COPY NO. 387

Robert F. Frocchi

DATAmatic 1000
PROGRAMMING MANUAL
VOLUME I

DATAmatic

A Division of Minneapolis-Honeywell Regulator Company

151 Needham Street

Newton Highlands 61, Massachusetts

Litho in U.S.A.

DSI - 5A

PREFACE

This manual is intended to serve as textbook and reference for the programmer who will be working with a DATA m a t i c 1000 Electronic Data-Processing System. To avoid duplication, attention will be focused upon the system operations, but a basic description of the various units and functioning of the system will be included.

Volume I describes primarily the computational handling of data within the system and defines the various DATA m a t i c 1000 instructions available to the programmer. Operation of the Central Processing System is discussed in Section I; the Peripheral Equipment is treated in Sections II - VII.

In Volume II, a typical program is developed and analyzed. Large-volume data-handling problems, including sorting and all types of file maintenance, are carefully discussed and the DATA m a t i c 1000 control aids for such problems emphasized. Various aids to the programmer, such as the DATA m a t i c ABC-1 Automatic Business Compiler System, are discussed. Compiler notation, including the use of the Library of Subroutines, is explained, and the use of Utility Routines to assist in the modification and correction of programs is described. Specific sections are devoted to sorting and multiple file maintenance.

TABLE OF CONTENTS

INTRODUCTION

SECTION I — CENTRAL PROCESSOR

TABLE OF CONTENTS (Continued)

SECTION II — MODEL 1200 CARD INPUT SYSTEM

SECTION III MODEL 1300 OUTPUT CONVERTER

Typical Layout of a DATAmatic 1000 Electronic Data-Processing System.

INTRODUCTION

The DATAmatic 1000 is a high-capacity electronic data-processing system designed specifically for application to the increasingly complex problems and procedures required in modern business. The system incorporates significant new systems techniques as well as several basically new component developments. One of the primary features of the DATAmatic 1000 is its exceptionally large capacity to store information on magnetic tape, coupled with its ability to feed information from magnetic tape to the processing section and back to magnetic tape at a sustained rate of 60,000 decimal digits per second. In addition, the operational speed of the processing section maintains full compatibility with this high speed of information transfer.

Two of the most cumbersome aspects of most business problems are sorting and file maintenance. The DATAmatic 1000 is equipped with an extensive and flexible set of instructions, designed specifically to excel in the performance of these functions and many others. These instructions may be automatically assembled into complete programs by the DATAmatic ABC-1 Automatic Business Compiler. Thereafter, a task which is repeated daily or weekly is initiated simply by reusing the program from its storage on the program magnetic tape.

In the DATAmatic 1000, reliability is a prime consideration throughout every aspect of engineering and design. The design of electronic circuitry is highly conservative. Every transfer of information within the system is carefully checked to insure that the information is transferred without alteration. In addition, all arithmetic and logical operations are completely checked. All units of the system are constructed of easily replaced standard packages to facilitate maintenance. A system of marginal checking includes circuitry and a special program which may be run periodically to locate any package which should be replaced because of marginal performance. With proper use of this facility, most machine malfunctions will be corrected before they occur.

Elements of the System

The system may be conceived functionally as comprising three main sections, the Input, Central Processor, and Output Sections, although its physical layout will generally not correspond with such a conception. Data is initially fed into the Input Section in the form of punched cards. This section, which includes a Card Reader, an Input Converter, and one Magnetic File Unit, reads the data from the cards, translates it into machine language, edits and arranges it into the desired format, and records it on magnetic tape.

A High-Speed Memory Amplifier package representative of the modular construction used throughout the DATAmatic 1000 system.

The Central Processing Section includes (1) Arithmetic and Control Units, (2) the High-Speed magnetic-core Memory, (3) Magnetic File Units, (4) Input and Output temporary storage Buffers, and (5) the Central Console. The Central Processor reads data stored permanently on magnetic tape, performs all manipulations of data, controls the sequence of functions performed, stores information temporarily while it is being processed and, after processing, stores it permanently on magnetic tape. By means of the Central Console, the operator may monitor the overall operation of the system. As needed, Magnetic File Units may be used by auxiliary equipment. Such action is controlled by switches.

The Output Section converts data from magnetic tape into either punched-card form or printed form, performing considerable editing in the process. The Model 1300 Output Converter, which feeds standard punching and/or printing equipment, may either replace or supplement the Model 1400 Output Printing System, depending on the quantitative requirements of the system for output information. One Magnetic File Unit may be considered a part of the Output Section.

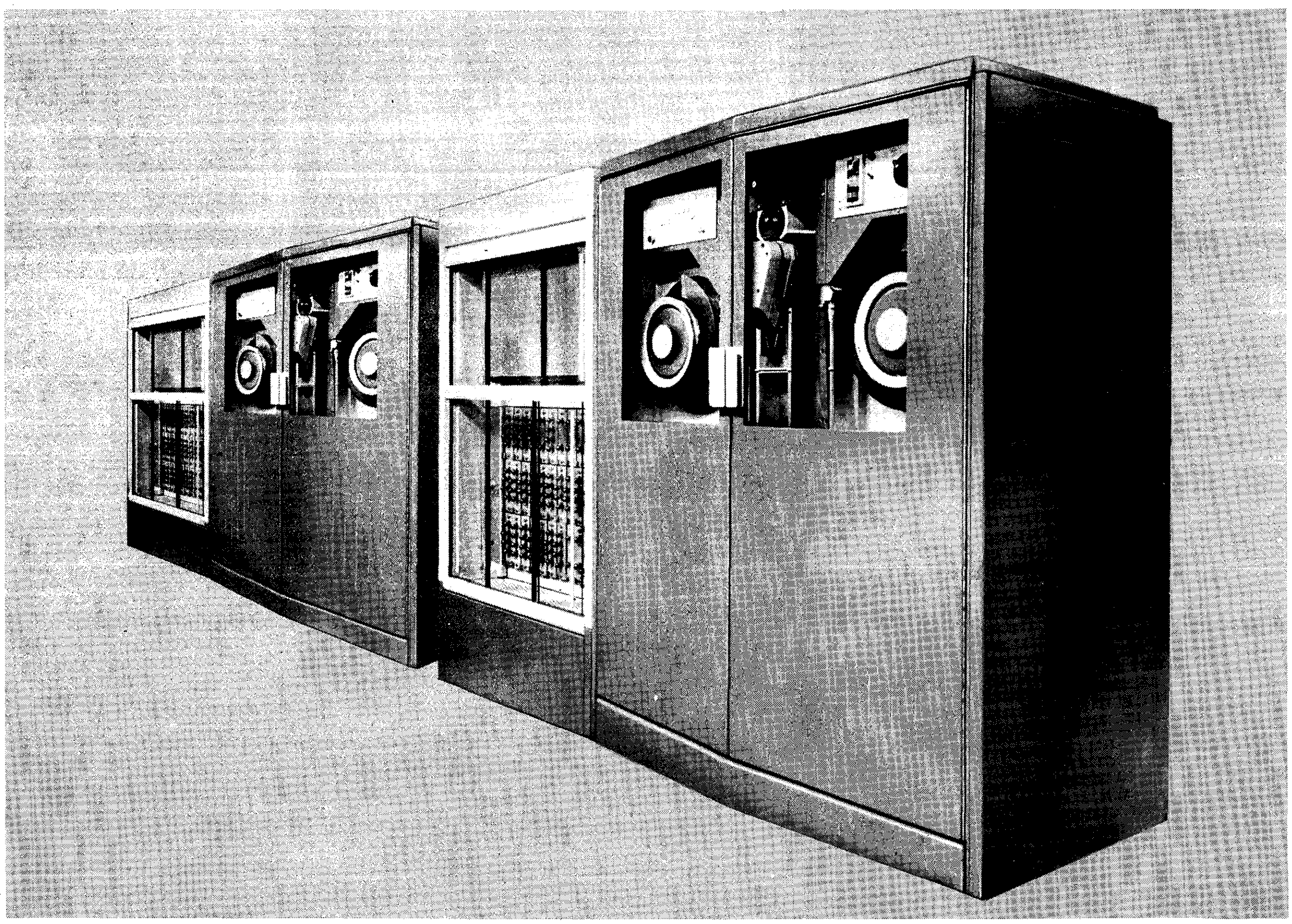
Magnetic Tape Storage

The basic medium for the storage of information in the DATAmatic 1000 is magnetic tape. The particular tape used, the method of recording information on it, and the tape-handling equipment have all been designed or selected to be mutually compatible and to provide high capacity, ease and speed of access to information, ultra-reliable storage and recovery of information, and maximum utilization of space on the tape.

Type VTR-179 magnetic tape has been selected for the DATAmatic 1000 because of its reliability and long life. This tape consists of a layer of iron oxide bonded to a tough, durable Mylar plastic base. A reel of tape is three inches wide and 2700 feet long and can store over 37,000,000 decimal digits of information, the equivalent of data which would require 465,000 punched cards.

Stored information is recorded on the magnetic tape in groups of magnetized spots. The length rather than the strength of these spots is used to form a dot-dash code representing the encoded digits, letters, and symbols. This, the first of a series of unique reliability features, assures that variations in the recorded signal strength will not result in errors.

The information is stored in standard quantities called blocks, which are arranged in a novel fashion along the tape. The virtual elimination of dead space and the optimum packing of information into the tape area is achieved by regarding the tape as a series of areas one block in length, then recording in every other



DATAmatic 1000 Magnetic File Units

block while the tape travels in one direction and in the blocks omitted while the tape travels in the reverse direction. The blocks not filled in a given direction of travel provide the space for starting and stopping the tape in that direction. As a result, information is recorded on almost the entire area of the tape. Moreover, since the reversal of tape direction is accomplished automatically, all of this information is written or read sequentially and the tape is positioned at its physical beginning at the conclusion of this process.

Information is recorded lengthwise to the tape in 31 channels, a system which greatly speeds the transfer of information and facilitates searching processes. Specifically, as many as ten tapes may be searched simultaneously, which means that the system is actually passing over 600,000 decimal digits per second while seeking the particular item desired. The read-record head will write on the tape at the sustained rate of 60,000 decimal digits per second and will recover this information at the same rate. The reading or searching operations may be performed with the tape travelling either forward or backward.

The tape-drive mechanism and the read-record head are contained in the Magnetic File Unit. An installation may include from four to one hundred Magnetic File Units, all directly connected into the system. They may be divided in any manner and at any time between the reading and recording operations. The volume of transactions and the complexity of operations govern the number of Magnetic File Units required for a given system. Furthermore, these units may be added to or removed from the system at any time as these requirements vary.

In order that any Magnetic File Unit may be interrogated and information recovered from it without interrupting the operation of the Central Processor, a File Reference Unit is available. Thus a Magnetic File Unit may, at different times, be recording data received from the Input Converter, reading data to the Output Converter, recording data from the Central Processor, reading data to the Central Processor, or reading data to a File Reference Unit. Also available is a File Switching Unit which increases the flexibility with which Magnetic File Units may be arranged into the various functional groups.

Input Section

Data enters the DATAmatic 1000 on standard 80-column punched cards which are initially read by the Card Reader. In this unit the card is read twice, the two readings are compared, and the card is stacked. The input system will produce a signal to indicate any card whose two readings are not identical. The Card Reader holds batches of over 3000 cards at one time and passes them at the rate of 900 fully punched cards per minute.

The information which is read from the punched cards is translated into the language of the system and arranged in the format of the magnetic tape by the Input Converter. In this process, it passes through two control panels and two temporary storage locations, providing great flexibility for transposition, duplication, and discarding of information. The operator manually sets an identifying control number into the Input Converter, which includes this number in the information to be written on magnetic tape. The control number may then be written on the batch of cards which it represents, in case it is desired later to cross-reference these cards with their corresponding tape.

The encoded information is first arranged in a 100-column format within the converter. In this conversion, any number of card columns may be duplicated provided that the total number of columns does not exceed 100. Triplication of columns is not permitted. Thirteen conversion rules are available for the translation of punch code into machine code. Any single card column may be translated by any one of these thirteen rules. The information is then translated into the final tape format, the contents of two punched cards being fed to each block on the magnetic tape. Several automatic checking features are built in to detect improperly punched cards or errors either in reading or in one of the conversion steps. The operator controls the settings of a group of panel switches which direct the course of action that the machine is to follow in each of these situations.

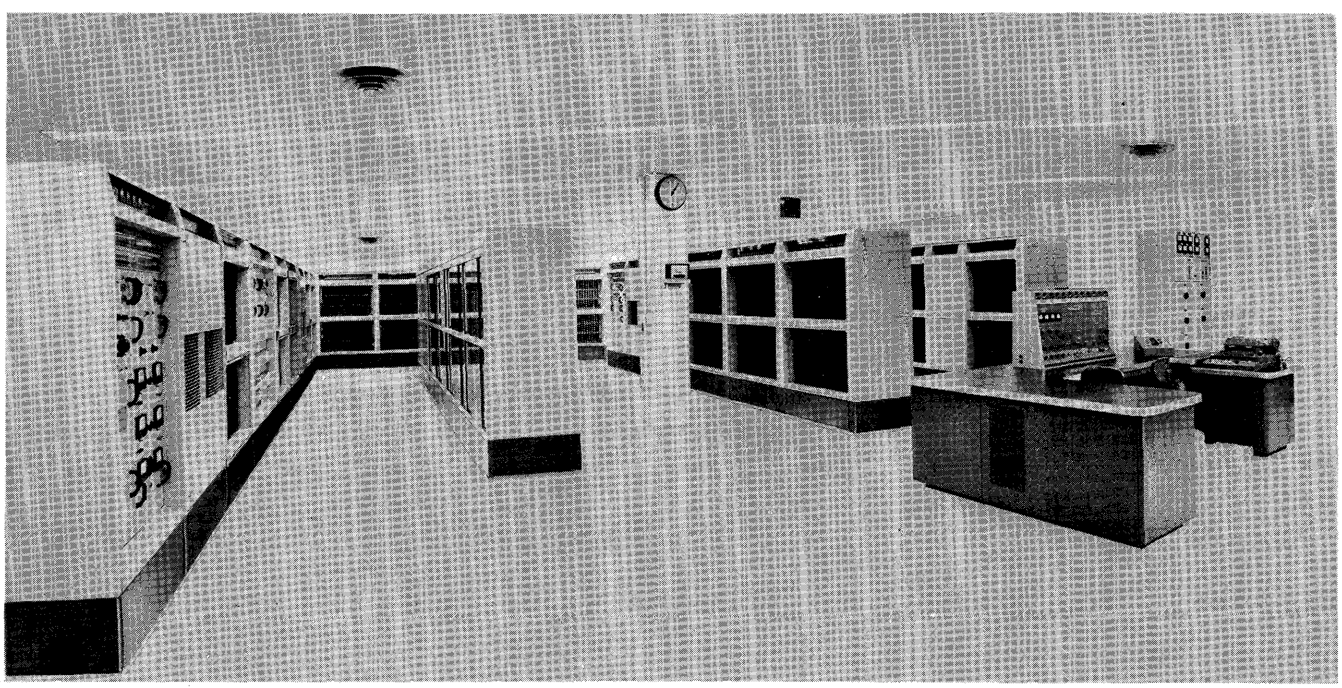
It must be emphasized that the operation of the Input Converter is strictly "off-line". That is, it proceeds entirely independently of and simultaneously with the data-processing and/or output functions. Normally, one or more specific Magnetic File Units are assigned the function of writing on tape all raw data received from input and communicating it to the Central Processor.

Binary Notation

Information which is manipulated, stored, or communicated other than by electronic systems is generally written using 10 decimal digits, 26 alphabetic characters, and a number of punctuation marks and other special symbols. Basic to the adaptation of information to electronic systems is the fact that such information can be written entirely in terms of two symbols, generally called zero and one. This presentation is called binary notation and is analogous to the presentation of information in the more familiar Morse code, in which the two symbols used are called dots and dashes. The symbols used in a binary notation are called binary digits or bits. For example, the ten familiar decimal digits, 0 through 9, are represented in binary notation as follows:

$\overline{0000}$	- 0	$\overline{0101}$	- 5
$\overline{0001}$	- 1	$\overline{0110}$	- 6
$\overline{0010}$	- 2	$\overline{0111}$	- 7
$\overline{0011}$	- 3	$\overline{1000}$	- 8
$\overline{0100}$	- 4	$\overline{1001}$	- 9

Bars will be placed over binary digits to distinguish them from decimal zeros and ones.



DATA m a t i c 1000 Central Processor Section

The storage of information by electronic equipment depends upon the ability to distinguish between two states which represent the two symbols used in binary codes. There are many electronic devices which can make such a distinction. An example of such a device which is both fast and reliable is the tiny, ring-shaped magnetic core. This core may be magnetized in either of two senses; in one sense it is considered to be storing a binary zero and in the other sense a binary one. These tiny magnetic cores constitute the principal element for the storage of information in the High-Speed Memory and buffer storage units of the DATAmatic 1000 Central Processor. In a group of four such magnetic cores, ten of the sixteen possible combinations of states may be used to represent the ten decimal digits.

Central Processing Section

The Central Processor has already been defined to include the Arithmetic and Control Units, the Input and Output Buffer Storage Units, the High-Speed Memory, the Magnetic File Units, and the Central Console. Together, these units contain the electronic elements and circuitry for high-speed performance of the stored programs.

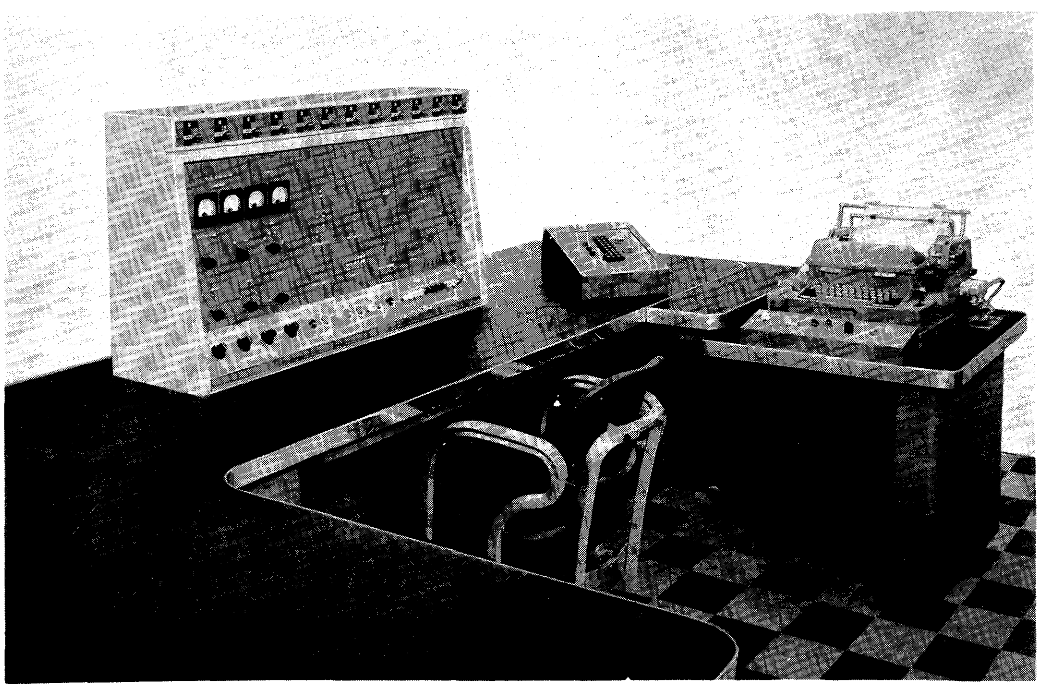
The fast and reliable internal memory is composed of over 100,000 magnetic cores and has a capacity of 24,000 decimal digits. Access is in parallel for rapid readout of stored information. Processing of data stored on magnetic tape is also enhanced by the inclusion of two Input and two Output Buffer Storage Units. These buffers, which are each capable of storing 744 decimal digits, permit a steady flow of information to and from memory and enable the memory to read from one tape and write on another simultaneously.

The design of the Central Processor and the provision of certain special instructions are specifically aimed at the attainment of high sorting, merging, and file-maintenance speeds. Some examples of the speeds achieved are:

Sort - 60,000 decimal digits per second (equivalent to 750 fully punched cards per second).
Merge - 60,000 decimal digits per second.
File Maintenance - 600,000 decimal digits per second.

Arithmetic instructions are carried out by the Arithmetic Unit. The sequence of performance of the stored instructions is directed by the Control Unit. The Central Console is the means of human communication with, and control over, the system. It affords active human control over starting and stopping the machine and passive communication in displaying a continuous picture of the status of the DATAmatic 1000 as it processes a program instruction by instruction. The latter property is an exceptionally useful diagnostic tool for program debugging. The Central Console also mounts a special automatic typewriter which is used for the manual insertion of data to the machine and for the interrogation of the machine. The components of the system can be checked out by running the marginal check program. The Console displays the results which indicate whether any package in the system is approaching an unsatisfactory level.

A fundamental reliability feature of the system is the fact that each basic unit of information includes a check digit called the weight count. This weight count is recomputed after each transfer of information within the Central Processor. The arithmetic comparison of



DATAmatic 1000 Central Console showing simplicity of layout achieved through functional design the original and the recomputed weight counts is an extremely positive and economical means of verifying all internal information transfers, plus arithmetic and logical operations.

Output Section

The Output Section, like the Input Section of the DATAmatic 1000, operates entirely "off-line". It accomplishes the conversion of information stored on magnetic tape into the form of punched cards or printed copy. Two alternative output sections are available which may be purchased either singly or together, depending on the quantity and speed requirements of the application. These are the Model 1300 Output Converter, which provides the required output to drive a standard card punch and/or a standard 150-line-a-minute printer, and the Model 1400 Output Printing System which includes a special DATAmatic High-Speed Printer capable of printing 900 lines per minute. As is the case in the Input Section, one or more Magnetic File Units are normally assigned to communicate between the Central Processor and the Output Converter.

MODEL 1300: The Model 1300 Output Converter reduces data stored on magnetic tape to a form acceptable to a standard 150-line-per-minute tabulator and/or a standard 100-card-per-minute card punch. The tabulator and card punch functions, governed by standard control panels, are preserved.

Information is read from magnetic tape to the converter in quantities of up to 192 decimal digits, 128 alphabetic characters, or equivalent. Each of these sets of data is processed individually and becomes the basis of one line of printed output and/or one 80-column punched card.

The data is then read into converter output storage through a code translator, controlled by a conversion control panel. There are 14 rules for the translation of machine language into standard punch-card codes.

The output storage section simulates 120 columns of punch-card data, in which form the information leaves the converter. Format arrangement and all other standard printout functions are governed in normal fashion by the control panels associated with the readout equipment. In the case of the card punch, the data is converted into the standard 80-column format and transposition and duplication of columns are effected, as desired, by proper wiring of the card punch control panel.

MODEL 1400: The Model 1400 Output Printing System operates from a completely flexible tape format and performs a considerable amount of editing and format arrangement while preparing information to be printed at the rate of nine hundred 120-character lines per minute. In fact, the most complicated printed formats are obtained with a minimum amount of pre-editing required in the Central Processor. Special symbols and legend material can be emitted. Also the printing of certain parts of the form may be suppressed, dependent upon the content of other data within the particular record. Furthermore, the same output tape may be used for several different types of printing runs by wiring and using all of the control panels in the equipment. The sequence of information on magnetic tape need not have any relation to the sequence of printing of information within a given line. It is, moreover, possible to scan a record on the tape several times, on each occasion deriving different combinations of data to be printed on a given form; data from the tape may be rejected or printed several times at will.

From the moment that information is read from the magnetic tape to the actual printing process, a complete train of information monitoring exists to preclude the possibility of printing erroneous information. This system includes a read-back signal from the actual printing hammer to the original stored information to verify the correctness of the character being printed in every column of the form.

The High-Speed Output Converter reads information from magnetic tape in discrete quantities of up to 192 decimal digits. These quantities may be read from any part of the block and are handled as separate units of information throughout the conversion process. Three control panels are used to select the input information, translate it into the language of the High-Speed Printer, edit and position it, and store it in the 120-position print storage. There are 160 printing positions available on the High-Speed Printer, of which any 120 may be used during a given run. Two additional control panels are used to select the particular 120 print positions to be used and to perform further editing.

Integrated Checking

The weight count feature of the DATAmatic 1000, previously described, is an integrated checking system which verifies every information transfer, arithmetic and logical operation from the original conversion to machine language through the final production of printed or punched output. The weight count digit is originally computed and checked during the input conversion process. It is then recorded on tape, one such digit being an integral part of each basic unit of information and remaining with this basic unit throughout all of the operations of the system. Thereafter, recomputation and checking of weight count verifies every transfer of information from tape to the Central Processor, and all internal operations within the Central Processor, transfers from the Central Processor to tape, transfers from tape to the Output Converter, and Output Converter decoding processes. Each of these checking sequences is integrated with the preceding and following sequence to form a single, system-wide verification of accuracy. The weight count system is augmented in various DATAmatic 1000 units with duplicate circuitry and other special circuits which further extend the checking system.

SECTION I

CENTRAL PROCESSOR

SECTION I - CENTRAL PROCESSOR

THE DATA m a t i c WORD

The fundamental parcel of information which the DATA m a t i c 1000 can manipulate is called a word. It is an assemblage of 52 bits which can have different interpretations depending upon the use of the word in the DATA m a t i c 1000. The three basic types of words are numeric, alphanumeric, and instruction, each of which has a prescribed bit structure. Moreover, the bits of a word may be arbitrarily grouped to serve a specific function.

NUMERIC: In a numeric word, the 52 bits are grouped into thirteen 4-bit sets called digits. The leftmost digit contains a code which represents the sign of the number; 1101 equals plus (+) and 0101 equals minus (−). The rightmost digit performs a special checking function, to be described shortly. The eleven remaining digits are interpreted as eleven decimal digits. Thus the number

+ 78901234567

is represented in the system as shown in Figure 1-1.

In order to perform most arithmetic operations on numeric words in the system, it is necessary that these words be signed as indicated above. Frequently, a numeric word contains reference information only and is not normally included in arithmetic operations. Under these conditions, the leftmost digit does not have to be a sign and can contain another decimal digit. Words of this type are called unsigned numeric, as distinguished from the more common signed numeric words.

A group of four bits can have 16 distinct configurations, and therefore can be used as a code having 16 different values. When a group of four bits has one of 16 different values, it is called a hexadecimal digit. (The term hexadecimal is frequently abbreviated as hex.) Numeric words are interpreted as eleven (signed) or twelve (unsigned) digits where each digit is represented by four bits. Therefore, a numeric word could also specify eleven (signed) or twelve (unsigned) hexadecimal digits. The external symbols which are used to express hexadecimal digits include the normal decimal digits plus an arbitrary selection of values to represent the hex numbers between 10 and 15. The complete table of hexadecimal codes is given in Table 1-1.

The rightmost four bits are called the Weight Count and represent a decimal digit used to check the correctness of the rest of the word. It is computed by adding the numeric values of the first twelve hexa-

Bit Code	Normal	Special Symbols	Numeric Value ε
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	8	8	8
1001	9	9	9
1010	B	$	⑩
1011	C	,	⑪
1100	D	.	⑫
1101	E	+	⑬
1110	F	(space)	⑭
1111	G	*	⑮

ε The circle is placed around 10 in this column in order to stress the fact that ⑩ corresponds to the four bits, 1010, rather than the eight bits 0001 0000. Similarly for ⑪, etc.

Table 1-1

Bit Position	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29
Binary Digit	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1

Bit Position	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Binary Digit	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	X	X	X	X

Figure 1-1

decimal digits and reducing the sum by subtracting enough nines to obtain a digit from 1 to 9. Figure 1-2 shows some examples.

Word	Sum	Reduced Sum or Weight Count
9 1 6 2 7 3 8 1 9 8 3 3	60	6
+ 1 6 2 7 3 8 1 9 8 3 3	64	1
9 B C D E F G 0 1 2 3 4	94	4

Figure 1-2

In the case where the first twelve hexadecimal digits are all zeros, the weight count will be nine. It is important to note that every word, numeric, alphanumeric, or instruction, is treated as numeric for the computation of its weight count. The weight count is computed by the Input Converter and attached to the word before recording on magnetic tape. There is no requirement for the manual computation of weight count. Whenever the word is moved from one part of the system to another, the weight count is recomputed and checked. As soon as an incorrect weight count is detected, the machine will stop immediately to permit corrective action.

ALPHANUMERIC: In an alphanumeric word the 52 bits are grouped into eight 6-bit groups and one 4-bit group. The 4-bit group comprises the four rightmost bits and, as in the case of the numeric word, is used to specify the weight count of the word. The eight 6-bit groups are called characters and contain codes which are interpreted as alphanumeric characters by the associated DATA m a t i c 1000 equipment. There are 64 possible configurations of the six bits in a code. All 64 configurations and the corresponding alphanumeric symbols are listed in Table 1-2.

The codes which are specified as unassigned have no interpretation. However, these unassigned codes should not be substituted for space codes in output data because they may stop the operation of the peripheral equipment. As an example of alphanumeric binary coding, the name John H. Smithers would be stored internally as shown in Figure 1-3.

INSTRUCTIONS: Words can be interpreted as instructions by the system, in which case the word directs the machine to perform a certain operation involving other words to be found at specified locations. Two hexadecimal digits are used to specify the internal operation which is to be performed. The rightmost four-bit group, as in all words, specifies the weight count of the word. The remaining digits are used to specify the location of words or to specify numbers required to perform the instruction. Prior to any further discussion of the interpretation of a word as an instruction, it is necessary to define registers and addresses.

CONSTANTS: A DATA m a t i c word whose contents remain inviolate throughout a program or a specified portion of a program is called a constant. A constant may be numeric, alphanumeric, or an instruction.

000000	0 (Zero)	100000	- (Hyphen or Minus)
000001	1	100001	J
000010	2	100010	K
000011	3	100011	L
000100	4	100100	M
000101	5	100101	N
000110	6	100110	O (letter)
000111	7	100111	P
001000	8	101000	Q
001001	9	101001	R
001010	'	101010	=
001011	#	101011	$
001100	@	101100	*
001101	+	101101	¢
001110	(Unassigned)	101110	(Unassigned)
001111	(Unassigned)	101111	(Unassigned)
010000	(Space)	110000	&
010001	A	110001	/
010010	B	110010	S
010011	C	110011	T
010100	D	110100	U
010101	E	110101	V
010110	F	110110	W
010111	G	110111	X
011000	H	111000	Y
011001	I	111001	Z
011010	Cr.	111010	:
011011	. (Decimal Point or Period)	111011	,
011100	□	111100	%
011101	(	111101	1/2
011110	)	111110	(Unassigned)
011111	(Unassigned)	111111	(Unassigned)

Table 1-2. Alphanumeric Binary Code

Alphanumeric Code	$\overline{100001}$	$\overline{100110}$	$\overline{011000}$	$\overline{100101}$	$\overline{010000}$	$\overline{011000}$	$\overline{011011}$	$\overline{010000}$	XXXX
Alphanumeric Symbol	J	O	H	N	(space)	H	.	(space)	
Alphanumeric Code	$\overline{110010}$	$\overline{100100}$	$\overline{011001}$	$\overline{110011}$	$\overline{011000}$	$\overline{010101}$	$\overline{101001}$	$\overline{110010}$	XXXX
Alphanumeric Symbol	S	M	I	T	H	E	R	S	

Figure 1-3

REGISTERS AND ADDRESSES

Registers: The DATAmatic 1000 word requires a set of 52 elementary storage devices. Such a set of storage devices is called a register. In the process of executing a typical instruction, information is taken from selected registers, manipulated, and stored in other registers. In order to select the proper words for each instruction, each register in the High-Speed Memory must have a unique identification, called an address. The internal High-Speed Memory consists of two banks of 1000 registers each, designated as bank 0 and bank 1.

Addresses: Within each bank, the registers are assigned decimal numbers from 000 to 999, called subaddresses. An address consists of a bank designator, $\bar{0}$ or $\bar{1}$, followed by a subaddress. For example, $\bar{1}100$ is the address of the word in register 100 of bank 1. Note that the bank designator is specified by a single bit, so that the complete address requires only 13 bits in an instruction.

Registers between $\bar{1}990$ and $\bar{1}999$ have special uses and are called special function registers. Of this group of registers, $\bar{1}991$, $\bar{1}996$, and $\bar{1}998$ do not exist. Special use is also made of registers $\bar{1}981$ to $\bar{1}989$. A bar has been placed over the first digit of each address to remind the reader that this digit represents one bit rather than four. When there is no danger of confusion, this bar will be omitted.

When a word is specified as an instruction, it has the format shown in Figure 1-4. The hexadecimal code which selects the internal operations is specified in bit positions 48 to 41. The A address is composed of the bank designator, bit 51, and the subaddress, bits 40 to 29. Similarly, the B address comprises Bits 50 and 28 to 17 and the C address bits 49 and 16 to 5. The space in an instruction which is usually reserved for an address is sometimes used for other purposes. Examples will be discussed in the detailed discussion of the DATAmatic 1000 instructions. Bit 52 is interpreted as the sign of an instruction and is used when an instruction is to be modified by addition or subtraction.

Bits	52	51-49	48-41	40-29	28-17	16-5	4-1
Function	Sign	A B C	Operation Code	A Subaddress	B Subaddress	C Subaddress	Weight Count
		bank designators					

Figure 1-4

SEQUENCE REGISTER AND ORDER OF EXECUTING INSTRUCTIONS

Before proceeding to describe all the instructions in detail, it is important to know how an instruction is executed and how the Central Processor knows which instructions to perform and in what sequence. There are two registers which are important in this regard. One is the Control Register (1990); the other is a special register (containing one address) which is called the Sequence Register. We start with the instruction to be performed in the Control Register. The process of executing the instruction includes:

(1) analyzing it to determine its type
(2) performing it
(3) selecting the next instruction and placing it in the Control Register.

In regard to the last step, instructions fall into two classes: "normal" instructions and "subsequence call" instructions. For a normal instruction, step (3) consists of putting into the Control Register the instruction whose address is in the Sequence Register and then adding "one" to the address in the Sequence Register. Thus a succession of normal instructions would be performed from consecutive registers. For a subsequence call instruction, step (3) consists of putting the instruction in the C address into the Control Register, ignoring the Sequence Register completely. Thus, a series of subsequence call instructions can be taken from a set of randomly chosen registers in the internal memory and the address in the Sequence Register is left unmodified during the execution of these subsequence call instructions. The execution of any normal instruction automatically leads to the use of the Sequence Register for the selection of the next instruction. The ability to

$D_sD_aD_bD_c$	Operation Code	A Subaddress	B Subaddress	C Subaddress	Weight Count
$\bar{1}\ \bar{0}\ \bar{0}\ \bar{0}$	$\overline{0000}\ \overline{0000}$	$\overline{0000}\ \overline{0000}\ \overline{0000}$	$\overline{0000}\ \overline{0001}\ \overline{0001}$	$\overline{0000}\ \overline{0010}\ \overline{0000}$	XXXX

Figure 1-5

perform a sequence of instructions in the subsequence mode and then to return to the main thread of programming via the Sequence Register without any additional instructions required to direct this operation is a unique feature of the DATAmatic 1000. In some cases, it leads to a significant reduction in the number of instructions and to considerable increase in speed.

CHANGE OF SEQUENCE: The function of certain instructions is to change the contents of the Sequence Register. Such instructions make it possible to perform instructions in any sequence desired by the programmer who prepares the program. Thus, in summary, normal instructions are performed in sequence unless one of them changes the contents of the Sequence Register in some fashion other than to add "one" to it. Subsequence call instructions are performed in the sequence determined by the C address of the instruction. The subsequence call instructions will be defined as each instruction is discussed. The normal instructions are all other instructions plus subsequence call instructions whose C address is zero. Certain instructions (e.g. ADD) result in a subsequence call to a special address under certain circumstances. These will be described in detail later.

At this point it would be well to define the Sequence Change and Subsequence Call instruction to illustrate the above remarks about the sequence of performing instructions. This instruction is designated SCS. Its operation code is 00 (appearing in bits 48-41 as $\overline{0000}\ \overline{0000}$). The A address is not used in this instruction. Prior to step (3), the B address is put in the Sequence Register unless it is zero, in which case it is ignored. If C is not zero, the instruction is a subsequence call and C is the address of the next instruction. If C is zero, the instruction is normal and the Sequence Register contains the address of the next instruction. For example, in the instruction shown in Figure 1-5, the A address is $\bar{0}\ \overline{0000}\ \overline{0000}\ \overline{0000}$ or 0000, which is written as 0. The B address is $\bar{0}\ \overline{0000}\ \overline{0001}\ \overline{0001}$ or 0011, which is written as 11. The C address is $\bar{0}\ \overline{0000}\ \overline{0010}\ \overline{0000}$ or 0020, which is written as 20. The operation code indicates that the instruction is SCS. The instruction can then be written in hexadecimal form:

$D_sD_aD_bD_c$	Operation Code	A Sub-address	B Sub-address	C Sub-address
8	00	000	011	020

It is simpler yet to write it: SCS/0/11/20

This is a shorthand notation to indicate the type of instruction and its addresses which are SCS, 0, 11, and 20, respectively. After a few more examples, this kind of notation will be used consistently without referring to the binary form of the word. The instruction SCS/0/11/20 causes the Central Processor to put 11 (the B address) into the Sequence Register and then to get its next instruction from register 20 (the C address).

Figure 1-6 illustrates the sequence of performing a series of SCS instructions. S. R. stands for the Sequence Register and C. R. for the Control Register. It is assumed that initially zero is in the Control Register, one is in the Sequence Register, and registers 1 through 6 contain the following instructions:

Register	Instruction
1	SCS/0/3/0
2	SCS/0/0/0
3	SCS/0/0/2
4	SCS/0/0/6
5	SCS/0/7/3
6	SCS/0/0/2

First instruction, SCS/0/0/0: The instruction is recognized to be type SCS in step 1. In step 2, the Sequence Register is not altered since the B address of the instruction is zero. In step 3, since this instruction is normal (its C address is zero), the next instruction is selected from register 1 (the address in the Sequence Register) and one is added to the address in the Sequence Register.

Second instruction, SCS/0/3/0: In step 1, the instruction is recognized to be type SCS. In step 2, B is put into the Sequence Register since B is not zero. In step 3, since this instruction is normal, the next instruction is selected from register 3 (the address in the Sequence Register) and one is added to the address in the Sequence Register.

The remaining instructions are self-explanatory. Notice that the instruction SCS/0/0/2 is performed three times. Each time it leads to the instruction in register two. However, each time the instruction in register two (SCS/0/0/0) is performed, there is a different address in the Sequence Register. Consequently, the next instruction after SCS/0/0/0 is different every time. The usefulness of this ability will be demonstrated later.

Instruction	Address	Current Instruction in Control Register at start	Contents of Sequence Register at start	Set up Sequence Register		Put next instruction in Control Register		Make final adjustment to Sequence Register	
				If B $\neq$ 0, put B in S. R.	If B = 0, let S. R. stand	If C $\neq$ 0, put contents of C in C. R.	If C = 0, put contents of S. R. address in C. R.	If current instruction is normal, Add 1 to S. R.	If current instruction is a subsequence call, let S. R. stand.
First	1990	SCS/0/0/0	1		1		SCS/0/3/0	2	
Second	1	SCS/0/3/0	2	3			SCS/0/0/2	4	
Third	3	SCS/0/0/2	4		4	SCS/0/0/0			4
Fourth	2	SCS/0/0/0	4		4		SCS/0/0/6	5	
Fifth	4	SCS/0/0/6	5		5	SCS/0/0/2			5
Sixth	6	SCS/0/0/2	5		5	SCS/0/0/0			5
Seventh	2	SCS/0/0/0	5		5		SCS/0/7/3	6	
Eighth	5	SCS/0/7/3	6	7		SCS/0/0/2			7
Ninth	3	SCS/0/0/2	7		7	SCS/0/0/0			7
Tenth	2	SCS/0/0/0	7		7		Instruction from 7	8	
Eleventh	7	Instruction from 7	8						

Figure 1-6

PROGRAMMING: The process that the programmer goes through to prepare a problem for solution consists of writing the necessary instructions to execute, step by step, the operations required. These instructions are then placed in the internal memory and the DATAmatic 1000 proceeds to execute the instructions in the sequence specified by the programmer. In order to do this work successfully, the programmer must be well aware of the distinction between normal instructions and subsequence call instructions. The use of normal and subsequence call instructions is illustrated further in a later section.

At this point, it is worth considering how a word is distinguished as an instruction, a number, or an alphanumeric. The operator determines which instruction will be first when he starts the system. After the computer has started, each instruction, in conjunction with the Sequence Register, determines which instruction will be next. Thus, one by one, the words which are to be interpreted as instructions are brought into the Control Register, analyzed, and executed. If by mistake the programmer (or machine) has put a number or a name into a register where an instruction should be, the system will put this number or name into the Control Register. The system may discover that the word cannot be an instruction and halt with an error indication. It is conceivable that a number could be interpreted as a legitimate instruction, but the programmer takes precaution to avoid this situation. In preparing a problem, he also needs certain numbers which are placed in the memory at desired locations by a series of appropriate instructions. The process of determining the appropriate instructions is called programming. In brief, then, it is the programmer who determines which words are instructions, numbers, or alphanumerics.

CLASSIFICATION OF INSTRUCTIONS

The instructions may be conveniently classified into six functional groups: Arithmetic Instructions, Tape and Transfer Instructions, Decision Instructions, Sorting Instructions, Control Instructions, and Print Instructions. In the detailed discussion of these groups, it must be realized that the address of a register is, of course, not the same as the word in the register. In order to avoid the confusion which sometimes arises, the symbol (A) will be used when referring to the word in the register at address A. This symbol represents the contents of A or the word in A.

ARITHMETIC INSTRUCTIONS

ADD: Add is typical of the arithmetic instructions. Its operation code is FF or $\overline{1110}$ $\overline{1110}$. This instruction is written ADD/A/B/C and it is read: "Add (A) to (B) and put the answer in C." That is, add the number in A to the number in B and store the sum in C. If the sum of the numbers at A and B is too large to store in a single register, there is said to be an overflow. In this case, the high-order digit of the sum is lost and there is an automatic subsequence call to 1988, the Accumulator Overflow Register. The programmer will presumably have stored in 1988 the first instruction of a special action to take care of this situation. The Add instruction does not disturb the words at A and B. In fact, it is a general rule that the machine does not alter the contents of a register just to use them.

Previously, it was noted that any string of 48 bits with four extra check bits is called a word and that groups of four of these bits can be used to represent decimal digits between 0 and 9 inclusive. It was also noted that a group of four bits may also represent six other hexadecimal quantities which have been arbitrarily designated as B, C, D, E, F, and G. These obviously are not decimal digits. The arithmetic instructions were designed basically to perform decimal arithmetic. This means that they will perform proper arithmetic operations on decimal numbers with valid signs ($\overline{1101}$ for plus and $\overline{0101}$ for minus) in digit positions 52 through 49. It has further been noted that certain words in the machine, instructions in particular, will contain hexadecimal digits which are not proper decimal digits. Unless he wishes to modify an instruction, the user will not normally attempt to add two words which are not honest decimal numbers. If he does attempt to add two words which are not legitimate decimal numbers, the machine may stop and indicate an error.

INSTRUCTION MODIFICATION: Instructions usually have illegitimate signs since the first four bits are sign bit and bank designators and could be any hexadecimal character. The two characters of the operation code can be hexadecimal digits. Furthermore, the subaddress positions of certain instructions can contain non-decimal characters. Since the DATAmatic 1000 is basically a decimal machine with decimal arithmetic, it has been necessary to make special provision for modifying instructions by addition and subtraction and these special provisions include certain restrictions which the programmer must understand.

(1) The instruction must be in address A and the number to be added in address B; otherwise the machine may stop.

(2) If a hexadecimal digit in the word at address A is added to any digit, decimal or hexadecimal, from the word at address B, the sum must not exceed hex G; otherwise the machine will stop.

(3) Addresses greater than 999 cannot be obtained (because of the position of the bank designator) by adding to addresses less than 1000.

With these three restrictions in mind, the programmer will have no difficulty modifying instructions as desired.

EXAMPLE: As an example of the use of the Add instruction, assume that in registers 1 through 8 the words shown in Figure 1-7 (in binary form) have been placed. When it does not matter what is in certain words, a dash instead of a digit is used. D_s is the sign bit for instructions. D_a, D_b, and D_c are the bank designators for addresses A, B, and C, respectively. The first three

Register	CONTENTS OF REGISTER					
	$D_s D_a D_b D_c$	Instruction Type	A Subaddress	B Subaddress	C Subaddress	Weight Count
	52-49	48-41	40-29	28-17	16-5	4 - 1
1	1101	0000 0000	0000 0000 0000	0000 0000 0000	0000 0001 0010	----
2	1101	0000 0000	0000 0000 0000	0000 0000 0000	0000 0000 1001	----
3	0101	0000 0000	0000 0000 0000	0000 0000 0000	0000 0001 0011	----
4	----	---- ----	---- ---- ----	---- ---- ----	---- ---- ----	----
5	----	---- ----	---- ---- ----	---- ---- ----	---- ---- ----	----
6	1000	1110 1110	0000 0000 0001	0000 0000 0010	0000 0000 0100	----
7	1000	1110 1110	0000 0000 0001	0000 0000 0011	0000 0000 0101	----
8	1000	1110 1110	0000 0000 0100	0000 0000 0101	0000 0000 0101	----

Figure 1-7

Register	Contents of Register				
	*	Instruction Type	A Sub-address	B Sub-address	C Sub-address
1	+	0 0	0 0 0	0 0 0	0 1 2
2	+	0 0	0 0 0	0 0 0	0 0 9
3	-	0 0	0 0 0	0 0 0	0 1 3
4	-	- -	- - -	- - -	- - -
5	-	- -	- - -	- - -	- - -
6	8	F F	0 0 1	0 0 2	0 0 4
7	8	F F	0 0 1	0 0 3	0 0 5
8	8	F F	0 0 4	0 0 5	0 0 5

* This column represents the sign if the word is a number and it represents the sign bit and bank designators if the word is an instruction. It represents unspecified bits in words 4 and 5 .

Figure 1-8

words are numbers and therefore their bits 52-49 are signs; the last three words are instructions which manipulate these numbers. The headings D_s, D_a, D_b, D_c, Instruction Type, etc., refer only to the instructions.

These instructions are shown in the more compact hexadecimal form in Figure 1-8. Even more compactly, +12 could be written as a shorthand notation for the word in register 1. Furthermore, as mentioned above it will be convenient and helpful to the reader's memory to write ADD instead of FF and 1 instead of 0001 for the A address of the instruction in register 6, etc. Thus Figure 1-8 could be further condensed to the form of Figure 1-9.

If 6 (i.e. 0006) is put in the Sequence Register and the computer is started, it will execute the instructions in registers 6, 7, and 8 successively, and then go on to whatever instruction is contained in register 9. As a result of performing the instruction in register 6, +12 +9 = +21 will be sent to register 4, replacing whatever may have been there. The contents of registers 1 and 2 are not affected by this instruction. Similarly, the result of the next instruction in register 7 is to put +12 +(−13) = −1 in register 5 and the result of the instruction in register 8 is to put +21 + (−1) = +20 in register 5, replacing the −1 which was there temporarily. Thus these three instructions may be summarized as shown in Figure 1-10.

