computer to the device.

3) Not Ready: This hub emits current after a pulse has been delivered to the test hubs if the device has not completed the input transfer. This current is usually used to test another device or to energize some program step to be done while the device is loading.

   Note: These last two hubs (Ready and Not Ready) are analogous to the scan multiplex operation of either connecting the device to the computer when the device is ready, or proceeding to the next unit if the device is not yet ready. Again the only difference lies in the fact that in the "Demand" operation the alternatives are under control of the programmer rather than an automatic function.

4) In: When a pulse is delivered to this hub the computer connects to the related device's drum input-output track, thereby making the input-output track related to the device available to the computer for values or results.

5) Out Inp. After a pulse has been delivered to the "In" hub this hub will emit current if the last operation the device accomplished was a reading function (tape read, etc). In other words this hub tells us two things -

   a) The computer is now connected to this device's input-output track and,

   b) The last thing this device did was read, that is accept input data.

   This hub is usually wired to the in of a program step in order to initiate the device's program routine.

   It should be noted that if the device is in the process of transferring input data, this hub will not emit until the transfer is completed.

6) Out-Outp: This hub performs exactly the same function as Out-Inp except that emission of current from this hub indicates that the last input-output function that the device performed was that of writing or recording output data, rather than reading as in 5.

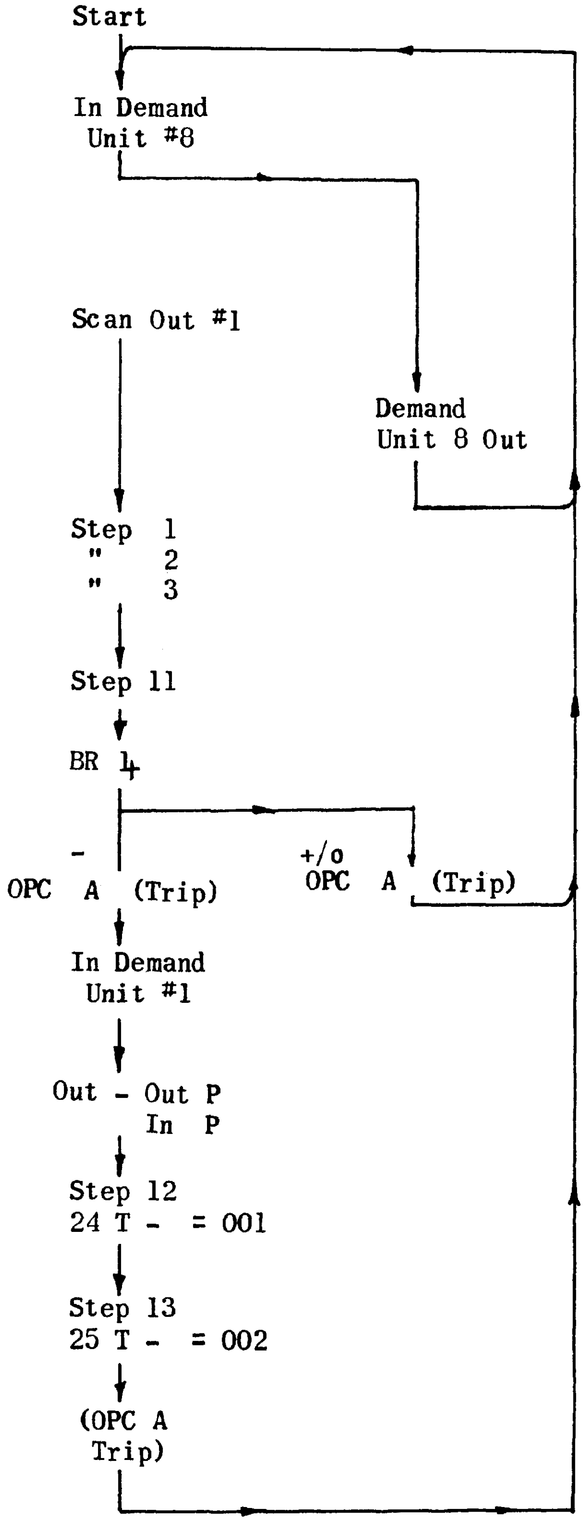
To illustrate the use of a demand unit assume the following case:

1) The system includes 4 scan units controlled by Demand Unit #8 which are all doing a simple pre-billing and inventory deletion program which runs from step 1 - step 11. These scan units are all assigned to routine #1.

2) If the result of step 11 is negative, place the values in intermediate storages 24 and 25 into output storages 001 and 002 of Demand unit #1 (a 150 CPM card punching device), and then return to the normal scan multiplexing operation.

Note: This might be a case of punching a below re-order point card when the inventory on any item went below its minimum balance.

The operations to be performed might be flow charted as:



The program controls for this operation would be:

START TO	DI #8

SCAN OUTS	
NO.	TO
1	IN ST. #1

DEMAND IN FROM	DEMAND OUT TO	
	INPUT	OUTPUT
OE_1	IN ST. #12	IN ST. #12
START, DOI #8, DOO #8, OE_2	DI #8	DI #8

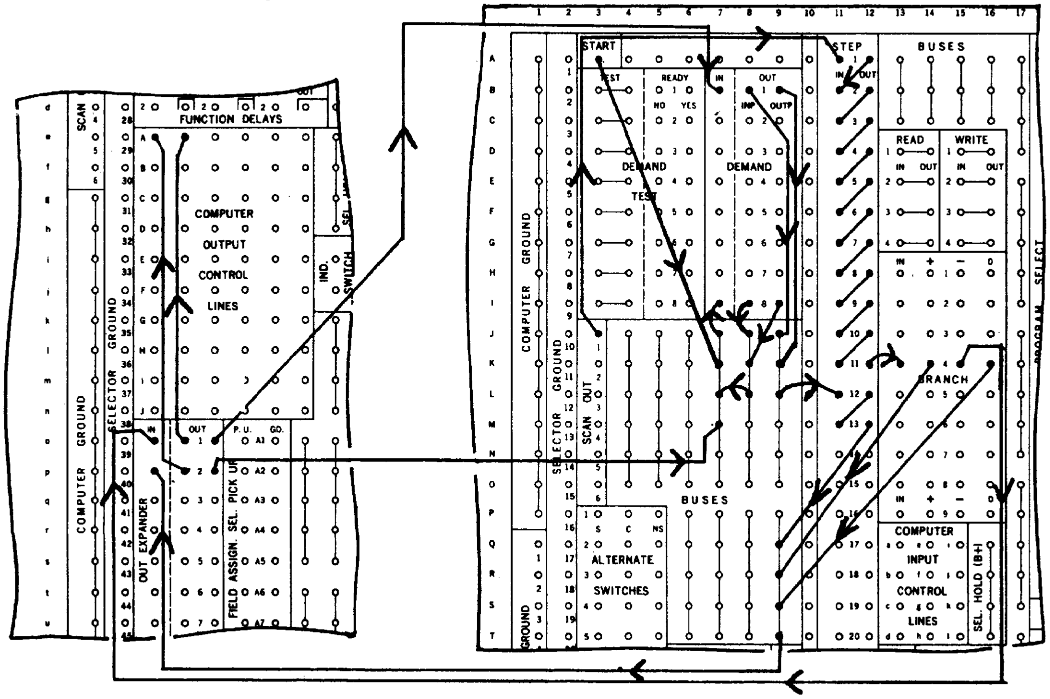
NO.	TO SCAN OUT
1	SO_1
2	SO_1
3	SO_1
4	SO_1
5	
6	
7	
8	

OUTPUT CONTROL LINES	
NO.	FROM
A	OE_1, OE_2
B	

BRANCHING				
NO.	IN FROM STEP	TO IF +	TO IF −	TO IF O
4	#11	OE_2	OE_1	OE_2
5				

OUT EXPANDERS			
NO.	IN	OUT	OUT
1	− BR_4	OPC A	DI #1
2	+/0 BR_4, ST#3	OPC A	DI #8
3			

The computer plugboard Multiplex and Demand Control Wiring would be:



Although this example indicates the most common use of Demand Units it should be noted that through use of the "Test in", the "Ready" and "Not Ready" hubs a demand unit can be made to operate just as a scan unit.

9. SUMMARY - TIMING

The time required to perform and prove a step depends on four (4) factors:

A. Access time to storage fields.
B. Shift time
C. Loading and Unloading time of the Arithmetic Unit registers
D. Process operation time.

The sum of these four factors will determine the length of time required to perform a complete three address step.

A. Access time

Access time is the time required to locate a specific storage field. This must be determined for V_1, V_2, and R. The average access times are:

1.	Input-output storage	2.5 milliseconds
2.	Intermediate storage	2.5 milliseconds
3.	High speed storage	2.5 milliseconds
4.	Large capacity revolver	.343 to 3.429 milliseconds

Access time to a field in the large capacity storage drum is dependent upon the size of the unit record area. The maximum access time is the length of time required for one unit record area to pass under the read or write head of the revolver. This can be found by multiplying the unit record length by .05714 milliseconds. The average time is one-half of this amount.

Unit Record Length	Maximum access		Average access	
12	.686	milliseconds	.343	milliseconds
15	.857	"	.429	"
20	1.143	"	.572	"
24	1.371	"	.686	"
30	1.714	"	.857	"
40	2.286	"	1.143	"
50	2.857	"	1.429	"
60	3.429	"	1.715	"
75	4.286	"	2.143	"
100	5.714	"	2.857	"
120	6.857	"	3.429	"

B. Shift time

Shift time is .042 milliseconds for each place shifted.

C. Register loading and unloading time

The time to transfer a storage to or from the Arithmetic Unit registers is 0.504 milliseconds or the equivalent of 12 shifts (12 X .042 = .504).

D. Process time

Process time is the time required by the arithmetic section to perform and prove the process. This time is not constant for each process. In an arithmetic process it varies with the number and value of the digits entering the calculation. In the transfer process it varies with the source and destination of the value. In the channel search process it varies with the ADR switching time and number of channels to be searched.

Listed below are the times of, or formulae for computing the time of, the nine (9) processes performed by the Arithmetic Unit. These times are measured from the "Initiate Process" pulse to the "End Process" pulse, and therefore, do not include access time to storage media nor the loading and unloading time of the Arithmetic Unit registers.

1. ADD:

(a) Alpha-Numeric: T = 0.71 Milliseconds.
(b) Numeric-Including Check: T = 1.89 Milliseconds.

2. SUBTRACT:

(a) Alpha-Numeric: T = 0.71 Milliseconds.
(b) Numeric, $VI-V2 \geq 0$: T = 1.89 Milliseconds.
(c) Numeric, $VI-V2 < 0$: T = 2.47 Milliseconds.

3. MULTIPLY:

$T = 1.60 + 0.588 (N1)$ Milliseconds.

4. MULTIPLY CHECK:

$T = 1.00 + 1.00 (N1)$ Milliseconds.
Where $N1$ = sum of the value of digits in the multiplier.

5. DIVIDE:

$T = 0.8 + 1.2\ (l-k+n+l) + 0.588\ (N2)$ Milliseconds.

Where: l = Dividend digits
k = Divisor digits
n = V1 Left shifts
$l-k+n+l$ = Number of digits in quotient.
$N2$ = Sum of the value of the quotient digits

6. DIVIDE CHECK:

$T = 1.6 + 0.042\ (k-n) + 0.588\ (N2)$ Milliseconds.

7. MASKING TRANSFER:

$T = 0.63$ Milliseconds.

8. COMPARE:

$T = 0.67$ Milliseconds.

9. LEFT ZERO ELIMINATION:

$T = 0.042 + 0\ 0.084n$ Milliseconds.

10. TRANSFER:

$T = 0.504$ Milliseconds.

11. CHANNEL SEARCH:

Four different factors are involved in determining channel search time when overflow channels are not included. If overflow channels are used, an additional factor is involved. The factors listed do not include the placing of the address in the ADR, since that would be done in a preceding step. The timing of this step would be the normal timing for the step involved, except that no access time need be allowed for the ADR. Instead 6 right shift time is required for ADR entry or 6 x .042 = .252 ms.

The factors involved in channel search without overflow are:

a. Switch time: The address register switches to the indicated drum section and channel as soon as it receives an entry (provided that a read or write operation is not in process). Therefore, this time may or may not be included in channel search time. The time required for the switching is:

1. To a different drum	0.1 Milliseconds
2. To a different drum section in the same drum	0.1 Milliseconds
3. To a different channel in the same section	0.1 Milliseconds
4. To same channel in same drum section	0.0 Milliseconds

b. Access time: The average time to find the starting point (index point or starting address) is 17 milliseconds.

c. Search time: The search time is the time required to locate the desired record. Since one complete drum revolution is 34 milliseconds, the average search time to a specific record is 17 milliseconds.

The time for a channel search, not including overflow, would be the sum of switch time, access time and the search time. For example, to locate a record in another drum would be:

0.1 + 17 + 17 = 34.1 milliseconds.

If there is a possibility of overflow channels, the time to transfer the overflow address to the ADR need not be included in the calculations since it will be overlapped with the time required to locate the starting point of the next channel to be searched.

12. Read Unit Record

When a complete address, DS/CH/URA (xxxxxx) is transferred to the ADR, it must be followed by a Read instruction to transfer this record to the revolver. Three factors are involved in determining Read time. The factors involved are:

a. Search time: 17 milliseconds (average)
b. Read time: This time is dependent on the unit record length and may be determined by the following table:

Unit Record Length	Read-write time
12	.686
15	.857
20	1.143
24	1.371
30	1.714
40	2.286
50	2.857
60	3.429
75	4.286
100	5.714
120	6.857

c. Access time: If switching is done to another drum it is necessary to add 17 milliseconds. This is the average time required to locate the starting point for the new drum.

13. Write Unit Record

The time required to transfer a unit record from the revolver back to its drum location (or some other location if a new address is transferred to the ADR before the write instruction is given) involves two factors. These factors, are as follows

a. Search Time : 17 milliseconds
b. Write Time: determined by table for Read Unit record.

Whether a complete address is used to capture a unit record or channel search is used, it is necessary to give a "write" instruction to transfer the record on the revolver back to the drum. This time may be overlapped if the write command occurs in the program at a time where additional steps, not utilizing unit record fields, are required.

V-93

UNIVAC FILE - COMPUTER

CONTROL & PROGRAM PANEL

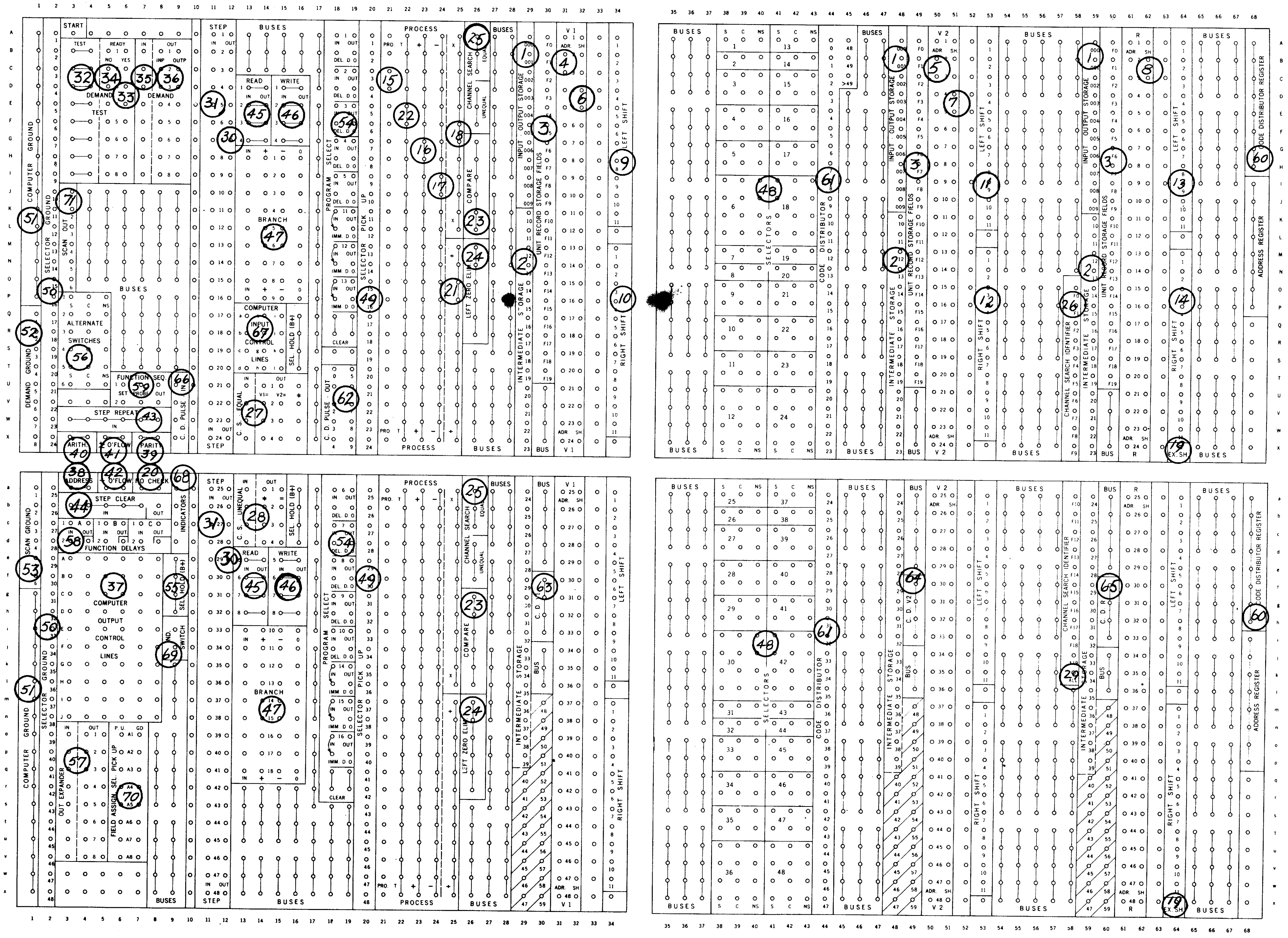
Remington Rand

DIVISION OF SPERRY RAND CORPORATION

TABULATING MACHINES

V - 93

APPLICATION:—



Remington Rand Univac

DIVISION OF SPERRY RAND CORPORATION

V - 94

UNIVAC FILE-COMPUTER SYSTEM

STORAGE ASSIGNMENT CHART

APPLICATION: ______________________ PROGRAM NO.: ______________

LARGE CAPACITY DRUM

ADDRESS FROM ____________

TO ____________

ENTER THE STARTING ADDRESS OF THE URA'S RELATED TO THIS APPLICATION, I.E., 000000.

ENTER THE ENDING ADDRESS OF THE URA'S RELATED TO THIS APPLICATION, I.E., 029949.

SYM		11	10	9	8	7	6	5	4	3	2	1	SN
F 0													
F 1													
F 2													
F 3													
F 4													
F 5													
F 6													
F 7													
F 8													
F 9													
F10													
F11													
F12													
F13													
F14													
F15													
F16													
F17													
F18													
F19													

ENTER DESCRIPTION OF THE FIELD, I.E., BALANCE ON HAND, EMPLOYEE NO., ETC.

ENTER THE CHARACTERS ASSIGNED TO A FIELD AND INDICATE DECIMAL POSITION, I.E., ▲

IF THE SIGN IS STORED, ENTER THE CHARACTER; IF THE SIGN IS NOT STORED, ENTER + TO INDICATE AN APPLIED PLUS SIGN MUST BE WIRED FOR THE FIELD.

LARGE CAPACITY DRUM FIELD ASSIGNMENT

ADDRESS FROM ____________

TO ____________

SYM	DESCRIPTION	CHARACTER AND DECIMAL 11	10	9	8	7	6	5	4	3	2	1	SN
F 0													
F 1													
F 2													
F 3													
F 4													
F 5													
F 6													
F 7													
F 8													
F 9													
F10													
F11													
F12													
F13													
F14													
F15													
F16													
F17													
F18													
F19													

IF TWO TYPES OF URA'S ARE STORED FOR ONE APPLICATION, USE THIS TO INDICATE THE SECOND FIELD ASSIGNMENT LAYOUT.

LARGE CAPACITY DRUM FIELD ASSIGNMENT SELECTORS

PICK-UP FROM	GROUND TO	A#	SELECT	COMMON	NON-SELECT
		1a			
		1b			
		1c			
		1d			
		2a			
		2b			
		2c			
		2d			
		3a			
		3b			
		3c			
		3d			
		4a			
		4b			
		4c			
		4d			
		5a			
		5b			
		5c			
		5d			
		6a			
		6b			
		6c			
		6d			
		7a			
		7b			
		7c			
		7d			
		8a			
		8b			
		8c			
		8d			

ENTER THE NUMBER OF THE INPUT CONTROL LINE OR PROGRAM SELECTOR WHICH IS TO IMPULSE THE PICK UP, I.E., IPC a, PSO 12, ETC.

ENTER TYPE OF GROUND, I.E., SCAN GROUND 3, DEMAND GROUND 4, COMPUTER GROUND, ETC.

ENTER THE CHARACTER, APPLIED PLUS SIGN, ETC. DESIRED WHEN THE SELECTOR IS ON THE NON-SELECT SIDE.

ENTER THE UNIT RECORD FIELD, CHARACTER, ETC THAT IS TO BE SELECTED.

ENTER THE CHARACTER, APPLIED PLUS SIGN, ETC. DESIRED WHEN THE SELECTOR IS ON THE SELECT SIDE.

INTERMEDIATE STORAGE FIELD ASSIGNMENT

TRACK 50

SYM	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
50													
51													
52													
53													
54													
55													
56													
57													
58													
59													

INTERMEDIATE STORAGE FIELD ASSIGNMENT

TRACK 10

SYM.	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													

ENTER DESCRIPTION OF THE STORAGE UNIT, I.E., ROUNDING 5, CONSTANT 14, WORKING STORAGE, ETC.

ENTER THE VALUE, IF IT IS KNOWN, I.E., CONSTANT 5.

TRACK 20

SYM.	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
20													
21													
22													
23													
24													
25													
26													
27													
28													
29													

INDICATE SIGN POSSIBILITIES, IF THEY ARE KNOWN, I.E., +, -, +/- etc.

TRACK 30

SYM.	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
30													
31													
32													
33													
34													
35													
36													
37													
38													
39													

TRACK 40

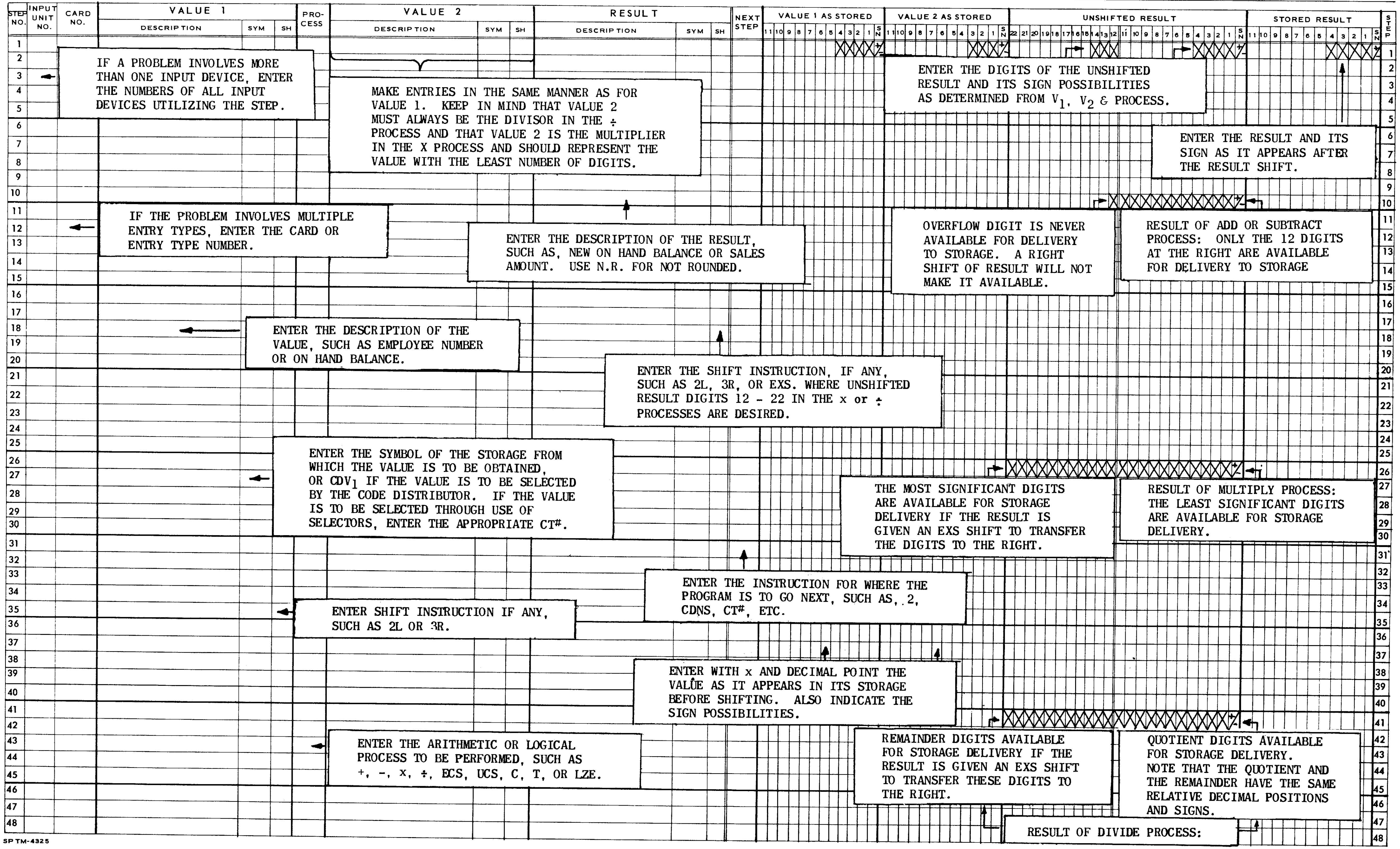
SYM.	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
40													
41													
42													
43													
44													
45													
46													
47													
48													
49													

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

V - 95

UNIVAC FILE-COMPUTER SYSTEM
PROGRAM PLANNING CHART

APPLICATION: ____________________ PROGRAM NO.: ____________



IF A PROBLEM INVOLVES MORE THAN ONE INPUT DEVICE, ENTER THE NUMBERS OF ALL INPUT DEVICES UTILIZING THE STEP.

MAKE ENTRIES IN THE SAME MANNER AS FOR VALUE 1. KEEP IN MIND THAT VALUE 2 MUST ALWAYS BE THE DIVISOR IN THE ÷ PROCESS AND THAT VALUE 2 IS THE MULTIPLIER IN THE X PROCESS AND SHOULD REPRESENT THE VALUE WITH THE LEAST NUMBER OF DIGITS.

ENTER THE DIGITS OF THE UNSHIFTED RESULT AND ITS SIGN POSSIBILITIES AS DETERMINED FROM V_1, V_2 & PROCESS.

ENTER THE RESULT AND ITS SIGN AS IT APPEARS AFTER THE RESULT SHIFT.

IF THE PROBLEM INVOLVES MULTIPLE ENTRY TYPES, ENTER THE CARD OR ENTRY TYPE NUMBER.

ENTER THE DESCRIPTION OF THE RESULT, SUCH AS, NEW ON HAND BALANCE OR SALES AMOUNT. USE N.R. FOR NOT ROUNDED.

OVERFLOW DIGIT IS NEVER AVAILABLE FOR DELIVERY TO STORAGE. A RIGHT SHIFT OF RESULT WILL NOT MAKE IT AVAILABLE.

RESULT OF ADD OR SUBTRACT PROCESS: ONLY THE 12 DIGITS AT THE RIGHT ARE AVAILABLE FOR DELIVERY TO STORAGE

ENTER THE DESCRIPTION OF THE VALUE, SUCH AS EMPLOYEE NUMBER OR ON HAND BALANCE.

ENTER THE SHIFT INSTRUCTION, IF ANY, SUCH AS 2L, 3R, OR EXS. WHERE UNSHIFTED RESULT DIGITS 12 - 22 IN THE x or ÷ PROCESSES ARE DESIRED.

ENTER THE SYMBOL OF THE STORAGE FROM WHICH THE VALUE IS TO BE OBTAINED, OR CDV_1 IF THE VALUE IS TO BE SELECTED BY THE CODE DISTRIBUTOR. IF THE VALUE IS TO BE SELECTED THROUGH USE OF SELECTORS, ENTER THE APPROPRIATE CT#.

THE MOST SIGNIFICANT DIGITS ARE AVAILABLE FOR STORAGE DELIVERY IF THE RESULT IS GIVEN AN EXS SHIFT TO TRANSFER THE DIGITS TO THE RIGHT.

RESULT OF MULTIPLY PROCESS: THE LEAST SIGNIFICANT DIGITS ARE AVAILABLE FOR STORAGE DELIVERY.

ENTER THE INSTRUCTION FOR WHERE THE PROGRAM IS TO GO NEXT, SUCH AS, .2, CDNS, CT#, ETC.

ENTER SHIFT INSTRUCTION IF ANY, SUCH AS 2L OR 3R.

ENTER WITH x AND DECIMAL POINT THE VALUE AS IT APPEARS IN ITS STORAGE BEFORE SHIFTING. ALSO INDICATE THE SIGN POSSIBILITIES.

ENTER THE ARITHMETIC OR LOGICAL PROCESS TO BE PERFORMED, SUCH AS +, -, X, ÷, ECS, UCS, C, T, OR LZE.

REMAINDER DIGITS AVAILABLE FOR STORAGE DELIVERY IF THE RESULT IS GIVEN AN EXS SHIFT TO TRANSFER THESE DIGITS TO THE RIGHT.

QUOTIENT DIGITS AVAILABLE FOR STORAGE DELIVERY. NOTE THAT THE QUOTIENT AND THE REMAINDER HAVE THE SAME RELATIVE DECIMAL POSITIONS AND SIGNS.

