IBM

General Information Manual

IBM 1401 Data Processing System

From Control Panel to Stored Program

Preface

This manual is introductory in nature and is a foundation for understanding stored programming.

Its purposes are threefold:

1. To acquaint the reader with stored-program machines and concepts.
2. To aid one in preparing for the 1401 course.
3. To serve as a reference.

The manual is divided into three sections. The first introduces the concepts and terms associated with the 1401 Data Processing System. The second describes the 1401 insofar as it must be understood for a reading of the third section. The third section illustrates functions of wired control panel machines and shows how the same functions are accomplished on the 1401 Data Processing System.

CONTENTS

Data Processing Machines

Data processing machines are not new. Historical milestones are dated as early as the 1600s. Pascal became a pioneer in the field with the first mechanical computer in 1641; others, such as Leibnitz in 1673 and Charles Babbage in the 1800s, contributed additional concepts and ideas with their machines. Little followed Babbage's machines until 1944 when Harvard University and IBM completed the Mark I, the first modern machine to employ Babbage's principles. Since 1945, rapid and important strides have been made in the field of data handling and processing.

Quite naturally, data processing machines were first used by mathematicians and scientists for processing and handling scientific information. But with an understanding of terms and concepts associated with the machines, accountants realize that they do an excellent job of processing accounting transactions as well. In present accounting applications, stored-program machines have proved very effective.

Data processing machines fall generally into one of two categories: those which are primarily controlled by stored instructions (stored-program machines) and those which are controlled by wired instructions (wired or mechanical control panel machines). Examples of stored-program machines are the IBM 1401 Data Processing System, the IBM RAMAC 305 and the IBM 650 Data Processing System. The 602 Calculating Punch, the 604 Electronic Calculating Punch, the 402 and 403 Alphabetical Accounting Machines and the 407 Accounting Machine are wired control panel machines.

Principles of Data Processing Machines

Data processing machines have associated with them five functions:

Input
Storage
Control
Arithmetic
Output

Input implies the taking in of information. Input can be a card, a magnetic tape or a paper tape. Some machines accept only certain types of input. For example, the 407 and 403 accept only punched cards.

Storage is the medium whereby data, once it is read by the machine, is held until it is used. The collator has storage – comparing magnets. Accounting machines use counters and storage units to retain information.

Control allows the machine to be set up to process different applications, thus utilizing certain features for some applications and other features for other applications. In accounting machines, reproducers, collators, calculators, etc., this is done with the wired control panel. In stored-program machines, control is accomplished with the stored instructions.

Arithmetic includes the ability to add, subtract, multiply and divide. Some machines use all arithmetic functions; others use only certain ones. Accounting machines use addition and subtraction. Calculators such as the 604 and 602 make use of all four functions.

Output consists of anything turned out by the machine as a result of processing the input. It includes punched cards, printed reports, paper tapes, magnetic tapes, and combinations of these.

Wired control panel machines and stored-program machines are alike in possessing these functions. Differences arise primarily in how the functions are accomplished. For example, control is accomplished on wired control panel machines with the control panel; on stored-program machines it is accomplished with coded instructions stored in the machine.

Another difference arises in reading the input. With wired control panel machines, (1) only the contents of card columns which are wired are usable, (2) of these columns which are wired, only the information going to a storage unit (if there is one on the machine) retains its original identity, and (3) all card information not in storage is lost after the card leaves the reading station. On the other hand, with stored-program machines such as the 1401, (1) when a card is read, all 80 columns of card data automatically enter the machine, with no control panel wiring necessary, (2) the 80 columns of data enter storage, so that original identity is retained, and (3) use of the card data is not limited to the time during which the card is passing the reading station, but can be made at a later time, since it is retained in storage.

The Storage Unit

Just as a counter is composed of smaller units known as counter wheels, a storage unit is composed of smaller units termed storage positions. Stored-program machines use the storage unit to retain (1) data to be processed and (2) instructions for processing the data.

Storage positions on the 1401 have several distinguishing characteristics:

1. Each position is capable of holding a single character which can be either an alphabetic, numerical or special character. Therefore, the amount of information which can be stored within a machine is determined directly by the number of storage positions available in the machine.
2. A stored character can be "read out" as often as desired without destroying it. To "read out" is to make available for processing purposes the characters which are in selected storage positions.
3. The content of a storage position is destroyed only by "reading in" another character. To "read in" is to store data in a storage position or positions. "Reading in" destroys the content of storage positions and replaces it with new data.
4. A character in storage retains its original identity until it is replaced by another.
5. Each storage position has an address. An address for each position permits the machine to store a character in a specific location, then call for and use that character by referring to the address of its storage location.

Generally there are six types of storage: magnetic tape, magnetic disk, magnetic drum, core, mechanical and vacuum tube. Although different from each other in physical make-up, they are alike in their function, that is, to store information. These various types of storage exist because of different needs created by the many applications processed on machines.

Core storage is one of the more recently developed storage devices. For machines utilizing it, operations requiring the reading of information from and writing of information into storage are extremely fast — so fast that the time required to read or write a character is expressed in terms of millionths of a second. (A millionth of a second is commonly known as a "microsecond." A thousandth of a second is a "millisecond.")

Each 1401 core storage position is composed of eight tiny, vertically aligned cores, similar in shape to a doughnut. If a storage position were viewed in the machine, its cores would be arranged vertically, as seen in Figure 1.

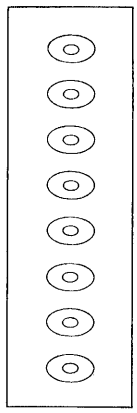


Figure 1. A Storage Position

The core is composed of a ferromagnetic material — "ferro" indicating the presence of iron and "magnetic" implying the ability to be magnetized.

Passing through each core are four wires (see Figure 2). These wires are used in storing information and then retrieving it.

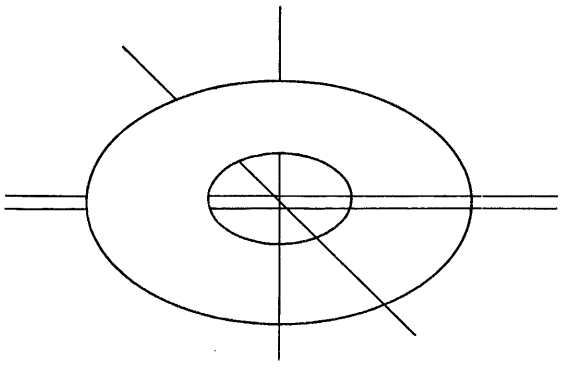


Figure 2.

If electric current is passed simultaneously through two of the wires, in the directions shown in Figure 3, a magnetic field clockwise in nature is established in the core.

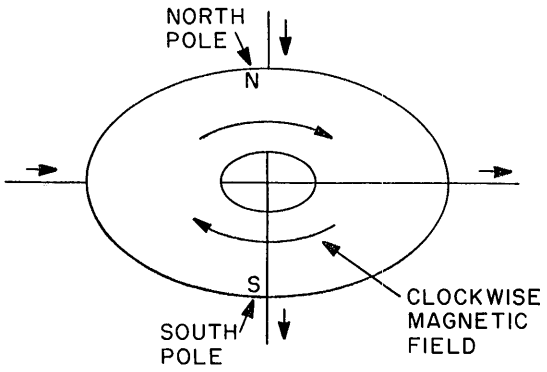


Figure 3.

If the current is reversed on the same two wires, the core's magnetic field becomes counterclockwise in nature (see Figure 4).

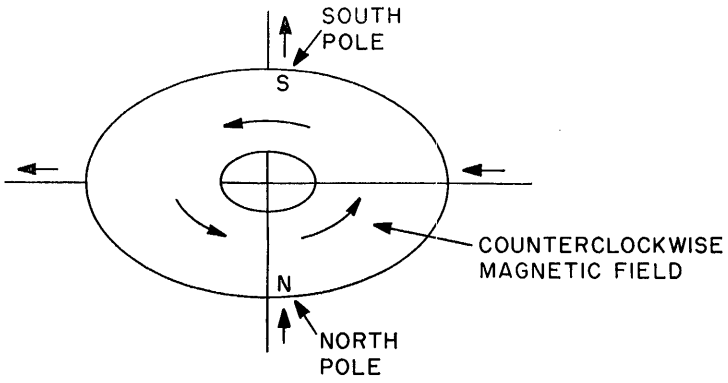


Figure 4.

Once the core's magnetic field is set up, the core remains magnetized in that direction until current is passed along the wires in the opposite direction, to change the direction of magnetism.

When the core is magnetized in a clockwise direction, it is said to be "on" (see Figure 3). When magnetized in a counterclockwise direction, it is "off" (see Figure 4). The "on" or "off" condition for each core is determined by the direction, clockwise or counterclockwise, of its magnetic field. Note that "on" and "off" are terms chosen arbitrarily; they do not indicate presence and absence of current.

In a particular storage position, some of the cores may be off while others are on (see Figure 5).

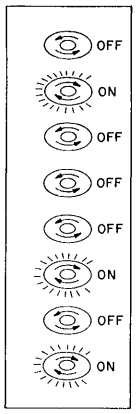


Figure 5. Storage Position

In Figure 5, the first, third and seventh cores, from the bottom up, are on. All other cores are off.

In order to store characters (numerical, alphabetic and special) each of the first six cores of any storage position has a value assigned to it. Moving from bottom to top, the first core has a value of 1 assigned to it; the next a value of 2, the next a value of 4, the next a value of 8, the next a value of A, and the next a value of B. Use of the two cores at the top of each storage position will be explained later.

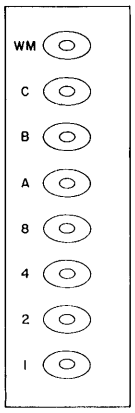


Figure 6. Storage Position

If a core is on, it represents the value assigned to it. If a core is off, it has no value. In the storage position illustrated in Figure 7 only the lowest core is on. This combination of on-off conditions is recognized by the 1401 as the digit 1.

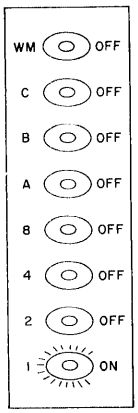


Figure 7. Storage Position

Another method of illustrating the same thing is shown in Figure 8. Circles represent the cores. If a core is on, the circle is shaded; all other cores are unshaded and therefore are off.

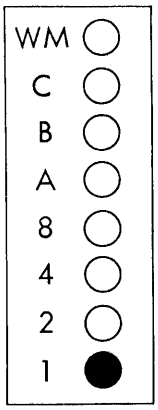


Figure 8.

Upon being read from the card into the 1401, every character (alphabetic, numerical or special) is converted from the IBM card code to the 1401's *Binary Coded Decimal* and stored. With the binary coded decimal, the four lower cores, for values 1, 2, 4 or 8, are used singly or in combinations to represent the digits 0-9.

The on or off condition of a single core is referred to as a "bit" of information. Figure 9 illustrates the bit combinations used to represent each of the digits 0-9.

When reading the binary coded decimal, it is said that:

For the digit 1 the 1 bit is on.
For the digit 2 the 2 bit is on.
For the digit 3 both the 1 and 2 bits are on, etc.
For the digit 0 the 8 and 2 bits are on.

To see the codes as they appear in the 1401 requires an additional step. Note that in Figure 9 the A and B bits are off for all digits. Furthermore, the digits 1, 2, 4, 7 and 8 contain an odd number of on bits in the 1, 2, 4 and 8 positions. Each of the digits 3, 5, 6, 9 and 0 contains an even number of on bits in those positions.

Built into the machine is an automatic checking feature called the *parity check.* A character stored in the machine is checked each time it is moved from one location to another to make certain that it has an odd number of bits. In order to meet this test, the seventh core, labeled the C bit, or the check bit (see Figure 10), is automatically set on or off for each character being converted from the IBM card code to the binary coded decimal. As this conversion is made, a device in the 1401 counts the number of on bits in positions 1, 2, 4, 8, A and B. If the character consists of an even number of bits the C bit is set on; if the character consists of an odd number of bits the C bit is set off.

Thereafter, every time a character is moved anywhere in the machine the parity check is proved. When any character fails to meet the odd bit count requirement, the machine stops processing immediately. This is insurance against having characters changed after they enter the machine.

The digits 0-9 with the check bit added to the appropriate characters for the odd bit count are illustrated in Figure 11.

Digits 3, 5, 6, 9 and 0 have a check bit in their respective storage positions. Without it, each would fail to pass the parity check. In addition, alphabetic *and* special characters must also meet the parity check.

With the binary coded decimal, alphabetic and special characters use the 1, 2, 4 and 8 bits for the numerical portion of the IBM card code, and A and B bits for the zone portion of the IBM card code.

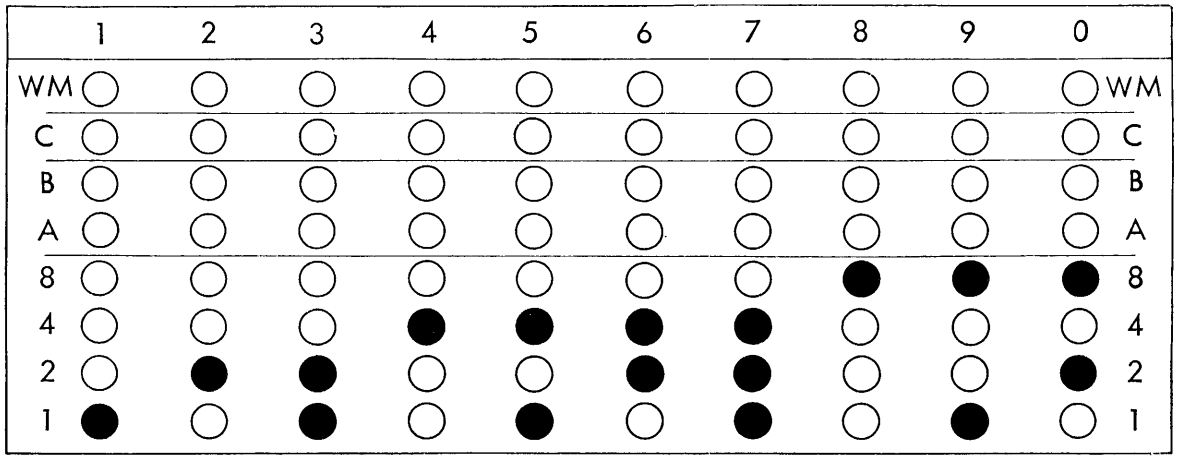


Figure 9.

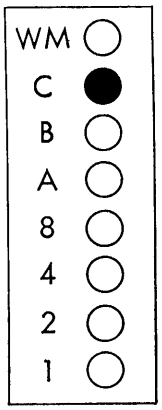


Figure 10. Storage Position

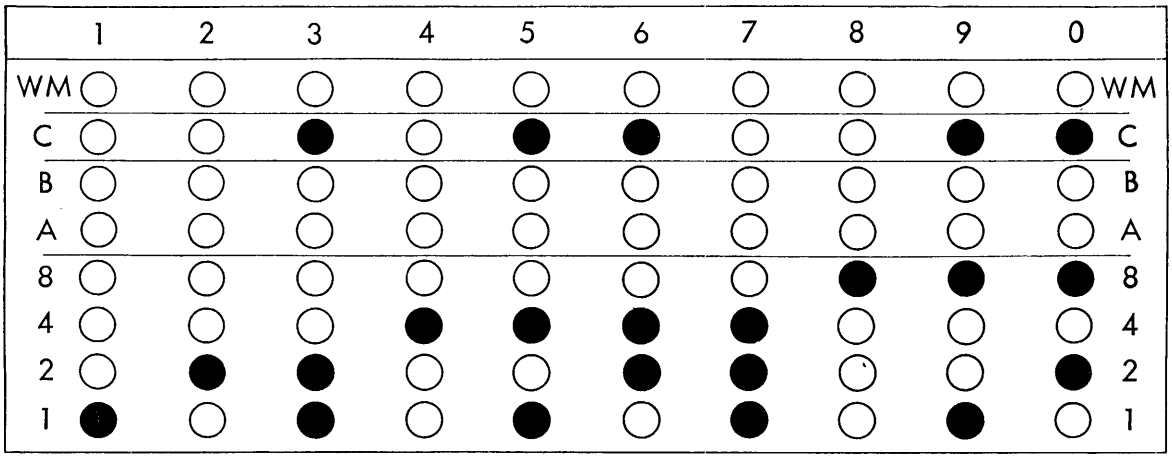


Figure 11.