Register	Contents
1	+12
2	+9
3	-13
4	----
5	----
6	ADD/1/2/4
7	ADD/1/3/5
8	ADD/4/5/5

Figure 1-9

FLOW DIAGRAMS: It is useful, especially in more complicated situations, to summarize the effect of the in-

Register	Instruction (Symbolic Form)	Effect of Instruction
6	ADD/1/2/4	(1) + (2) = +21⟶4
7	ADD/1/3/5	(1) + (3) = -1⟶5
8	ADD/4/5/5	(4) + (5) = +20⟶5

Figure 1-10

structions still more compactly by means of a flow diagram. The flow diagram is a basic programming tool, the usefulness of which will soon become apparent. Other basic elements of programming will be introduced as they are needed. Figure 1-11 is a flow diagram of the example above.

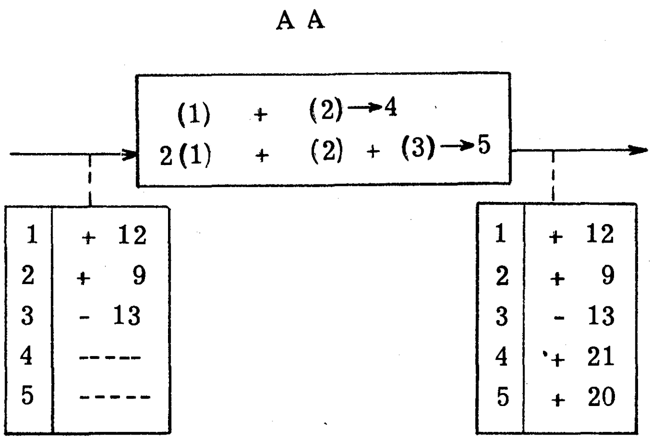


Figure 1-11

The rectangle labelled AA is called an operation box because it summarizes some operations to be performed by the machine. The instructions which it represents will put the sum of the contents of registers 1 and 2 in register 4 and the sum of twice the contents of register 1 plus the contents of registers 2 and 3 in register 5. The rectangles attached by dotted lines to the flow diagrams are called storage tables. They show the pertinent storage of information in certain registers before and after the operation box. Thus, operation box AA did not affect registers 1, 2, and 3, but put +12 + 9 = +21 into register 4, and put +24 + 9 − 13 = +20 into register 5.

Speed: In Figure 1-54 (page 1-40), which summarizes a description of the instructions, is a column labelled "Word Cycles", and under this column opposite "Add" is 8. A word cycle is 28.8 microseconds (millionths of a second). Addition requires eight word cycles or about 230 microseconds. This is the time for the complete operation of obtaining the instruction, obtaining the number from the A address, adding it to the number in the B address, storing the answer in the C address, and changing the contents of the Sequence Register. In the preceding example there are three Add instructions which would require 24 word cycles or about 690 microseconds. The time required for the SCS instruction described earlier is six word cycles, except that it will be reduced one word cycle if the Sequence Register is *not* changed and reduced another word cycle if there *is* a subsequence call. In fact, it is a general timing rule that a change in the Sequence Register adds one word cycle and a subsequence call subtracts one word cycle from the time required to perform any instruction. The instruction times stated in this manual are the so-called "maximum" times for the case in which there is a change in the Sequence Register but no subsequence call.

Decimal Point: When performing addition on a desk computer or in an electronic computer, the decimal point can be in any position provided it is the same for both numbers added. The answer, of course, also has its decimal point in the same place. No bits are used in the number to specify the decimal point. It is the programmer who knows where he wants it and it is his responsibility to place numbers correctly so that their decimal points line up. It is possible to print decimal points where desired in the output process. This will be described in the Output Converter sections (Sections III and IV).

Subtract: The Subtract instruction is written SUB/A/B/C, where SUB stands for the operation code GG; in other words, bits 48-41 will be $\overline{1111}$ $\overline{1111}$. The effect of the Subtract instruction is to put the contents of A minus the contents of B into C.

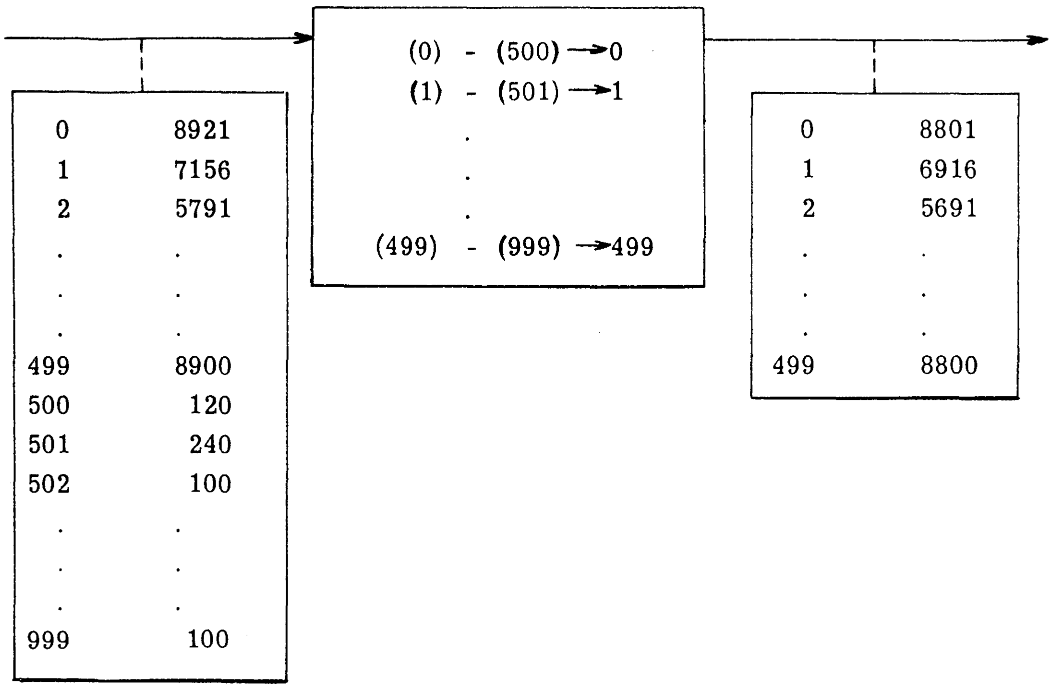
$$(A) - (B) \longrightarrow C$$

The result has the correct sign if the contents of A and B are decimal numbers with either legitimate sign, + or −. As in the case of an Add instruction, an overflow results in the loss of the high-order digit and an automatic subsequence call to 1988. The use of the Subtract instruction for modifying instructions is analogous to the use of the Add instruction. It should be remembered that if bit 52 equals 1, the instruction will be considered positive by the Arithmetic Unit, and if bit 52 equals 0, the instruction will be considered negative. Bit 52 of the instruction will be used with the sign of the number to determine the proper computation to be performed in accordance with standard algebraic rules.

Example: An example of the use of the Subtraction instruction is shown in Figure 1-12. The function of operation box AA is to subtract from the week's pay of each of 500 individuals the amount of their charity deductions in order to obtain their net pay after this deduction. Before entering box AA, the amounts of the week's pay for the 500 individuals are in registers 0 to 499 with the pennies in the far right of the registers, that is, register 0 contains +00 000 008 921, etc. Registers 500 through 999 contain the charity deductions for each individual. Thus, the charity deduction of $1.20 in register 500 corresponds to the gross pay of $89.21 in register 0. The instructions in registers 1000 through 1499 are labelled AA1, AA2, etc., to indicate that they accomplish the operations required by operation box AA. In other words, they replace $89.21 in register 0 by $88.01 in register 0; $71.56 in register 1 by $69.16 in register 1; and so on.

The instructions in hexadecimal form are shown in the fifth column. The first digit of each instruction is 8 because the instructions are all taken positive which

A A



Instruction	Space Assignment	Descriptive Form of Instruction	Effect of Instruction	Instruction in Hexadecimal Form
AA1	1000	SUB/0/500/0	8801 → 0	8 GG 000 500 000
AA2	1001	SUB/1/501/1	6916 → 1	8 GG 001 501 001
AA3	1002	SUB/2/502/2	5691 → 2	8 GG 002 502 002
.	.	.	.	.
.	.	.	.	.
	.	.	.	.
AA499	1499	SUB/499/999/499	8800 → 499	8 GG 499 999 499

Figure 1-12

makes the first bit a one. The three addresses in each of these cases are less than 1000, so that the bits 51, 50, and 49 are zero in all of these examples. GG is the operation code for the Subtract instruction.

It should be noted in this example that 500 instructions have been used to perform 500 subtractions. The Subtract instruction, like the Add instruction, requires eight word cycles, so the time required to make 500 subtractions is 0.12 seconds. If it were always necessary to write down 500 instructions in order to perform 500 operations, the memory capacity of 2000 words would not suffice to perform complicated operations. For this reason, another method which may be used is shown in Figure 1-13.

In Figure 1-13, the effect of the instruction in register 1000 is the same as before, namely $1.20 is subtracted from $89.21; the result, $88.01, is placed in register 0. The effect of the next instruction in register 1001 is more interesting. This instruction causes the word in register 1003 to be added to the instruction in register 1000 and the result is placed in register 1000. The ones are in such a position as to add one to each of the three addresses in this instruction. The operation code GG is not affected by adding zero to it, nor does the plus sign affect the 8. Consequently, register 1000 contains as a result of this instruction the instruction SUB/1/501/1. The next instruction, taken from register 1002, places 1000 in the Sequence Register and is

Instruction	Location	Descriptive Form of Instruction	Effect of Instruction	Instruction in Hexadecimal Form
AA1	1000	(SUB/0/500/0)	8801 ⟶ 0	8 GG 000 500 000
AB1	1001	ADD/1000/1003/1000	8 GG 001 501 001⟶1000	G FF 000 003 000
AB2	1002	SCS/0/1000/0	(return to 1000 for next instruction)	2 00 000 000 000
	1003	+0 1, 1, 1		+ 00 001 001 001

Figure 1-13

a normal instruction since the C address is zero. The next instruction is then taken from register 1000. However, the instruction which is now in register 1000 has been changed since originally written there. Therefore, the effect of performing the instruction in register 1000 is not the same as it was the last time.

To stress the fact that this instruction keeps changing and is not entirely determined by what originally was there, parentheses are put around it. It can now be seen that the second time the instruction in register 1000 is performed, $2.40 is subtracted from $71.56, obtaining $69.16, which will be placed in register 1. Next, the instruction in 1001 will be performed and this time it will again modify the instruction in register 1000. Thus, step by step, the same arithmetic operations are performed, but now only three instructions are used.

The set of operations in boxes AA and AB is called a loop and the DATAmatic 1000 goes through the loop AA and AB 500 times in this example. It should be noted that this is not a practical code at this point, because after going through the loop 500 times, the instruction in 1000 will be SUB/501/0/500 (see Figure 1-14). This illustrates the fact that adding one to address 999 does not yield address 1000. Figure 1-14 shows that when an instruction is modified by addition, overflow ones are carried to the left through all nine subaddress digits (bits 40-5), just as though both words were numeric. For this reason, the programmer must exercise caution in crossing from one memory bank to another.

8 GG 499 999 499
+ 00 001 001 001
8 GG 501 000 500

Figure 1-14

Note that the time required to go through the loop once is 22 word cycles and the time required to go through it 500 times is 0.33 seconds. In other words, it took 2.75 times as long to accomplish the task with three instructions repeated 500 times as it took to do it with 500 instructions performed once. After discussing the Branch and Return instruction, this example will be reexamined and a method introduced to make it practical. The process of preparing a problem for solution by a digital computer, known as programming, involves making economic compromises between various methods of accomplishing the same goal.

MULTIPLY: MUL/A/B/C. The operation code of Multiply is FE. This instruction causes the computer to multiply the number at A by the number at B and put the answer at C.

$$(A) \cdot (B) \longrightarrow C$$

There is no provision for multiplying anything but legitimate decimal numbers with 4-bit signs. The time required for the Multiply instruction is 35 word cycles.

DECIMAL POINT: Just as on a desk computer, the decimal point of each number may be anywhere and the decimal point of the answer must be computed as follows: if the decimal points of (A) and (B) are a and b places, respectively, to the right of the sign, then the decimal point of C is c places to the right of the sign, where $c = a + b$; a, b, and c may be positive or negative. Thus suppose that registers 0 and 1 contain +12 000 000 000 and +13 000 000 000; the instruction MUL/0/1/2 will cause +01 560 000 000 to be put in register 2 (see Figure 1-15). This act may be interpreted in many different ways. For example, if +12 000 000 000 in register 0 stands for +12. 000 000 000 and +13 000 000 000 in 1 stands for +13. 000 000 000, then the number +01 560 000 000 which instruction

Register	Contents
0	+12 000 000 000
1	+13 000 000 000

Instruction	Effect of Instruction
MUL/0/1/2	+01 560 000 000 ⟶ 2

Figure 1-15

MUL/0/1/2 put into register 2 will stand for +01 56.0 000 000. In this case, a = 2, b = 2, c = 4, and the answer has its decimal point four places to the right of the sign. On the other hand, if register 0 contains .12 and register 1 contains .13 then the same instruction puts .0156 in register 2. In this case, a = 0, b = 0, c = 0. If the number in register 0 is interpreted as .0012 and the number in register 1 as .013, then the number put into register 2 by the same instruction should be interpreted as .0000156. In this case, a = −2, b = −1, c = −3; i.e., the decimal point of the answer is three places to the left of the first digit in the register. Thus, the position of the decimal point is arbitrary in both factors but not in the answer. It will frequently be useful to take the point of view that one deals only with numbers having the decimal point at the far left. In this way the results of multiplication would also place the decimal point at the far left.

ROUNDING POSITIVE NUMBERS: If the contents of A and B are 11-digit numbers, the exact product is a 22-digit number. For convenience, the DATAmatic 1000 rounds the product by adding five to the twelfth digit from the left, puts the 11-digit rounded number in register C and the rest of the number in register 1995, the Remainder Register. The exact product of +11 111 111 111 and +99 000 000 000 is +10 999 999 999 89 000 000 000.

Register	Contents
0	+11 111 111 111
1	+99 000 000 000

Instruction	Effect of Instruction
MUL/0/1/2	+11 000 000 000 ⟶ 2 +39 000 000 000 ⟶ 1995

Figure 1-16

Figure 1-16 shows the method of rounding the product of a multiplication. Five is added as shown in Figure 1-17, putting the first eleven digits of the sum in register 2 and the rest of the number in 1995.

NON-POSITIVE NUMBERS: In multiplying numbers that are not positive, the DATAmatic 1000 multiplies and rounds the corresponding positive numbers (called absolute values), then attaches the correct sign of the product to both halves.

DIVIDE: DIV/A/B/C. Operation code EE. Divide the number at address B by the number at address A and put the quotient at C and the remainder at 1995. If

+10 999 999 999	89 000 000 000
	50 000 000 000
+11 000 000 000	39 000 000 000

Figure 1-17

the numerator equals or exceeds the denominator, an automatic subsequence call is made to 1986, the Division Overcapacity Subsequence Register. Since no purpose would be served by dividing names and instructions, division only works for numeric words. The time required for division varies between 31 and 130 word cycles.

$$\frac{(B)}{(A)} \longrightarrow C$$

DECIMAL POINTS: If the decimal points of (A), (B), (C), and (1995) are interpreted as being a, b, c, and r places, respectively, from the left end of the 11-digit number, then

$$c = b - a$$
$$r = -11 + b$$

(a is positive if the decimal point is to the right of the left end and negative if the decimal is to the left of the left end.)

Figure 1-18 may be interpreted in many different ways. If the number in B is 10.0 and the number in A is interpreted as 3.0, then the quotient is 3.3 + .1/3; i.e., the remainder is .1. In this case, a = 1, b = 11, c = 10, and r = 0. If, on the other hand, the number in B is interpreted as .00 000 000 010 and the number at A is interpreted as 0.3, then (C) must be interpreted as .00 000 000 033 and (1995) as .000 000 000 001. Thus, a = b = c = 0, r = −11. Again if the number at B represents +10.0 and the number at A represents +30 000 000 000., then (C) will represent .00 000 000 033 and (1995) will represent .10 000 000 000. Thus a = b = 11, c = r = 0.

Register	Original Contents
A	+30 000 000 000
B	+00 000 000 010
C	--- --- --- ---
1995	--- --- --- --

Instruction	Effects of Instruction
DIV/A/B/C	(+00 000 000 033 ⟶ C) (+10 000 000 000 ⟶ 1995)

Figure 1-18

These examples should make it clear that the decimal point is in the mind of the user, not the computer, and the same set of digits in A and B may be interpreted as having their decimal points in various places as convenient. Wherever they may be chosen, the position of the decimal point in C and 1995 is then determined by the rules above.

SHIFT: For convenience in shifting numbers and alphanumeric words to the right or left, there are four Shift instructions in the DATAmatic 1000. For signed numbers there are two Shift instructions, SLP and SRP.

SHIFT LEFT PRESERVING SIGN: SLP/A/n/C. Operation code 5G. This instruction causes the DATAmatic 1000 to shift the eleven decimal digits, excluding the sign, of the word at A, n numeric places left and put the result at C. The sign is unchanged. The n digits on the left end of the word are discarded; the n new digits on the right end of the word are zeros (see Figure 1-19). n is the hundreds digit, or B_h (i.e. four bits), of the B address section of the instruction. It should be emphasized that to shift left ten places, $B_h = \overline{1010}$, not $\overline{0001}\ \overline{0000}$. Time: 7 + n word cycles.

Register	Contents
A	-00 016 000 016

Instruction	Effect of Instruction
SLP/A/4/C	-60 000 160 000 ——➤C

Figure 1-19

SHIFT RIGHT PRESERVING SIGN: SRP/A/n/C. Operation code 6G. The effect of this instruction is an obvious modification of SLP; namely, the shift is right instead of left. For alphanumeric words or numeric words without sign, there are two instructions, called SLW and SRW.

SHIFT LEFT WORD: SLW/A/n$\left\{\begin{matrix}A\\N\\T\end{matrix}\right\}$/C. Operation code 4F. The SLW/A/nA/C instruction and the corresponding right shift have been provided to make it possible to shift alphanumeric words for editing. The effect of the instruction is to shift the whole word at A (excluding check bits) $4B_h.B_t$ binary places to the left and put the result at C. As in the SLP instruction, the n digits on the left end of the word are discarded; the n digits on the right end of the word are zeros. B_h is again the hundreds digit of the B address and B_t is the tens digit. B_t will always be a 0 or a 5. For example, if B_h is 4 and B_t is 5, the amount of the shift will be four times 4.5, i.e., 18 binary places. Since one alphabetic character is six binary places the amount, nA, of alphabetic characters shifted is $4B_h.B_t/6$ as shown in Figure 1-20. In this case, the time required is 7 + 1.5n word cycles.

In addition, this instruction and the corresponding Right Shift may be used to shift numeric words of twelve decimal digits without sign. In this case, the symbolic form of the instruction is SLW/A/nN/C; the amount nN is the number of numeric characters to be shifted and is equal to B_h (B_t being normally zero). The time required to shift numeric words with the SLW instruction is 7 + n word cycles.

Alphabetic Characters Shifted	Hexadecimal Form of B Subaddress	Binary Places Shifted
nA		$4\ B_h\ B_t$
0	000	0
1	150	6
2	300	12
3	450	18
4	600	24
5	750	30
6	900	36
7	B50	42
8	D00	48

Figure 1-20

Register	Contents	Instruction (Symbolic Form)	Instruction (Actual Form)	Effect of Instruction
A	Johnston	SLW/A/4A/C	8 4F 000 600 001	ston ⟶ C
A	Johnston	SLW/A/5A/C	8 4F 000 750 001	ton ⟶ C
A	123456789876	SLW/A/4N/C	8 4F 000 400 001	567898760000 ⟶ C
A	123456789876	SLW/A/5N/C	8 4F 000 500 001	678987600000 ⟶ C
A	AC113456789	SLW/A/10T/C	8 4F 000 500 001	345678900000 ⟶ C

Figure 1-21

A third possible use of the Shift instruction is the case where it is desired to shift a word 2n binary places, as in words containing both alphabetic and numeric characters. The symbolic form of the instruction is SLW/A/nT/C; the amount nT is the number of two-bit units to be shifted and is equal to $2B_h \cdot B_t$. The time required in this case is 7 + n/2 word cycles. Some examples of SLW instructions are shown in Figure 1-21.

SHIFT RIGHT WORD: SRW/A/n{A N T}/C. Operation code 7F. This instruction differs from SLW only in the direction of the shift. The symbolic forms SRW/A/nN/C and SRW/A/nT/C are analogous to the corresponding Shift Left forms described above.

In order to facilitate the storage of more than one item of information in a word, use is made of the instruction:

SUBSTITUTE: SST/A/B/C. Operation code GD. The effect of SST is the following: (1) Consider the words at A, B, and C as 48-bit numbers, with the word at B as the controlling factor in the operation. (2) For every binary 1 in (B), take the corresponding bit from (A) and substitute it in (C); for every binary 0 in (B), preserve the corresponding bit in (C). (3) The new word in C is formed from bits of (A) and (C) as controlled by (B). Time: 7 word cycles.

In Figure 1-22 the bits of (A) and (C) which are rejected have been crossed out. The new word at C is composed of the bits of (A) and (C) which were accepted. As seen in this example, the hexadecimal digit G, i.e., $\overline{1111}$ in address B will extract a corresponding digit of the word at address A, and the hexadecimal 0, i.e., $\overline{0000}$ in address B will protect the corresponding digit of the word at address C.

EXAMPLE: The use of the Substitute and Shift instructions in performing arithmetic operations will be illustrated, where the numbers involved have been stored to save space. Each word consists of four positive three-digit numbers. To be precise, suppose that in register 0 there is a word composed as in Figure 1-23 and that in register 1 there is the employee's hourly pay rate in the corresponding situations. The decimal points are shown in Figures 1-23 and 1-24 for the reader's benefit. The week's gross pay for the employee is p where:

$$p = h_1p_1 + h_2p_2 + h_3p_3 + h_4p_4$$

A set of instructions and constants in storage, which accompiish the computation of the gross pay and store it in register 2, is shown in Figure 1-25. The reader is urged to trace the course of this example, step by step.

Contents Before Instruction												
A	1111	1101	1111	1010	1000	0000	0101	0111	1111	0101	1111	0101
B	0110	1111	0000	1010	0000	1111	1111	0000	0000	1010	1010	0001
C	0000	0101	0000	1111	1111	0000	1010	1111	1111	1111	0000	0000
Instruction: SST/A/B/C												
Contents After Instruction												
C	0110	1101	0000	1111	1111	0000	0101	1111	1111	0101	1010	0001

Figure 1-22

Hours Worked per Week (h_1)	Overtime Hours Worked on Regular Shift (h_2)	Night Bonus Hours (h_3)	Overtime Night Bonus Hours (h_4)
26.9	00.0	13.1	06.9

Figure 1-23

Straight Pay Rate (p_1)	Overtime Pay Rate (p_2)	Night Bonus Pay Rate (p_3)	Overtime Night Bonus Pay Rate (p_4)
1.95	2.93	2.15	3.22

Figure 1-24

Register	Contents
0	269 000 131 069
1	195 293 215 322
2	+00 000 000 000
3	G00 0GG GGG GGG
4	GGG GGG G00 0GG
5	+00 000 000 000
6	+00 000 000 000
7	+00 000 000 000

Instruction	Effect of Instruction
SRW/0/1/6	026 900 013 106 → 6
SST/5/3/6	+26 900 000 000 → 6
SRW/1/7/7	000 000 019 529 → 7
SST/5/4/7	+00 000 019 500 → 7
MUL/6/7/2	h_1p_1 = +00 000 005 245 → 2
SLW/0/2/6	900 013 106 900 → 6
SST/5/3/6	+00 000 000 000 → 6
SRW/1/4/7	000 019 529 321 → 7
SST/5/4/7	+00 000 029 300 → 7
MUL/6/7/6	+00 000 000 000 → 6
ADD/2/6/2	$h_1p_1 + h_2p_2$ = + 00 000 005 245 → 2
SLW/0/5/6	013 106 900 000 → 6
SST/5/3/6	+13 100 000 000 → 6
SRW/1/1/7	019 529 321 532 → 7
SST/5/4/7	+00 000 021 500 → 7
MUL/6/7/6	+00 000 002 817 → 6
ADD/2/6/2	$h_1p_1 + h_2p_2 + h_3p_3$ = +00 000 008 062 → 2
SLW/0/8/6	106 900 000 000 → 6
SST/5/3/6	+06 900 000 000 → 6
SLW/1/2/7	529 321 532 200 → 7
SST/5/4/7	+00 000 032 200 → 7
MUL/6/7/6	+00 000 002 222 → 6
ADD/2/6/2	$h_1p_1 + h_2p_2 + h_3p_3 + h_4p_4$ = +00 000 010 284 → 2

Figure 1-25

TAPE AND TRANSFER INSTRUCTIONS

Magnetic tape is used as the principal medium for the storage of information in the DATAmatic 1000. Information is stored on the magnetic tape in the form of magnetized spots. The magnetic tape used with the DATAmatic 1000 has tremendous storage capacity and speed. As in the Central Processor, 52 bits make up one word, and 62 words make up one block. Each reel of magnetic tape contains the storage capacity for 50,000 blocks, or 3,100,000 words. A block of 62 words is stored, two words per channel, in 31 channels, lengthwise along the tape. One of these channels (the one which holds words 1 and 2) is called the Key Channel. The other 30 channels are called Satellite Channels. There are five other channels holding control information which cannot be affected by the programmer. The blocks are arranged as shown in Figure 1-26.

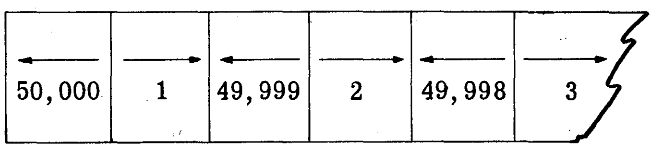


Figure 1-26

Blocks 1, 2, 3, etc., with the arrows pointing to the right are said by convention to belong to the first tape half; blocks 50,000, 49,999, etc., with the arrows pointing to the left are said to belong to the second tape half. The direction of the arrow is the direction in which writing is done and the direction in which forward reading is done. Suppose that one of the tapes is stopped with the reading head approximately at the middle of block 50,000. A single Read Forward instruction causes the tape to accelerate to full speed, cross block 1 at full speed, and decelerate to a stop with the reading head in the middle of block 49,999. Thus blocks 50,000 and 49,999 serve as start and stop spaces for the reading of block 1. This relationship holds along the length of the tape. At the physical end of the tape, tape direction is reversed and further Read Forward instructions cause the reading of the blocks on the second tape half which were previously used as start-stop spaces.

The Magnetic File Units read and record on the tape at the rate of 5000 words per second, and one may have as many Magnetic File Units as desired up to 100. Thus, the total tape storage capacity is 310,000,000 eleven-decimal-digit words. This capacity is available to the computer at all times.

A special feature built into the DATAmatic 1000 makes it possible to search 10 tapes simultaneously at full speed. Thus, it is possible to bypass 50,000 words per second in searching for a record.

To make these features possible, two Input Buffers, which will be called A Buffer and B Buffer, and two Output Buffers are provided. A buffer is a set of registers designed in such a way that it can accept information from magnetic tape at one rate and communicate it to the Central Processor at another (or vice versa). Each buffer holds 62 words.

Information flows, usually a block at a time, from some Magnetic File Unit to A or B Buffer. This is effected by means of the Read or Search instructions which will be discussed shortly. The information in the buffers can be sent to memory, one to 32 words at a time, by means of Transfer In instructions, also to be described. There is an electronic switch, controlled by certain of the Tape and Transfer instructions, which connects A or B Buffer to memory. A second switch, controlled by the Tape instructions, connects A or B Buffer to one of the Magnetic File Units to receive information. The Output Buffers are arranged in a simpler fashion. The First Output Buffer is always connected to memory and to the Second Output Buffer. The Second Output Buffer is connected to the first and, under control of the Write instructions, to some one Magnetic File Unit. Information is transferred (one to 32 words at a time) from memory to the First Output Buffer by the TXO instruction discussed below. The information then goes to the Second Output Buffer, a block at a time, and thence to some tape. This operation is performed by using Write instructions, to be defined later. The ability to take information from the Input Buffers as needed and transfer information to the Output Buffers as needed is a feature of the DATAmatic 1000 which frequently saves programming time.

The words in a tape block come into the Central Processor in the same order as they went from the Output Buffer onto tape. This is true whether the Read instruction is "forward" or "backward" (as discussed later). It should be emphasized that writing does not affect adjacent blocks; in fact, it is possible to make corrections by changing random blocks without affecting the adjoining information.

INTERLOCKS: In order to avoid interference, the DATAmatic 1000 is provided with controls called "interlocks" which automatically prevent the simultaneous performance of two instructions involving the same Magnetic File Unit. This feature also prevents the simultaneous performance of the following pairs of instructions involving different Magnetic File Units: (1) Read, Read, (2) Read, Search, (3) Write, Write. Another interlock involving the Search instruction is discussed later.

READ FORWARD: The Read Forward instructions are RFA, RFB, and RFD. The operation code for all three instructions is 36. The instructions are distinguished one from another by the bank designator and hundreds digit of the A address (D_aA_h).

RFA/A/B/C. The function of RFA is to initiate the reading in the forward direction of the next block from some magnetic tape into Buffer A. The forward direction is the direction indicated by the arrows in Figure 1-26, i.e., the direction of increasing block numbers. The Sequence Register is changed to B (unless B is zero) and the next instruction is obtained by a subsequence call to address C (unless C is zero). The tape to be read into Buffer A is determined by the tens and units digits of the A address (A_tA_u); any one of 100 tapes may be specified. The bank designator and the hundreds digit of the A address (D_aA_h) are 02.

If the Read instruction is given after the end of tape has been reached, it is not executed. Instead, it is put into the Current Instruction Register, and an automatic subsequence call to 1989 (the End of Tape Register) is made. This is true of all Read and Search instructions. After seven word cycles, and before the instruction is complete, the Central Processor can continue with the execution of other instructions provided they do not call for further use of the same Magnetic File Unit or the use of a Read or Search instruction on any other Magnetic File Unit.

It takes 427 word cycles for the complete process —initiating the Read instruction, setting up the proper interlocks, moving the tape to the beginning of the block, reading the block into the buffer, sensing the end-block mark, and releasing the proper interlocks. Only then is the information which was read into the buffer available to the Central Processor. However, since there are many operations to be carried out in connection with information previously obtained from the tape, the time required to fill Buffer A by this Read instruction is virtually free. In other words, this time is gained because the filling of A Buffer is done in parallel with the other instructions. If during the 427 word cycles there should be another instruction involving the same Magnetic File Unit or a Read or Search instruction on another Magnetic File Unit, then the completion of this particular Read instruction must be awaited.

Each tape mechanism contains switches which determine the use of the read or write state. All three Read Forward instructions, RFA, RFB, and RFD, put these switches for the selected tape mechanism into the read state. Switching between read and write is relatively time-consuming when compared with arithmetic operations. It is usually advisable, therefore, to arrange the work in such a way that one reads from certain tapes and writes on others. Specifically, a Read instruction which follows a Write instruction, or vice versa, takes about 500 word cycles; however, as in normal read or write situations, any instructions which do not conflict can be executed within seven word cycles.

The RFB instruction differs from the RFA instruction only in that it fills Buffer B instead of Buffer A and that the hundreds digit of the A address is 8 instead of 2.

Example: If the tape on a particular Magnetic File Unit contains a series of one-block items to be processed, these items can be read into one buffer, e.g. the A Buffer, transferred to memory and processed, as shown in simplified form in Figure 1-27. The purpose of box AA is to "prime the pump" by loading the A Buffer originally. In box AB an RFA instruction is used to start refilling the A Buffer as soon as it is emptied. Processing goes on in parallel with buffer refilling. For simplicity, operations which would control stopping have been omitted. If the processing takes 300 word cycles (case 1), the total time to go through box AB is 497 word cycles: 70 word cycles for the Transfer instruction (see page 1-19) plus 427 word cycles for the Read instruction. Processing is free and the program is said to be "tape limited." If the processing time were 500 word cycles (case 2), then the total time to go through box AB would be 577 word cycles: 70 for the Transfer instruction, seven for the Read instruction, and 500 for the processing. In this case, reading is essentially free and the program is said to be "machine limited."

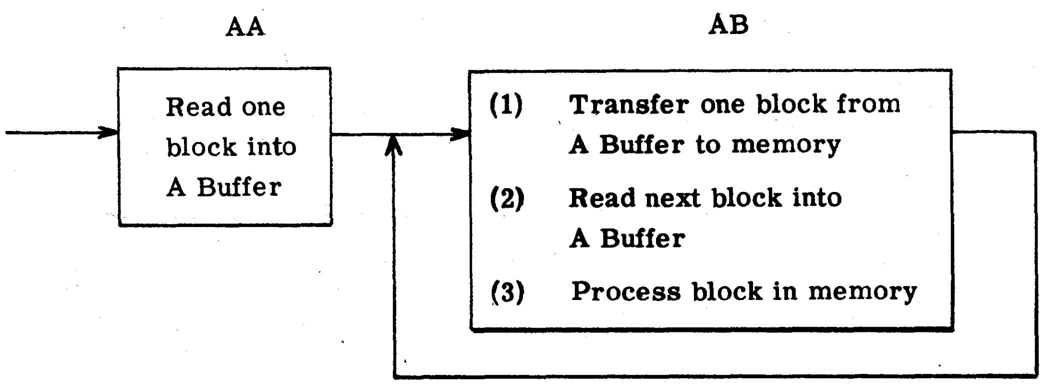


Figure 1-27

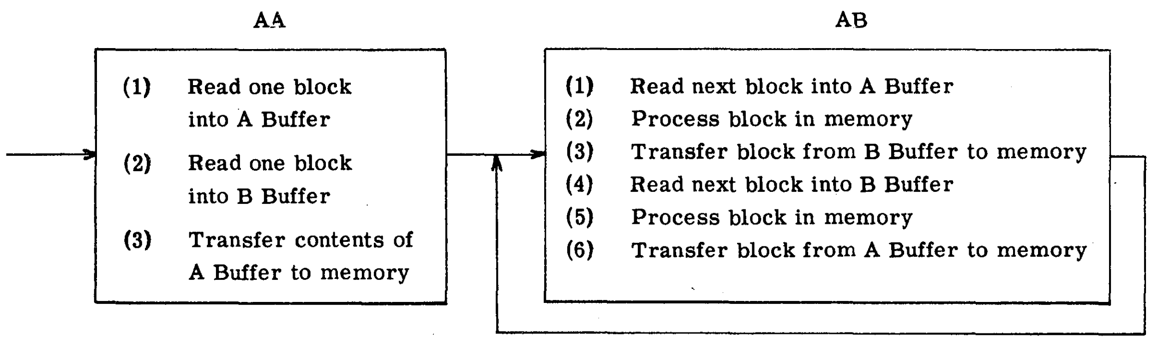


Figure 1-28

In case 1, a slight gain in speed could be obtained by using the A and B Buffers alternately so that transferring and processing could both parallel reading as shown in Figure 1-28. In this figure not only are steps 2 and 5 free, but steps 3 and 6 are also free. Thus the total time per block processed is 427 word cycles.

If the items to be processed are smaller than one block, it is convenient, as will be made clear in the discussion of the Transfer In instruction, to withdraw the items from the buffers as needed. It would be especially desirable in this case to use both buffers so that one can be filling as the items in the other are being used. This can be arranged according to Figure 1-29.

In Figure 1-29, if the items are 10 words long, and if the time required to transfer in an item, process it, and check whether the buffer is empty is less than 70 word cycles, then those operations are free and the program is tape limited. If the operations mentioned take more than 70 word cycles, then the Read instructions are free and the program is machine limited.

An instruction has been provided to simplify programming and to facilitate the alternate use of both buffers. This is the Read Forward into Different Buffer instruction.

RFD/A/B/C. Operation code is again 36. The bank designator and hundreds digit of the A address (D_aA_h) are 00. In executing this instruction, the Central Processor causes a block of information from the tape designated by the tens and units digits of the A address (A_tA_u) to be read into a buffer which is different from the last buffer used by a Read instruction. That is, if the last block was read into A Buffer, the next block will be read into B Buffer and conversely. There is a Transfer In instruction associated with each of these three Read Forward instructions which will be discussed later. The RFD instruction simultaneously connects the High-Speed Memory to the buffer not being read into.

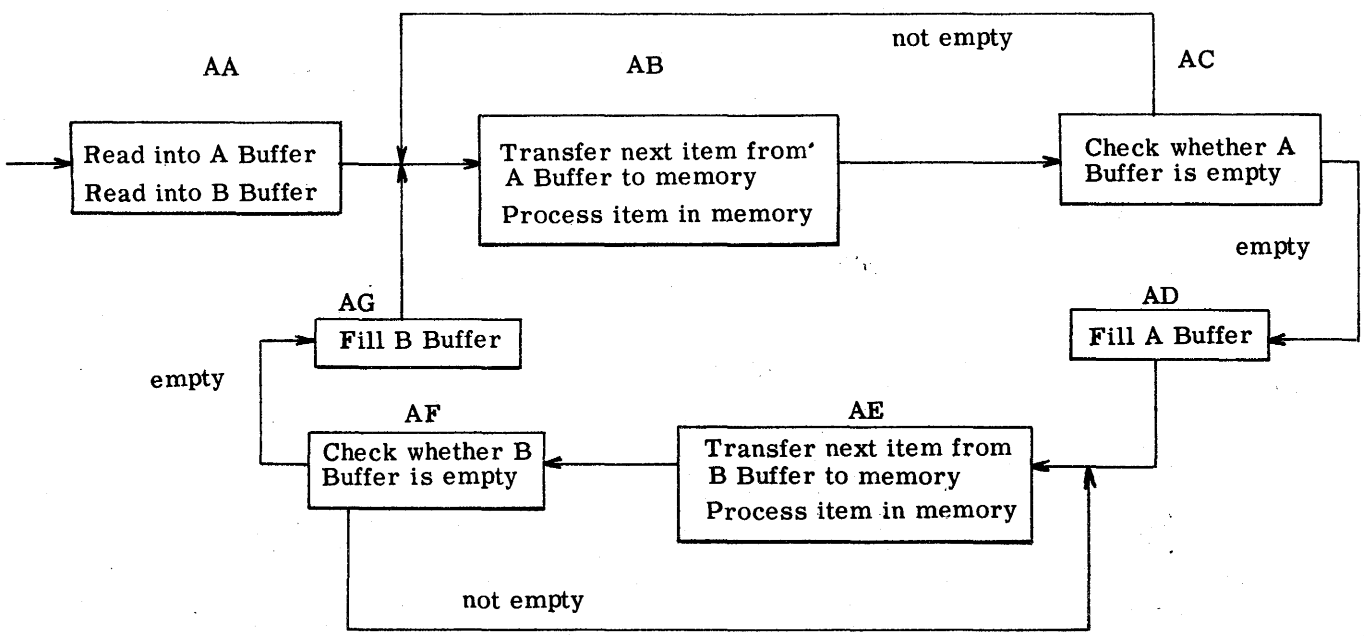


Figure 1-29

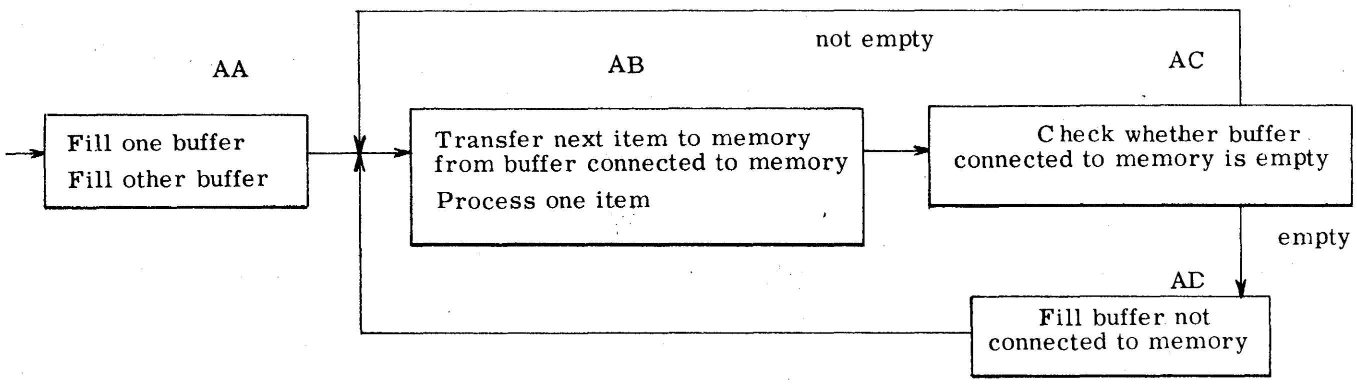


Figure 1-30

This makes it possible to replace Figure 1-29 by Figure 1-30. In each case, the appropriate buffer is filled by means of the RFD instruction which automatically connects the other buffer to memory. The timing is unaffected by this programming simplification.

In each of the Read Forward instructions, if the end of tape is reached, the instruction is not performed. Instead, the Read instruction being processed is transferred to register 1999, the Current Instruction Register, and a subsequence call is made to register 1989 where the first instruction of an end-of-tape process may be stored.

READ BACKWARD: The three instructions, RBA, RBB, and RBD, differ from the Read Forward instructions only in their operation code, which is 24 instead of 36, and in causing the tape to be read backward instead of forward. It should be observed that the complete time to execute a Read Backward instruction after a Read Forward instruction, or any other instruction which involves switching tape directions, is about 600 word cycles, all or part of which may be overlapped by other instructions which do not conflict.

READ FORWARD KEY CHANNEL: RFK/A/B/C. Operation code again 36. $D_a = 0$, $A_h = 6$. This instruction causes the two key words from the tape specified by A_tA_u to be read into the B Buffer. The remainder of the B Buffer is cleared and filled with fillers. The satellite channels are put into the write state by this instruction.

READ BACKWARD KEY CHANNEL: RBK/A/B/C. Operation code again 24. $D_a = 0$, $A_h = 6$. This instruction differs from RFK only in that the tape is read backward.

TRANSFER INSTRUCTIONS: There is a whole family of instructions designed to facilitate the transfer of 1 to 32 words at a time from one part of the machine to another. In each of these instructions, the words moved should not come from the non-existent registers 1991, 1996, or 1998, nor from 999 and 1000 simultaneously. Likewise the words should not be sent to 1990 nor to 999 and 1000 simultaneously.

For all of the Transfer instructions except "Transfer Internally", the B address section of the instruction is used for control purposes rather than as an address. As summarized in Figure 1-31, n - m is used only in the Sorting instructions, B_p is used to determine whether or not certain interlocks apply, B_r is used in the input transfers to determine which buffer is involved. On the other hand, n is used in every Transfer instruction to determine the number of words from 1 to 32 to be moved (in "Transfer Internally" it is in the C address

B Address Section of Instruction				
Bit Position	50 28 27 26 25	24	23 22	21 20 19 18 17
Symbol	n - m	B_p	B_r	n
Use	Used only in Sorting instructions to determine sorting word	Controls interlock	Determines buffer involved	Number of words to be moved
Applicable to	Sorting instructions	Input instructions		

Figure 1-31

portion of the instruction). The code for n is shown in Figure 1-32. It is a straight binary code except that $\overline{00000}$ is used for 32 instead of for zero.

TRANSFER IN: TIA/A/n/C, TIB/A/n/C, TIS/A/n/C, TID/A/n/C. The Transfer In instructions take information from the Input Buffers to the High-Speed Memory. They are designed to be efficient and flexible; consequently they are somewhat complicated.

	Bit Position 21	20	19	18	17	Number of words to be moved
n =	0	0	0	0	0	32
	0	0	0	0	1	1
	0	0	0	1	0	2
	0	0	0	1	1	3
	0	0	1	0	0	4
	0	0	1	0	1	5
	0	0	1	1	0	6
	0	0	1	1	1	7
	0	1	0	0	0	8
	0	1	0	0	1	9
	0	1	0	1	0	10
	0	1	0	1	1	11
	0	1	1	0	0	12
	0	1	1	0	1	13
	0	1	1	1	0	14
	0	1	1	1	1	15
	1	0	0	0	0	16
	1	0	0	0	1	17
	1	0	0	1	0	18
	1	0	0	1	1	19
	1	0	1	0	0	20
	1	0	1	0	1	21
	1	0	1	1	0	22
	1	0	1	1	1	23
	1	1	0	0	0	24
	1	1	0	0	1	25
	1	1	0	1	0	26
	1	1	0	1	1	27
	1	1	1	0	0	28
	1	1	1	0	1	29
	1	1	1	1	0	30
	1	1	1	1	1	31

Bars omitted for simplicity.

Figure 1-32

TIA/A/n/C. Operation code G3. The basic function of this instruction is to transfer n words from Buffer A to the High-Speed Memory starting at address A. In the TIA instruction, m is ignored (see Figure 1-31), B_p is $\overline{0}$, and B_r is $\overline{01}$; n determines how many words shall be transferred from the Buffer. The C address of this instruction is used for a subsequence call to the next instruction. Time: 4 + n word cycles.

In the process of transferring information into the High-Speed Memory from the Input Buffer, all of the other words in the buffer move forward one step. Thus, suppose the instruction TIA/A/2/C is given 31 times. The first time, words one and two will come into the High-Speed Memory; the second time words three and four will come in, etc. As shown in Figure 1-32, if n is $\overline{00000}$, then 32 words will be brought in from the buffer. The time required to execute this instruction is 4 + n word cycles, where n is the number of words transferred.

All TI and TB instructions are subject to the restrictions that they should not send information to 1990 nor 1992, nor to both 999 and 1000. Address C must not be 1990.