RESULT OF DIVIDE PROCESS:

SP TM-4325

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

V - 96

UNIVAC FILE-COMPUTER SYSTEM

FUNCTION CONTROL CHART

APPLICATION: ____________ PROGRAM NO.: ____________

DEMAND UNITS

NO.	TYPE OF UNIT	TEST IN FROM	TEST OUT TO NOT READY	TEST OUT TO READY	DEMAND IN FROM	DEMAND OUT TO INPUT	DEMAND OUT TO OUTPUT
1							
2							
3							
4	1	2	3	4	5	6	7
5							
6							
7							
8							

SCAN UNITS

NO.	TYPE OF UNIT	TO SCAN OUT	NO.	TYPE OF UNIT	TO SCAN OUT	NO.	TYPE OF UNIT	TO SCAN OUT
1			9			17		
2			10			18		
3			11			19		
4	8	9	12	8	9	20	8	9
5			13			21		
6			14			22		
7			15			23		
8			16			24		

CODE DISTRIBUTOR

FROM V1 OF STEPS	47
FROM V2 OF STEPS	48
FROM R OF STEPS	49

NO.	TO	NO.	TO
00		26	
01		27	
02		28	
03		29	
04		30	
05		31	
06		32	
07		33	
08		34	
09		35	
10		36	
11	50	37	50
12		38	
13		39	
14		40	
15		41	
16		42	
17		43	
18		44	
19		45	
20		46	
21		47	
22		48	
23		49	
24		>49	
25			

FROM OUT OF FUNCTION: 51

NO.	TO	NO.	TO
0		5	
1		6	
2	52	7	52
3		8	
4		9	

INPUT CONTROL LINES

NO.	TO
a	
b	
c	
d	
e	
f	10
g	
h	
i	
i	
k	
l	

OUTPUT CONTROL LINES

NO.	FROM
A	
B	
C	
D	
E	11
F	
G	
H	
I	
J	

START TO	12

SCAN OUTS

NO.	TO
	13

ERROR CHECKS

FROM	TO
ARITH.	
±O'FLOW	
÷O'FLOW	14
PARITY	
ADDRESS	

INDICATORS

NO.	FROM
1	
2	15
3	
4	

IND. SWITCH	FROM 16	TO 17	
NO CHECK	18		

READ UNIT RECORD

NO.	IN FROM	OUT TO
1		
2		
3		
4	19	20
5		
6		
7		
8		

WRITE UNIT RECORD

NO.	IN FROM	OUT TO
1		
2		
3		
4	21	22
5		
6		
7		
8		

STEP CLEAR

IN	OUT
23	24
CLEAR PROG. SEL.	25

STEP REPEAT

IN FROM	26

CHANNEL SEARCH EQUAL

NO.	FROM OUT OF STEP NO.	TO IF = V1	TO IF = V2	TO IF ≠
1				
2				
3	27	28	29	30
4				

BRANCHING

NO.	IN FROM STEP	TO IF +	TO IF −	TO IF O
1				
2				
3				
4				
5				
6				
7				
8	34	35	36	37
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				

CHANNEL SEARCH UNEQUAL

NO.	FROM OUT OF STEP NO.	TO IF =	TO IF ≠
1			
2			
3	31	32	33
4			

OUT EXPANDERS

NO.	IN	OUT	OUT
1			
2			
3			
4	38	39	40
5			
6			
7			
8			

FUNCTION DELAYS

NO.	IN 1	IN 2	OUT
1			
2	41	42	43
3			

ALTERNATE SWITCHES

NO.	SELECT	COMMON	NON-SELECT
1			
2			
3	45	44	46
4			
5			
6			

FUNCTION SEQUENCE

NO.	SET	PROBE	OUT
1	53	54	55
2			

SYMBOLS FOR PROGRAMMING:

Processes:	PR
Transfer	T
Addition	+
Subtraction	–
Multiplication	X
Division	÷
Left Zero Elim	LZE
Compare	C
Channel Search =	ECS
Channel Search ≠	UCS

SHIFTS:

Shift 2 Left	2L
Shift 2 Right	2R
Expand Prod. Shift	EXS

SELECTORS:

Computor Selector	$T_1 \rightarrow T_{48}$
Drum Selector	$A_1 \rightarrow A_8$
Input-Output Selectors	$S_1 \rightarrow S_{10}$
Common	CT(#) or CA (#)
Select	ST (#) or SA (#)
Non-Select	NST(#) or NSA (#)
Pick-Up	PUT(#) or PUA (#)
Program Selector	$PS_1 \rightarrow PS_{16}$
Delay of PS	$DPS_1 \rightarrow DPS_{10}$
Imm. of PS	$IPS_{11} \rightarrow IPS_{16}$
Power of PS	$PSO_1 \rightarrow PSO_{16}$
Drop Out of PS	$PSDO_1 \rightarrow PSDO_{16}$
Ground of Sel	GT (#)
Computer Ground	CG
Demand Ground	$DG_1 \rightarrow DG_8$
Scan Ground	$SG_1 \rightarrow SG_6$
Selector Hold	SH

Function Delay	FD
In FD 1	$FD1_1 \rightarrow FD1_3$
In FD 2	$FD2_1 \rightarrow FD2_3$
Out FD	$FDO_1 \rightarrow FDO_3$

STORAGES:

Input-Output	000 → 009
Intermediate	10 → 59
Revolver	$F_0 \rightarrow F_{19}$
Channel Search Fields	$CSF_0 \rightarrow CSF_{19}$

DEMAND UNITS:

Test In	TST (#)
Test Out Ready	TSTR (#)
Test Out Not Ready	TSTN (#)
Demand In	DI (#)
Demand Out Input	DOI (#)
Demand Out Output	DOO (#)

Scan Out	$SO_1 \rightarrow SO_6$
Program Step In	In ST (#)
Program Step Out	From ST (#)
Out Expanders	$OE_1 \rightarrow OE_8$
In Branching	$BR_1 \rightarrow BR_{18}$
+ Branch Out	$+BR_1 \rightarrow BR_{18}$
– Branch Out	$-BR_1 \rightarrow BR_{18}$
o Branch Out	$oBR_1 \rightarrow BR_{18}$
Read Unit Record	$R_1 \rightarrow R_8$
Write Unit Record	$W_1 \rightarrow W_8$
Equal Channel Search In	ECSI (#)
Equal Channel Search Out=V_1	$ECSV_1$ (#)
Equal Channel Search Out=V_2	$ECSV_2$ (#)
Equal Channel Search Out ≠	ECS ≠ (#)
Unequal Channel Search In	UCSI (#)
Unequal Channel Search Out =	UCS = (#)
Unequal Channel Search Out ≠	UCS ≠ (#)
Program Address Counter	PAK
Address Register	ADR

Code Distributor	CD
In CD V_1	CDV_1
In CD V_2	CDV_2
In CD R	CDR
Out CD 00 to > 49	$CD_{00} \rightarrow CD > 49$
In CD Pulse In	CDP
Out CD Pulse Out	CDP_0--CDP_9
Function Sequence Set	$FSS_1 \rightarrow FSS_2$
Function Sequence Probe	$FSP_1 \rightarrow FSP_2$
Function Sequence Out	$FSO_1 \rightarrow FSO_2$

ALTERNATE SWITCHES:

Common	$SW_1C \rightarrow SW_6C$
Select	$SW_1S \rightarrow SW_6S$
Non-Select	$SW_1NS \rightarrow SW_6N_6$

V-96

SP TM-4327

FUNCTION CONTROL CHART

EXPLANATION KEY

1. Enter type of unit, i.e., 150 CPM, Magnetic Tape, etc.

2. TST - Enter source of pulse, i.e., start, step exit, etc.

3. TSTN - Enter step number or function to be initiated if unit is not ready.

4. TSTR - Enter step number or function to be initiated if unit is ready.

5. DI - Enter source of pulse, i.e., start, step exit, etc.

6. DOI - Enter step number or function to be initiated if last operation was the transfer of data to input.

7. DOO - Enter step number or function to be initiated if last operation was the transfer of data to output.

8. Enter type of unit.

9. Enter the number of the scan out routine assigned to the unit.

10. IPC - Enter the number of the selector(s) to be picked up, i.e., PU T12, PU A8, etc.

11. OPC - Enter the number of the step exit or other function which impulses the output control line.

12. Enter the step number or function to be initiated by the start impulse.

13. Enter step number or function to be initiated by the scan out impulse.

14. Enter the function to be initiated if the related error occurs.

15. Enter source of pulse, usually B+ through a selector, i.e., ST#2, etc.

16. Enter source of pulse, usually B+.

17. Enter the destination of the pulse routed through the indicator switch, i.e., CT#2, etc.

18. Enter an x if no check is to be wired.

19. R - Enter the step exit or out of function which is to initiate the reading of a specific URA from the drum to the revolver.

20. Enter the step number or other function to be initiated immediately following the read instruction.

21. W - Enter the step exit or out of function which is to initiate the writing from the revolver to the drum.

22. Enter the step number or other function to be initiated immediately following the write instruction.

23. Enter the error check exit(s) which are to initiate step clear.

24. Enter the step number or function to be initiated after step clear, i.e., step 24, PS 5, etc.

25. Enter source of pulse, i.e., out of step 36, etc.

26. Enter the error check exit(s) which are to initiate step repeat.

27. ECSI - Enter the step number of the channel search equal step, i.e., out of step 4, etc.

28. $ECSV_1$ - Enter the step number or function to be initiated if a match was found for value 1.

29. $ECSV_2$ - Enter the step number or function to be initiated if a match was found for value 2.

30. ECS≠ - Enter the step number or function to be initiated if no match was found.

31. UCSI - Enter the step number of the channel search unequal step.

32. UCS= - Enter the step number or function to be initiated if all URA's matched.

33. UCS ≠ - Enter the step number or function to be initiated if a non-matching URA was found.

34. BR - Enter the number of the step exit whose result is to be tested for +, - or O.

35. +BR - Enter the step number or function to be initiated if the result is +.

36. -BR - Enter the step number or function to be initiated if the result is -.

37. OBR - Enter the step number or function to be initiated if the result is O.

38. OE - Enter the number of the step exit or function exit which is to be expanded.

39. Enter the step number or function to be initiated, (maximum of two).

40. Enter the step number or function to be initiated, (maximum of two).

41. FD1 - Enter the number of the step exit or function exit.

42. FD2 - Enter the number of the step exit or function exit.

43. FDO - Enter the step number or function to be initiated after both FD1 and FD2 have been impulsed.

44. SW C - Enter the step number, function, value, etc. which is to be selected by the manual alternate switch.

45. SW S - Enter the step number, function, value, etc. which is to be chosen if the alternate switch is on the select side.

46. SW NS - Enter the step number, function, value, etc. which is to be chosen if the alternate switch is on the non select side.

47. CD V_1 - Enter the step number(s) whose value 1 is to be chosen through the code distributor.

48. CD V_2 - Enter the step number(s) whose value 2 is to be chosen through the code distributor.

49. CD R - Enter the step number(s) whose result is to be chosen through the code distributor.

50. CD 00 through 49 - Enter the address of the storage unit to be called for if the related value is in the code distributor.

51. CD P - Enter the number of the step exit or function.

52. CDP 0 through 9 - Enter the step number or function to be initiated if the related value is in the code distributor.

53. FSS - Enter the number of the step exit or function exit.

54. FSP - Enter the number of the step exit or function exit.

55. FSO - Enter the step number or function to be initiated after the FSS and then the FSP have been impulsed.

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

V - 100

UNIVAC FILE-COMPUTER SYSTEM

COMPUTER SELECTOR CONTROL CHART

APPLICATION:__________ PROGRAM NO. __________

PICK-UP FROM	GROUND TO	T#	SELECT	COMMON	NON-SELECT
		1			
		2			
		3a			
		3b			
		4a			
		4b			
		5a			
		5b			
		6a			
		6b			
		6c			
		6d			
		7			
		8			
		9a			
		9b			
		10a			
		10b			
		11a			
		11b			
		12a			
		12b			
		12c			
		12d			
		13			
		14			
		15a			
		15b			
		16a			
		16b			
		17a			
		17b			
		18a			
		18b			
		18c			
		18d			
		19			
		20			
		21a			
		21b			
		22a			
		22b			
		23a			
		23b			
		24a			
		24b			
		24c			
		24d			

PICK-UP FROM	GROUND TO	T#	SELECT	COMMON	NON-SELECT
		25			
		26			
		27a			
		27b			
		28a			
		28b			
		29a			
		29b			
		30a			
		30b			
		30c			
		30d			
		31			
		32			
		33a			
		33b			
		34a			
		34b			
		35a			
		35b			
		36a			
		36b			
		36c			
		36d			
		37			
		38			
		39a			
		39b			
		40a			
		40b			
		42b			
		42c			
		42d			
		43			
		47a			
		47b			
		48a			
		48b			
		48c			
		48d			

PROGRAM SELECTS

PS#	IN FROM	DELAY OUT TO	D. O. FROM	POWER TO
1				
2				
3				
4				
8				
9				
10				
PS#	**IN FROM**	**IMMEDIATE OUT TO**	**D. O. FROM**	**POWER TO**
11				
12				
13				
14				
15				
16				

REMARKS:

ENTER THE NUMBER OF THE INPUT CONTROL LINE OR PROGRAM SELECTOR WHICH IS TO IMPULSE THE PICK UP, I.E., IPC a, PSO 12, ETC.

ENTER THE TYPE OF GROUND I.E., SCAN GROUND #1, DEMAND GROUND #4, COMPUTER GROUND, ETC.

ENTER THE ADDRESS OF THE VALUE, THE STEP NUMBER, THE FUNCTION, ETC. DESIRED WHEN THE SELECTOR IS ON THE SELECT SIDE.

ENTER THE VALUE, RESULT, STEP NUMBER, FUNCTION, ETC. THAT IS TO BE SELECTED.

ENTER THE ADDRESS OF THE VALUE, THE STEP NUMBER, THE FUNCTION, ETC. DESIRED WHEN THE SELECTOR IS ON THE NON-SELECT SIDE.

ENTER THE NUMBER OF THE STEP EXIT OR FUNCTION WHICH IMPULSES THE PS IN.

ENTER STEP NUMBER OR FUNCTION TO BE INITIATED. NOTE, PS 1 IS A 20 ms. DELAYED OUT; PS 2 THROUGH 10 ARE 15 ms. DELAYED OUTS; PS 11 THROUGH PS 16 ARE IMMEDIATE OUTS.

ENTER SELECTOR(S) TO BE PICKED UP, I.E., PU T8, PU A6, ETC.

ENTER THE NUMBER OF THE STEP EXIT OR FUNCTION WHICH IMPULSES THE DROP OUT.

SECTION VI

ILLUSTRATIVE PROBLEMS AND SOLUTIONS

ILLUSTRATIVE PROBLEMS AND SOLUTIONS

INDEX - SECTION VI

1. Problem Approach

Every problem applied to a Univac File-Computer consists of several areas of interest. In other words, all the aspects of the system must be controlled through the program by analysis of the problem. These areas of interest or break-up of the overall problem can be stated as:

1) Analyze the problem.

   a) What end results are required?
   b) When and on what schedule must the results be obtained?
   c) What is the source data?
   d) What is the source media?
   e) What reference file information is needed?
   f) What calculations are necessary to transform the source and reference data into the desired end results.

2) Specify the System Required.

   After the problem has been analyzed and the questions mentioned above are answered the following questions should be considered:

   a) What type of input devices are required?
   b) Is more than one input device needed?
   c) What information is to be stored on drums?
   d) Is external tape storage needed?
   e) Is stored instruction programming needed?
   Note: This question can be better answered after the problem has been flow charted.

   Note: After specifying a system in this way, subsequent programming may require a change in the original specifications. However, it is necessary to have a general idea of what equipment is needed before starting the flow programming.

3) After completing 2) and preparing tentative machine specifications the entire application should be analyzed in detail. This may be best accomplished by:

   a) Referring to the information obtained when analyzing the problems, prepare a rough flow chart indicating what runs are to be made by the computer.

After these general flow charts are completed, a schedule of computer runs such as the following might be prepared:

UNIVAC FILE COMPUTER – APPLICATION SCHEDULE

PRO. #	DESCRIPTION	VOLUME	MODE		INPUT MEDIA	INPUT-OUTPUT REQUIRED	FREQUENCY	ELAPSED TIME	INPUT REC'D	OUTPUT DUE	OUTPUT MEDIA	URA'S
			ON	OFF								
1	Payroll & Labor Distribution	20M		✓	Card Forms #A	1- 150CPH 90-COL.	Daily	2½hrs	3:00PM	Daily	Card Form A	I&II
2	Net Pay Run	8M		✓	Card Forms B&C	"	Weekly	1 hr	Tue. 9AM	Tue 12 Noon	Card Forms B&C	I

The card forms and unit record areas referred to would, of course, be tentatively designed through the operations described in 1) and 2).

It is obvious that this format of scheduling computer runs would not be necessary where only one or two runs are required; however, in more complex applications involving many runs of various frequency such a schedule provides an excellent check point for scheduling the work in the best fashion. The elapsed time column must necessarily remain blank until the next step of the procedure is completed.

4) Program Flow Charting.

After we have completed the preceding steps the following information should be available for each scheduled computer run:

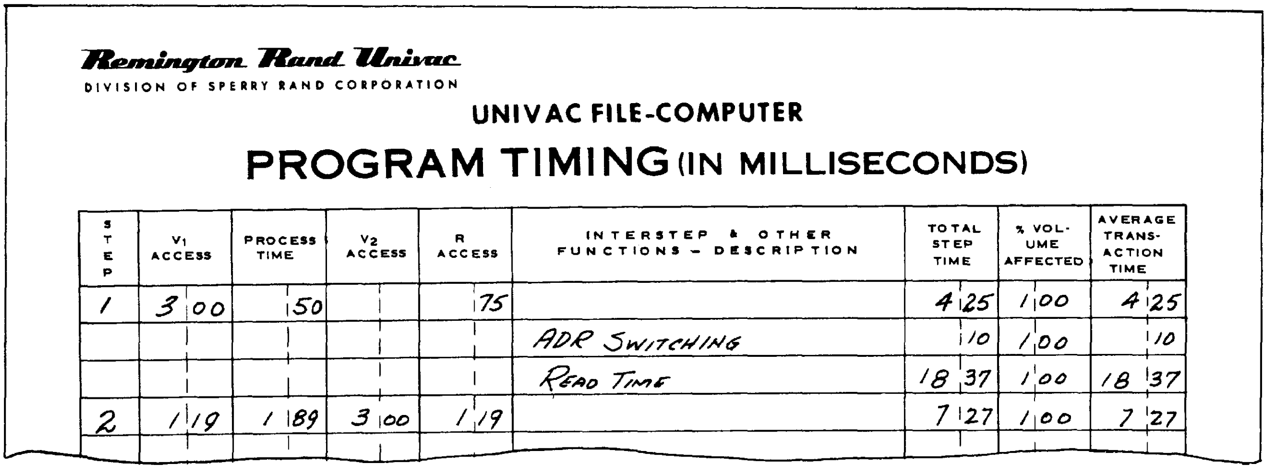
a) Input media and data description.
b) Output media and data description.
c) Reference or drum URA data description.
d) Calculation formulae necessary to transform the input data to the output or reference results.

Having this information the next step is to flow chart the necessary program.

The flow chart breaks down the step by step operation of the computer. This type of chart is an invaluable aid to the programmer since it exposes all the potential flaws in his logic. It forces him to think logically, in the manner in which the computer will operate. It must be remembered that the computer does not "think" by itself; it merely performs mathematical processing of data in a logical fashion as set forth by the programmer. Without such charts it is easy for the programmer to lose the thread of alternatives which are presented to the machine, and thereby miss important steps in the logical sequence.

It is not necessary to be mechanically technical in these charts. It is better to prepare it in everyday terminology. Writing the actual program after having charted the steps in this logical fashion is simple and virtually all errors can be avoided.

5) After the program flow chart has been completed it is possible to proceed to the next step of "timing" the program. Form SPTM-4329 is used for this purpose.



Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER

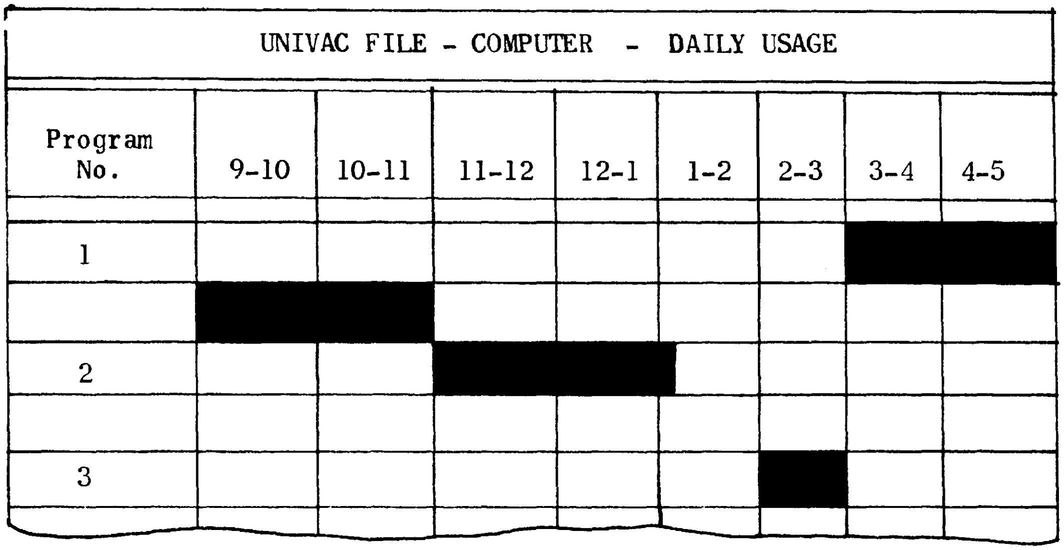
PROGRAM TIMING (IN MILLISECONDS)

STEP	V1 ACCESS	PROCESS TIME	V2 ACCESS	R ACCESS	INTERSTEP & OTHER FUNCTIONS – DESCRIPTION	TOTAL STEP TIME	% VOLUME AFFECTED	AVERAGE TRANSACTION TIME
1	3 00	50		75		4 25	1 00	4 25
					ADR SWITCHING	10	1 00	10
					READ TIME	18 37	1 00	18 37
2	1 19	1 89	3 00	1 19		7 27	1 00	7 27

The entries made refer to Problem I calculations in part. Symbols should be entered in the description column to cross-reference the volume % figure used for the calculations performed to obtain the result. This will also prove of value when changing or altering a program. The % volume figure should also include provision for iterative processes where a step is performed more than once for the routine.

The various program step times can be obtained from the time summary section of this manual, however, care should be exercised in accurately applying these times since program techniques of simultaneous operations often eliminate process times from the total elapsed time total. The timing is done directly from the program flow chart and should include all interstep functions (ADR switching, Read, Write etc.) as well as the step process and access times. Note that the total times computed in this step should be entered on the application schedule.

6) After the program flow charts and their related timing charts have been completed it is possible to chart the computer usage against available time to determine how the program sequence can be best arranged in light of input receipt time and desired output time in order to produce the most satisfactory results. This is done most easily by means of a Gantt chart such as:



The above elapsed time usage report would indicate the following daily schedule:

	Starts	Ends	Total Time
Run 1	3:00 PM	11:00 AM	4 hours
Run 2	11:00 AM	1:20 PM	2 hours 20 min.
Run 3	2:00 PM	3:00 PM	1 hour

The preparation of this chart enables us to get a clear picture of the time end results will be available, and also forces us to time our preparatory work to fit in with the scheduled computer usage.

7) The information provided by the preceding six (6) steps is sufficient to allow us to determine the feasibility of our applications, and in addition it gives us an excellent basis for creation of the detailed program.

As mentioned before, if we are complete enough in preparing our program flow chart we will have little or no difficulty in transforming the flow chart to a detailed program. Complete examples of this entire process are shown in the examples covered in this section.

Illustrative Problems:

The following sample problems and their solutions are designed to be used as practice problems in programming the Univac File-Computer. They are presented in a sequence of ascending complexity, and each problem employs one or more machine features or program techniques not utilized in the preceding problems.

Each problem consists of 7 parts, these are:

1) Statement of problem
2) Machine specifications and comments
3) General procedure flow chart
4) Input-Output data format and explanation
5) Drum stored data
6) Program flow chart and comments
7) Program planning sheets

A list of the training problems is as follows:

I Billing and Inventory Control,

An off-line, punched card, single input device problem.

II Billing and Inventory Control,

An off-line, punched card, single input device problem, including back-ordered item control.

III Multi-Card Inventory and Sales Problem.

An off-line, single input device, punched card problem.

IV Incentive Payroll - Daily Job Card Calculation and Summarization by Employee.

Involves two types of URA's (operation standards, and employee)

V Weekly Payroll,

Gross to net calculations.

2. PROBLEM I

1. Statement of Problem

This problem involves the continuous daily processing of approximately 7500 order item cards. These cards are produced from the customers order by means of a Card-O-Matic punch and are then placed into the UFC operation in order receipt sequence. As far as the computing operation is concerned the only significant data in these cards is the quantity ordered and the item number. The item number is so designed as to be the complete drum address of the item's URA. There are approximately 10,000 inventoried items. The operations to be performed by the computer are:

a) Reduction of the drum balance on hand figure for the item by the quantity ordered. (For purposes of simplicity we will assume that the balance on hand will never become negative. We will also assume that every order can be shipped i.e. - No backorder conditions will arise.)

b) Quantity ordered is to be multiplied by the sales price obtained from the item URA and the result, sales amount, is to be rounded with .005 and punched into the card.

c) Quantity ordered is to be multiplied by the cost price obtained from the item URA and the result, cost amount, is to be rounded with .005 and punched into the card.

It should be noted that normally an application such as this could also include customer name cards which would be used with the computed item cards to produce the invoice. Consideration of these name cards has not been included in this problem in the interests of simplicity.

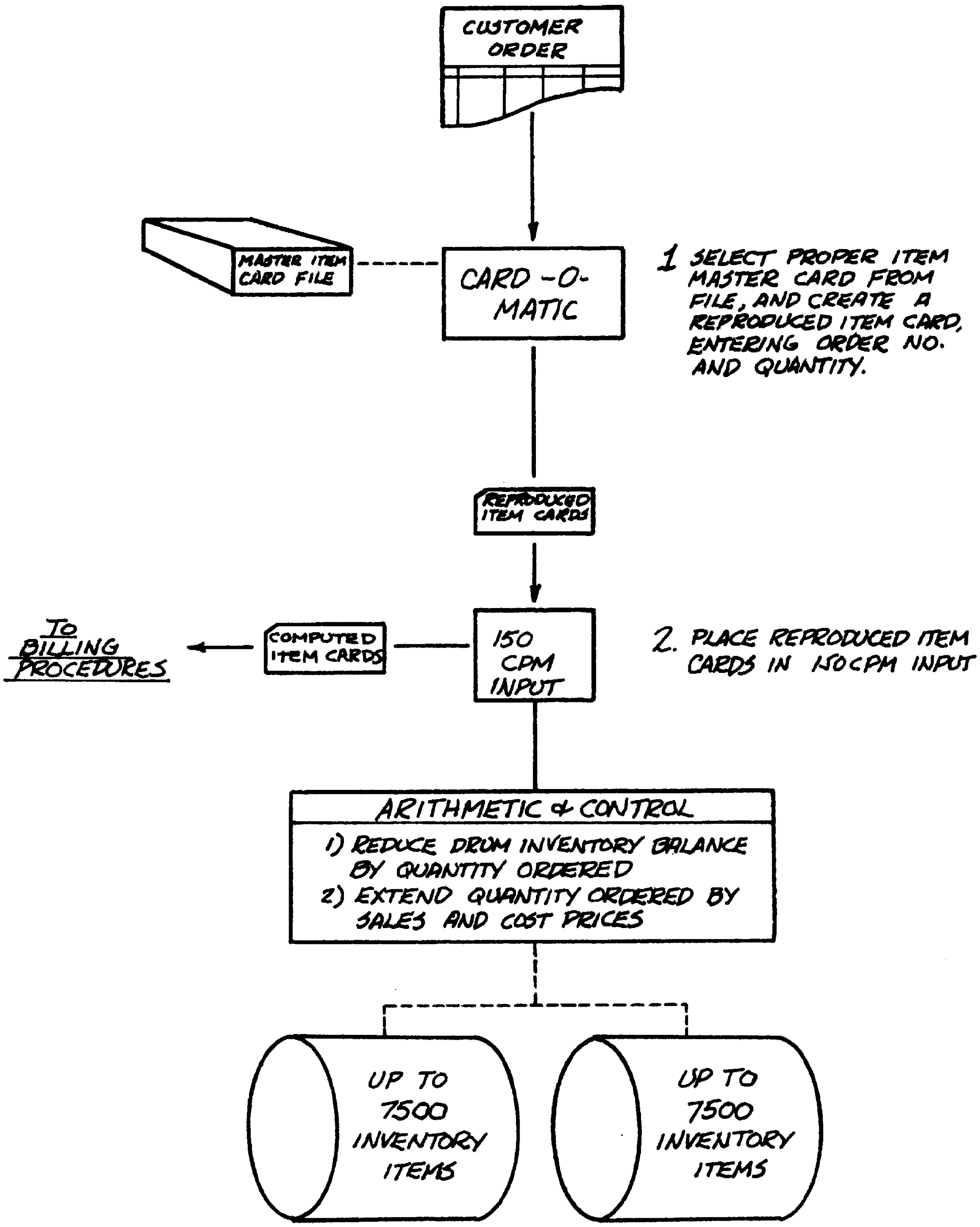
2. Machine Specifications

1 - 150 CPM Input-Output Card Sensing Punching Unit

2 - Large Capacity Drums with 24 digit URA length.

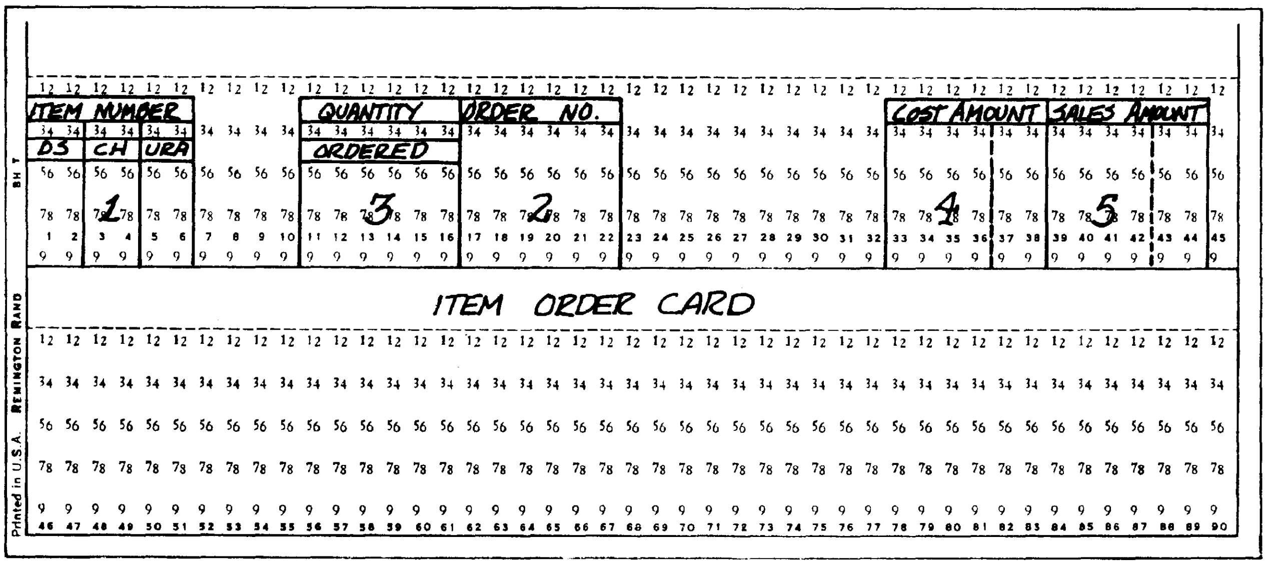
1 - Arithmetic and Control Unit (External Program)

3. General Procedure Flow Chart


CUSTOMER ORDER
MASTER ITEM CARD FILE
CARD-O-MATIC
1 SELECT PROPER ITEM MASTER CARD FROM FILE, AND CREATE A REPRODUCED ITEM CARD, ENTERING ORDER NO. AND QUANTITY.
REPRODUCED ITEM CARDS
TO BILLING PROCEDURES
COMPUTED ITEM CARDS
150 CPM INPUT
2. PLACE REPRODUCED ITEM CARDS IN 150CPM INPUT
ARITHMETIC & CONTROL
1) REDUCE DRUM INVENTORY BALANCE BY QUANTITY ORDERED
2) EXTEND QUANTITY ORDERED BY SALES AND COST PRICES
UP TO 7500 INVENTORY ITEMS
UP TO 7500 INVENTORY ITEMS

4. Input-Output Data and Definition

The input source media are punched cards which appear as follows:



Definitions:

Input

1. Item Number - This is the inventory number for the item ordered. It is also the actual large capacity drum address of the item's stored data.

2. Order number - This is the customers order number. This information is not required in the calculation.

3. Quantity Ordered - This field contains the quantity of the item ordered by the customer.

Output

4. Cost Amount - This field is the output resulting from multiplying quantity ordered by the drum stored cost price.

5. Sales Amount - This field is the output resulting from multiplying quantity ordered by the drum stored sales price.

The above input and output data will be assigned to input-output storage as follows:

INPUT - 000 - Item Number (Drum Address)
002 - Quantity Ordered

OUTPUT - 003 - Cost Amount
004 - Sales Amount

Note: The two fields of input are assigned to even input storage units so that the "even" input transfer may be used. This will transfer the data to input storage in one drum revolution.

5. Drum Stored Data and Definition

a) Drum Unit Records for 10,000 items, Drum locations 000000 - 039924.