The zero zone is represented by the A bit on.

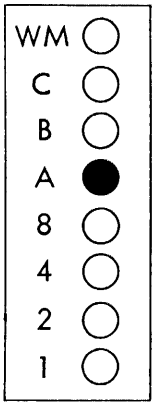


Figure 12. Zero Zone

The 11 zone is represented by the B bit on.

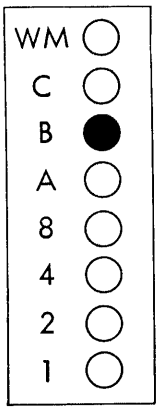


Figure 13. 11 Zone

The 12 zone is represented by both the A and B bits on.

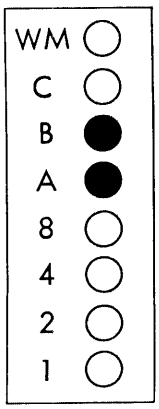


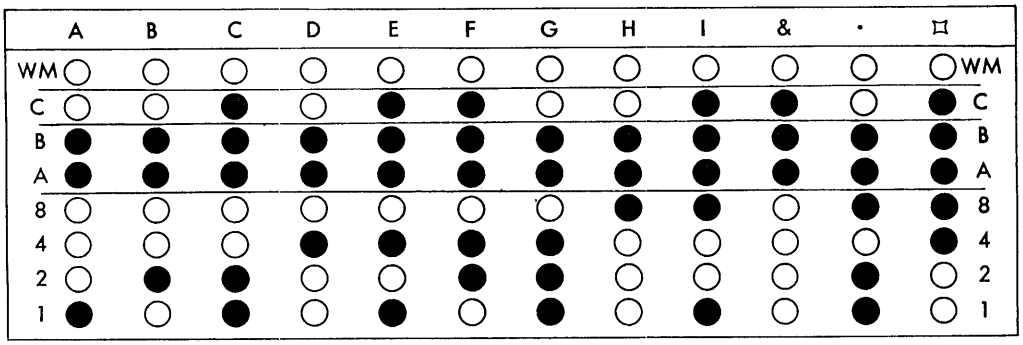
Figure 14. 12 Zone

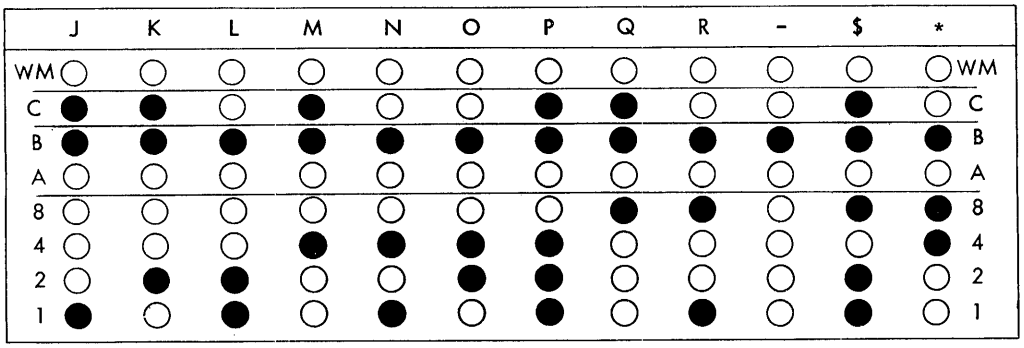
Combining the bits representing zones, the bits representing digits, and any necessary check bit, alphabetic and special characters appear as illustrated on the opposite page in Figure 15.

Core Storage Addressing in the 1401

The number of characters required to address each location varies with different types of machines. Some machines use a five-character address, others a four-character address, and still others a three-character address. The number of storage positions in the machine affects, but does not necessarily determine, the number of digits required for addressing. The 1401 Card System uses a three-character scheme; therefore, each position of core storage has a three-character address assigned to it. The purpose of assigning an address to each position is the same as that of numbering houses on a street — namely, to enable reference to a specific position. Storage positions are numbered consecutively. The address of the first position is 000; the address of the second, 001; the address of the third, 002; and so forth.

The first one thousand positions are numbered 000-999. Thereafter, the first character of the address changes and becomes either a special or alphabetic character. For example, positions 1000-1099 are addressed as +00-+99; positions 1100-1199 as /00-/99; positions 1200-1299 as S00-S99, etc. Alphabetic and special characters are used until each of the possible four thousand positions has an address.


A B C D E F G H I & · ⌑
WM C B A 8 4 2 1


J K L M N O P Q R - $ *
WM C B A 8 4 2 1

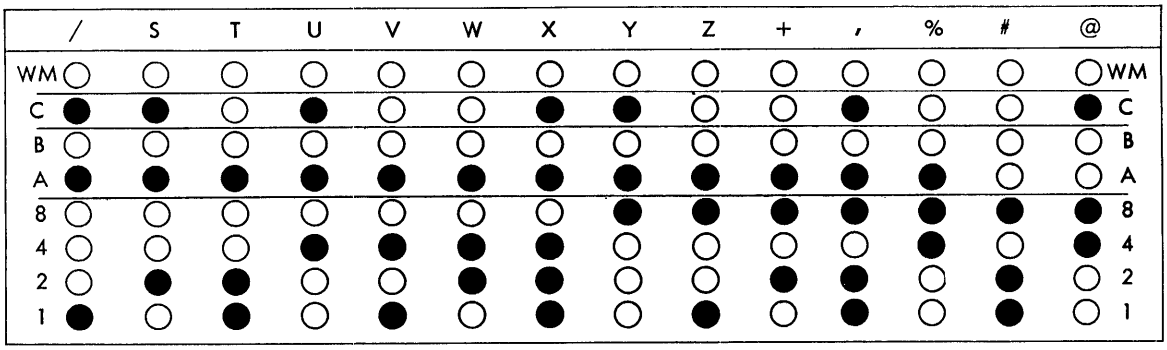

/ S T U V W X Y Z + , % # @
WM C B A 8 4 2 1

Figure 15.

Addressing

LOCATION	ADDRESS	
000-999	000-999	First character has no zone over digit
1000-1099	+00-+99	First character uses zero zone over digit
1100-1199	/00-/99	
1200-1299	S00-S99	
1300-1399	T00-T99	
1400-1499	U00-U99	
1500-1599	V00-V99	
1600-1699	W00-W99	
1700-1799	X00-X99	
1800-1899	Y00-Y99	
1900-1999	Z00-Z99	
2000-2099	J00-J99	First character uses eleven zone over digit
2100-2199	K00-K99	
2200-2299	L00-L99	
2300-2399	M00-M99	
2400-2499	N00-N99	
2500-2599	O00-O99	
2600-2699	P00-P99	
2700-2799	Q00-Q99	
2800-2899	R00-R99	
2900-2999		
3000-3099	A00-A99	First character uses twelve zone over digit
3100-3199	B00-B99	
3200-3299	C00-C99	
3300-3399	D00-D99	
3400-3499	E00-E99	
3500-3599	F00-F99	
3600-3699	G00-G99	
3700-3799	H00-H99	
3800-3899	I00-I99	
3900-3999		

Figure 16.

Since whole fields of information are stored more often than single characters, adjacent core storage positions are thought of as "grouped" to form storage fields, in much the same way that card columns are grouped into card fields. Grouping of core positions changes from one application to another. A numerical field occupying storage positions 952-956, and containing the value 17562, is illustrated in Figure 17. Position 952 is the high-order position and 956 is the low-order position.

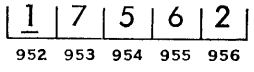


Figure 17.

Storage fields are defined to the machine with the aid of *word marks*. A word mark is recorded in the high-order storage position by using the topmost core in that position. (See Figure 18.)

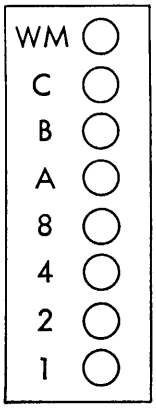


Figure 18.

If the topmost core is on, a word mark is said to be set in that storage position. (See Figure 19.)

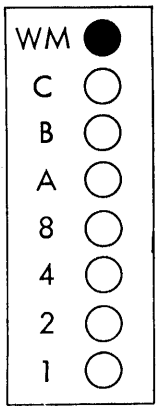


Figure 19.

If the word mark core is off, there is no word mark in that position. Machine-coded instructions set and erase word marks; and can do so either before or during the processing of the application.

A word mark is *not* a character; it is a single bit.

For the machine to properly identify the limits of a field, a word mark should appear in the high-order – and only in the high-order – position of each field. A numerical field occupying storage positions 952-956, containing the value 17562, and with a word mark over position 952, would look as follows:

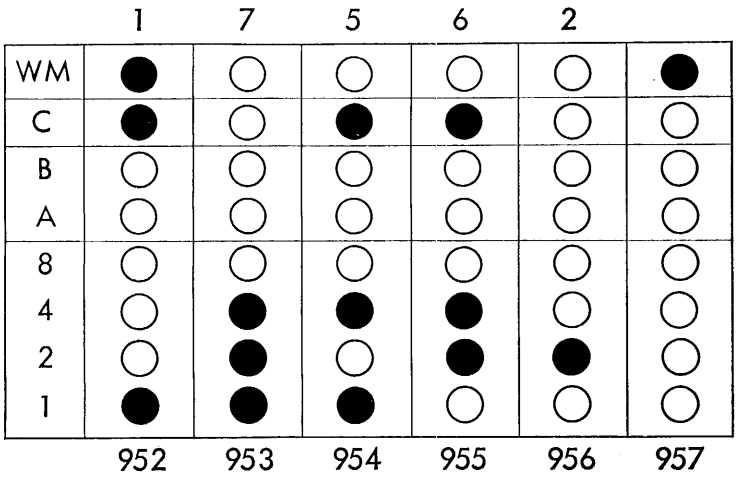


Figure 20.

A storage field begins with the position containing the word mark and includes it and all other positions to the right, up to, but not including, the next position containing a word mark. In Figure 20, the amount field is composed of positions 952-956. The next field beyond the amount field begins in position 957.

From this point on, the presence of a word mark in a position of storage will be signaled by a line drawn under the character in that position. The previous example now appears as:

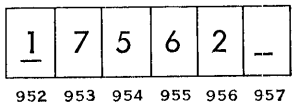


Figure 21.

Positions 952 and 957 contain the word marks.

A storage field can be either of two types — a data field or an instruction field. A data field is one which contains information to be processed (such as sales amount), information to aid in processing (such as a control code), or information which is already processed (such as an accumulated amount). Instruction fields contain coded instructions which the machine must execute in processing the data. The size of the data field is determined by the greatest number of characters which can be in the data amount; the size of an instruction field is determined in the same manner.

Variable Word Length

Certain stored-program machines, of which the 1401 is an example, have a desirable characteristic called "variable word length." The ability to have grouped together any number of storage positions to accommodate fields of any size, gives rise to this term. The fact that a machine has variable word length allows efficient use of the storage area.

It is recalled that on both the 604 and 602 each storage unit contains a specific number of positions; each is fixed in length. For this reason, high-order positions are frequently wasted because, in size, the field is smaller than the unit.

In a variable word length machine this does not happen; there are no fixed groupings of storage positions. Instead, the size of the grouping varies with the length of the data or instruction field to be accommodated. The word mark makes this possible; it can be set to make any storage position the high-order position. It can also be erased from storage positions when field limits change.

Other stored-program machines have "fixed word length." This means that storage positions are grouped and used in multiples of a constant number, generally multiples of ten positions. Each group of ten positions, and the data in those positions, is known as a "word."

Addressing Fields

To use the data in a field or to interpret and execute an instruction, the machine must be able to determine the limits of the field — the low-order position and the

high-order position. The word mark in storage, together with an address found in the instruction, enables the machine to determine field limits.

Word marks are always set in the high-order position regardless of type field. A data field is addressed in its low-order position. The machine then treats that position, and all successive positions to the left, up to and including the one containing the next word mark, as the data field.

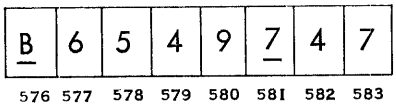


Figure 22.

In Figure 22, if the digits in positions 576-580 represent data, the field is addressed in its low-order position, 580. The machine uses the character stored in position 580 and all characters in successive locations to the left through position 576.

Instruction fields are addressed in their high-order position (the position containing the word mark). The machine treats that position and all positions to the right, *up to,* but not including, the next containing a word mark, as being part of the instruction field.

If the digits in positions 576-580 of Figure 22 represent an instruction, the field is addressed in position 576. Characters in all successive positions to the right, through position 580, are included as part of the instruction.

Input-Output Areas

Every stored-program machine must have storage areas specified to the machine for the purposes of (1) reading in, (2) punching and (3) printing, if the machine performs all these functions. Otherwise, the machine, when instructed to read in, has no way of knowing where in the storage unit to put the card data; or, when instructed to punch, where in storage to get the data it punches into the card; or, when instructed to print, where in storage to get the data for printing on the report.

Methods of specifying these areas are (1) wired circuitry in the machine or (2) stored instructions in the machine. Each stored-program machine uses one of these types of input-output area specification.

Examples of machines which have input-output storage areas specified by internal circuitry are the 305, the 650 and the 1401. For example, on the 1401 during reading, the 80 columns of card data automatically go into storage positions 001-080 (see Figure 23). The content of card column 1 goes into storage position 001, etc. When the 1401 is instructed to punch, it automatically punches into the card columns the contents of storage positions 101-180. If instructed to print, the 1401 automatically prints in the print positions the contents of positions 201-300 (or 201-332, depending upon the machine model).

The fact that a storage area is used at one time for one of the input-output functions does not prevent its use for ordinary storage purposes at another time when that function is not being performed.

The Instruction

Instructions are the means of controlling a machine. All machines, wired control panel as well as stored-program, must be controlled to perform functions in the proper sequence, at the proper time, and with the proper data in order to process an application successfully. In stored-program machines this is accomplished with the stored *instruction.* In wired control panel machines it is accomplished with the control panel.

In the 1401, an instruction has up to four parts:

1. The operation code (commonly referred to as the "op code")
2. The A/I address
3. The B address
4. The digit modifier

Each stored-program machine has a set of functions which it can perform, and for every function there is a corresponding op code. Each machine has its own set of operation codes.

The 1401 has single-character op codes. The codes are letters, numbers or special characters. For example:

A causes addition.
1 reads a card.
C compares two fields for control purposes.
S causes subtraction.
4 causes punching.
2 causes printing.