SENTINELS: To provide extra flexibility and to simplify programming, the DATAmatic 1000 has been provided with the ability to recognize special words known as sentinels. A sentinel is a word whose bits 51-52 are $\overline{00}$. Thus, no number with a legitimate sign can be a sentinel. An instruction is a sentinel only if it is negative and if its A address is less than 1000. An alphanumeric word could be a sentinel if the first two bits are $\overline{00}$. A few illustrations on the use of sentinels will be given shortly. In the process of executing the TIA instruction, each word transferred is examined to see if it is a sentinel, and if it is, it is stored in register 1997. Furthermore, if the last word being transferred is not a sentinel, then the next word that would be transferred is also examined and if it is a sentinel, it is transferred to 1997. If two sentinels are discovered, only the first one is put in 1997. If no sentinel is discovered, a Pass instruction is stored in 1997. Pass instructions simply do nothing and go on to the next instruction. They are discussed under Control instructions. By making the C address of the TIA instruction equal to 1997, one may use a sentinel to control the operations following the TIA instruction. When a Transfer In instruction is performed, n words are always transferred to the memory even though there may be less than n words remaining in the buffer. After the words remaining in the buffer are stored, words called fillers are transferred. The

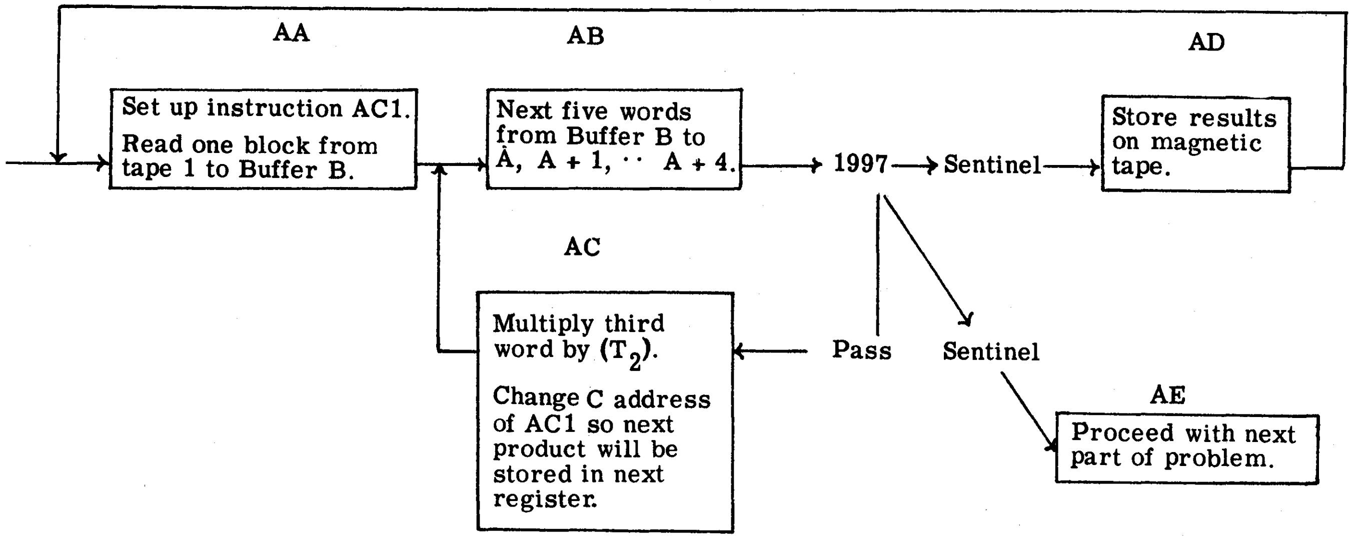


Figure 1-33

fillers are in the form of Sentinel instructions: SCS/0/0/1985. The use of these fillers is shown in the following example (see Figures 1-33 and 1-34).

EXAMPLE: Assume that a tape is arranged with twelve 5-word items per block. (An item is a set of information, any number of words, having some coherence and making it desirable to manipulate it as a unit. A file is a set of such items.) Furthermore, after the last item on tape, there is a control item whose second word is the sentinel SCS/0/AE1/0. (AE1 is a symbol representing the address of the first instruction in operation box AE. Similarly, AC3 represents the address of the third instruction in operation box AC.) It is desired to multiply the third word of each item by a constant and store the results in consecutive positions in the memory, starting at address Y. After processing 12 items, it is desired to record the 12 products on tape and then repeat with the next 12 items.

In each of the first 12 iterations, five words are transferred to A, A + 1, ...A + 4. When the subsequence call is made to 1997 from AB1, a Pass instruction is performed because this is the instruction which is automatically transferred to the Sentinel Register when no sentinels are encountered. On the 13th iteration, words 61 and 62 are sent to A and A + 1, then three sentinels of the form SCS/0/0/1985 are transferred to A + 2, A + 3, and A + 4. The first of these sentinels is sent to 1997. The subsequence call in AB1, which is made to 1997, this time causes a subsequence call to be made to 1985. Then the 12 products are stored on tape and the next instruction is from box AA. This process continues until the sentinel SCS/0/AE1/0 is read in from Buffer B by instruction AB1. The subsequence call to 1997 this time finds a different sentinel there and the machine proceeds to box AE.

This example shows how useful sentinels can be. They reduce the need for counting, thereby eliminating instructions and increasing the speed.

WEIGHT COUNT ERROR: In any Transfer In instruction, if there is a weight count error in the words being transferred, the machine will stop if the CHECK SUBSEQUENCE switch on the Central Console is set to "off", or it will make a subsequence call to 1987 if this switch is set to "on". Therefore, in the example coded in Figure 1-34, if it is desired to stop upon detection of an incorrect weight count, the CHECK SUBSEQUENCE switch should be set to "off". If, upon detection of an error, it is desired to transfer control to some other program, for instance to print out some information to be examined, the CHECK SUBSEQUENCE switch should be set to "on" and the instruction SCS/0/B/0 should be stored at 1987 previous to read-in, where B is the address of the first instruction in the program to transfer information to the Console Typewriter for printing.

TIB/A/n/C: For this instruction, $B_r = \overline{1}\overline{0}$ and the information is transferred from B Buffer; otherwise, this instruction is the same as TIA.

TIS/A/n/C: For this instruction $B_r = \overline{0}\overline{0}$. S means use the buffer which is connected to the Central Processor.

TID/A/n/C: For this instruction $B_r = \overline{1}\overline{1}$. D means use the buffer which is not connected to the Central Processor.

TRANSFER IN (BYPASSING INTERLOCK): TB(A, B, S, D)/A/n/C. Operation code again G3. $B_p = \overline{1}$. B_r is used as in the TI instructions to indicate the buffer involved. These four instructions have been provided to speed up the transfer of information from

Space Assignment	STORAGE						
	Register	Contents					
		Sign and Bank Designator	Operation Code	A	B	C	
0001	T1	8	FE	013	002	100	or MUL/A+2/T2/Y in symbolic form.
0002	T2	+	55	555	555	555	T1, T2, T3, T4, Y, Y+1, A, etc., are symbols used for the names of registers where words are stored that are useful in the code below. T1 holds an instruction to be put in AC1 (called setting up instruction AC1). T2, T3, and T4 hold constants. The input items are stored temporarily in A, A+1, ··· A+4 and 12 answers are stored in Y, Y+1 ··· Y+11 before being stored on magnetic tape.
3	T3	+	00	000	000	001	
4	T4	+	00	000	000	000	
0100	Y						
0101	Y+1						
	.						
	.						
	.						
0011	A						

Space Assignment	INSTRUCTION					EFFECT OF INSTRUCTION		HEX FORM OF INSTRUCTION				
	Instruction Name	Instruction Type	A	B	C		Footnote	Sign and Bank Designator	Operation Code	A	B	C
0005	AA1	ADD	T1	T4	AC1	MUL/A+2/T2/Y $\longrightarrow$ AC1	1	8	FF	001	004	008
6	2	RFB	1	0	0	Read into Buffer B.		8	36	801	000	000
7	AB1	TIB	A	5	1997	5 words from Buffer B to A, A+1, ··· A+4.	2	9	G3	011	045	997
8	AC1	(MUL	A+2	T2	Y)	(A+2) · (T2) $\longrightarrow$ Y, Y+1, ··	3	8	FE	013	002	100
9	2	ADD	AC1	T3	AC1	AC1 + (T3) $\longrightarrow$ AC1	4	8	FF	008	003	008
10	3	SCS	0	AB1	0	Return to AB1 for next instruction.		8	00	000	007	000
	1997	PSS	0	0	0	Go to Sequence Register for address of next instruction.		8	82	000	000	000
		or				or				or		
		SCS	0	0	1985	Go to 1985 for next instruction		1	00	000	000	985
		or										
	AD1	SCS	0	AE1	0							
	1985	First instruction of box AD				Store results on magnetic tape						

1. This sets up instruction AC1. If the Transfer and Subsequence Call instruction (see page 1-21) had been defined, it would have been used instead of Add to move a word from AC1.
2. The five registers A, A+1, ···, A+4 are used to hold one item temporarily.
3. The instruction is written here as it was set up in operation box AA. Brackets are written around this instruction to indicate that it keeps changing.
4. This instruction modifies instruction AC1 so that the C address is increased by one each time. Thus, the results of the multiplication will be stored in successive registers.

Figure 1-34

tape to memory by making it possible to remove information from a buffer at the same time that a Read instruction is progressing. They must be given shortly (approximately 150 word cycles) after a Read instruction.

Consider, for example, the instruction TBB/A/n/C. When this instruction is given, n words will be moved from Buffer B to the n successive registers starting at A, concurrently with the execution of an RFB instruction. (An interlock would have made a TIB instruction wait until the RFB instruction was completed, but the TBB instruction bypasses the interlock.) The words will be the n words left in Buffer B by an earlier Read instruction. More precisely, there are about 200 word cycles after a Read instruction is initiated before any information enters the buffer. This is the period when the space between blocks is crossing the reading heads. During this period, there is plenty of time to move all the information in the buffer to memory and perform many operations on it. However, the programmer should use the TBB instruction before the information he wants is replaced.

Example: As an example of the use of this instruction, consider a payroll problem in broad outline. Suppose that tape one holds information about the hours that employees have worked. This information might be arranged in 3-word items giving the employee number, normal hours worked, and hours worked on overtime. There could be twenty of these 3-word items per block arranged by employee number order. On tape two, suppose that there is stored a master file of information about employees giving name, address, pay rate, etc. This may be arranged with two 30-word items per block, in order by employee number. Thus, even though the two files are in the same order by employee number, tape two will move more often.

Greatly oversimplified, certain features of the computation of pay checks may be represented as shown in Figure 1-35. In this arrangement, the RFB instruction must be completed before the TIB instruction is started; these two instructions take about 495 word cycles. Assuming that the pay check can be computed in 350 word cycles, the total time per check is about 845 word cycles.

Simply by rearranging the process, it can be speeded up considerably as in Figure 1-36. In this case, the TIB instructions take 68 word cycles and the RFB instruction takes 427 word cycles. The pay check is computed while the RFB instruction proceeds. Thus, the total time per check is 495 word cycles.

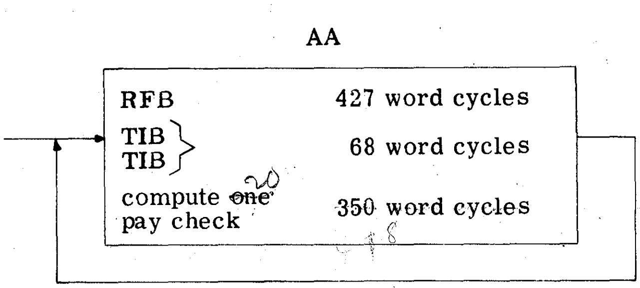


Figure 1-35

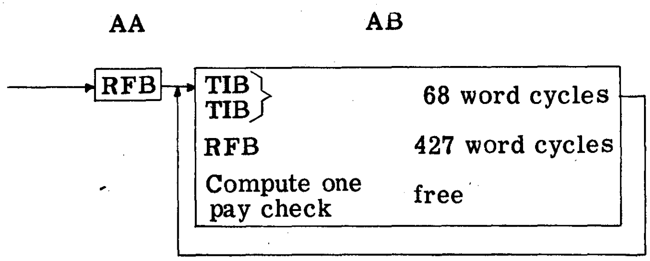


Figure 1-36

Using TBB, the process can be rearranged as in Figure 1-37. In this case, the TBB instructions and the computation of one pay check are all performed during the Read instruction which is setting up the next block of information. The total time per check is 427 word cycles. Thus, the Bypass Interlock instructions made it possible to save up to 70 word cycles by initiating the next Read instruction immediately before removing information from the buffer. The TB instructions are otherwise identical to the TI instructions.

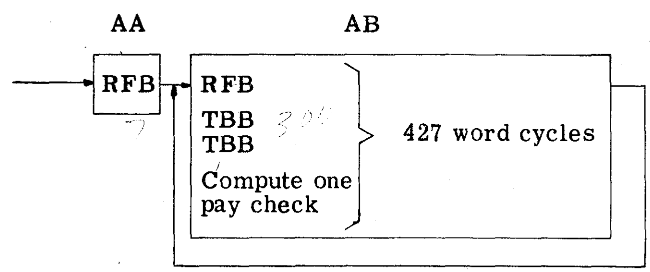


Figure 1-37

Transfer Internally: TXI/A/B/n. Operation code E1. This instruction causes n words to be moved from the n registers starting at A to the n registers starting at B. n is determined by the five least significant bits of address C (in this it differs from the other instructions that move n words). As usual, 32 is specified by n = 0, and 1, 2, 3, ... by n = 1, 2, 3, ... The TXI instruction clearly saves time and space when it is desired to move several words. Time: 3 + 2n word cycles.

Note: The range of addresses specified by A and n or by B and n must not include both 999 and 1000. Address B must not be 1990 and if n is more than 1, the range of address B plus n must not include a special address.

Transfer and Subsequence Call: TXS/A/B/C. Operation code D0. Since the special case of moving just one word is of frequent occurrence, the TXS instruction has been provided which moves one word from A to B and subsequences to C. Address B must not be 1990. This is a fast instruction and its subsequence feature makes it useful in the cyclic counter (see page 1-27). Time: 5 word cycles.

TRANSFER OUT: TXO/A/n/C. Operation code G0. Just as there are input buffers A and B which each hold a block of information, there are two buffers used for output which each hold a block of information. One of these is called "The Output Buffer". The other, which may be called "the Second Output Buffer," is not under control of the programmer. It is used as temporary storage during a Write instruction to free the Output Buffer to receive new information before completion of the Write instruction. The instruction which puts information in the Output Buffer is TXO. It causes the computer to transfer n words to the Output Buffer from the consecutive registers starting at A. When the 63rd word is transferred to the Output Buffer, the first is pushed out as an overflow word. The 64th word pushes out the second as an overflow, etc. The first overflow word which is a sentinel is stored in 1997, otherwise a Pass is stored in 1997. Subsequence to C. The use of sentinels to control the output of information is illustrated on page 1-20. Time: 4 + n word cycles.

NOTE: No information should be transferred out from 1991, 1992, 1996, 1998, nor both 999 and 1000. C should not be 1990.

WRITE FORWARD, WFA AND WFP. Operation code 17. It was previously stated that one of the 31 channels on a tape is called the key channel since it can be used to hold a number which identifies an item of information. The others are called satellite channels. The difference between WFA and WFP is concerned with putting the key channel into the write or read state, respectively. Time: 7 or 427 word cycles.

WFA/A/B/C. Write forward one block (from the Output Buffer) onto the tape specified by the tens and units digits of the A address (A_tA_u). Both the key and satellite channels of the tape specified are put into the write state, if they are not already in this state. A_h is zero for this instruction.

WFP/A/B/C. Write forward, except on key channel, one block (from the Output Buffer) onto the tape specified by A_tA_u. The satellite channels of the specified tape are put into the write state and the key channel into the read state to protect the key channel and prevent complete accidental loss of file information. A_h is two for this instruction.

The following description applies to both Write instructions. If the Output Buffer is not full, fillers are added, each consisting of a sentinel SCS/0/0/1984. The contents of the Output Buffer are then written on file Unit A_tA_u, moving the tape forward. Actually, the contents of the Output Buffer are moved temporarily into the Second Output Buffer freeing the Output Buffer, after seven word cycles, to receive further information via the TXO instruction; then the information is written onto magnetic tape. The Sequence Register is changed to B and a subsequence call is made to C.

If the end of tape has been reached, this instruction will not be performed. Instead it will be transferred to the Current Instruction Register (1999) with its subaddresses permuted in the order C/A/B. The bank designators are not permuted. An automatic subsequence call is made to 1989.

REWIND TAPE: REW/A/B/C. Operation code 05. Rewind the tape specified by the tens and units digits of address A. Upon completion of this instruction the tape is positioned for reading or writing in the first block. The execution of this instruction may be simultaneous, except for the first eight word cycles, with the processing of other instructions not using this particular file unit. Put B in the Sequence Register and subsequence to C. Time: 7 word cycles until start of next instruction.

FILE MAINTENANCE INSTRUCTIONS: There are four Tape instructions (the Search instructions) and two Decision instructions (the Key Comparison instructions) which were specifically designed to make the maintenance of large files highly efficient. These will be discussed later.

SORTING INSTRUCTIONS: There are 17 Transfer instructions (the Transfer and Select instructions and the Twin Transfer) which were specifically designed to make sorting efficient. These will also be discussed later.

SWITCH TAPE HALF: Two instructions have been provided to make it possible to start at any block and go to any other block in a minimum of time.

SWITCH TO FIRST HALF: SWF/A/B/C. Operation code 05. D_aA_h = 02. This instruction prepares the Magnetic File Unit specified by A_tA_u to read a block from the first half of the tape, puts B in the Sequence Register, and subsequences to C. Time: 7 or 250 word cycles.

SWITCH TO SECOND HALF: SWS/A/B/C. Operation code 05. D_aA_h = 08. This instruction prepares the Magnetic File Unit specified by A_tA_u to read a block from the second half of the tape, puts B in the Sequence Register, and subsequences to C. Time: 7 or 250 word cycles.

Seven word cycles after the start of a Switch Tape Half instruction, the system is free to perform another instruction not involving the same Magnetic File Unit. 250 word cycles are required before the initiation of a Read instruction involving the same Magnetic File Unit. Because of timing considerations there are simple programming restrictions recommended in conjunction with either of these instructions:

1. The Switch instruction must not be followed by a Write instruction.
2. The pair of instructions, Read Backward and Switch, must not be followed by Read Forward.
3. The information left in the Input Buffer as a result of the first Read instruction following the Switch instruction is invalid. Consequently, it must not be transferred to memory.

DECISION INSTRUCTIONS

The instructions which make it possible for any computer to choose between two different courses of action all through its computation are its Decision Instructions. The DATAmatic 1000 has six of them. Two of them, the Key Comparisons, are designed for file maintenance and will be discussed under that instruction group. Each of these instructions looks for some special circumstance and, if it finds it, puts C in the Sequence Register. This, of course, has the effect of making the instruction at C the next one executed. If the condition sought is not found, the next instruction executed will be the one following the comparison. In either case, the next instruction will be the one in the Sequence Register.

NUMERIC COMPARISONS: The numeric comparisons compare the numbers at addresses A and B. They are "algebraic" comparisons, i.e.

−0 equals +0
−7 is less than −5
−7 is less than +3
+3 is less than +5, etc.

LESS THAN COMPARISON, NUMERIC: LCN/A/B/C. Operation code 3D. If the number at A is less than or equal to the number at B, put C in the Sequence Register. Time: 7 word cycles.

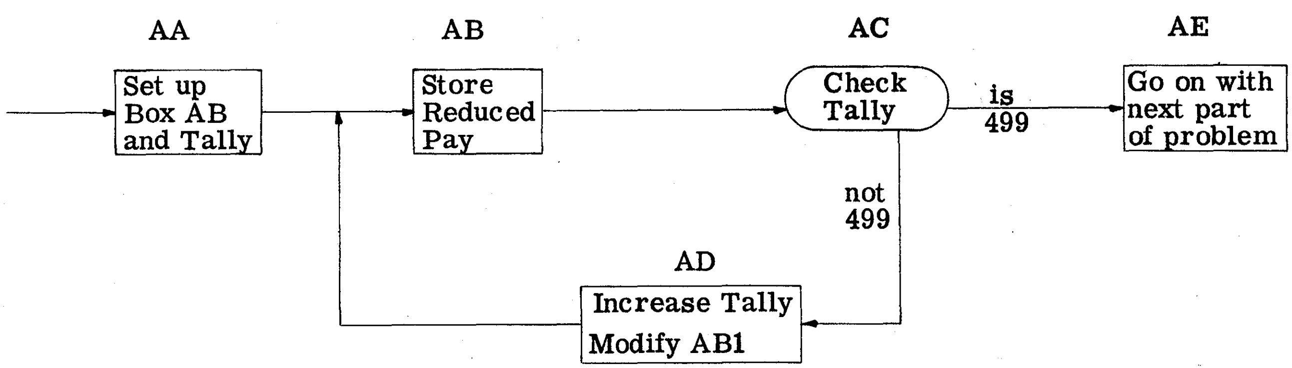
NOTE: A change in the Sequence Register, other than adding one to it, will subtract one word cycle from the time required by all of the comparison instructions.

INEQUALITY COMPARISON, NUMERIC; ICN/A/B/C. Operation code 0D. If the number at A is not equal to the number at B, put C in the Sequence Register. Time: 7 word cycles.

ALPHABETIC COMPARISONS: The alphabetic comparisons compare the words at addresses A and B, bit by bit. If the word at A has a zero in the first position where they differ reading from left to right, it is considered "alphabetically less" than the word at B. The order of the alphanumeric characters is 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, ..., Z. Furthermore,

−0 is less than +0,
3 is less than 7,
−3 is less than −7,

Any negative number is less than any positive number.



Storage		Instruction	Type	A	B	C	Effect of Instruction
Reg	Contents						
S1	+0	AA1	TXS	S1	T1	0	+ 0 ⟶ T1
S2	+ 499	AA2	TXS	S4	AB1	0	SUB/0/500/0 ⟶ AB1
S3	0/1/1/1*	AA3	SCS	0	AB1	0	next instruction is AB1
S4	SUB/0/500/0	AD1	ADD	AB1	S3	AB1	SUB/1/501/1 ⟶ AB1
S5	+ 1	AD2	ADD	T1	S5	T1	Add 1 to T1
T1	Tally originally equal to 0.	AB1	(SUB	0	500	0)	(0) - (500) ⟶ 0
		AC1	ICN	T1	S2	AD1	
		AE1					

* 0/1/1/1 is shorthand notation for a positive number having ones in the positions corresponding to the addresses of an instruction: +00 001 001 001.

Figure 1-38

LESS THAN COMPARISON, ALPHABETIC: LCA/A/B/C. Operation code 2E. Put C in the Sequence Register if the word at A is alphabetically less than or equal to the word at B. Time: 7 word cycles.

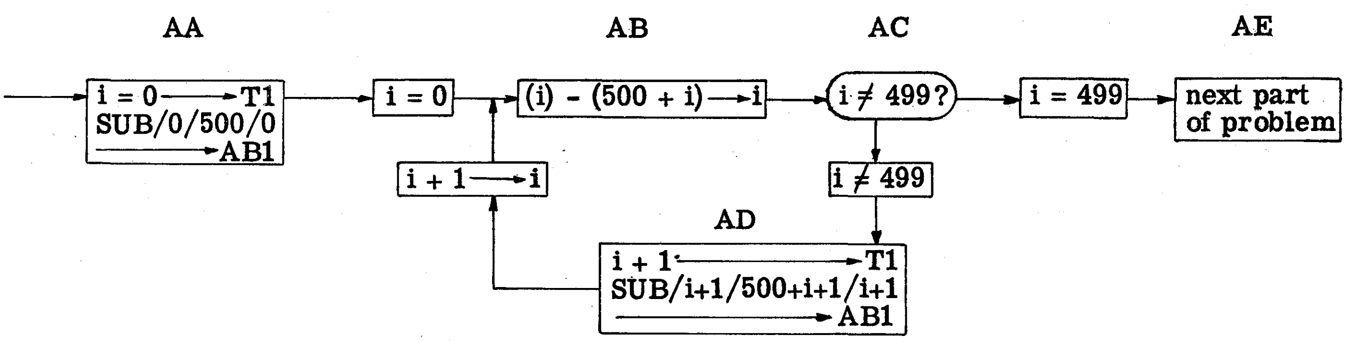
INEQUALITY COMPARISON, ALPHABETIC: ICA/A/B/C. Operation code 1E. If the words at A and B are not identical, put C in the Sequence Register. Time: 7 word cycles.

Only a few simple illustrations of the uses of these instructions are given here. However, the user will quickly visualize others.

COUNTING: On page 1-9, Figure 1-12, the Subtract instruction was illustrated by subtracting 500 numbers from 500 other numbers. On page 1-9, it was shown that by repeating three instructions the machine could do in 0.33 seconds what it could do with 500 instructions in 0.17 seconds. Actually, Figure 1-13 does not quite represent a practical solution to the problem because there is no provision for passing from the repetition of the loop to the next operation, called an "exit" from the loop. Figure 1-38 shows how the ICN instruction solves this problem.

TALLY: A tally (also known as a counter, induction variable, or bound variable) is an integer which keeps track of the number of times a loop of operations has been performed. The flow diagram in Figure 1-38 is self-explanatory. 499 times a man's pay is reduced, the tally is checked, the tally is increased, and instruction AB1 is modified. The 500th time, the tally is found to equal 499 so the control jumps to box AE1 as it should. Thus the computer counted to 499 and then went on to other operations.

PROGRAMMING AND CODING: In box AB of Figure 1-38 the instruction is written the way it was first stored in register AB1. Actually, however, this instruction is never the same twice around the loop. Therefore, the form SUB/0/500/0 is an appropriate description of this instruction only once. Likewise, there was no attempt to list the contents of T1, since it starts at 0 and changes to 1, 2, 3, ... 499. In order to be able to write flow diagrams and variable instructions in a generally meaningful form, a letter such as "i" is introduced for the tally. "i" is said to be zero the first time through the loop and to increase by one each time through the loop. When "i" is 499, the computer goes to box AE. Using "i", the instruction AB1 is written as SUB/i/500+i/i. (Thus on the sixth pass through the loop, "i" equals five and AB1 is SUB/5/505/5.) In box AD, where one is added to each address of AB1, it is considered that SUB/i+1/500+i+1/i+1 is put in AB1. (Thus if "i" equals five, it is SUB/6/506/6 that is put in AB1.) In going from box AD to box AB, the point of view changes and it is considered that "i" has increased by one and that the instruction in AB1 is now SUB/i/500+i/i again. Thus "i" has increased to six so the instruction represented by SUB/i/500+i/i is now SUB/6/506/6. This is the purpose of the substitution box in Figure 1-39 where i + 1 ➤ i means that "i" is increased by 1. The substitution box is introduced as a convenience to the programmer to make



Storage	
Reg.	Contents
S1	+ 0
S2	+ 499
S3	0/1/1/1
S4	SUB/0/500/0
T1	i

Instruction	Type	A	B	C	Effect of Instruction
AA1	TXS	S1	T1	0	i = 0 → T1
AA2	TXS	S4	AB1	0	SUB/0/500/0 → AB1 (i. e., i = 0)
AA3	SCS	0	AB1	0	AB1 → S. R.
AD1	ADD	AB1	S3	AB1	SUB/i+1/500+i+1/i+1 → AB 1
AD2	ADD	T1	S1	T1	i+1 → T1
AB1	(SUB	i	500+i	i)	(i) - (500+i) → i
AC1	ICN	T1	S 2	AD1	
AE1					

Figure 1-39

precise the point where he changes notation. No instructions are involved.

The flow diagram is further clarified by introducing, where convenient, boxes called assertion boxes which contain assertions about the status of certain induction variables. Thus, the assertion box after AA says that "i" equals zero and that between box AC and box AE says that "i" equals 499. Only the operation boxes correspond to instructions executed by the computer. The substitution and assertion boxes are optional refinements to be used where needed. They are not assigned letter designations since there are no associated operations. These refinements are included in Figure 1-39. In addition, the ends of box AC are rounded to symbolize the fact that it executes a Decision instruction. Such boxes are called decision boxes.

The execution by the computer of a series of instructions is a dynamic process which modifies the contents of various registers. Consequently, statements about the contents of these registers are only valid at specific times in the course of the problem. The place to make such statements is by the lines called "constancy intervals" between the operation boxes, where nothing is happening. This is done by labelling the constancy intervals and attaching storage tables which are labelled the same way. Only information about the contents of registers affected by nearby operation boxes is put in these storage tables. This is shown in Figure 1-40. Such complete detail is rarely required and constancy-interval labelling and attached storage tables are used only where they clarify the flow diagrams.

Note that instruction AB1 has the form SUB/499/999/499 the last time it is used. Then instruction AB2 changes it to SUB/501/000/500. The instruction is, however, never used in this form because the ICN instruction finds that "i" is the same as the word in S2 and AE1 then becomes the next instruction.

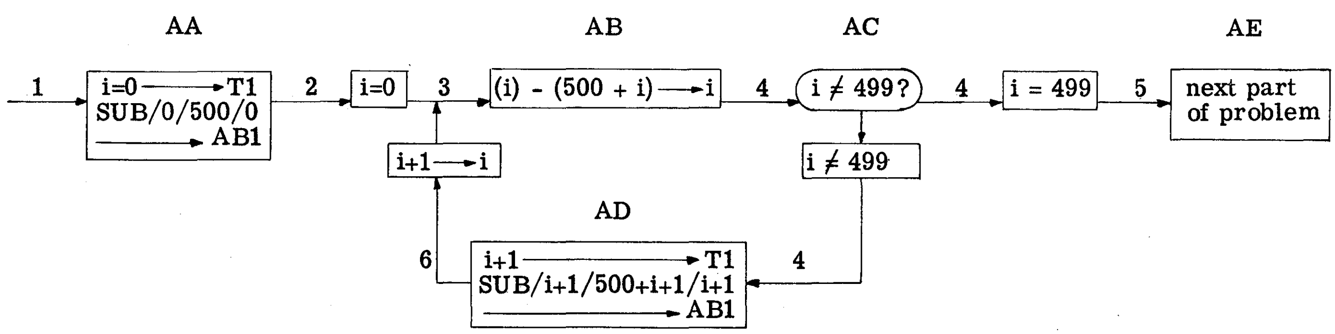
It is possible to save three instructions and illustrate the use of the ICN instruction by using the instruction AB1 itself as a tally as shown in Figure 1-41.

Marking: In some situations, it is not known in advance how many times a loop should be executed, but a marker can be used to stop the process. For example, it is desired to find the average performance of the personnel of a department. To be precise, suppose that on tape one there is a series of items of information about company personnel. In each item there are two words as follows:

+00 ddd eee eee
+00 00p. pp0 000

ddd stands for three digits of department number; eee eee stands for six digits of employee number; p. pp stands for three digits of performance. It is assumed that these items have been sorted into order by department and in each department by employee number.

The program shown in Figure 1-42 accumulates the individual performance in register 0 and the number of individuals in the department in register 1. Each time it has completed the two accumulations for one individual, it checks the next individual to see whether he is in the same department or not. This is done by



1	
T1	---
AB1	---
S1	1
S4	SUB/0/500/0

2	
T1	0
AB1	SUB/0/500/0

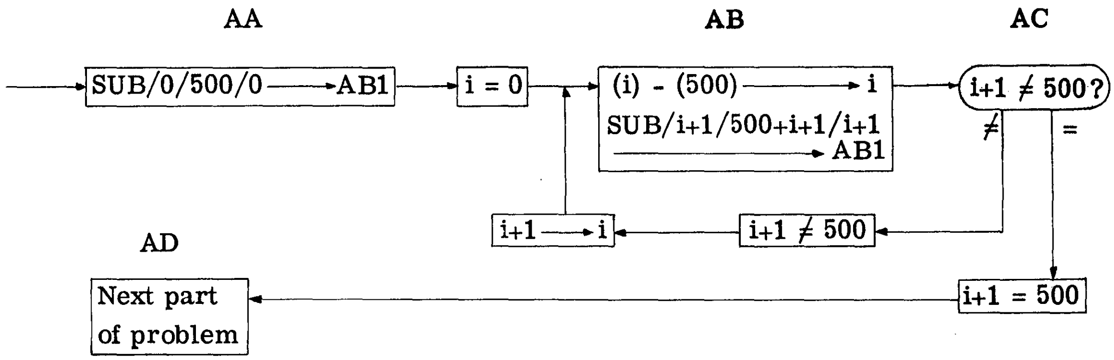
3	
i	i^{th} man's gross pay
500+i	i^{th} man's deduction
T1	1
AB1	SUB/i/500+i/i

6	
T1	i+1
AB1	SUB/i+1/500+i+1/i+1

4	
T1	i
AB1	SUB/i/500+i/i
i	i^{th} man's net pay

5	
T1	499
AB1	SUB/499/999/499
999	499^{th} man's pay

Figure 1-40



Storage	
Reg.	Contents
S1	SUB/0/500/0
S2	SUB/ 501/000/500
S3	0/1/1/1

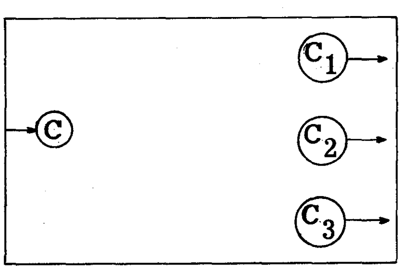
Instruction	Type	A	B	C	Effect of Instruction
AA1	TXS	S1	AB1	0	SUB/0/500/0 → AB1
AB1	(SUB	i	500+i	i)	i^{th} man's net pay → i
AB2	ADD	AB1	S3	AB1	SUB/i+1/500+i+1/i+1 → AB1
AC1	ICN	AB1	S2	AB1	
AD1	next part of problem				

Figure 1-41

looking at the department number. If the department number is the same as the preceding one, the loop is reentered, but if the number is different it serves as a marker. In this case the department performance is computed, written onto tape as a two-word item in a full block, the accumulations in registers 0 and 1 are set back to zero, and the process continues for the next department. To stop the computation of department performance and start some other computation, it is assumed that the last item is followed by an artificial empty item and then a sentinel which leads to the next operation. The output items will have the form:

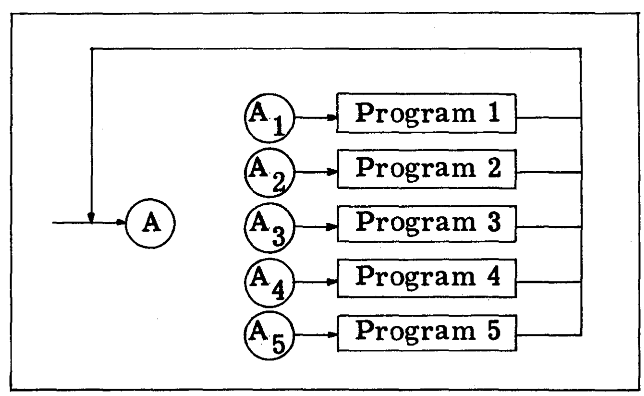
+00 ddd 000 000
+p. pppp ppp ppp

Variable Remote Connector: Figure 1-42 illustrates the use of a diagramming device called a variable remote connector. It indicates that several different paths are possible. The path chosen at any time is determined by the program. Thus, instruction AD1 ends with a subsequence call to 1997. What happens next depends on the contents of 1997. There are three possibilities summarized in the flow diagram by C_1, C_2, and C_3, in the following fashion:



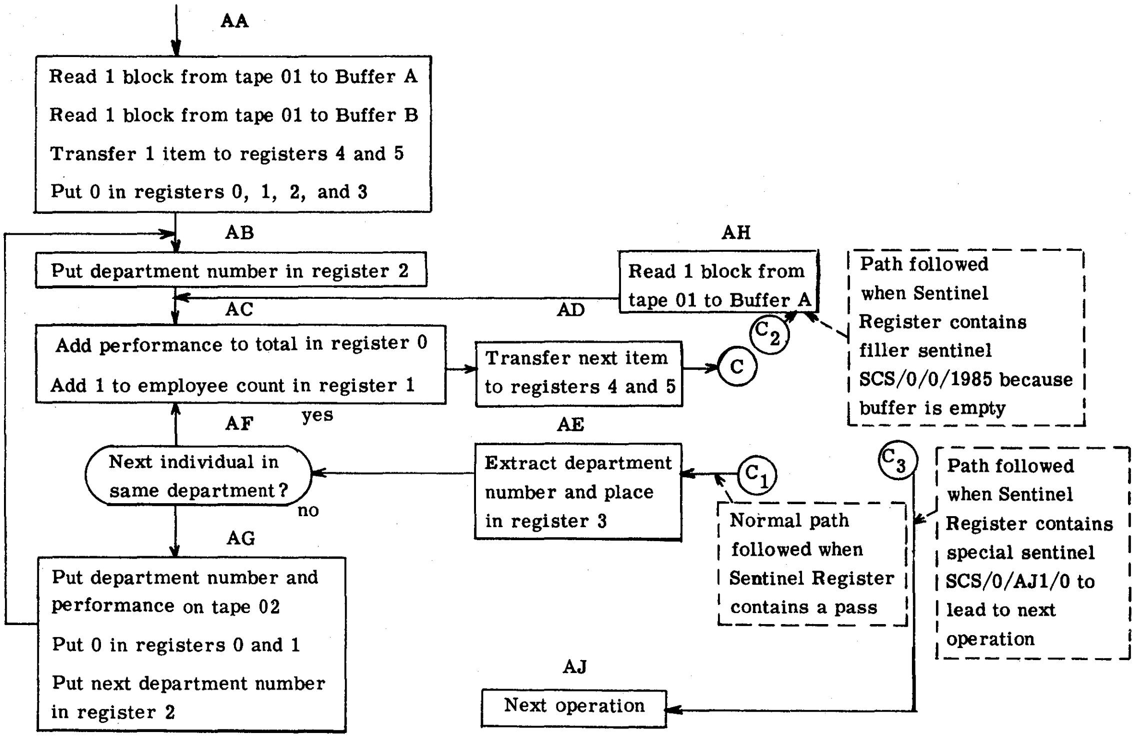
There are many other ways to obtain a variable remote connector, several of which will be illustrated later. One useful example of the variable remote connector is:

The Cyclic Counter: Suppose that there are any number, such as five, of different programs to be repeated in a cyclic fashion. The sequence of operations



consists of program 1, then program 2, then program 3, then program 4, then program 5, then program 1, program 2, and so on, continuously. The code for this can be arranged as follows: Let a set of five TXS instructions be stored in registers T1, T2, ... T5, in the form

TXS/T2/w/A1
TXS/T3/w/A2
TXS/T4/w/A3
TXS/T5/w/A4
TXS/T1/w/A5



Storage	
Register	Contents
0	Sum of individual performances for department
1	Number of individuals considered
2	Department number
3	Department number of current individual
4	+00 ddd eee eee
5	+00 00p pp0 000
6, 7, 8, 9	+00 000 000 000
10	+00 010 000 000
11	000 GGG 000 000

Instruction	Type	A	B	C	Effect of instruction
AA1	RFA	1	0	0	First block to A Buffer
AA2	RFD	1	0	0	Second block to B Buffer. Connect A Buffer to memory.
AA3	TIS	4	2	0	First item to registers 4 and 5.
AA4	TXI	6	0	4	+0 to registers 0, 1, 2, and 3.
AB1	SST	4	11	2	First department number to register 2 in form +00 ddd 000 000.
AC1	ADD	0	5	0	Sum of performances for employees considered.
AC2	ADD	10	1	1	Number of employees considered.
AD1	TIS	4	2	1997	Put next item in registers 4 and 5.
AE1	SST	4	11	3	
AF1	ICN	2	3	AG1	
AF2	SCS	0	AC1	0	
AG1	DIV	1	0	3	Obtain department performance.
AG2	TXO	2	2	0	Department number and performance to Output Buffer.
AG3	TX1	6	0	2	+0 to registers 0 and 1.
AG4	WFA	2	AB1	0	Department number and performance to tape 02. Next instruction is AB1 which puts new department number in register 2 and goes to AC1.
AH1= 1985	RFD	1	0	0	Put next block in other buffer and return to AE.
AJ1	Next operation				

Figure 1-42

where w is what is called a working location and A1, A2, A3, A4, A5 are the addresses where the five programs start. The process can be started by performing the instruction in T1. This puts (T2) in w and subsequences to A1. The end of each program, including the first, is a subsequence to w, which puts the next TXS instruction into w and subsequences to the next program.

EXAMPLE: Suppose that the tape on Magnetic File Unit number one (MFU1) contains a succession of five-word payroll items of the form

Word	Symbolic Form	Contents
1	eee eee 000 000	employee number
2	+00 000 0hh∧h00	regular hours
3	+p∧pp00 000 000	regular pay rate
4	+00 000 0hh∧h00	overtime hours
5	+p∧pp00 000 000	overtime pay rate

where the ∧ indicates the position of the decimal point (which is not carried in the word by the machine). There are twelve such items in each block. Two 16-word items are to be put in a block on tape two in a suitable form for the Model 1300 Output Converter. Each of these 16-word items (see Figure 1-43) will contain three 4-word pay items, three voids, and a control word (see Model 1300 Output Converter, Section III).

Word	Contents	Word	Contents
1	Employee Number	9	Employee Number
2	Regular Pay	10	Regular Pay
3	Overtime Pay	11	Overtime Pay
4	Gross Pay	12	Gross Pay
5	Employee Number	13	+00 000 000 000
6	Regular Pay	14	+00 000 000 000
7	Overtime Pay	15	+00 000 000 000
8	Gross Pay	16	Control Word

Figure 1-43

A twelve-stage cyclic counter is used which has four distinct outlets. The code for this operation is presented in Figure 1-44, together with a flow chart for the essential part of the operation.

FILE MAINTENANCE

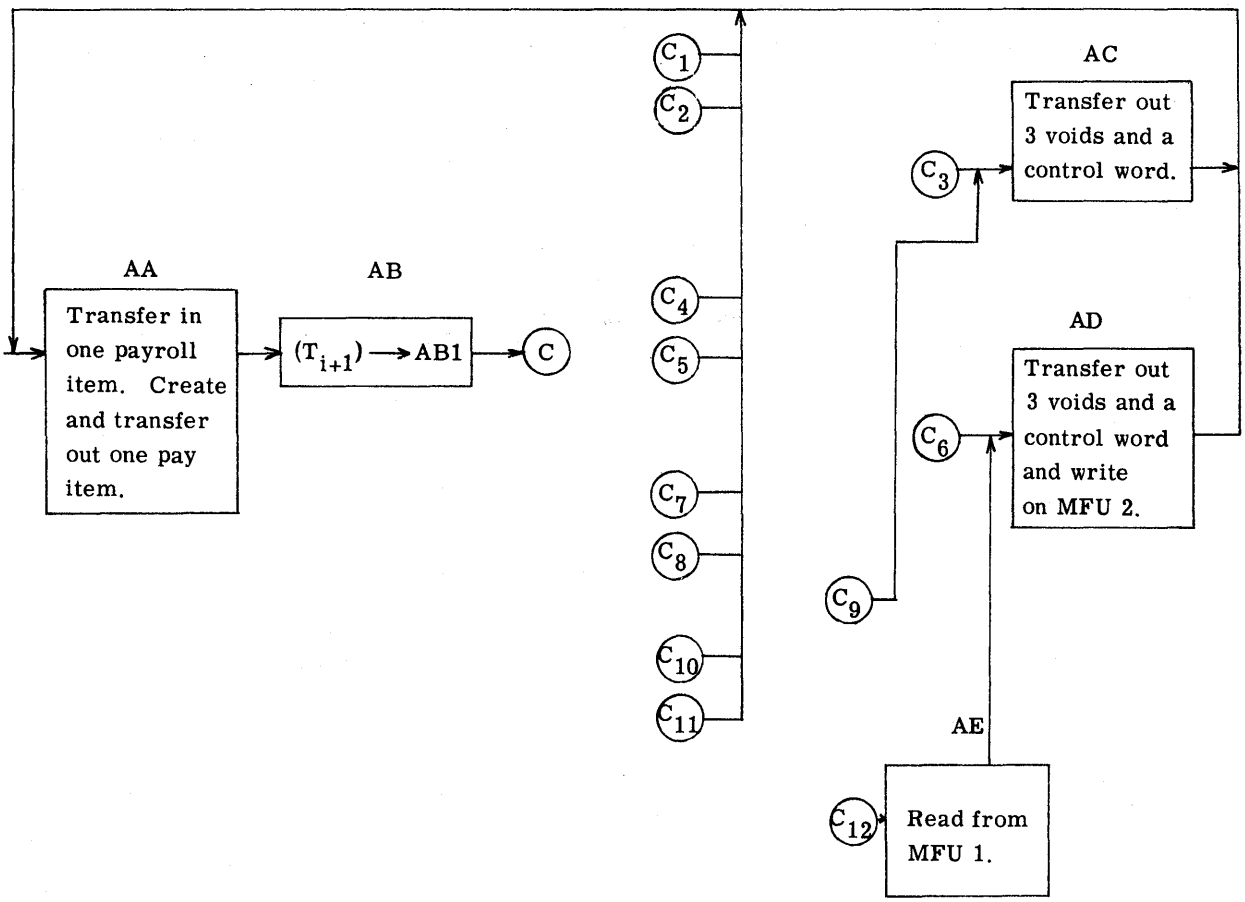
An extremely important and very common business application of electronic data-processing machines is the maintenance of a large main file of information. Whether the items in the file are the policies of a life insurance company, the inventory of a mail order house, or the checks outstanding of a bank, the maintenance of the file involves a number of operations. A typical breakdown of such operations might include (1) assembling periodically (daily) a transaction tape affecting only a small percentage of the items of the main file; (2) sorting these transaction items to put them in the same order as the items of the main file; (3) searching through the main file to locate the items with the same key number (e.g. policy number, part number, or check number); (4) performing some operations with the information contained in the matching items; (5) updating the main file by putting back a modified item; and (6) writing reports based on information extracted from the active items.

It will be noted that operation (3), the searching operation, involves going through the whole file even though only a few items are sought. The DATAmatic 1000 has made this, the bulkiest part of file maintenance, very efficient by means of special Search instructions. These instructions make it possible to run ten tapes at once, looking only at a key word in each block. Thus searching can progress at 50,000 words per second.

SEARCH INSTRUCTIONS: The search instructions make use of the fact that the A Buffer can be divided into ten 2-word channels, each one of which can receive two words from the key channel of a Magnetic File Unit. The remaining 42 registers of the A Buffer are not used during the search process (see Figure 1-45). Each DATAmatic 1000 tape has 31 channels, of which one is called the key channel. The others are called satellite channels. The four Search instructions have certain features in common. They each search, that is, cause the two words in the key channel of the Magnetic File Unit specified by A_tA_u to be read into the A Buffer channel specified by A_u. In addition, B is put in the Sequence Register and there is a subsequence call to C.

SEARCH FORWARD READING: SFR/A/B/C. Operation code 36. $D_a = 1$, $A_h = 2$. Put all channels in the read state and search forward. Time: 7 or 427 word cycles. As in all of the other tape-handling instructions, the minimum time after the Search instruction has started before the next instruction can start is 7 word cycles. The time to complete the instruction is 427 word cycles, which may be partially or completely in parallel with the execution of other instructions.

SEARCH FORWARD WRITING: SFW/A/B/C. Operation code 36. $D_a = 1$, $A_h = 4$. Put the key channel in the read state, the satellite channels in the write state, and search forward. Time: 7 or 427 word cycles.



Storage	
Reg.	Contents
T1	TXS/T2/AB1/AA1
T2	TXS/T3/AB1/AA1
T3	TXS/T4/AB1/AC1
T4	TXS/T5/AB1/AA1
T5	TXS/T6/AB1/AA1
T6	TXS/T7/AB1/AD1
T7	TXS/T8/AB1/AA1
T8	TXS/T9/AB1/AA1
T9	TXS/T10/AB1/AC1
T10	TXS/T11/AB1/AA1
T11	TXS/T12/AB1/AA1
T12	TXS/T1/AB1/AE1
S1	+00 000 000 000
S2	+00 000 000 000
S3	+00 000 000 000
S4	Control word

Instruction	Type	A	B	C	Effect of instruction
C_1 = AA1	TIS	w_1	5	0	Bring in next payroll item.
AA2	MUL	w_2	w_3	w_2	Create pay item.
AA3	MUL	w_4	w_5	w_3	
AA4	ADD	w_2	w_3	w_4	
AA5	TXO	w_1	4	AB1	Transfer out pay item.
AA6	SCS	0	AA2	0	Change Sequence Register.
AB1	(TXS	T_{i+1}	AB1	C_i)	Cyclic counter
C_3 = C_9 = AC1	TXO	S1	4	AA1	Complete first output item with 3 voids and control word.
C_6 = AD1	TXO	S1	4	AD2	Complete second output item.
AD2	WFA	2	0	AA1	Put 2 output items on tape 2.
C_{12} = AE1	RFD	1	0	AD1	Fill Input Buffer.