UNIT RECORD AREA - FIELD ASSIGNMENT

SYM.	DESCRIPTION	DIGITS	SIGN
F0	Item Number	xxxxxx.	+
F1	Balance on Hand	xxxxxxx.	+
F2	Cost Price	xx.xxx	+
F3	Selling Price	xx.xxx	+
F4	Constant 5	.005	+

b) Definitions

F0 - Item Number - Actual drum address of the inventory item. Since the input media contains the item address there is no need for this information in this program.

F1 - Balance On Hand - This represents the quantity of the item on hand and available for shipment. The statement of the problem precludes the possibility of this balance ever becoming negative.

F2 - Cost Price - Cost price per each item sold.

F3 - Selling Price - Sales price per each item sold.

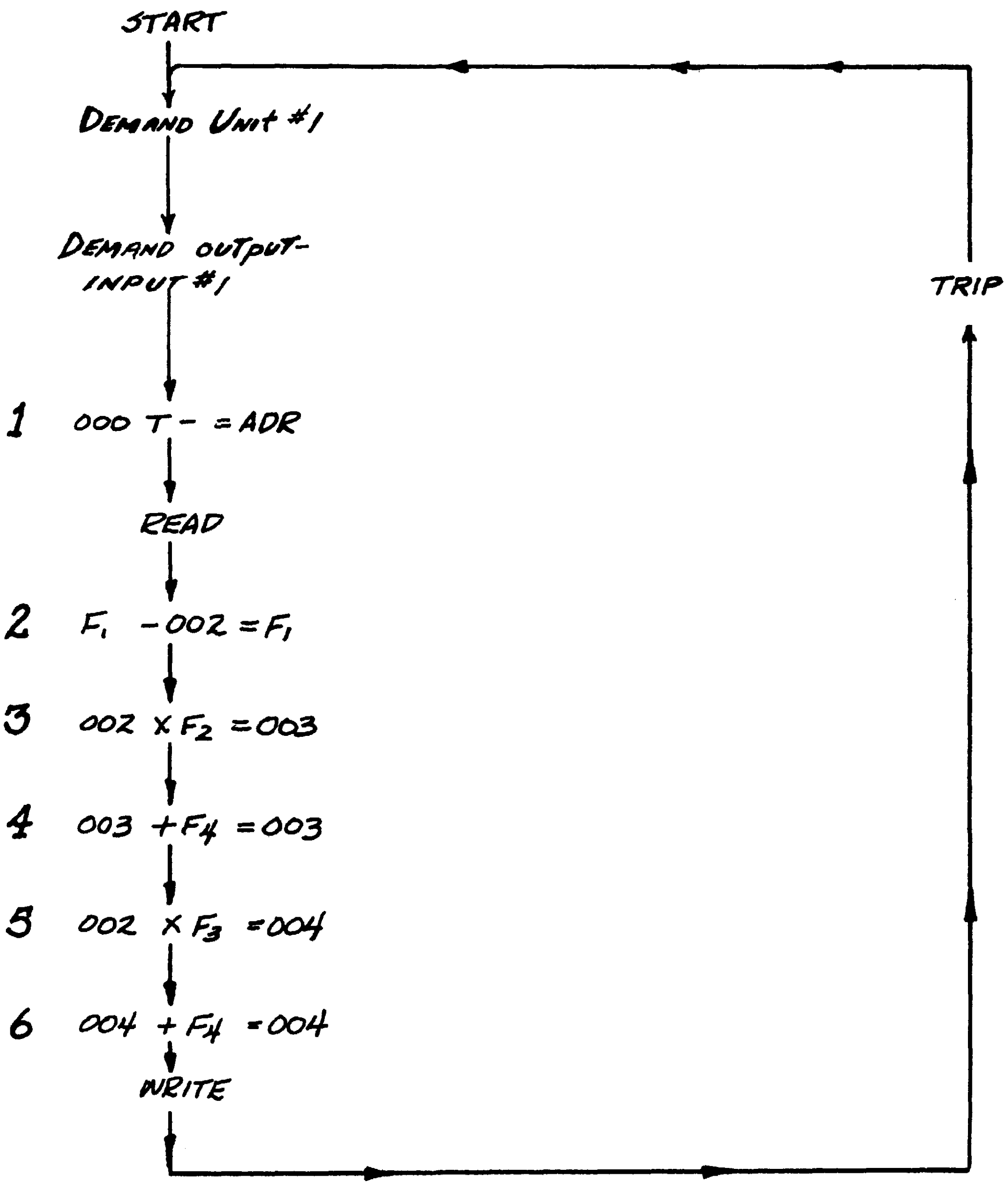
F4 - Constant 5 - Since the problem requires rounding of the cost and sales amounts to the nearest penny we need this constant. We have placed the constant in the unit record area for two reasons:

1) We have the necessary space to include this digit in the 24 digit URA chosen.

2) Access time to this URA field will be faster than to an input-output or intermediate storage.

For purposes of this problem, assume the above data has been loaded onto the drums.

6. PROGRAM FLOW CHART



Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM

150 CPM INPUT-OUTPUT CHART

APPLICATION: BILLING & INVENTORY PROGRAM NO.: I

INPUT-OUTPUT STORAGE FIELD ASSIGNMENT

DEMAND UNIT NO. 1 OR SCAN UNIT NO. ____

INPUT FIELD ASSIGNMENT

SYM.	DESCRIPTION	CARD COLUMNS AND DECIMALS 11	10	9	8	7	6	5	4	3	2	1	SN.	SIGN CTL.
000	ITEM NUMBER (DRUM ADDRESS)						1	2	3	4	5	6	+	
001														
002	QUANTITY ORDERED						11	12	13	14	15	16	+	
003														
004														
005														
006														
007														
008														
009														

OUTPUT FIELD ASSIGNMENT

SYM.	DESCRIPTION	CARD COLUMNS AND DECIMALS 11	10	9	8	7	6	5	4	3	2	1	SN.	SIGN CTL.
000														
001														
002														
003	COST AMOUNT					33	34	35	36	37	38		+	
004	SALES AMOUNT					39	40	41	42	43	44		+	
005														
006														
007														
008														
009														

INPUT CONTROL LINES

SYM.	FROM
a	
b	
c	
d	
e	
f	
g	
h	
i	
j	
k	
l	

OUTPUT CONTROL LINES

SYM.	TO
A	TRIP
B	
C	
D	
E	
F	
G	
H	
I	
J	

INPUT-OUTPUT SELECTORS

PICK UP	S#	SELECT	COMMON	NON-SELECT
	1a			
	1b			
	1c			
	2a			
	2b			
	2c			
	3a			
	3b			
	3c			
	4			
	5a			
	5b			
	5c			
	6a			
	6b			
	6c			
	7a			
	7b			
	7c			
	8a			
	8b			
	8c			
	9a			
	9b			
	9c			
	10a			
	10b			
	10c			

INPUT TRANSFER

IN (from)	EVEN
NO (to)	
EVEN (to)	IN
ALL (to)	

PROGRAM SELECTS

NO.	IN FROM	POWER OUT TO
1		
2		

REMARKS:

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM

STORAGE ASSIGNMENT CHART

APPLICATION: BILLING & INVENTORY PROGRAM NO.: I

LARGE CAPACITY DRUM FIELD ASSIGNMENT

ADDRESS FROM 00 00 00
TO 039924

SYM	DESCRIPTION	CHARACTER AND DECIMAL											
		11	10	9	8	7	6	5	4	3	2	1	SN
F 0	ITEM NUMBER						5	4	3	2	1	0	+
F 1	BALANCE ON HAND					12	11	10	9	8	7	6	+
F 2	COST PRICE							17	16	15	14	13	+
F 3	SELLING PRICE							22	21	20	19	18	+
F 4	CONSTANT 5											23	+
F 5													
F 6													
F 7													
F 8													
F 9													
F10													
F11													
F12													
F13													
F14													
F15													
F16													
F17													
F18													
F19													

LARGE CAPACITY DRUM FIELD ASSIGNMENT

ADDRESS FROM ____________
TO ____________

SYM	DESCRIPTION	CHARACTER AND DECIMAL											
		11	10	9	8	7	6	5	4	3	2	1	SN
F 0													
F 1													
F 2													
F 3													
F 4													
F 5													
F 6													
F 7													
F 8													
F 9													
F10													
F11													
F12													
F13													
F14													
F15													
F16													
F17													
F18													
F19													

LARGE CAPACITY DRUM FIELD ASSIGNMENT SELECTORS

PICK-UP FROM	GROUND TO	A#	SELECT	COMMON	NON-SELECT
		1a			
		1b			
		1c			
		1d			
		2a			
		2b			
		2c			
		2d			
		3a			
		3b			
		3c			
		3d			
		4a			
		4b			
		4c			
		4d			
		5a			
		5b			
		5c			
		5d			
		6a			
		6b			
		6c			
		6d			
		7a			
		7b			
		7c			
		7d			
		8a			
		8b			
		8c			
		8d			

INTERMEDIATE STORAGE FIELD ASSIGNMENT

TRACK 50

SYM	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
50													
51													
52													
53													
54													
55													
56													
57													
58													
59													

INTERMEDIATE STORAGE FIELD ASSIGNMENT

TRACK 10

SYM.	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													

TRACK 20

20													
21													
22													
23													
24													
25													
26													
27													
28													
29													

TRACK 30

30													
31													
32													
33													
34													
35													
36													
37													
38													
39													

TRACK 40

40													
41													
42													
43													
44													
45													
46													
47													
48													
49													

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM

PROGRAM PLANNING CHART

APPLICATION: BILLING & INVENTORY PROGRAM NO.: I

STEP NO.	INPUT UNIT NO.	CARD NO.	VALUE 1			PROCESS	VALUE 2			RESULT			NEXT STEP
			DESCRIPTION	SYM	SH		DESCRIPTION	SYM	SH	DESCRIPTION	SYM	SH	
1	1	1	Item no. (DRUM ADDRESS)	000		T				Item No.	ADR		R1
2	1	1	Balance on Hand	F1		−	Quantity Ordered	002		NEW BALANCE ON HAND	F1		3
3	1	1	Quantity ORDERED	002		X	COST PRICE	F2		COST AMOUNT N.R.	003		4
4	1	1	COST AMOUNT N.R.	003		+	.005	F4		" " RDD.	003		5
5	1	1	Quantity ORDERED	002		X	SELLING PRICE	F3		SALES " N.R.	004		6
6	1	1	SALES AMOUNT N.R.	004		+	.005	F4		" " RDD.	004		W1
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													
41													
42													
43													
44													
45													
46													
47													
48													

VALUE 1 AS STORED												VALUE 2 AS STORED												UNSHIFTED RESULT																							STORED RESULT												STEP
11	10	9	8	7	6	5	4	3	2	1	SN	11	10	9	8	7	6	5	4	3	2	1	SN	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	SN	11	10	9	8	7	6	5	4	3	2	1	SN	
					X	X	X	X	X	X	+																													X	X	X	X	X	X	+						X	X	X	X	X	X		1
				X	X	X	X	X	X	X	+						X	X	X	X	X	X	+																X	X	X	X	X	X	X	+					X	X	X	X	X	X	X	+	2
					X	X	X	X	X	X	+							X	X	X	X	X	+																X	X	X	X	X	X	X	+					X	X	X	X	X	X	X	+	3
				X	X	X	X	X	X	X	+											5	+																X	X	X	X	X	X	X	+					X	X	X	X	X	X	X	+	4
					X	X	X	X	X	X	+							X	X	X	X	X	+																X	X	X	X	X	X	X	+					X	X	X	X	X	X	X	+	5
				X	X	X	X	X	X	X	+											5	+																X	X	X	X	X	X	X	+					X	X	X	X	X	X	X	+	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
																																																										41
																																																										42
																																																										43
																																																										44
																																																										45
																																																										46
																																																										47
																																																										48

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM

FUNCTION CONTROL CHART

APPLICATION: Billing & Inventory PROGRAM NO.: I

DEMAND UNITS

NO.	TYPE OF UNIT	TEST IN FROM	TEST OUT TO NOT READY	TEST OUT TO READY	DEMAND IN FROM	DEMAND OUT TO INPUT	DEMAND OUT TO OUTPUT
1	150 CPM				START & OE_1	IN ST #1	
2							
3							
4							
5							
6							
7							
8							

SCAN UNITS

NO.	TYPE OF UNIT	TO SCAN OUT	NO.	TYPE OF UNIT	TO SCAN OUT	NO.	TYPE OF UNIT	TO SCAN OUT
1			9			17		
2			10			18		
3			11			19		
4			12			20		
5			13			21		
6			14			22		
7			15			23		
8			16			24		

CODE DISTRIBUTOR

FROM V1 OF STEPS	
FROM V2 OF STEPS	
FROM R OF STEPS	

NO.	TO	NO.	TO
00		26	
01		27	
02		28	
03		29	
04		30	
05		31	
06		32	
07		33	
08		34	
09		35	
10		36	
11		37	
12		38	
13		39	
14		40	
15		41	
16		42	
17		43	
18		44	
19		45	
20		46	
21		47	
22		48	
23		49	
24		>49	
25			

FROM OUT OF FUNCTION:

NO.	TO	NO.	TO
0		5	
1		6	
2		7	
3		8	
4		9	

INPUT CONTROL LINES

NO.	TO
a	
b	
c	
d	
e	
f	
g	
h	
i	
j	
k	
l	

OUTPUT CONTROL LINES

NO.	FROM
A	OE_1
B	
C	
D	
E	
F	
G	
H	
I	
J	

START TO	DI #1

SCAN OUTS

NO.	TO

ERROR CHECKS

FROM	TO
ARITH.	STEP REPEAT
±O'FLOW	STOP
÷O'FLOW	
PARITY	STOP
ADDRESS	STOP

INDICATORS

NO.	FROM	
1		
2		
3		
4		
IND. SWITCH	FROM	TO
NO CHECK		

READ UNIT RECORD

NO.	IN FROM	OUT TO
1	ST#1	ST#2
2		
3		
4		
5		
6		
7		
8		

WRITE UNIT RECORD

NO.	IN FROM	OUT TO
1	ST#6	OE_1
2		
3		
4		
5		
6		
7		
8		

STEP CLEAR

IN	OUT
CLEAR PROG. SEL.	

STEP REPEAT

IN FROM	ARITH. ERROR

CHANNEL SEARCH EQUAL

NO.	FROM OUT OF STEP NO.	TO IF = V1	TO IF = V2	TO IF ≠
1				
2				
3				
4				

BRANCHING

NO.	IN FROM STEP	TO IF +	TO IF −	TO IF O
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				

CHANNEL SEARCH UNEQUAL

NO.	FROM OUT OF STEP NO.	TO IF =	TO IF ≠
1			
2			
3			
4			

OUT EXPANDERS

NO.	IN	OUT	OUT
1	W_1	OPCA	DI #1
2			
3			
4			
5			
6			
7			
8			

FUNCTION DELAYS

NO.	IN 1	IN 2	OUT
1			
2			
3			

ALTERNATE SWITCHES

NO.	SELECT	COMMON	NON-SELECT
1			
2			
3			
4			
5			
6			

FUNCTION SEQUENCE

NO.	SET	PROBE	OUT
1			
2			

SYMBOLS FOR PROGRAMMING:

Processes: ....PR
Transfer........T
Addition........+
Subtraction ....—
Multiplication X
Division........÷
Left Zero Elim........LZE
Compare C
Channel Search = ...ECS
Channel Search ≠........UCS

SHIFTS:

Shift 2 Left........2L
Shift 2 Right........2R
Expand Prod. Shift........EXS

SELECTORS:

Computor Selector ....$T_1 \rightarrow T_{48}$
Drum Selector ....$A_1 \rightarrow A_8$
Input-Output Selectors........$S_1 \rightarrow S_{10}$
Common ....CT(#) or CA (#)
Select........ST (#) or SA (#)
Non-Select........NST(#) or NSA (#)
Pick-Up ....PUT(#) or PUA (#)
Program Selector ....$PS_1 \rightarrow PS_{16}$
Delay of PS ....$DPS_1 \rightarrow DPS_{10}$
Imm. of PS........$IPS_{11} \rightarrow IPS_{16}$
Power of PS ....$PSO_1 \rightarrow PSO_{16}$
Drop Out of PS ....$PSDO_1 \rightarrow PSDO_{16}$
Ground of Sel ....GT (#)
Computer Ground ....CG
Demand Ground........$DG_1 \rightarrow DG_8$
Scan Ground ....$SG_1 \rightarrow SG_6$
Selector Hold ....SH

Function Delay........FD
In FD 1 ....$FD1_1 \rightarrow FD1_3$
In FD 2 ....$FD2_1 \rightarrow FD2_3$
Out FD........$FDO_1 \rightarrow FDO_3$

STORAGES:

Input-Output ....000→009
Intermediate ....10→59
Revolver ....$F_0 \rightarrow F_{19}$
Channel Search Fields ....$CSF_0 \rightarrow CSF_{19}$

DEMAND UNITS:

Test In........TST (#)
Test Out Ready ....TSTR (#)
Test Out Not Ready........TSTN (#)
Demand In ....DI (#)
Demand Out Input........DOI (#)
Demand Out Output ....DOO (#)

Scan Out ....$SO_1 \rightarrow SO_6$
Program Step In ....In ST (#)
Program Step Out........ From ST (#)
Out Expanders ....$OE_1 \rightarrow OE_8$
In Branching ....$BR_1 \rightarrow BR_{18}$
+ Branch Out ....$+BR_1 \rightarrow BR_{18}$
– Branch Out ....$-BR_1 \rightarrow BR_{18}$
o Branch Out ....$oBR_1 \rightarrow BR_{18}$
Read Unit Record........$R_1 \rightarrow R_8$
Write Unit Record........$W_1 \rightarrow W_8$
Equal Channel Search In........ECSI (#)
Equal Channel Search Out=V_1 ..$ECSV_1$ (#)
Equal Channel Search Out=V_2 ..$ECSV_2$ (#)
Equal Channel Search Out ≠ ....ECS ≠ (#)
Unequal Channel Search In........UCSI (#)
Unequal Channel Search Out =..UCS = (#)
Unequal Channel Search Out ≠..UCS ≠ (#)
Program Address Counter ....PAK
Address Register ....ADR

Code Distributor........CD
In CD V_1 ....CDV_1
In CD V_2 ....CDV_2
In CD R........CDR
Out CD 00 to > 49........$CD_{00} \rightarrow CD > 49$
In CD Pulse In ....CDP
Out CD Pulse Out ....CDP_0--CDP_9

Function Sequence Set ....$FSS_1 \rightarrow FSS_2$
Function Sequence Probe........$FSP_1 \rightarrow FSP_2$
Function Sequence Out........$FSO_1 \rightarrow FSO_2$

ALTERNATE SWITCHES:

Common ....$SW_1C \rightarrow SW_6C$
Select........$SW_1S \rightarrow SW_6S$
Non-Select........$SW_1NS \rightarrow SW_6N_6$

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM

COMPUTER SELECTOR CONTROL CHART

APPLICATION: BILLING & INVENTORY PROGRAM NO. I
(NOT USED IN THIS PROBLEM)

PICK-UP FROM	GROUND TO	T#	SELECT	COMMON	NON-SELECT
		1			
		2			
		3a			
		3b			
		4a			
		4b			
		5a			
		5b			
		6a			
		6b			
		6c			
		6d			
		7			
		8			
		9a			
		9b			
		10a			
		10b			
		11a			
		11b			
		12a			
		12b			
		12c			
		12d			
		13			
		14			
		15a			
		15b			
		16a			
		16b			
		17a			
		17b			
		18a			
		18b			
		18c			
		18d			
		19			
		20			
		21a			
		21b			
		22a			
		22b			
		23a			
		23b			
		24a			
		24b			
		24c			
		24d			

PICK-UP FROM	GROUND TO	T#	SELECT	COMMON	NON-SELECT
		25			
		26			
		27a			
		27b			
		28a			
		28b			
		29a			
		29b			
		30a			
		30b			
		30c			
		30d			
		31			
		32			
		33a			
		33b			
		34a			
		34b			
		35a			
		35b			
		36a			
		36b			
		36c			
		36d			
		37			
		38			
		39a			
		39b			
		40a			
		40b			
		41a			
		41b			
		42a			
		42b			
		42c			
		42d			
		43			
		44			
		45a			
		45b			
		46a			
		46b			
		47a			
		47b			
		48a			
		48b			
		48c			
		48d			

PROGRAM SELECTS				
PS#	IN FROM	DELAY OUT TO	D. O. FROM	POWER TO
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
PS#	IN FROM	IMMEDIATE OUT TO	D. O. FROM	POWER TO
11				
12				
13				
14				
15				
16				

REMARKS:

3. PROBLEM II

1. Statement of Problem

This problem is a further extension of problem I in that consideration has been given to back-ordered quantities. The problem again involves continuous daily processing of 7500 order item cards against a drum stored inventory file of 10,000 items. The cards are created from the customers order by means of a Card-O-Matic punch and fed into the 150 CPM sensing punching unit of the UFC. The computer is to perform the following operations:

a) Reduce the quantity on hand (drum) by the quantity ordered. If the result is negative, then the result equals the quantity to be back-ordered. Punch the quantity back-ordered. In addition the quantity on hand should be changed to equal zero when a back-order occurs. The quantity shipped is determined by reducing the quantity ordered by the quantity back-ordered (if any).

b) Quantity shipped is to be extended by the cost and selling prices to obtain cost and sales amounts.

c) When a back-order occurs a one (1) should be added to a number of back-orders count maintained on the drum by item.

d) Gross profit is to be calculated and punched as well as added to the gross profit total on the drum. Gross profit equals sales amount less cost amount.

The item number is again the complete drum URA address of the item's unit record area. Assume that the cost price is never more than the sales price. Further conditions of the problem are:

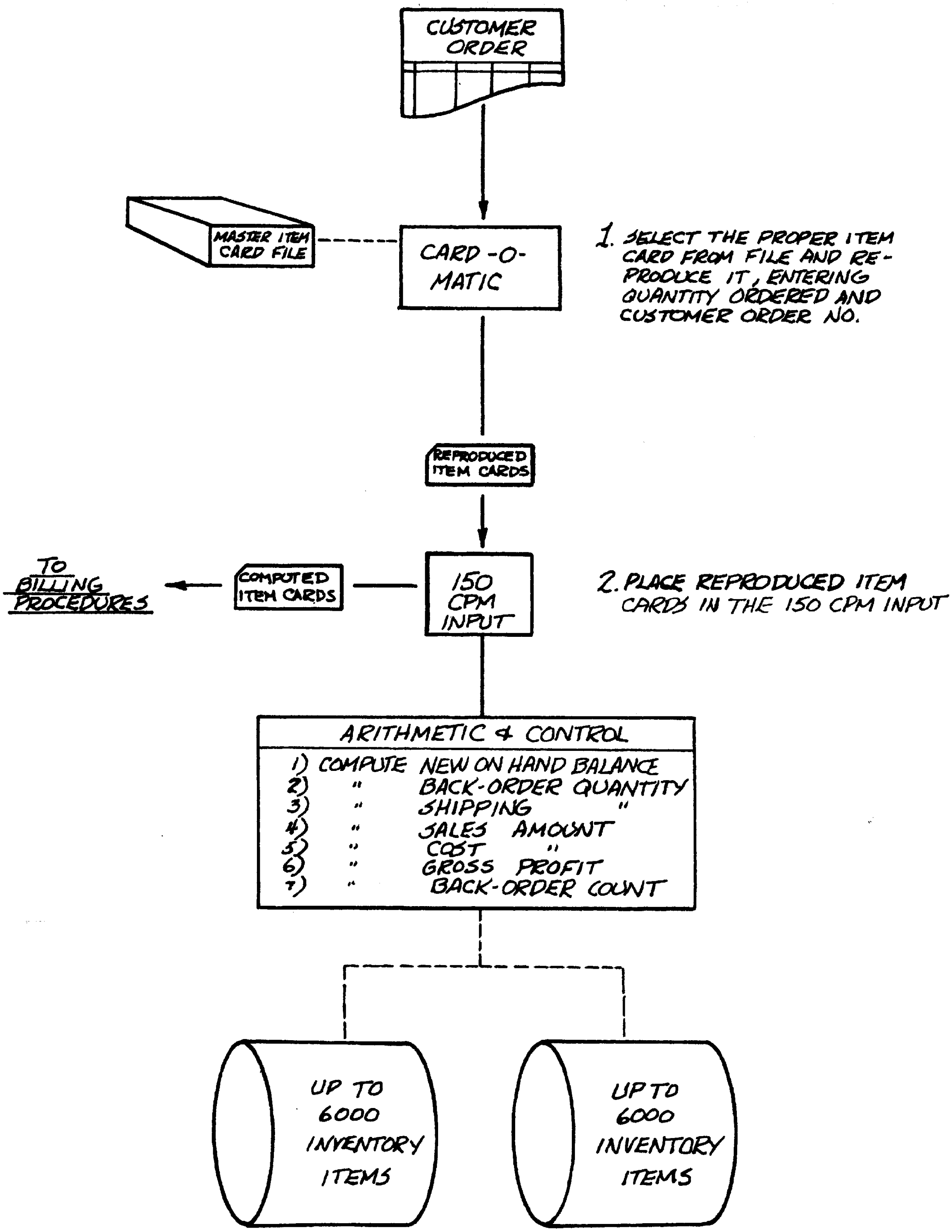
1) Assume the item unit record areas are already on the drum.

2) Assume all intermediate storages are clear (i.e. spaces).

2. Machine Specifications

1 - 150 CPM Input-Output Card Sensing Punching Unit
2 - Large Capacity Drums with a 30 digit URA length
1 - Arithmetic and Control Unit (External Program)

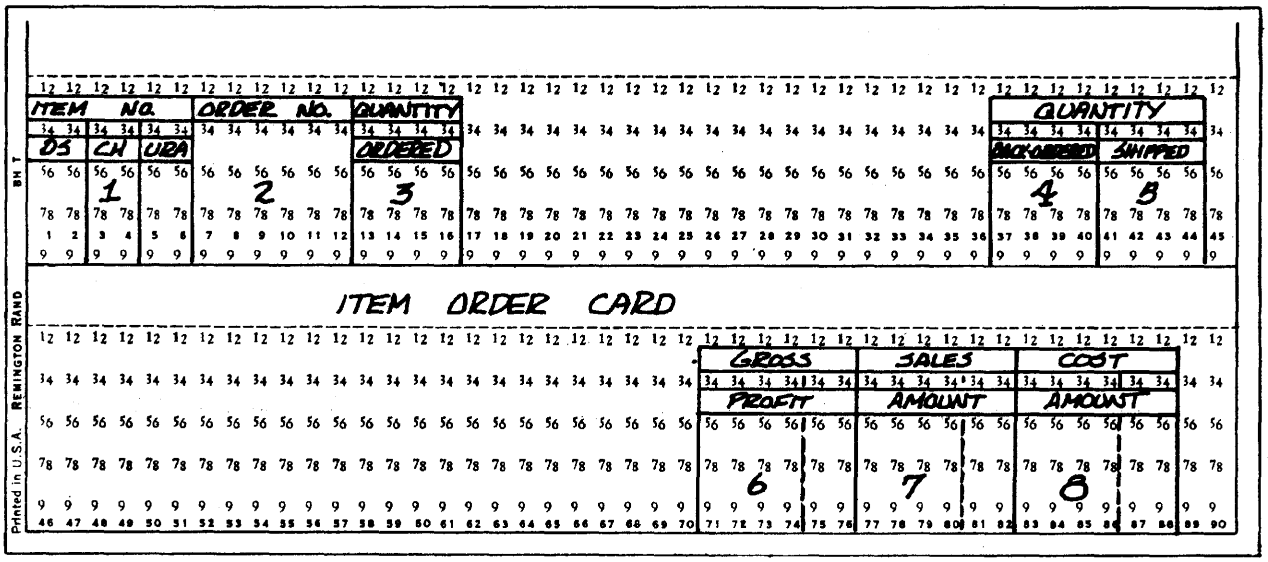
3. General Procedure Flow Chart



1. SELECT THE PROPER ITEM CARD FROM FILE AND REPRODUCE IT, ENTERING QUANTITY ORDERED AND CUSTOMER ORDER NO.

2. PLACE REPRODUCED ITEM CARDS IN THE 150 CPM INPUT

4. Input-Output Data and Definition



Data Definition

Input

1. Item Number - This is the inventory number for the item ordered. It is also the specific large capacity drum address for the item's URA.

2. Order Number - This is the customer's order number. This information is not required in the computations.

3. Quantity Ordered - This field contains the quantity of the item ordered by the customer

Output

4. Quantity Back Ordered - This field represents that portion of the quantity ordered which cannot be filled from the on hand balance.

5. Quantity Shipped - This field represents the portion of quantity ordered, which can be shipped from the balance on hand.

6. Gross Profit - The gross profit realized on the sale of this item: the difference between sales amount and cost amount.

7. Sales Amount - The dollar amount rounded to two (2) decimal places resulting from the extension of quantity shipped by sales price.

8. Cost Amount - Same as 7 except substitute cost price for sales price.

The above input and output data is assigned to input-output storage as follows:

INPUT	000	Item Number (Drum address)
	002	Quantity Ordered
OUTPUT	002	Quantity Shipped
	004	Quantity Back Ordered
	006	Cost Amount
	007	Gross Profit Amount
	008	Sales Amount

5. Drum Stored Data

a) Drum unit records for 10,000 items, Drum locations 000000-049919.

UNIT RECORD AREA - FIELD ASSIGNMENT

SYM.	DESCRIPTION	DIGITS	SIGN
F0	Sales Price	xx.xxx	+
F1	Cost Price	xx.xxx	+
F2	On Hand Balance	xxxxxx.	+
F3	Number of Back Orders	xxxx.	+
F4	Gross Profit to Date	xxxxxx.xx	+
F5	Constant 1	1.	+
F6	Constant 5	.005	+

b) Definitions

F0 - Sales Price - Sales Price for each unit sold.

F1 - Cost Price - Cost Price for each unit sold.

F2 - On Hand Balance - Quantity of the item on hand and available for shipment on customer orders.

F3 - Number of Back Orders - A statistical total kept to keep track of the number of times the item was not available to fill a customers order.

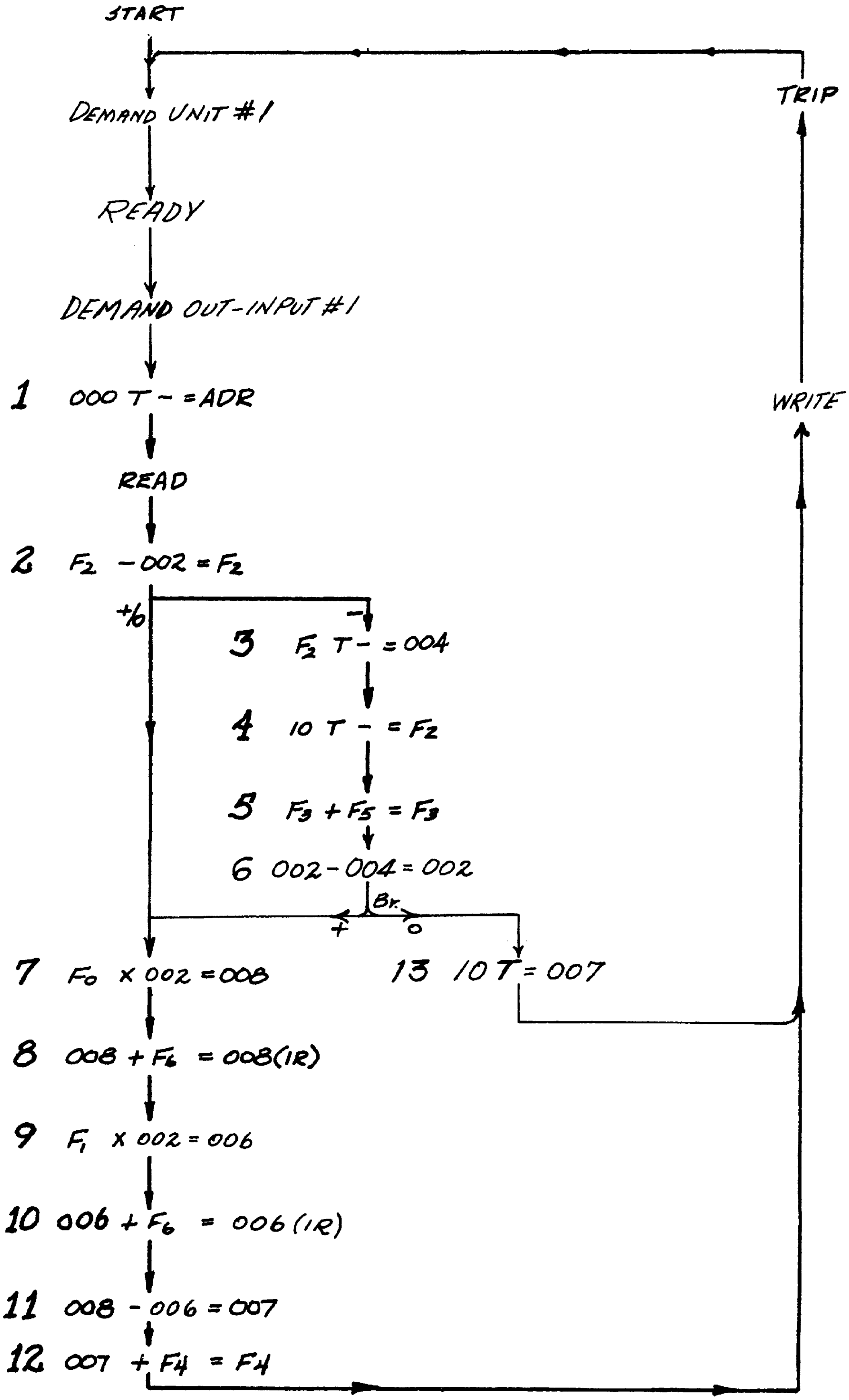
F4 - Gross Profit to Date - A cumulative total of the realized gross profit on shipments of the item to date (i.e. since beginning of the accounting period).