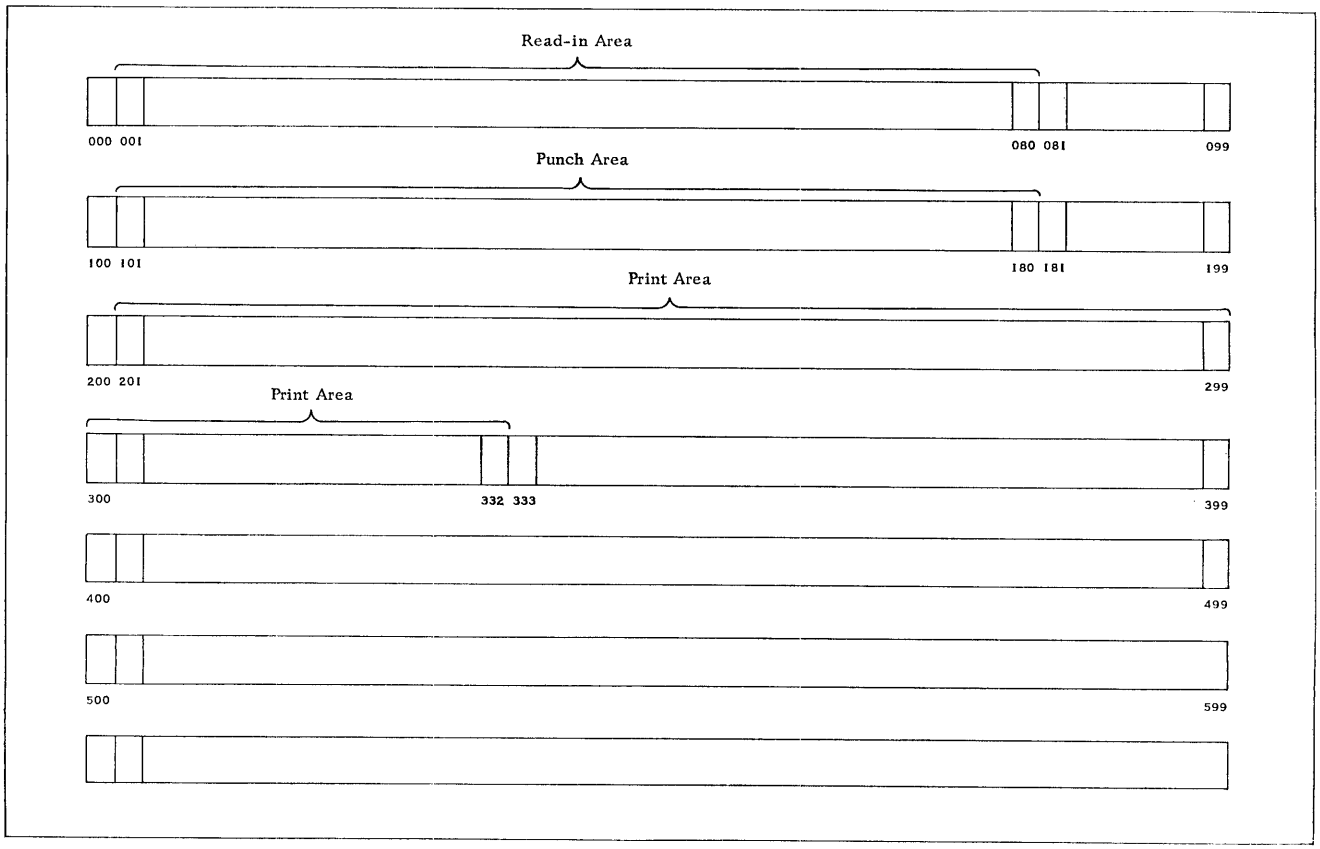


Figure 23. 1401 Storage — Input-Output Areas

The A/I address locates a field which is to be used or affected. It is generally the field from which data comes. The address always consists of three characters —the addressing scheme for the 1401.

The B address also locates a field to be affected or used when the machine performs the function specified by the op code. This is generally the field to which data goes. This address, also, contains three characters.

The digit modifier, used only with certain op codes, always consists of a single character. It adds power and flexibility to the op code.

An instruction containing an op code, an A address and a B address is:

OP Code	A Address			B Address		
A	0	8	0	9	9	9

This is an "add" operation. The op code, A, instructs the machine to add. The A address locates one of the fields to be used in the addition and the B address locates the other field to be used in the same operation. In this case the A address may represent sales amount coming into the machine from a card, and the B address may represent the accumulated amount of all sales cards up to this one.

After seeing this instruction, it is apparent that every instruction will not use all four components. The number of characters in the instruction is determined by the function being performed.

Examples of different combinations are:

1. Op code, I address, B address, digit modifier
2. Op code, A address, B address
3. Op code, I address, digit modifier
4. Op code, A/I address
5. Op code, digit modifier
6. Op code

Since the instructions can vary in length or in the number of characters contained, they are referred to as *variable length instructions.*

A Comparison of Stored-Program Control and Wired-Panel Control

Figure 24 illustrates a 604 planning chart and Figure 25 a wiring diagram for crossfooting factors A, B, C and D. The control panel wiring instructs the machine as follows:

1. Read card columns 40-43 into general storage unit 4.
2. Read card columns 44-48 into factor storage unit 2.
3. Read card columns 49-52 into factor storage unit 4.
4. Read card columns 53-56 into general storage unit 2.
5. Read factor A out of general storage 4 and add it into the counter on program step 1.
6. Read factors B, C and D out of their respective storage units and add them into the counter on successive program steps.
7. Read out the sum T and reset the counter. T is punched in card columns 57-61.

To properly plan and diagram this problem for the 604 requires, among other things, a knowledge of control panel hubs and the functions controlled by these hubs. To properly plan and diagram the same problem for the 1401 requires a knowledge of operation codes rather than hubs. The following chart illustrates this principle of control by op codes vs. control by control panel hubs. Crossfooting is the example. (Under the column headed "1401 Control," only the op code portion of the instruction is listed. For 604 control refer to the wiring diagram in Figure 25.)

FUNCTION	604 CONTROL	1401 CONTROL
1. Reading factors A, B, C, D.	Accomplished by wires labeled 1.	The op code 1 reads the entire card into storage positions 001-080.
2. Add factor A.	Wires labeled 2.	The op code $\overset{+}{0}$ (plus zero) resets a storage area for accumulation and adds factor A.
3. Add factor B to A.	Wires labeled 6.	The op code A causes addition.
4. Add factor C to (A + B).	Wires labeled 3.	The op code A causes addition.
5. Add factor D to (A + B + C).	Wires labeled 4.	The op code A causes addition.
6. Read out and reset T. Punch T.	Wires labeled 5.	The op code M reads out T to the proper storage punch positions. The op code 4 causes a card to be punched.

Miscellaneous Earnings + Regular Earnings + Extra Shift Earnings + Overtime Earnings = Gross Earnings

APPLICATION _____ PROBLEM A + B + C + D = T

PROG NO READ	OPERATION NOTES / READ	READ UNITS INTO / OUT OF	PROGRAM SUPPRESS	PROG NO READ	FACTOR STORAGE ASSIGNMENT 6-4 8-6 1	B 2	8-4 3	C 4	MULT. QUOT.	COUNTER	GENERAL STORAGE ASSIGNMENT 6-4 8-6 1	D 2	3	A 4	PROG NO READ
						REG. EARN.		EXTRA SHIFT				O/T PREM.		MISC.	
	NOTES:														
1	Transfer A To Counter									RI+				RO	1
2	Transfer B To Counter					RO				RI+					2
3	Transfer C To Counter							RO		RI+					3
4	Transfer D To Counter									RI+		RO			4
20															20
P	Punch									R & R GROSS EARNINGS					P

Figure 24.

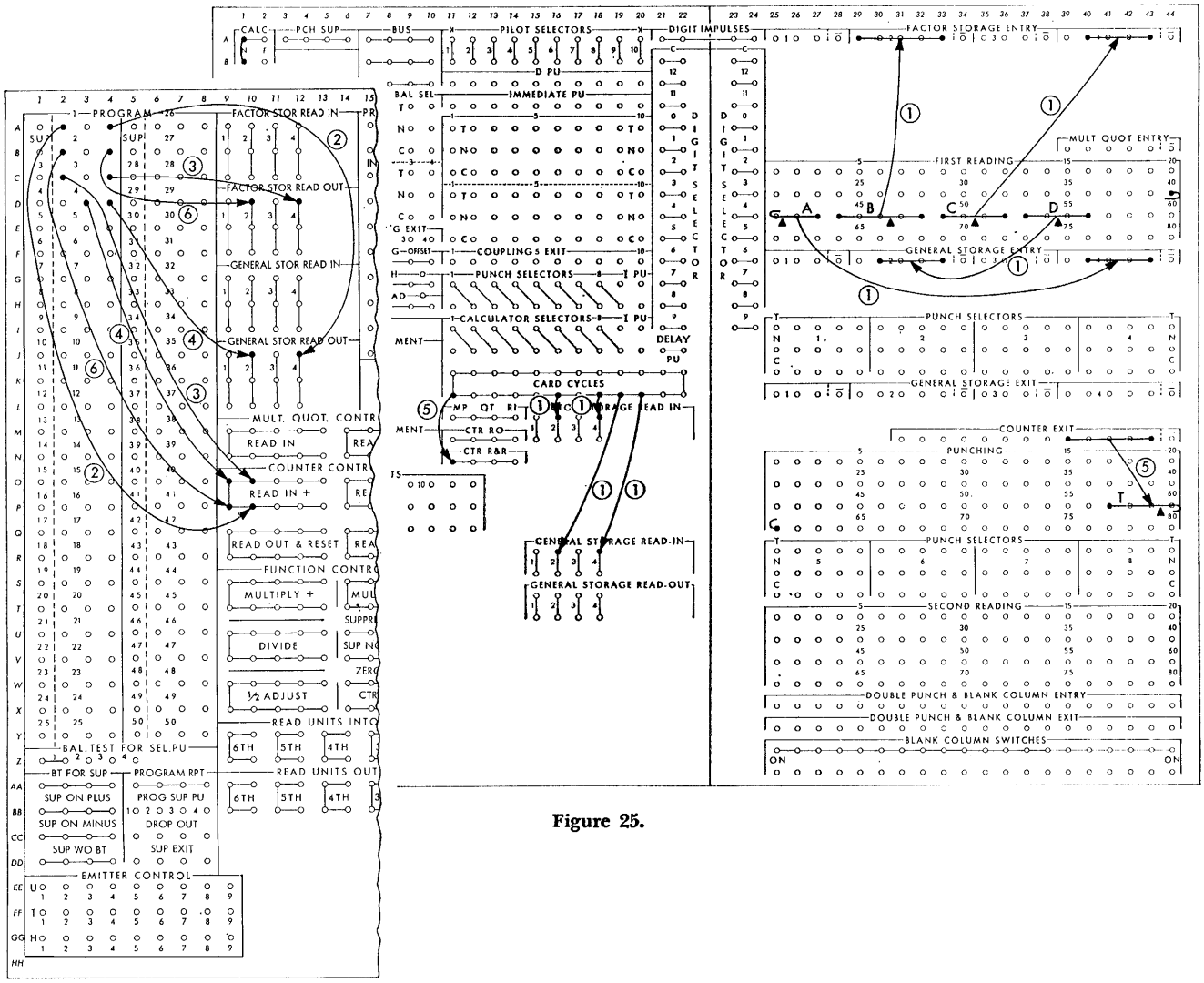


Figure 25.

How Instructions and Data Get into the Machine

After the application to be processed by the 1401 is chosen, it is planned thoroughly. Planning includes (1) analysis of the application, (2) planning and sequencing steps to be used, (3) writing the instructions, (4) designing storage (determining which areas will house data to be processed, data after it is processed, and instructions—keeping in mind the read-in, punch and print areas), and (5) *testing* the application.

In order to process the application, the machine must have stored inside it (1) the instructions which process the card data and (2) the card data to be processed. This requires two machine steps.

First the instructions are stored; this is termed "loading the program." This is equivalent to the insertion of a panel in wired control panel machines. In order to load the program, the instructions must already be in "program load cards" (see Figure 26). A program load card contains (1) instructions to be loaded and (2) load instructions which store in the proper area the instructions to be loaded.

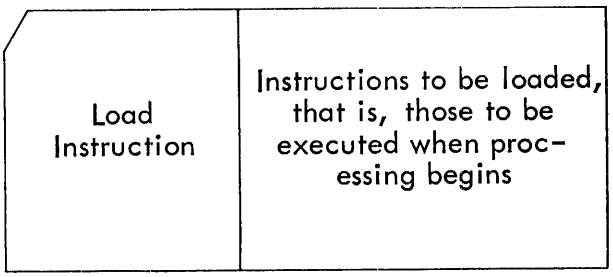


Figure 26. Program Load Card

To load the program, the load cards are placed in the read hopper of the machine and the load button is depressed. After all the load cards are read, the program is loaded. The loaded instructions will be executed each time a data card feeds into the machine. They remain in storage until instructions for a different routine replace them.

Processing of the application begins as soon as (1) the punched data cards are placed in the read hopper, (2) all units are readied, and (3) the start button is depressed.

If storage could be seen once processing is begun (see Figure 27), there would be:

1. The read-in area containing the contents of a data card just coming in to be processed.
2. The punch area where information to be punched is assembled.
3. The print area where information to be printed is assembled.
4. The area necessary to hold data while it is being processed and before it can be put in one of the output areas.
5. The area containing instructions which are being executed in order to process data and place it in the proper output areas.

Only the location of the read-in, punch and print areas is fixed. The location of the other areas can vary.

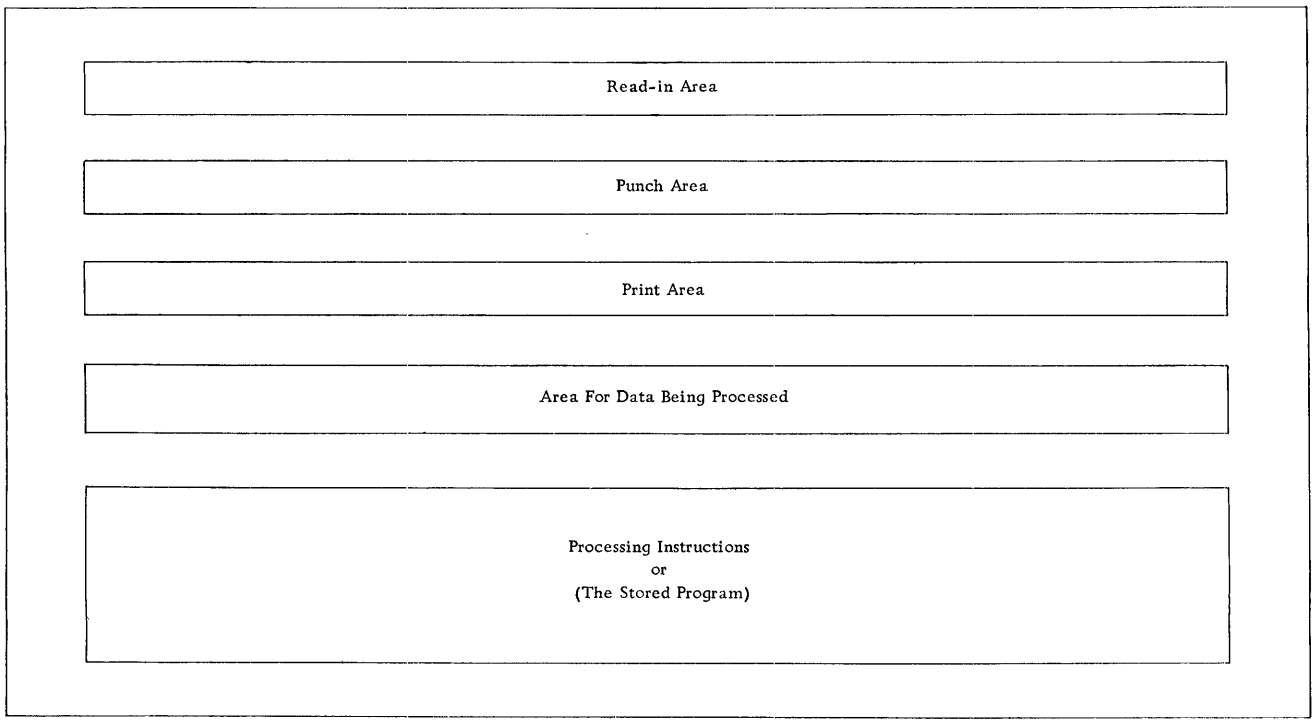


Figure 27. 1401 Storage

Planning is most important in putting an application into operation. The efficiency and output resulting from a well planned application will compensate for the effort and time spent in planning and organizing the application. The ability to plan an application for a particular machine requires an understanding of the concept of the machine, the machine's components, the functions associated with the machine, and the instructions which initiate these functions.