Figure 1-44

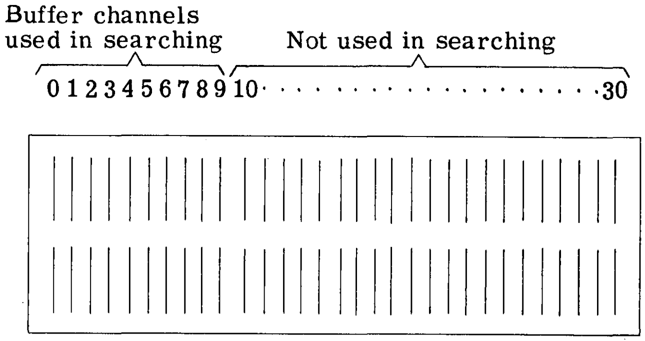


Figure 1-45

SEARCH BACKWARD READING: SBR/A/B/C. Operation code 24. $D_a = 1$, $A_h = 2$. Same as SFR except for tape direction. Time 7 or 427 word cycles.

SEARCH BACKWARD WRITING: SBW/A/B/C. Operation code 24. $D_a = 1$, $A_h = 4$. Same as SFW except for tape direction. Time: 7 or 427 word cycles.

KEY COMPARISONS: There are two instructions, called the First and Second Key Comparisons, which are designed to operate in conjunction with the Search instructions.

FIRST KEY COMPARISON: FKC/A/B/C. Operation code 78. (1) If the first half of the A Buffer channel designated by A_u is full or filling, transfer its contents to address A, waiting if necessary until it is full. If empty and not filling, eliminate this step. (2) If the word at A is a sentinel, put it in 1997; if not, put a Pass instruction (PSS/0/0/0) in 1997. (3) If the word at B is alphabetically less than or equal to the word at A, put C in the Sequence Register. (For definition of "alphabetically less than" see page 1-24.) (4) Connect the A Buffer to memory. If a weight count error is detected during performance of a Key Comparison instruction, the machine will stop unconditionally. Time: 8 word cycles.*

SECOND KEY COMPARISON: SKC/A/B/C. Operation code 69. This instruction differs from the First Key Comparison in the fact that it deals with the second half of the buffer channel designated by A_u. If the word at B is alphabetically more than, or equal to the word at A, C is put in the Sequence Register. Time: 8 word cycles.*

EXAMPLE: Searching is such an important feature of the DATAmatic 1000 that a whole section (Section IX) has been devoted to it. Consequently, the present example is limited to a simple illustration of the rudiments of searching. Suppose that file units zero, one, two, etc., through nine contain a master file of one million 30-word insurance policies, ordered by increasing policy number. Word one of each block is the policy number of the second policy in the block. In this example, no use is made of word two in each block.

* See note, page 1-24.

A series of ten thousand 30-word transactions affecting random policies are stored in the required order on file unit ten.

Transactions are brought into memory through Buffer B and their policy numbers, p_0, p_1, ... p_9, are stored at P_0, P_1, ... P_9. That is, P_0 contains p_0, a policy number for a policy on tape zero; P_1 contains p_1, a policy number for a policy on tape one, etc. A starting process, not shown, loads A Buffer channels zero through nine with words one and two from file units zero through nine. The inner loop of Key Comparison and Search instructions brings the key, k_0, from file zero into K_0 and compares it with p_0. If k_0 is greater than or equal to p_0, there is said to be a strike and searching is interrupted by a jump to A_0 to process this transaction by reading the appropriate policy into memory via the B Buffer and taking what action is required. Searching is then resumed after a new key is put in P_0. If k_0 is less than p_0, tape zero is moved forward one block and the next key is brought into buffer channel zero by a Search instruction. The key from tape one is then compared to p_1, etc.

Figure 1-46 shows the inner loop of the operation. The Search instruction related to file nine sets the Sequence Register to AA1 to close the loop. As long as no strikes occur, searching is performed simultaneously on all ten Magnetic File Units and information is bypassed at the rate of 50,000 words per second.

Instruction	Type	A	B	C
AA1	FKC	K_0	P_0	A_0
AB1	SFR	0	0	0
AC1	FKC	K_1	P_1	A_1
AD1	SFR	1	0	0
.				
.				
.				
AQ1	FKC	K_8	P_8	A_8
AR1	SFR	8	0	0
AS1	FKC	K_9	P_9	A_9
AT1	SFR	9	AA1	0

Figure 1-46

SORTING INSTRUCTIONS

The most frequent operation performed in business applications with electronic data-processing machines is that of sorting. A set of items of information is stored on magnetic tape in one order (possibly random order) and must be rearranged to some other order.

There are two rather different ways to accomplish this ordering: merge sorting and digital sorting. Because of its high tape speed, the DATAmatic 1000 is very efficient at merge sorting (see Section X). The DATAmatic 1000 is, however, even more efficient at digital sorting because of special instructions designed for this purpose. (Whether digital or merge sorting is most efficient in a particular case depends principally on K and k. K is the number of items, while k is the number of digits in the sort key. This point is discussed more fully in Section X.) These instructions make it possible to sort using approximately two instructions per item, per digit sorted. It is not surprising, therefore, that these sorting instructions are quite complicated. Many users will never have to use these instructions to create a sorting program since a library of sorting routines will be available. However, since the instructions have many other uses, they are described here.

For all of the sixteen Transfer and Select instructions, as for the Transfer In and Transfer Out instructions, the B address is used to govern the instruction. The B address is laid out as in Figure 1-47 (see also Figure 1-31).

DOUBLE TRANSFER AND SELECT: DTA (DTB, DTS, DTD)/A/B/C. Operation code GB. Time: 6 + n word cycles.

1. Transfer n words to the Output Buffer from the consecutive registers starting at address A. Output Buffer overflow is not examined for sentinels. A+n−1 must be less than 1989.
2. Transfer n words from the Input Buffer to the n registers starting at Address A. The n words moved and the next one are examined for sentinels. The first sentinel found is stored in 1997. If none is found, a PSS/0/0/0 is stored in 1997.
3. Under control of the word in register 1993, substitute, from the mth word transferred, into a special register which contained only zeros. Call the word so obtained the "extracted word." For example:

m^{th} word in item moved	+09 541 057 041
word in 1993	G00 GGG 000 0G0
extracted word	+00 541 000 040

4. Generate a hexadecimal digit from the extracted word as follows. Place the 12 hexadecimal characters of the extracted word above one another. After each character, write out the binary form of this digit. Then put a one at the foot of each column which contains one or more ones and a zero at the foot of all other columns. The result will be the binary form of the generated digit. This process is shown in Figure 1-48.

Frequently, the word in 1993 will contain just one G to extract a single decimal digit and then the generated digit will be the extracted digit.

5. Add the digit generated in step 4 to the units digit of address C of the instruction

B Address of Transfer and Select Instructions				
Bit position	50 28 27 26 25	24	23 22	21 20 19 18 17
Name	D_b B_h	B_p	B_r	n
Other name	n - m			
Use	Determines sorting word	Bypass interlock if B_p is 1.	(00)-S (01)-A (10)-B (11)-D	Determines number of words moved.

Figure 1-47

Transfer 32 words & don't select on any?

Digits of the extracted word	+	1 1 0 1
	0	0 0 0 0
	0	0 0 0 0
	5	0 1 0 1
	4	0 1 0 0
	1	0 0 0 1
	0	0 0 0 0
	0	0 0 0 0
	0	0 0 0 0
	0	0 0 0 0
	4	0 1 0 0
	0	0 0 0 0
Generated digit	E	1 1 0 1

Figure 1-48

itself as shown in Figure 1-49. If this sum exceeds nine, there will be a carry and if it exceeds nineteen there will be an error and the machine will stop.

If n is greater than the number of words left in the buffer, the buffer will make up the deficiency with sentinel fillers of the form SCS/0/0/1985. If m is less than or equal to zero, C is unmodified. If C is less than 1000, modified C must be less than 1000.

6. Make a subsequence call to the new address C, computed in step 5, unless the original C address was zero or an error was detected in the n words transferred from the Input Buffer.
7. Store the original instruction in 1994.

Steps 1 and 2 move items in and out of the Central Processor very efficiently, requiring only one word cycle per pair of words moved. The extractor used in step 3 is usually a G which extracts a particular decimal digit. Thus step 4 is generally quite simple. The addition of the extracted digit to the C address makes it possible to go to one of ten (or even sixteen) different places for the next instruction. This is the part of the instruction which is equivalent to the brush in a mechanical sorter detecting a hole in a particular column and putting the card in a corresponding pocket. Thus, in summary, the general function of this instruction in combination with the Twin Transfer instruction is to put items of arbitrary size onto different magnetic tapes as a mechanical sorter puts cards into different hoppers.

In general, the sorting process takes only two instructions per item, per digit in the sort word. The process is limited only by the speed of the tape for items containing more than two words. It also requires only one pass, per key-word decimal digit of the items, through the Central Processor. Thus, the time required to put a file of K n-word items in order by a k-digit sort key, if there are 20 MFU's in the system, is

$$\frac{K\ n(k+1)}{5000} \text{ seconds.}$$

For example, if K is 1,000,000, n is three, and k is five, the sort time is 3600 seconds or 60 minutes.

The details of one digital sorting routine and the properties of digital sorting routines in general are discussed in section X.

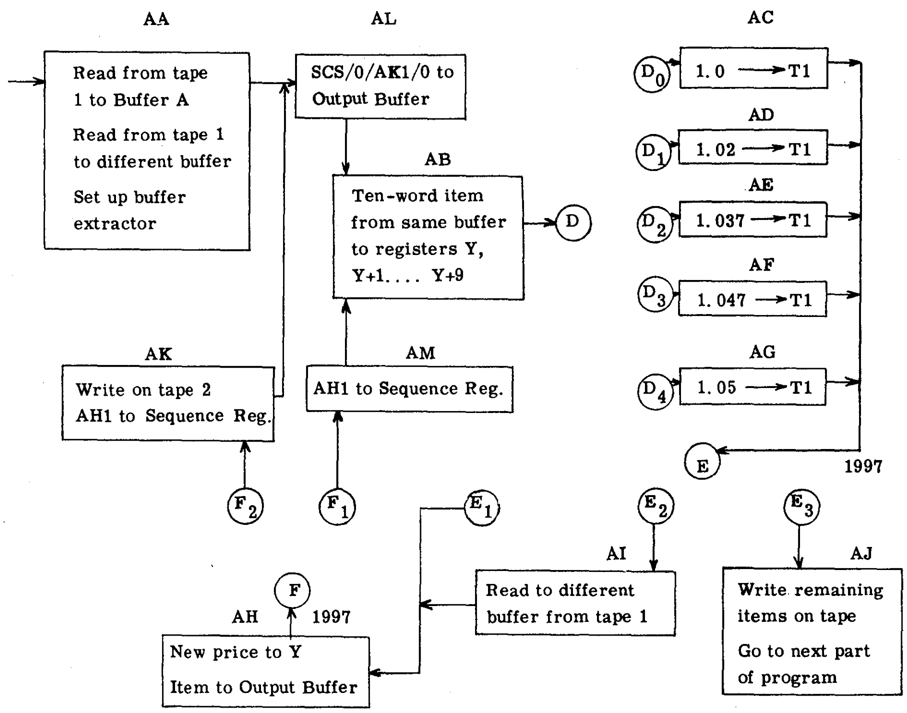
INTERNAL SELECT: ISL/A/B/C. Operation code D9. Generate a hexadecimal digit using the words at A and B. Add this digit to the C address and subsequence to C. Store the instruction in 1994. The generation of the hexadecimal digit from the words at A and B is performed in the same way as it was from the word at 1993 and the m_{th} word transferred in the Double Transfer and Select Instructions. Time: 7 word cycles.

TRANSFER AND SELECT: TSA (TSB, TSS, TSD)/A/B/C. Operation code FC. These instructions are identical to the Double Transfer and Select instructions except that step one is omitted. They are useful for picking out the operations to be performed on an item from tape, according to the value of a code digit. Time: 6+n word cycles.

NOTE: All the Transfer and Select instructions are subject to the same restrictions listed under Transfer In instructions (page 1-19 except that the A address may be special but not 1990.

Extracted C Address	014	015	019	013	015	016	000	001	009	023
Generated digit	G	G	G	G	F	F	F	B	B	7
New C Address	029	error	error	028	029	error	014	011	019	030

Figure 1-49



Storage							
Reg.	Contents	Instruction	Type	A	B	C	Effect of instruction
Y	+pp. pp0 000 00c	AA1	RFA	1	0	0	First block to Buffer A
Y+1	+0w. ww0 000 000	AA2	RFD	1	0	0	Second block to Buffer B
Y+2	+nn nnn nnn nnn	AA3	TXS	S9	1993	AL1	Extractor to 1993
.		E_1 = AH1	MUL	S2	Y	T2	1.0425 p_0, dec. pt. 3 places right of sign
.							
.		AH2	MUL	S3	Y+1	Y	.01 w, dec. pt. 3 places right
Y+9		AH3	ADD	Y	T2	Y	1.0425 p_0 + .01 w, same dec. pt.
S1	SCS/0/AK1 /0	AH4	MUL	Y	T1	Y	p rounded to nearest penny
S2	+1.0425	AH5	TXO	Y	10	1997 F	item to output
S3	+0.01	F_1 = AM1	SCS	0	AH1	AB1	
S4	+00 000 1.00000	AB1	TSS	Y	9,10	AC1	item to Y... Y+9; state code
S5	+00 000 1.02000	D_0 = AC1	TXS	S4	T1	1997 E	1.0 → T1, dec. pt. 6 places rt.
S6	+00 000 1.03700	D_1 = AD1	TXS	S5	T1	1997 E	1.02 → T1, " " " " "
S7	+00 000 1.04700	D_2 = AE1	TXS	S6	T1	1997 E	1.037 → T1, " " " " "
S8	+00 000 1.05000	D_3 = AF1	TXS	S7	T1	1997 E	1.047 → T1, " " " " "
S9	000 000 000 00G	D_4 = AG1	TXS	S8	T1	1997 E	1.05 → T1, " " " " "
S10		E_2 = AI1	RFD	1	AH1	0	Next block to other buffer
T1	1+t_s, dec. pt. 6 places right	F_2 = AK1	WFA	2	AH1	AL1	2 irrelevant words, then 6 ten-word items written on tape 2
		AL1	TXO	S1	3	AB1	Sentinel and 2 irrelevant words in Output Buffer
T2	working space	AJ1	First instruction of box AJ				

Figure 1-50

EXAMPLE: Suppose that tape one contains 10,000 ten-word items of the form:

Word	Form
1	+pp. pp0 000 00c
2	+0w. ww0 000 000
3	+nn nnn nnn nnn
4	description
5	description
6	description
7	description
8	description
9	quantity available
10	manufacturing time

where pp.pp represents four digits of price, w.ww represents three digits of weight, c is a code for the state, and nn nnn nnn nnn represents the number on hand. These items contain inventory information about devices manufactured. None of the words of the ten-word item is a sentinel.

Suppose it is desired to reprice these items by the formula,

$$p = (1 + t_s)(1.0425\, p_0 + .01\, w)$$

where t_s is the state tax, as given below, and p_0 is the present price.

It is intended to write a new file on tape two. As a signal to refill the buffers when depleted, assume that

c	0	1	2	3	4
t_s	.0	.02	.037	.047	.05

the 61st word of each block is a sentinel SCS/0/AI1/0. The second word after the last item is a sentinel SCS/0/AJ1/0, which leads to the end of the program after all of the items have been processed.

Figure 1-50 shows one program for solving this problem. After loading both buffers, the basic loop is entered. The usual path through this loop involves transferring an item to memory (AB1), using the Transfer and Select instruction to pick the proper tax (AC through AG), and repricing the item (AH). An empty buffer is signalled by the sentinel in word 61 and a full Output Buffer is signalled by the sentinel that was inserted in box AL .

BYPASS INTERLOCK: The instructions DBA, DBB, DBS, DBD, BSA, BSB, BSS, and BSD differ from DTA, DTB, DTS, DTD, TSA, TSB, TSS, and TSD in that B_p is 1, causing the instruction to be executed without waiting for the completion of any Read instruction being executed. A Read instruction must be in progress whenever one of these instructions is initiated.

TWIN TRANSFER: TTX/A/B/C. Operation code E2. Transfer the contents of the Select Instruction Register (1994) to address A and transfer the word at B to C. This instruction is specifically designed for digital sorting and the reader may see it used by referring to Section X. Time: 5 word cycles.

CONTROL INSTRUCTIONS

PASS: PSS/0/0/0. Operation code 82. Go to the Sequence Register for next instruction. Addresses are ignored. They can be arbitrary and can thus be used to store information. The use of this instruction has already been illustrated. Time: 2 word cycles.

SEQUENCE CHANGE AND SUBSEQUENCE CALL: SCS/A/B/C. Operation code 00. Put B in the Sequence Register and subsequence to C. A can be an arbitrary decimal number used for storage of information. It is not used in the instruction. Time: 6 word cycles.

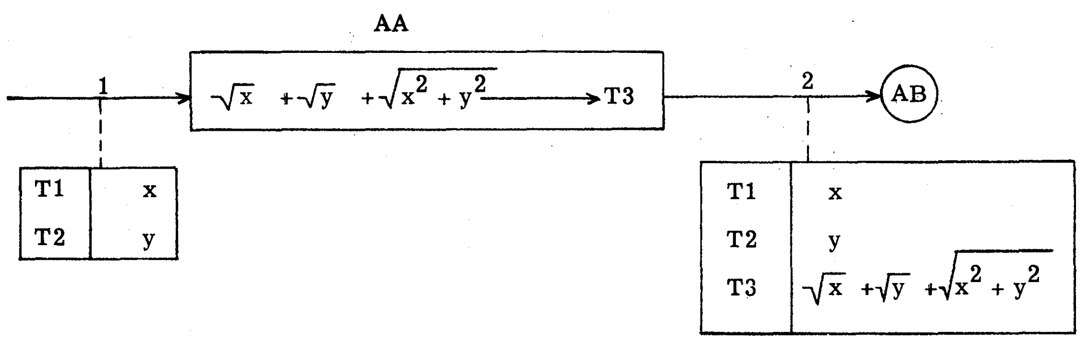
BRANCH AND RETURN: BAR/A/B/C. Operation code 42. Put a non-sentinel SCS/0/(SR)/0 instruction in register A, then put B in the Sequence Register and subsequence to C. The purpose of this instruction is to facilitate the use of subroutines. A subroutine is a set of instructions which performs a specific job often needed by many different users. Time: 7 word cycles. EXAMPLE: Suppose that starting at register R1 there is a subroutine which computes the square root of the number in register 0 and puts the answer back in register 0. Suppose that its exit is at R2; that is, after putting the answer at 0, it does the instruction at R2. R1 is called the entry to this subroutine and R2 is called its exit. A section of coding using the square root routine is given in Figure 1-51. This coding causes x+y to be put in T1 and then moved to 0. The square root subroutine is then entered and it puts $\sqrt{x+y}$ into 0. SCS/0/AB12/0, placed in R2 by the BAR instruction, causes the return to AB12 after $\sqrt{x+y}$ is found. The last instruction puts $\sqrt{x+y}$ in T3.

As a slightly more elaborate example of the BAR instruction, consider the code corresponding to the flow diagram of Figure 1-52. In this case the single BAR instruction in R3 is used three times for three different entries to and exits from the square root subroutine. Following each of the three instructions whose C address is R3, the square root subroutine is entered and puts the square root, first of x, then of y, then of $x^2 + y^2$, in register 0. Thus any process may easily be used repeat-

Storage	
Register	Contents
T1	x
T2	y

Instruction		Effect of instruction
AB9	ADD/T1/T2/T1	x+y → T1
AB10	TXS/T1/0/0	x+y → 0
AB11	BAR/R2/R1/0	SCS/0/AB12/0 → R2 (go to square root subroutine for next instruction)
AB12	TXS/0/T3/0	$\sqrt{x+y}$ → T3

Figure 1-51



Storage	
Reg.	Contents
T1	x
T2	y
T3	

Instruction		Effect of instruction
AA1	TXS/T1/0/R3	x → 0; AA2 in S.R.
AA2	TXS/0/T3/0	$\sqrt{x}$ → T3
AA3	TXS/T2/0/R3	y → 0; AA4 in S.R.
AA4	ADD/0/T3/T3	$\sqrt{x} + \sqrt{y}$ → T3
AA5	MUL/T1/T1/T4	x^2 → T4
AA6	MUL/T2/T2/0	y^2 → 0
AA7	ADD/T4/0/0	$x^2 + y^2$ → 0
AA8	SCS/0/0/R3	AA9 in S.R.
AA9	ADD/T3/0/T3	$\sqrt{x^2 + y^2} + \sqrt{x} + \sqrt{y}$ → T3
AB1		
R3	BAR/R2/R1/0	SCS/0/AA2/0 → R2; enter $\sqrt{sr}$
		SCS/0/AA4/0 → R2; enter $\sqrt{sr}$
		SCS/0/AA9/0 → R2; enter $\sqrt{sr}$

Figure 1-52

Digit	D_b	B_h				B_t				B_u			
	x	x	x	x	x	x	x	x	x	x	x	x	x
Name	Not used	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_g	B_s
Function	Arbitrary	Correspond to 10 breakpoint switches on console											Stop if one

Figure 1-53

edly by means of the BAR instruction. It is again assumed that the entry and exit of the square root subroutine are at R1 and R2, respectively.

OPTIONAL STOP: OST/A/B/C. Operation code G6. The A address is an arbitrary decimal number not used in the execution of the instruction. The B address, in conjunction with certain switches, determines whether the machine will stop. Subsequence to C. Time: 5 word cycles. The bits of the B address may be designated as in Figure 1-53.

The Console contains a Conditional Stop switch which has three positions: Stop, Compare, and Proceed. It also has ten breakpoint switches, which have the two positions "On" and "Off"

1. If B_s is 1, the machine stops regardless of the switches or other bits.
2. If B_s is 0 and B_g is 1, the machine goes regardless of the switches and other bits.
3. If B_s and B_g are 0 and the Conditional Stop switch is set to "Stop", the machine will stop regardless of the breakpoint switches and B_1 to B_{10}.
4. If B_s and B_g are 0 and the Conditional Stop switch is set to "Proceed", the machine will go regardless of the breakpoint switches and B_1 to B_{10}.
5. If B_s and B_g are 0 and the Conditional Stop switch is set to "Compare", any coincidence of a 1 among B_1 to B_{10} and an "On" of the corresponding breakpoint switch will stop the machine.

The flexibility of this instruction gives it many uses. For example, OST instructions having B_1 equal to 1 might be introduced at places in the program where it would be handy to stop during program testing. OST instructions having B_2 equal to 1 might be introduced at places in the program where the run could later be interrupted to introduce new information or to check the size of some number. Also, OST instructions having B_3 equal to 1 could be introduced at places to interrupt the program in order that someone else might use the machine. Then, if breakpoint switch one is "on" and the others "off", the machine would only stop at the testing stops, whereas, if breakpoint switch two is "on" and the others "off", the machine would no longer stop at the testing stops but would stop to allow the introduction of new information, etc.

PRINT INSTRUCTIONS

PRINT NUMERIC: PRN/A/B/C. Operation code 21. Print the word at A on the Console Typewriter in numeric form; put B in the Sequence Register and subsequence to C. Time till start of next instruction: 6 word cycles.

PRINT ALPHABETIC: PRA/A/B/C. Operation code 11. Print the word at A on the Console Typewriter in alphabetic form; put B in the Sequence Register and subsequence to C. Time till start of next instruction: 6 word cycles.

These instructions will be used infrequently because the time required to print one word varies from two to five seconds. Specific times will be discussed in Section V in the description of the Console Typewriter. The transmission from address A to the Console is unchecked. Thus, the Print instruction can be used if an input word is wrong and the CHECK SUBSEQUENCE switch is "on". Any other use of the erroneous word in memory stops the machine. The weight count digit is printed for visual comparison.

ADDRESSES OF SIGNIFICANCE

Of the 2000 addresses in the DATAmatic 1000, ten are special addresses which contain words for control purposes. Of the remaining 1990 addresses, ten (0000 and 1981 to 1989) may be used for special control purposes and may be selected automatically as instruction sources under certain circumstances. Some of the possible control uses of these registers are described in the following discussion.

ADDRESSES OF SIGNIFICANCE

0000 One of the ways to start the DATAmatic 1000 (see Sect. V) is to push the Special Start Transfer Control button. This causes the Sequence Register to be set to zero (0000) and causes the first instruction to be selected from this location.

1981 During the editing pass, for data recorded on tape by the Input Converter, the control word of each blockette of information (word 16 or 32) will be examined by means of a subsequence call to the Sentinel Register. If an illegal procedure occurred during conversion, the control word will be a sentinel SCS/0/0/1981. Consequently, register 1981 should contain the first instruction of a routine to do something about the illegal procedure.

1982 Another way to start the DATAmatic 1000 (see Sect. V) is to push the Special Start Subsequence button. This causes the computer to subsequence to 1982 without altering the Sequence Register. 1982 may contain the first instruction of an appropriate routine. For example, 1982 may be used for an automatic rerun procedure making it possible to carry on immediately if a random error should cause the machine to stop.

1983 Illegal punches, present during the input conversion process, will produce a sentinel instruction: SCS/0/0/1983 in the control word of the blockette (see Sect. II).

1984 Fillers, transferred from the Output Buffer, will be sentinel instructions: SCS/0/0/1984.

1985 Fillers, transferred from the Input Buffer, will be sentinel instructions: SCS/0/0/1985.

1986 Division overflow results in an automatic subsequence call to 1986.

1987 If the CHECK SUBSEQUENCE switch is "on" and a weight count error is detected in a transfer of data from an input buffer to the High-Speed Memory, a subsequence call to 1987 is produced.

1988 Addition or subtraction overflow results in an automatic subsequence call to 1988.

1989 Reaching the end of tape results in an automatic subsequence call to 1989.

The address 0000 is of further interest. Except in the case of Comparison instructions, an instruction to change the Sequence Register or make a subsequence call to address 0000 (void address) will be ignored and the Sequence Register will be used in the normal fashion.

Storage addresses 1990 through 1999, located in the Arithmetic and Control Units, are special addresses which contain words used for control purposes. They can be read into and out of by instructions.

1990—CONTROL REGISTER: The Control Register is part of the DATAmatic 1000's control circuitry. It stores each instruction during the time the instruction is being performed by the system.

The primary use of this Control Register in programming is as a C address for instruction modification purposes. Modifications to an instruction are usually made by means of an Addition instruction which adds a constant to the instruction being modified. Because the Addition instruction does not provide for a subsequence call, the program would ordinarily continue the current instruction sequence after an instruction modification. However, since the Control Register is addressable, it is possible to modify an Addition instruction and thus to make an immediate subsequence call to 1990. This is accomplished by sending the result of the Add instruction modification to the Control Register (i.e., the C address of the Add instruction is 1990). This will result in an automatic subsequence call to 1990, so that the modified instruction is the next one executed.

The address 1990 may be used as the A, B, or C address of many instructions. (However, do not transfer internally to 1990 or use 1990 as the B address of any TXS instruction. Furthermore, 1990 may not be used as the C address in any Comparison instruction nor should one ever subsequence to 1990.) Because the address portions of an instruction are shifted end-around during its execution, the programmer must know the relative positions of the A, B, and C addresses of the word in the Control Register if this register is used as a word source. If 1990 is used as an A address, the instruction in the Control Register is in normal (unshifted) form. If 1990 is used as the B address, the address portions of the instruction have been shifted so that they are in the revised sequence BCA when the actual selection is made. In general, the programmer should not address 1990 without taking special precautions.

1992—OUTPUT BUFFER REGISTER: The Output Buffer Register contains the same word as that found in the first word position of the Output Buffer when the Output Buffer is full. Thus, when 62 words have been sent to the Output Buffer, this register (1992) will contain the first word which was transferred. Then if another word is sent to the Output Buffer, the first word is pushed out and replaced by the second word which was transferred, and so forth. The word in this register is then the same as the first word which will be written on tape by the next Write instruction. When the Output Buffer is not full, i.e., contains less than 62 words, the Output Buffer Register still contains the word that previously occupied the final word position of the Output Buffer. This is the word which was written into the key channel on tape by the previous Write instruction. With the exception of the Transfer Out instruction, information may not be sent to 1992.

1993—EXTRACTOR REGISTER: This register contains the constant used for the extraction that is performed in the two Transfer and Select instructions. Whenever this register is used during the execution of several of the arithmetic operations, its contents, prior to the operation, are temporarily stored elsewhere and are then restored after the completion of the operation. It may be used as the A, B, or C address of any instruction. If 1993 is used as the C address, it will first be cleared; then the instruction will be performed normally. If 1993 is used as the C address of a Substitute instruction, (B) will be transferred to 1993 prior to the substitution, which will then proceed normally.

1994—SELECT INSTRUCTION REGISTER: Both Transfer and Select instructions and the Internal Select instruction are stored in this register during and after their execution. Four times the multiplicand is stored in this register as a result of the Multiply instruction and the previous contents are thereby destroyed. This register may be the A, B, or C address of all instructions except the Transfer and Select or Multiply instructions.

1995—REMAINDER REGISTER: This register receives the low-order product of a multiplication after it has been modified by rounding. The low-order product is made up of the eleven low-order decimal digits of the complete product. Rounding is accomplished by adding five to the high-order digit of the low-order portion of the product and allowing any end carry to be added to the low-order digit of the high-order portion of the product. This register also receives the remainder after the execution of a Divide instruction. It may be used as the A, B, or C address of an instruction.

1997—SENTINEL REGISTER: Any instruction which implements a transfer between the High-Speed Memory and a buffer will also implement the transfer of a word to the Sentinel Register. If the Transfer instruction senses a sentinel, then the first sentinel sensed by the instruction is stored in the Sentinel Register; if no sentinel is sensed, then a Pass instruction is stored in the Sentinel Register.

The Transfer instructions in question are Transfer In, Transfer Out, Transfer and Select, Double Transfer and Select, First Key Comparison, and Second Key Comparison. In the case of Transfer In, Transfer and Select, and Double Transfer and Select, all words transferred in and also the next word in the Input Buffer are examined for sentinels. In the case of the Key Comparison instructions, the sentinel sensing is performed on the word at address A, that is, on the word that has been transferred. In the case of Transfer Out, the overflow words are sensed for sentinels. Notice that output words are not sensed for sentinels in the Double Transfer and Select instruction.

This register also plays a special role in multiplication and contains seven times the multiplicand upon completion of a Multiplication instruction.

1999—CURRENT INSTRUCTION REGISTER: DATAmatic 1000 instructions are processed in eight word cycles. These eight cycles are executed starting with cycles 2 through 8 and ending with an optional cycle 1. The Current Instruction Register is used for storing a number of different items of information related to these eight word cycles.

The Central Processor will have completed an instruction at the conclusion of cycle 8 if the instruction specifies a subsequence or at the conclusion of cycle 1 if the instruction is normal. In the first case, register 1999 contains the instruction just executed. In the second case, register 1999 contains the address of the next instruction to be executed. In either case, the contents of 1999 show the source of the instruction currently in the Control Register. Thus register 1999 may be used as a diagnostic aid when the Central Processor stops.

When the Current Instruction Register contains the address of the instruction to be executed, the subaddress portion of this address is located in the A subaddress position. However, the bank designator for this address resides in the B bank designator position. All other bit positions are blank. When the Current Instruction Register contains the instruction just executed, the address positions of the instruction are in the order CAB rather than the normal ABC. Thus, the subaddress of the instruction being executed always resides in the same bit positions (A subaddress), whether it is a programmed C address or a normal instruction sequence address.

SUMMARY OF INSTRUCTIONS

Figure 1-54 summarizes the instructions in a compact form. It omits some fine points of each instruction but is a useful reference to the main features of each instruction. The word "order" is frequently used as a synonym for instruction in data-processing terminology and it has been used in Figures 1-54 and 1-55 with this meaning. The column in Figure 1-54 labelled "word type" refers to the four types of order words shown in Figure 1-55.

SUMMARY OF INSTRUCTIONS

Order	Symbol	Operation Code	A	B	C	Word Type	Effect of Order	Word‡ Cycles
CALCULATING ORDERS								
Add	ADD	FF				1	(A) + (B) → C If overflow, subsequence to 1988	8
Subtract	SUB	GG				1	(A) − (B) → C See ADD	8
Multiply	MUL	FE				1	(A) × (B) → C High-order 11 digits (rounded). Low-order 11 digits with 5 added to high-order position → 1995.	35
Divide	DIV	EE				1	(B) ÷ (A) → C If overflow, subsequence to 1986 Remainder → 1995	31-130
Shift left, Preserving sign	SLP	5G				4	Shift (A) N decimal places left and send to C. Sign does not move, N = Bh	7 + N
Shift right, Preserving sign	SRP	6G				4	Same as SLP, except for direction	7 + N
Shift left, Word	SLW	4F				4	Shift (A) 4Bh . Bt binary places left and send to C. Whole word moves.	7 + N(†)
Shift right, Word	SRW	7F				4	Same as SLW, except for direction	7 + N(†)
Substitute	SST	GD				1	Replace (C) by (A) as specified by (B)	7
TRANSFER ORDERS**								
Transfer In	TI(A,B,S,D)	Bp = 0 G3			S	2	A = 01, B = 10, S = 00, D = 11 } Br — If sentinel is sensed store in 1997	4 + n
Transfer In bypassing interlock	TB(A,B,S,D)	Bp = 1 G3			S	2		4 + n
Double Transfer and Select	DT(A,B,S,D)	Bp = 0 GB			S	2		6 + n
Double Transfer and Select, bypassing interlock	DB(A,B,S,D)	Bp = 1 GB			S	2		6 + n
Transfer and Select	TS(A,B,S,D)	Bp = 0 FC			S	2		6 + n
Transfer and Select, bypassing interlock	BS(A,B,S,D)	Bp = 1 FC			S	2		6 + n
Transfer Out	TXO	G0			S	2		4 + n
Transfer Internally	TXI	E1				3	Transfer n words from A to B. n = lowest 5 bits of C	3 + 2n
Transfer and Subsequence Call	TXS	D0			S	1	Transfer one word from A to B	5
Twin Transfer	TTX	E2				1	(SOR) → A and (B) → C	5
Internal Select	ISL	D9			S	1	Subsequence Call to location (C) + d, where d is digit formed by extracting (A) against (B). This order stores itself in 1994.	6
TAPE ORDERS								
Read Forward	RF (A,B,D,K)	36		Seq.	S	1	(Da.Ah) = 0,2 A 0,8 B 0,0 D 0,6 K — AtAu specifies tape mechanism	7
Read Backward	RB (A,B,D,K)	24		Seq.	S	1		7
Search Forward	SF (R,W)	36		Seq.	S	1	(Da.Ah) = 1,2 R 1,4 W — AtAu specifies tape mechanism	7
Search Backward	SB (R,W)	24		Seq.	S	1		7
Write Forward	WF (A,P)	17		Seq.	S	1	Ah = 0 for WFA, 2 for WFP	7
Rewind Tape	REW	05		Seq.	S	1	(Da,Ah) = 0,0	7
Switch Tape to First Half	SWF	05		Seq.	S	1	Position tape unit At Au to first half (Da,Ah) = 0,2	7
Switch Tape to Second Half	SWS	05		Seq.	S	1	Position tape unit At Au to second half (Da,Ah) = 0,8	7
DECISION ORDERS								
Less than Comparison, Numeric	LCN	3D				1	If (A) ≤ (B), change (SR) to "C"	7 or 6*
Less than Comparison, Alphabetic	LCA	2E				1	If (A) ≤ (B), change (SR) to "C"	7 or 6*
Inequality Comparison, Numeric	ICN	0D				1	If (A) not equal to (B), change (SR) to "C"	7 or 6*
Inequality Comparison, Alphabetic	ICA	1E				1	If (A) not equal to (B), change (SR) to "C"	7 or 6*
First Key Comparison	FKC	78				1	Au ≡ input buffer A channel — If (B) ≤ (A), go to C	8 or 7*
Second Key Comparison	SKC	69				1	Au ≡ input buffer A channel — If (B) ≥ (A), go to C	8 or 7*
CONTROL ORDERS								
Sequence Change	SCS	00		Seq.	S	1	Changes Sequence Register to B (unless B = 0) Causes next order performed to be a subsequence call to C (unless C = 0)	6
Branch and Return	BAR	42		Seq.	S	1	Stores "SCS000(SR)000" at A	7
Pass	PSS	82				1	Consults Sequence Register for next order to be performed	2
Optional Stop	OST	G6		Con.	S	1	Programmed stop	5
PRINT ORDERS								
Print Alphabetic	PRA	11		Seq.	S	1	Print (A) on console typewriter	6
Print Numeric	PRN	21		Seq.	S	1	Print (A) on console typewriter	6

SIGNIFICANT ADDRESSES

1981	Input Converter Error Subsequence
1982	Special Start Subsequence
1983	Illegal Punch Subsequence
1984	Output Buffer Filler Subsequence
1985	Input Buffer Filler Subsequence
1986	Division Overcapacity Subsequence
1987	Input Error Subsequence
1988	Accumulator Overflow
1989	End of Tape
1990	Control Register
1992	Output Buffer Register
1993	Extractor Register
1994	Select Order Register
1995	Remainder Register
1997	Sentinel Register
1999	Current Order Register

NOTES

CON.	Digits used to define order in addition to operation code
SEQ.	Change (SR) to B. If not used, subtract one word cycle
S	Subsequence call to C. If used, subtract one word cycle
SR	Sequence Register
SOR	Select Order Register
(A)	Contents of Address A
(B)	Contents of Address B
Bh	Hundreds digit of B address
Bt	Tens digit of B address
(C)	Contents of Address C
A	Da, Ah, At, Au
n	Number of words to be transferred
N	Number of four-bit positions to be shifted
m	Number of word used for extraction
*	Applies if sequence is changed
(†)	Rounded to next higher integer
‡	One word cycle is equivalent to 28.8 micro-seconds
**	"B" Address of Buffer Transfer Orders

50, 28,27,26,25	24	23,22	21,20,19,18,17
$B_1 = n - m$	Bp	Br	$B_2 = n$

Fig. 1-54

THE DATAMATIC WORD

BIT POSITION	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
TRANSFER WEIGHT	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1

ORDER WORD NORMAL (TYPE 1)		memory designator			OPERATION CODE	ADDRESS A			ADDRESS B			ADDRESS C			WEIGHT COUNT
	D_s	D_a	D_b	D_c	O_P	hundreds digits	tens digits	units digits	hundreds digits	tens digits	units digits	hundreds digits	tens digits	units digits	W

ORDER WORD BUFFER TRANSFER (TYPE 2)					OPERATION CODE	ADDRESS A							ADDRESS C			WEIGHT COUNT
	D_s	D_a	B_1	D_c	O_P	hundreds digits	tens digits	units digits	B_1	B_p	B_r	B_2	hundreds digits	tens digits	units digits	W

ORDER WORD INTERNAL TRANSFER (TYPE 3)		memory designator			OPERATION CODE	ADDRESS A			ADDRESS B					WEIGHT COUNT
	D_s	D_a	D_b	*	O_P	hundreds digits	tens digits	units digits	hundreds digits	tens digits	units digits	*	C_2	W

ORDER WORD SHIFT (TYPE 4)		memory designator			OPERATION CODE	ADDRESS A						ADDRESS C			WEIGHT COUNT
	D_s	D_a	*	D_c	O_P	hundreds digits	tens digits	units digits	B_h	B_t	*	hundreds digits	tens digits	units digits	W

NUMERIC WORD	SIGN 1 0 1 or 12TH DIGIT	11th DIGIT	10th DIGIT	9th DIGIT	8th DIGIT	7th DIGIT	6th DIGIT	5th DIGIT	4th DIGIT	3rd DIGIT	2nd DIGIT	1st DIGIT	WEIGHT COUNT W

ALPHANUMERIC WORD	8th CHARACTER (most significant)	7th CHARACTER	6th CHARACTER	5th CHARACTER	4th CHARACTER	3rd CHARACTER	2nd CHARACTER	1st CHARACTER (least significant)	WEIGHT COUNT W

LEGEND

- B_h, B_t Determines Number of Binary Places to be Shifted
- B_p Determines if Interlock is to be Bypassed
- B_r Determines Buffer Involved
- B_1 Determines Word Inspected
- B_2 Determines Number of Words Transferred
- C_2 Determines Number of Words Transferred
- D_a The Memory Designator (Thousands Digit) of Address A
- D_b The Memory Designator (Thousands Digit) of Address B
- D_c The Memory Designator (Thousands Digit) of Address C
- D_s The Sign of the Order
- * Irrelevant

Fig. 1-55

SECTION II

MODEL 1200 CARD INPUT SYSTEM

SECTION II—MODEL 1200 CARD INPUT SYSTEM

INTRODUCTION

The DATAmatic Model 1200 Card Input System, consisting of a Card Reader and an Input Converter, reads information from standard 80-column punched cards and converts it to a binary representation for writing on magnetic tape. This operation is "off-line", i.e., completely separate from the Central Processor. One Magnetic File Unit is assigned to write original data converted by the Input Converter. This unit may be switched to communicate with the Central Processor when it is not writing.

The conversion from cards to tape (see Figure 2-1) may be thought of as divided into three steps:

(1) Reading cards by means of the Card Reader under control of the Card Reading Control Panel;

(2) Converting the card information under control of the Conversion Control Panel to a magnetic tape representation and composing a 16-word record; and

(3) Writing this record on tape.

These operations are automatically checked within the Card Reader or during subsequent tape operations. Each punched card is converted to one 16-word record in an array of magnetic cores called the Tape Data Storage section. This record includes 14 words derived from the card itself, one word of identification, and one check word. Two such records, called "blockettes", are written on each tape block. The rest of the block is unchanged. During input conversion, considerable editing may be performed under control of the Card Reading and Conversion Control Panels.

CONTROL PANELS: These removable panels are wired to control the operation of various equipment units. The wiring is planned by the programmer as a part of his program preparation. A control panel contains a number of connection points called hubs. These include exit hubs, which emit an electrical signal, and entry hubs, which can receive such a signal. The wires, which have a spring-type connector at either end, are installed with one end inserted in an exit hub and the other in an entry hub. They may be easily removed for wiring changes or, if desired, they may be inserted together with a special sleeve which effectively locks the connection. These more permanent connections require a special tool for removal and are thus safer for long-term use after the final control panel wiring has evolved. The diversity of possible connections which may be made on a control panel is a major source of flexibility in the equipment controlled.

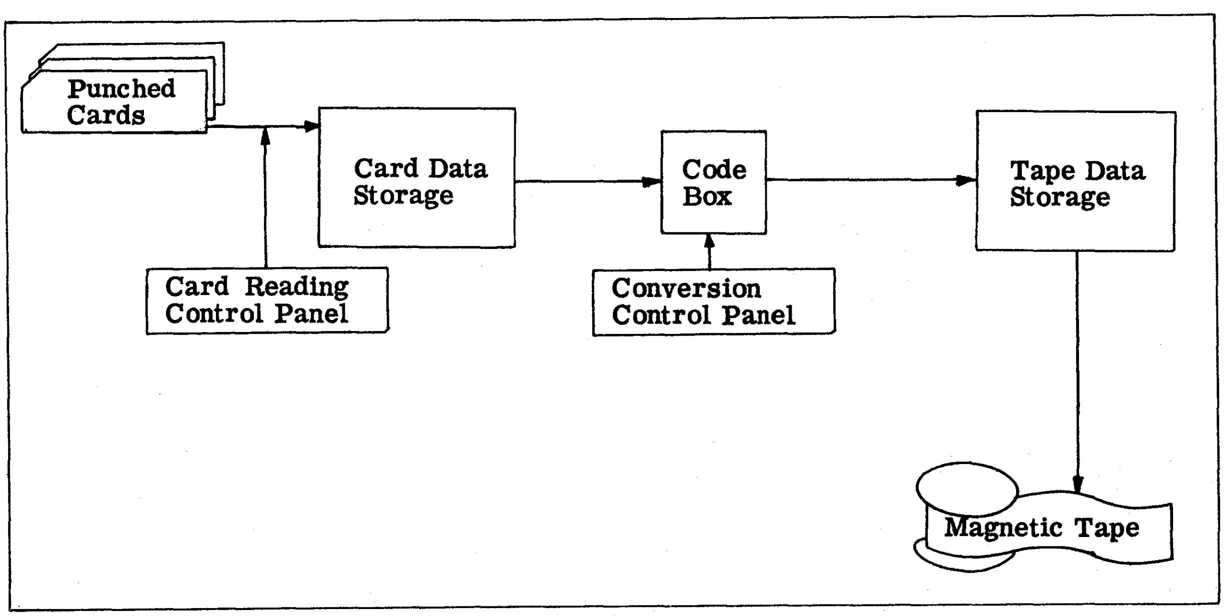


Figure 2-1

CARD READER AND CARD READING CONTROL PANEL

The Card Reader operates at the rate of nine hundred 80-column cards per minute. Over 3000 cards may be stacked in the input tray. Each card is read at two reading stations and the two readings are compared. If the two readings are not identical, the converter will follow the course of action directed by certain switch settings (see page 2-7). After the second reading, the card is passed to a stacker and the information from the card is sent to the Card Data Storage section in accordance with the wiring of the Card Reading Control Panel. Under control of this panel, any or all of the 80 card columns may be selected for transmission to card data storage, where up to 100 columns of punched-card data are stored. Enough columns may be duplicated to furnish 100 columns of information to card data storage; however, triplication of columns is not permitted. Note that all transposition, duplication, and discarding of information must be done with the Card Reading Control Panel.

During the planning for input conversion, a tape format must be designed to specify the location in the blockette of each character converted. The first 14 words in the blockette may contain converted information. To promote understanding of the Card Reading Control Panel, it is helpful to visualize these 14 words in one row with the high-order bits of word 1 at the left and the low-order bits of word 14 at the right, as shown in Figure 2-2. The Card Reading Control Panel is then wired according to this array, working from right to left. Spaces to contain zero-bit fillers may be ignored while wiring this control panel. The rightmost **converted** character in the above array is wired first by connecting the corresponding card column to storage column 100. Working to the left, the next converted character is wired to storage column 99, etc., until the leftmost converted character in the array has been wired.