F5 - Constant 1 - This constant is required to enable us to add a one (1) to F3 when a back-order occurs. It is kept in the unit record area rather than intermediate storage since we have space in the URA and the field access time to a 30 digit URA is faster than to an intermediate storage.

F6 - Constant 5 - Required to round the cost and sales amounts.

For the purpose of this problem we will assume the above data has been loaded on the drums.

6. Program Flow Chart



Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM

150 CPM INPUT-OUTPUT CHART

APPLICATION: Billing & Inventory PROGRAM NO.: II

INPUT-OUTPUT STORAGE FIELD ASSIGNMENT

DEMAND UNIT NO. 1 OR SCAN UNIT NO. ____

INPUT FIELD ASSIGNMENT

SYM.	DESCRIPTION	CARD COLUMNS AND DECIMALS 11	10	9	8	7	6	5	4	3	2	1	SN.	SIGN CTL.
000	ITEM NUMBER (DRUM ADDRESS)						1	2	3	4	5	6	+	
001														
002	QUANTITY ORDERED								13	14	15	16	+	
003														
004														
005														
006														
007														
008														
009														

OUTPUT FIELD ASSIGNMENT

SYM.	DESCRIPTION	CARD COLUMNS AND DECIMALS 11	10	9	8	7	6	5	4	3	2	1	SN.	SIGN CTL.
000														
001														
002	QUANTITY SHIPPED								41	42	43	44	+	
003														
004	QUANTITY BACK ORDERED								37	38	39	40	+	
005														
006	COST AMOUNT						83	84	85	86	87	88	+	
007	GROSS PROFIT AMOUNT						71	72	73	74	75	76	+	
008	SALES AMOUNT						77	78	79	80	81	82	+	
009														

INPUT CONTROL LINES

SYM.	FROM
a	
b	
c	
d	
e	
f	
g	
h	
i	
j	
k	
l	

OUTPUT CONTROL LINES

SYM.	TO
A	TRIP
B	
C	
D	
E	
F	
G	
H	
I	
J	

INPUT-OUTPUT SELECTORS

PICK UP	S#	SELECT	COMMON	NON-SELECT
	1a			
	1b			
	1c			
	2a			
	2b			
	2c			
	3a			
	3b			
	3c			
	4			
	5a			
	5b			
	5c			
	6a			
	6b			
	6c			
	7a			
	7b			
	7c			
	8a			
	8b			
	8c			
	9a			
	9b			
	9c			
	10a			
	10b			
	10c			

INPUT TRANSFER

IN (from)	EVEN
NO (to)	
EVEN (to)	IN
ALL (to)	

PROGRAM SELECTS

NO.	IN FROM	POWER OUT TO
1		
2		

REMARKS:

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM

STORAGE ASSIGNMENT CHART

APPLICATION: BILLING & INVENTORY PROGRAM NO.: II

LARGE CAPACITY DRUM FIELD ASSIGNMENT

ADDRESS FROM 000000
TO 049919

SYM	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
F 0	SALES PRICE							4	3	2	1	0	+
F 1	COST PRICE							9	8	7	6	5	+
F 2	ON HAND BALANCE						15	14	13	12	11	10	+
F 3	NUMBER OF BACK ORDERS								19	18	17	16	+
F 4	GROSS PROFIT TO DATE				27	26	25	24	23	22	21	20	+
F 5	CONSTANT 1											28	+
F 6	CONSTANT 5											29	+
F 7													
F 8													
F 9													
F10													
F11													
F12													
F13													
F14													
F15													
F16													
F17													
F18													
F19													

LARGE CAPACITY DRUM FIELD ASSIGNMENT

ADDRESS FROM ________
TO ________

SYM	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
F 0													
F 1													
F 2													
F 3													
F 4													
F 5													
F 6													
F 7													
F 8													
F 9													
F10													
F11													
F12													
F13													
F14													
F15													
F16													
F17													
F18													
F19													

LARGE CAPACITY DRUM FIELD ASSIGNMENT SELECTORS

PICK-UP FROM	GROUND TO	A#	SELECT	COMMON	NON-SELECT
		1a			
		1b			
		1c			
		1d			
		2a			
		2b			
		2c			
		2d			
		3a			
		3b			
		3c			
		3d			
		4a			
		4b			
		4c			
		4d			
		5a			
		5b			
		5c			
		5d			
		6a			
		6b			
		6c			
		6d			
		7a			
		7b			
		7c			
		7d			
		8a			
		8b			
		8c			
		8d			

INTERMEDIATE STORAGE FIELD ASSIGNMENT

TRACK 50

SYM	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
50													
51													
52													
53													
54													
55													
56													
57													
58													
59													

INTERMEDIATE STORAGE FIELD ASSIGNMENT

TRACK 10

SYM.	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
10	CLEAR (SPACE CODES)	—	—	—	—	—	—	—	—	—	—	—	+
11													
12													
13													
14													
15													
16													
17													
18													
19													

TRACK 20

20													
21													
22													
23													
24													
25													
26													
27													
28													
29													

TRACK 30

30													
31													
32													
33													
34													
35													
36													
37													
38													
39													

TRACK 40

40													
41													
42													
43													
44													
45													
46													
47													
48													
49													

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM

PROGRAM PLANNING CHART

APPLICATION: BILLING & INVENTORY PROGRAM NO.: II

STEP NO.	INPUT UNIT NO.	CARD NO.	VALUE 1			PRO-CESS	VALUE 2			RESULT			NEXT STEP
			DESCRIPTION	SYM	SH		DESCRIPTION	SYM	SH	DESCRIPTION	SYM	SH	
1	/	/	ITEM No. (DRUM ADD)	000		T				ITEM No.	ADR		R1
2	/	/	ON HAND BALANCE	F2		—	QUANTITY ORDERED	002		O/H NEW O/H BALANCE – B/O QTY	F2		BR1
3	/	/	B/O QTY	F2		T				BACK-ORDER QTY	004		4
4	/	/	SPACE CODES	10		T				CLEARED O/H BAL.	F2		5
5	/	/	No. OF BACKORDERS	F3		+	CONSTANT 1	F5		No. OF BACK-ORDERS	F3		6
6	/	/	QUANTITY ORDERED	002		—	BACKORDER QTY.	004		QUANTITY SHIPPED	002		BR2
7	/	/	SALES PRICE	F0		X	QTY. SHIPPED	002		SALES AMT. (N.R.)	008		8
8	/	/	SALES AMT. (N.R.)	008		+	CONSTANT 5	F6		SALES AMT. (RDD)	008	1R	9
9	/	/	COST PRICE	F1		X	QTY. SHIPPED	002		COST AMT. (N.R.)	006		10
10	/	/	COST PRICE (N.R.)	006		+	CONSTANT 5	F6		COST AMT. (R.D.D)	006	1R	11
11	/	/	SALES AMOUNT	008		—	COST AMOUNT	006		GROSS PROFIT AMT.	007		12
12	/	/	GROSS PROFIT AMT.	007		+	GROSS PROFIT T.D.	F4		GROSS PROFIT T.D.	F4		W1
13	/	/	SPACE CODE	10		T				CLEAR GROSS PROFIT	007		W1
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													
41													
42													
43													
44													
45													
46													
47													
48													

VALUE 1 AS STORED

STEP	11	10	9	8	7	6	5	4	3	2	1	SN
1						X	X	X	X	X	X▲	+
2						X	X	X	X	X	X▲	+
3								X	X	X	X▲	+
4	-	-	-	-	-	-	-	-	-	-	-	+
5								X	X	X	X▲	+
6								X	X	X	X▲	+
7							X	X▲	X	X	X	+
8					X	X	X	X▲	X	X	X	+
9							X	X▲	X	X	X	+
10					X	X	X	X▲	X	X	X	+
11						X	X	X	X▲	X	X	+
12						X	X	X	X▲	X	X	+
13	-	-	-	-	-	-	-	-	-	-	-	+

VALUE 2 AS STORED

STEP	11	10	9	8	7	6	5	4	3	2	1	SN
1												
2								X	X	X	X▲	+
3												
4												
5											1▲	+
6								X	X	X	X▲	+
7								X	X	X	X▲	+
8											5▲	+
9								X	X	X	X▲	+
10											5▲	+
11						X	X	X	X▲	X	X	+
12				X	X	X	X	X	X▲	X	X	+
13												

UNSHIFTED RESULT

STEP	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	SN
1																	X	X	X	X	X	X	+
2																	X	X	X	X	X	X▲	±
3																			X	X	X	X▲	+
4												-	-	-	-	-	-	-	-	-	-	-	+
5																			X	X	X	X▲	+
6																			X	X	X	X▲	+
7																X	X	X	X▲	X	X	X	+
8																X	X	X	X▲	X	X	X	+
9																X	X	X	X▲	X	X	X	+
10																X	X	X	X▲	X	X	X	+
11																	X	X	X	X▲	X	X	+
12															X	X	X	X	X	X▲	X	X	+
13												-	-	-	-	-	-	-	-	-	-	-	+

STORED RESULT

STEP	11	10	9	8	7	6	5	4	3	2	1	SN
1						X	X	X	X	X	X	
2						X	X	X	X	X	X▲	+
3								X	X	X	X▲	+
4						-	-	-	-	-	-	+
5								X	X	X	X▲	+
6								X	X	X	X▲	+
7					X	X	X	X▲	X	X	X	+
8						X	X	X	X▲	X	X	+
9					X	X	X	X▲	X	X	X	+
10						X	X	X	X▲	X	X	+
11						X	X	X	X▲	X	X	+
12				X	X	X	X	X	X▲	X	X	+
13	-	-	-	-	-	-	-	-	-	-	-	+

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM
FUNCTION CONTROL CHART

APPLICATION: Billing & Inventory PROGRAM NO.: II

VI-25

DEMAND UNITS

NO.	TYPE OF UNIT	TEST IN FROM	TEST OUT TO		DEMAND IN FROM	DEMAND OUT TO	
			NOT READY	READY		INPUT	OUTPUT
1	150 CPM				START, OE_1	IN ST#1	
2							
3							
4							
5							
6							
7							
8							

SCAN UNITS

NO.	TYPE OF UNIT	TO SCAN OUT	NO.	TYPE OF UNIT	TO SCAN OUT	NO.	TYPE OF UNIT	TO SCAN OUT
1			9			17		
2			10			18		
3			11			19		
4			12			20		
5			13			21		
6			14			22		
7			15			23		
8			16			24		

CODE DISTRIBUTOR

FROM V1 OF STEPS	
FROM V2 OF STEPS	
FROM R OF STEPS	

NO.	TO	NO.	TO
00		26	
01		27	
02		28	
03		29	
04		30	
05		31	
06		32	
07		33	
08		34	
09		35	
10		36	
11		37	
12		38	
13		39	
14		40	
15		41	
16		42	
17		43	
18		44	
19		45	
20		46	
21		47	
22		48	
23		49	
24		>49	
25			

FROM OUT OF FUNCTION:	

NO.	TO	NO.	TO
0		5	
1		6	
2		7	
3		8	
4		9	

INPUT CONTROL LINES

NO.	TO
a	
b	
c	
d	
e	
f	
g	
h	
i	
j	
k	
l	

OUTPUT CONTROL LINES

NO.	FROM
A	OE_1
B	
C	
D	
E	
F	
G	
H	
I	
J	

START TO	DI#1

SCAN OUTS

NO.	TO

ERROR CHECKS

FROM	TO
ARITH.	STEP REPEAT
±O'FLOW	STOP
÷O'FLOW	
PARITY	STOP
ADDRESS	STOP

INDICATORS

NO.	FROM	
1		
2		
3		
4		
IND. SWITCH	FROM	TO
NO CHECK		

READ UNIT RECORD

NO.	IN FROM	OUT TO
1	STEP #1	STEP #2
2		
3		
4		
5		
6		
7		
8		

WRITE UNIT RECORD

NO.	IN FROM	OUT TO
1	ST#12, ST#13	OE_1
2		
3		
4		
5		
6		
7		
8		

STEP CLEAR

IN	OUT
CLEAR PROG. SEL.	

STEP REPEAT

IN FROM	ARITH. ERROR

CHANNEL SEARCH EQUAL

NO.	FROM OUT OF STEP NO.	TO IF = V1	TO IF = V2	TO IF ≠
1				
2				
3				
4				

BRANCHING

NO.	IN FROM STEP	TO IF +	TO IF −	TO IF O
1	ST# 2	STEP #7	STEP#3	STEP #7
2	ST#6	STEP#7		STEP #13
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				

CHANNEL SEARCH UNEQUAL

NO.	FROM OUT OF STEP NO.	TO IF =	TO IF ≠
1			
2			
3			
4			

OUT EXPANDERS

NO.	IN	OUT	OUT
1	W_1	OPC A	DI #1
2			
3			
4			
5			
6			
7			
8			

FUNCTION DELAYS

NO.	IN 1	IN 2	OUT
1			
2			
3			

ALTERNATE SWITCHES

NO.	SELECT	COMMON	NON-SELECT
1			
2			
3			
4			
5			
6			

FUNCTION SEQUENCE

NO.	SET	PROBE	OUT
1			
2			

SYMBOLS FOR PROGRAMMING:

Processes:	PR
Transfer	T
Addition	+
Subtraction	−
Multiplication	X
Division	÷
Left Zero Elim	LZE
Compare	C
Channel Search =	ECS
Channel Search ≠	UCS

SHIFTS:

Shift 2 Left	2L
Shift 2 Right	2R
Expand Prod. Shift	EXS

SELECTORS:

Computor Selector	$T_1 \to T_{48}$
Drum Selector	$A_1 \to A_8$
Input-Output Selectors	$S_1 \to S_{10}$
Common	CT(#) or CA (#)
Select	ST (#) or SA (#)
Non-Select	NST(#) or NSA (#)
Pick-Up	PUT(#) or PUA (#)
Program Selector	$PS_1 \to PS_{16}$
Delay of PS	$DPS_1 \to DPS_{10}$
Imm. of PS	$IPS_{11} \to IPS_{16}$
Power of PS	$PSO_1 \to PSO_{16}$
Drop Out of PS	$PSDO_1 \to PSDO_{16}$
Ground of Sel	GT (#)
Computer Ground	CG
Demand Ground	$DG_1 \to DG_8$
Scan Ground	$SG_1 \to SG_6$
Selector Hold	SH

Function Delay	FD
In FD 1	$FD1_1 \to FD1_3$
In FD 2	$FD2_1 \to FD2_3$
Out FD	$FDO_1 \to FDO_3$

STORAGES:

Input-Output	000→009
Intermediate	10→59
Revolver	$F_0 \to F_{19}$
Channel Search Fields	$CSF_0 \to CSF_{19}$

DEMAND UNITS:

Test In	TST (#)
Test Out Ready	TSTR (#)
Test Out Not Ready	TSTN (#)
Demand In	DI (#)
Demand Out Input	DOI (#)
Demand Out Output	DOO (#)

Scan Out	$SO_1 \to SO_6$
Program Step In	In ST (#)
Program Step Out	From ST (#)
Out Expanders	$OE_1 \to OE_8$
In Branching	$BR_1 \to BR_{18}$
+ Branch Out	$+BR_1 \to BR_{18}$
− Branch Out	$-BR_1 \to BR_{18}$
o Branch Out	$oBR_1 \to BR_{18}$
Read Unit Record	$R_1 \to R_8$
Write Unit Record	$W_1 \to W_8$
Equal Channel Search In	ECSI (#)
Equal Channel Search Out=V_1	$ECSV_1$ (#)
Equal Channel Search Out=V_2	$ECSV_2$ (#)
Equal Channel Search Out ≠	ECS ≠ (#)
Unequal Channel Search In	UCSI (#)
Unequal Channel Search Out =	UCS = (#)
Unequal Channel Search Out ≠	UCS ≠ (#)
Program Address Counter	PAK
Address Register	ADR

Code Distributor	CD
In CD V_1	CDV_1
In CD V_2	CDV_2
In CD R	CDR
Out CD 00 to > 49	$CD_{00} \to CD > 49$
In CD Pulse In	CDP
Out CD Pulse Out	CDP_0--CDP_9
Function Sequence Set	$FSS_1 \to FSS_2$
Function Sequence Probe	$FSP_1 \to FSP_2$
Function Sequence Out	$FSO_1 \to FSO_2$

ALTERNATE SWITCHES:

Common	$SW_1C \to SW_6C$
Select	$SW_1S \to SW_6S$
Non-Select	$SW_1NS \to SW_6N_6$

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM

COMPUTER SELECTOR CONTROL CHART

APPLICATION: BILLING & INVENTORY (NOT USED IN THIS PROBLEM) PROGRAM NO. II

PICK-UP FROM	GROUND TO	T#	SELECT	COMMON	NON-SELECT
		1			
		2			
		3a			
		3b			
		4a			
		4b			
		5a			
		5b			
		6a			
		6b			
		6c			
		6d			
		7			
		8			
		9a			
		9b			
		10a			
		10b			
		11a			
		11b			
		12a			
		12b			
		12c			
		12d			
		13			
		14			
		15a			
		15b			
		16a			
		16b			
		17a			
		17b			
		18a			
		18b			
		18c			
		18d			
		19			
		20			
		21a			
		21b			
		22a			
		22b			
		23a			
		23b			
		24a			
		24b			
		24c			
		24d			

PICK-UP FROM	GROUND TO	T#	SELECT	COMMON	NON-SELECT
		25			
		26			
		27a			
		27b			
		28a			
		28b			
		29a			
		29b			
		30a			
		30b			
		30c			
		30d			
		31			
		32			
		33a			
		33b			
		34a			
		34b			
		35a			
		35b			
		36a			
		36b			
		36c			
		36d			
		37			
		38			
		39a			
		39b			
		40a			
		40b			
		41a			
		41b			
		42a			
		42b			
		42c			
		42d			
		43			
		44			
		45a			
		45b			
		46a			
		46b			
		47a			
		47b			
		48a			
		48b			
		48c			
		48d			

PROGRAM SELECTS

PS#	IN FROM	DELAY OUT TO	D. O. FROM	POWER TO
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

PS#	IN FROM	IMMEDIATE OUT TO	D. O. FROM	POWER TO
11				
12				
13				
14				
15				
16				

REMARKS:

Program Comments

Step 2) Although a minus result may occur on this step the actual numeric result will be stored in F2 as a positive number since we have an applied plus sign in F2.

Step 8) & 10)

The sales and cost amounts must be shifted 1 position to the right in order to drop off the trailing decimal in the rounded result. If this is not done it is possible to get an incorrect gross profit figure in step 11. (Incorrect in the sense that the punched gross profit plus the punched cost amount may not equal the punched sales amount).

4. PROBLEM III

1. Statement of Problem

This problem is a further refinement of problems I and II in that it includes provision for other types of transactions which can affect a finished stock inventory item i.e.:

a. Customer Order Cards
b. Stock Receipt Cards
c. Stock Order Cards
d. Cards representing customer orders which have been previously back-ordered.

The inclusion of these new transactions makes this problem more comprehensive in scope than the prior inventory cases. The calculations required, by card, are as follows:

a. Customer Order Cards

(1) Reduce drum on hand balance by quantity ordered
(2) If the result of (1) is minus:
    (a) Increase the drum quantity back-ordered by the quantity back-ordered on this transaction.
    (b) Reduce the drum on hand balance to zero.
    (c) Place the quantity back-ordered in storage for punching.
    (d) Reduce quantity ordered by quantity back-ordered to obtain quantity shipped.

(3) If the result of (1) was plus or zero:
    (a) Quantity shipped equals quantity ordered.
(4) On all items, quantity shipped times sales price equals sales amount, round to two decimals.

b. Stock Receipt Cards

(1) Increase on hand balance by the quantity received.
(2) Reduce drum quantity on order by the quantity ordered.
(3) Punch the difference between quantity ordered and quantity received in the card as the stock overage or shortage. Also punch a zero (0) in column 18 if the difference represents a shortage.

Note: Assume that each receipt card represents the completion of a stock order.

c. Stock Order Card

(1) Increase the drum on order balance by the quantity ordered.

d. Back-Order Card - Prior day transactions.

(1) Treat the same as a customer order card, except
(2) Reduce the drum quantity back-ordered total by the quantity ordered field.

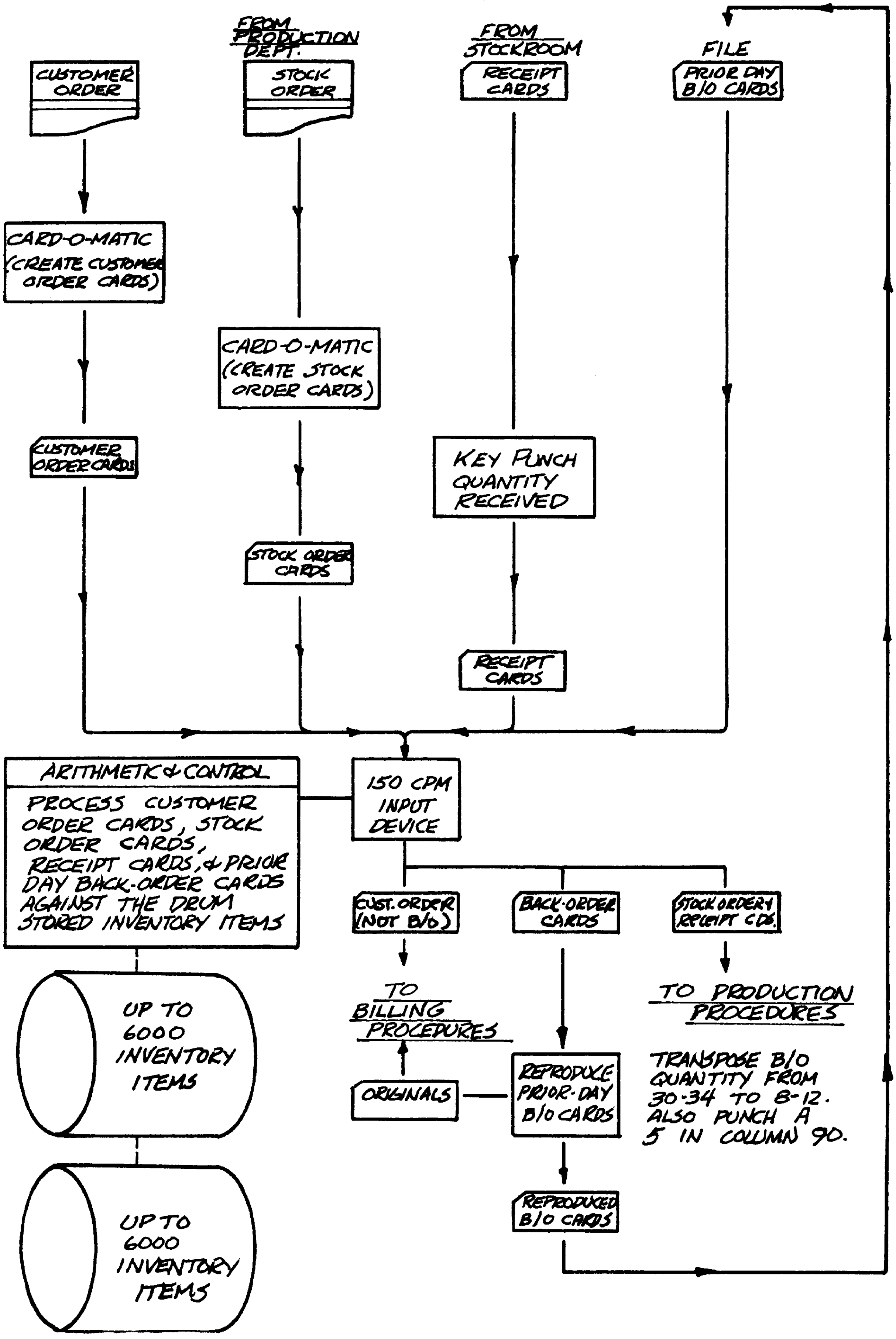
Note: Quantity ordered in this card, represents the quantity back-ordered in a customer's order at some previous time.

The problem involves 7500 cards a day entered at random to apply against 10,000 items. The arrangement of the items on the drums creates over-flow channels on approximately 10% of the channels used in the system. Of the 7500 transactions there are approximately 7000 customer order cards, 215 receipt cards, 215 stock order cards and 70 prior day back-order cards.

2. Machine Specifications

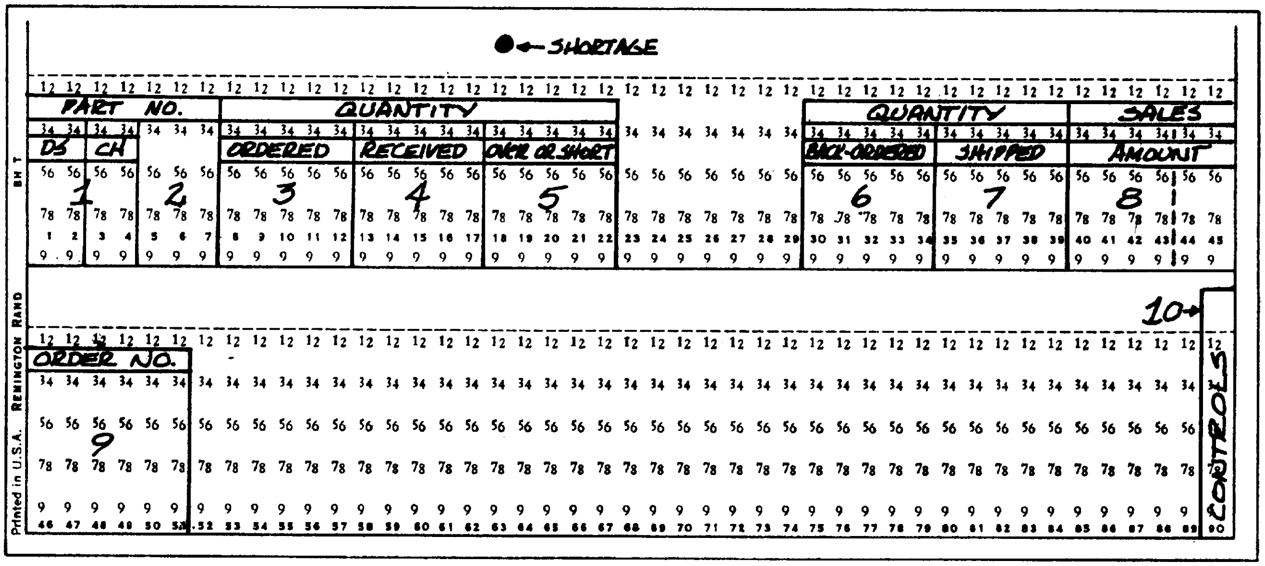
1 - 150 CPM - Card Sensing Punching Unit
2 - Large Capacity Drums with a 30 digit URA length
1 - Arithmetic and Control Unit (external programming)

3. GENERAL PROCEDURE FLOW CHART


CUSTOMER ORDER
FROM PRODUCTION DEPT.
STOCK ORDER
FROM STOCKROOM
RECEIPT CARDS
FILE
PRIOR DAY B/O CARDS
CARD-O-MATIC (CREATE CUSTOMER ORDER CARDS)
CARD-O-MATIC (CREATE STOCK ORDER CARDS)
KEY PUNCH QUANTITY RECEIVED
CUSTOMER ORDER CARDS
STOCK ORDER CARDS
RECEIPT CARDS
ARITHMETIC & CONTROL
PROCESS CUSTOMER ORDER CARDS, STOCK ORDER CARDS, RECEIPT CARDS, & PRIOR DAY BACK-ORDER CARDS AGAINST THE DRUM STORED INVENTORY ITEMS
150 CPM INPUT DEVICE
CUST. ORDER (NOT B/O)
BACK-ORDER CARDS
STOCK ORDER & RECEIPT CDS.
TO BILLING PROCEDURES
TO PRODUCTION PROCEDURES
UP TO 6000 INVENTORY ITEMS
ORIGINALS
REPRODUCE PRIOR-DAY B/O CARDS
TRANSPOSE B/O QUANTITY FROM 30-34 TO 8-12. ALSO PUNCH A 5 IN COLUMN 90.
REPRODUCED B/O CARDS
UP TO 6000 INVENTORY ITEMS

4. Input-Output Data and Definition

The source media consists of 90 column punched cards, entered in random sequence, and punched in the following form:



Definitions

Input

(1) All Cards - Drum section and channel number of the item's large capacity drum URA.

(2) All Cards - Complete part number (cols. 1-7) of the item.

(3) Customer Order Card - Quantity of the item which the customer has ordered.

Stock Receipt - Quantity of the item which was ordered to be produced for stock.

Stock Order Card - Quantity of the item which is now being ordered to be produced for stock.

Back Order Cards - Quantity of the item which was back-ordered on a customers order prior to this time. This quantity is to be re-tested against the inventory balance to determine if it can now be shipped.

(4) Stock Receipt Card - Quantity received by the stockroom for the complete production order. This may or may not agree with the quantity ordered (3).

All Other Cards - Blank

(9) Order No. - Stock order number, Stock Receipt number and Customers order number not required in the calculation.

(10) Customer Order Card - 0 in column 90
Stock Receipt Card - 1 in column 90
Stock Order Card - 3 in column 90
Back Order Card - 5 in column 90

Output

(5) Stock Receipt Card - Difference between (3) and (4) represents a production shortage or overage, a 0 in column 18 should be punched if the quantity represents a shortage.

All Other Cards - Blank

(6) Customer Order Card - That portion of the quantity ordered which cannot be shipped out of stock.

Back Order Cards - Same as for the customer order card except that this represents the second (or third etc.) time the item has been back-ordered.

All Other Cards - Blank

(7) Customer Order Card and Back-Order Card - That portion of the quantity ordered which can be shipped from stock.

All Other Cards - Blank

(8) Customer Order and Back-Order Cards - Result of quantity shipped multiplied by the sales price and rounded to two decimal places.

All Other Cards - Blank

The above input-output is assigned to the input-output storages as follows:

INPUT STORAGES

Sym.	Description
000	Part No.
002	Quantity Received
004	Quantity Ordered

OUTPUT STORAGES

Sym.	Description
002	Overage or Shortage
004	Quantity Shipped
006	Sales Amount
008	Quantity Back-ordered

5. Drum Stored Date and Definition

a. Drum unit record areas for 10,000 items, Drum locations 000000 to 054919.