Planning a problem for the 1401 Data Processing System is very similar in principle to planning a problem for the 602 or 604. In either case, the data which is available for input and the data desired as output must be known in order to bridge the gap between the two. The bridge represents machine processing; it consists of machine-coded instructions necessary to supply the missing data and to process all data in order to come up with the desired output.

Block Diagramming

Determining, sequencing and expressing the necessary steps with a standard set of symbols constitutes block diagramming. It is the preliminary work for writing the machine-coded instructions.

Block diagramming is similar to completing the planning of a 602 or 604 problem. The machine must be instructed in each detail of operation. In Figure 24 is the planning chart for a crossfooting problem on the 604.

The planning chart illustrated in Figure 24 becomes block diagramming by substituting the proper symbols. (See reference manual, *Flow Charting and Block Diagramming Techniques,* form C20-8008.) Figure 28 is a block diagram of the same operation for the 1401 Data Processing System.

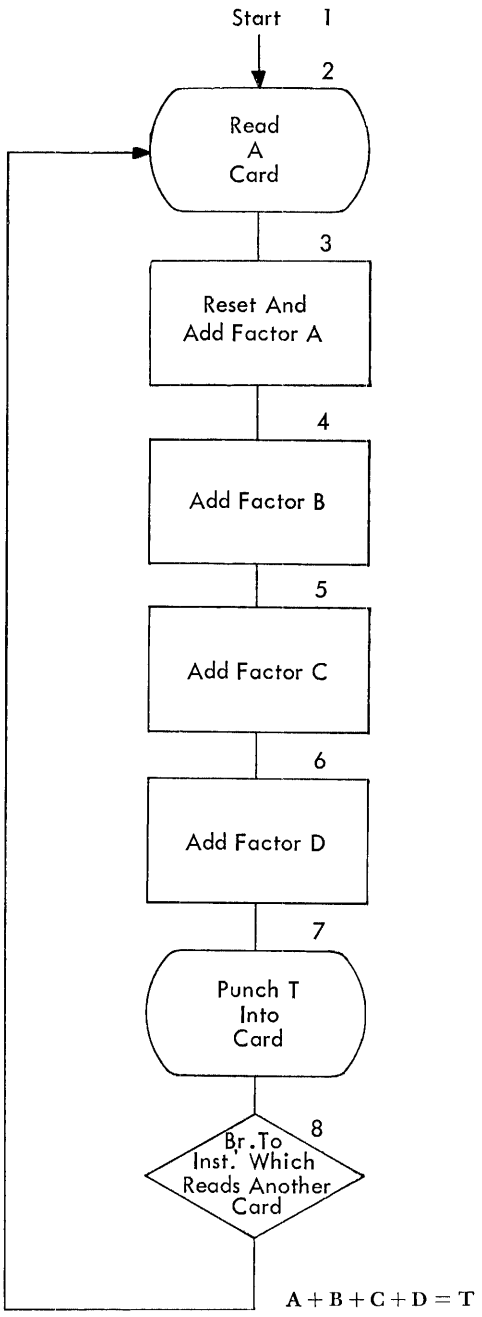


Figure 28. Crossfooting and Punching Results

EXPLANATION OF FIGURE 28:

1. Start. – The card read punch and printer are readied and the start button is depressed. Start applies only for the first card.
2. For the first step a card is read into storage positions 001-080.
3. For the second step certain positions of the storage punch area which are to be used for accumulation are reset and factor A is added into them. On the 1401 it is necessary to reset the accumulate area before it is used; there is no operation code which will reset the area when T is read out.
4. For the third step factor B is added into the same area.
5. For the fourth step factor C is added into the same area.
6. For the fifth step factor D is added into the same area.
7. For the sixth step punching of the result, T, from storage positions 101-180 is initiated.
8. For the seventh step the machine is instructed to go back to the first step, which causes the following card to be read and its processing begun.

Composing machine-coded instructions for each step in the block diagram is called programming. The machine-coded instruction for the first step ("read a card") is 1. Each step thereafter has an instruction which will accomplish the operation indicated. It becomes more obvious now why block diagramming is a preliminary for programming: the necessary steps must be determined before the coded instructions can be written. During block diagramming it should be kept in mind that there must be an instruction for everything the machine is to do:

1. For causing each card to be read.
2. For moving data from the storage read-in area, and to and from the process area.
3. For performing arithmetic functions—add, subtract, multiply and divide.
4. For moving data to either the punch or print area.
5. For punching a card.
6. For printing each line of a report.
7. For form control.

Block diagramming can also be compared to flow charting. For both, a sequence of events is determined and recorded clearly and simply. The steps are then converted to machine instructions—stored-program instructions or wired control panel instructions, whichever is appropriate.

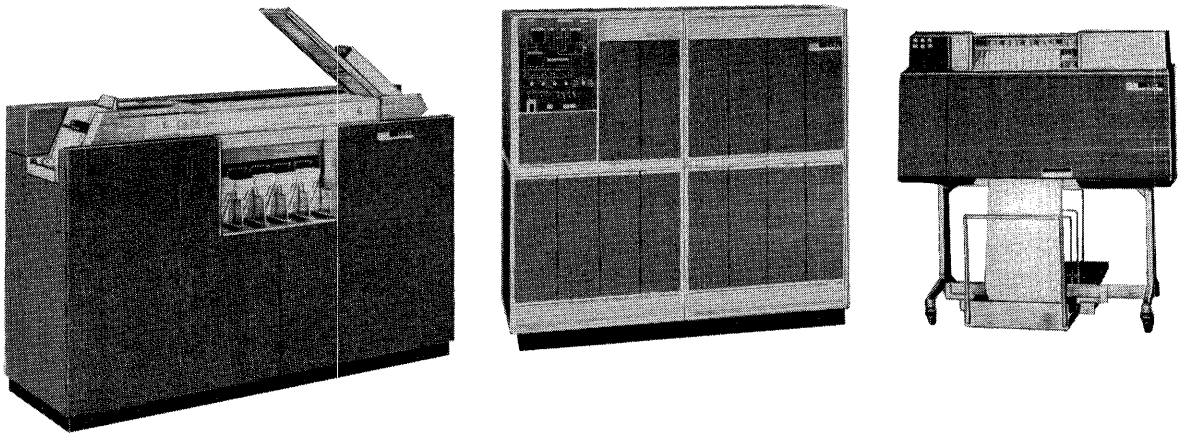
When the machine-coded instructions are written, the instructions become known collectively as the "program" or "routine."

A program or routine is a set of coded instructions arranged in proper sequence to direct the computer to perform a desired operation or series of operations.

Another term, *subroutine,* implies a part of a routine or program. *It is a short or repeated sequence of instructions necessary to solve a part of a problem, and is, therefore, a part of a routine.*

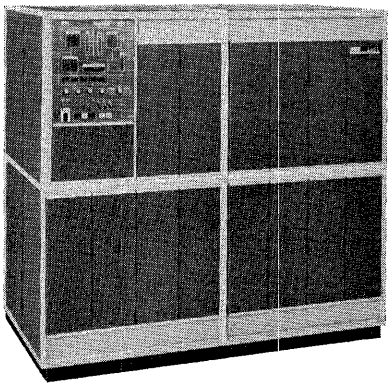
The 1401 Data Processing System

Components

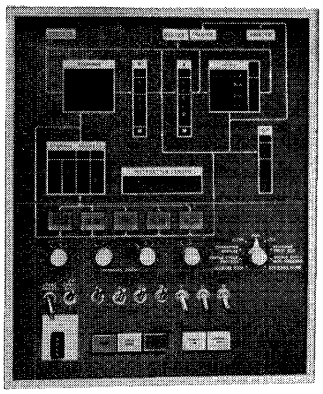


The 1401 Data Processing System is a serial, variable word length, stored-program machine. It is controlled entirely by stored-program instructions; there are *no* control panels in any of the units.

Three units compose the 1401 Data Processing System:

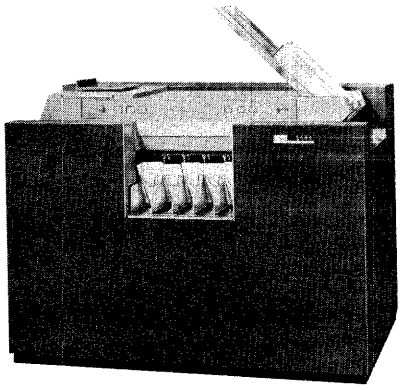


1. The 1401 Processing Unit. In it is found the circuitry necessary to perform machine functions—arithmetic operations, comparing, forms control, etc.; it also houses core storage and the console.



The console consists of (1) buttons and switches necessary to start routines, stop routines, test routines in the planning stages, and restart a routine after it is stopped; and (2) display lights which indicate the contents of storage positions and functions which are occurring.

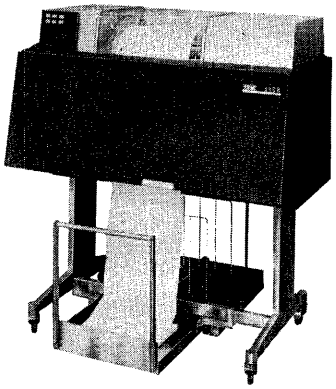
2. The 1402 Card Read Punch. All card reading and punching occurs in this unit. The reading occurs in the right-hand side at a maximum of 800 cards per minute. Punching occurs in the left-hand side at a maximum of 250 cards per minute.



Cards to be read pass two sets of reading brushes. The first set of reading brushes is used for a checking function; the second set completes the check and allows the data to enter storage.

On the punch side, cards first pass a set of punches where punching occurs and then a checking station where the punching is checked.

Both feeds have a "pretest" station for sensing misfeeds. All card input and output is checked for validity. Invalid codes cause the machine to stop.

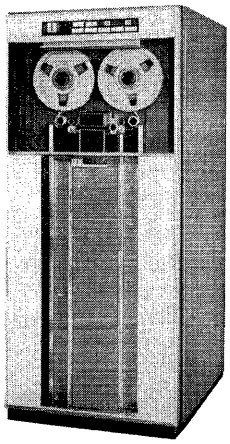


3. The 1403 Printer prints at a maximum of 600 lines per minute and has a print span of 100 positions or 132 positions. Horizontal spacing is 10 characters per inch. Any one of 48 characters—26 alphabetic, 10 numerical and 12 special —can print in each print position. The twelfth special character is the + symbol or the record mark, depending upon the 1401 model.

Output to the printer is checked.

Vertical spacing can be six or eight lines to the inch and is under operator control.

Tapes



Up to six magnetic tape units can be attached to the 1401 Data Processing System. They can be either the 729 II or 729 IV.

1401 Storage

There is a maximum of 4,000 positions of core storage. Each position has a three-digit address.

Three areas of storage are reserved for specific purposes— (1) positions 001-080 for card read-in, (2) positions 101-180 for punching, and (3) positions 201-300 (or 201-332) for printing. When not used for these specified purposes, these areas may be used for any other purpose.

In the 1401, characters are in the binary coded decimal; a parity check is performed for each character on every move.

Instruction Format and Operation Codes

The 1401 processes an application by executing a series of instructions. These instructions must first be determined by the programmer, written, punched into program load cards, tested and then stored in the machine. The machine must have all necessary instructions stored in it before processing the data.

The Instruction

The instruction format is:

OP Code	A/I Address			B Address			Digit Modifier
X	X	X	X	X	X	X	X

(Each x represents a character.)

1. The operation code always consists of a single character. It indicates the function to be performed.
2. The A or I address is always a three-digit address. A identifies a data field and I an instruction field.
3. The B address is always a three-digit address. It always identifies a data field.
4. The digit modifier adds flexibility to the op codes with which it is appropriately used.

Some instructions do not use all four parts. The machine function to be performed determines which parts are used.

Operation Codes

Each of the following sections describes and illustrates a group of operation codes*. The symbols I, A and B refer to type of address; "d" indicates the digit modifier.

(1) Word Mark Instructions

, — WORD MARK SET

OP Code	A Address			B Address		
,	X	X	X	X	X	X

This instruction may include one or two addresses, and will cause a word mark to be set at each of the specified addresses without disturbing data already there.

*For more information on the op codes described in this section, and others not described, refer to the General Information Manual for the 1401 Data Processing System (form D24-1401).

⌑ — WORD MARK CLEAR

OP Code	A Address			B Address		
⌑	X	X	X	X	X	X

Same as WORD MARK SET except that word marks are cleared at the specified addresses.

(2) Input-Output Instructions

1 — READ

OP Code
1

This instruction stops the execution of instructions and reads a card. Then the execution of instructions continues. Contents of all eighty columns automatically go into storage positions 001-080.

2 — PRINT

OP Code
2

This instruction stops the program, prints on the report the contents of the print area, and spaces the form. The print area includes positions 201-300 or 201-332.

3 — PRINT AND READ

OP Code
3

This instruction combines the operations of READ and PRINT. The program stops until printing and reading occur.

4 — PUNCH

OP Code
4

This instruction punches a card with the contents of the punch area (positions 101-180).

5 — READ AND PUNCH

OP Code
5

This instruction combines the READ and PUNCH operations.

6 — PRINT AND PUNCH

OP Code
6

This instruction combines the functions of PRINT and PUNCH.

7 — PRINT, READ, PUNCH

OP Code
7

This instruction combines the functions of READ, PRINT and PUNCH.

(3) Move Operations

M — MOVE

OP Code	A Address			B Address		
M	X	X	X	X	X	X

This instruction causes the field at the A address to be stored in the field at the B address, thus changing the contents of the B field.

Z — MOVE AND ZERO SUPPRESS

OP Code	A Address			B Address		
Z	X	X	X	X	X	X

This instruction is similar to the MOVE (M) instruction except for one important difference: upon completion of the operation, the B field will contain blanks, instead of zeros, to the left of the most significant digit.

D — MOVE DIGIT

OP Code	A Address			B Address		
D	X	X	X	X	X	X

This instruction causes the numerical portion *only*, of the character in the storage position indicated by the A address, to be stored in the position indicated by the B address.

L — LOAD

OP Code	A Address			B Address		
L	X	X	X	X	X	X

This instruction is the same as MOVE, except that the word mark of the A field is transferred to the B field and all other word marks in the B field are cleared.

Y — MOVE ZONE

OP Code	A Address			B Address		
Y	X	X	X	X	X	X

This instruction is similar to MOVE DIGIT, but only the zone portion is moved.

(4) Arithmetic Codes

A — ADD

OP Code	A Address			B Address		
A	X	X	X	X	X	X

This instruction causes the field at the A address to add algebraically to the field at the B address.

S — SUBTRACT

OP Code	A Address			B Address		
S	X	X	X	X	X	X

This instruction causes the field at A address to be algebraically subtracted from the field at B address.

$\overset{+}{0}$ — RESET ADD

OP Code	A Address			B Address		
$\overset{+}{0}$	X	X	X	X	X	X

This instruction is similar to the ADD instruction, except that the B field is set to zero before the A field data is added to it.