An example, which illustrates the method of wiring the Card Reading Control Panel, including transposition and duplicating card information, is shown in Figure 2-3. Note that this control panel contains 80 exit hubs labelled Reading Station 1 and 80 exit hubs labelled Reading Station 2. The sections labelled Converter Entry 1 and Converter Entry 2 each contain 100 commoned entry hubs (those joined by solid lines are connected internally). Each set of exit hubs sends a set of signals from one of the card reading stations, representing the information punched on the card. The control panel wiring relays these signals via the Converter Entry hubs to the desired locations in card data storage. All wiring on the upper half of this panel must be duplicated on the lower half. In the example of Figure 2-3, card column 80 is connected to storage column 100; card column 79 is discarded; card columns 78-75 are connected to storage columns 99-96, with card column 78 also duplicated in storage column 95; card column 71 is transposed to storage column 94; and card columns 74-72 are connected to storage columns 93-91. Only those columns are wired which contain information to be converted, however few they may be.

WORD	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
BITS	**52 -- 5**	**52 -- 5**	**etc.**											**52 -- 5**

Figure 2-2

CONVERSION CONTROL PANEL

This control panel performs the following two functions:

(1) Determining which conversion rule will be performed on each column of information in card data storage. For example, a decimal digit can be converted to four bits or to six bits on tape, as shown in Figure 2-4. The conversion rule desired is specified by appropriate wiring of the Conversion Control Panel.

(2) Supplying zeros to fill in the unused parts of the blockette on tape.

There are 13 different rules by which a card column can be converted and four different rules by which groups of zeros can be inserted as desired (see Table 2-1, page 2-13). For example, an X-punch on a card can be transmitted to tape as:

100000	by using conversion rule 1, 2, or 3,
0000	by using conversion rule 7 or 8,
0101	by using conversion rule 10,
1101	by using conversion rule 11,
0001 0001	by using conversion rule 12, or
000000000010	by using conversion rule 13.

DATAmatic

Layout Form — Input Converter, Model 1200

(CARD READING CONTROL PANEL)

Title..

Prepared by...For Program..

By Programmer...Checked by..

Date...Remarks..

....

Modification...Page.................of..........

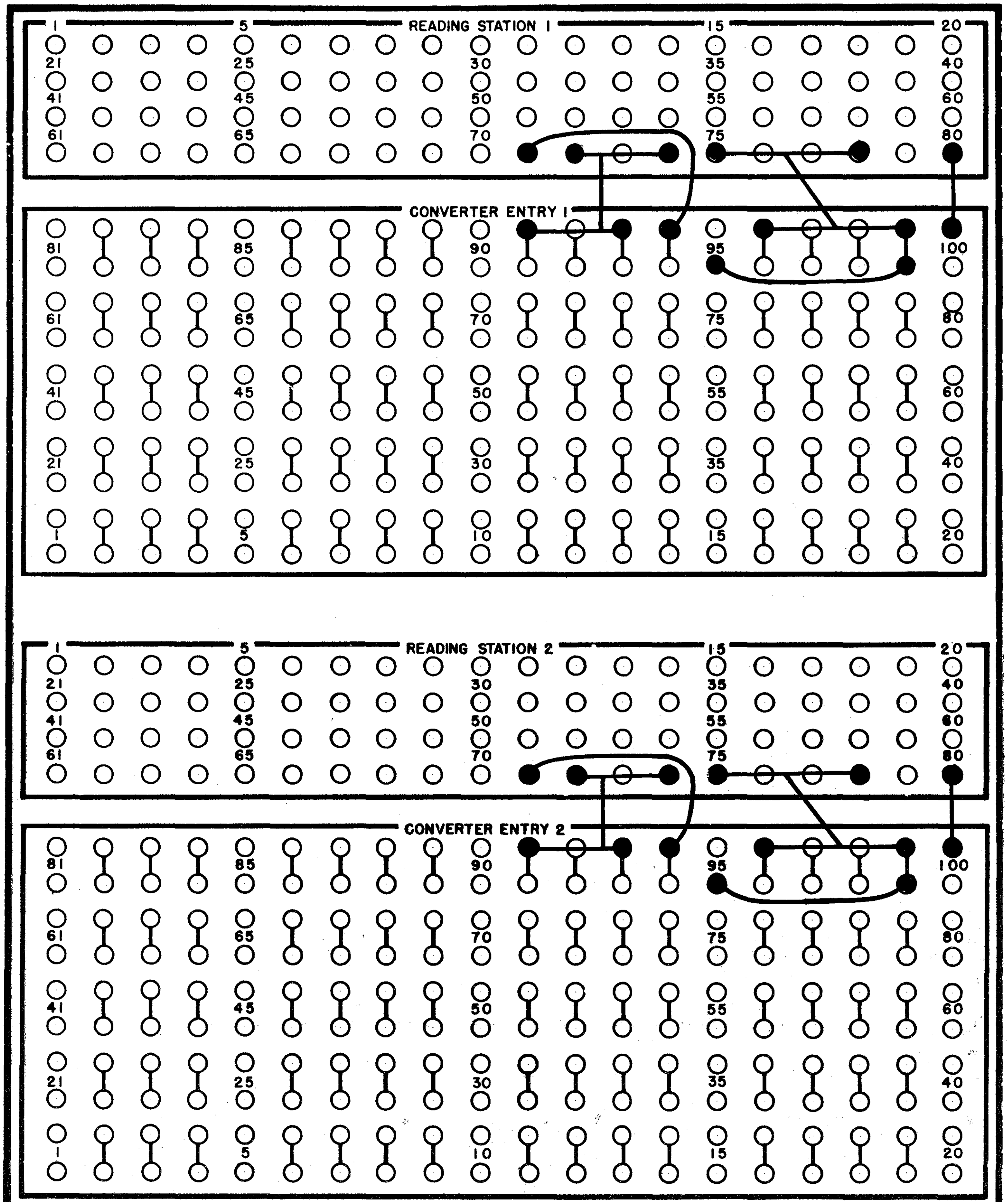


Figure 2-3

BINARY 4-BIT and 6-BIT CODES

Decimal Digit	Binary Notation	Alphanumeric Character	Binary Notation
0	0000	0	000000
1	0001	1	000001
2	0010	2	000010
3	0011	.	.
4	0100	.	.
5	0101	A	010001
6	0110	B	010010
7	0111	C	010011
8	1000	.	.
9	1001	.	.

Figure 2-4

Each card column may be converted by a different conversion rule, provided that it contains no punch configurations which are illegal under the rule selected (see Table 2-2, page 2-15). Between successive converted card columns, zero-bit fillers may be introduced as required by the tape format. Each conversion rule hub may be used up to 34 times in converting the information from one card. Sets of rules hubs which are connected in common may be used up to 34 times per set on one card. The storage columns are connected sequentially, starting with column 100, then column 99, etc., to the lowest-numbered storage column in use. The information in card data storage must be converted in sequence and no columns between the first and last columns converted may be omitted. This emphasizes the earlier statement that all transposition, duplication, and discarding of card information must be done with the Card Reading Control Panel.

Each punched card is converted to 14 words on magnetic tape, although some of these words may contain no converted information. The Conversion Control Panel, by applying the conversion rules and inserting zero-bit fillers, must account for 14 x 48 bits to be transmitted to tape. These bits are filled in successively, starting with the rightmost (low-order) bit of the 14th word and working toward the leftmost (high-order) bit of the first word (see Figure 2-2). If the required number of bits are not transmitted to tape, the converter will signal this fact.

In the example of Figure 2-5, rule 1 converts the information in storage columns 100 and 99 to 6-bit alphanumeric characters in steps 1 and 2; rule 17 transmits two groups of 12 binary zeros each in steps 3 and 4; and rule 4 converts the information in storage columns 98, 97, and 96 to 4-bit decimal digits in steps 5, 6, and 7 for a total of 48 bits. Word 14 of the blockette might then look as follows:

```
  7    3    9                                         A      C
0111 0011 1001 000000000000 000000000000 010001 010011
```

Each word must be prepared by an integral number of steps. In other words, no step may supply the bits to complete one word and start the following word.

WEIGHT COUNT GENERATION

Forty-eight of the 52 bits in a DATAmatic word (bits 52 through 5) are used to represent stored information. Bits 4 through 1 constitute a check digit called the weight count of the word. This weight count is originally generated at the time that the word is first assembled by the Input Converter and it remains an integral part of the word throughout the DATAmatic 1000 System. The details of the generation of the weight count are described in Section I, page 1-1.

IDENTIFICATION AND CHECKING

IDENTIFICATION: The 15th word of each blockette contains a fixed control number inserted by means of the eight RUN DESIGNATOR DIGITS switches on the converter console panel (see Figure 2-6). Each of these switches can be set to supply a decimal digit in one of the eight low-order digit positions of the control number. The remainder of this control number will be automatically generated as a plus sign and three decimal zeros. The weight count of the control number is automatically computed and attached to the 15th word by the converter. This control number may be assigned to the batch of cards being processed and may thus be used as a cross reference.

```
                      RUN DESIGNATOR DIGITS switches
                      8  7  6  5  4  3  2  1
                      ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓
15th word: +  0  0  0 X  X  X  X  X  X  X  X
```

DATAmatic Layout Form — Input Converter, Model 1200

(CONVERSION CONTROL PANEL)

Title..

Prepared by...For Program...

By Programmer...Checked by...

Date...Remarks...

....

Modification..Page...............of..........

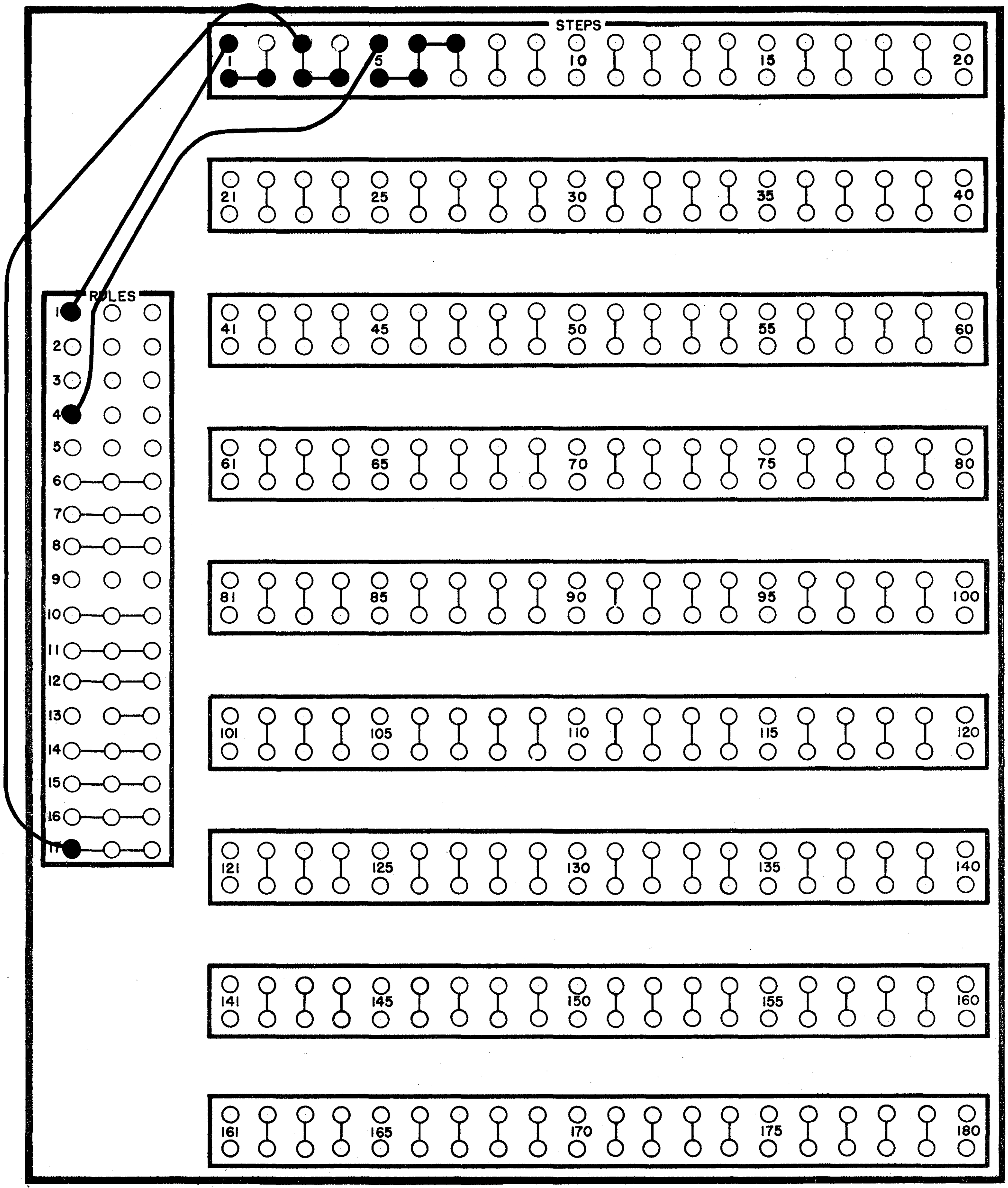


Figure 2-5

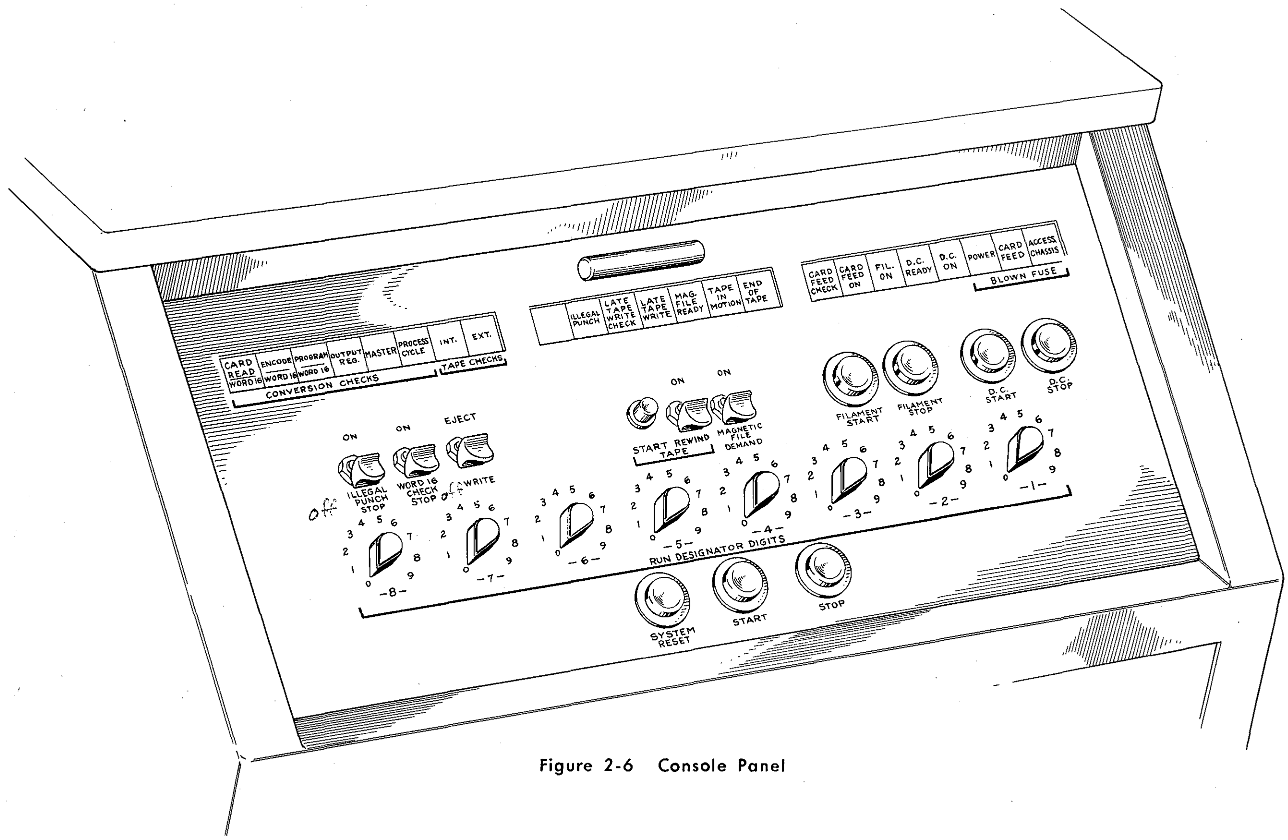


Figure 2-6 Console Panel

CHECKING: The 16th word of each blockette is a control word. Its contents are determined by programming and card-punching conditions, in conjunction with the positions of the ILLEGAL PUNCH STOP, WORD 16 CHECK STOP, and WRITE-EJECT switches on the console panel. Figure 2-7 is a summary of the course of action resulting from any combination of these conditions and switch settings.

One of three conditions will pertain to the conversion of every card: the card is correct, the card was illegally punched, or some other situation exists (improper control panel wiring, card mutilated, card punches not registered, etc.). The converter senses for these conditions and detecting one or more of them, takes the action specified in Figure 2-7. It should be noted that the WRITE-EJECT switch indicates a course of action only if the converter has been directed to proceed for a given condition. If the converter has been directed to stop for this condition, the position of the WRITE-EJECT switch is irrelevant; the converter will stop and eject the card concerned into the reject hopper without writing on tape. Because of the card-reading speed of the converter, a stop will cause ejection not only of the card concerned but also of the two following cards. These cards will not have been read and they must be placed back in the read hopper before the converter is restarted. When a stop occurs, a light will signal the condition which has caused the stop.

If the ILLEGAL PUNCH STOP switch is set to OFF and the WRITE-EJECT switch to WRITE, all blockettes in which an illegal punch is detected will have a sentinel SCS/0/0/1983 generated and stored in the 16th word. If the WORD 16 CHECK STOP switch is set to OFF and the WRITE-EJECT switch to WRITE, all blockettes in which some other illegal condition is detected will have a sentinel SCS/0/0/1981 stored in the 16th word. If both conditions are detected on the same card, the Word 16 Check control word takes precedence. All other blockettes will have a sentinel PSS/0/0/0 instruction stored in the 16th word.

A section of unusable tape up to three blocks in length will be passed over and writing will continue on the next block of good tape with no loss of time. Should the unusable tape exceed three blocks, the converter will write the last converted blockette in the first good block encountered, then stop. Three cards following the last card converted will be ejected. These cards may either be returned to the feed hopper before restarting or accumulated for conversion at the end of the run. To continue normal operation, press the START button. Do not press SYSTEM RESET to restart in this situation.

SWITCHES			CONDITION AND RESULT		
ILLEGAL PUNCH STOP	WORD 16 CHECK STOP	WRITE-EJECT	ILLEGAL PUNCH	WORD 16 CHECK	BOTH
OFF	OFF	WRITE	SCS/0/0/1983 in word 16	SCS/0/0/1981 in word 16	SCS/0/0/1981 in word 16
OFF	OFF	EJECT	Eject 1 card	Eject 1 card	Eject 1 card
ON	OFF	WRITE	Eject 3 cards Stop card feed	SCS/0/0/1981 in word 16	Eject 3 cards Stop card feed
ON	OFF	EJECT	Eject 3 cards Stop card feed	Eject 1 card	Eject 3 cards Stop card feed
OFF	ON	WRITE	SCS/0/0/1983 in word 16	Eject 3 cards Stop card feed	Eject 3 cards Stop card feed
OFF	ON	EJECT	Eject 1 card	Eject 3 cards Stop card feed	Eject 3 cards Stop card feed
ON	ON	WRITE	Eject 3 cards Stop card feed	Eject 3 cards Stop card feed	Eject 3 cards Stop card feed
ON	ON	EJECT	Eject 3 cards Stop card feed	Eject 3 cards Stop card feed	Eject 3 cards Stop card feed

Figure 2-7

APPLICATION

Assume that a batch of punched cards contains information about transactions (for example, new insurance policies) as shown in Figure 2-8. The pertinent information is to be transmitted in 14 words to magnetic tape. Therefore, a tape format must be selected (see Figure 2-9).

The desired flow of information through the Input Converter is shown in Figure 2-10. Columns are selected from the punched card by means of the Card Reading Control Panel. To wire this control panel, examine the tape format and begin by wiring the card column corresponding to the last character on the tape format to Converter Entry hub 100. The card column corresponding to the next to last character is wired to Converter Entry hub 99, etc. Spaces to be filled with zeros in the tape format are ignored in wiring this control panel. It is not necessary to wire every reading station hub or every converter entry hub, but there must be no converter entry hubs omitted between number 100 and the lowest number used. Do not forget to duplicate all wiring on the lower half of the control panel.

Before wiring a control panel, it is helpful to prepare a worksheet. Figure 2-11 shows such a worksheet for wiring the Card Reading Control Panel to read cards having the layout shown at the top of Figure 2-10 and to rearrange the information into the card data storage layout shown in the same figure. The control panel wiring is shown in Figure 2-12. Note first that card columns 78-80, containing the salesman's number, are not converted to tape. The address, in card columns 38-77, is the last item in the tape format and, therefore, the first item wired. It is placed in storage columns 61-100. Then, continuing with the amount, each preceding blockette word is wired to consecutively lower-numbered storage columns.

Figure 2-13 shows a worksheet for a Conversion Control Panel which produces the tape layout of Figure 2-9. The corresponding wiring is shown in Figure 2-14. Step 1 is wired first and represents the least significant bits of blockette word 14. Successive bit groups are then wired to consecutive steps. Two hubs are provided for each step so that repeated usage of any conversion rule may be achieved by chain wiring. Note that each

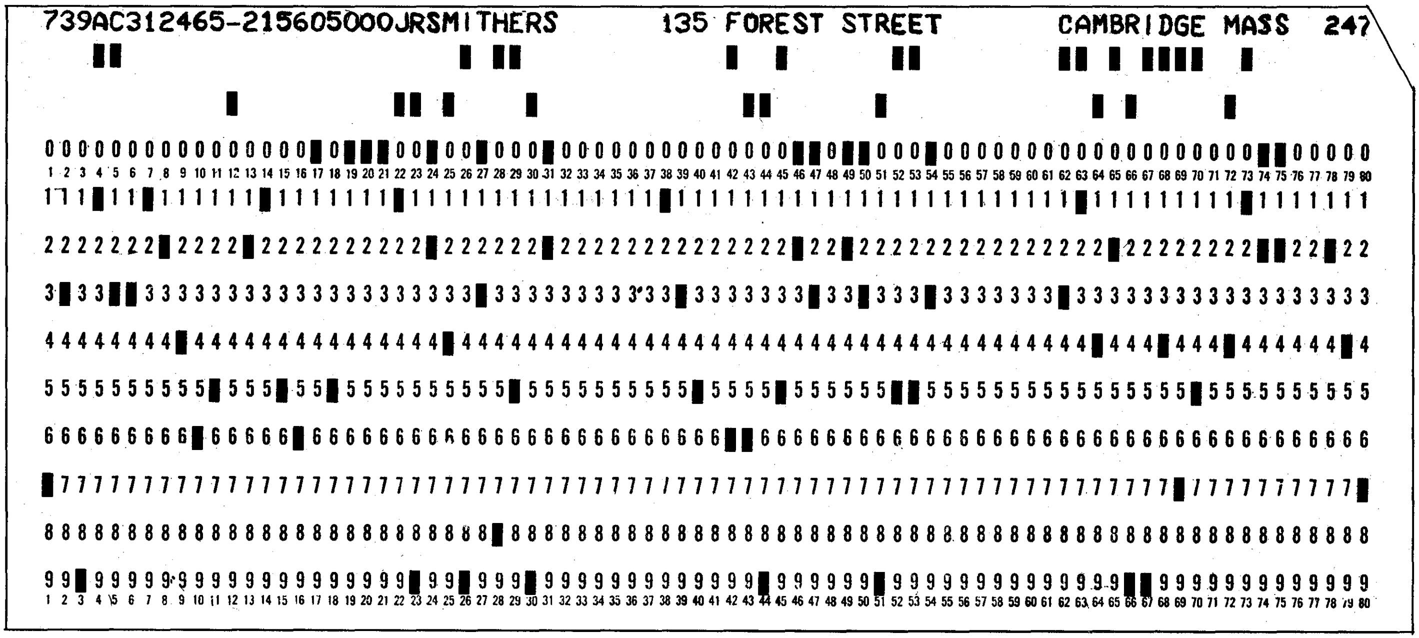


Figure 2-8

Word												Contents
1	7	3	9	A	C	3	1	2	4	6	5	District and Policy No.
2	S	M	I	T	H	E	R	S				Name
3	Bk	Bk	Bk	Bk	Bk	Bk	J	R				
4						1	1	2	1	5	6	Date
5	+						0	5	0	0	0	Amount of Policy
6	1	3	5	Bk	F	O	R	E				Street
7	S	T	Bk	S	T	R	E	E				
8	T	Bk	Bk	Bk	Bk	Bk	Bk	Bk				
9	C	A	M	B	R	I	D	G				City and State
10	E	Bk	M	A	S	S	Bk	Bk				
11												
12												
13												
14												

2 bits = ☐
4 bits (numeric) = ☐
6 bits (alphanumeric) = ☐

Bk = Blank

All unused space to be filled with zeros

Figure 2-9

Flow of Information Through Input Converter

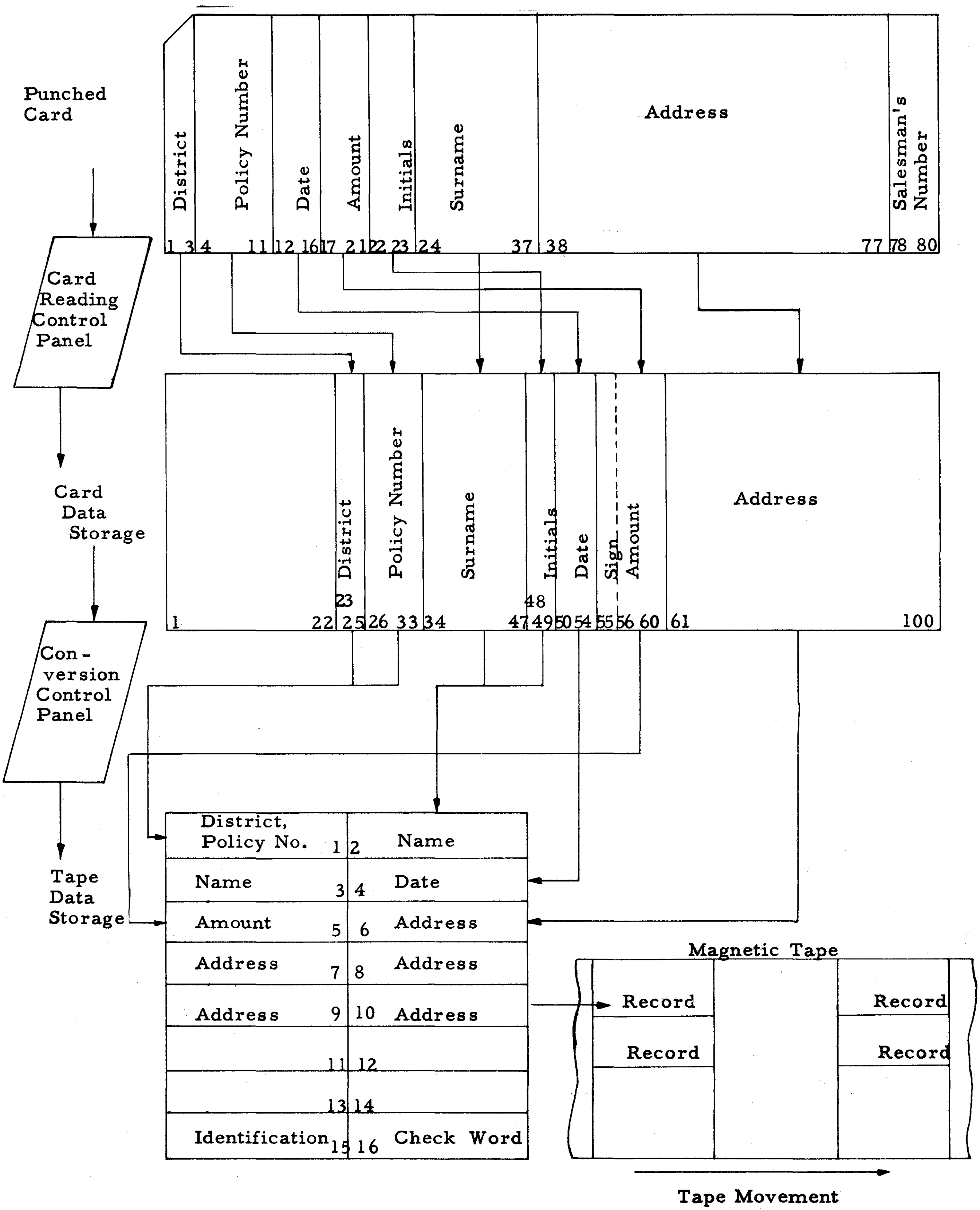


Figure 2-10

CARD READING CONTROL PANEL WORKSHEET

Operation	Blockette Word	Information	Card Data Storage	Card Column
(1)	10 - 6	135 Forest Street Cambridge, Mass.	100 - 61	77 - 38
(2)	5	05000 (amount)	60 - 56	21 - 17
(3)	5	+	55	21
(4)	4	56 (year), 21 (day)	54 - 51	16 - 13
(5)	4	11 (month)	50	12
(6)	3	J R (initials)	49 , 48	23 , 22
(7)	3, 2	Smithers	47 - 34	37 - 24
(8)	1	AC312456	33 - 26	11 - 4
(9)	1	739 (district)	25 - 23	3 - 1

Figure 2-11

step represents one application of one conversion rule but not necessarily the conversion of one character.

In Figure 2-14, it is first necessary to wire a sufficient number of zero-bit fillers to complete blockette words 14 through 11. This is accomplished by wiring conversion rule 17 to the first 16 steps (four steps of rule 17 form one word of zeros). Words 10 through 6 are next converted using 40 applications of conversion rule 2 in steps 17 through 56. Since only 34 of these steps may be wired from a given hub, two of the Rule 2 hubs must be utilized. Words 5 through 1 are then wired, working from their least-significant to their most-significant bits. Care must be taken to insure that a full 48 bits are supplied to complete each word, that no rules hub is used more than 34 times, and that no steps are omitted between step 1 and the last step used.

CONVERSION CONTROL PANEL WORKSHEET

Blockette Word	Composition	Rule	Steps
14	all zeros	17	1-4
13	all zeros	17	5-8
12	all zeros	17	9-12
11	all zeros	17	13-16
10-6	all alphanumeric	2	17-56
5	5 decimal digits	4	57-61
5	24 binary zeros	17	62-63
5	1 sign	10	64
4	4 decimal digits	4	65-68
4	1 month	12	69
4	24 binary zeros	17	70-71
3-2	all alphanumeric	2	72-87
1	6 decimal digits	4	88-93
1	2 alphanumeric	1	94-95
1	3 decimal digits	4	96-98

Figure 2-13

DATAmatic

Layout Form — Input Converter, Model 1200

(CARD READING CONTROL PANEL)

Title...

Prepared by.. For Program...

By Programmer.. Checked by..

Date.. Remarks..

....

Modification.. Page.................. of..........

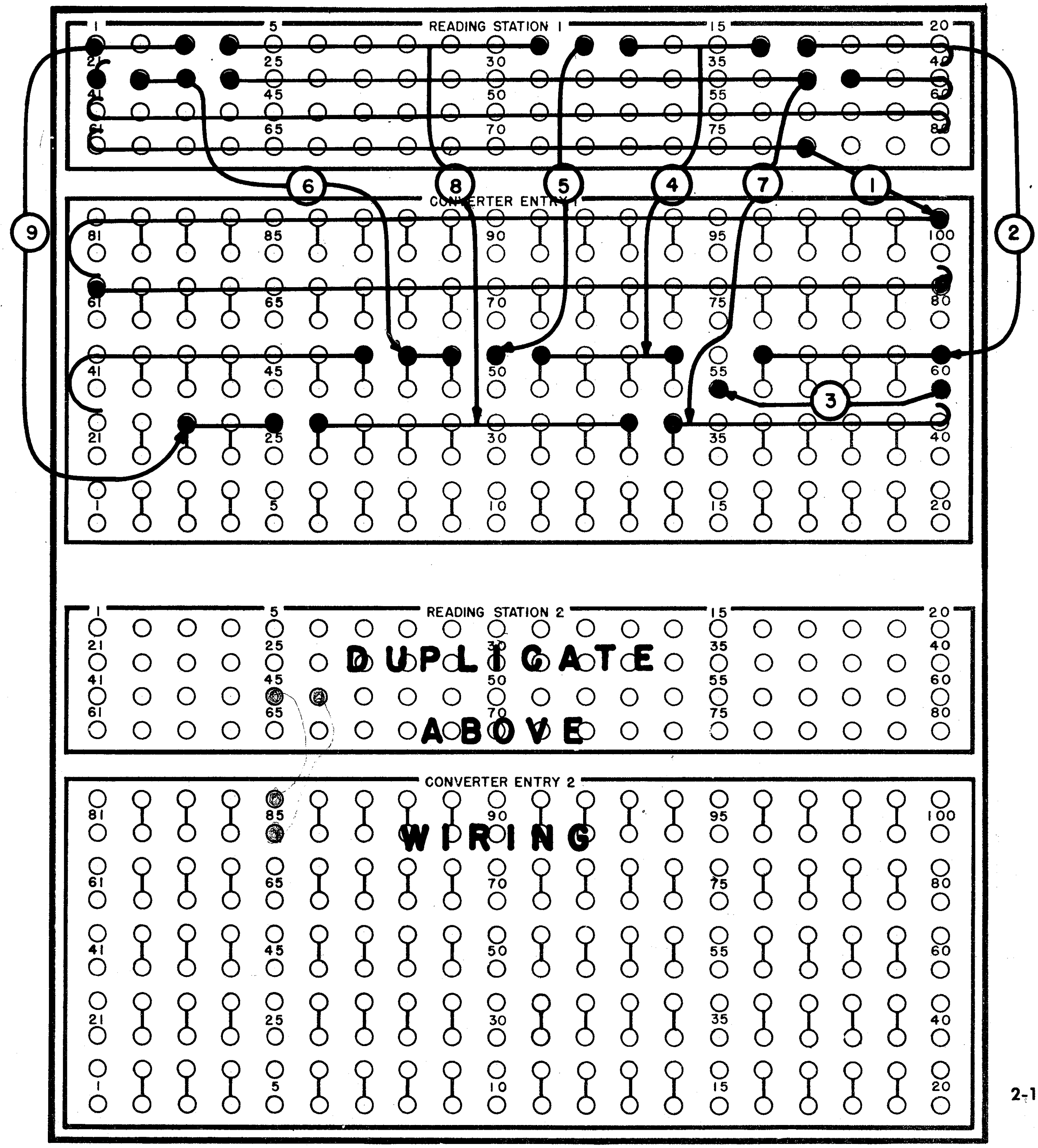


Figure 2-12

DATAmatic

Layout Form — Input Converter, Model 1200

(CONVERSION CONTROL PANEL)

Title..

Prepared by.. For Program..

By Programmer.. Checked by..

Date... Remarks..

....

Modification.. Page.............. of..........

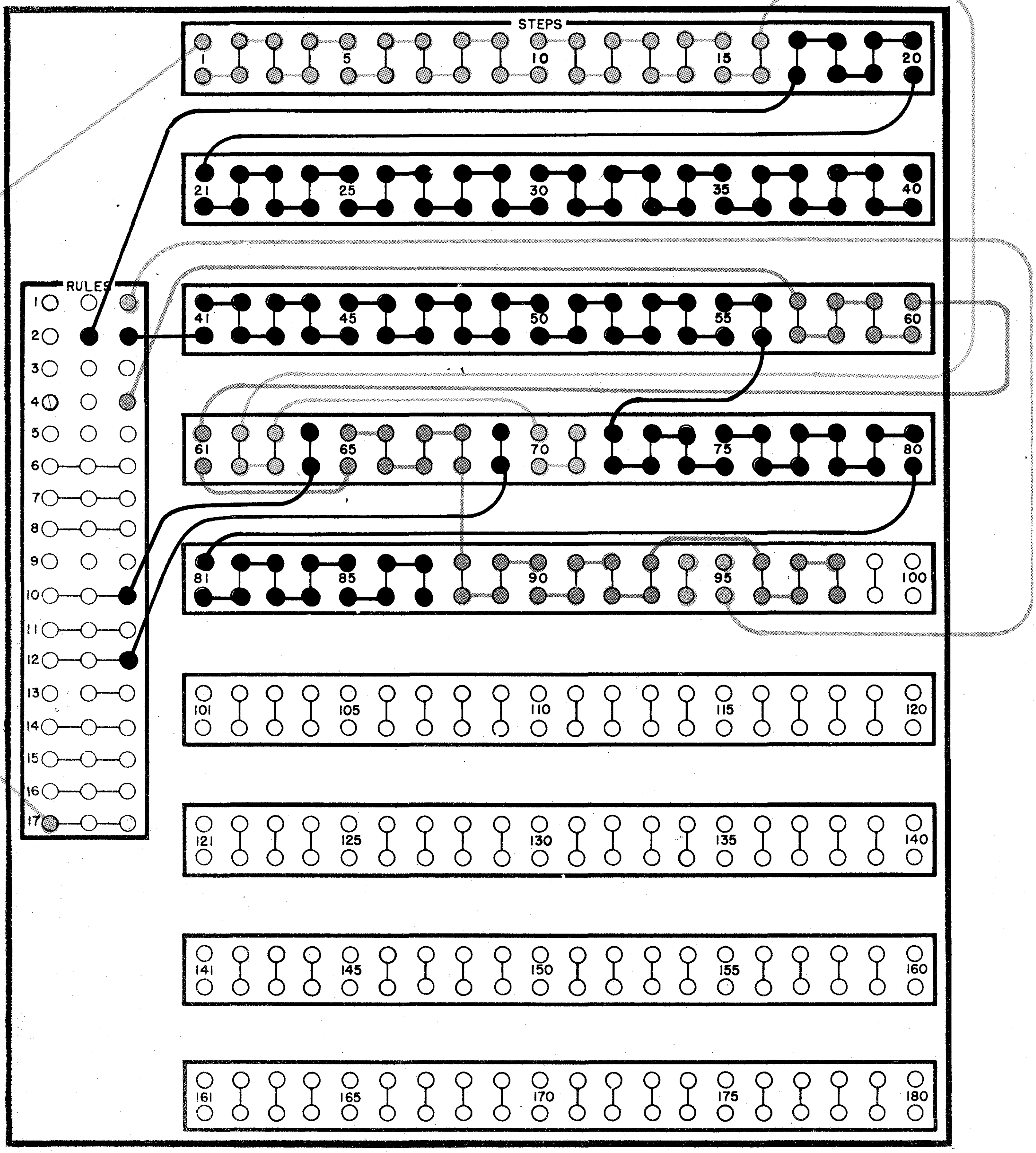


Figure 2-14

TAPE EDITING

Due to extensive editing, the magnetic tape produced by the Card Input System will be ready for processing. Further editing of the tape, if desired, may be performed during the first pass through the Central Processor. This may include compaction of information or deletion of words with incorrect weight counts and records with signals indicating possible illegal conditions.

MODEL 1200 CONVERSION RULES

The following 17 conversion rules are provided in the Model 1200 Card Input System to govern the translation of the various punch configurations in a card column into the bit configurations within the word structure on magnetic tape:

Rule 1: Alphanumeric (Blank Illegal)
The punch array within each card column is converted into six bits on magnetic tape in accordance with the Alphanumeric Code Chart (see Table 2-2). A blank column on the card, as shown on the chart, is considered an illegal punch. The complete standard set of 47 Hollerith punch codes is covered by this rule.

Rule 2: Alphanumeric (Blank Legal)
The punch array within each card column is converted into six bits in accordance with the Alphanumeric Code Chart. A blank column on the card, as shown in the chart, is a permissible array. The complete standard set of 47 Hollerith punch codes is covered by this rule.

Rule 3: Alphanumeric (Full Code)
The punch array within each card column is converted into six bits in accordance with the Alphanumeric Code Chart. All of the 56 DATAmatic punch configurations, as well as a blank column, are permissible under this rule.

Rule 4: Numeric (Blank Illegal; Zone Illegal)
The punch array within each card column is converted into four bits in accordance with the Numeric Code Chart (see Table 2-2). A blank card column yields an illegal-punch signal. The existence of an R-punch, an X-punch, or both, causes any columnar punch array to be considered illegal.

Rule 5: Numeric (Blank = Zero; Zone Illegal)
The punch array within each card column is converted into four bits in accordance with the Numeric Code Chart. A blank card column is permissible and is converted into the configuration 0000. The existence of an R-punch, an X-punch, or both, causes any columnar punch array to be considered illegal.

Rule 6: Numeric (Blank = G; Zone Illegal)
The punch array within each card column is converted into four bits in accordance with the Numeric Code Chart. A blank card column is permissible and is converted into the code 1111. The existence of an R-punch, an X-punch, or both, causes any columnar punch array to be considered illegal.

Rule 7: Numeric (Blank = 0; Zone Legal)
The punch array within each card column is converted into four bits in accordance with the Numeric Code Chart. A blank card column is permissible and is converted into the code 0000. The existence of an R-punch, an X-punch, or both, is permitted and does **not** yield an illegal-punch signal.

Rule 8: Numeric (Blank Illegal; Zone Legal)
The punch array within each card column is converted into four bits in accordance with the Numeric Code Chart. A blank card column yields an illegal-punch signal. An R-punch, or an X-punch, or both, is permitted in any column of a card.

Rule 9: Hexadecimal
The punch array in each card column is converted into four bits in accordance with the Hexadecimal Code Chart (see Table 2-2). A blank card column or the existence of an X-punch in any column yields an illegal-punch signal.

Rule 10: Sign (R Positive)
A card column is converted into four bits in accordance with the R-Positive Sign Code Chart (see Table 2-2). A blank card column is a permissible configuration.

Rule 11: Sign (X Positive)
A card column is converted into four bits in accordance with the X-Positive Sign Code Chart (see Table 2-2). A blank card column is a permissible configuration.

Rule 12: Month
A card column is converted into eight bits in accordance with the Month Code Chart (see page 2-14). A blank card column is an illegal configuration.

Table 2-1, Page 1

Rule 13: Transcription

Each card column is converted into a 12-bit array, each punch in the card column being independently converted into a corresponding "one" bit in the 12-bit array. Any configuration of punches in a card column is permissible. The correspondence between punch positions in a card column and "one" bits in the 12-bit array is:

Punch position in column:
9 8 7 6 5 4 3 2 1 0 X R
"One"-bit position in 12-bit array:
12 11 10 9 8 7 6 5 4 3 2 1

Rule 14: Filler (Two-bit)

Each use of this rule transmits two "zero" bits to magnetic tape.

Rule 15: Filler (Four-bit)

Each use of this rule transmits four "zero" bits to magnetic tape.

Rule 16: Filler (Six-bit)

Each use of this rule transmits six "zero" bits to magnetic tape.

Rule 17: Filler (Twelve-bit)

Each use of this rule transmits twelve "zero" bits to magnetic tape.

MONTH CODE CHART FOR CONVERSION RULE 12

In the Month Code Chart, as in Table 2-2, the zone section of a card column comprises the R, X, and 0 punch positions. The numeric section comprises the punch positions 0 through 9. A dash (—) in either the zone or the numeric section of the charts indicates that no punch exists in that section.

Punch Array in Card Column		Bit Configuration on Magnetic Tape	Equivalent DATAmatic 1000 Symbol
Zone	Numeric		
—	1	0000 0001	01
—	2	0000 0010	02
—	3	0000 0011	03
—	4	0000 0100	04
—	5	0000 0101	05
—	6	0000 0110	06
—	7	0000 0111	07
—	8	0000 1000	08
—	9	0000 1001	09
R	—	0001 0010	12
X	—	0001 0001	11
0	—	0001 0000	10

Note: Any punch configuration not listed in the above chart should never be used under Conversion Rule 12. The erroneous use of any such disallowed configuration will lead to an illegal-punch signal.