UNIT RECORD AREA - FIELD ASSIGNMENT

Sym.	Description	Digits	Sign
FO	Part No.	xxxxxxx	+/-
F1	Balance on Hand	xxxxxx.	+
F2	Balance on Order	xxxxxx.	+
F3	Unfilled Back-Order Qty. to Date	xxxxx.	+
F4	Sales Price	xx.xxx	+

b. Definitions

Item URA's

FO - Part Number - Complete part number of the item, corresponds to field (2) in the input cards.

F1 - Balance on Hand - Quantity of the item which is on hand and available for shipment.

F2 - Balance on Order - Quantity of the item which has been ordered and will eventually be received in the stock room. This is work in process for a fabricated item or the quantity ordered, but not yet received from vendors for purchased items.

F3 - Unfilled Back-Order Quantity to Date - This is the total of the quantity backordered to date.

F4 - Sales Price - Unit sales price for the item.

Overflow URA's - Unit Record #19 in channels where overflow is required.

F0 - Drum Section and Channel Number of overflow channel i.e.
xx xx 00 -
DS CH URA SN

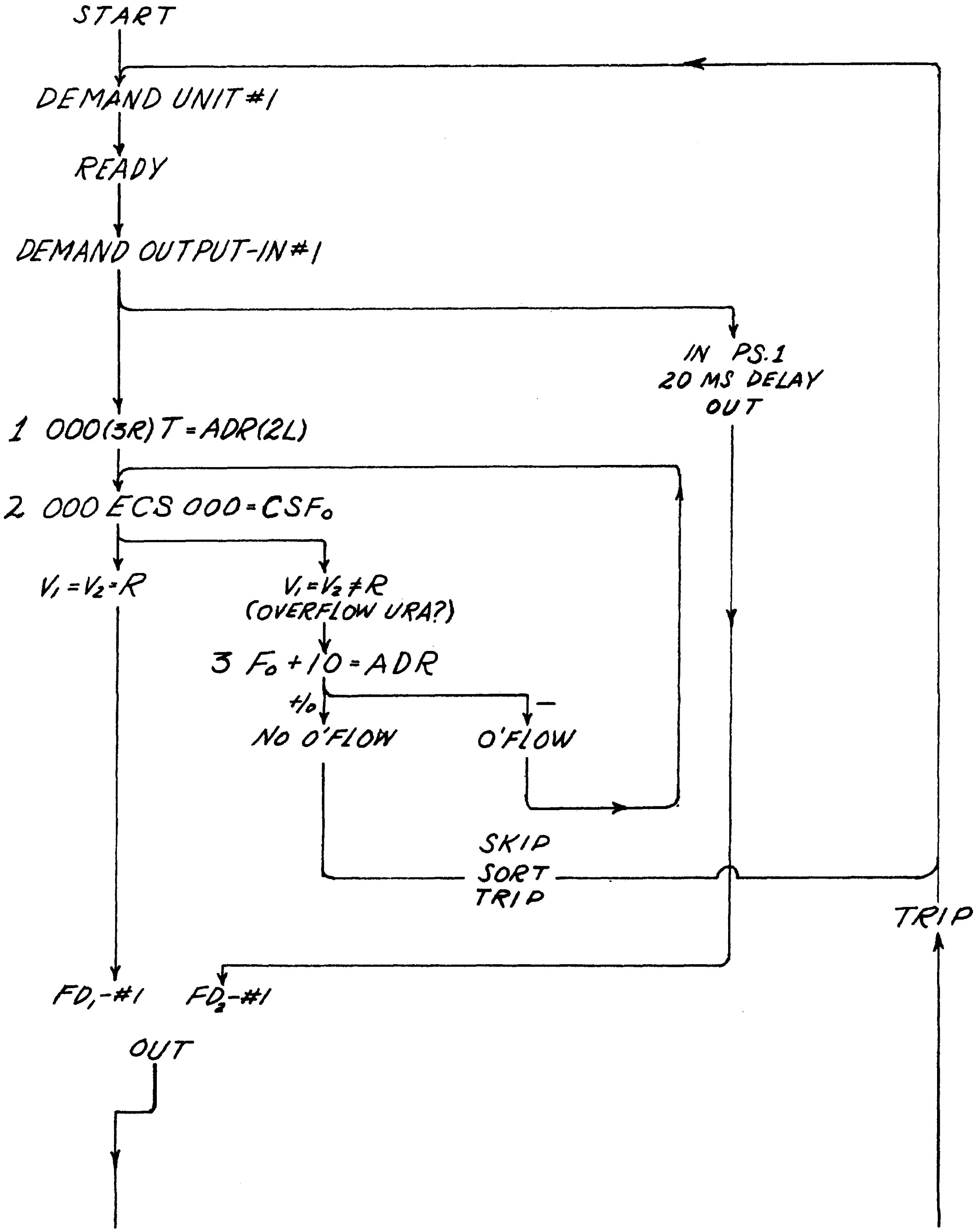
F1 - Spaces
F2 - Spaces
F3 - Spaces
F4 - Spaces

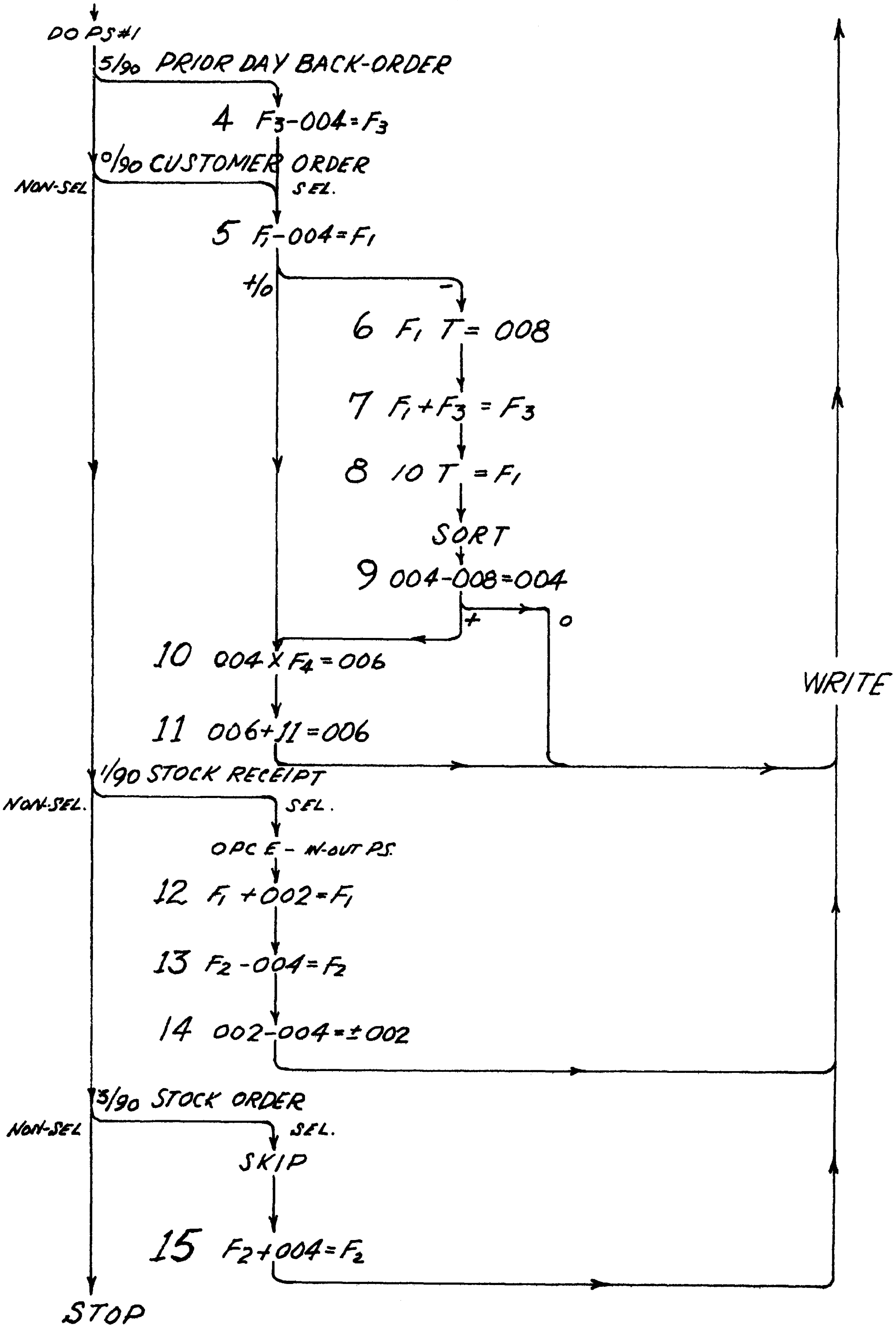
For purposes of this problem we will assume the drum data has been stored. In addition, channel 00 of drum section 00 is not used for overflow.

The following constants are stored in intermediate storage:

storage 10 - spaces
storage 11 - constant 5

6. PROGRAM FLOW CHART




DO PS#1
5/90 PRIOR DAY BACK-ORDER
4 F3-004=F3
0/90 CUSTOMER ORDER
NON-SEL
SEL.
5 F1-004=F1
+/0
-
6 F1 T= 008
7 F1+F3 = F3
8 10 T = F1
SORT
9 004-008=004
+
0
10 004×F4=006
11 006+11=006
WRITE
1/90 STOCK RECEIPT
NON-SEL.
SEL.
OPC E - IN-OUT PS.
12 F1 +002=F1
13 F2 -004=F2
14 002-004=±002
3/90 STOCK ORDER
NON-SEL
SEL.
SKIP
15 F2+004=F2
STOP

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

VI - 37

UNIVAC FILE-COMPUTER SYSTEM

150 CPM INPUT-OUTPUT CHART

APPLICATION: MULTI-CARD INVENTORY & SALES PROGRAM NO.: III

VI-37

INPUT-OUTPUT STORAGE FIELD ASSIGNMENT

DEMAND UNIT NO. 1 OR SCAN UNIT NO. ______

INPUT FIELD ASSIGNMENT

SYM.	DESCRIPTION	CARD COLUMNS AND DECIMALS												SIGN CTL.
		11	10	9	8	7	6	5	4	3	2	1	SN.	
000	PART NUMBER (DS-Col. 1&2, CH-Col. 3&4)					1	2	3	4	5	6	7	+	
001														
002	QUANTITY RECEIVED							13	14	15	16	17	+	
003														
004	QUANTITY ORDERED							8	9	10	11	12	+	
005														
006														
007														
008														
009														

OUTPUT FIELD ASSIGNMENT

SYM.	DESCRIPTION	CARD COLUMNS AND DECIMALS												SIGN CTL.
		11	10	9	8	7	6	5	4	3	2	1	SN.	
000														
001														
002	OVERAGE OR SHORTAGE							18	19	20	21	22	+/-	0/18
003														
004	QUANTITY SHIPPED							35	36	37	38	39	+	
005														
006	SALES AMOUNT					40	41	42	43	44	45		+	
007														
008	QUANTITY BACKORDERED							30	31	32	33	34	+	
009														

INPUT CONTROL LINES

SYM.	FROM
a	0/90
b	1/90
c	3/90
d	5/90
e	
f	
g	
h	
i	
j	
k	
l	

OUTPUT CONTROL LINES

SYM.	TO
A	TRIP
B	SORT & TRIP
C	SORT
D	SKIP
E	IN-OUT PS#1
F	
G	
H	
I	
J	

INPUT-OUTPUT SELECTORS

PICK UP	S#	SELECT	COMMON	NON-SELECT
	1a			
	1b			
	1c			
	2a			
	2b			
	2c			
	3a			
	3b			
	3c			
PS #1	4	——	EVEN OUTPUT CALL LINE	PUNCHING COLS. 35-39
	5a			
	5b			
	5c			
	6a			
	6b			
	6c			
	7a			
	7b			
	7c			
	8a			
	8b			
	8c			
	9a			
	9b			
	9c			
	10a			
	10b			
	10c			

INPUT TRANSFER

IN (from)	EVEN
NO (to)	
EVEN (to)	IN
ALL (to)	

PROGRAM SELECTS

NO.	IN FROM	POWER OUT TO
1	OPC E	PU S#4
2		

REMARKS: CARD #1 - CUSTOMER ORDER CARD
CARD #2 - STOCK RECEIPT CARD
CARD #3 - STOCK ORDER CARD
CARD #4 - BACK ORDER CARD

UNIVAC FILE-COMPUTER SYSTEM

STORAGE ASSIGNMENT CHART

APPLICATION: MULTI-CARD PROGRAM NO.: III

INVENTORY & SALES

PART URA'S LARGE CAPACITY DRUM FIELD ASSIGNMENT ADDRESS FROM 000000 TO 054919

SYM	DESCRIPTION	CHARACTER AND DECIMAL											
		11	10	9	8	7	6	5	4	3	2	1	SN
F 0	PART NUMBER					7	6	5	4	3	2	1	0
F 1	BALANCE ON HAND						13	12	11	10	9	8	+
F 2	BALANCE ON ORDER						19	18	17	16	15	14	+
F 3	UNFILLED BACK-ORDER QTY T.D.							24	23	22	21	20	+
F 4	SALES PRICE							29	28	27	26	25	+
F 5													
F 6													
F 7	NOTE:												
F 8	PART Number ALWAYS												
F 9	HAS A + SIGN STORED.												
F10	OVERFLOW URA WILL												
F11	HAVE A − SIGN IN												
F12	CHARACTER Zero.												
F13													
F14													
F15													
F16													
F17													
F18													
F19													

OVERFLOW URA'S
URA 19 of O'FLOW CH'S LARGE CAPACITY DRUM FIELD ASSIGNMENT ADDRESS FROM ______ TO ______

SYM	DESCRIPTION	CHARACTER AND DECIMAL											
		11	10	9	8	7	6	5	4	3	2	1	SN
F 0	DS-CH of O'FLOW CH. (DS CH URA SN XX XX 00 −)						↑						−
F 1	SPACES					SAME							
F 2	SPACES			CHARACTER									
F 3	SPACES		ASSIGNMENT AS ABOVE										
F 4	SPACES						↓						
F 5													
F 6													
F 7													
F 8													
F 9													
F10													
F11													
F12													
F13													
F14													
F15													
F16													
F17													
F18													
F19													

LARGE CAPACITY DRUM FIELD ASSIGNMENT SELECTORS

PICK-UP FROM	GROUND TO	A#	SELECT	COMMON	NON-SELECT
		1a			
		1b			
		1c			
		1d			
		2a			
		2b			
		2c			
		2d			
		3a			
		3b			
		3c			
		3d			
		4a			
		4b			
		4c			
		4d			
		5a			
		5b			
		5c			
		5d			
		6a			
		6b			
		6c			
		6d			
		7a			
		7b			
		7c			
		7d			
		8a			
		8b			
		8c			
		8d			

INTERMEDIATE STORAGE FIELD ASSIGNMENT
TRACK 50

SYM	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
50													
51													
52													
53													
54													
55													
56													
57													
58													
59													

INTERMEDIATE STORAGE FIELD ASSIGNMENT
TRACK 10

SYM.	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
10	SPACES	−	−	−	−	−	−	−	−	−	−	−	+
11	CONSTANT 5											5	+
12													
13													
14													
15													
16													
17													
18													
19													

TRACK 20

20													
21													
22													
23													
24													
25													
26													
27													
28													
29													

TRACK 30

30													
31													
32													
33													
34													
35													
36													
37													
38													
39													

TRACK 40

40													
41													
42													
43													
44													
45													
46													
47													
48													
49													

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

CARD #1 – CUSTOMER ORDER
CARD #2 – STOCK RECEIPT
CARD #3 – STOCK ORDER
CARD #4 – BACK ORDER

VI - 39 UNIVAC FILE-COMPUTER SYSTEM

PROGRAM PLANNING CHART

APPLICATION: MULTI-CARD INVENTORY & SALES PROGRAM NO.: III

STEP NO.	INPUT UNIT NO.	CARD NO.	VALUE 1			PROCESS	VALUE 2			RESULT			NEXT STEP	VALUE 1 AS STORED												VALUE 2 AS STORED												UNSHIFTED RESULT																							STORED RESULT												STEP
			DESCRIPTION	SYM	SH		DESCRIPTION	SYM	SH	DESCRIPTION	SYM	SH		11	10	9	8	7	6	5	4	3	2	1	SN	11	10	9	8	7	6	5	4	3	2	1	SN	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	SN	11	10	9	8	7	6	5	4	3	2	1	SN	
1	/	1,2,3,4	PART NUMBER	000	3R	T				PART NUMBER	ADR	2L	2					X	X	X	X	X	X	X	+																															X	X	X	X	+						X	X	X	X	-	-		1
2	/	1,2,3,4	PART NUMBER	000		ECS	PART NUMBER	000		STORED PART No.	CSF_0		ECS_1					X	X	X	X	X	X	X	+					X	X	X	X	X	X	X	+																X	X	X	X	X	X	X	+													2
3	/	1,2,34	O'FLOW ADDRESS	F_0		+	SPACE CODES	10		+/0 -No overflow UBA – -STORED overflow ADD.	ADR		BR_1						X	X	X	X	-	-	±	-	-	-	-	-	-	-	-	-	-	-	+																	X	X	X	X	O	O	±						X	X	X	X	O	O		3
4	/	4	UNFILLED B/O TODATE	F3		–	QUANTITY ORDERED	004		NEW UNFILLED B/O T.D.	F3		5							X	X	X	X	X	+							X	X	X	X	X	+																		X	X	X	X	X	+							X	X	X	X	X	+	4
5	/	1-4	BALANCE O/H	F1		–	QTY ORDERED	004		+/0 NEW BALANCE O/H – QTY. B/O	F1		BR_2						X	X	X	X	X	X	+							X	X	X	X	X	+																	X	X	X	X	X	X	±						X	X	X	X	0	0	+	5
6	/	1-4	QTY. B/O	F1		T				QTY. BACKORDERED	008		7							X	X	X	X	X	+																														X	X	X	X	X	+						X	X	X	X	X	X	+	6
7	/	1-4	QTY BACKORDER	F1		+	UNFILLED B/O QTY. T.D.	F3		NEW B/O QTY. T.D.	F3		8							X	X	X	X	X	+							X	X	X	X	X	+																		X	X	X	X	X	+							X	X	X	X	X	+	7
8	/	1-4	BLANK (SPACES)	10		T				ZEROIZE BAL. O/H	F1		OPCC 9	-	-	-	-	-	-	-	-	-	-	-	+																								-	-	-	-	-	-	-	-	-	-	-	+						-	-	-	-	-	-	+	8
9	/	1-4	QTY. ORDERED	004		–	QTY. BACKORDER	008		QTY. SHIPPED	004		BR_3							X	X	X	X	X	+							X	X	X	X	X	+																		X	X	X	X	X	+							X	X	X	X	X	+	9
10	/	1-4	QTY. SHIPPED	004		X	SALES PRICE	F4		SALES AMT (N.R.)	006		11							X	X	X	X	X	+							X	X	X	X	X	+																X	X	X	X	X	X	X	+					X	X	X	X	X	X	X	+	10
11	/	1-4	SALES AMT. (N.R.)	006		+	CONSTANT 5	11		SALES AMT (R.D.D.)	006		W_1					X	X	X	X	X	X	X	+											5	+																X	X	X	X	X	X	X	+					X	X	X	X	X	X	X	+	11
12	/	2	BALANCE ON HAND	F1		+	QTY. RECEIVED	002		NEW BAL. O/H	F1		13						X	X	X	X	X	X	+							X	X	X	X	X	+																	X	X	X	X	X	X	+						X	X	X	X	X	X	+	12
13	/	2	BALANCE ON ORDER	F2		–	QTY. ORDERED	004		NEW BAL. ON ORDER	F2		14						X	X	X	X	X	X	+							X	X	X	X	X	+																	X	X	X	X	X	X	+						X	X	X	X	X	X	+	13
14	/	2	QTY. RECEIVED	002		–	QTY. ORDERED	004		OVERAGE OR SHORTAGE	002		W_1							X	X	X	X	X	+							X	X	X	X	X	+																		X	X	X	X	X	±							X	X	X	X	X	±	14
15	/	3	QTY. ORDERED	004		+	BALANCE ON ORDER	F2		NEW BAL. ON ORDER	F2		W_1							X	X	X	X	X	+						X	X	X	X	X	X	+																	X	X	X	X	X	X	+						X	X	X	X	X	X	+	15
16																																																																									16
17																																																																									17
18																																																																									18
19																																																																									19
20																																																																									20
21																																																																									21
22																																																																									22
23																																																																									23
24																																																																									24
25																																																																									25
26																																																																									26
27																																																																									27
28																																																																									28
29																																																																									29
30																																																																									30
31																																																																									31
32																																																																									32
33																																																																									33
34																																																																									34
35																																																																									35
36																																																																									36
37																																																																									37
38																																																																									38
39																																																																									39
40																																																																									40
41																																																																									41
42																																																																									42
43																																																																									43
44																																																																									44
45																																																																									45
46																																																																									46
47																																																																									47
48																																																																									48

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM

FUNCTION CONTROL CHART

APPLICATION: MULTI - CARD INVENTORY & SALES

PROGRAM NO.: III

DEMAND UNITS

NO.	TYPE OF UNIT	TEST IN FROM	TEST OUT TO		DEMAND IN FROM	DEMAND OUT TO	
			NOT READY	READY		INPUT	OUTPUT
1	150 CPM				START, OE_1, OE_3	IN ST#1, PS#1	
2							
3							
4							
5							
6							
7							
8							

SCAN UNITS

NO.	TYPE OF UNIT	TO SCAN OUT	NO.	TYPE OF UNIT	TO SCAN OUT	NO.	TYPE OF UNIT	TO SCAN OUT
1			9			17		
2			10			18		
3			11			19		
4			12			20		
5			13			21		
6			14			22		
7			15			23		
8			16			24		

CODE DISTRIBUTOR

FROM V1 OF STEPS	
FROM V2 OF STEPS	
FROM R OF STEPS	

NO.	TO	NO.	TO
00		26	
01		27	
02		28	
03		29	
04		30	
05		31	
06		32	
07		33	
08		34	
09		35	
10		36	
11		37	
12		38	
13		39	
14		40	
15		41	
16		42	
17		43	
18		44	
19		45	
20		46	
21		47	
22		48	
23		49	
24		>49	
25			

FROM OUT OF FUNCTION:	

NO.	TO	NO.	TO
0		5	
1		6	
2		7	
3		8	
4		9	

INPUT CONTROL LINES

NO.	TO
a	PUT #2 (0/90)
b	PUT #7 (1/90)
c	PUT #8 (3/90)
d	PUT #1 (5/90)
e	
f	
g	
h	
i	
j	
k	
l	

OUTPUT CONTROL LINES

NO.	FROM
A	OE_1
B	OE_3
C	OUT ST #8
D	OE_2, OE_4
E	ST # 7
F	
G	
H	
I	
J	

START TO	DI #1

SCAN OUTS

NO.	TO

ERROR CHECKS

FROM	TO
ARITH.	STEP REPEAT
±O'FLOW	STOP
÷O'FLOW	
PARITY	STOP
ADDRESS	STOP

INDICATORS

NO.	FROM	
1		
2		
3		
4		
IND. SWITCH	FROM	TO
NO CHECK		

READ UNIT RECORD

NO.	IN FROM	OUT TO
1		
2		
3		
4		
5		
6		
7		
8		

WRITE UNIT RECORD

NO.	IN FROM	OUT TO
1	$^{o}Br_3$, steps 11, 14 & 16	OE_1
2		
3		
4		
5		
6		
7		
8		

STEP CLEAR

IN	OUT
CLEAR PROG. SEL.	

STEP REPEAT

IN FROM	ARITH. ERROR

CHANNEL SEARCH EQUAL

NO.	FROM OUT OF STEP NO.	TO IF = V1	TO IF = V2	TO IF ≠
1	2	FD_1#1	—	IN ST #3
2				
3				
4				

CHANNEL SEARCH UNEQUAL

NO.	FROM OUT OF STEP NO.	TO IF =	TO IF ≠
1			
2			
3			
4			

BRANCHING

NO.	IN FROM STEP	TO IF +	TO IF −	TO IF O
1	# 3	OE_2	IN ST # 2	OE_2
2	#5	IN ST #10	IN ST #6	IN ST #10
3	#9	IN ST #10	——	W_1
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				

OUT EXPANDERS

NO.	IN	OUT	OUT
1	W_1	OPC A	DI #1
2	+/o Br_1	OE_3	OPC D
3	OE_2	OPC B	DI #1
4	ST #8	OPC D	IN ST #15
5			
6			
7			
8			

FUNCTION DELAYS

NO.	IN 1	IN 2	OUT
1	DPS #1	ECS #1 (=)	PSDO #1, CT #1
2			
3			

ALTERNATE SWITCHES

NO.	SELECT	COMMON	NON-SELECT
1			
2			
3			
4			
5			
6			

FUNCTION SEQUENCE

NO.	SET	PROBE	OUT
1			
2			

SYMBOLS FOR PROGRAMMING:

Processes:	PR
Transfer	T
Addition	+
Subtraction	–
Multiplication	X
Division	÷
Left Zero Elim	LZE
Compare	C
Channel Search =	ECS
Channel Search ≠	UCS

SHIFTS:

Shift 2 Left	2L
Shift 2 Right	2R
Expand Prod. Shift	EXS

SELECTORS:

Computor Selector	$T_1 \rightarrow T_{48}$
Drum Selector	$A_1 \rightarrow A_8$
Input-Output Selectors	$S_1 \rightarrow S_{10}$
Common	CT(#) or CA (#)
Select	ST (#) or SA (#)
Non-Select	NST(#) or NSA (#)
Pick-Up	PUT(#) or PUA (#)
Program Selector	$PS_1 \rightarrow PS_{16}$
Delay of PS	$DPS_1 \rightarrow DPS_{10}$
Imm. of PS	$IPS_{11} \rightarrow IPS_{16}$
Power of PS	$PSO_1 \rightarrow PSO_{16}$
Drop Out of PS	$PSDO_1 \rightarrow PSDO_{16}$
Ground of Sel	GT (#)
Computer Ground	CG
Demand Ground	$DG_1 \rightarrow DG_8$
Scan Ground	$SG_1 \rightarrow SG_6$
Selector Hold	SH

Function Delay	FD
In FD 1	$FD1_1 \rightarrow FD1_3$
In FD 2	$FD2_1 \rightarrow FD2_3$
Out FD	$FDO_1 \rightarrow FDO_3$

STORAGES:

Input-Output	000→009
Intermediate	10→59
Revolver	$F_0 \rightarrow F_{19}$
Channel Search Fields	$CSF_0 \rightarrow CSF_{19}$

DEMAND UNITS:

Test In	TST (#)
Test Out Ready	TSTR (#)
Test Out Not Ready	TSTN (#)
Demand In	DI (#)
Demand Out Input	DOI (#)
Demand Out Output	DOO (#)

Scan Out	$SO_1 \rightarrow SO_6$
Program Step In	In ST (#)
Program Step Out	From ST (#)
Out Expanders	$OE_1 \rightarrow OE_8$
In Branching	$BR_1 \rightarrow BR_{18}$
+ Branch Out	$+BR_1 \rightarrow BR_{18}$
– Branch Out	$-BR_1 \rightarrow BR_{18}$
o Branch Out	$oBR_1 \rightarrow BR_{18}$
Read Unit Record	$R_1 \rightarrow R_8$
Write Unit Record	$W_1 \rightarrow W_8$
Equal Channel Search In	ECSI (#)
Equal Channel Search Out=V_1	$ECSV_1$ (#)
Equal Channel Search Out=V_2	$ECSV_2$ (#)
Equal Channel Search Out ≠	ECS ≠ (#)
Unequal Channel Search In	UCSI (#)
Unequal Channel Search Out =	UCS = (#)
Unequal Channel Search Out ≠	UCS ≠ (#)
Program Address Counter	PAK
Address Register	ADR

Code Distributor	CD
In CD V_1	CDV_1
In CD V_2	CDV_2
In CD R	CDR
Out CD 00 to > 49	$CD_{00} \rightarrow CD > 49$
In CD Pulse In	CDP
Out CD Pulse Out	CDP_0--CDP_9
Function Sequence Set	$FSS_1 \rightarrow FSS_2$
Function Sequence Probe	$FSP_1 \rightarrow FSP_2$
Function Sequence Out	$FSO_1 \rightarrow FSO_2$

ALTERNATE SWITCHES:

Common	$SW_1C \rightarrow SW_6C$
Select	$SW_1S \rightarrow SW_6S$
Non-Select	$SW_1NS \rightarrow SW_6N_6$

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM

COMPUTER SELECTOR CONTROL CHART

APPLICATION: MULTI-CARD PROGRAM NO. III
INVENTORY & SALES

PICK-UP FROM	GROUND TO	T#	SELECT	COMMON	NON-SELECT
1PC d (5/90)	CG	1	IN ST #4	FDO #1	CT # 2
1PC a (0/90)	CG	2	IN ST # 5	NST #1	CT#7
		3a			
		3b			
		4a			
		4b			
		5a			
		5b			
		6a			
		6b			
		6c			
		6d			
1PC b (1/90)	CG	7	INST#12, OPC E	NST #2	CT #8
1PC c (3/90)	CG	8	OE4	NST #7	—
		9a			
		9b			
		10a			
		10b			
		11a			
		11b			
		12a			
		12b			
		12c			
		12d			
		13			
		14			
		15a			
		15b			
		16a			
		16b			
		17a			
		17b			
		18a			
		18b			
		18c			
		18d			
		19			
		20			
		21a			
		21b			
		22a			
		22b			
		23a			
		23b			
		24a			
		24b			
		24c			
		24d			

PICK-UP FROM	GROUND TO	T#	SELECT	COMMON	NON-SELECT
		25			
		26			
		27a			
		27b			
		28a			
		28b			
		29a			
		29b			
		30a			
		30b			
		30c			
		30d			
		31			
		32			
		33a			
		33b			
		34a			
		34b			
		35a			
		35b			
		36a			
		36b			
		36c			
		36d			
		37			
		38			
		39a			
		39b			
		40a			
		40b			
		41a			
		41b			
		42a			
		42b			
		42c			
		42d			
		43			
		44			
		45a			
		45b			
		46a			
		46b			
		47a			
		47b			
		48a			
		48b			
		48c			
		48d			

PROGRAM SELECTS

PS#	IN FROM	DELAY OUT TO	D. O. FROM	POWER TO
1	DOI #1	IN FD_2-#1	FDO #1	
2				
3				
4				
5				
6				
7				
8				
9				
10				
PS#	IN FROM	IMMEDIATE OUT TO	D. O. FROM	POWER TO
11				
12				
13				
14				
15				
16				

REMARKS:

Program Comments

Before Step (1) - From the demand unit out we have gone to two places (step 1 and PS 1). This is done so that we may get a timed pulse of 20 ms before we enter the selectors after step 2. This is necessary since 20 ms are required before the selectors are picked up from the input device (5 ms delay on the device itself and 15 ms delay on the transfer of the computer selectors).

Step (3) - The last URA is captured. If this is an overflow URA, FO will have a negative sign plus the overflow DS and CH. By adding FO to spaces and delivering the result to the ADR, we can also determine if this is a legitimate overflow URA.

FD_1 - FD_2 - After we have found the URA from the channel search process we are ready to enter our selectors if 20ms have elapsed since start time. This is controlled by delivering the channel search equal pulse and the exit of the 20ms delay from PS1 to a Function Delay unit. With this wiring the entry to the selectors from the Function Delay Out will occur when channel search is completed or 20ms has elapsed since start time, whichever occurs last.

Step (5) - By our drum URA field assignment we have determined that F1 will always be considered as a plus amount. However, since the branching function examines the result sign from the register, not the result storage, we will still get the correct negative branching. This drum wiring does, of course, make the back-order quantity a plus figure.

Note: Certain steps could be combined for the various cards and the values, processes, results and exits, selected on the basis of card control. However, the only saving would be in the number of plugboard steps used, since the number of steps each card went through would remain the same. Since plenty of steps were available this approach would be an unnecessary complication of the problem.