$\bar{0}$ — RESET SUBTRACT

OP Code	A Address			B Address		
$\bar{0}$	X	X	X	X	X	X

B field is reset to zero before A field is algebraically subtracted from it.

(5) Compare

C — COMPARE

OP Code	A Address			B Address		
C	X	X	X	X	X	X

This instruction causes the field at the A address to be compared with the field at the B address.

The result of the compare (equal or unequal) is stored in the machine and then checked by a later instruction.

(6) Editing For Numerical Fields

E — EDIT

OP Code	A Address			B Address		
E	X	X	X	X	X	X

Editing is the process of setting up numerical fields for printing. It allows control of symbol and zero printing. Two fields are needed — the data field and a control field. The data field contains only the information to be printed. The control field specifies (1) where commas, decimals, conditional CR or minus symbols are to print, (2) where zero suppression is to stop, and (3) where characters from the data field are to print.

Figure 29 is an example of a data field.

0	0	2	5	7	4	2	6

Figure 29.

It is desired to print the data as it is illustrated in Figure 30, but with a CR between the last digit and the asterisks if the amount is negative.

$			2	,	5	7	4	.	2	6			*	*

Figure 30.

The control word to cause printing as illustrated in Figure 30 is:

$	b	b	b	,	b	b	0	.	b	b	C	R	*	*

Figure 31.

IN FIGURE 31:

(a) Every character that is to print *as itself* (, $. * CR) is placed in the control word position which corresponds to the position in which it is to print.

(b) Any position which should appear as a space during printing contains an & symbol.

(c) Any position representing a digit from the data field is blank in the control word. In Figure 31, b represents a blank.

(d) The rightmost position in which zero suppression is to occur must have a zero.

Among other things editing accomplishes on the 1401 what zero print control and raised hammersplit levers accomplish on the 407 and 402-3 respectively.

(7) Test and Branch Instructions

Branch instructions accomplish on the 1401 what selectors accomplish on wired control panel machines. These instructions test for conditions which can arise during processing, such as negative balances, zero balances, overflow, X or NX, control change, etc. If the condition tested for is in existence, this type instruction causes the machine to skip from the regular instructions to a group which handles that particular condition. Such an instruction must specify the address of the first instruction in the group. If the condition tested for does not exist, the machine executes the next instruction in sequence.

Switching control from one instruction to another instruction which is not the next in sequence is called *"branching"* or "a branch." The instruction which tests for a condition is a branch instruction.

Branch instructions take one of several forms:

(a)

OP Code	I Address		
B	X	X	X

Always branch to the address indicated by I for the next instruction.

(b)

OP Code	I Address			Digit Modifier
B	X	X	X	X

Program branch to the address indicated by I for the next instruction if the condition indicated by the digit modifier exists. Some of the digit modifier codes and the conditions for which they test are illustrated below.

DIGIT MODIFIER	CONDITION
9	A punch in carriage channel 9
@	A punch in carriage channel 12
A	Last Card
Z	Overflow
/	Unequal Compare

(c)

OP Code	I Address			B Address			Digit Modifier
B	X	X	X	X	X	X	X

Program branch to the address indicated by I for the next instruction *if* the storage position indicated by the B address contains the same character that is in the digit modifier portion of this instruction. With this type branch, the digit modifier can be any character (alphabetic, numerical or special) such as C, 1 or an 11 punch.

(d) Another type instruction which accomplishes a test and branch operation is the "test for zone or word mark and branch." Its format is:

OP Code	I Address			B Address			Digit Modifier
V	X	X	X	X	X	X	X

1. The op code, V, identifies it as the "test for zone or word mark and branch" instruction.
2. The digit modifier identifies the type of test. Characters which can appear as digit modifiers, and the condition for which each tests, are illustrated below.

DIGIT MODIFIER	CONDITION FOR WHICH IT TESTS
1	Word Mark
2	No zone (no A bit and no B bit)
B	12 zone (A bit and B bit)
K	11 zone (B bit and no A bit)
S	Zero zone (A bit and no B bit)
3	Either a word mark or no zone
C	Either a word mark or 12 zone
L	Either a word mark or 11 zone
T	Either a word mark or zero zone

3. The B address is the storage portion which is tested.
4. The I address is the location of the next instruction to be executed if the condition exists.
5. If the condition does not exist the machine will execute the next instruction in sequence.

(8) Miscellaneous Instructions

F — FORMS CONTROL

OP Code	Digit Modifier
F	X

This instruction causes the carriage to move as specified by the digit modifier. In Figure 32 are digit modifiers which are used. Opposite each is the carriage operation it will perform.

/ — CLEAR

OP Code	A Address		
/	X	X	X

This instruction is used to clear storage of data and word marks. It clears the position designated by the A address and all successively lower-numbered positions through 00 of that hundred. If the A address is 563, positions 563 through 500 are cleared. If the A address is 599, positions 599 through 500 are cleared.

N — NO OPERATION

OP Code
N

This operation code prevents the machine from doing anything for the instruction in which it appears. The machine just executes the next instruction in sequence.

• — STOP

OP Code
•

This instruction will cause the program to stop and the Stop Process light to turn on.

CHARACTER AT (d) FOR FORMS CONTROL INSTRUCTIONS

DIGIT MODIFIER	IMMEDIATE SKIP TO
1	Channel 1
2	Channel 2
3	Channel 3
4	Channel 4
5	Channel 5
6	Channel 6
7	Channel 7
8	Channel 8
9	Channel 9
0	Channel 10
#	Channel 11
@	Channel 12

DIGIT MODIFIER	SKIP AFTER PRINT TO
A	Channel 1
B	Channel 2
C	Channel 3
D	Channel 4
E	Channel 5
F	Channel 6
G	Channel 7
H	Channel 8
I	Channel 9
$\dot{0}$	Channel 10
.	Channel 11
⌑	Channel 12

DIGIT MODIFIER	IMMEDIATE SPACE
J	1 space
K	2 spaces
L	3 spaces

DIGIT MODIFIER	AFTER PRINT-SPACE
/	1 space
S	2 spaces
T	3 spaces

Figure 32.

How Functions Associated with Wired Control Panel Machines Are Accomplished on the 1401

Card Reading

Depression of the start key on the 1401 reads the first card automatically; thereafter, no cards are read unless the machine executes a read instruction. The op code which initiates card reading is 1, and it appears in storage as:

OP Code
1

Addition

As illustrated in Figure 33, when a card is read, its contents automatically go to the read-in area, positions 001-080. The amount (positions 076-080 in this example) is to be accumulated. It can be accumulated in any available storage positions; positions 575-580 have been chosen.

The instruction to add the amount into position 575-580 is:

OP Code	A Address			B Address		
A	0	8	0	5	8	0

The op code, A, instructs the machine to add. The A address, 080, identifies the card data field. The B address, 580, identifies the units position of the area for accumulation; an area for accumulation is the equivalent of a counter.

When the 1401 adds, the adding mechanism takes the factor indicated by the A address and the factor indicated by the B address, adds the two together and puts the sum of the two back in the B field.

The sales amount from the card remains in positions 076-080 until another card feeds in and replaces it.

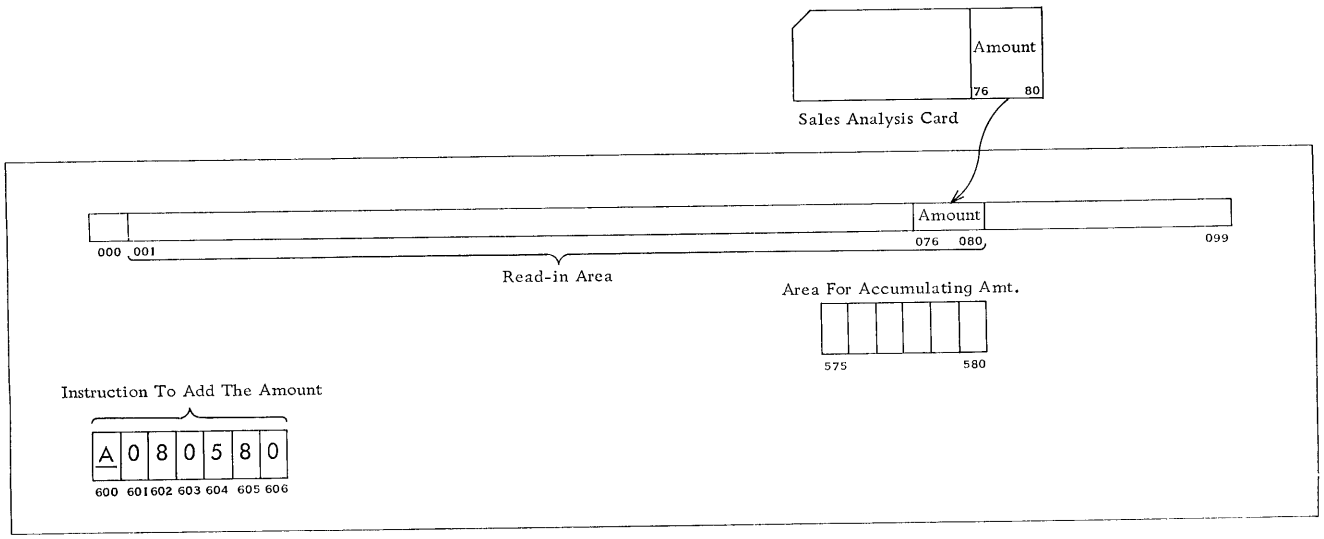


Figure 33. 1401 Storage — Addition

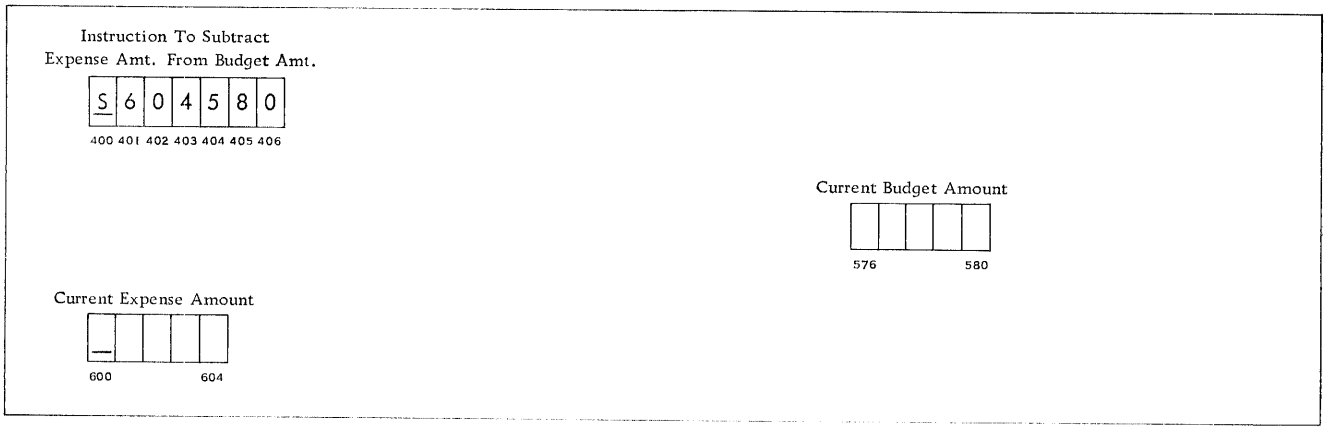


Figure 34. 1401 Storage — Subtraction. Current Budget Amount — Current Expense Amount = Current Over/Under Budget

Subtraction

In Figure 34 appear the data fields and the instruction in 1401 storage. Current Expense Amount (storage positions 600-604) is to be subtracted from Current Budget Amount (positions 576-580).

The 1401 instruction (positions 400-406) to subtract Current Expense from Current Budget Amount is:

OP Code	A Address			B Address		
S	6	0	4	5	8	0

S is the subtract op code; the A address identifies the amount to be subtracted, and the B address identifies the factor from which it is subtracted. The machine subtracts Current Expense Amount (positions 600-604) from Current Budget Amount (positions 576-580) and places the difference, Current Over/Under Budget, in positions 576-580.

Crossfooting

Figure 35 represents data factors and the instructions for crossfooting. OASI, federal withholding tax and UCI are crossfooted to give total deductions for each employee.

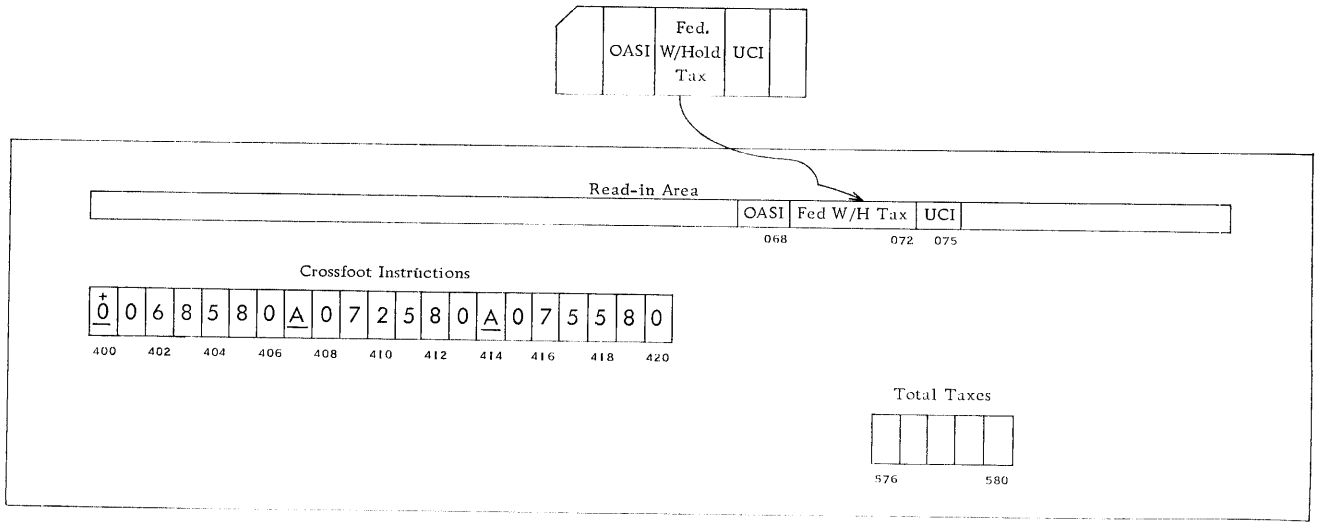


Figure 35. 1401 Storage — Crossfooting

The 1401 requires a series of program instructions to accomplish crossfooting just as a wired control panel accounting machine requires a series of program cycles. They are:

	OP Code	A Address			B Address		
1.	$\overset{+}{0}$	0	6	8	5	8	0
2.	A	0	7	2	5	8	0
3.	A	0	7	5	5	8	0

The first instruction, $\overset{+}{0}$ 068 580, resets to zero the accumulate area before adding OASI into positions 576-580. Instruction 2 adds federal withholding tax, and the third adds UCI. After the third instruction is executed, total taxes are in positions 576-580.

Recognizing Negative Balances

The 1401 signals a negative balance by placing a credit X over the units position of the amount, as shown in Figure 36. (The X is a B bit.)

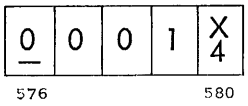


Figure 36. Negative Amount in 1401 Storage

In order to recognize a negative balance when it arises, a "test for zone and branch" instruction is executed. It tests the units position of the amount field for the X. The instruction is:

OP Code	I Address			B Address			Digit Modifier
V	6	8	4	5	8	0	K

1. The B address indicates the units position, which is to be tested for the credit X.
2. The digit modifier, K, tells the machine what to test for — a credit X.
3. The op code, V, instructs the machine that this is a "test for zone and branch" instruction.
4. The I address indicates the location of the next instruction to be executed if the amount is negative.
5. If the amount is positive, the next instruction executed will be the next one in succession.