Table 2-1, Page 2

Punch Card To Magnetic Tape Conversion Chart

Card Punch		Illegal Punch Under Rule											Alphanumeric Rules 1, 2, 3		Numeric Rules 4, 5, 6, 7, 8		Hexadecimal Rule 9		R-Pos. Sign Rule 10		X-Pos. Sign Rule 11	
Zone	Num.	1	2	3	4	5	6	7	8	9	10	11	Tape Code	Sym.	Tape Code	Sym.	Code	Sym.	Code	Sym.	Code	Sym.
-	-	X			X				X	X			010000	Blank	0000 * 1111 *	0 (Zero) G	-		1101	+	1101	+
-	1												000001	1	0001	1	0001	1	1101	+	1101	+
-	2												000010	2	0010	2	0010	2	1101	+	1101	+
-	3												000011	3	0011	3	0011	3	1101	+	1101	+
-	4												000100	4	0100	4	0100	4	1101	+	1101	+
-	5												000101	5	0101	5	0101	5	1101	+	1101	+
-	6												000110	6	0110	6	0110	6	1101	+	1101	+
-	7												000111	7	0111	7	0111	7	1101	+	1101	+
-	8												001000	8	1000	8	1000	8	1101	+	1101	+
-	9												001001	9	1001	9	1001	9	1101	+	1101	+
-	8, 2	X	X		X	X	X	X	X	X			001010	'	-		-		1101	+	1101	+
-	8, 3				X	X	X	X	X	X			001011	#	-		-		1101	+	1101	+
-	8, 4				X	X	X	X	X	X			001100	@	-		-		1101	+	1101	+
-	8, 5	X	X		X	X	X	X	X	X			001101	+	-		-		1101	+	1101	+
-	8, 6	X	X		X	X	X	X	X	X			001110	Error	-		-		1101	+	1101	+
-	8, 7	X	X		X	X	X	X	X	X			001111	Error	-		-		1101	+	1101	+

* 0000 Rules 5 and 7
1111 Rule 6 only

Table 2-2, page 1

Card Punch		Illegal Punch Under Rule											Alphanumeric Rules 1, 2, 3		Numeric Rules 4,5,6,7,8		Hexadecimal Rule 9		R-Pos. Sign Rule 10		X-Pos. Sign Rule 11	
Zone	Num.	1	2	3	4	5	6	7	8	9	10	11	Tape Code	Sym.	Tape Code	Sym.	Code	Sym.	Code	Sym.	Code	Sym.
R	-				X	X	X			X			110000	&	0000	0 (Zero)	-		1101	+	0101	-
R	1				X	X	X			X			010001	A	0001	1	-		1101	+	0101	-
R	2				X	X	X						010010	B	0010	2	1010	B	1101	+	0101	-
R	3				X	X	X						010011	C	0011	3	1011	C	1101	+	0101	-
R	4				X	X	X						010100	D	0100	4	1100	D	1101	+	0101	-
R	5				X	X	X						010101	E	0101	5	1101	E	1101	+	0101	-
R	6				X	X	X						010110	F	0110	6	1110	F	1101	+	0101	-
R	7				X	X	X						010111	G	0111	7	1111	G	1101	+	0101	-
R	8				X	X	X			X			011000	H	1000	8	-		1101	+	0101	-
R	9				X	X	X			X			011001	I	1001	9	-		1101	+	0101	-
R	8, 2	X	X		X	X	X	X	X	X			011010	CR	-		-		1101	+	0101	-
R	8, 3				X	X	X	X	X	X			011011	.	-		-		1101	+	0101	-
R	8, 4				X	X	X	X	X	X			011100	▢	-		-		1101	+	0101	-
R	8, 5	X	X		X	X	X	X	X	X			011101	(	-		-		1101	+	0101	-
R	8, 6	X	X		X	X	X	X	X	X			011110	)	-		-		1101	+	0101	-
R	8, 7	X	X		X	X	X	X	X	X			011111	Error	-		-		1101	+	0101	-
R,X	-	X	X		X	X	X			X					0000	0 (Zero)			1101	+	1101	+
R,X	any	X	X		X	X	X			X					0000	0 (Zero)			1101	+	1101	+
R,0	-	X	X		X	X	X			X					0000	0 (Zero)			1101	+	0101	-
R,0	any	X	X		X	X	X	X	X	X					-				1101	+	0101	-
R,X,0	-	X	X		X	X	X			X					0000	0 (Zero)			1101	+	1101	+
R,X,0	any	X	X		X	X	X	X	X	X					-				1101	+	1101	+

Table 2-2, page 2

Card Punch		Illegal Punch Under Rule											Alphanumeric Rules 1, 2, 3		Numeric Rules 4,5,6,7,8		Hexadecimal Rule 9		R-Pos. Sign Rule 10		X-Pos. Sign Rule 11	
Zone	Num.	1	2	3	4	5	6	7	8	9	10	11	Tape Code	Sym.	Tape Code	Sym.	Code	Sym.	Code	Sym.	Code	Sym.
X*	-				X	X	X			X			100000	-	0000	0 (Zero)	-		0101	-	1101	+
X	1				X	X	X			X			100001	J	0001	1	-		0101	-	1101	+
X	2				X	X	X			X			100010	K	0010	2	-		0101	-	1101	+
X	3				X	X	X			X			100011	L	0011	3	-		0101	-	1101	+
X	4				X	X	X			X			100100	M	0100	4	-		0101	-	1101	+
X	5				X	X	X			X			100101	N	0101	5	-		0101	-	1101	+
X	6				X	X	X			X			100110	O letter	0110	6	-		0101	-	1101	+
X	7				X	X	X			X			100111	P	0111	7	-		0101	-	1101	+
X	8				X	X	X			X			101000	Q	1000	8	-		0101	-	1101	+
X	9				X	X	X			X			101001	R	1001	9	-		0101	-	1101	+
X	8, 2	X	X		X	X	X	X	X	X			101010	=	-		-		0101	-	1101	+
X	8, 3				X	X	X	X	X	X			101011	$	-		-		0101	-	1101	+
X	8, 4				X	X	X	X	X	X			101100	*	-		-		0101	-	1101	+
X	8, 5	X	X		X	X	X	X	X	X			101101	¢	-		-		0101	-	1101	+
X	8, 6	X	X		X	X	X	X	X	X			101110	Error	-		-		0101	-	1101	+
X	8, 7	X	X		X	X	X	X	X	X			101111	Error	-		-		0101	-	1101	+
R,X	-	X	X		X	X	X			X					0000	0 (Zero)			1101	+	1101	+
R,X	any	X	X		X	X	X			X					0000	0 (Zero)			1101	+	1101	+
X,0	-	X	X		X	X	X			X					0000	0 (Zero)			0101	-	1101	+
X,0	any	X	X		X	X	X	X	X	X					-				0101	-	1101	+
R,X,0	-	X	X		X	X	X			X					0000	0 (Zero)			1101	+	1101	+
R,X,0	any	X	X		X	X	X	X	X	X					-				1101	+	1101	+

Table 2-2, page 3

Card Punch		Illegal Punch Under Rule											Alphanumeric Rules 1, 2, 3		Numeric Rules 4, 5, 6, 7, 8		Hexadecimal Rule 9		R-Pos. Sign Rule 10		X-Pos. Sign Rule 11	
Zone	Num.	1	2	3	4	5	6	7	8	9	10	11	Tape Code	Sym.	Tape Code	Sym.	Code	Sym.	Code	Sym.	Code	Sym.
0	-												000000	0 (Zero)	0000	0 (Zero)	0000	0 (Zero)	1101	+	1101	+
0	1				X	X	X	X	X	X			110001	/	-		-		1101	+	1101	+
0	2				X	X	X	X	X	X			110010	S	-		-		1101	+	1101	+
0	3				X	X	X	X	X	X			110011	T	-		-		1101	+	1101	+
0	4				X	X	X	X	X	X			110100	U	-		-		1101	+	1101	+
0	5				X	X	X	X	X	X			110101	V	-		-		1101	+	1101	+
0	6				X	X	X	X	X	X			110110	W	-		-		1101	+	1101	+
0	7				X	X	X	X	X	X			110111	X	-		-		1101	+	1101	+
0	8				X	X	X	X	X	X			111000	Y	-		-		1101	+	1101	+
0	9				X	X	X	X	X	X			111001	Z	-		-		1101	+	1101	+
0	8, 2	X	X		X	X	X	X	X	X			111010	:	-		-		1101	+	1101	+
0	8, 3				X	X	X	X	X	X			111011	,	-		-		1101	+	1101	+
0	8, 4				X	X	X	X	X	X			111100	%	-		-		1101	+	1101	+
0	8, 5	X	X		X	X	X	X	X	X			111101	1/2	-		-		1101	+	1101	+
0	8, 6	X	X		X	X	X	X	X	X			111110	Error	-		-		1101	+	1101	+
0	8, 7	X	X		X	X	X	X	X	X			111111	Space	-		-		1101	+	1101	+
R,0	-	X	X		X	X	X			X					0000	0 (Zero)			1101	+	0101	-
R,0	any	X	X		X	X	X	X	X	X					-				1101	+	0101	-
X,0	-	X	X		X	X	X			X					0000	0 (Zero)			0101	-	1101	+
X,0	any	X	X		X	X	X	X	X	X					-				0101	-	1101	+
R,X,0	-	X	X		X	X	X			X					0000	0 (Zero)			1101	+	1101	+
R,X,0	any	X	X		X	X	X	X	X	X					-				1101	+	1101	+

Table 2-2, page 4

SECTION III

MODEL 1300 OUTPUT SYSTEM

SECTION III—MODEL 1300 OUTPUT SYSTEM

INTRODUCTION

The DATAmatic Model 1300 Output System accepts information recorded on magnetic tape and generates electrical signals which drive specific equipment to print or punch this information. The operation of this system is completely independent of the Central Processor. One Magnetic File Unit is assigned to read from tape the information to be converted. At the user's option, the system may be arranged to include a switch, permitting this Magnetic File Unit to communicate with the Central Processor when the Output System is not in operation.

The Model 1300 processes information from tape in groups of 16 words, called blockettes. Two such blockettes are derived from each block on tape, the first comprising words 1 through 16 and the second comprising words 17 through 32. Information to be converted by the Model 1300 must be stored within words 1-32 of the tape block. Each blockette is stored temporarily in binary form by an array of magnetic cores called Converter Input Storage. From there it is converted under control of a Conversion Control Panel into standard punched-card code and sent to the Output Storage section. This section electrically simulates 120 columns of punched-card data and stores one blockette of information at one time. The information in output storage is processed by a standard 150-line-per-minute tabulator or a standard 100-card-per-minute card punch, each blockette becoming the contents of one or more lines of printing or one punched card. Most of the tabulator or card punch functions, governed by standard control panels, are preserved and normal summary punching provisions are retained. Figure 3-1 is a block diagram of the Model 1300 and its associated units.

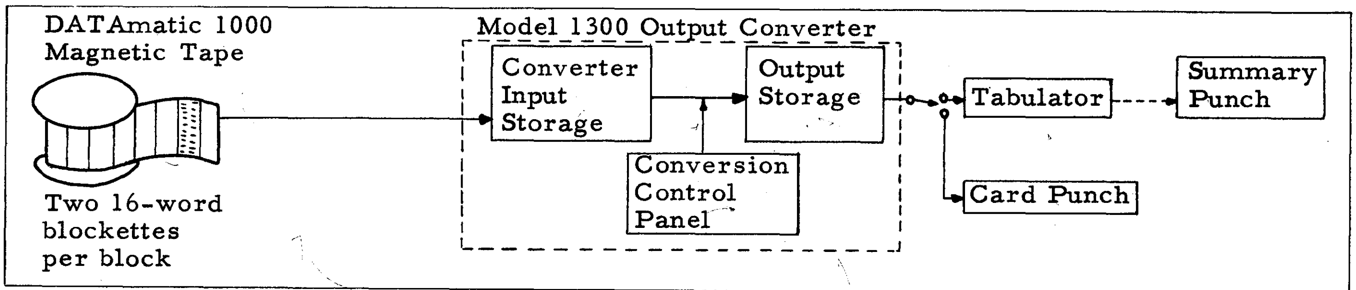


Figure 3-1

OPERATION

Figure 3-2 shows the console panel from which the Model 1300 Output System is operated. The programmer may elect one of the following four basic modes of operation, as determined by the settings of the PROCESS CONTROL WORD and PROCESS ALL BLOCKS switches at the left of the panel.

(1) Process all 16 words of each blockette without the use of a control word.

(2) Process the first 15 words of those blockettes indicated by certain control digits which the programmer stores in the 16th word.

(3) Process all 16 words of those blockettes indicated by certain control digits stored in the 16th word.

(4) Process only the first 15 words of each blockette without the use of a control word.

The settings of the PROCESS CONTROL WORD and PROCESS ALL BLOCKS switches which establish these modes are summarized in Figure 3-3. Modes 1 and 2 are the most commonly used. Mode 3 is provided for the case where the programmer requires more than 15 words of information storage and less than a complete word of control digits. Mode 4 will be used infrequently.

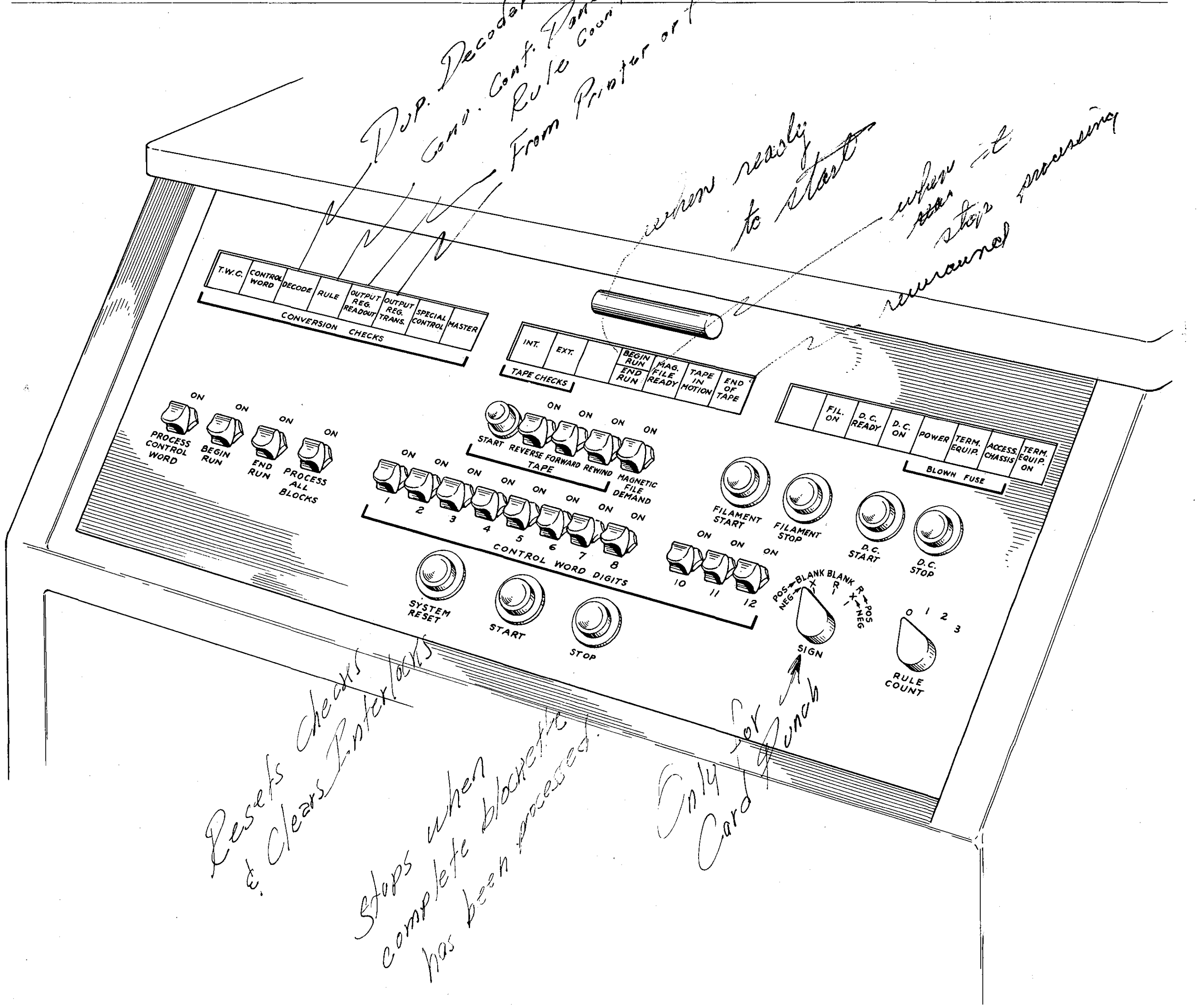


Figure 3-2

PROCESS CONTROL WORD	PROCESS ALL BLOCKS	RESULT
ON	ON	Process 16 words without control word function (mode 1)
OFF	OFF	Process 15 words, word 16 is control word (mode 2)
ON	OFF	Process 16 words, word 16 is control word (mode 3)
OFF	ON	Process 15 words without control word function (mode 4)

Figure 3-3

CONTROL WORD DIGIT POSITIONS

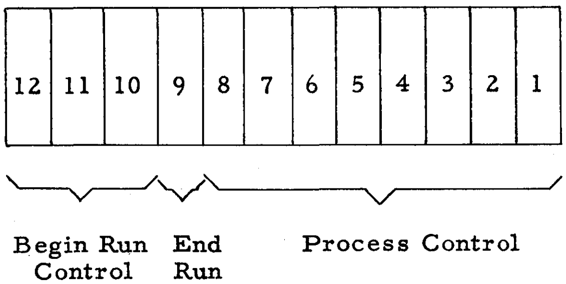


Figure 3-4

CONTROL WORD: From one to twelve digits of the 16th blockette word may be used for control purposes. The functions of these digits, shown in Figure 3-4, are governed by the settings of the BEGIN RUN, END RUN, and the CONTROL WORD DIGITS switches on the console panel. Note that these switches correspond to the breakdown of digits shown in Figure 3-4 with the following two exceptions:

(1) The ninth CONTROL WORD DIGITS switch is omitted because the logical function of this switch is performed by the END RUN switch, and

(2) The 11 remaining switches are numbered from left to right, while the control digit positions are numbered from right to left.

Whenever all or part of the 16th word is used for control purposes, the programmer takes precautions that this word shall not contain more than 26 binary "1"s, including control digits, weight count bits, and stored information, if any. Those digits which are used for control will be either hexadecimal "0"s or "5"s and their use is described in the following paragraphs. Therefore, an example of mode 3 operation might utilize a blockette of fifteen information words plus a sixteenth word consisting of four control digits and certain additional stored information. Considering that four control digits may all be "5"s, for a total of eight "one" bits, and that the weight count bits may include up to three more "one" bits, it can be seen that information up to a total of 15 "one" bits may be stored in the rest of the word.

BEGIN RUN: To search for the tape block containing the first record to be processed, set the BEGIN RUN switch to ON, set *one* of the eight Process Control switches (1-8) and *one* of the three Begin Run switches (10-12) to ON, and press the START button. The 16th word of each tape block will be searched until one such word is found which has the digit "5" in the two positions specified by the "on" switches. The tape is then positioned so that the tape block in which the coincidence occurs is the next to be read, the converter stops, and the BEGIN RUN light is illuminated. At the block on tape where the run is to start, the programmer will have placed in the 16th word a control word which has hex "5"s in the appropriate digit positions. If more than one file is recorded on the same tape, the first block of each file will have a different begin run control word in its 16th word.

PROCESS RUN: The BEGIN RUN switch is set to OFF and the PROCESS CONTROL WORD and PROCESS ALL BLOCKS switches are set to correspond to the desired mode of operation (see Figure 3-3). If operating under control of blockette word 16 (mode 2 or 3), one of the CONTROL WORD DIGITS switches 1 through 8 is set to ON. Only one of these switches at a time may be set to ON. The processing run is started by pressing SYSTEM RESET, followed by START. Each blockette which has a hex "5" in the control word digit position corresponding to the "on" switch will be processed. If this digit is a "0", the blockette will not be processed. If operating without the use of control words (mode 1 or 4), it is only necessary to press SYSTEM RESET, followed by START, and all blocks will be processed until the end of the run.

END RUN: When the END RUN switch is "on", the occurrence of a "5" in the end run digit (see Figure 3-4) of blockette word 16 causes the converter to stop after processing the associated blockette. Since the end run function is independent of the mode of operation, this digit position is not available for information storage in any mode.

TAPE CONTROL: The TAPE switches on the console panel govern rewinding the tape and advancing the tape one block at a time in either direction. To move the tape one block, set the REVERSE or the FORWARD switch to ON and push the TAPE START button. To rewind the tape, set the REWIND switch to ON and press the TAPE START button. The REWIND switch may then be set to OFF as the rewind will proceed automatically. The TAPE IN MOTION light remains lit while the rewind is in progress and the END OF TAPE light signals the end of the rewind. Only one of the three switches REVERSE, FORWARD, or REWIND may be "on" when the TAPE START button is pushed. The SYSTEM RESET button must be pressed before and after each manual tape motion or group of motions.

The MAGNETIC FILE DEMAND switch demands control of the associated Magnetic File Unit when it is set to ON and releases control of the MFU to the Central Processor when it is set to OFF. If a Magnetic File Unit is demanded by both the converter and the Central Processor, it will be controlled by the unit making the prior demand.

When the converter stops with an error indication, the blockette in which the indication occurred will be printed or punched for inspection. The operator may then reverse the tape manually (one or more blocks) and resume processing, or he may resume processing with the following blockette. The programmer should provide instructions regarding the desired error-stop procedure.

TIMING: During the processing run, each blockette (two from each tape block) is brought into the converter and processed either as 15 or 16 words of information, with or without the control word function. The standard rate of processing these blockettes is 150 per minute. Under normal conditions, at least six blockettes may be discarded between each blockette processed without any loss of printing speed.

CONVERSION

Information in converter input storage is translated into punched-card code by a code translator, controlled by the Conversion Control Panel (see Figure 3-5). Up to 120 columns of information in punched-card code are stored in output storage. The conversion from input storage to output storage occurs in steps, of which a maximum of 240 may be wired on the Conversion Control Panel. Each step extracts a specified number of bits and converts them to a column of punched-card code in output storage or rejects them. The number of bits extracted is specified by the wiring of the step. Successive conversion steps extract bits sequentially from input storage and insert characters sequentially in output storage. Bits are extracted in the same sequence as they were written on tape (see Section II, page 2-2); i.e., starting with the low-order bits of the last information word (15 or 16) of the blockette and proceeding from low order to high order through blockette word 1. The first character converted is stored in storage column 120. Note that this may not correspond to step 1 if that step is wired for rejection. Successive converted characters are stored sequentially by decreasing order of storage columns.

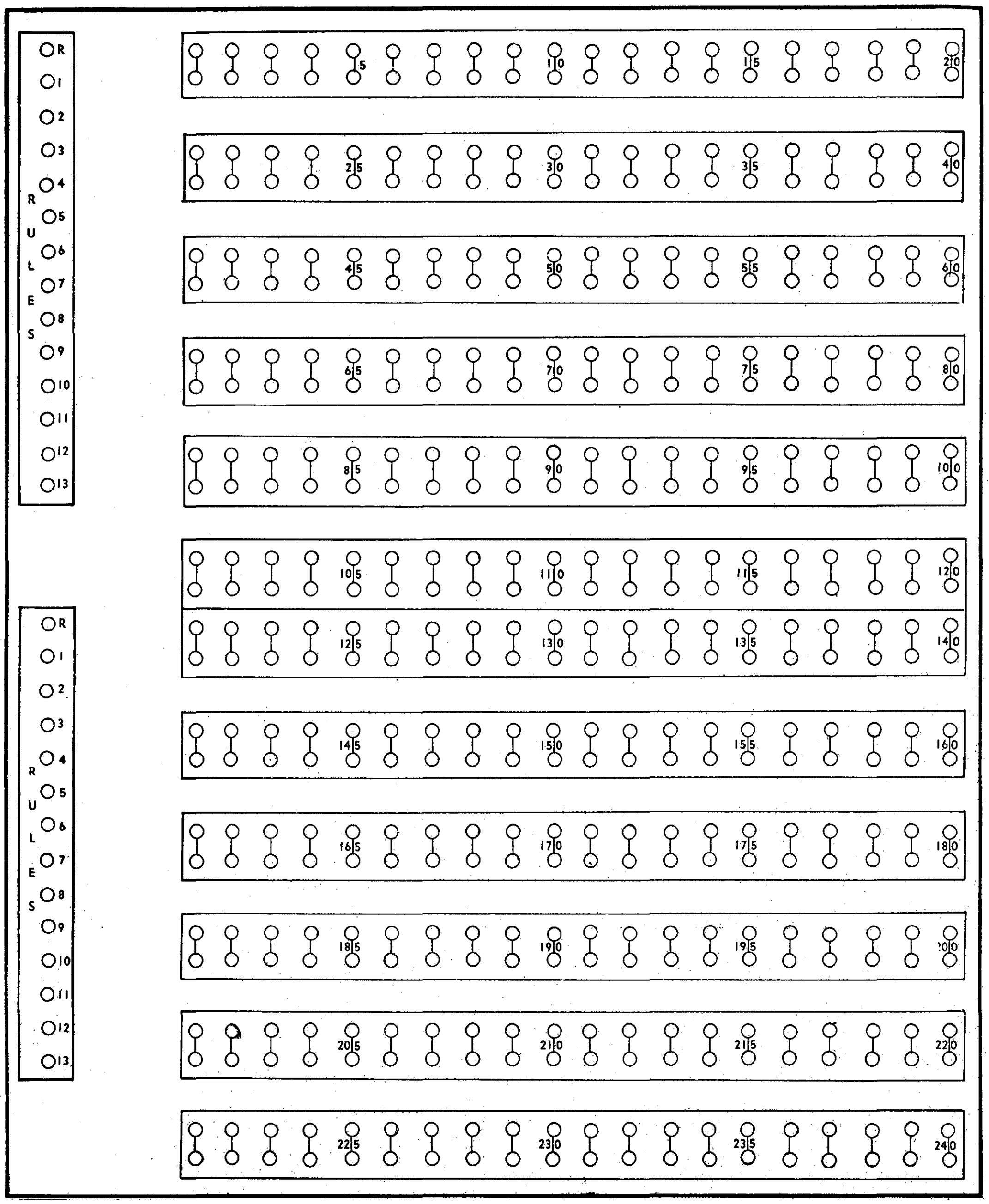


Figure 3-5

Each step in the conversion process executes one of the 14 rules listed in Table 3-1 (pages 3-11 through 3-12). Rules are selected by means of the control panel wiring. There are two sets of rules hubs on the Conversion Control Panel, as shown in Figure 3-5. The upper rules hubs are for use with steps 1 through 120, and the lower hubs are for use with steps 121 through 240.

Step 1 is first wired to the appropriate rules hub to convert or reject the low-order bits of blockette word 15 (or 16). Whenever a step repeats the rule used by the preceding step, it must be wired to an R (repeat) hub. The upper R hub is used in steps 1 through 120 and the lower R hub in steps 121 through 240. Each word must be extracted from input storage in an integral number of steps. Each time that a complete word has been extracted the next step must be wired to rule 11 (end-of-word), except that rule 12 is used at the end of blockette word 9 and rule 13 at the end of word 1. The first step of a word must not be wired to an R hub.

In planning the conversion wiring, it is necessary to count the number of bits extracted from input storage. Exactly 48 bits must be extracted or rejected for each word converted (the weight count bits are ignored in this process). The full 15 words must be wired (16 if the entire blockette is to be processed). If less than the full blockette is to be converted, enough steps must be wired to rejection rules and end-of-word rules to account for the full total; otherwise one of the CONVERSION CHECKS lights will signal a programming error.

RULE COUNT: The RULE COUNT switch on the console panel is provided as a check against the possibility of attempting a processing run with the wrong Conversion Control Panel in the converter. Before a control panel can be used in the converter, the programmer must manually compute a parameter known as the Rule Count of the panel and the RULE COUNT switch must be set to this number. If a processing run is attempted using an incorrect rule count setting, the converter will stop and the RULE light will light. Note that the available rules include conversion rules (1, 2, 3, 4, 5, and 10); rejection rules (6, 7, 8, and 9); special control rules (9 and 10); end-of-word rules (11, 12, and 13); and repeats. To compute the rule count of a panel, start with the total number of steps wired, i.e., the highest number used. To this number, add the number of times that actual conversion rules (1 through 5 and 10) are wired and subtract the number of *repeats* of rejection rules (not the rejection rules themselves). The result is the unreduced rule count and it is reduced by subtracting an integral number of fours, so that the remainder is a number from 0 to 3. This remainder is the rule count of the panel.

For example, the rule count of the panel shown in Figure 3-10 is computed as follows. There are 148 steps wired. Note that conversion rules are shown in light green, rejection rules in gray, end-of-word rules in dark green, and repeats in black for the reader's convenience. There are nine steps wired from conversion rules and 60 *repeats* of rejection rules. The unreduced rule count is

$$148 + 9 - 60,$$

which equals 97. Subtracting 24 fours, the remainder is 1, which is the rule count of the panel. Consequently, the RULE COUNT switch must be set on 1 for successful converter operation with this panel.

OUTPUT PROVISIONS

To effect the transfer of information out of output storage, this section is wired to 120 reading hubs on the tabulator control panel (see Figure 3-6) in such a way that signals from the 120 output storage columns have the same pulse sequence and timing as signals from a card read at the tabulator second reading station. Alternatively, the same connections may be made to 120 reading hubs on the card punch control panel. Connections are also made to 16 tabulator or card punch special control hubs. In general, the normal rules for the tabulator or card punch control panel will be operative and transposition, omission, or duplication of columns may be accomplished by proper wiring of these panels.

All format arrangements and special tabulator (or card punch) functions are performed under control of the special control hubs. Each execution of rule 9 or rule 10 activates the next special control hub in sequence if the proper code is encountered (see Table 3-1); otherwise that hub is bypassed and the next special control rule governs the following special control hub. The special control signals are sent to the output unit prior to the processed information in order to set up all format circuits before printing or punching.

TAPE EDITING

Before the start of the output conversion process, it is necessary to arrange the output information in the desired tape format. This operation, which includes placing all of the information desired for output in the first 32 words of the tape block, can normally be incorporated in the processing of the data by the Central Processor. Occasionally, the programmer may wish to take advantage of a separate editing pass for this purpose. Note that each converted blockette is retained in output storage until the next blockette is converted and may be used by the programmer to print a number of *different* lines.

Lower Right Quadrant of Standard Tabulator Control Panel Showing Hubs Used by the Type 1300 Output Converter

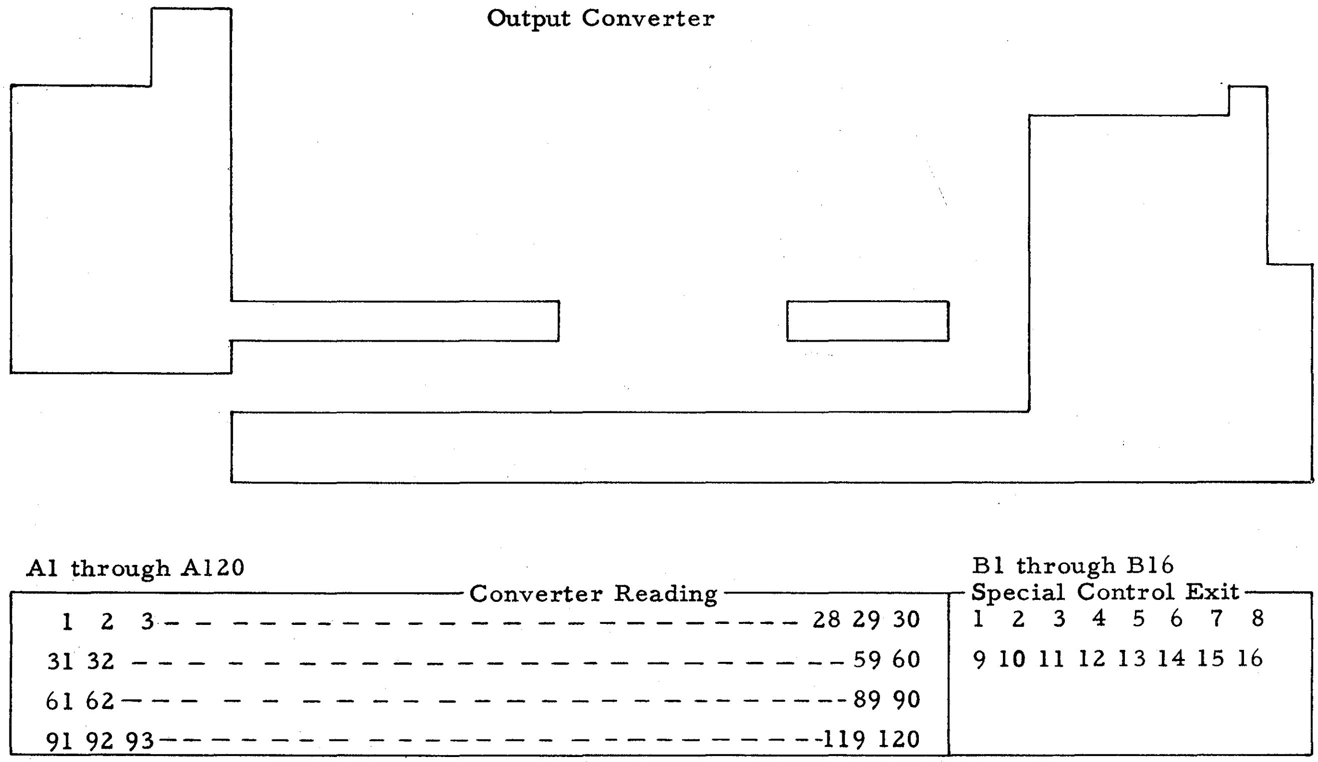


Figure 3-6

APPLICATION

As an example of the production of printed output with the Model 1300 Converter, it may be desired to prepare a batch of dividend checks, such as those shown in Figure 3-7. The information necessary to prepare a check may be stored in 16 words, comprising one blockette, and two blockettes may be stored in each tape block, as shown in Figure 3-8. The detailed layout of two typical blockettes is shown in Figure 3-9. The small numbers indicate the conversion rule to be wired to each step. Where no number is indicated, the step is wired to a repeat hub.

Figure 3-10 shows the wiring of the Conversion Control Panel to convert the information from one blockette into punched-card code. Since word 15 of the blockette is not used to store information in this example, the first step wired is a 6-bit reject (rule 8) followed by seven repeat steps chain-wired together. Step 9 is an end-of-word step (rule 11). Rule 1 (6-bit alphanumeric) is used in step 10 to convert the low-order character of word 14 and is followed by seven more repeat steps. Note that the first of these (step 11) is wired to the last previous repeat step (step 8). The wiring proceeds in this fashion from word 15 to word 1 and within words from the low-order to the high-order bits. Care is taken to convert all 48 bits of each word, to follow each word by an end-of-word step, to use the end-of-word-nine and end-of-word-one rules at the specified places, and to terminate all chain wiring at step 120 and resume wiring from the lower set of rules hubs, starting with step 121. Step 121 may be wired to the lower repeat hub if desired. The computation of the rule count of this panel has been described on page 3-5.

Figure 3-11 shows two arrays of information in output storage, each converted from one of the blockettes of Figure 3-9. Column 120 is filled first and contains the first information converted from input storage (the result of step 10). Only the execution or repetition of one of the rules 1 through 5 or rule 10 will fill a storage column, and all information so converted appears in consecutive columns. The output format of the converted information is governed by the tabulator or card punch control panel.

Dividend Number	Check Date	Check Number	Check No.	Account No.	Dividend Amount Mailed	Dividend Amount Deposited
187	JUNE 28 1956	87654321	87654321	091246	32.76	

Pay to the order of	Pay this amount	Name	Dividend No.
WILLIAM J FARRINGTON JR	$*****32.76	WILLIAM J FARRINGTON JR	187

Dividend Number	Check Date	Check Number	Check No.	Account No.	Dividend Amount Mailed	Dividend Amount Deposited
187	JUNE 28 1956	87654322	87654322	091253		20185.94

Pay to the order of	Pay this amount	Name	Dividend No.
ALVAH T EDISON	$**20185.94	ALVAH T EDISON	187

Figure 3-7

Tape Channel	Digits 12 11 10 9 8 7 6 5 4 3 2 1	Word Number		Digits 12 11 10 9 8 7 6 5 4 3 2 1
0		1	2	
1		3	4	
2		5	6	
3	DIV. NO.	7	8	MONTH DAY YEAR
4	CHECK NUMBER	9	10	ACCOUNT NUMBER
5	CHECK AMOUNT	11	12	NAME
6	NAME	13	14	NAME
7		15	16	CONTROL WORD
8		17	18	
9		19	20	
10		21	22	
11	DIV. NO.	23	24	MONTH DAY YEAR
12	CHECK NUMBER	25	26	ACCOUNT NUMBER
13	CHECK AMOUNT	27	28	NAME
14	NAME	29	30	NAME
15		31	32	CONTROL WORD

Figure 3-8

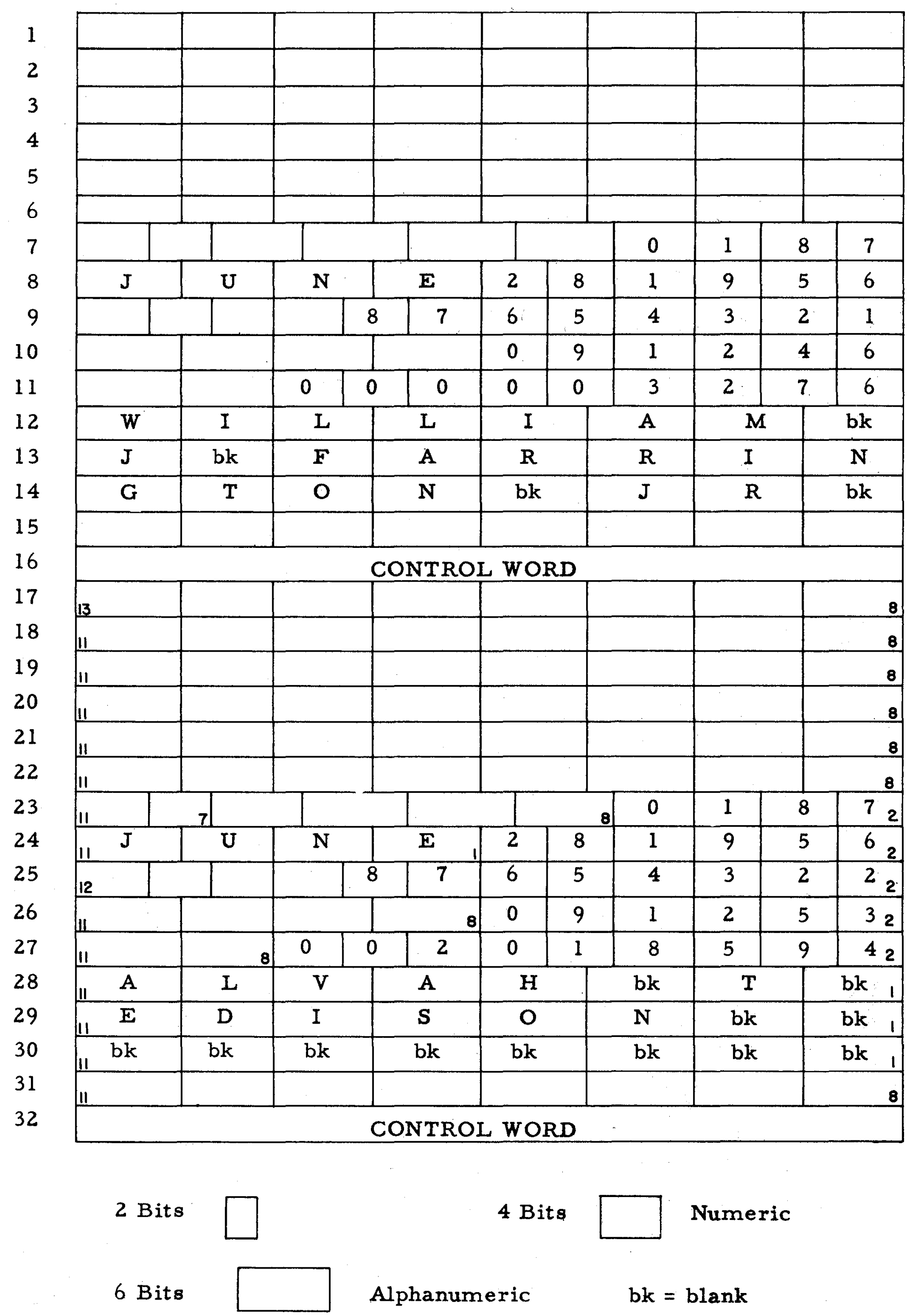

1
2
3
4
5
6
7 0 1 8 7
8 J U N E 2 8 1 9 5 6
9 8 7 6 5 4 3 2 1
10 0 9 1 2 4 6
11 0 0 0 0 0 3 2 7 6
12 W I L L I A M bk
13 J bk F A R R I N
14 G T O N bk J R bk
15
16 CONTROL WORD
17 13 8
18 11 8
19 11 8
20 11 8
21 11 8
22 11 8
23 11 7 8 0 1 8 7 2
24 11 J U N E 1 2 8 1 9 5 6 2
25 12 8 7 6 5 4 3 2 2 2
26 11 8 0 9 1 2 5 3 2
27 11 8 0 0 2 0 1 8 5 9 4 2
28 11 A L V A H bk T bk 1
29 11 E D I S O N bk bk 1
30 11 bk bk bk bk bk bk bk bk 1
31 11 8
32 CONTROL WORD
2 Bits
4 Bits
Numeric
6 Bits
Alphanumeric
bk = blank

Figure 3-9

DATAmatic

A DIVISION OF MINNEAPOLIS-HONEYWELL REGULATOR COMPANY

Layout Form-Output Converter, Model 1300

Title..

Prepared by.. For Program..

By Programmer.. Checked by..

Date.. Remarks..

....

Modification.. ....Page..................of..............

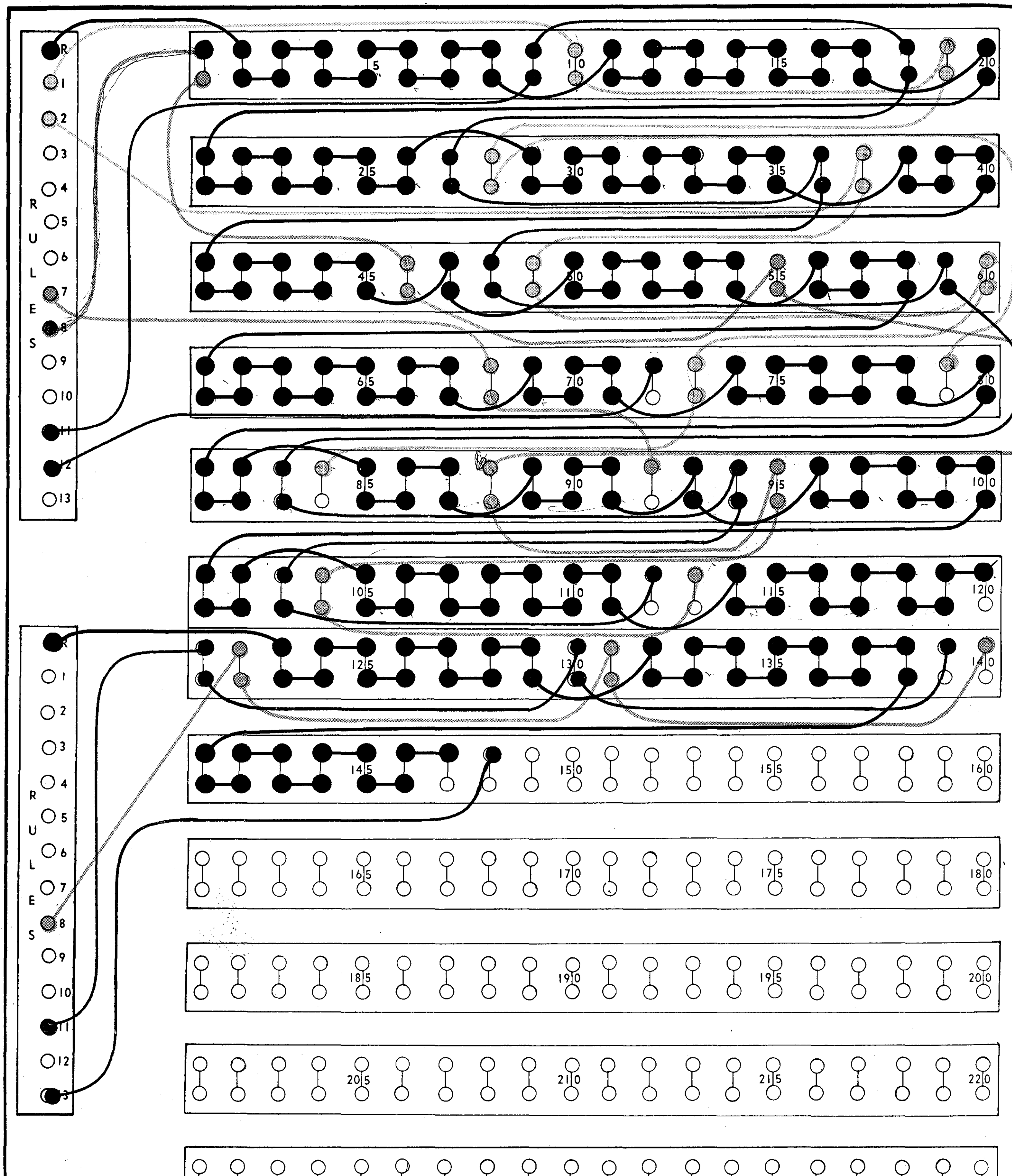
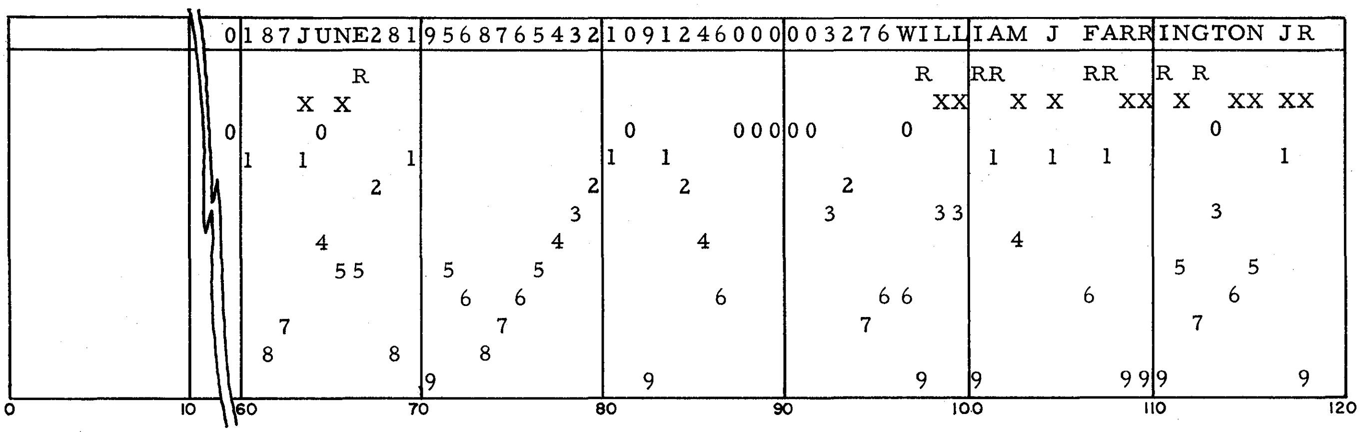
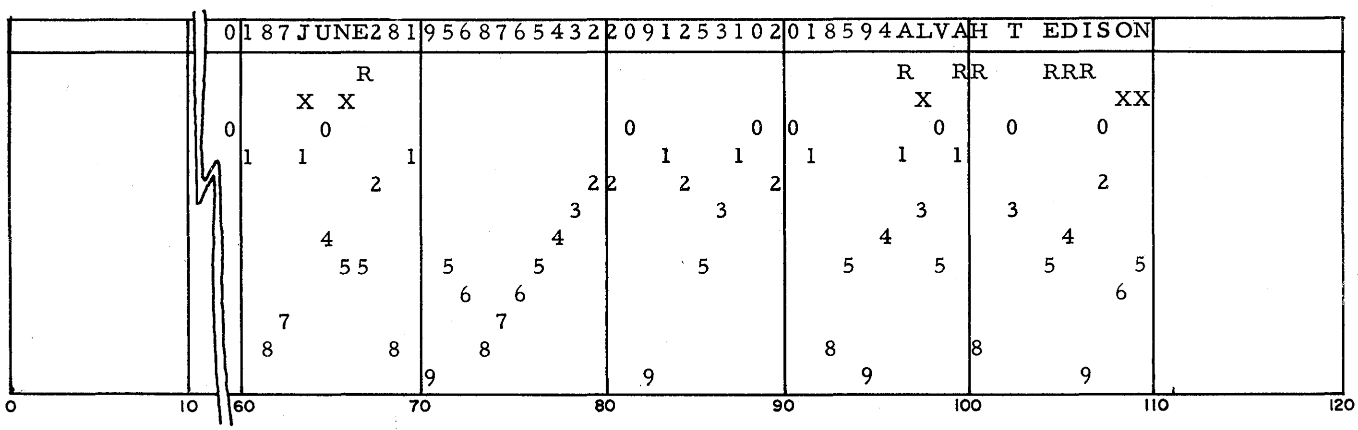


Figure 3-10



(a)



(b)

Figure 3-11

MODEL 1300 CONVERSION RULES AND CODES

Rule R: Repeat Rule
This rule should be wired from any step which is to repeat the rule executed by the previous step.

Rule 1: 6-bit Alphanumeric
Six bits are extracted from input storage and converted to an alphanumeric code as shown in the chart below.

TAPE	PUNCH CARD CODE RX0: R	X	0	1 2 3: 1	2	3	4 5 6: 4	5	6	7 8 9: 7	8	9	PRINT
000000			X										0 (zero)
000001				X									1
000010					X								2
000011						X							3
000100							X						4
000101								X					5
000110									X				6
000111										X			7
001000											X		8
001001												X	9
001010													Blank
001011						X					X		#
001100							X				X		@
001101													Blank
001110													Blank
001111													Blank
010000													Blank
010001	X			X									A
010010	X				X								B
010011	X					X							C
010100	X						X						D
010101	X							X					E
010110	X								X				F
010111	X									X			G
011000	X										X		H
011001	X											X	I
011010													Blank
011011	X					X					X		. (Dec pt, per.)
011100	X						X				X		□
011101													Blank
011110													Blank
011111													Blank
100000		X											- (Hyphen)
100001		X		X									J
100010		X			X								K
100011		X				X							L
100100		X					X						M
100101		X						X					N
100110		X							X				O (letter)
100111		X								X			P
101000		X									X		Q
101001		X										X	R
101010													Blank
101011		X				X					X		$ (dollar sign)
101100		X					X				X		* (asterisk)
101101													Blank

Rule 1: 6-bit Alphanumeric (cont.)