5. PROBLEM IV

1. Statement of problem

The processing of daily job cards is illustrated in this problem. There are approximately 5000 employees with an average of four job cards per employee. A daily run will therefore involve the processing of 20,000 cards. These cards are produced from the employee daily record sheet to show employee clock number, operation number, elapsed hours, pieces produced and the various codes. They are then placed in the UFC. The employee clock number is four digits, of which two are used for channel number. The drum section is obtained from the department number. The operation number is six digits, of which four are used for drum section and channel number. A channel search operation is required in both cases to locate the employee unit record and the operation unit record.

The operation's unit record is used first to determine the employee's standard hours produced. The standard hours produced is obtained by multiplying the standard hours per 100 pieces by the pieces produced. This is the only use of the operation unit record.

The employee unit record is then obtained and the operations performed by the computer are:

a) Determine the employee's earnings at standard. This is the standard hours produced multiplied by the employee's day rate.
b) Calculate any shift pay by multiplying the shift rate, which is coded in the job card by a 1,2 or 3 in column 45, by the elapsed time.
c) Calculate outside work pay which is noted by a code 1 in column 46 (10 cents) multiplied by the elapsed hours.
d) Determine if there is any make up pay due the employee. This is obtained by taking the difference between his guaranteed pay (day rate x elapsed hours) and his earnings at standard (calculated in operation a).
e) Determine employee's total pay, (a) + (b) + (c) + (d).

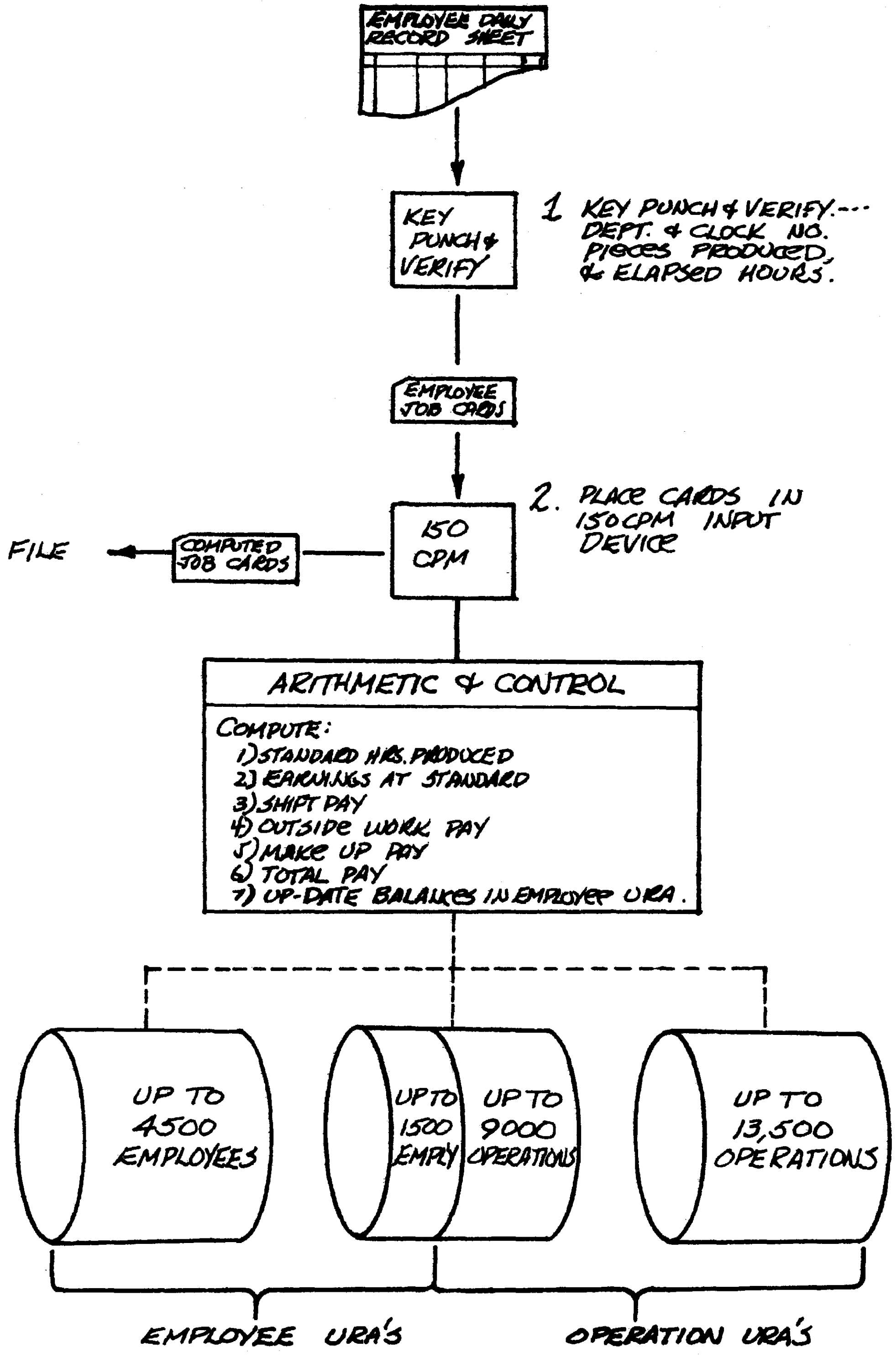
f) Update employee week to date fields.

The problem involves 5,000 employees and 20,000 different operations. The employee URA length is 40 characters and each operation URA required 12 characters. We will put 3 operations in each unit record and maintain the 40 characters unit record length. Therefore 3 drums with a capacity of 13,500 URA's is adequate to store all employee and operation data.

2. Machine Specifications

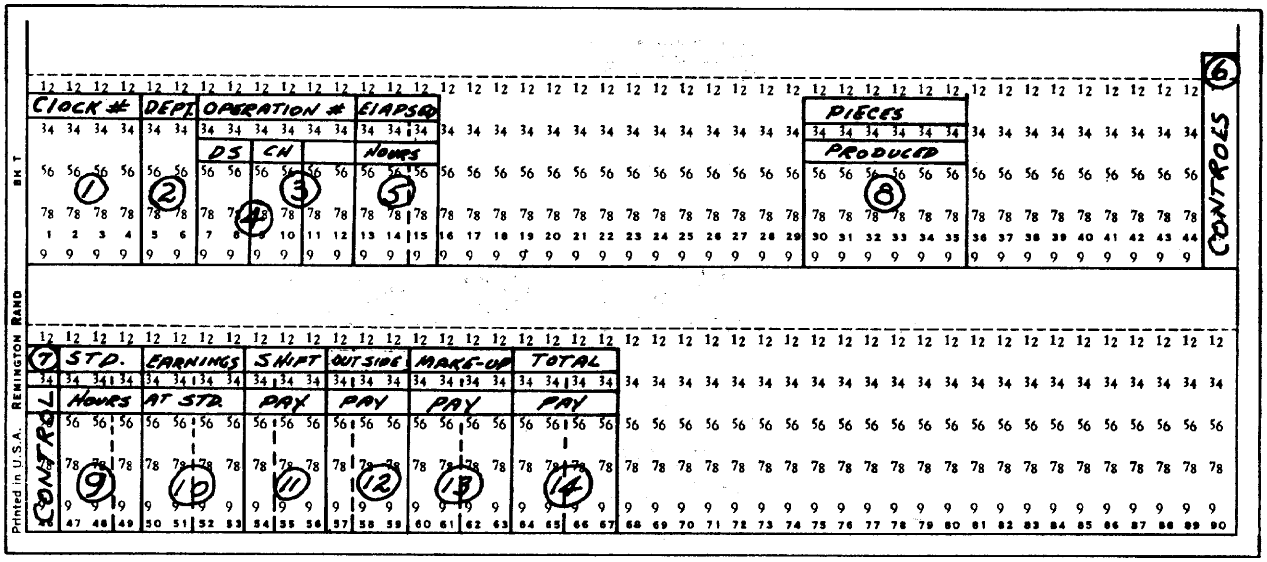
1 - 150 CPM - Card Sensing Punching Unit
3 - Large Capacity Drums (3) with a 40 digit URA length
1 - Arithmetic and Control Unit (external programming)

3 GENERAL PROCEDURE FLOW CHART



4. Input-Output Data and Definition

The input source media are 90 column punched cards, entered in random sequence, and punched in the following form:



Definitions:

Input

(1) Employee clock number (cols. 1-4)

(2) Dept. number (cols. 5 & 6). Drum section is cols. 5 & 6 and channel is cols. 3 & 4

(3) Complete operation number (cols. 7-11) (In incentive cards only.)

(4) Drum section and channel number of operation URA (cols. 7-10)

(5) Elapsed hours - numbers of hours the employee has worked on this particular operation (cols. 13-15)

(6) Shift code - 1 in col. 45 - 1st Shift .00 premium
2 in col. 45 - 2nd Shift .06 premium
3 in col. 45 - 3rd Shift .09 premium

(7) Work code - 1 in col. 46 - outside work - .10 per hour premium
3 in col. 46 - no incentive (no operation number in card)

(8) Pieces produced - quantity of pieces produced by employee on this operation (cols. 30-35)

Output

(9) Standard hours produced -- used to determine employee's earnings at the standard.

(10) Earnings - if incentive worker, standard hours produced x day rate: non-incentive, elapsed hours x day rate.

(11) Shift pay - additional pay due employee for 2nd or 3rd shift work.

(12) Outside work pay - if employee works out of building he receives an additional .10 for every hour worked

(13) Make up pay - difference of employee's rated pay (day rate x elapsed hours) and (10). The employee is guaranteed a minimum wage and if his pay at standard is less he will receive his minimum wage

(14) Total pay - sum of (10), (11), (12) & (13)

The above input-output data is assigned to the following input-output storage locations:

INPUT STORAGES

Sym.	Description
000	Clock Number & Dept. No.
002	Operation Number
004	Elapsed Hours
006	Pieces Produced

OUTPUT STORAGES

Sym.	Description
000	Make-Up Pay
001	Earnings
002	Standard Hours Produced
003	Total Pay
004	Outside Work Pay
008	Shift Pay

5. Drum Stored Data and Definition

a. Drum unit record areas for 5000 employees - Drum location 000000 to 039914.

LARGE CAPACITY DRUM REVOLVER
UNIT RECORD AREA - FIELD ASSIGNMENT
(EMPLOYEE RECORD) A (Select)

Sym.	Description	Digits	Sign
F0	Clock No. & Dept. No.	xxxxxx.	+
F1	Day Rate	x.xxx	+
F2	Reg. Earnings WTD	xxx.xx	+
F3	Shift Pay WTD	x.xx	+
F4	Outside Work Premium WTD	x.xx	+
F5	Gross Earnings YTD	xxxx.xx	+
F6	FICA YTD	xx.xx	+
F7	Fed. W. Tax YTD	xxxx.xx	+
F8	Elapsed Hours WTD	xx.x	+

b. **Definition**

Employee URA's

F0 - Clock No. & Department No. — Employee complete clock and department number. Corresponds to fields (1) and (2) in the input cards.

F1 - Day Rate — Employee's hourly rate of pay.

F2 - Regular Earnings WTD — Weekly accumulation of earnings. This is the sum of the employee's daily earnings and make-up pay.

F3 - Shift Pay WTD — Accumulation of daily shift pay.

F4 - Outside Work Premium WTD — Accumulation of daily outside work pay.

F5 - Earnings YTD — Accumulation of employee's total earnings on a yearly basis (not used in this problem).

F6 - FICA YTD — Federal Insurance Contributions Act deductions accumulated on a yearly basis. (not used in this problem).

F7 - Federal Withholding Tax YTD — Accumulation of Withholding tax on a yearly basis (not used in this problem).

F8 - Elapsed Hours - WTD — Accumulation of hours worked during the week.

c. Drum Unit Record Area for 20,000 operations - Drum location 040000 to 089914.

LARGE CAPACITY DRUM REVOLVER
UNIT RECORD AREA - FIELD ASSIGNMENT
(OPERATION STD - URA)

Sym.	Description	Digits	Sign
F0	Operation No.	xxxxxx.	+
F1	Std. Hrs. Per 100 Pcs.	xxx.xxx	+
F2	Operation No.	xxxxxx.	+
F3	Std. Hrs. Per 100 Pcs.	xxx.xxx	+
F4	Operation No.	xxxxxx.	(stored +)
F5	Std. Hrs. Per 100 Pcs.	xxx.xxx	+

d. Definition

F0, F2, F4 - Operation No. - complete operation number.

F1, F3, F5 - Std. Hrs. per 100 pcs. - This defines the number of hours alloted to the employee for his production of every 100 pieces. This figure multiplied by the total number of pieces produced in this operation is the employee's earnings at standard.

The following constants are stored in intermediate storage:

Storage	20 -	Constant	5
"	21 -	"	spaces
"	22 -	"	06
"	23 -	"	09

Overflow URA's - Unit Record #14 in channels where overflow is required:

a. Operation URA

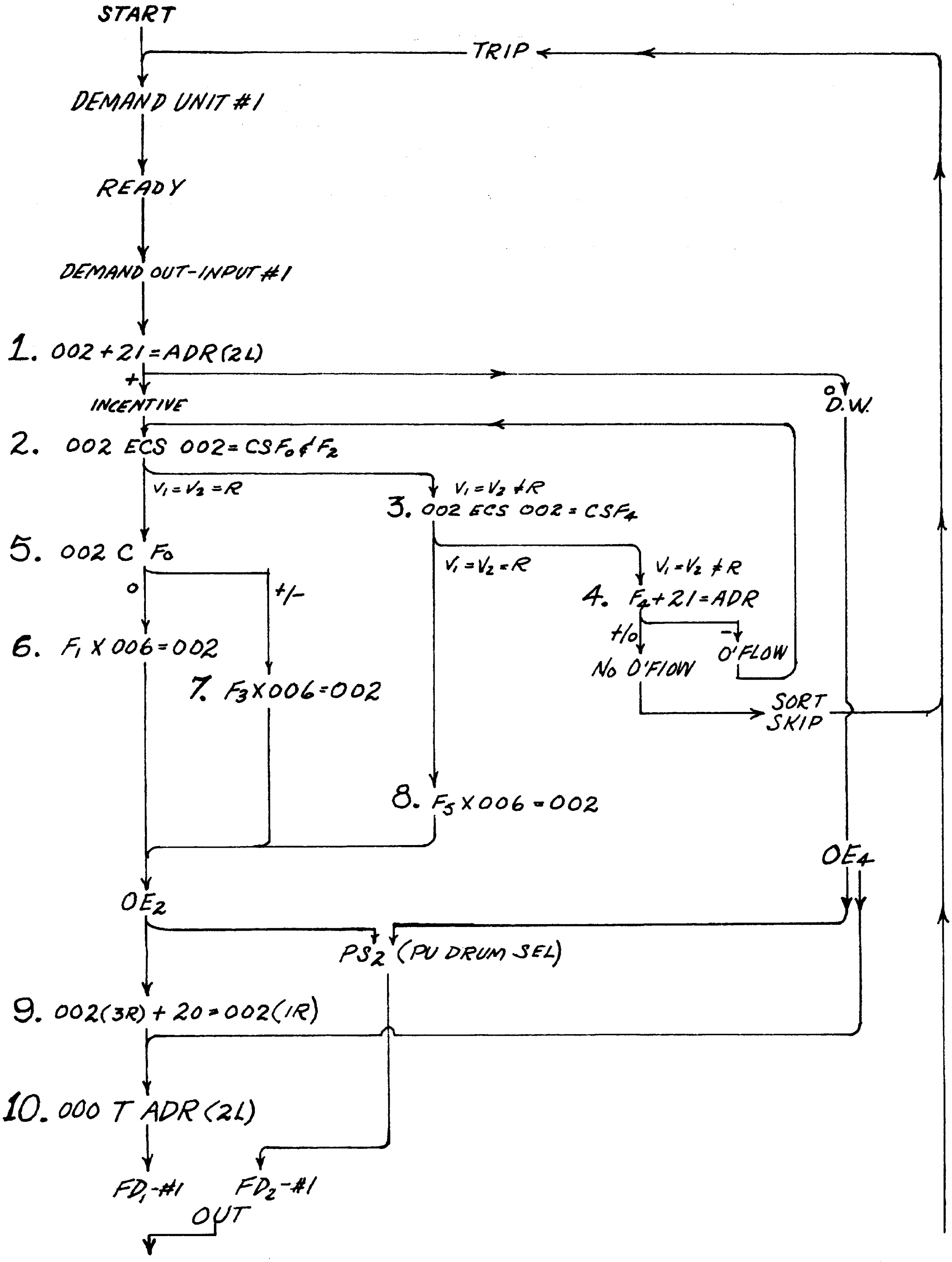
F0-F3 = 2 operations
F4 = Drum Section and channel number of overflow channel (xxxx00-)
F5 = Spaces

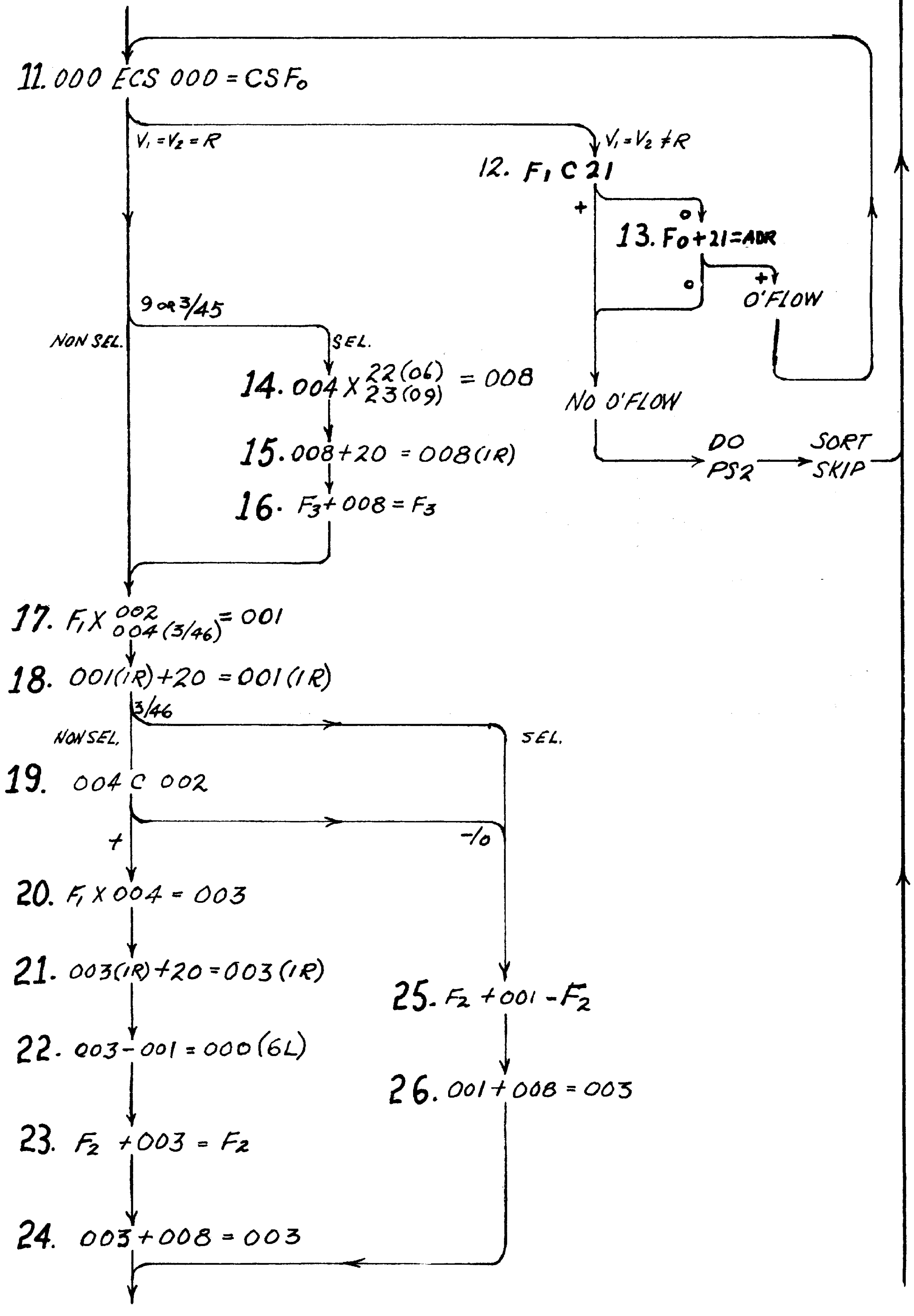
b. Employee URA

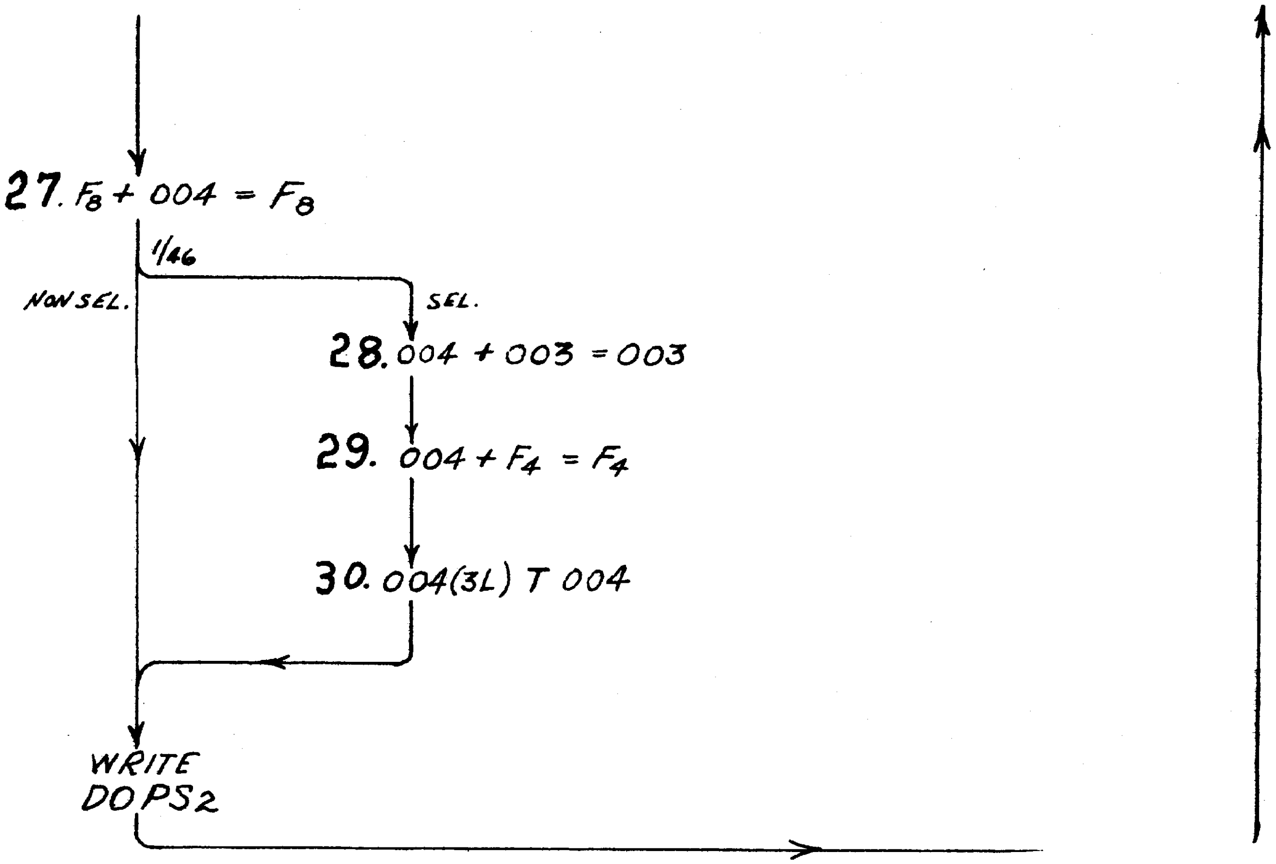
F0 = DS & channel (xxxx00)
F1-F8 = Spaces

Channel 00 of drum section 00 is not used for overflow.

6. PROGRAM FLOW CHART




11. 000 ECS 000 = CS F₀
V₁ = V₂ = R
V₁ = V₂ ≠ R
12. F₁ C 21
+
0
13. F₀ + 21 = ADR
0
+
O'FLOW
NO O'FLOW
DO PS2
SORT SKIP
9 or 3/45
NON SEL.
SEL.
14. 004 X 22 (06) 23 (09) = 008
15. 008 + 20 = 008 (1R)
16. F₃ + 008 = F₃
17. F₁ X 002 004 (3/46) = 001
18. 001 (1R) + 20 = 001 (1R)
3/46
NON SEL.
SEL.
19. 004 C 002
+
-/0
20. F₁ X 004 = 003
21. 003 (1R) + 20 = 003 (1R)
22. 003 - 001 = 000 (6L)
23. F₂ + 003 = F₂
24. 003 + 008 = 003
25. F₂ + 001 - F₂
26. 001 + 008 = 003


27. $F_8 + 004 = F_8$
1/46
NON SEL.
SEL.
28. $004 + 003 = 003$
29. $004 + F_4 = F_4$
30. 004(3L) T 004
WRITE
DOPS2

UNIVAC FILE-COMPUTER SYSTEM

150 CPM INPUT-OUTPUT CHART

APPLICATION: DAILY PAYROLL PROGRAM NO.: IV

INPUT-OUTPUT STORAGE FIELD ASSIGNMENT

DEMAND UNIT NO. ________ OR SCAN UNIT NO. ________

INPUT FIELD ASSIGNMENT

SYM.	DESCRIPTION	CARD COLUMNS AND DECIMALS												SIGN CTL.
		11	10	9	8	7	6	5	4	3	2	1	SN.	
000	CLOCK NUMBER & DEPARTMENT *						1	2	5	6	3	4	+	
001														
002	OPERATION NUMBER *						11	12	7	8	9	10	+	
003														
004	ELAPSED HOURS									13	14▲	15	+	
005														
006	PIECES PRODUCED						30	31	32	33	34	35▲	+	
007														
008														
009														

OUTPUT FIELD ASSIGNMENT

SYM.	DESCRIPTION	CARD COLUMNS AND DECIMALS												SIGN CTL.
		11	10	9	8	7	6	5	4	3	2	1	SN.	
000	MAKE-UP PAY		60	61▲	62	63							+	
001	EARNINGS								50	51▲	52	53	+	
002	STD. HOURS PRODUCED									47	48▲	49	+	
003	TOTAL PAY								64	65▲	66	67	+	
004	OUTSIDE WORK						57▲	58	59				+	
005														
006														
007														
008	SHIFT PAY									54▲	55	56	+	
009														

INPUT CONTROL LINES

SYM.	FROM
a	9/45
b	3/45
c	1/46
d	3/46
e	
f	
g	
h	
i	
j	
k	
l	

OUTPUT CONTROL LINES

SYM.	TO
A	TRIP
B	SORT & SKIP
C	
D	
E	
F	
G	
H	
I	
J	

INPUT-OUTPUT SELECTORS

PICK UP	S#	SELECT	COMMON	NON-SELECT
	1a			
	1b			
	1c			
	2a			
	2b			
	2c			
	3a			
	3b			
	3c			
	4			
	5a			
	5b			
	5c			
	6a			
	6b			
	6c			
	7a			
	7b			
	7c			
	8a			
	8b			
	8c			
	9a			
	9b			
	9c			
	10a			
	10b			
	10c			

INPUT TRANSFER

IN (from)	EVEN
NO (to)	
EVEN (to)	IN
ALL (to)	

PROGRAM SELECTS

NO.	IN FROM	POWER OUT TO
1		
2		

REMARKS: * NOTE DATA STORED ON DRUM USING SAME COLUMNAR SEQUENCE.