The instructions and the amount to be tested, as they appear in 1401 storage, are illustrated in Figure 37:

1. Positions 401-408 contain the instruction which tests the amount.
2. Positions 576-580 contain the amount to be tested.
3. Positions 409-415 contain the instruction to be executed if the amount is positive.
4. Position 684 contains the instruction to be executed if the amount is negative. It stops the machine.

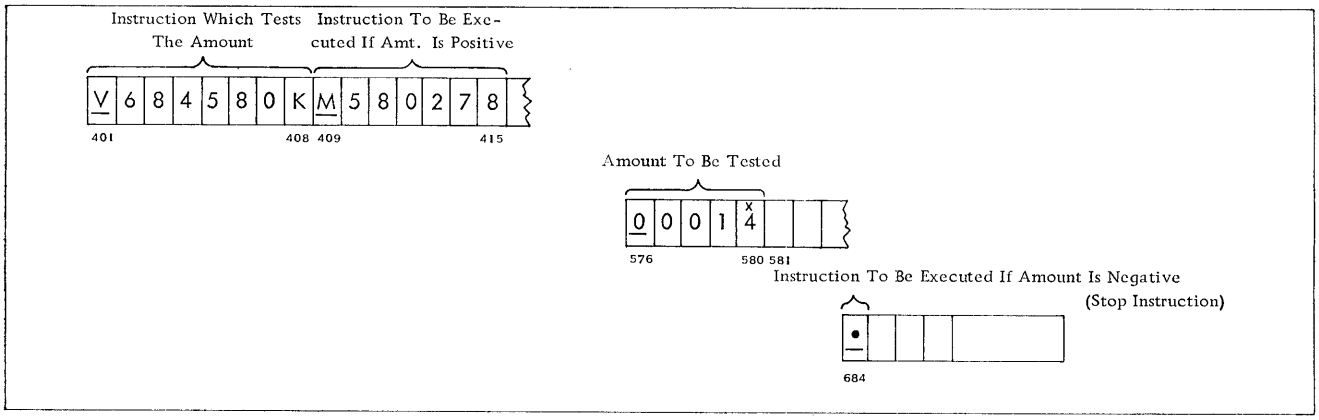


Figure 37. 1401 Storage — Recognizing Negative Balances

Counter Coupling

There is no counter coupling on the 1401. The size of the area for accumulation is determined by the programmer when writing the program; it can be any size. It is defined in the machine by a word mark in the high-order position of the field (see Figure 38).

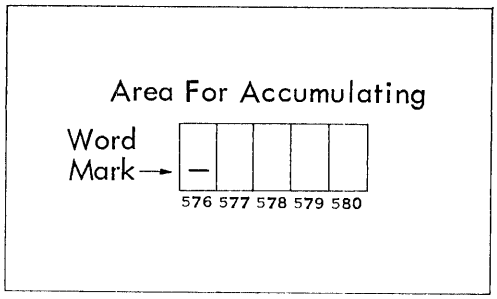


Figure 38. 1401 Storage

If it is desirable to change the size of the accumulate area from five to seven positions, it is done with two instructions:

	OP Code	A Address		
1.	¤	5	7	6
2.	,	5	7	4

Instruction 1 erases the word mark from position 576. Instruction 2 sets a word mark in position 574. Figure 39 illustrates the enlarged accumulate area and the instructions which increased its size.

Comparing

To compare, on wired control panel machines, the control number in a card at one reading station is compared with the control number in the following card at another reading station.

On the 1401, two cards cannot be read simultaneously for comparing purposes. However, the control number in each card must be compared with the control number in the following card. To accomplish this, a section of storage is used to hold the control number of each card after it is processed and until the next card is brought into the machine (see Figure 40).

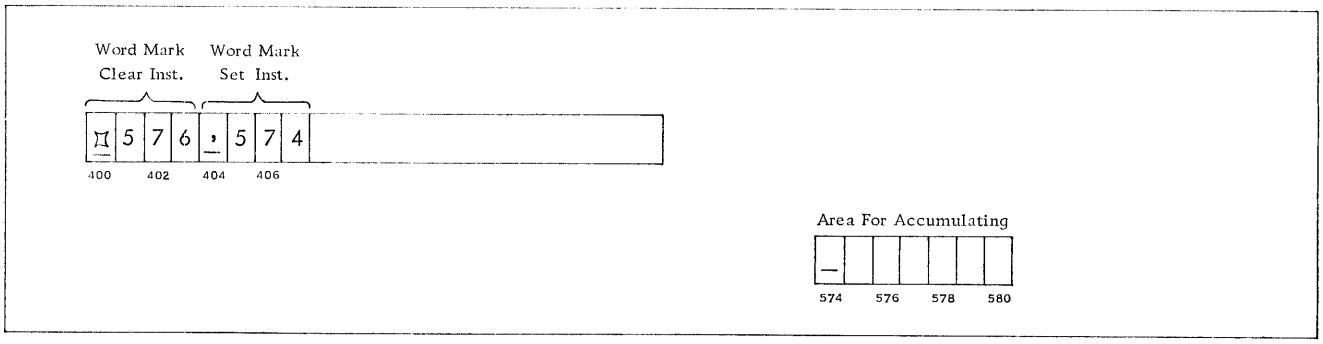


Figure 39. 1401 Storage

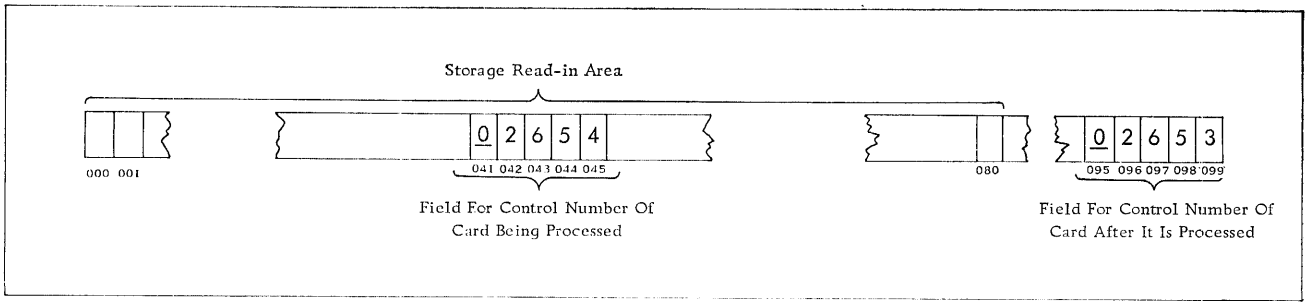


Figure 40. 1401 Storage

The control numbers are compared by this instruction:

OP Code	A Address			B Address		
C	0	4	5	0	9	9

This instruction accomplishes for the 1401 the same thing that wiring from reading brushes to comparing entry accomplishes for wired control panel machines. The result of the comparison (equal or unequal) is remembered by the machine and can be determined by a branch instruction such as:

OP Code	I Address			Digit Modifier
B	6	0	0	/

This instruction accomplishes on the 1401 the same thing that wiring from comparing exit to minor program start accomplishes on wired control panel accounting machines.

The digit modifier causes the test for the condition of the previous compare. The I address is the location of instructions to be executed if the comparison was unequal; if the comparison was equal, the next instruction executed will be the following instruction. The op code B identifies this as a branch instruction.

Storage containing instructions and data is illustrated in Figure 41.

Program Control

On wired control panel accounting machines, program control is the recognition of change in different control fields (minor, intermediate and major) and then the initiation of a program step for each level in which there is a change.

To accomplish program control on the 1401, changes in minor, intermediate and major control fields must be recognized and a series of program instructions executed for each level of change. Assuming minor, intermediate, and major program control, there must be:

1. For minor program control
   a. a minor control number
   b. a minor compare and a minor branch instruction
   c. a set of minor instructions which are executed when there is a minor control change
2. For intermediate program control
   a. an intermediate control number
   b. an intermediate compare and an intermediate branch instruction
   c. a set of intermediate instructions which are executed when there is an intermediate control change
3. For major program control
   a. a major control number
   b. a major compare and a major branch instruction
   c. a set of major instructions which are executed when there is a major control change

1401 storage illustrating minor, intermediate and major program control appears in Figure 42.

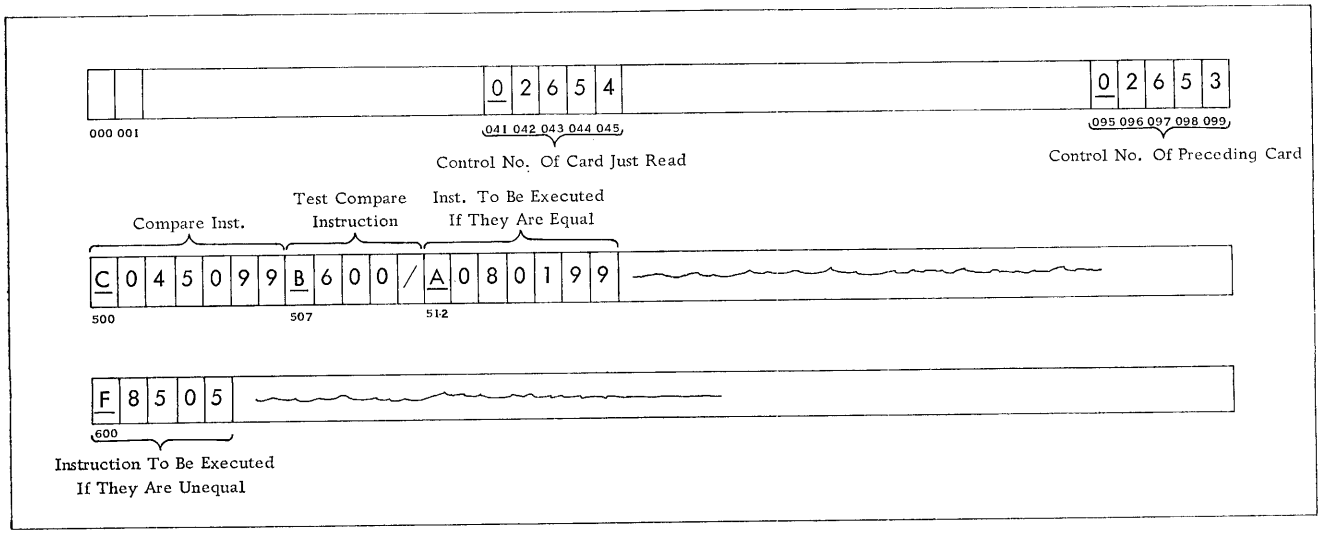


Figure 41. 1401 Storage

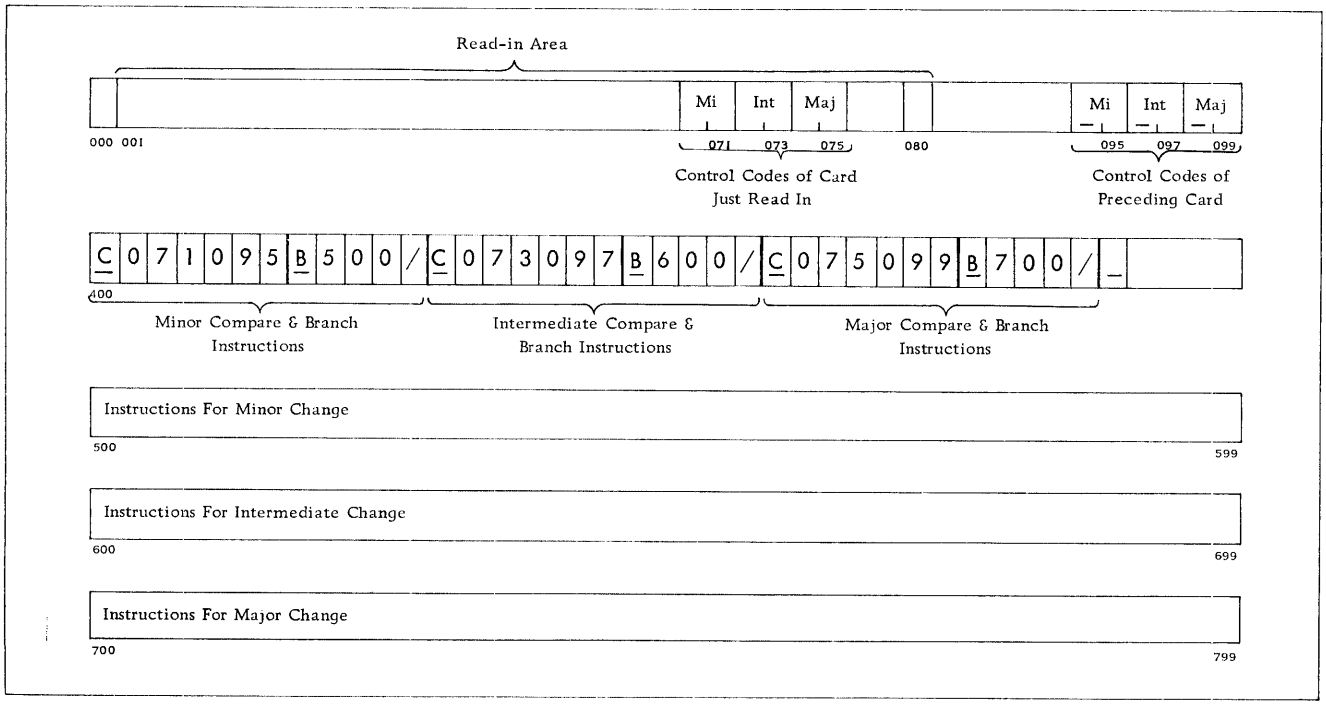


Figure 42. 1401 Storage — Program Control

In Figure 42, each card's control fields are retained in the machine until the following card is read. A compare instruction and a branch instruction exist for each level of control. The minor branch instruction sends program control to storage position 500. Beginning in position 500 are the instructions for a minor change. Intermediate branch sends control to position 600 for intermediate instructions, and major branch sends control to position 700 for major instructions.

Recognizing Zero Balances

Figure 43 illustrates, and the text following it explains, a 1401 method of "stop for zero balance."

In Figure 43:

1. The balance to be tested for zeros is in positions 495-499.
2. The field of zeros against which comparison is to be made is in positions 400-404. These zeros are stored in these positions at the same time the program instructions are loaded.
3. The compare instruction in positions 600-606 determines whether or not the balance is zero. If the compare is equal, the balance is zero; if the compare is unequal, the balance is non-zero.
4. The branch instruction, positions 607-611, tests the compare operation. For unequal compare, indicating a non-zero balance, program control is sent to position 613 for the next instruction. If the compare is equal, indicating a zero balance, a branch does not occur. Instead, the machine executes the next instruction in sequence, which is a machine stop instruction.

X Selection

With wired control panel machines, X selection is performed by wiring a card column signaling the condition X or NX to the X pickup of a selector. If nothing is punched in the card column, the selector remains normal; if the card column contains the character, the selector transfers. The impulse traveling through the points of the selector determines what happens. If it emerges from the transferred hub, it represents an X condition. If it emerges from the normal hub, it represents a NX condition.

Wired control panel machines accomplish X selection with wires and selectors. In stored-program machines, instructions accomplish selection.

To test the condition X or NX, a branch instruction is used. Figure 44 illustrates X selection on the 1401.