TAPE	PUNCH CARD CODE RX0: R	X	0	1 2 3: 1	2	3	4 5 6: 4	5	6	7 8 9: 7	8	9	PRINT
101110													Blank
101111													Blank
110000	X												&
110001			X	X									/ (slant bar)
110010			X		X								S
110011			X			X							T
110100			X				X						U
110101			X					X					V
110110			X						X				W
110111			X							X			X
111000			X								X		Y
111001			X									X	Z
111010													Blank
111011			X			X					X		, (comma)
111100			X				X				X		% (percent)
111101													Blank
111110													Blank
111111													Blank

Rule 2: 4-bit Hexadecimal
Four bits are extracted from input storage and converted to a hexadecimal code as shown in the chart below.

TAPE	PUNCH CARD CODE RX0: R	X	0	1 2 3: 1	2	3	4 5 6: 4	5	6	7 8 9: 7	8	9	PRINT
0000			X										0
0001				X									1
0010					X								2
0011						X							3
0100							X						4
0101								X					5
0110									X				6
0111										X			7
1000											X		8
1001												X	9
1010	X				X								B
1011	X					X							C
1100	X						X						D
1101	X							X					E
1110	X								X				F
1111	X									X			G

Rule 3: 4-bit Sign (Card Punch Function)
Four bits are extracted from input storage and converted, under control of the 3-position SIGN switch on the console panel, into the punch code shown in the following chart.

Table 3-1, page 1

SIGN SWITCH	TAPE	PUNCH CARD CODE R	PUNCH CARD CODE X
Pos.-blank Neg.-X	0101		X
	all other codes	(no punch)	
Pos.-blank Neg.-R	0101	X	
	all other codes	(no punch)	
Pos.-R Neg.-X	0101		X
	1101	X	
	all other codes	(no punch)	

4-bit Sign (Tabulator Function)

The position of the SIGN switch is irrelevant. Four bits are extracted from input storage. A minus sign (hex 5) in input storage is converted to a punch code 9 in output storage; any other code in input storage is converted to a punch code blank in output storage. Proper wiring of the tabulator control panel will accomplish the printing of plus and minus signs, credit and debit symbols, etc., as desired. If the storage column were wired directly to the Print entry on the tabulator control panel, plus and minus tape codes would be printed as blanks and 9's, respectively.

Rule 4: 6-bit Month

The use of this rule is restricted to card punch operation. Six bits are extracted from input storage and converted to a month code as shown in the chart below. Note that in tabulator operation, month conversion is accomplished by executing two successive 4-bit hexadecimal rules.

TAPE	R	X	0	1	2	3	4	5	6	7	8	9	
000001				X									
000010					X								
000011						X							
000100							X						
000101								X					
000110									X				
000111										X			
001000											X		
001001												X	
010000			X										
010001		X											
010010	X												
all others	(no punch)												

Rule 5: 1-bit Conversion

One bit is extracted from input storage and is converted as shown in the chart below.

TAPE	R	X	0	1	2	3	4	5	6	7	8	9	PRINT
0			X										0
1				X									1

Rule 6: 1-bit Reject

One bit is extracted from input storage and discarded.

Rule 7: 4-bit Reject

Four bits are extracted from input storage and discarded.

Rule 8: 6-bit Reject

Six bits are extracted from input storage and discarded.

Rule 9: 1-bit Reject (Special Control)

One bit is extracted from input storage and discarded. A "one" bit in this position activates a special control exit on the control panel of the tabulator or card punch.

Rule 10: 4-bit Sign (Special Control)

Four bits are extracted from input storage and converted according to rule 3. A minus sign (hex 5) code activates a special control exit on the control panel of the tabulator or card punch.

Rule 11: End of Word (Not one or nine)

This rule must be executed at the end of each word except words one and nine. No conversion takes place.

Rule 12: End of Word Nine

This rule must be executed at the end of word nine. No conversion takes place.

Rule 13: End of Word One

This rule must be executed at the end of word one. No conversion takes place.

Notes:

1. As many as 16 steps may be wired to execute a special control function (rule 9 or 10); i.e., a maximum of 16 special control selections may be made.
2. The Repeat rule (rule R) may be wired as many times as required.
3. All other rules may be wired as many as 34 times from each hub; i.e., any of the rules 1 through 8 may be wired 34 times in the first 120 steps and 34 times after step 120.
4. The upper set of rules hubs may only be wired from steps 1 through 120 and the lower set may only be wired from steps 121 through 240.
5. The repeat rule may not be used to repeat rule 9 or rule 10.
6. Step 121 may be wired from the lower Repeat hub if desired.

SECTION IV

MODEL 1400 OUTPUT PRINTING SYSTEM

SECTION IV—MODEL 1400 OUTPUT PRINTING SYSTEM

INTRODUCTION

The DATAmatic Model 1400 Output Printing System consists of a Converter Control Unit and a 900-line-per-minute Printer. The Converter Control Unit accepts information recorded on magnetic tape and generates electrical signals which cause the printing of this information. At the same time, it performs extensive editing and format arrangement functions. Its operation is completely independent of the Central Processor. One Magnetic File Unit is assigned to read the information from the output tape. This unit may be switched to communicate with the Central Processor when the Output System is not in operation.

The operation of the Output Printing System may be summarized with reference to the flow diagram of Figure 4-1. Information is read from a block on magnetic tape into the Converter Control Unit eight channels (16 words) at a time. The processing and printing of each 8-channel group (called a blockette) constitutes one print cycle and results in one line of print. Each blockette contains the information which forms the contents of the line of print and, normally, a control word which defines the parameters of the conversion, editing, and layout functions. (An alternative mode of operation will be discussed in which the control word is generated internally.) The eight channels comprising a blockette may be any permutation of any eight channels in the tape block. Fifteen such permutations of channels (called channel selections) are wired on the *Tape Channel Selection Control Panel.* It is stressed that the channels so selected may come from any part of the tape block in any order, and that the same channels may be reused in the same selection, as well as in different selections.

A blockette of information, as defined by the Tape Channel Selection Control Panel, is stored initially in binary form in an array of magnetic cores called Tape Data Storage. The control work, either from tape or internally generated, is stored in the Control Word Register. A *Conversion Control Panel* selects specified groups of bits from Tape Data Storage and executes five conversion rules to convert them to 6-bit codes representing decimal digits, alphabetic characters, or special symbols. Conversion proceeds serially from the high-order bits of the first word to the low-order bits of the last word in Tape Data Storage. Each time that 48 bits have been removed from Tape Data Storage, their weight count is checked. Flexibility is enhanced by the inclusion of four conversion panels, permitting any one of four conversion programs to be used in processing a given blockette. The control word of a blockette selects the conversion panel to apply to that blockette.

The bit groups selected by the Conversion Control Panel are translated into printer codes by duplicate decoders and the results are compared. A *Horizontal Format Control Panel* edits the converted information and positions it in the Print Storage section. This control panel can be wired to transpose, duplicate, and delete converted information and also to emit characters and symbols not recorded on magnetic tape. Again, there are four such panels in the converter, of which one is made active for each blockette by the control word.

The Print Storage section stores each character (up to a maximum of 120) in 6-bit binary code for printing. The *Output Format Control Panel* specifies which of the 160 available print positions (up to a maximum of 120) will be active for a given run. A run consists of processing and printing all of the information which can be handled with a given set of control panels and with a given setting of the console panel switches. The same print positions remain active throughout the run.

The print drum is a constant-speed cylinder on which 56 axial rows of characters and symbols are embossed. Each row consists of 160 identical characters or symbols. During a print drum revolution, each row passes the printing surface and prints its characters in the specified print positions. One row will print all of the "E"s in the line, another row all of the "7"s, and another row all of the commas. Thus after a complete revolution, one line has been printed. Vertical spacing of printed lines is governed by an 8-channel, paper Format Control Tape, in conjunction with the control word. As the printing hammers are struck, a signal is fed back to compare the characters printed with the codes produced in the decoders.

Normally, the selection of the information to be processed, the selection of Conversion and Horizontal Format Control Panels, and several other control functions are performed by one of the 16 words read from tape. In addition to this type of operation, provision is made for operating the system under control of a *Manual Control Panel* which generates the information normally contained in the control word. In some cases, this manual control mode of operation minimizes the need for editing tapes in the Central Processor prior to output conversion and makes it possible to print the contents of completely unedited tape. Details of this type of operation are discussed on page 4-17 and illustrated in the example on page 4-34. When processing partially edited tape, it is possible to convert certain blockettes under control of the Manual Control Panel and then return control to one of the words on tape.

FLOW CHART MODEL 1400

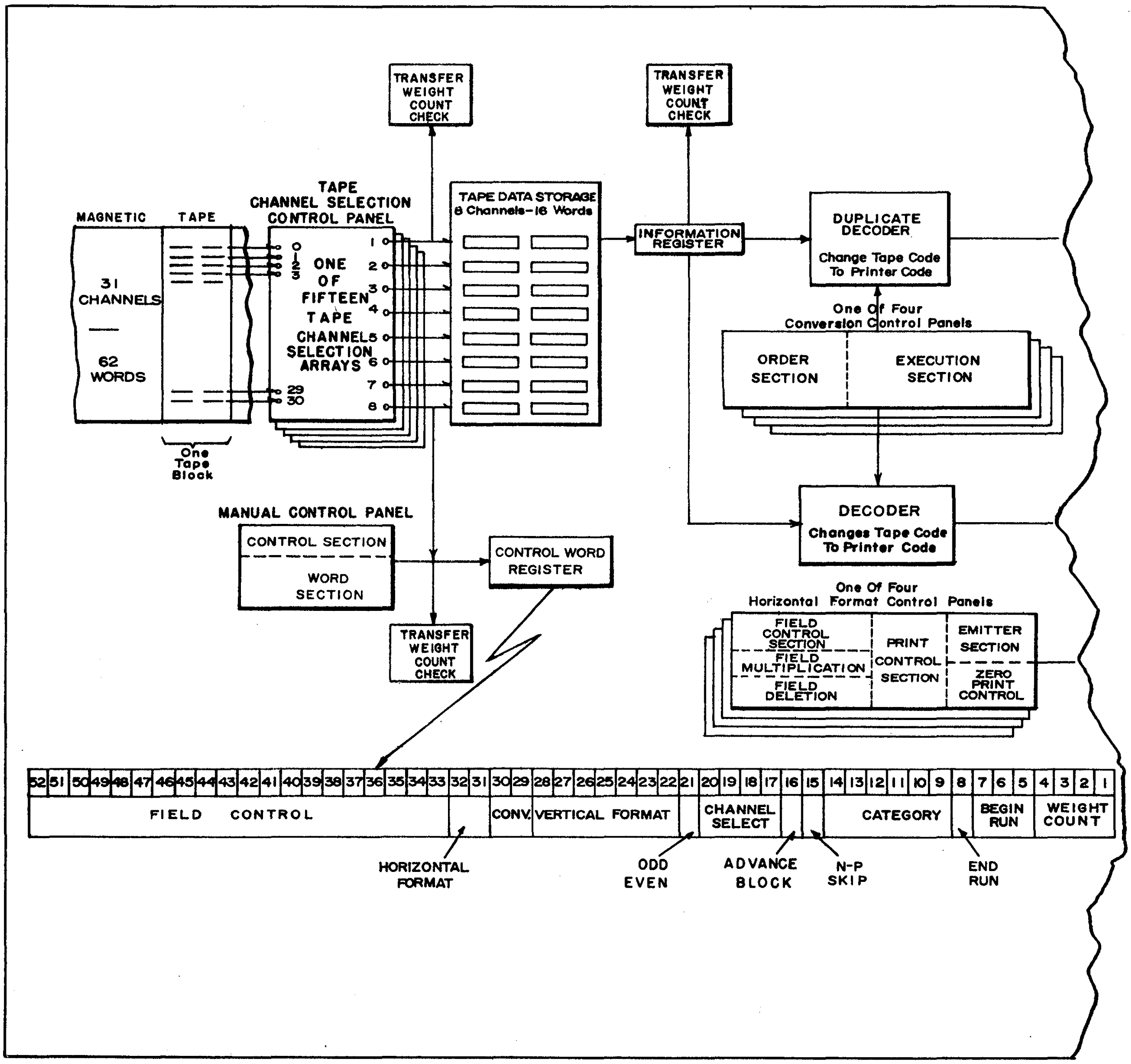


Figure 4-1

CONTROL WORD

Processing and printing of each blockette of information is governed by a control word which is either recorded on tape as a part of the blockette or wired on the Manual Control Panel. In addition, each control word designates the contents of the following blockette and the location of the following control word. The bits of the control word are divided into groups and each group performs a specific function, as shown in Figure 4-2. It is the responsibility of the programmer to plan the logical flow of control and to place the necessary control words in his tape layout. Note that control words may be located anywhere in the tape block, except that the first blockette of a run requires a control word in the "even" half of a tape channel. The wiring of the tape channel selection panel must place each control word in the eighth channel of the associated blockette. The form shown in Figure 4-2 may be used in planning the layout of control words.

OUTPUT PRINTING SYSTEM

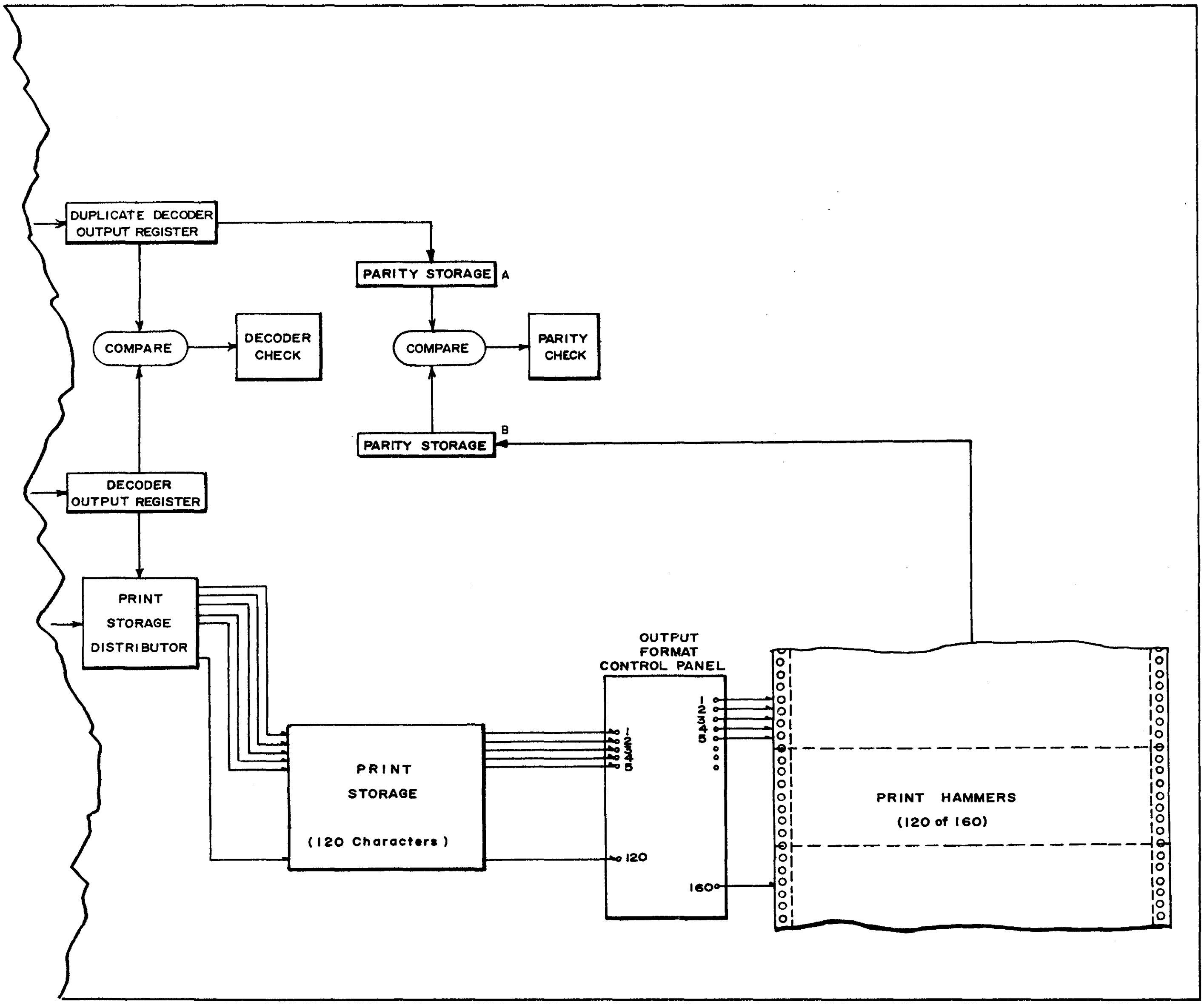


Figure 4-1

Weight Count (bits 1-4): These bits perform the check function described in Section I, page 1-3.

Begin Run (bits 5-7): Frequently, a number of different printing runs may be recorded on the same output tape. These runs may be numbered from one to six. The begin-run bits (5-7) are used to identify the first block of each run. Tape Channel Selection 1, the first 8-channel group wired on the tape channel selection panel, forms the first blockette of each run. The even word of the eighth channel in this selection is the control word for this blockette; in the case of the first block of a run, it is called the begin run control word. Bits 5-7 of the begin run control word contain the binary form of the number (from 1 to 6) which identifies the run. The configurations $\overline{000}$ and $\overline{111}$ are not used as begin run identification. The end run bit (8) must always be a "zero" in a begin run control word.

On the Converter Console Panel (see Figure 4-11, page 4-23), there is a rotary switch, called the Run Selector switch, which has six positions numbered from

CONTROL WORD LAYOUT

MCB PROG. STEPS (bits 21–16) | ← MEANINGLESS ON MCB → (bits 15–5)

← TRANSFER WEIGHT ZERO ON MCB → (bits 21–5)

Bit Position	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
Transfer Weight	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	
Control Word	FIELD SELECTION																				HORIZONTAL	FORMAT BD	CONVERTER	BOARD	VERTICAL FORMAT CHANNELS							ODD-EVEN	TAPE CHANNEL SELECTION				ADV. BLOCK	N-P SKIP	CATEGORY						END RUN	BEGIN RUN			WEIGHT COUNT				
	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1					7	6	5	4	3	2	1								6	5	4	3	2	1									
																																																					B
																																																					H
																																																					B
																																																					H
																																																					B
																																																					H
																																																					B
																																																					H
																																																					B
																																																					H
																																																					B
																																																					H
																																																					B
																																																					H

Remarks:

BIT USE, HFB #

20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1   7 6 5 4 3 2 1   6 5 4 3 2 1

Figure 4-2

1 to 6. The begin-run search is made by setting the MANUAL BOARD, MFU MANUAL CONTROLS, PRIMARY CATEGORY, and all STOP switches "off", setting the RUN SELECTOR switch to the number of the run to be printed, and then pressing the SYSTEM RESET button followed by the BEGIN RUN SEARCH button. The SEARCH DEMAND and TAPE IN MOTION indicators will light and the system will search the tape, examining the Tape Channel Selection 1 control word in each tape block. When a begin-run control word is located which has the desired run number encoded in bits 5-7, the system will position the tape at the start of this block and will stop with the SEARCH ENDED light illuminated. When all necessary switches are set and the START PRINT button is pressed, the converter will begin the printing run by reading Tape Channel Selection 1 and re-examining its control word for processing instructions.

Note that in the begin-run mode, the machine searches bits 5-7 of the specified word in every tape block until a "hit" occurs. If another run on the same tape uses this specified word for information storage, it is possible to obtain a false "hit" should the configuration being sought occur in bits 5-7 of one of these information words. Therefore, it is wise to print one form for visual inspection prior to continuing a printing run.

END RUN (BIT 8): To stop the system automatically at the end of a run, the programmer places a "one" in the eighth bit position of the last control word of the run. This control word must contain legitimate configurations in all of its bit groupings. If the END OF RUN switch is "on" when this control word is encountered, the line will be processed and the system will then stop with the END OF RUN light illuminated.

CATEGORY (BITS 9-14): These six bits are used for external selective suppression of printing and paper feeding. The settings of the PRIMARY CATEGORY and SECONDARY CATEGORY switches on the console panel (Figure 4-11) determine the manner in which the category bits control printing and paper feeding. The combination of the category bit configuration and the CATEGORY switch settings assign each blockette processed to one of these three operational categories:

(1) print one line, then advance paper,

(2) do not print but advance paper, or

(3) do not print or advance paper.

The PRIMARY CATEGORY switch is a rotary switch with seven positions labelled OFF, 1, 2, 3, 4, 5, and 6, respectively. When this switch is set to OFF, the system ignores the category bits (9-14) as well as the settings of the SECONDARY CATEGORY switches, prints each blockette processed, and advances paper according to the vertical format controls. The six "on" positions correspond to the six category bits; position 1 corresponds to bit 9, etc. If the PRIMARY CATEGORY switch is set for any "on" position, the corresponding category bit performs the following function:

(1) If the designated bit is a "zero", the current line is not printed, the paper is not advanced, the SECONDARY CATEGORY switch settings are ignored, and the non-print skip bit (15) is consulted to determine the source of the next blockette.

(2) If the designated bit is a "one", paper is advanced according to the vertical format controls, the non-print skip bit is *not* consulted, and the SECONDARY CATEGORY switch settings determine whether or not the line is printed.

The six SECONDARY CATEGORY toggle switches correspond to the six category bits in the same manner as do the six primary switch positions. If the SECONDARY CATEGORY switches are consulted and are all in the OFF position, the system ignores bits 9-14 and prints the line. If a SECONDARY CATEGORY switch is "on", the corresponding category bit performs the following function:

(1) If the designated bit is a "zero", the line is printed and the paper advanced.

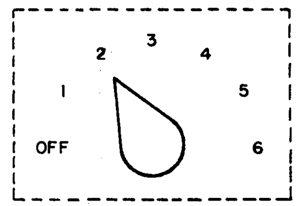
(2) If the designated bit is a "one", the line is not printed but the paper is advanced.

These six toggle switches may be set in any combination of ON and OFF positions. If more than one switch is "on", printing occurs only if a "zero" exists in each bit position corresponding to an "on" switch. A "one" bit corresponding to an "on" switch will suppress printing regardless of the remaining category bits and switch settings.

Figure 4-3 is a diagram of the actions which result from the various configurations of the category bits and switch settings. Several generalizations can be derived from the combined action of these rules. First, if the PRIMARY CATEGORY switch is in one of the "on" positions (1-6) and the SECONDARY CATEGORY switches are all "on", no printing can be done regardless of the contents of the category bits, although paper may or may not be advanced. If the SECONDARY CATEGORY switches are all "off", every line accepted for processing at the primary level will be printed. A maximum of five of the six category bits may be used to control printing at the secondary level since one bit will have been used at the primary level.

The bit groups Non-Print Skip (bit 15), Advance Block Control (bit 16), Tape Channel Selection (bits 17-20), and Odd-Even Control Word (bit 21) differ from the other bit groups in the control word. Whereas all other controls govern the processing of the *current* blockette, these four controls direct the selection of the *following* blockette and its control word. As will be shown, the following blockette may consist of any one of 15 channel selections read from either the current tape block or the next tape block in the logically forward direction. The control word for this blockette may be

PRIMARY CATEGORY Switch



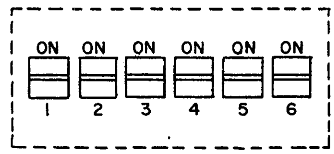
OFF

1	2	3	4	5	6
Read bit 9	Read bit 10	Read bit 11	Read bit 12	Read bit 13	Read bit 14

If "zero", do not print, do not advance paper, ignore SECONDARY CATEGORY switches, consult Non-Print Skip (bit 15).

If "one", advance paper, do not consult Non-Print Skip (bit 15), print unless suppressed by SECONDARY CATEGORY switches.

SECONDARY CATEGORY Switches



1	2	3	4	5	6
bit 9	bit 10	bit 11	bit 12	bit 13	bit 14

If OFF, ignore corresponding category bit.

If ON, consult corresponding category bit.

If all consulted bits are "zero", print and advance paper.

If any consulted bit is "one", do not print but advance paper.

Print line,

Advance paper

Figure 4-3

either word in the eighth channel of the designated channel selection.

Non-Print Skip (bit 15): The non-print skip bit is effective only when a line has been suppressed at the primary category level. Following a "miss" between the PRIMARY CATEGORY switch setting and the corresponding category bit, the non-print skip bit governs the selection of the following blockette according to these two rules:

(1) If the non-print skip bit is a "zero", the next blockette is selected in accordance with the Advance Block Control (bit 16) and the Tape Channel Selection control (bits 17-20).

(2) If the non-print skip bit is a "one", bits 16 and 17-20 are ignored and the next blockette is selected from the next block on tape according to Tape Channel Selection 1, the even control word.

Advance Block Control (bit 16): A "zero" in this bit designates that the next blockette shall be read from the same tape block as the current blockette. A "one" in this bit directs that the next blockette shall be read from the next logically forward tape block.

Tape Channel Selection (bits 17-20): The tape channel selection bits of the *current* control word specify the eight channels to comprise the *following* blockette. The only exceptions to this statement are the non-print skip and the manual control modes of operation. The 15 configurations $\overline{0001}$ through $\overline{1111}$ in bits 17-20 each designate the desired tape channel selection (from 1 to 15) in binary form. The configuration $\overline{0000}$ transfers control to the Manual Control Panel, discussed on page 4-17.

Odd-Even Control Word (bit 21): When the control word associated with a blockette is read from tape, it must reside in the eighth channel of the blockette. Bit

21 of the *current* control word specifies which word of this channel will control the printing of the *following* blockette. A "one" in bit 21 designates the odd word; a "zero" designates the even word.

VERTICAL FORMAT CONTROL (BITS 22-28):These bits, together with an 8-channel, paper Format Control Tape, govern the vertical spacing between printed lines. Two modes of vertical spacing are provided and the particular mode to be used is designated by bit 28. A "zero" in bit 28 specifies the normal mode of operation. In this case, bits 22-27 correspond, respectively, to channels 1-6 of the format control tape. A "one" in any of these bits designates the active format control channel and the paper is advanced until the format control mechanism senses the next punched hole in this channel. Only one control channel may be made active by each control word. A "one" in bit 28 specifies the lines-count technique of vertical spacing, in which format control channel 7 is active. In this mode, bits 22-27 represent a pure binary number from 0 to 63. When the lines-count technique is indicated, the paper is advanced until as many punched holes are sensed in channel 7 as are designated by the number in bits 22-27. The format control tape is illustrated and described further on page 4-20.

CONVERSION CONTROL PANEL (BITS 29-30): Four separate Conversion Control Panels (numbered from 1 to 4) are included in the Converter Control Unit, each of which may contain a complete conversion program. The number of the active conversion panel for a given blockette is specified in binary form in bits 29-30 of the control word (the configuration $\overline{00}$ representing panel 4).

HORIZONTAL FORMAT CONTROL PANEL (BITS 31-32): There are four Horizontal Format Control Panels (numbered from 1 to 4) in the Converter Control Unit, each capable of governing a complete horizontal format. The number of the active horizontal format panel for a given blockette is specified in binary form in bits 31-32 of the control word (the configuration $\overline{00}$ representing panel 4).

FIELD SELECTION (BITS 33-52): The Horizontal Format Control Panel divides data into fields, from 1 to 20 of which may be laid out in a single line of print. A field is any group of characters which is to be printed in the same sequence as it is converted. Fields are processed sequentially, using the characters in the order in which they are removed from Tape Data Storage. Bits 33-52 correspond to fields 1 through 20, respectively. The location of fields is specified on the horizontal format panel using sets of field control hubs. These sets include "1" hubs, "0" hubs, and "u" or unconditional hubs. A "one" in any of the field selection bits specifies that during the processing of the corresponding field, all "1" hubs and all "u" hubs will be active. Similarly, a "zero" in any field selection bit specifies that during the processing of the corresponding field, all "0" hubs and all "u" hubs will be active.

In summary, the following bit groups govern the processing of the blockette in whose control word they occur:

Bits	Function
5 - 7	Begin Run
9 - 14	Category
29 - 30	Conversion Control Panel
31 - 32	Horizontal Format Control Panel
33 - 52	Field Selection

The following bit groups govern the selection of the *following* blockette and its control word:

15	Non-Print Skip
16	Advance Block Control
17 - 20	Tape Channel Selection
21	Odd-Even Control Word

The following bit groups govern the action of the system after the current blockette has been processed:

8	End Run
22 - 28	Vertical Format Control

TAPE CHANNEL SELECTION CONTROL PANEL

The Tape Channel Selection Control Panel, shown in Figure 4-4, contains two sections of exit hubs, labelled "Magnetic File Unit System Channels", and two sections of entry hubs, labelled "Blockette". The "Blockette" hubs are arranged in 15 groups, each group consisting of eight pairs of commoned hubs. Each "Magnetic File Unit" section contains 31 hubs (numbered from 0 to 30) representing the 31 tape channels.

The upper group of MFU Channel hubs is used to form Channel Selections 1 through 8 and the lower group to form Channel Selections 9 through 15. Channel Selection 1 is wired by connecting eight of the MFU Channel hubs to the eight "Converter Input Channel" positions in the first blockette group. Subsequent channel selections are then wired in the same manner. When a channel is wired to more than one blockette from the same MFU Channel group, chain wiring is used between the common-entry hubs, as shown in Figure 4-4. The following rules govern the wiring of the Tape Channel Selection Control Panel:

(1) Within each channel selection used, all eight channels must be wired. Undesired channels wired to comply with this requirement may be deleted later.

TAPE CHANNEL SELECTION CONTROL PANEL

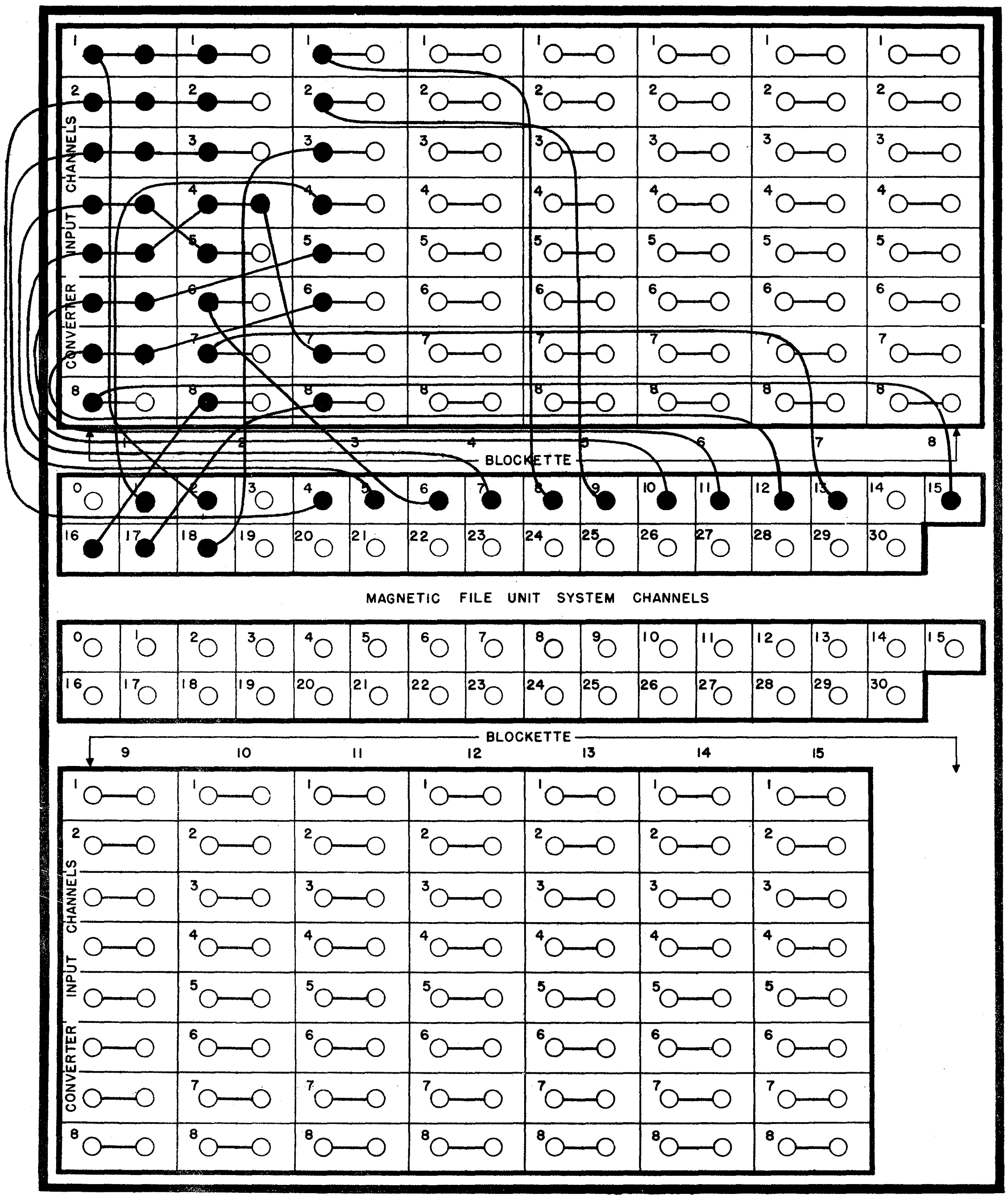


Figure 4-4

(2) Any channel may be used up to three times within a given channel selection. Furthermore, the same channel may be used up to three times in each channel selection wired.

(3) Channel 8 of each channel selection must contain the control word of that blockette if the control word is to come from tape.

(4) After the first eight channel selections are wired, all chain wiring is terminated. Channel Selections 9 through 15 are wired from the lower section of MFU Channel hubs.

The control panel shown in Figure 4-4 has been partially connected to demonstrate the method of wiring. Tape Channel Selection 1 consists of channels 2, 4, 5, 7, 10, 11, 12, and 15. Channel Selection 2 comprises channels 2, 4, 5, 10, 7, 6, 13, and 16, while Channel Selection 3 comprises channels 8, 9, 18, 1, 11, 12, 10, and 17, respectively.

As previously mentioned, Channel Selection 1 always assembles the first blockette of any printing run, except when starting in the manual control mode. Bits 17-20 of the control word for this blockette specify one of the 15 channel selections to assemble the second blockette, etc., for the remainder of the run. Once one of the channel selections has been specified for a blockette, the other 14 selections become inactive during the processing of that blockette. The channels selected are read from high-order to low-order bits and stored in the Tape Data Storage section.

CONVERSION CONTROL PANEL

Figure 4-5 shows the layout of the Conversion Control Panel, which is used to control the conversion of information from tape code into printer code and to discard unwanted information. Each conversion program must convert or discard an integral number of 48-bit words. Weight count bits are never processed but are automatically reserved to check each 48-bit group extracted from Tape Data Storage. Five rules may be used in the conversion process.

(1) Six-bit Alphanumeric: Six bits are removed from Tape Data Storage and sent to Print Storage. The 6-bit codes may represent any character capable of being printed. For details of these codes see Table 4-1, page 4-42.

(2) Four-bit Hexadecimal: Four bits are removed from Tape Data Storage and converted to the 6-bit code representing the corresponding hexadecimal character (0 through 9 and B through G). Details of these codes are shown in Table 4-1, page 4-42.

(3) Four-bit Sign: Four bits are removed from Tape Data Storage and discarded after the high-order bit has been used to set a flip-flop circuit into either a "plus" or a "minus" state. A "one" bit represents a plus sign and a "zero" bit represents a minus sign. The state of the Sign Flip-flop is accessible on all four horizontal format panels. In wiring these panels, the state of this flip-flop may be used for any desired control purpose, including the insertion of "plus" and "minus" identifications.

(4) One-bit Sign: One bit from Tape Data Storage is converted to the 6-bit code for plus or minus; a "one" represents "plus" while a "zero" represents "minus".

(5) One-bit Numeric: One bit from Tape Data Storage is converted to the 6-bit code for "0" or "1".

Three rules may be used to discard data:

(1) 52-bit Discard: A complete word is discarded from Tape Data Storage. This rule must not be executed while a partial word remains in Tape Data Storage.

(2) One-bit Discard: A single bit is discarded from Tape Data Storage.

(3) End Conversion: The final instruction of any conversion program must be an end conversion instruction. This instruction must not be executed when a partial word remains in Tape Data Storage. (A partial word should be eliminated by the use of the 1-bit discard instruction.) The end conversion instruction causes the rapid rejection of all remaining whole words from Tape Data Storage and the initiation of printing. The step which executes end conversion is not wired to a repetitions hub.

One 52-bit discard rule discards an entire word, including weight count. A one-bit discard repeated 52 times discards 48 information bits, the attached 4-bit weight count, and four bits from the following word.

Information from Tape Data Storage is converted in steps and delivered sequentially in 6-bit codes to Print Storage. Each step is wired by specifying one of the conversion or discard instructions and the number of times this instruction is to be executed. Any conversion instruction or the 1-bit discard instruction may be executed up to 63 times in a single step. The 52-bit discard instruction may be executed up to 15 times in a single step. The end of the last word containing information to be printed should be followed immediately by the end conversion instruction. The system alarm circuits will be disabled if an intervening 52-bit discard instruction is wired. A conversion program may consist of from 2 to 30 steps. End conversion must not be wired to the first step of a program. Information is processed high-order first, i.e., the first step converts the high-order bit(s) of the first word in Tape Data Storage and succeeding steps proceed toward the low-order bit of the last word in Tape Data Storage.

The upper half of the conversion panel (Figure 4-5) consists of a "Conversion Steps" zone and an "Orders" zone. The "Conversion Steps" zone contains 30 pairs of commoned hubs, corresponding to the maximum of 30 steps in a conversion program. The "Orders" zone contains one hub for each of the converter instructions. Each instruction used is wired to the first step in which it is used. The commoned hub for this step may then be wired to a later step if the instruction is to be reused.

The lower half of the conversion panel consists of a "Conversion Steps" zone and a "Repetitions" zone. Each step which is used on the upper half must be wired in the "Conversion Steps" zone to a repetitions hub to designate the number of times that the instruction is to be performed. Note that consecutive repetitions (up to 63) are specified by the wiring of these lower zones, while repetitions in excess of 63, as well as non-consecutive repetitions, are achieved by chain wiring. If a step is wired to an "Orders" hub but not to a "Repetitions" hub, the specified instruction is performed 64 times. An electronic stepping switch activates the various "Conversion Steps" hubs in succession.

The conversion program represented by the example of Figure 4-5 converts a blockette in which words 1 and 2 are alphanumeric; word 3 contains 24 unused bits, one 1-bit sign, three more unused bits, and five hexadecimal digits; words 4-6 are not used; and word 7 is alphanumeric. Step 1 specifies that the alphanumeric conversion instruction is to be performed 16 times (two words). Step 2 performs the 1-bit discard 24 times. Step 3 performs one 1-bit sign conversion and step 4 discards three more bits. Five hexadecimal digits are converted in step 5. In step 6, words 4 through 6 are discarded. The alphanumeric conversion rule is chain wired from step 1 to step 7 to convert the seventh word. Step 8 is the end conversion rule.

There are four Conversion Control Panels in the Converter Control Unit, each of which may contain a complete conversion program. The conversion panel containing the desired program to convert a specific blockette is activated by bits 29-30 of the control word.

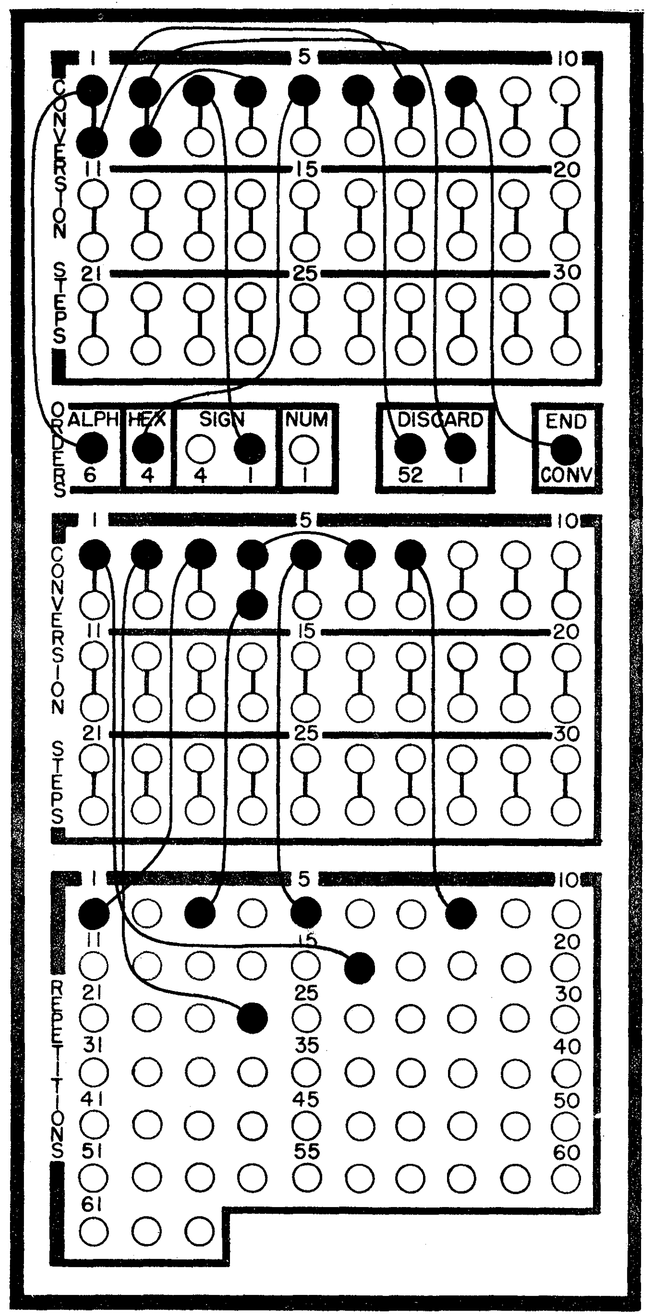


Figure 4-5

HORIZONTAL FORMAT CONTROL PANEL

The major purpose of the Horizontal Format Control Panel (shown in Figure 4-6) is to position all of the converted information in the 120 print storage positions. In addition, this panel controls deletion, field multiplication, character emission (with or without the deletion of data), and zero suppression or substitution. There are four horizontal format panels in the Converter Control Unit, each capable of defining a different horizontal line format. The panel to be used for a particular blockette is specified by bits 31 and 32 of the associated control word. The selection of the horizontal format panel is independent of the selection of the conversion panel. When any one panel has been activated, the other three panels become inactive for that particular printing cycle.

In general, the horizontal format panel treats data in fields, of which as many as 20 may be established within a given line. A field is any specified group of characters, regardless of length, which, excepting character emissions, are drawn from Tape Data Storage in order and which retain this order during horizontal format processing. Twenty bits in the control word (bits 33-52) are designated for conditional control of individual fields. Any, but not all, of these bits may be used in a given format for special control purposes. Fields are processed on a sequential basis.

FIELD POSITIONING: A field may be placed in any set of consecutive print storage positions. This is accomplished by proper wiring of the Print Control A and Print Control B hubs, the Field hubs, and the End-of-Field hubs (see Figure 4-6). Each Print Control zone contains a hub to correspond to each of the 120 print storage positions. The Print Control A hubs are used to define the starting positions of the fields; Print Control B hubs are used to define the positions where fields end and those where special functions are executed. The Field zone contains a set of hubs to correspond to each of the 20 field selection bits or to each of the 20 possible fields of data in a printed line. Each set consists of a "one" hub, a "zero" hub, and two unconditional (u) hubs. Unless bit 33 is to be used for a special control function, field 1 is established by wiring from one of the Field 1 hubs to the position in Print Control A where it is desired to start the field and from the position in Print Control B where it is desired to end the field to an End-of-Field hub. (Note that the first field which is used to extract information from Tape Data Storage will contain the first information converted, although it may not necessarily be the left-most field in the printed line.)

Two electronic stepping switches work in conjunction with these hubs. These are the 20-position Field Control Stepper and the 120-position Print Position Stepper. The field control stepper is automatically set to stage 1 at the beginning of each print cycle and remains in this stage until it is indexed to the next consecutive stage by an end-of-field impulse or a special end impulse. Stage 1 corresponds to field 1 and to bit 33 of the control word; stage 2 corresponds to field 2 and to bit 34, etc. While the field control stepper is in stage 1, the "one" hub of field 1 is active and the "zero" hub is inactive if bit 33 is a "1". If this bit is a "0", the "zero" hub is active and the "one" hub is inactive. In either case, the unconditional hubs are active. Thus, the field control stepper permits the control word to exercise conditional control over the lengths and contents of specific fields.

The 120-position print position stepper is activated by the print control A hubs. Whenever an impulse is received at a print control A hub from an active field hub, the print position stepper is set to a position corresponding to the number of the print control hub. Then the next character specified is sent to the indicated print storage position. The print position stepper then advances to the next consecutive position and the next character specified is sent to the following print storage position. This process continues until an end-of-field impulse is received. After the last character of the field has been sent to print storage, the field control stepper is advanced to the next stage. The print position stepper is then set to the position corresponding to the print control A hub which is wired to the next active field hub.

Each of the 120 hubs in the print control B zone is activated when the print position stepper is in the corresponding position. If a print control B hub is wired to a special function, that function will be performed when the print position stepper is in the indicated position. Similarly, if a print control B hub is wired to an active end-of-field hub, a field will be ended and the field control stepper advanced. In the case of the last field of a print cycle, the proper print control B hub is wired to an end-horizontal-format hub rather than to an end-of-field hub. If end-horizontal-format is reached while any 6-bit alpha, 4-bit hex, 1-bit numeric, or 1-bit sign rules remain to be executed, an indicator will light and the system will stop.