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM

STORAGE ASSIGNMENT CHART

APPLICATION: DAILY PAYROLL PROGRAM NO.: IV

EMPLOYEE URA
(A1-A4 SELECT)

LARGE CAPACITY DRUM
FIELD ASSIGNMENT

ADDRESS FROM 00 00 00
TO 03 99 14

SYM	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
F 0	CLOCK NUMBER & DEPARTMENT						5	4	3	2	1	0	+
F 1	DAY RATE								9	8	7	6	+
F 2	REGULAR EARNINGS WTD							14	13	12	11	10	+
F 3	SHIFT PAY WTD									17	16	15	+
F 4	OUTSIDE WORK PAY WTD									20	19	18	+
F 5	GROSS EARNINGS YTD						26	25	24	23	22	21	+
F 6	FICA YTD								30	29	28	27	+
F 7	FWT YTD						36	35	34	33	32	31	+
F 8	ELAPSED HOURS WTD									39	38	37	+
F 9													
F10													
F11													
F12	NOTE: OVERFLOW URA												
F13	F_0 = DS CH URA XX XX 00												
F14	F_1-F_8 = SPACES												
F15													
F16													
F17													
F18													
F19													

OPERATION URA
(A1-A4 NON-SELECT)

LARGE CAPACITY DRUM
FIELD ASSIGNMENT

ADDRESS FROM 04 00 00
TO 08 99 14

SYM	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
F 0	OPERATION No.						5	4	3	2	1	0	+
F 1	STD. HOURS PER C						11	10	9	8	7	6	+
F 2	OPERATION No.						17	16	15	14	13	12	+
F 3	STD HOURS PER C						23	22	21	20	19	18	+
F 4	OPERATION No. (+ STORED IN CH 24)						30	29	28	27	26	25	24
F 5	STD. HOURS PER C						36	35	34	33	32	31	+
F 6													
F 7													
F 8	NOTE: O'FLOW ADDRESS IS												
F 9	IN F_4 of LAST URA.												
F10	F_0 = OPER. No.												
F11	F_1 = STD HRS.												
F12	F_2 = OPER. No.												
F13	F_3 = STD HRS.												
F14	F_4 = DS CH URA SN. XX XX 00 —												
F15	F_5 = SPACES												
F16													
F17													
F18													
F19													

LARGE CAPACITY DRUM
FIELD ASSIGNMENT SELECTORS

PICK-UP FROM	GROUND TO	A#	SELECT	COMMON	NON-SELECT
PS0 2	CG	1a	+ CH 10	APPLIED + F_2	+ CH 12
		1b	+ CH 15	APP. + F_3	+ CH 18
		1c	+ CH 18	APP. + F_4	—
		1d	+ CH 21	APP + F_5	+ CH 31
PS0 2	CG	2a	+ CH 27	APP + F_6	—
		2b	+ CH 31	APP + F_7	—
		2c	F_1	CH 9	CH 10
		2d	CH 12	CH 11	F_1
PS0 2	CG	3a	F_2	CH 14	CH 15
		3b	F_3	CH 17	F_2
		3c	F_4	CH 20	CH 21
		3d	CH 24	CH 23	F_3
PS0 2	CG	4a	F_5	CH 26	CH 27
		4b	F_6	CH 30	F_4
		4c	F_7	CH 36	F_5
		4d			
		5a			
		5b			
		5c			
		5d			
		6a			
		6b			
		6c			
		6d			
		7a			
		7b			
		7c			
		7d			
		8a			
		8b			
		8c			
		8d			

INTERMEDIATE STORAGE FIELD ASSIGNMENT
TRACK 50

SYM	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
50													
51													
52													
53													
54													
55													
56													
57													
58													
59													

INTERMEDIATE STORAGE FIELD ASSIGNMENT
TRACK 10

SYM.	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													

TRACK 20

20	CONSTANT 5									0	0	5	+
21	SPACES	—	—	—	—	—	—	—	—	—	—	—	+
22	CONSTANT .06										0	6	+
23	CONSTANT .09										0	9	+
24													
25													
26													
27													
28													
29													

TRACK 30

30													
31													
32													
33													
34													
35													
36													
37													
38													
39													

TRACK 40

40													
41													
42													
43													
44													
45													
46													
47													
48													
49													

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM

PROGRAM PLANNING CHART

APPLICATION: DAILY PAYROLL PROGRAM NO.: IV

STEP NO.	INPUT UNIT NO.	CARD NO.	VALUE 1 DESCRIPTION	SYM	SH	PROCESS	VALUE 2 DESCRIPTION	SYM	SH	RESULT DESCRIPTION	SYM	SH	NEXT STEP	VALUE 1 AS STORED	VALUE 2 AS STORED	UNSHIFTED RESULT	STORED RESULT
1			OPERATION NO.	002		+	SPACES	21		TEST: +INCENT : ADDRESS 0 DAY WORK	ADR	2L	BR1	XXXXXX+/0	-----------+	XXXXXX+/0	XXXX00
2			OPERATION NO.	002		ECS	OPERATION NO.	002		OPER. NO.	CSF0 CSF2		ECS1	XXXXXX+	XXXXXX+	XXXXXX+	
3			OPERATION NO.	002		ECS	OPERATION NO.	002		OPER. NO.	CSF4		ECS2	XXXXXX+	XXXXXX+	XXXXXX+	
4			O'FLOW ADDRESS	F4		+	SPACES	21		TEST: − O'FLOW ADD. 0/+ NO O'FLOW	ADR		BR2	XXXX00+/−/0	-----------+	XXXX00+/−/0	XXXX00
5			OPERATION NO.	002		C	1st OPER. NO.	F0		TEST: 0 - 1ST OPER. −/+ 2ND OPER.	—		BR3	XXXXXX+	XXXXXX+		
6			STD. HRS. PER C	F1		X	PCS. PROD.	006		STD. HRS. PROD. N.R.	002		OE2	XXXXXX+	XXXXXX+	XXXXXXX+	XXXXXXX+
7			STD. HRS. PER C	F3		X	PCS. PROD.	006		STD. HRS. PROD. N.R.	002		OE2	XXXXXX+	XXXXXX+	XXXXXXX+	XXXXXXX+
8			STD. HRS. PER C	F5		X	PCS. PROD.	006		STD. HRS. PROD. N.R.	002		OE2	XXXXXX+	XXXXXX+	XXXXXXX+	XXXXXXX+
9			STD. HRS. PROD. N.R.	002	3R	+	CONSTANT 5	20		STD. HRS. PROD. RDD.	002	1R	10	XXXXXXX+	.05+	XXXXX+	XXX+
10			CLOCK + DEPT. NO.	000		T		—		ADDRESS	ADR	2L	FD1	XXXXXX+		XXXXXX+	XXXX00
11			CLOCK + DEPT. NO	000		ECS	CLOCK + DEPT. NO.	000		CLOCK + DEPT. NO.	CSF0		ECS3	XXXXXX+	XXXXXX+	XXXXXX+	
12			DAY RATE	F1		C	SPACES	21		TEST: + NO O'FLOW 0 POSS. O'FLOW	—		BR4	XXXX+	-----------+		
13			O'FLOW ADDRESS	F0		+	SPACES	21		TEST: + O'FLOW ADD. 0 NO O'FLOW	ADR		BR5	XXXX00+/0	-----------+	XXXX00+/0	XXXX00
14			ELAPSED HRS.	004		X	.06 OR .09	ST 36		SHIFT PAY N.R.	008		15	XXX+	.0X+	XXXX+	XXXX+
15			SHIFT PAY N.R	008		+	.005	20		SHIFT PAY RDD.	008	1R	16	XXXX+	.005+	XXXX+	XXX+
16			SHIFT PAY W.T.D.	F3		+	SHIFT PAY	008		NEW SHIFT PAY WTD	F3		17	XXX+	XXX+	XXX+	XXX+
17			DAY RATE	F1		X	ELAP. OR STD. HRS.	CT 4a		EARNINGS N.R.	001		18	XXXX+	XXX+	XXXXXX+	XXXXXX+
18			EARNINGS N.R.	001	1R	+	.005	20		EARNINGS RDD.	001		CT 4b	XXXXXX+	.005+	XXXXX+	XXXX+
19			ELAPSED HRS.	004		C	STD. HRS.	002		TEST: + ELAP HRS. > STD. −/0 ELAP. HRS. ≤ STD.	—		BR6	XXX+	XXX+		
20			DAY RATE	F1		X	ELAPSED HRS.	004		GUARANTEED PAY N.R.	003		21	XXXX+	XXX+	XXXXXX+	XXXXXX+
21			GUARANTEED PAY N.R.	003	1R	+	.005	20		GUARANTEED PAY RDD.	003	1R	22	XXXXXX+	.005+	XXXXX+	XXXX+
22			GUARANTEED PAY RDD.	003		−	EARNINGS	001		MAKE-UP PAY	000	6L	23	XXXX+	XXXX+	XXXX+	XXXX000000+
23			REG. EARNINGS WTD	F2		+	GUARANTEED PAY	003		NEW REG. EARN. WTD	F2		24	XXXXX+	XXXX+	XXXXX+	XXXXX+
24			GUAR. PAY (Σ EARN + MAKE)	003		+	SHIFT PAY	008		Σ EARN. + MAKE-UP + SHIFT	003		27	XXXX+	XXX+	XXXX+	XXXX+
25			REG. EARNINGS WTD	F2		+	EARNINGS	001		NEW REG EARN. WTD	F2		26	XXXXX+	XXXX+	XXXXX+	XXXXX+
26			EARNINGS	001		+	SHIFT PAY	008		Σ EARN. + SHIFT	003		27	XXXX+	XXX+	XXXX+	XXXX+
27			ELAPSED HRS. WTD	F8		+	ELAPSED HRS.	004		NEW ELAP. HRS. WTD	F8		CT1	XXX+	XXX+	XXX+	XXX+
28			ELAP. HRS. = OUT. PAY	004		+	Σ EARN. + MAKE-UP + SHIFT	003		TOTAL PAY	003		29	XXX+	XXXX+	XXXX+	XXXX+
29			ELAP. HRS. = OUT. PAY	004		+	OUTSIDE PAY WTD	F4		NEW OUT. PAY WTD	F4		30	XXX+	XXX+	XXX+	XXX+
30			ELAP. HRS. = OUT. PAY	004	3L	T		—		OUTSIDE PAY	004		W1	XXX+		XXX000+	XXX000
31																	
32																	
33																	
34																	
35																	
36																	
37																	
38																	
39																	
40																	
41																	
42																	
43																	
44																	
45																	
46																	
47																	
48																	

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

VI - 55

UNIVAC FILE-COMPUTER SYSTEM

FUNCTION CONTROL CHART

APPLICATION: DAILY PAYROLL   PROGRAM NO.: IV

DEMAND UNITS

NO.	TYPE OF UNIT	TEST IN FROM	TEST OUT TO NOT READY	TEST OUT TO READY	DEMAND IN FROM	DEMAND OUT TO INPUT	DEMAND OUT TO OUTPUT
1	150 CPM				START; OE 3	IN STEP 1	
2							
3							
4							
5							
6							
7							
8							

SCAN UNITS

NO.	TYPE OF UNIT	TO SCAN OUT	NO.	TYPE OF UNIT	TO SCAN OUT	NO.	TYPE OF UNIT	TO SCAN OUT
1			9			17		
2			10			18		
3			11			19		
4			12			20		
5			13			21		
6			14			22		
7			15			23		
8			16			24		

CODE DISTRIBUTOR

FROM V1 OF STEPS	
FROM V2 OF STEPS	
FROM R OF STEPS	

NO.	TO	NO.	TO
00		26	
01		27	
02		28	
03		29	
04		30	
05		31	
06		32	
07		33	
08		34	
09		35	
10		36	
11		37	
12		38	
13		39	
14		40	
15		41	
16		42	
17		43	
18		44	
19		45	
20		46	
21		47	
22		48	
23		49	
24		>49	
25			

FROM OUT OF FUNCTION:

NO.	TO	NO.	TO
0		5	
1		6	
2		7	
3		8	
4		9	

INPUT CONTROL LINES

NO.	TO
a	PU T3 (9/45)
b	PU T2 (3/45)
c	PU T1 (1/46)
d	PU TH (3/46)
e	
f	
g	
h	
i	
i	
k	
l	

OUTPUT CONTROL LINES

NO.	FROM
A	OE 3
B	OE 1
C	
D	
E	
F	
G	
H	
I	
J	

START TO: DI #1

SCAN OUTS

NO.	TO

ERROR CHECKS

FROM	TO
ARITH.	STEP REPEAT
±O'FLOW	STOP
÷O'FLOW	
PARITY	STOP
ADDRESS	STOP

INDICATORS

NO.	FROM
1	
2	
3	
4	

IND. SWITCH	FROM	TO
NO CHECK		

READ UNIT RECORD

NO.	IN FROM	OUT TO
1		
2		
3		
4		
5		
6		
7		
8		

WRITE UNIT RECORD

NO.	IN FROM	OUT TO
1	NS T1 STEP 30	OE 3
2		
3		
4		
5		
6		
7		
8		

STEP CLEAR

IN	OUT

CLEAR PROG. SEL.

STEP REPEAT

IN FROM: ARITH. ERROR

CHANNEL SEARCH EQUAL

NO.	FROM OUT OF STEP NO.	TO IF = V1	TO IF = V2	TO IF ≠
1	2	STEP 5	—	STEP 3
2	3	STEP 8	—	STEP 4
3	11	CT3a	--	STEP 12
4				

BRANCHING

NO.	IN FROM STEP	TO IF +	TO IF −	TO IF O
1	1	STEP 2	—	OE 4
2	4	OE 1	STEP 2	OE 1
3	5	STEP 7	STEP 7	STEP 16
4	12	OE 1	—	STEP 13
5	13	STEP 11	—	OE 1
6	19	STEP 20	STEP 25	STEP 25
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				

CHANNEL SEARCH UNEQUAL

NO.	FROM OUT OF STEP NO.	TO IF =	TO IF ≠
1			
2			
3			
4			

OUT EXPANDERS

NO.	IN	OUT	OUT
1	oBR 2+5 +BR 2+4	OPC B	OE 3
2	FROM STEPS 6, 7 + 8	IN PS 2	IN STEP 9
3	W1, OE1	OPCA, DOPS2	DI #1
4	oBR 1	IN PS 2	IN STEP 10
5			
6			
7			
8			

FUNCTION DELAYS

NO.	IN 1	IN 2	OUT
1	DPS 2	FROM STEP 10	IN STEP 11
2			
3			

ALTERNATE SWITCHES

NO.	SELECT	COMMON	NON-SELECT
1			
2			
3			
4			
5			
6			

FUNCTION SEQUENCE

NO.	SET	PROBE	OUT
1			
2			

SYMBOLS FOR PROGRAMMING:

Processes:	PR
Transfer	T
Addition	+
Subtraction	–
Multiplication	X
Division	÷
Left Zero Elim	LZE
Compare	C
Channel Search =	ECS
Channel Search ≠	UCS

SHIFTS:

Shift 2 Left	2L
Shift 2 Right	2R
Expand Prod. Shift	EXS

SELECTORS:

Computor Selector	$T_1 \to T_{48}$
Drum Selector	$A_1 \to A_8$
Input-Output Selectors	$S_1 \to S_{10}$
Common	CT(#) or CA (#)
Select	ST (#) or SA (#)
Non-Select	NST(#) or NSA (#)
Pick-Up	PUT(#) or PUA (#)
Program Selector	$PS_1 \to PS_{16}$
Delay of PS	$DPS_1 \to DPS_{10}$
Imm. of PS	$IPS_{11} \to IPS_{16}$
Power of PS	$PSO_1 \to PSO_{16}$
Drop Out of PS	$PSDO_1 \to PSDO_{16}$
Ground of Sel	GT (#)
Computer Ground	CG
Demand Ground	$DG_1 \to DG_8$
Scan Ground	$SG_1 \to SG_6$
Selector Hold	SH

Function Delay	FD
In FD 1	$FD1_1 \to FD1_3$
In FD 2	$FD2_1 \to FD2_3$
Out FD	$FDO_1 \to FDO_3$

STORAGES:

Input-Output	000→009
Intermediate	10→59
Revolver	$F_0 \to F_{19}$
Channel Search Fields	$CSF_0 \to CSF_{19}$

DEMAND UNITS:

Test In	TST (#)
Test Out Ready	TSTR (#)
Test Out Not Ready	TSTN (#)
Demand In	DI (#)
Demand Out Input	DOI (#)
Demand Out Output	DOO (#)

Scan Out	$SO_1 \to SO_6$
Program Step In	In ST (#)
Program Step Out	From ST (#)
Out Expanders	$OE_1 \to OE_8$
In Branching	$BR_1 \to BR_{18}$
+ Branch Out	$+BR_1 \to BR_{18}$
– Branch Out	$-BR_1 \to BR_{18}$
o Branch Out	$oBR_1 \to BR_{18}$
Read Unit Record	$R_1 \to R_8$
Write Unit Record	$W_1 \to W_8$
Equal Channel Search In	ECSI (#)
Equal Channel Search Out=V_1	$ECSV_1$ (#)
Equal Channel Search Out=V_2	$ECSV_2$ (#)
Equal Channel Search Out ≠	ECS ≠ (#)
Unequal Channel Search In	UCSI (#)
Unequal Channel Search Out =	UCS = (#)
Unequal Channel Search Out ≠	UCS ≠ (#)
Program Address Counter	PAK
Address Register	ADR

Code Distributor	CD
In CD V_1	CDV_1
In CD V_2	CDV_2
In CD R	CDR
Out CD 00 to > 49	$CD_{00} \to CD > 49$
In CD Pulse In	CDP
Out CD Pulse Out	CDP_0--CDP_9
Function Sequence Set	$FSS_1 \to FSS_2$
Function Sequence Probe	$FSP_1 \to FSP_2$
Function Sequence Out	$FSO_1 \to FSO_2$

ALTERNATE SWITCHES:

Common	$SW_1C \to SW_6C$
Select	$SW_1S \to SW_6S$
Non-Select	$SW_1NS \to SW_6N_6$

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

VI - 56

UNIVAC FILE-COMPUTER SYSTEM

COMPUTER SELECTOR CONTROL CHART

APPLICATION: DAILY PAYROLL PROGRAM NO. IV

PICK-UP FROM	GROUND TO	T#	SELECT	COMMON	NON-SELECT
1PCc 1/46	CG	1	IN STEP 28	FROM STEP 27	W1
1PCb 3/45	CG	2	IN STEP 14	NS T3a	IN STEP 17
1PCa 9/45	CG	3a	IN STEP 14	ECS V1 #3	C T2
		3b	22 (.06)	V2 ST. 14	23 (.09)
1PCd 3/46	CG	4a	004	V2 ST. 17	002
		4b	IN STEP 25	FROM STEP 18	IN STEP 19
		5a			
		5b			
		6a			
		6b			
		6c			
		6d			
		7			
		8			
		9a			
		9b			
		10a			
		10b			
		11a			
		11b			
		12a			
		12b			
		12c			
		12d			
		13			
		14			
		15a			
		15b			
		16a			
		16b			
		17a			
		17b			
		18a			
		18b			
		18c			
		18d			
		19			
		20			
		21a			
		21b			
		22a			
		22b			
		23a			
		23b			
		24a			
		24b			
		24c			
		24d			

PICK-UP FROM	GROUND TO	T#	SELECT	COMMON	NON-SELECT
		25			
		26			
		27a			
		27b			
		28a			
		28b			
		29a			
		29b			
		30a			
		30b			
		30c			
		30d			
		31			
		32			
		33a			
		33b			
		34a			
		34b			
		35a			
		35b			
		36a			
		36b			
		36c			
		36d			
		37			
		38			
		39a			
		39b			
		40a			
		40b			
		41a			
		41b			
		42a			
		42b			
		42c			
		42d			
		43			
		44			
		45a			
		45b			
		46a			
		46b			
		47a			
		47b			
		48a			
		48b			
		48c			
		48d			

PROGRAM SELECTS

PS#	IN FROM	DELAY OUT TO	D. O. FROM	POWER TO
1	OE 2, OE 4	IN FD 1	OE 3	PU A1,2,3 & 4
2				
3				
4				
5				
6				
7				
8				
9				
10				
PS#	**IN FROM**	**IMMEDIATE OUT TO**	**D. O. FROM**	**POWER TO**
11				
12				
13				
14				
15				
16				

REMARKS:

Program Comments

Before Step (11) - Drum field selectors are picked up because two unit records are used in this run and they each have different field layouts. To insure using the proper fields in calculations, the field layout must be changed for individual processing.

6. PROBLEM V

1. Statement of Problem

This problem is the processing of the weekly payroll for 5000 employees. In problem IV daily job cards were processed to reflect the employee's daily activity in his URA. The employee unit record used in problem IV is again used in this problem. An employee master deck is used to produce this weekly payroll. Each card has, in addition to the employee name, the employee's clock number, department number, number of exemptions and other deductions. This master deck is then placed in the UF-C for processing.

The employee clock number, department number, number of exemptions and other deductions are the only factors in this card used by the computer in arriving at an employee's paycheck. The operations performed by the computer are to calculate: (Note WTD = week to date, YTD = year to date, FWT = federal withholding tax.)

1) Regular Pay WTD - This has already been summarized in his unit record, F2.

2) Overtime Premium Pay - Since an employee will have various rates of pay for different jobs during the week, the overtime rate is an average of his rates. Standard work week is 40 hours. Overtime hours = elapsed hours - 40. The formula used to calculate overtime pay is:

$$\left(\frac{\text{Req. Pay} + \text{Shift} + \text{Outside Work}}{\text{Elapsed Hours}}\right) \left(\frac{\text{Overtime Hours}}{2}\right)$$

3) Shift Pay - Accumulated in field F3.

4) Outside Work - Accumulated in field F4.

5) Gross Pay - Regular pay + shift pay + outside work + overtime pay.

6) FICA This Week - 2% of the employee's gross pay to $4200.00.

7) Withholding Tax This Week - This is equal to (Gross pay - the number of exemptions x 13) 18%.

8) Net Pay - Gross pay minus FICA, Withholding Tax and other deductions.

9) Withholding Tax YTD - FWT to date + this week's FWT.

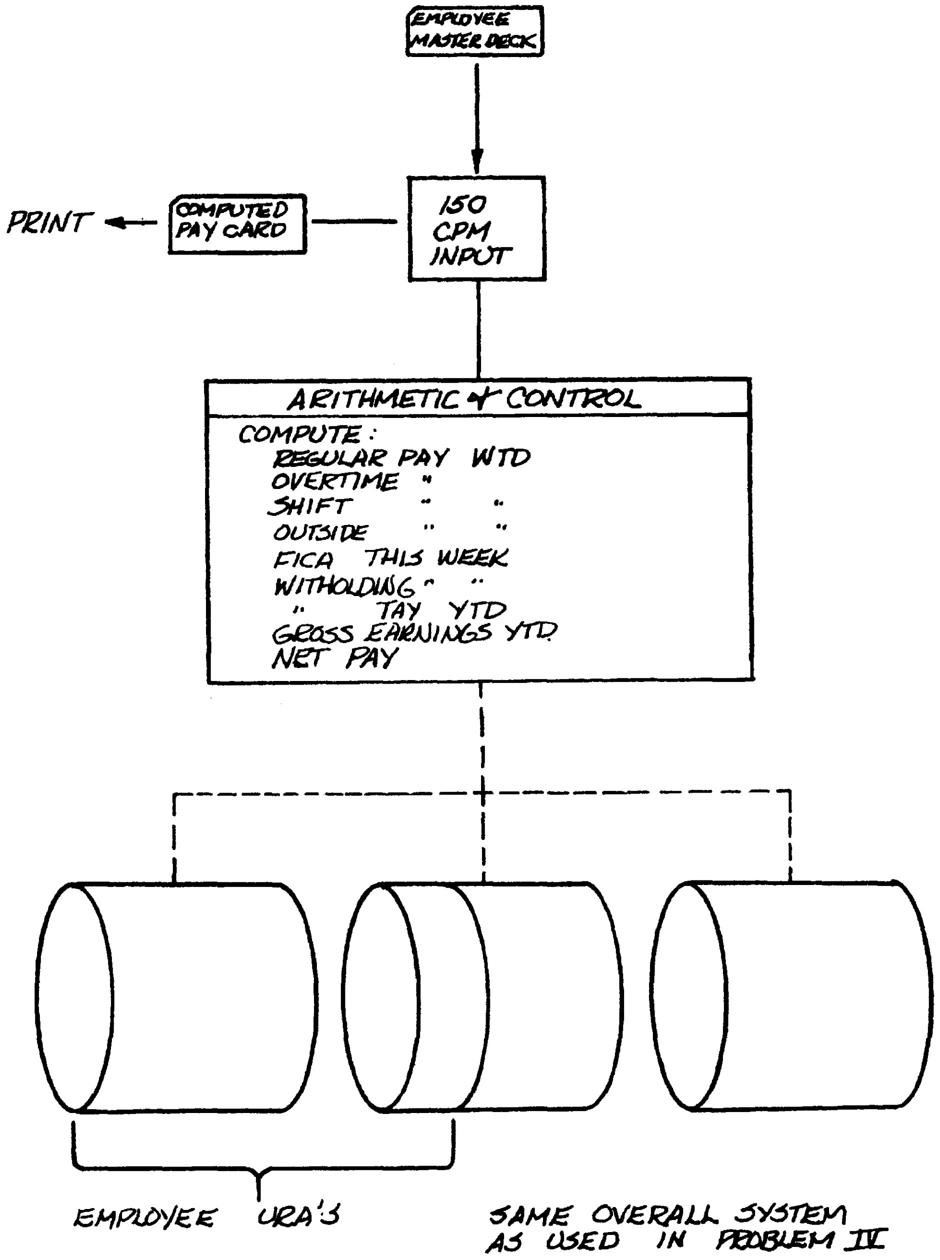
10) Gross Earnings YTD - Gross earnings to date + this week's gross earnings.

11) FICA YTD - FICA to date + this week's FICA.

2. Machine Specifications

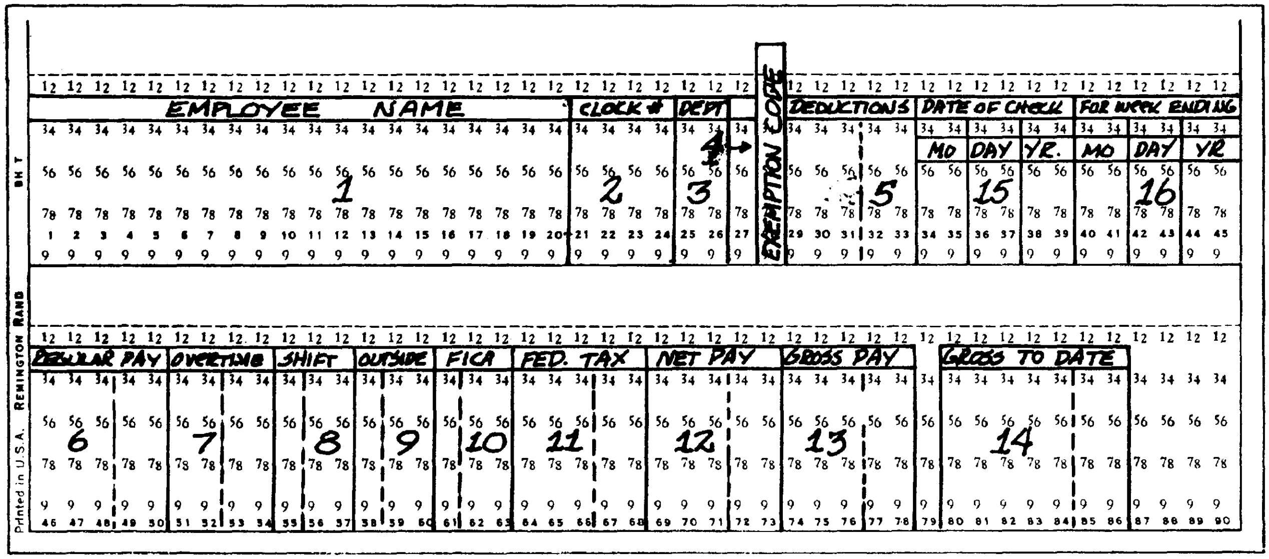
1 - 150 CPM Card Sensing Punching Unit
2 - Large Capacity drums with 40 digit URA length
1 - Arithmetic & Control Unit (external programming)

3 GENERAL PROCEDURE FLOW CHART



4. Input - Output Data & Definition

The input source medium are 90 column punched cards, entered in random sequence and punched in the following form:



Definitions:

Input

(1) Employee name (col 1-20)

(2) Clock No. (col 21-24). Col 23 & 24 are used for Channel number.

(3) Dept. No. (col 25-26) - These two digits are used for drum section number.

(4) Number of exemptions (col 28) - The number of federal withholding tax exemptions. 10 exemptions is punched as an over-capacity zero.

(5) Deductions (col 29-33) - The sum of all payroll deductions except taxes.

Output

(6) Regular Pay WTD (col 46-50) - The employee's accumulated regular pay for this week. Item in URA cleared after processing.

(7) Overtime Pay WTD (col 51-54) - This is the employee's overtime pay.

(8) Shift Pay WTD (col 55-57) - Accumulated shift pay for this week. Item in URA cleared after processing.

(9) Outside Work Pay WTD (col 58-60) - Accumulated outside work for this week. Item in URA cleared after processing.

(10) FICA This Week (col 61-63) - Employee's FICA deduction this week.

(11) Fed. Withholding Tax This Week (col 64-68) - Withholding tax deduction this week.

(12) Net Pay (col 69-73) - Gross pay less all taxes and other deductions. If gross pay is not sufficient to cover tax deductions plus the total of other deductions, punch gross pay minus tax deductions as net pay and also punch a zero in column 89 to identify this card.

(13) Gross Pay (col 74-78) - The employee's gross earnings for the week.

(14) Gross Earnings YTD (col 80-86) - Accumulation of Gross Earnings on a yearly basis.

The above input-output data is assigned to the following input-output storage locations.

Input Storage	Sym.	Description
	000	Clock number and department
	002	Other deductions
Output Storage	000	Outside work pay
	001	Gross to date
	002	Net Pay
	003	Regular Pay
	004	Overtime Pay
	005	Shift Pay
	006	FWT
	007	Gross Pay
	008	FICA

5. Drum Stored Data and Definition

a) Same as problem IV - Employee's Unit Record.

LARGE CAPACITY DRUM REVOLVER
UNIT RECORD AREA - FIELD ASSIGNMENT

Sym.	Description	Digits	Sign
F0	Clock & Dept. No.	xxxxxx.	+
F1	Day Rate	x.xxx	+
F2	Reg. Earning WTD	xxx.xx	+
F3	Shift Pay WTD	x.xx	+
F4	Outside Work	x.xx	+
F5	Gross Earnings YTD	xxxx.xx	+
F6	FICA YTD	xx.xx	+
F7	Fed. W. Tax YTD	xxxx.xx	+
F8	Elapsed Hours WTD	xx.x	+

b) Definition - Same as problem IV.

The overflow address is located in URA #14 of overflow channels.

F0 = Drum section & channel number of overflow channel i.e.

xx	xx	00
DS	CH	URA

F1 - F8 = Spaces

The following constants are stored in intermediate storage:

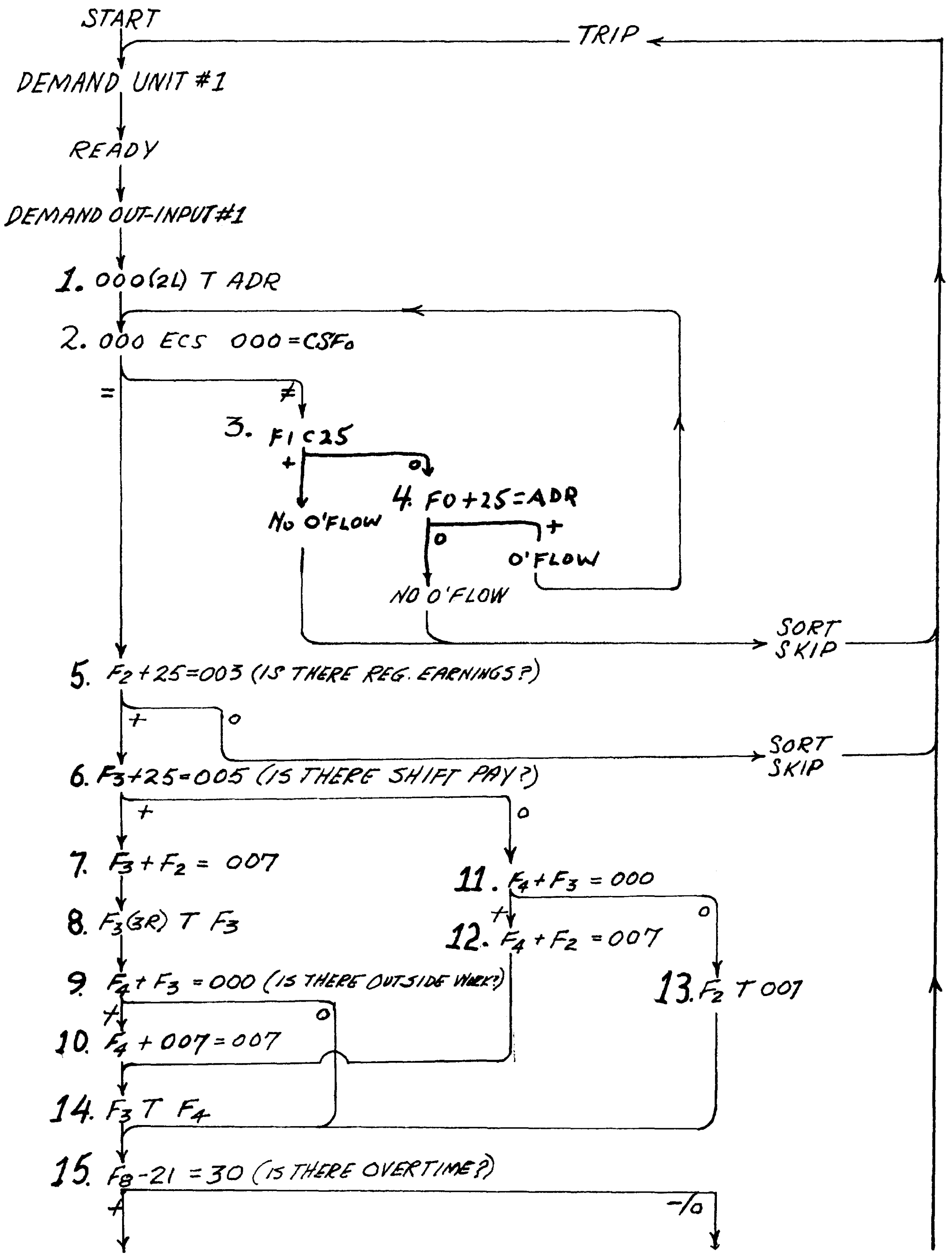
Intermediate Storage #	-	Constant
10	-	2.335 +
11	-	4.675 +
12	-	7.015 +
13	-	9.355 +
14	-	11.695 +
15	-	14.035 +
16	-	16.375 +
17	-	18.715 +
18	-	21.055 +
19	-	23.395 +
20	-	.005 -
21	-	40.0 +
22	-	84.00 +
23	-	.02 +
24	-	5 +
25	-	Spaces
26	-	.18 +