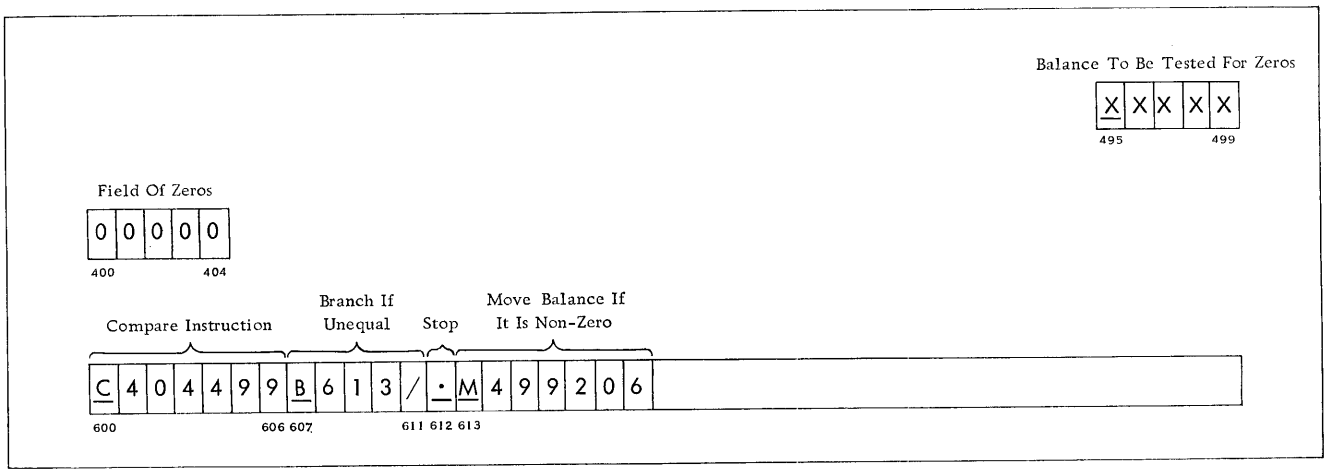


Figure 43. 1401 Storage — Recognizing Zero Balances

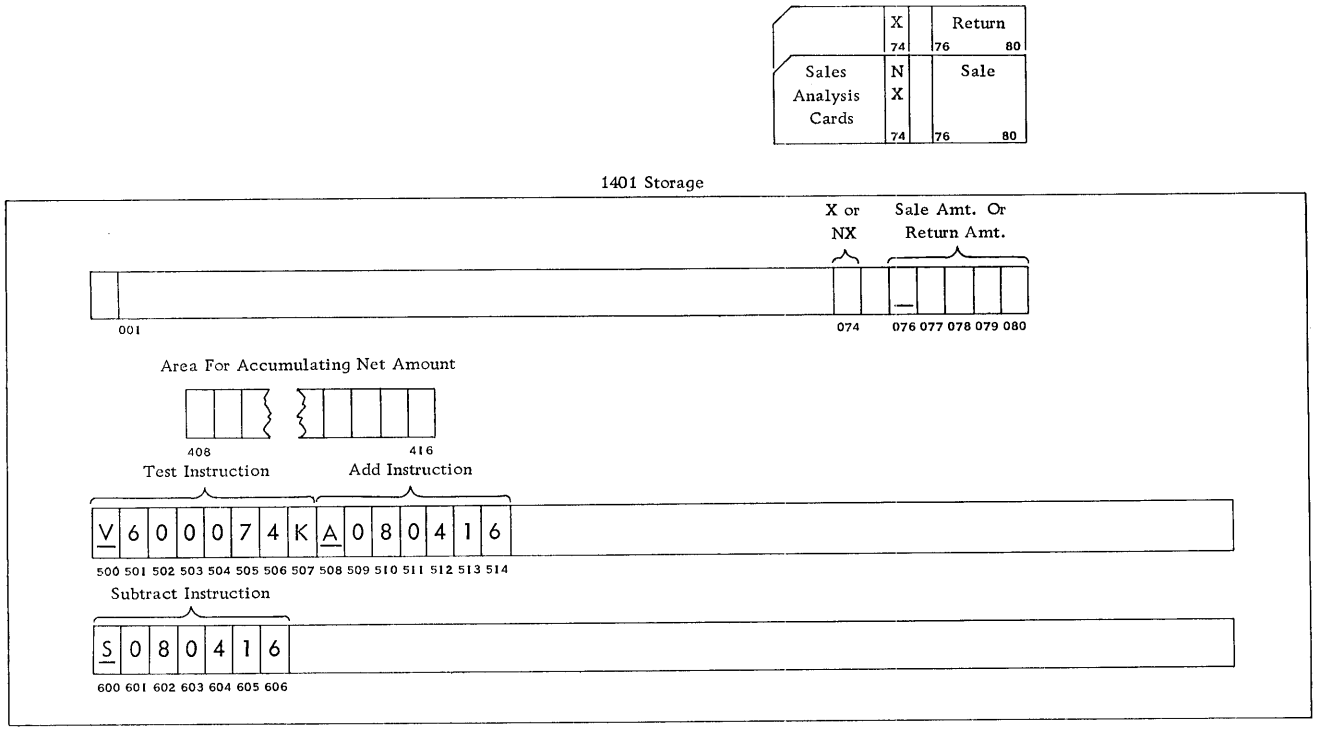


Figure 44. X Selection on the 1401

IN FIGURE 44:

1. The test instruction is in positions 500-507.
2. The op code, V, identifies this as a test instruction.
3. The digit modifier, K, indicates that the B address (position 074) is to be tested for the presence of the 11 zone or X.
4. The A address (position 600) is the location of the instruction to be executed if there is an X in position 074; it is a subtract instruction.
5. If the test results in a NX, the next instruction in sequence will be executed. It is an add instruction.

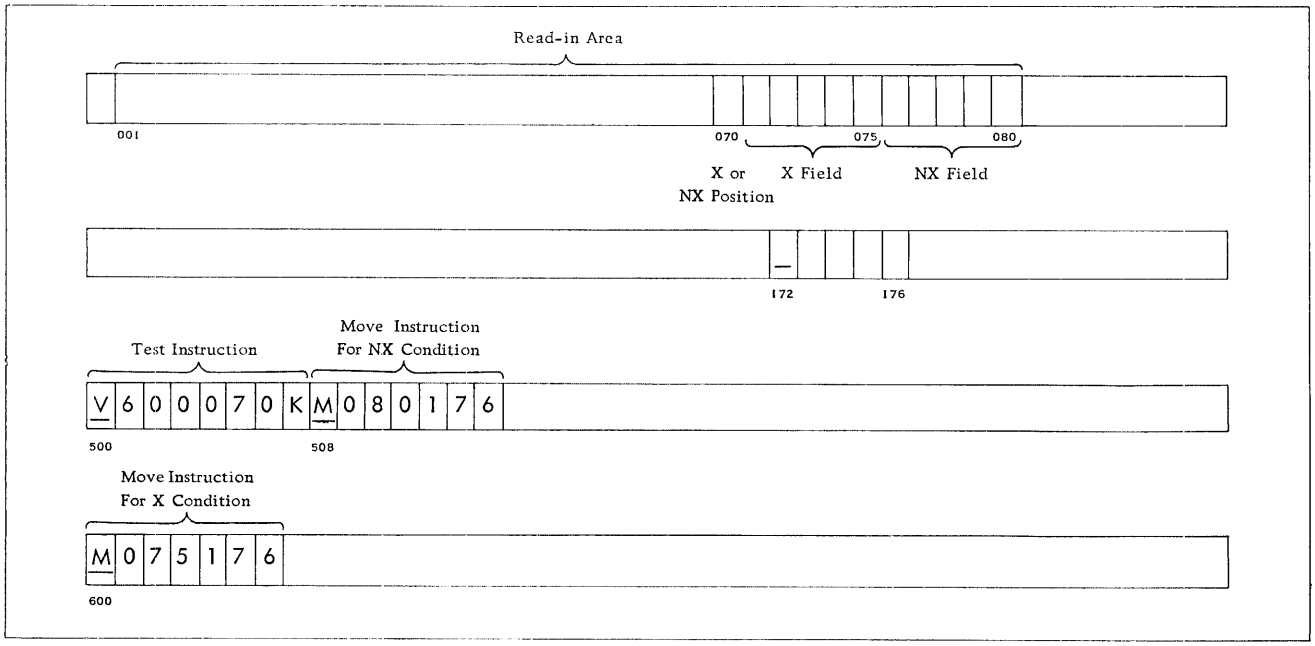


Figure 45. 1401 Storage — Field Selection

Digit Selection

Refer to Figure 44 and assume the same problem with the exception that a 1 or no 1 in column 74 is the condition.

The instruction which tests is:

OP Code	A Address			B Address			Digit Modifier
B	6	0	0	0	7	4	1

It says, branch (op code B) to location 600 (A address) for the next instruction if position 074 (B address) contains a 1 (digit modifier). If position 074 does not contain a 1, the next instruction in sequence will be executed.

Field Selection

Figure 45 illustrates 1401 storage containing the instructions and the data fields for field selection.

For this example, if position 070 contains an X, positions 071-075 are to be moved to positions 172-176; if position 070 is NX, positions 076-080 are to be moved to positions 172-176.

In Figure 45:

1. The instruction in positions 500-507 tests for the X or NX condition.
2. In positions 508-514 is the move instruction for the NX condition, that is, move the contents of positions 076-080 to 172-176.
3. In positions 600-606 is the move instruction for the X condition, that is, move the contents of positions 070-075 to 172-176.

Class Selection

1401 class selection is illustrated in Figure 46:

1. If position 040 contains an X, the amount in positions 076-080 is moved to positions 186-190; if NX, it is moved instead to positions 195-199.
2. Positions 500-507 contain the "test for zone" instruction.
3. The move instruction for a NX condition is located in positions 508-514; it moves amount to positions 195-199. The move instruction for an X condition is in positions 600-606; it moves the amount to positions 186-190.

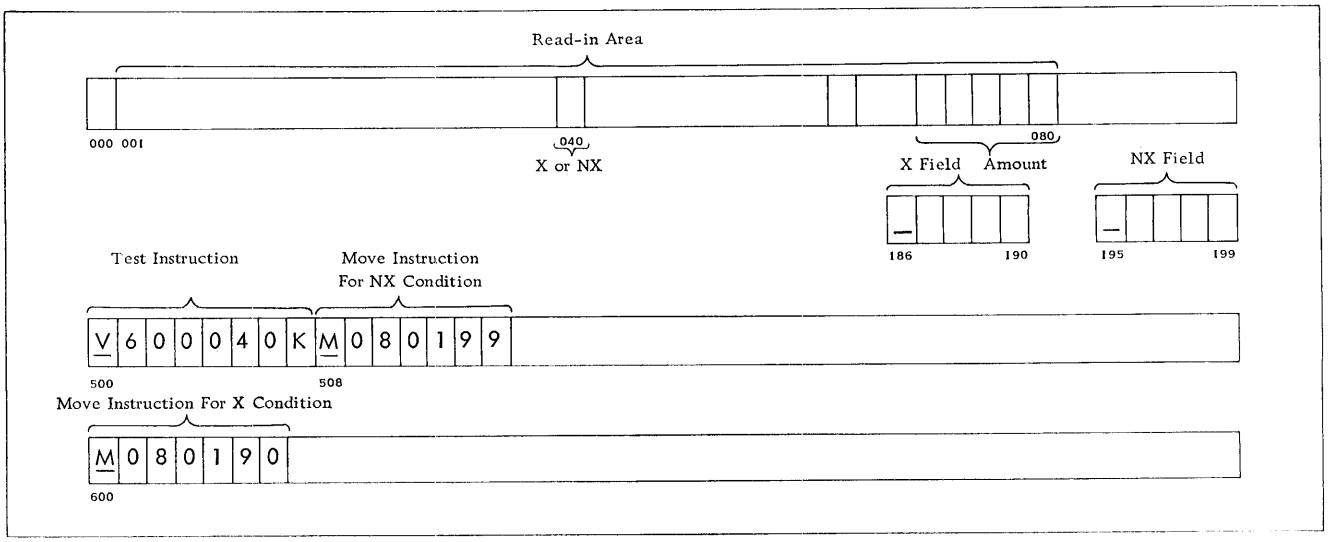


Figure 46. 1401 Storage — Class Selection

Carriage Control

The 1403 Printer has a tape-controlled carriage. Channel usage is much the same as it is with wired control panel accounting machines. Channels 1-12 are used to stop the form at any one of the twelve positions; channel 12 is also used for sensing overflow.

Carriage skips are initiated by instructions and stopped by punches in the carriage tape. The operation code signals to the machine a carriage skip, and the digit modifier indicates the channel to which the skip is made. Skips can occur immediately or after the next print operation.

Examples of "immediate skip to" and "skip after printing to" instructions are in Figure 47. The op code, F, initiates carriage skip. The digit modifier indicates the channel.

FUNCTION	INSTRUCTION	
	OP CODE	DIGIT MODIFIER
Immediate Skip to Channel 1	F	1
Immediate Skip to Channel 2	F	2
Immediate Skip to Channel 3	F	3
Immediate Skip to Channel 4	F	4
Immediate Skip to Channel 5	F	5
Immediate Skip to Channel 6	F	6
Immediate Skip to Channel 7	F	7
Immediate Skip to Channel 8	F	8
Immediate Skip to Channel 9	F	9
Immediate Skip to Channel 10	F	0
Immediate Skip to Channel 11	F	#
Immediate Skip to Channel 12	F	@

FUNCTION	INSTRUCTION	
	OP CODE	DIGIT MODIFIER
Skip after print to Channel 1	F	A
Skip after print to Channel 2	F	B
Skip after print to Channel 3	F	C
Skip after print to Channel 4	F	D
Skip after print to Channel 5	F	E
Skip after print to Channel 6	F	F
Skip after print to Channel 7	F	G
Skip after print to Channel 8	F	H
Skip after print to Channel 9	F	I
Skip after print to Channel 10	F	+ 0
Skip after print to Channel 11	F	.
Skip after print to Channel 12	F	␣

Figure 47. Carriage Control

Another carriage function is overflow control. The branch instruction which checks for overflow is:

OP Code	I Address			Digit Modifier
B	X	X	X	@

1. The digit modifier @ instructs the machine to determine whether a punch in channel 12 has been sensed.
2. The I address indicates the location of the first instruction in the overflow routine. The overflow routine contains the necessary instructions for skipping to the next form, printing indicative information, etc., and then skipping to the body line of that form.
3. If there is no overflow, the next instruction in sequence is executed.

Printing

The instruction which initiates printing is:

OP Code
2

In 1401 storage is a constant print area. It consists either of positions 201-300 or positions 201-332. (The machine can be ordered with either 100 or 132 print positions.)

Data to be printed is assembled prior to execution of the print instruction. When the print instruction is executed, each storage position prints in the correspondingly numbered print position. (See Figure 48.)

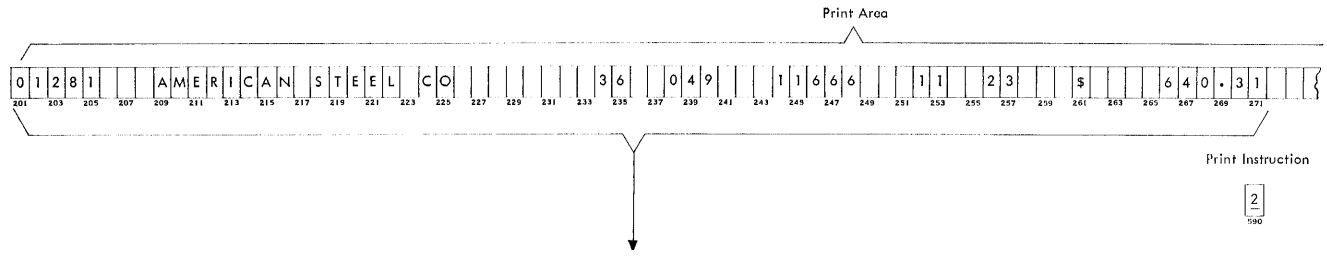
To assemble the print area means to put it together by moving to the print area all fields to be printed. This can be done with a series of move instructions. For example, assume that customer name in Figure 48 is located in storage positions 008-029. To assemble the customer name in positions 209-230 of the output area, the move instruction is:

OP Code	A Address			B Address		
M	0	2	9	2	3	0

Each other field to be printed can be moved with a similar move instruction.