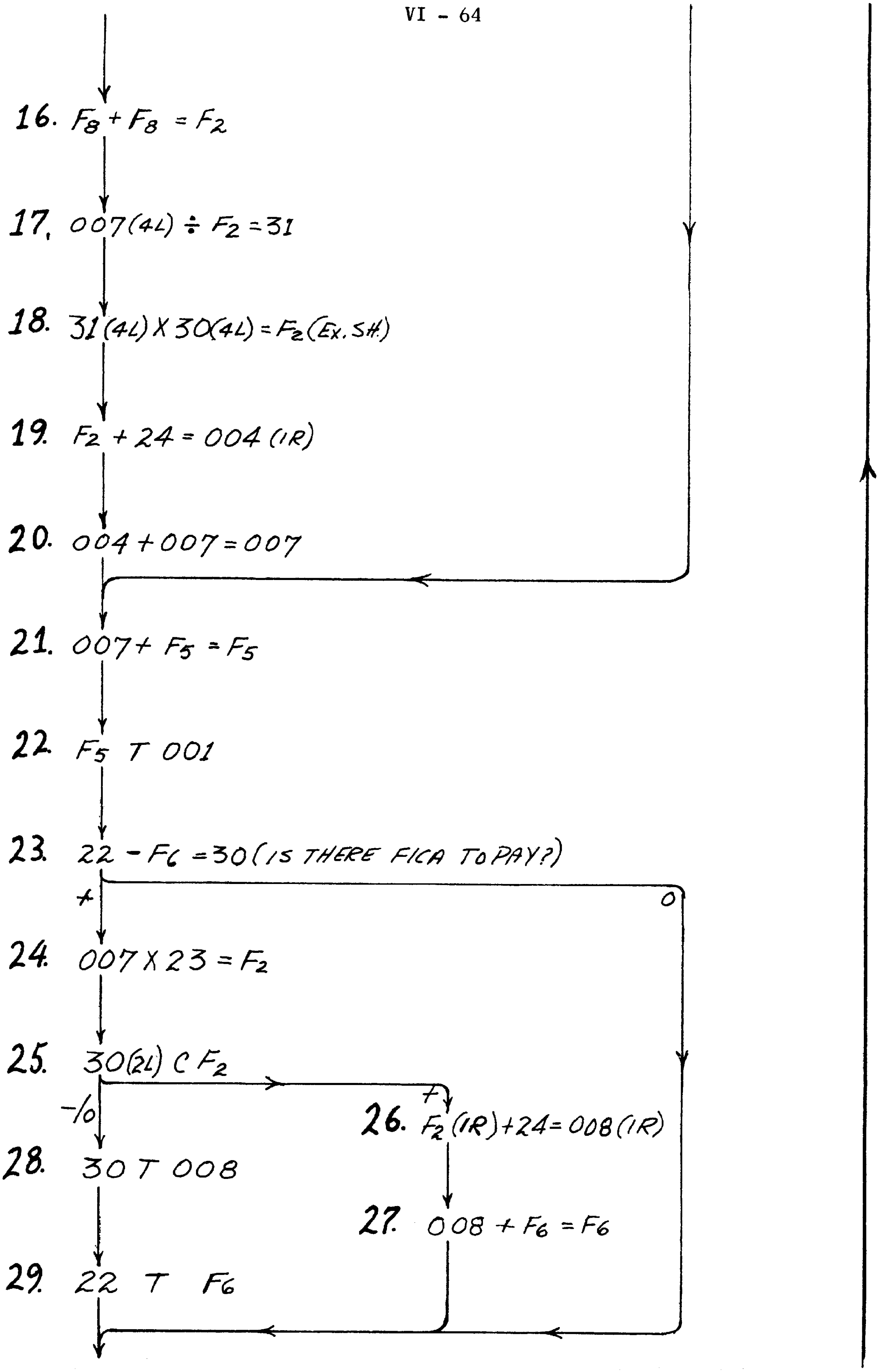
NOTE:

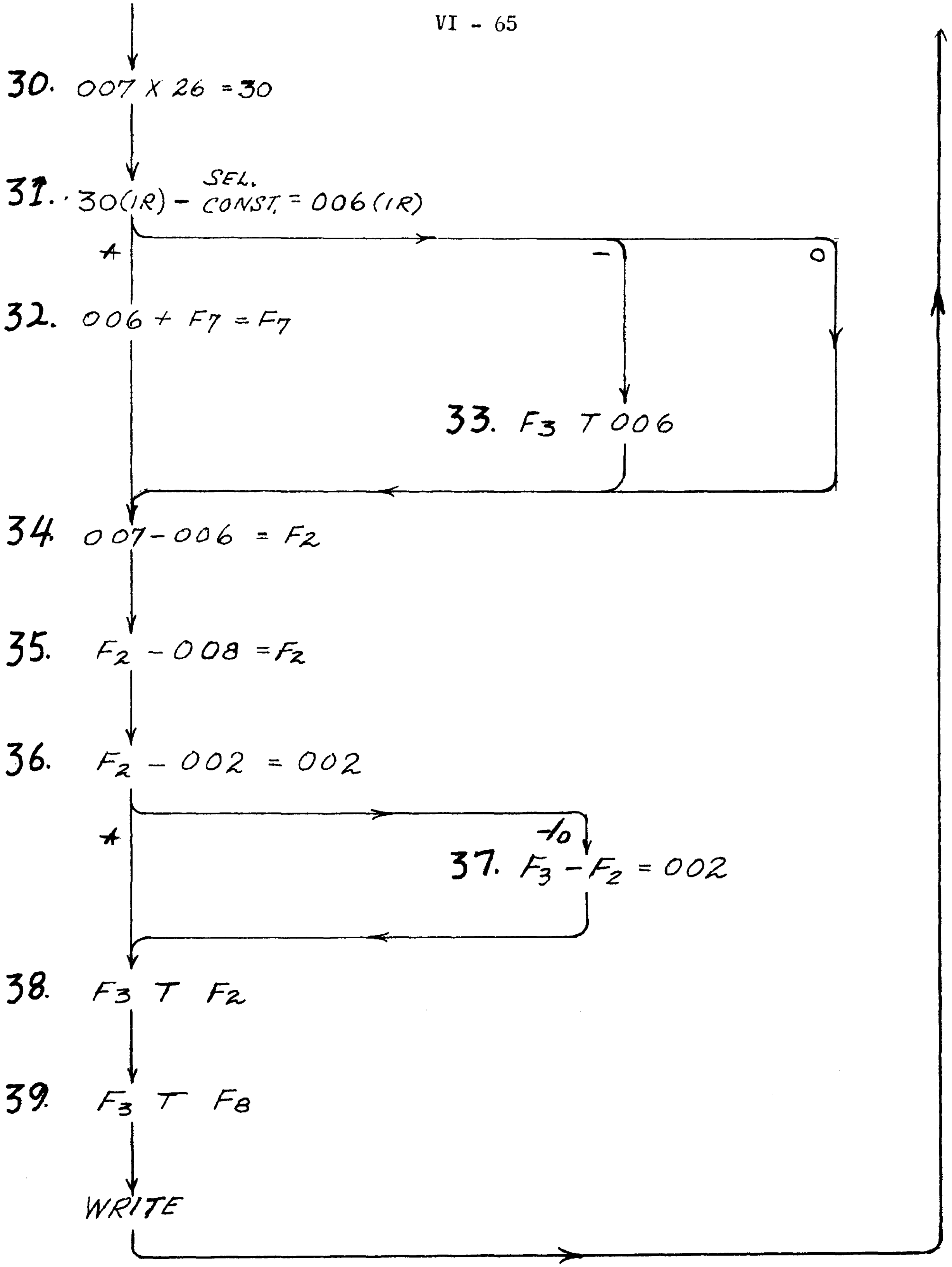
Constants in storages 10-20 are formed as follows:

[18% x (No. of Exemptions x $13.00)] - .005

6 Program Flow Chart




16. $F_8 + F_8 = F_2$
17. $007(4L) \div F_2 = 31$
18. $31(4L) \times 30(4L) = F_2$ (Ex. SH.)
19. $F_2 + 24 = 004$ (1R)
20. $004 + 007 = 007$
21. $007 + F_5 = F_5$
22. F_5 T 001
23. $22 - F_6 = 30$ (IS THERE FICA TO PAY?)
+
0
24. $007 \times 23 = F_2$
25. 30(2L) C F_2
+
-/0
26. F_2(1R)$+24 = 008$(1R)
28. 30 T 008
27. $008 + F_6 = F_6$
29. 22 T F_6


30. 007 X 26 = 30
SEL.
31. 30(IR) − CONST. = 006(IR)
+
−
0
32. 006 + F7 = F7
33. F3 T 006
34. 007 − 006 = F2
35. F2 − 008 = F2
36. F2 − 002 = 002
+
−/0
37. F3 − F2 = 002
38. F3 T F2
39. F3 T F8
WRITE

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

VI - 66

UNIVAC FILE-COMPUTER SYSTEM

150 CPM INPUT-OUTPUT CHART

APPLICATION: WEEKLY PAYROLL PROGRAM NO.: V

66-1

INPUT-OUTPUT STORAGE FIELD ASSIGNMENT

DEMAND UNIT NO. 1 OR SCAN UNIT NO. ______

INPUT FIELD ASSIGNMENT

SYM.	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN.	SIGN CTL.
000	CLOCK NUMBER & DEPARTMENT						21	22	25	26	23	24	+	
001														
002	TOTAL-OTHER DEDUCTIONS							29	30	31	32	33	+	
003														
004														
005														
006														
007														
008														
009														

OUTPUT FIELD ASSIGNMENT

SYM.	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN.	SIGN CTL.
000	OUTSIDE WORK									58	59	60	+	
001	GROSS EARNINGS TO DATE						81	82	83	84	85	86	+	
002	NET PAY							69	70	71	72	73	±	0/89
003	REGULAR PAY							46	47	48	49	50	+	
004	OVERTIME PREMIUM PAY								51	52	53	54	+	
005	SHIFT PAY									55	56	57	+	
006	FEDERAL WITHHOLDING TAX							64	65	66	67	68	+	
007	GROSS PAY							74	75	76	77	78	+	
008	FICA									61	62	63	+	
009														

INPUT CONTROL LINES

SYM.	FROM
a	0/28
b	1/28
c	3/28
d	5/28
e	7/28
f	9/28
g	
h	
i	
j	
k	
l	

OUTPUT CONTROL LINES

SYM.	TO
A	TRIP
B	SORT & SKIP
C	
D	
E	
F	
G	
H	
I	
J	

INPUT-OUTPUT SELECTORS

PICK UP	S#	SELECT	COMMON	NON-SELECT
	1a			
	1b			
	1c			
	2a			
	2b			
	2c			
	3a			
	3b			
	3c			
	4			
	5a			
	5b			
	5c			
	6a			
	6b			
	6c			
	7a			
	7b			
	7c			
	8a			
	8b			
	8c			
	9a			
	9b			
	9c			
	10a			
	10b			
	10c			

INPUT TRANSFER

IN (from)	EVEN
NO (to)	
EVEN (to)	IN
ALL (to)	

● PROGRAM SELECTS

NO.	IN FROM	POWER OUT TO
1		
2		

REMARKS:

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM

STORAGE ASSIGNMENT CHART

APPLICATION: WEEKLY PAYROLL PROGRAM NO.: V

LARGE CAPACITY DRUM FIELD ASSIGNMENT

ADDRESS FROM 000000
TO 039914

SYM	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
F 0	CLOCK NUMBER & DEPARTMENT						5	4	3	2	1	0	+
F 1	DAY RATE								9	8	7	6	+
F 2	REGULAR EARNINGS WTD							14	13	12	11	10	+
F 3	SHIFT PAY W.T.D.									17	16	15	+
F 4	OUTSIDE WORK PAY WTD									20	19	18	+
F 5	GROSS EARNINGS YTD						26	25	24	23	22	21	+
F 6	FICA YTD								30	29	28	27	+
F 7	FWT YTD						36	35	34	33	32	31	+
F 8	ELAPSED HOURS WTD									39	38	37	+
F 9													
F10													
F11													
F12	NOTE: OVERFLOW URA												
F13	$F_0 = \overset{DS}{XX}\ \overset{CH}{XX}\ \overset{URA}{00}$												
F14	$F_1 - F_8$ = SPACES												
F15													
F16													
F17													
F18													
F19													

LARGE CAPACITY DRUM FIELD ASSIGNMENT

ADDRESS FROM
TO

SYM	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
F 0													
F 1													
F 2													
F 3													
F 4													
F 5													
F 6													
F 7													
F 8													
F 9													
F10													
F11													
F12													
F13													
F14													
F15													
F16													
F17													
F18													
F19													

LARGE CAPACITY DRUM FIELD ASSIGNMENT SELECTORS

PICK-UP FROM	GROUND TO	A#	SELECT	COMMON	NON-SELECT
		1a			
		1b			
		1c			
		1d			
		2a			
		2b			
		2c			
		2d			
		3a			
		3b			
		3c			
		3d			
		4a			
		4b			
		4c			
		4d			
		5a			
		5b			
		5c			
		5d			
		6a			
		6b			
		6c			
		6d			
		7a			
		7b			
		7c			
		7d			
		8a			
		8b			
		8c			
		8d			

INTERMEDIATE STORAGE FIELD ASSIGNMENT

TRACK 50

SYM	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
50													
51													
52													
53													
54													
55													
56													
57													
58													
59													

INTERMEDIATE STORAGE FIELD ASSIGNMENT

TRACK 10

SYM.	DESCRIPTION	11	10	9	8	7	6	5	4	3	2	1	SN
10	CONSTANT (1 EXEMPTION)								2	3	3	5	+
11	CONSTANT (2 EX)								4	6	7	5	+
12	CONSTANT (3 EX)								7	0	1	5	+
13	CONSTANT (4 EX)								9	3	5	5	+
14	CONSTANT (5 EX)							1	1	6	9	5	+
15	CONSTANT (6 EX)							1	4	0	3	5	+
16	CONSTANT (7 EX)							1	6	3	7	5	+
17	CONSTANT (8 EX)							1	8	7	1	5	+
18	CONSTANT (9 EX)							2	1	0	5	5	+
19	CONSTANT (10 EX)							2	3	3	9	5	+

TRACK 20

20	CONSTANT (0 EXEMPTION)									0	0	5	−
21	CONSTANT									4	0	0	+
22	CONSTANT								8	4	0	0	+
23	CONSTANT										0	2	+
24	CONSTANT											5	+
25	CONSTANT	–	–	–	–	–	–	–	–	–	–	–	+
26	CONSTANT										1	8	+
27													
28													
29													

TRACK 30

30													
31													
32													
33													
34													
35													
36													
37													
38													
39													

TRACK 40

40													
41													
42													
43													
44													
45													
46													
47													
48													
49													

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

VI - 68

UNIVAC FILE-COMPUTER SYSTEM

PROGRAM PLANNING CHART

APPLICATION: WEEKLY PAYROLL PROGRAM NO.: V

STEP NO.	INPUT UNIT NO.	CARD NO.	VALUE 1			PROCESS	VALUE 2			RESULT			NEXT STEP	VALUE 1 AS STORED	VALUE 2 AS STORED	UNSHIFTED RESULT	STORED RESULT	STEP
			DESCRIPTION	SYM	SH		DESCRIPTION	SYM	SH	DESCRIPTION	SYM	SH		11 10 9 8 7 6 5 4 3 2 1 SN	11 10 9 8 7 6 5 4 3 2 1 SN	22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 SN	11 10 9 8 7 6 5 4 3 2 1 SN	
1	1		CLOCK → DEPT. NO.	000	2L	T				ADDRESS	ADR		2	XXXXXX▲+		XXXXXX00+	XXXX00	1
2	1		CLOCK → DEPT. NO.	000		ECS	CLOCK → DEPT. NO.	000		CLOCK → DEPT. NO.	CSFO		ECS1	XXXXXX▲+	XXXXXX▲+	XXXXXX+		2
3	1		DAY RATE	F1		C	SPACES	25		TEST: + NO O'FLOW 0 POSS. O'FLOW			BR1	X▲XXX+	------------+			3
4	1		O'FLOW ADDRESS	FO		+	SPACES	25		TEST: + O'FLOW ADD. 0 NO O'FLOW	ADR		BR2	XXXX00+	------------+	XXXX000	XXXX00	4
5	1		REG. EARN. WTD	F2		+	SPACES	25		REG. EARN.	003		BR3	XXX▲XX+/0	------------+	XXX▲XX+/0	XXX▲XX+/0	5
6	1		SHIFT PAY WTD	F3		+	SPACES	25		SHIFT PAY	005		BR4	X▲XX+/0	------------+	X▲XX+/0	X▲XX+/0	6
7	1		SHIFT PAY WTD	F3		+	REG. PAY	F2		Σ REG. + SHIFT	007		8	X▲XX+	XXX▲XX+	XXX▲XX+	XXX▲XX+	7
8	1		SHIFT PAY WTD	F3	3R	T				CLEAR SHIFT	F3		9	X▲XX+		------------+	---+	8
9	1		OUTSIDE PAY WTD	F4		+	SPACES	F3					BR5	X▲XX+/0	---+	X▲XX+/0	X▲XX+/0	9
10	1		OUTSIDE PAY WTD	F4		+	Σ REG. + SHIFT	007		Σ SHIFT + REG. + OUT.	007		14	X▲XX+	XXX▲XX+	XXX▲XX+	XXX▲XX+	10
11	1		OUTSIDE PAY WTD	F4		+	SPACES	F3		OUTSIDE PAY	000		BR6	X▲XX+/0	---+	X▲XX+/0	X▲XX+/0	11
12	1		OUTSIDE PAY WTD	F4		+	REG. PAY	F2		Σ REG. + OUT (NO SHIFT)	007		14	X▲XX+	XXX▲XX+	XXX▲XX+	XXX▲XX+	12
13	1		REG. EARN	F2		T				REG. (NO SHIFT OR OUT.)	007		15	XXX▲XX+		XXX▲XX+	XXX▲XX+	13
14	1		SPACES	F3		T				CLEAR OUTSIDE PAY	F4		15	---+		------------+	---+	14
15	1		ELAPSED HRS. WTD	F8		−	40.0	21		TEST: + O.T. HRS. −/0 NO O.T. HRS.	30		BR7	XX▲X+	40▲0+	XX▲X+/0	XX▲X+/0	15
16	1		ELAPSED HRS. WTD	F8		+	ELAPSED HRS. WTD	F8		2 X ELAP. HRS.	F2		18	XX▲X+	XX▲X+	XXX▲X+	XXX▲X+	16
17	1		Σ REG + SHIFT + OUT.	007	4L	÷	2 X ELAP. HRS.	F2		1/2 AVE. HRLY. RATE	31		18	XXX▲XX+	XXX▲X+	▲00XXXX X▲XXXXX+	X▲XXXXX+	17
18	1		1/2 AVE. HRLY. RATE	31	4L	X	O.T. HRS.	30	4L	O.T. PREM. PAY N.R.	F2	EX SH	19	X▲XXXXX+	XX▲X+	X▲XXXXXX X000000000+	XX▲XXX+	18
19	1		O.T. PREM. PAY N.R.	F2		+	.005	24		O.T. PREM. PAY RDD.	004	1R	20	XX▲XXX+	▲005+	XX▲XXX+	XX▲XX+	19
20	1		O.T. PREM. PAY RDD.	004		+	Σ REG. + SHIFT + OUT	007		GROSS PAY	007		21	XX▲XX+	XXX▲XX+	XXX▲XX+	XXX▲XX+	20
21	1		GROSS PAY	007		+	GROSS TO DATE	F5		NEW GROSS TO DATE	F5		22	XXX▲XX+	XXXX▲XX+	XXXX▲XX+	XXXX▲XX+	21
22	1		GROSS PAY TO DATE	F5		T				GROSS TO DATE	001		23	XXXX▲XX+		XXXX▲XX+	XXXX▲XX+	22
23	1		84.00	22		−	FICA TO DATE	F6		TEST: + AMT. TO GO 0 NO FICA	30		BR8	84▲00+	XX▲XX+	XX▲XX+/0	XX▲XX+/0	23
24	1		GROSS PAY	007		X	.02	23		TENTATIVE FICA N.R.	F2		25	XXX▲XX+	▲02+	X▲XXXX+	X▲XXXX+	24
25	1		AMT. FICA TO GO	30	2L	C	TENTATIVE FICA N.R.	F2		TEST: + AMT. > TENT. −/0 AMT ≤ TENT.			BR9	XX▲XX+	X▲XXXX+			25
26	1		TENTATIVE FICA N.R.	F2	1R	+	.005	24		FICA	008	1R	27	X▲XXXX+	▲005+	X▲XXX+	X▲XX+	26
27	1		FICA	008		+	FICA TO DATE	F6		NEW FICA TO DATE	F6		30	X▲XX+	XX▲XX+	XX▲XX+	XX▲XX+	27
28	1		AMT. FICA TO GO	30		T				FICA	008		29	X▲XX+		X▲XX+	X▲XX+	28
29	1		84.00	22		T				NEW FICA TO DATE	F6		30	84▲00+		84▲00+	84▲00+	29
30	1		GROSS PAY	007		X	.18	26		18 % GROSS	30		31	XXX▲XX+	▲18+	XXX▲XXXX+	XXX▲XXXX+	30
31	1		18% GROSS PAY	30	1R	−	TAX EXEMPT. + RDD.	CT1		FWT RDD.	006	1R	BR10	XXX▲XXXX+	XX▲XXX+	XXX▲XXX+/0	XXX▲XX+/0	31
32	1		FWT	006		+	FWT TO DATE	F7		NEW FWT TO DATE	F7		34	XXX▲XX+	XXXX▲XX+	XXXX▲XX+	XXXX▲XX+	32
33	1		SPACES	F3		T				CLEAR FWT	006		34	---+		------------+	------------+	33
34	1		GROSS PAY	007		−	FWT	006		GROSS − FWT	F2		35	XXX▲XX+	XXX▲XX+	XXX▲XX+	XXX▲XX+	34
35	1		GROSS − FWT	F2		−	FICA	008		GROSS − TAXES	F2		36	XXX▲XX+	X▲XX+	XXX▲XX+	XXX▲XX+	35
36	1		GROSS − TAXES	F2		−	OTHER DEDUCTIONS	002		NET PAY	002		BR11	XXX▲XX+	XXX▲XX+	XXX▲XX+	XXX▲XX+	36
37	1		SPACES	F3		−	GROSS − TAXES	F2		NET PAY	002		38	---+	XXX▲XX+	XXX▲XX+	XXX▲XX+	37
38	1		SPACES	F3		T				CLEAR REG. PAY	F2		39	---+		------------+	-----+	38
39	1		SPACES	F3		T				CLEAR ELAP. HRS.	F8		W1	---+		------------+	-----+	39
40																		40
41																		41
42																		42
43																		43
44																		44
45																		45
46																		46
47																		47
48																		48

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM

FUNCTION CONTROL CHART

APPLICATION: WEEKLY PAYROLL

PROGRAM NO.: V

DEMAND UNITS

NO.	TYPE OF UNIT	TEST IN FROM	TEST OUT TO		DEMAND IN FROM	DEMAND OUT TO	
			NOT READY	READY		INPUT	OUTPUT
1	150 CPM				START, OE2	IN STEP 1	
2							
3							
4							
5							
6							
7							
8							

SCAN UNITS

NO.	TYPE OF UNIT	TO SCAN OUT	NO.	TYPE OF UNIT	TO SCAN OUT	NO.	TYPE OF UNIT	TO SCAN OUT
1			9			17		
2			10			18		
3			11			19		
4			12			20		
5			13			21		
6			14			22		
7			15			23		
8			16			24		

CODE DISTRIBUTOR

FROM V1 OF STEPS	
FROM V2 OF STEPS	
FROM R OF STEPS	

NO.	TO	NO.	TO
00		26	
01		27	
02		28	
03		29	
04		30	
05		31	
06		32	
07		33	
08		34	
09		35	
10		36	
11		37	
12		38	
13		39	
14		40	
15		41	
16		42	
17		43	
18		44	
19		45	
20		46	
21		47	
22		48	
23		49	
24		>49	
25			

FROM OUT OF FUNCTION:

NO.	TO	NO.	TO
0		5	
1		6	
2		7	
3		8	
4		9	

INPUT CONTROL LINES

NO.	TO
a	PU T1
b	PU T2
c	PU T7
d	PU T8
e	PU T13
f	PU T6 → 14
g	
h	
i	
j	
k	
l	

OUTPUT CONTROL LINES

NO.	FROM
A	OE 2
B	OE 1
C	
D	
E	
F	
G	
H	
I	
J	

START TO	DI #1

SCAN OUTS

NO.	TO

ERROR CHECKS

FROM	TO
ARITH.	STEP REPEAT
±O'FLOW	STOP
÷O'FLOW	STOP
PARITY	STOP
ADDRESS	STOP

INDICATORS

NO.	FROM	
1		
2		
3		
4		
IND. SWITCH	FROM	TO
NO CHECK		

READ UNIT RECORD

NO.	IN FROM	OUT TO
1		
2		
3		
4		
5		
6		
7		
8		

WRITE UNIT RECORD

NO.	IN FROM	OUT TO
1	FROM STEP 39	OE 2
2		
3		
4		
5		
6		
7		
8		

STEP CLEAR

IN	OUT
CLEAR PROG. SEL.	

STEP REPEAT

IN FROM	ARITH. ERROR

CHANNEL SEARCH EQUAL

NO.	FROM OUT OF STEP NO.	TO IF = V1	TO IF = V2	TO IF ≠
1	STEP 2	STEP 5	—	STEP 3
2				
3				
4				

BRANCHING

NO.	IN FROM STEP	TO IF +	TO IF −	TO IF O
1	3	OE 1	—	STEP 4
2	4	STEP 2	—	OE1
3	5	STEP 6	—	OE1
4	6	STEP 7	—	STEP 11
5	9	STEP 10	—	STEP 15
6	11	STEP 12	—	STEP 13
7	15	STEP 16	STEP 21	STEP 21
8	23	STEP 24	—	STEP 30
9	25	STEP 26	STEP 28	STEP 28
10	31	STEP 32	STEP 33	STEP 34
11	36	STEP 38	STEP 37	STEP 37
12				
13				
14				
15				
16				
17				
18				

CHANNEL SEARCH UNEQUAL

NO.	FROM OUT OF STEP NO.	TO IF =	TO IF ≠
1			
2			
3			
4			

OUT EXPANDERS

NO.	IN	OUT	OUT
1	+BR1 0BR2 + 3	OPC B	OE 2
2	OE1, W1	OPC A	DI #1
3			
4			
5			
6			
7			
8			

FUNCTION DELAYS

NO.	IN 1	IN 2	OUT
1			
2			
3			

ALTERNATE SWITCHES

NO.	SELECT	COMMON	NON-SELECT
1			
2			
3			
4			
5			
6			

FUNCTION SEQUENCE

NO.	SET	PROBE	OUT
1			
2			

SYMBOLS FOR PROGRAMMING:

Processes: ....PR
Transfer........T
Addition........+
Subtraction ....−
Multiplication X
Division........÷
Left Zero Elim........LZE
Compare C
Channel Search = ...ECS
Channel Search ≠........UCS

SHIFTS:

Shift 2 Left........2L
Shift 2 Right........2R
Expand Prod. Shift........EXS

SELECTORS:

Computor Selector ....$T_1 \rightarrow T_{48}$
Drum Selector........$A_1 \rightarrow A_8$
Input-Output Selectors........$S_1 \rightarrow S_{10}$
Common ....CT(#) or CA (#)
Select........ST (#) or SA (#)
Non-Select........NST(#) or NSA (#)
Pick-Up........PUT(#) or PUA (#)
Program Selector ....$PS_1 \rightarrow PS_{16}$
Delay of PS ....$DPS_1 \rightarrow DPS_{10}$
Imm. of PS........$IPS_{11} \rightarrow IPS_{16}$
Power of PS ....$PSO_1 \rightarrow PSO_{16}$
Drop Out of PS........$PSDO_1 \rightarrow PSDO_{16}$
Ground of Sel ....GT (#)
Computer Ground ....CG
Demand Ground........$DG_1 \rightarrow DG_8$
Scan Ground ....$SG_1 \rightarrow SG_6$
Selector Hold ....SH

Function Delay........FD
In FD 1 ....$FD1_1 \rightarrow FD1_3$
In FD 2 ....$FD2_1 \rightarrow FD2_3$
Out FD........$FDO_1 \rightarrow FDO_3$

STORAGES:

Input-Output ....000→009
Intermediate ....10→59
Revolver ....$F_0 \rightarrow F_{19}$
Channel Search Fields ....$CSF_0 \rightarrow CSF_{19}$

DEMAND UNITS:

Test In........TST (#)
Test Out Ready ....TSTR (#)
Test Out Not Ready........TSTN (#)
Demand In ....DI (#)
Demand Out Input........DOI (#)
Demand Out Output ....DOO (#)

Scan Out ....$SO_1 \rightarrow SO_6$
Program Step In ....In ST (#)
Program Step Out........ From ST (#)
Out Expanders ....$OE_1 \rightarrow OE_8$
In Branching........$BR_1 \rightarrow BR_{18}$
+ Branch Out ....$+BR_1 \rightarrow BR_{18}$
− Branch Out ....$-BR_1 \rightarrow BR_{18}$
o Branch Out ....$oBR_1 \rightarrow BR_{18}$
Read Unit Record........$R_1 \rightarrow R_8$
Write Unit Record........$W_1 \rightarrow W_8$
Equal Channel Search In........ECSI (#)
Equal Channel Search Out=V_1 ..$ECSV_1$ (#)
Equal Channel Search Out=V_2 ..$ECSV_2$ (#)
Equal Channel Search Out ≠ ....ECS ≠ (#)
Unequal Channel Search In........UCSI (#)
Unequal Channel Search Out =..UCS = (#)
Unequal Channel Search Out ≠..UCS ≠ (#)
Program Address Counter ....PAK
Address Register ....ADR

Code Distributor........CD
In CD V_1 ....CDV_1
In CD V_2 ....CDV_2
In CD R........CDR
Out CD 00 to > 49........$CD_{00} \rightarrow CD > 49$
In CD Pulse In........CDP
Out CD Pulse Out ....CDP_0--CDP_9

Function Sequence Set ....$FSS_1 \rightarrow FSS_2$
Function Sequence Probe ....$FSP_1 \rightarrow FSP_2$
Function Sequence Out........$FSO_1 \rightarrow FSO_2$

ALTERNATE SWITCHES:

Common ....$SW_1C \rightarrow SW_6C$
Select........$SW_1S \rightarrow SW_6S$
Non-Select........$SW_1NS \rightarrow SW_6N_6$

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

UNIVAC FILE-COMPUTER SYSTEM

COMPUTER SELECTOR CONTROL CHART

APPLICATION: WEEKLY PAYROLL PROGRAM NO. V

PICK-UP FROM	GROUND TO	T#	SELECT	COMMON	NON-SELECT
1PCa 0/28	CG	1	19	V_2 STEP 31	CT2
1PCb 1/28	CG	2	CT 6a	NS T1	CT7
		3a			
		3b			
		4a			
		4b			
		5a			
		5b			
1PCf 9/28	CG	6a	11	ST2	10
		6b	13	ST7	12
		6c	15	ST 8	14
		6d	17	ST13	16
1PCc 3/28	CG	7	CT6b	NST2	CT8
1PCd 5/28	CG	8	CT6c	NST7	CT13
		9a			
		9b			
		10a			
		10b			
		11a			
		11b			
		12a			
		12b			
		12c			
		12d			
1PCe 7/28	CG	13	CT6d	NST8	CT14
1PCf 9/28	CG	14	18	NST13	20
		15a			
		15b			
		16a			
		16b			
		17a			
		17b			
		18a			
		18b			
		18c			
		18d			
		19			
		20			
		21a			
		21b			
		22a			
		22b			
		23a			
		23b			
		24a			
		24b			
		24c			
		24d			

PICK-UP FROM	GROUND TO	T#	SELECT	COMMON	NON-SELECT
		25			
		26			
		27a			
		27b			
		28a			
		28b			
		29a			
		29b			
		30a			
		30b			
		30c			
		30d			
		31			
		32			
		33a			
		33b			
		34a			
		34b			
		35a			
		35b			
		36a			
		36b			
		36c			
		36d			
		37			
		38			
		39a			
		39b			
		40a			
		40b			
		41a			
		41b			
		42a			
		42b			
		42c			
		42d			
		43			
		44			
		45a			
		45b			
		46a			
		46b			
		47a			
		47b			
		48a			
		48b			
		48c			
		48d			

PROGRAM SELECTS

PS#	IN FROM	DELAY OUT TO	D. O. FROM	POWER TO
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
PS#	IN FROM	IMMEDIATE OUT TO	D. O. FROM	POWER TO
11				
12				
13				
14				
15				
16				

REMARKS:

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

315 FOURTH AVENUE, NEW YORK 10, N. Y.

SPTM 4287 REV. 2

PRINTED IN U.S.A.