In Figure 48:

1. The first position of the print area prints in the first print position on the report. Each storage position thereafter prints in correspondingly numbered print positions.
2. Each print position that is to appear blank on the report must have its corresponding print position blank.



ACCOUNTS RECEIVABLE REGISTER

CUSTOMER NUMBER	CUSTOMER NAME	LOCATION		INVOICE NUMBER	INV. DATE		INVOICE AMOUNT
		STATE	CITY		MO.	DAY	
01281	AMERICAN STEEL CO	36	49	11666	11	23	$ 640.31

Figure 48. 1401 Printing

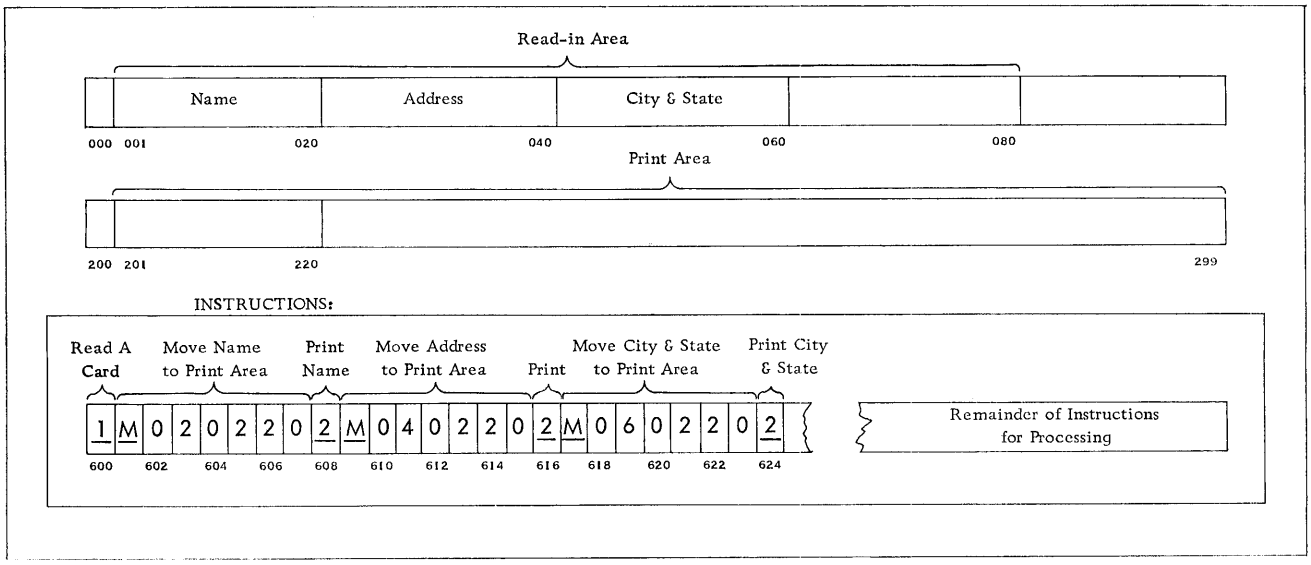


Figure 49. 1401 Storage — Printing Multiple Lines from a Single Card

Multiple Line Read or Multiple Line Print

On the 1401, the printing of multiple lines from a single card is accomplished with program instructions. Figure 49 illustrates, and the following text explains, printing of multiple lines from a single card.

1. When the 1401 reads a card, the entire 80 columns enter storage.
2. The section labeled "Instructions" in Figure 49 accomplishes multiple line printing from a single card by reading a card and then executing, several times, assemble and print instructions for that card before reading another card.

Punching

The instruction which initiates punching is:

OP Code
4

The 1401 has an area of storage (positions 101-180) designated as the punch area. The data to be punched should be assembled in this area prior to punching, because, when the punch instruction is executed, every punch position is punched into the correspondingly numbered card column. (See Figure 50.) If a card column is to be blank, the corresponding storage position of the punch area must contain a blank.

To assemble the information to be punched requires the execution of a series of move instructions, generally a move instruction for each field to be punched. Figure 50 illustrates (1) the assemble instructions for punching the Current Earnings card, (2) the data to be assembled, (3) the punch area with the data after assembly and (4) the Current Earnings card punched from the output area.

Detail Printing and Group Printing

Detail printing is accomplished on the 1401 by having the machine execute a print instruction for every card that is read, thereby printing something from each card.

Group printing is performed by locating the print instruction in the program so that it is executed only once per control group, thus allowing one line of printing per group.

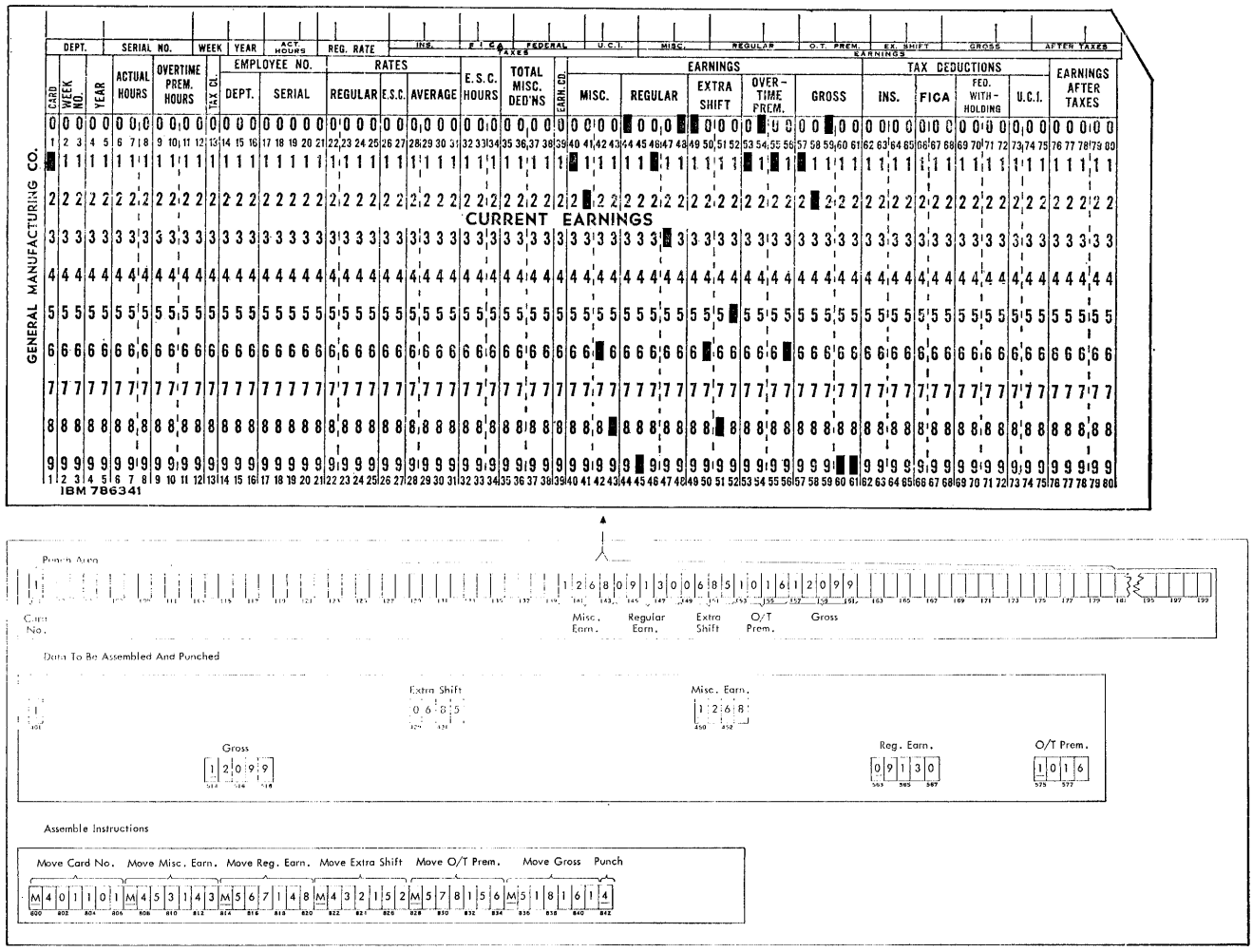


Figure 50. 1401 Storage — Punching

Emitting

The effect of emitting is accomplished on the 1401 by (1) storing in the machine in chosen positions the characters (alphabetic, numerical and special) and phrases needed and (2) executing a move instruction each time a character or phrase is needed. Examples of constants are:

1. The digits 0-9
2. The alphabetic characters A-Z
3. The twelve special characters
4. Phrases such as "Minor Total," "Intermediate Total," etc.
5. Words such as "Discount," "Net," etc.

The phrases, words and characters which will be needed are determined during the application's planning stages. They are punched into cards and loaded into the machine at the same time instructions are loaded.

Figure 51 illustrates a 1401 storage area containing constants. To have the phrase "Minor Total" appear beside the minor amount when it is printed, the move instruction in positions 500-506 is executed. It moves "Minor Total" from positions 736-746 to positions 265-275 of the print area. To print, then, a print instruction is executed after all necessary information is assembled in the print area.

For every character or group of characters generated from within the machine, the machine must execute a move instruction which moves the constant or constants to the print area.

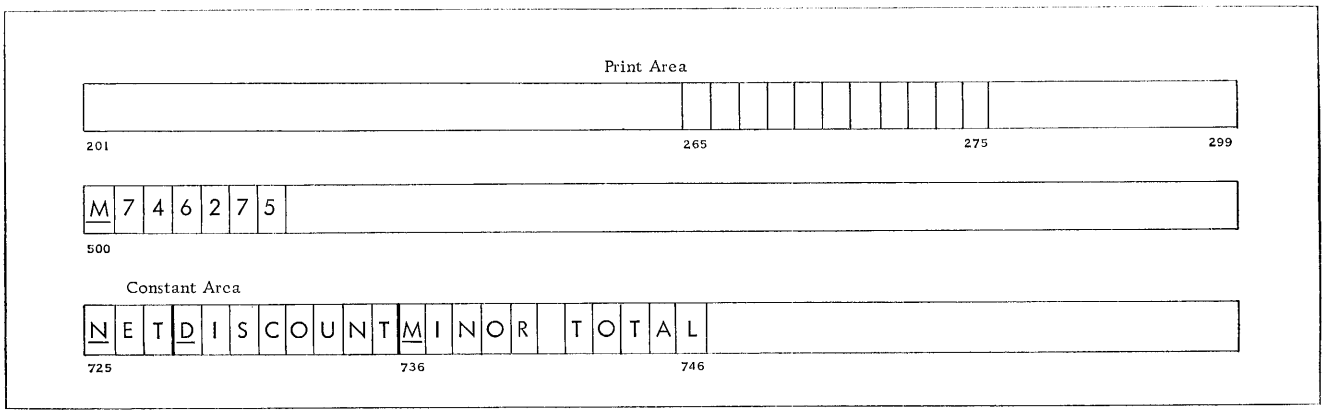


Figure 51. 1401 Storage — Emitting

Glossary

ADDRESS — A designation, usually numerical, of a location or of a device where information is stored.

ARITHMETIC — The four functions: addition, subtraction, multiplication and division.

BINARY CODED DECIMAL — The internal coding used by the 1401 to represent numbers, letters and special characters.

BIT — The smallest element of the binary coded decimal. Each bit can be either on or off. Each coded character is composed of seven bits.

BLOCK DIAGRAM — A graphic illustration of the machine steps necessary to perform a given application.

BRANCH — (1) Noun. — A point in a routine at which one of two or more alternatives is chosen under control of the routine. (2) Verb. — To break out of the one-after-the-other sequence of instruction execution.

BRANCH INSTRUCTION — An instruction which causes a branch to occur.

CONSOLE — That part of the computer where the operator has his working place, equipped with keyboard to the computer and all operating switches and lights.

CONTROL — (1) The section of a computer which controls all operations of the system. It may be compared with an automatic telephone exchange. (2) Method used to monitor or guide a program.

CORE (Magnetic) — Small doughnut-shaped device used for storage in computers. Its value is determined by its condition of magnetization.

DIGIT MODIFIER — A single character which, when included in an instruction, makes more specific the function controlled by the op code.

EDITING — Rearranging information. Editing may involve deletion of unwanted data, selection of pertinent data, insertion of information prior to printing, zero suppression, etc.

FIELD — Data in one or more adjacent storage locations to be treated as a unit.

FIXED WORD LENGTH — Condition in which all storage fields have a set capacity or length, in contrast to variable word length.

HIGH-ORDER POSITION — Leftmost position of a field.

INPUT — Any information which enters a machine for the purpose of being processed or to aid in processing.

INPUT AREA — A part of storage allocated to receive a record from an input unit. Input areas usually are defined further according to the kind of input unit, i.e., tape input area, card input area, and drum input area.

INSTRUCTION — A set of characters which, as a unit, cause a computer to perform one of its operations.

INSTRUCTION FIELD — A storage field containing an instruction.

LOAD OR LOADING — The operation of transferring data from an input device into storage.

LOW-ORDER POSITION — Rightmost position of a field.

MICROSECOND — A millionth of a second. (0.000001 seconds)

MILLISECOND — A thousandth of a second. (0.001 seconds)

ODD BIT COUNT — See parity check.

OPERATION (op) CODE — That part of an instruction designating the operation to be performed.

OUTPUT — The results produced by a computer, in the form of tape, punched cards, or printed documents.

OUTPUT AREA — A part of storage allocated to hold a record to be written on an output unit. Output areas are usually defined further according to the kind of output unit, i.e., print area, punch area, tape output area.

PARITY CHECK — In the 1401, a check in which the number of "on" bits in a character is determined and the sum checked for an odd number. It is a process of testing the correctness of information transmitted within the machine.

PROGRAM — (1) Noun. — A set of machine instructions which causes a computer to process data and to produce specific results. (2) Verb. — To plan the method of attack and the necessary instructions to completely process specific data.

PROGRAM LOAD CARDS — Cards containing instructions and constants used in loading the program.

PROGRAMMING — See Program.

READ IN — To transcribe, usually from input devices to storage, or from one storage area to another storage area.

READ OUT — To transfer information from a storage area.

ROUTINE — A set of coded instructions arranged in proper sequence to direct the computer to perform a desired operation or series of operations.

SERIAL TRANSMISSION — Moving data in sequence, one character at a time.

STORAGE FIELD — One or more adjacent storage positions which are grouped and used as a unit.

STORAGE POSITION — A device capable of storing one character.

STORAGE UNIT OR STORAGE — (1) The unit which holds or retains items of information. (2) Any device into which information can be introduced, held, and extracted at a later time.

STORED INSTRUCTIONS — Instructions located within the storage area of the machine.

STORED-PROGRAM MACHINE — A machine whose functions are controlled by coded instructions within the machine.

SUBROUTINE — A short or repeated sequence of instructions necessary to solve a part of a problem. Hence, a part of a routine.

VARIABLE WORD LENGTH — Condition in which the number of positions in a storage field or computer word is almost completely under the control of the programmer or coder. (Contrast to fixed word length.)

WIRED CONTROL PANEL MACHINE — A machine whose functions are controlled by a wired control panel.

WIRED INSTRUCTIONS — Instructions which are wired in on a control panel.

WORD — A set of characters treated by the computer circuits as a unit and transported as such.

WORD MARK — An "on" bit (signaled in the 1401 by the eighth core position) used in defining the limits of storage fields.