The end-of-field, end-horizontal-format, and many of the special function hubs are arranged in sets of "one" hubs, "zero" hubs, and unconditional hubs to permit conditional control of their respective functions by the control word. As in the case of the field hubs, the "one" hubs are active and the "zero" hubs inactive during the processing of any field whose selection bit is a "1". If the selection bit is a "0", the "zero" hub is active and the "one" hub inactive. In either case, the unconditional hubs are active. These sets differ, however, from the field hubs in that they are activated by

HORIZONTAL FORMAT CONTROL PANEL

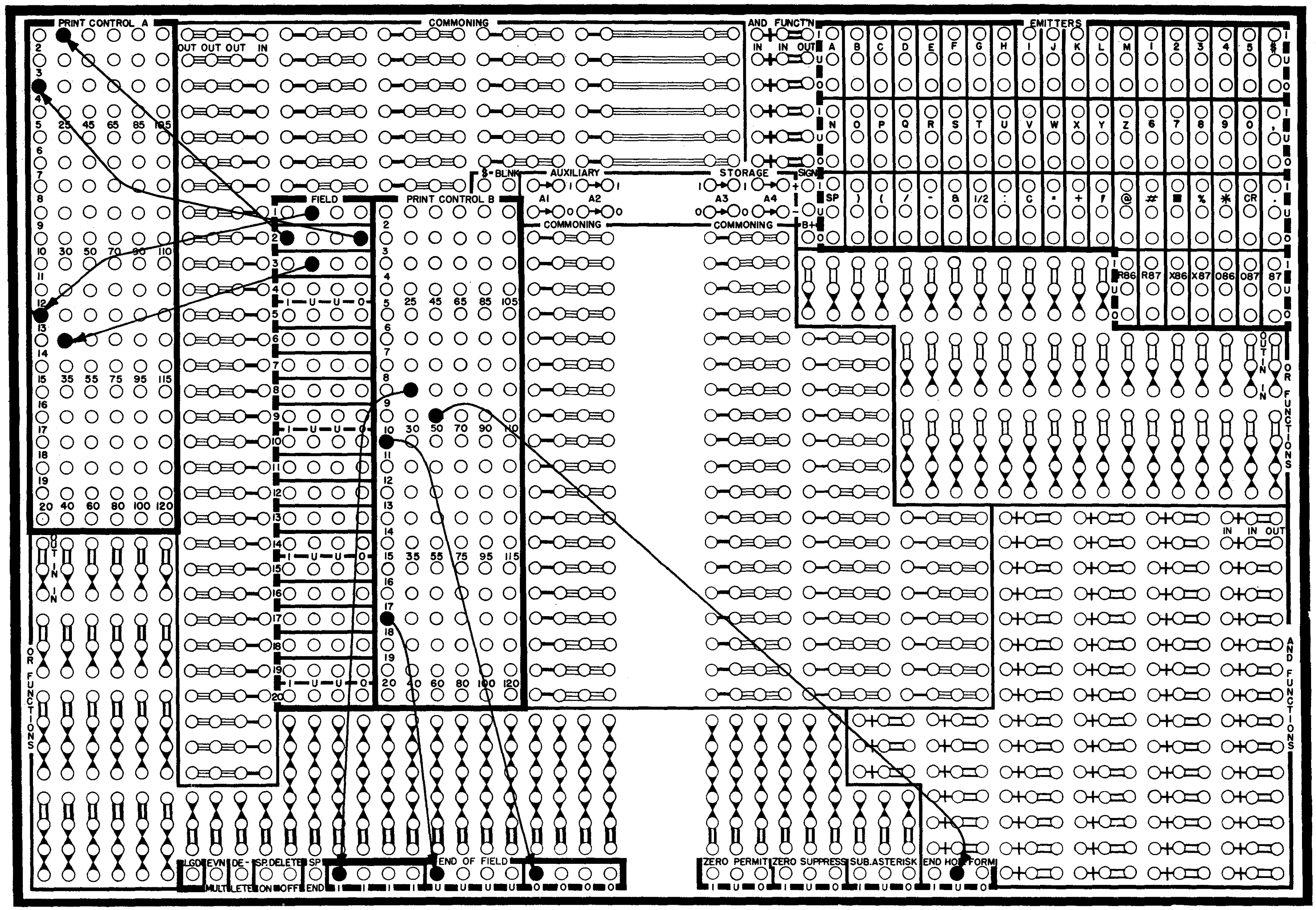


Figure 4-6

their field selection bits to receive an impulse, while the field hubs are activated by their field selection bits to transmit an impulse.

The alternate stepping of the field control stepper and the setting and stepping of the print position stepper removes converted information from Tape Data Storage and arranges it in print storage. Wiring is always done *from* the field hubs *to* the print control A or special function hubs and *from* the print control B hubs *to* the end-of-field, end-horizontal-format, or special function hubs.

EXAMPLE: Figure 4-6 shows the wiring to perform a simple example involving three fields of data. Field 1 is wired to start unconditionally at print storage position 12 and to end unconditionally at print storage position 17. If the field selection bit governing field 2 (bit 34) is a "1", this field will be stored in storage positions 21-28; however, if this bit is a "0", field 2 will be stored in storage positions 3-10. Note that in this case, the order in which data is stored in Tape Data Storage has been altered by the horizontal format panel. Field 3 is wired to be stored unconditionally in storage positions 33-49 and ends with an unconditional end-horizontal-format signal. In this example, the field selection bits exercise only one option, namely the positioning of field 2. In general, most horizontal line formats will involve other operations, such as emission, deletion, and field multiplication.

OR FUNCTION (CONVERGENCE) GATES: Frequently, a special function is to be executed more times than the number of hubs provided. A special type of gate, labelled an OR FUNCTION and represented by the above symbol, permits the necessary multiple usage of such functions. OR gates, also known as Convergence Hubs, are connected so that a signal received at any of the "in" hubs is available at the "out" hub but not at any other "in" hub. They are always used when wiring from several points to a single hub, hence the name "convergence hubs". The control hubs along the lower edge of the control panel frequently require the use of OR gates. For example, when it is necessary to connect more than four print control B hubs to end-of-field hubs, they must be wired to "in" hubs of an OR gate and the "out" hub wired to end-of-field. To achieve a greater amount of convergence, the "out" hubs of several OR gates may be wired to "in" hubs or one or more other OR gates.

EMITTERS: There are 57 sets of emitter hubs on the Horizontal Format Control Panel, one set for each of the 56 print-wheel characters and one set to emit a space. Each set consists of a "one" hub and a "zero" hub, which are rendered active or inactive by the field selection bit for the field being processed, and an unconditional hub. (Seven additional character emitters are indicated on the control panel. These are not intended for current use by programmers.) To emit a character from the horizontal format panel for printing, the print control B hub corresponding to the desired print storage position is wired to the appropriate hub of the emitter set, through an OR gate if necessary. In this case, no character is converted from Tape Data Storage; the next character to be converted is delayed and placed in the following print storage position, provided that position is not also wired to an emitter. A field, then, consists of both converted and emitted characters and its length is determined by the wiring of the print control A and B hubs, in conjunction with the field and end-of-field hubs. No Tape Data Storage characters are lost through emission, but they are retained for later use. An emitted space is treated like any other emitted character.

LEGEND: When the system is in the Legend mode, any characters emitted *replace* information converted from Tape Data Storage. In other words, instead of the converted character being delayed, it is deleted, and the emitted character takes its place. The legend mode may be entered by wiring either from a field hub or from a print control B hub to the Legend hub. In either case, it takes effect for the remainder of the field being processed. Note that when entered from a field hub, the legend mode will be in effect during the processing of the entire field. If the legend hub is wired from a "one" hub of a field set, it has no effect on emitters wired to "zero" hubs, and vice versa. When the legend mode is operative, it has no effect on hubs to which no emitters are wired.

DELETE: A field may be deleted entirely or in part by using the Delete hub. To delete an entire field, the "one" or the "zero" hub of the field set is wired to the delete hub. In this case, the length of the field must be defined by connecting its beginning and ending positions with unconditional hubs. The field selection bit then determines whether the field is printed or deleted. If the delete hub is wired from a print control B hub, the character in that position and the remainder of the field are deleted. In either case, character deletion continues until ended by an end-of-field pulse.

SPECIAL DELETE: Deletion of a portion of a field or of more than one field may be accomplished by the use of the Special Delete hubs. Wiring from a field hub or a print control B hub to the Special Delete—On hub initiates the deletion of all characters until the process is halted by wiring from print control B to the Special Delete—Off hub. Thus, the special delete condition may be initiated at the beginning of a field or in the middle of a field, may continue through any number of fields, and may end at any point in a field. A special delete condition initiated on one horizontal format panel continues in effect, even though subsequent blockettes are processed by other horizontal format panels, until it is terminated by wiring to Special Delete—Off, not necessarily on the panel on which it was initiated. The print control position which is wired to Special Delete—Off will contain the next character processed after the deletion. Both the Delete and Special Delete functions may be selected through AND gates (see below).

Commoning (Divergence) Hubs: Commoning hubs are arranged in groups of four, one entry hub and three exit hubs, according to the above symbol. They are used to receive a signal from one hub and route it to several points (thus the term "divergence" hubs). Any signal wired to a commoning entry hub is available at each of the associated exit hubs. Care must be taken in wiring commoning hubs to insure that two exit hubs are not interconnected, as serious interference problems may result.

Field Multiplication: A character or group of characters in Tape Data Storage may be printed as many times as desired within a single line, provided that they are defined as a field. Commoning hubs are used in this function by wiring from the field hub to a commoning entry hub and wiring from the associated exit hubs to the print control A positions where the fields are to begin. Internal checking circuitry requires that a signal be brought to the Even Multiply hub whenever a field is multiplied an even number of times. For duplication of a field, two of the three exits are brought to print control A positions and the third to the even multiply hub. To triplicate a field, all three commoning exits are brought to print control A positions and no even-multiply signal is needed. To print a field four times, one exit from the first commoning set is wired to the entry of a second commoning set, providing five available commoning exits. Four of these are wired to print control A positions and the fifth to the even multiply hub. Note that for field multiplication purposes, commoning hubs are always used in complete sets and an unwired hub of a set used for this purpose is indicative of a wiring error.

If part of a multiplied field is deleted, the same characters are automatically deleted in each of the multiple fields. Similarly, if the multiplied field contains one or more emitted characters, these characters will appear in each of the multiple fields. However, if different characters are emitted to corresponding positions in multiplied fields, the actual codes generated will contain "ones" wherever there is a "one" in any of the emitted codes.

Zero Functions: Suppression of zeros begins automatically at the beginning of each field and continues until the first non-zero hexadecimal or first-quadrant character is converted (see Table 4-1, page 4-42). Zeros following this character are deemed significant and are printed where they occur until either (1) the field is ended, or (2) a second, third, or fourth-quadrant character is printed from tape, after which zero suppression is resumed. Zero suppression is not affected by emitted characters. Table 4-1 shows that the 56 print-drum characters are grouped in four quadrants, with the numeric digits in the first quadrant, the alphabetic characters in the second, third, and fourth quadrants, and special symbols in all four quadrants. Zeros which are suppressed are replaced by space characters for printing. Deleted characters exercise the same control over zero suppression and permission as printed characters.

Zeros may be suppressed or permitted, other than by the above rule, or asterisks may be substituted for non-significant zeros by means of the horizontal format panel wiring. To set up any of these conditions, the first print control B position to be affected is wired to the proper hub under Zero Suppress, Zero Permit, or Substitute Asterisk (according to the value of the current field selection bit). Wired zero suppression or permission is terminated by the same conditions as automatic zero suppression or permission. Asterisk substitution is terminated by any character other than a zero or an emitted comma. If it is terminated by a first-quadrant character, zero permission will ensue; if terminated by a second, third, or fourth-quadrant character, zero suppression will ensue.

AND Gates: AND gates, represented by the above symbol, consist of two entry hubs and one exit hub. The circuitry is so designed that a signal is available at the exit *only* if signals are simultaneously present at both entries. AND gates are usually used in connection with the Sign Flip-flop, the Auxiliary Storage Flip-flops, the field hubs or the print control B hubs to allow selective control of emission, field location, deletion, or the other special functions along the lower edge of the panel.

Sign Flip-flop: A flip-flop is an electronic circuit which remains stable in either of two states until it receives a signal which causes it to change to the opposite state. In the Sign Flip-flop, these two states are defined as representing "plus" and "minus". A 4-bit sign conversion in the conversion panel does not result in the formation of a 6-bit character as do all of the other conversion rules. Instead, the execution of this rule sets the Sign Flip-flop in the state which represents the indicated sign. No character is printed by this rule nor is the print position stepper advanced. This means that a sign has no place in the sequence of characters being converted, but is set apart to be drawn on at a subsequent point in the sequence.

The Sign Flip-flop normally remains in a given state until a 4-bit sign rule reads the opposite sign configuration. However, for continuous use of a given state of the Sign Flip-flop, it is advisable to reinstate its value at least once for each line being printed.

If a step is wired for more than one execution of the 4-bit sign rule, the flip-flop will be set according to the first four bits read; subsequent 4-bit groups will be converted hexadecimally. If a 4-bit sign step is followed by another 4-bit sign step (or steps), the flip-flop will remain set according to the last step executed.

Since the 4-bit sign rule does not transmit a character to Print Storage, wiring must be done from the Sign Flip-flop to make use of this rule. (If it is not wired, no error indication will result.) If some symbol is to be sent to a given print storage position under control of the sign value, a hub of the Sign Flip-flop is wired into one AND gate entry, the desired print control B position is wired to the other AND gate entry, and the exit of the AND gate is wired to the desired symbol emitter. The exit of the AND gate might alternatively be brought to a zero-function hub or to an entry of another AND gate for selectively controlling other horizontal format panel functions according to the value of the sign.

The setting of the Sign Flip-flop may not be used to emit a character immediately before the execution of the next 4-bit sign rule, as the sign rule will take precedence and reset the flip-flop prior to the emission. Therefore, the flip-flop setting may only be used prior to the conversion step which immediately precedes the sign rule. In order to print sign information immediately before performing a sign rule, the flip-flop setting must be used earlier to set an auxiliary flip-flop which in turn may control the desired emission.

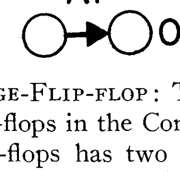
Floating Dollar Sign: An auxiliary emitter (marked $, above the print control B hubs) is provided for floating a dollar sign. A wire from any print control B hub to the Floating Dollar Sign hub causes a dollar sign to be inserted in the corresponding print storage position if the next character to the right is a numeral other than zero. If the next character to be converted is a non-significant zero, a blank is stored in the position from which the dollar sign is wired. The dollar sign is shifted automatically one place to the right and the following character is examined. As long as the next-right character is a non-significant zero, the dollar sign continues to be shifted to the right. The dollar sign is stored in the first position encountered which has a permitted zero or other numeral to the right of it. All commas to be emitted in conjunction with the floating dollar sign must be wired one print control B position prior to the point where they are to be used. An emitted comma following a non-significant zero is deleted. Zero permit should be wired from the print control B position which represents the farthest point to the right that the dollar sign will be printed. For example, to print $.05, wire zero permit from the position prior to the decimal point.

Floating Blank: Adjacent to the Floating Dollar Sign hub is a Floating Blank hub (marked BLNK). The latter is used to substitute blanks for emitted commas whenever zero suppression is in effect.

This floating ability applies only to the dollar sign and blank emitted from these two special hubs. Symbols emitted from the regular emitter hubs are stored only in the storage positions from which they are wired.

Special End: Each time a horizontal format panel is selected, the field control stepper is set to begin with field one. Each end-of-field impulse advances this stepper until an end-horizontal-format impulse is received. Special End effectively simulates an end-of-field impulse to advance the field control stepper without storing any characters. A number of completely different horizontal line formats can be established on a given horizontal format panel by the use of this function. This is illustrated in Figure 4-7. Here, if bits 33-35 (controlling fields 1-3) are "zeros", only these fields will be printed and the horizontal format will end at position 115. If these bits are "ones", only fields 4 and 5 will be printed and the horizontal format will end at position 117. In this example, bits 33-35 must be either all "zeros" or all "ones". Otherwise, more than one character might be sent to the same print storage position, which results in the formation of a code containing "ones" in each position where any processed character contained a "one".

If the unconditional hub for a field is wired to Special End, the corresponding bit is used for control purposes only. One or more of the other hubs in the same field set may then be wired to some control function, either directly or through an AND gate.



Auxiliary Storage-Flip-flop: There are four Auxiliary Storage Flip-flops in the Converter Control Unit. Each of these flip-flops has two entry hubs and two corresponding exit hubs. Whenever one of the entry hubs receives a signal, the corresponding exit hub is

HORIZONTAL FORMAT CONTROL PANEL

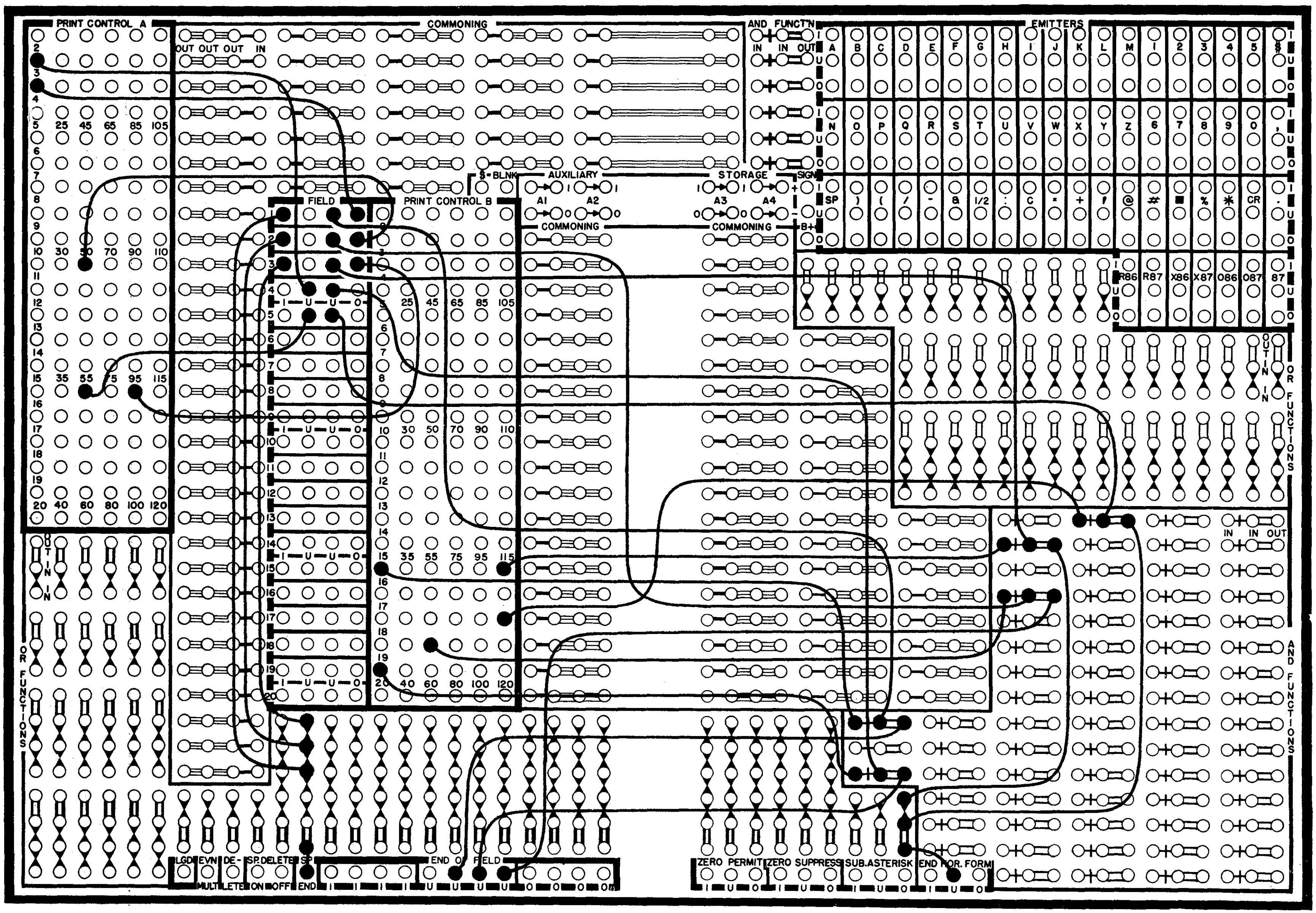


Figure 4-7

activated and remains active until a signal is received at the other entry. The auxiliary and sign flip-flop circuitry is common to all four horizontal format panels. Since they are *not* reset at the end of each printed line, a setting established during one print cycle will remain in effect during subsequent print cycles using other horizontal format panels, until a signal is received which reverses this setting. However, for continuous use of one setting, it is advisable to reinstate its value at least once for each line being printed. In most cases where they are used, these flip-flops receive entry signals from the Sign Flip-flop or from the field hubs, and their output signals are wired, via AND gates, to control printing of subsequent fields.

B+: The hub labelled B+ is connected to a steady positive voltage. It is used mainly for check purposes and will seldom concern the programmer. However, it may occasionally be used in programming. For example, if one of the control functions, such as Even Multiply, Zero Permit, or Zero Suppress, is to be in effect during the processing of an entire line, this function might be wired from the B+ hub.

CAUTION: Particular care must be taken not to connect two exit hubs, either directly or through a commoning set. Where more than one exit is to activate an entry, OR gates must be used rather than commoning hubs (as implied by convergence vs. divergence). The exit hubs on the horizontal format panel are (1) field hubs, (2) print control B, (3) outputs of AND gates, OR gates, and commoning hubs, (4) flip-flop exits, and (5) B+. *Wiring one exit to another will cause unsatisfactory performance and will seriously reduce the life of the circuits involved. Always check horizontal format panel wiring before operating the panel in the system.*

OUTPUT FORMAT CONTROL PANEL

There are 120 print storage positions, all of which may be active during a print cycle, and 160 print-drum positions. The Output Format Control Panel, shown in Figure 4-8, selects up to 120 print-drum positions and activates them for a given run by wiring them from print storage positions. Unwired print-drum positions will contain no printed characters and are used for spacing.

Each half of the output format panel consists of a 120-hub Print Storage zone and a 160-hub Output Column zone. The lower half is identical to the upper half and must duplicate all wiring in the upper half to satisfy the requirements of the converter checking circuitry. Each storage position containing a character to be printed is wired directly to the intended output column hub (in both halves of the panel). The hubs marked PRINT are not used in current applications of the output format panel. Each print storage position containing a converted or emitted character must be wired to an output column or an error signal will light. However, print storage positions wired on the output format panel need not contain information to be printed.

MANUAL CONTROL PANEL

As previously mentioned, the Manual Control Panel can be wired to generate control words in certain cases where it is desired to print from partially edited or unedited tapes. Up to eight control words may be wired on this panel, of the same form as those described under "Control Word", except that bits 5 through 15 are fixed in value and cannot be selected. When operating from a tape which contains no control words, the begin-run and end-run functions must be performed manually. Also, of course, the concept of category does not exist for the manual mode of operation since the category bits are fixed in value.

The manual control mode of operation proceeds in steps, under control of a 33-position stepping switch. Each step corresponds to a print cycle of normal operation in that it selects one blockette of information from tape and processes it to yield one line of print. When operating entirely in the manual mode, the 33 steps represent the maximum number of lines that can be printed before repeating the control sequence. The stepping switch is automatically set to 1 at the start of the printing run, advances one step as each line is printed, and returns to 1 according to the wiring of the Last Step hub. It is also possible to make a printing run using some control words from tape and others from the Manual Control Panel. In this case, more than 33 lines may be printed during a control sequence. The stepping switch advances one step for each manually controlled print cycle and one step each time that control is transferred to tape.

The lower portion of the panel contains eight double rows of hubs labelled Manual Control Words (see Figure 4-9). Each double row is capable of setting the Field Selection, Horizontal Format Panel, Conversion Panel, and Vertical Format areas of a different control word. A pair of hubs connected by a shunt wire in this area represents a "one" bit. Since all of these bit positions are automatically set at 0 at the start

OUTPUT FORMAT CONTROL PANEL

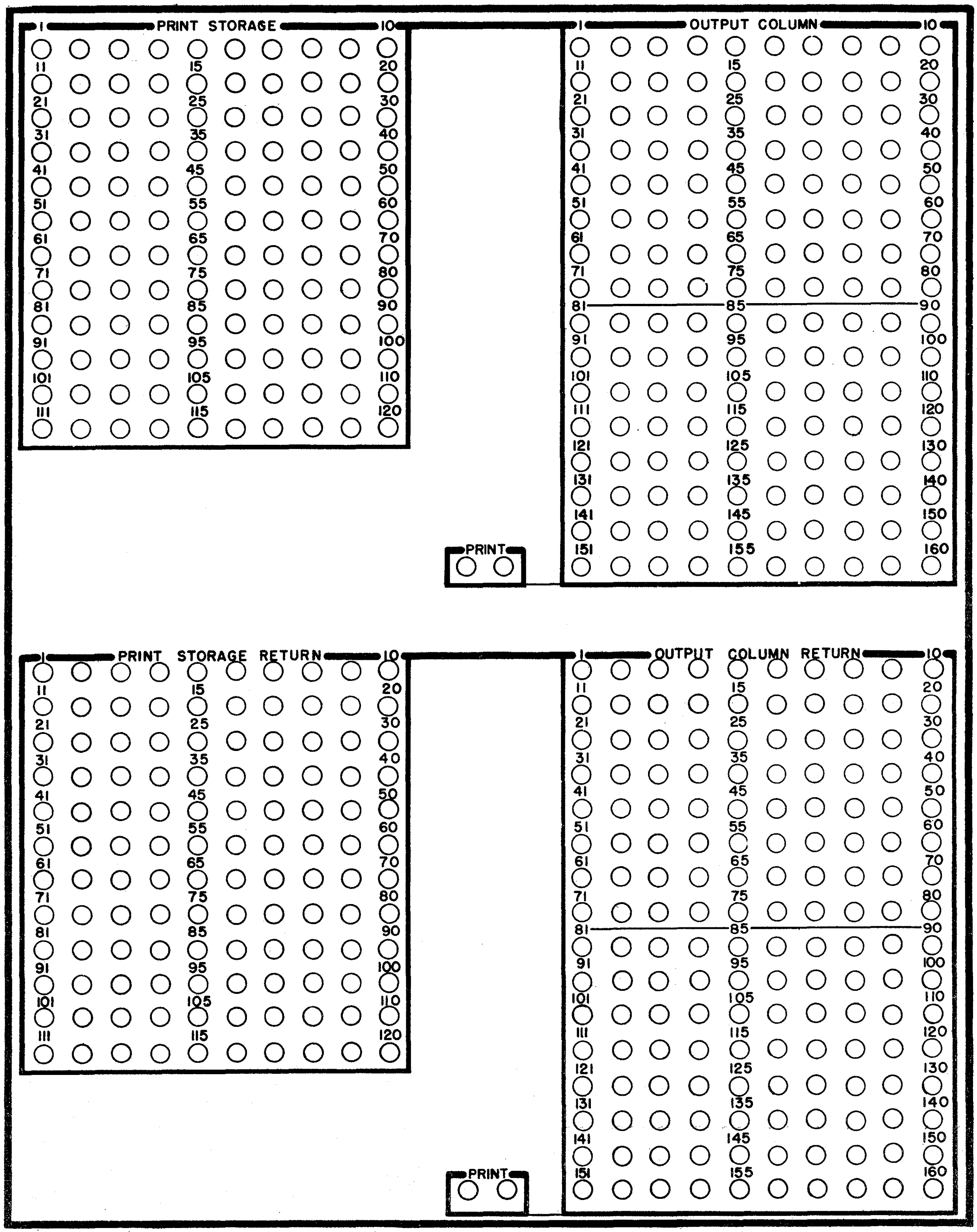


Figure 4-8

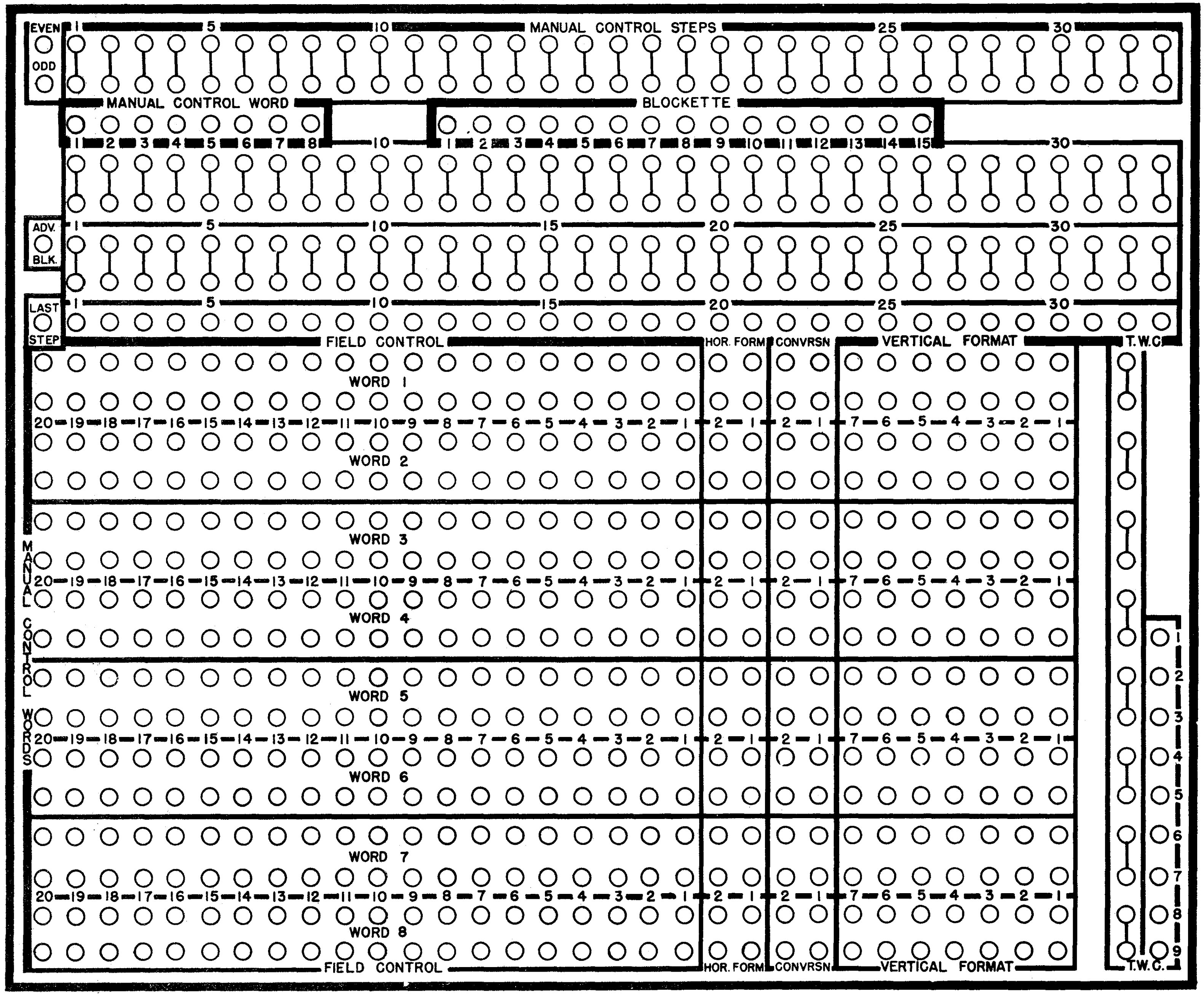

EVEN
ODD
MANUAL CONTROL STEPS
MANUAL CONTROL WORD
BLOCKETTE
ADV.
BLK.
LAST
STEP
FIELD CONTROL
HOR. FORM
CONVRSN
VERTICAL FORMAT
T.W.C.
WORD 1
WORD 2
WORD 3
WORD 4
WORD 5
WORD 6
WORD 7
WORD 8
MANUAL CONTROL WORDS

Figure 4-9

of a run, each hub pair not so connected represents a "zero" bit. To the right of this area are the Transfer Weight Count hubs, a single hub to represent each possible weight count from 1 to 9 and a pair of commoned hubs for each manual control word. For each control word wired, the transfer weight count of bits 22-52 (the bits wired in the Manual Control Word area) is computed manually. The transfer weight of each of these bits is shown in Figure 4-2, page 4-4. The computed weight count is wired to one of the commoned hubs at the right of the control word. The same weight count may be connected to more than one control word by chain wiring. The weight count value of bits 5-21 is zero for all manual control words. (Weight count computation is described in Section I, pages 1-1 and 1-2.)

The upper portion of the panel contains three double rows and one single row of hubs for each of the 33 steps. The first double row of these Manual Control Step hubs specifies the control word to govern the processing of the current blockette, either from tape or from the Manual Control Panel. To use a manual control word in a given step, the desired Manual Control Word hub (from 1 to 8) is wired to that step in the first double row. The same manual control word may be used in several steps by chain wiring. To use a control word from tape, the Odd or Even hub is wired to that step and designates whether the control word is the odd or the even word of channel 8 in the channel selection indicated for that step. When control is turned over to the tape in this fashion, the subsequent path of control depends on the setting of the MANUAL BOARD switch on the console panel. If this switch is set to OFF, the control word from tape will continue to designate subsequent control words in the normal manner until one is encountered which has $\overline{0000}$ in the tape channel selection bits, indicating that control is to be turned back to the next manual control panel step. If the MANUAL BOARD switch is set to ON when a control word from tape is specified, the next blockette will be processed under control of the designated control word, except that the tape channel selection bits will be ignored and control will automatically be returned to the next manual control panel step. Every step used must be wired in the first double row, either to a manual control word or to the Odd or Even hub.

The 15 hubs labelled "Blockette" are used with the second double row of manual control step hubs to specify a tape channel selection from 1 to 15. Each step wired in the row above must be wired in this row from one of the blockette hubs. Chain wiring may be used to repeat a given tape channel selection.

The third double row of hubs in this area is used with the Advance Block hub to specify the point at which it is desired to advance the tape to the next block. The tape will be advanced before the execution of the step to which Advance Block is wired. The tape may be advanced any number of times in a form by chain wiring. Normally, Advance Block is wired to step 1 to advance the tape before starting a sequence of manually controlled steps. However, if printing is begun under control of the manual control panel and Advance Block is wired to step 1, the tape must be advanced one block and reversed two blocks in the manual mode before starting, thus permitting the first tape block to be processed with a manual control word.

The single row *must* be used with the Last Step hub to specify the last step of the control sequence, in order to reset the stepping switch to 1. Last Step must not be wired to step 1 but the control cycle must proceed at least to step 2 before resetting.

Note that the control word wired to a given step governs both the selection and the processing of the blockette for the *same* step. This differs somewhat from the normal situation, in which the control word governs the printing of one line and the selection of information for the next line.

FORMAT CONTROL TAPE

Vertical spacing is controlled by an 8-channel Format Control Tape, located next to the print drum, in conjunction with bits 22-28 of the control word. Two techniques of format control are available, both of which may be used within a form, although only one may be specified in any given control word. The usual technique is to designate a format control channel which will govern paper feed after the line is printed. The other technique specifies the number of lines that the paper shall be advanced after printing the current line. The Format Control Tape is illustrated in Figure 4-10.

A "zero" in control word bit 28 designates the channel control technique, in which the paper is advanced until the next hole is sensed in the controlling channel. The controlling channel is specified by a "one" in one of the bits 22-27, which correspond to format control channels 1-6, respectively. When using this method, only one of these bits may contain a "one". This method is generally used except where printout format necessitates the use of the lines-count technique.

A "one" in bit 28 designates the lines-count technique, in which format control channel 7 controls vertical spacing. In this case, bits 22-27 store a number from 0 to 63 in binary form. The format control tape will be advanced until this number of punched holes are counted in channel 7. Each punch position in a channel corresponds to one line on the printing form; thus if channel 7 were punched in every position, the number

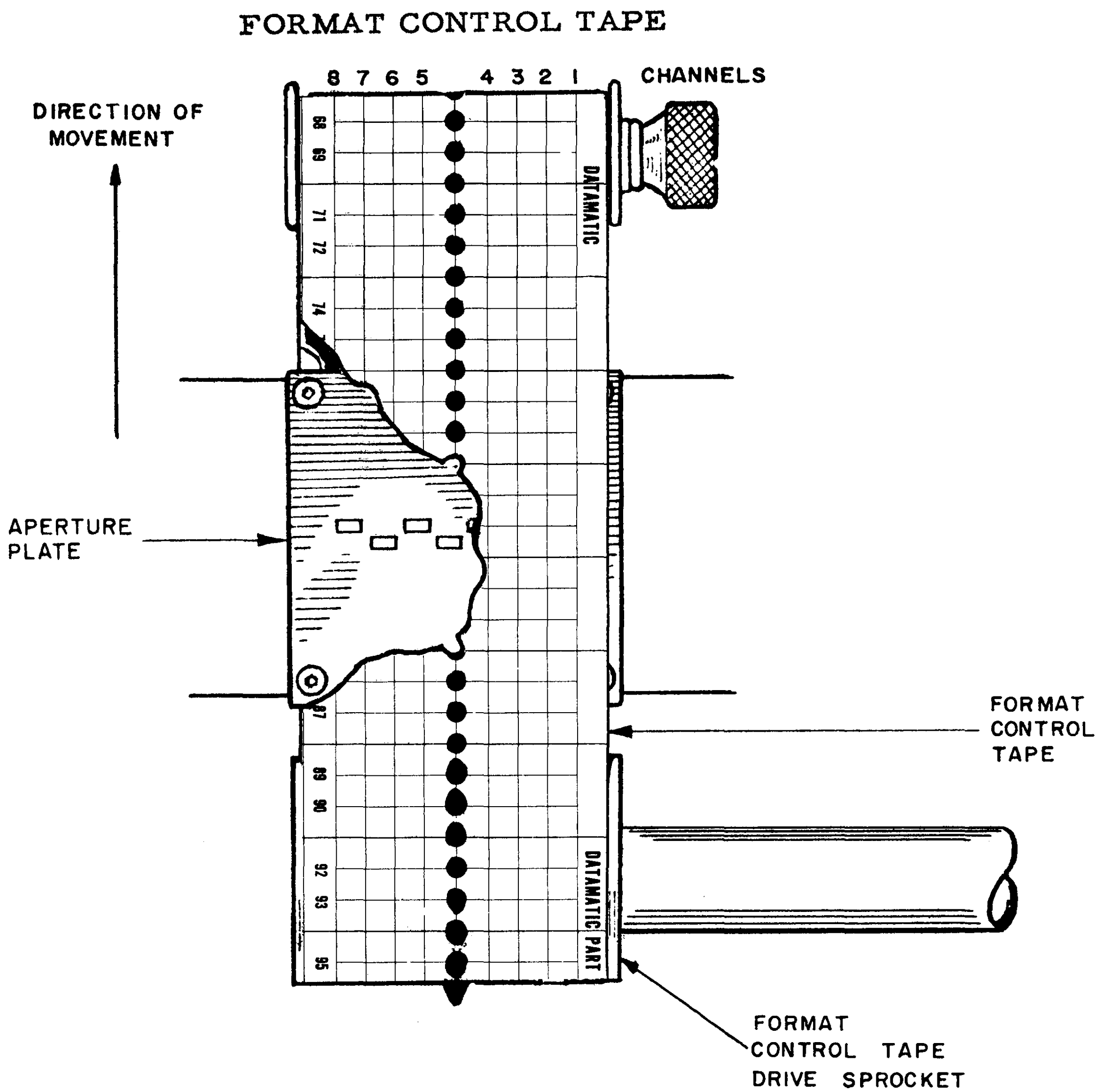


Figure 4-10

stored in bits 22-27 would also represent the number of lines that the paper is to be advanced after the printing of the current line. The lines-count technique may be used, for example, with preprinted forms where the positioning of the output is specified by the preprinted headings but is non-repetitive from form to form.

As previously stated, there are eight channels on the format control tape. Any of the first six channels may be selected by the control word to advance the form to any desired position. In addition, channel 1 is used to designate the head of the form. When the PRINTER-HEAD OF FORM button is pushed, the paper advances until the next hole is sensed in channel 1. The use of channel 7 in the lines-count technique has already been described. Channel 8 is used to define the end of the form when the STOP-END OF FORM switch on the console panel is "on". When this switch is "on", printing will continue until the end-of-form punch is sensed in channel 8. The printer will then stop and position the paper according to the last control word, normally at the head of the next form. During continuous operation, when the STOP-END OF FORM switch is "off", channel 8 has no effect on the operation of the printer.

It is recommended that the use of channels 1-6 be standardized for each installation. No standard uses are suggested here, however, since the requirements will vary from one application to another.

CONVERTER CONSOLE PANEL

The console panel of the Model 1400 Output Printing System (shown in Figure 4-11) is located on the Printer Unit. Many of the console controls have already been described in connection with the related functions. The following is a brief summary of the functions of all of the controls:

(1) POWER-ON. This is the master power switch for the Output Printing System.

(2) PRINTER-HEAD OF FORM. This button advances the Format Control Tape and the paper to the next head-of-form punch in channel 1.

(3) RESET. The CONVERTER SYSTEM RESET button resets all of the converter check circuits to their initial conditions prior to the start of a run. The CONVERTER CHECK RESET button resets the converter check circuits to the normal condition and turns off the associated indicator alarm light(s). The PRINTER RESET button resets the printer after power turn-on or a ribbon alarm. The MFU CHECK RESET button resets the Magnetic File Unit error circuit.

(4) START PRINT. This button initiates a printing run upon the completion of the system reset or resumes printing after a stop in the middle of a run.

(5) STOP. When the STOP-END OF LINE switch is "on", the printer will print one line and then stop. An entire form may be printed in this manner by setting this switch "on" and pressing the START PRINT button once for each line desired. When the STOP-END OF FORM switch is "on", the printer will stop after it senses an end-of-form punch in format control channel 8. This is explained under "Format Control Tape". When the STOP-END OF RUN switch is "on", the printer will stop after printing the line whose control word contains a "one" in bit position 8. To restart after such a stop, move the STOP-END OF RUN switch to OFF and then back to ON; then press START PRINT. One of these three switches should always be "on" while operating, except in the begin-run mode.

(6) MANUAL BOARD. This switch is set "on" when starting a printing run on the Manual Control Panel. If this switch is left "on", each time control is transferred to tape, it will automatically be transferred back to the Manual Control Panel after one print cycle, ignoring the tape channel selection bits. To start a run on the Manual Control Panel and then proceed, transferring control normally between the manual panel and tape, set the MANUAL BOARD switch "on" and press the CONVERTER SYSTEM RESET button. Then set the MANUAL BOARD switch "off" and press the START PRINT button.

(7) MFU DEMAND. This switch must be "on" to make the associated Magnetic File Unit available. In the case of a conflict between the Output System and the Central Processor for the same Magnetic File Unit, the MFU will be controlled by the unit making the prior demand.

(8) BEGIN RUN. As described under "Control Word", the begin-run search is made by selecting the desired position (from 1 to 6) of the RUN SELECTOR switch and pressing the BEGIN RUN SEARCH button. This will position the magnetic tape to read the block in which the "hit" occurred when START PRINT is pressed.

(9) PRIMARY and SECONDARY CATEGORY. As described under "Control Word", the 7-position PRIMARY CATEGORY switch and the six SECONDARY CATEGORY toggle switches are used in conjunction with control word bits 9-14 to allow external selective suppression of printing and paper feeding. These switches are inoperative during the processing of manually controlled print cycles.

(10) MFU MANUAL CONTROLS. When the toggle switch at the left of this group is "off", the seven MFU MANUAL CONTROLS are inoperative. When this toggle switch is "on", the BEGIN RUN SEARCH button is inoperative, control of the Converter Control Unit is released, and the seven MFU MANUAL CONTROLS are active. These buttons enable the operator to move the magnetic tape logically forward or backward, one block or continuously, to rewind the tape, or to switch to the other logical tape half.

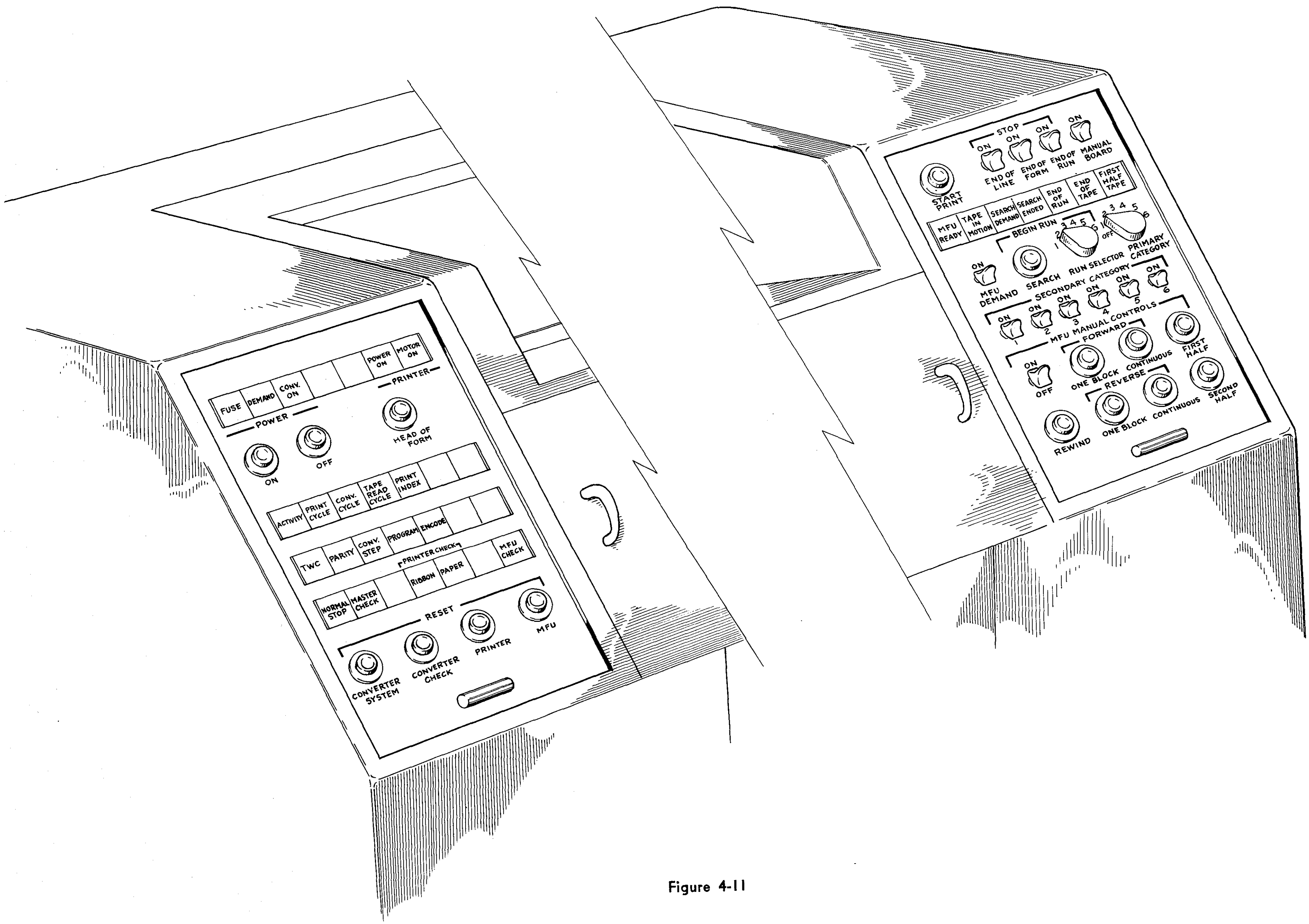

FUSE
DEMAND
CONV. ON
POWER ON
MOTOR ON
PRINTER
POWER
ON
OFF
HEAD OF FORM
ACTIVITY
PRINT CYCLE
CONV. CYCLE
TAPE READ CYCLE
PRINT INDEX
TWC
PARITY
CONV. STEP
PROGRAM
ENCODE
PRINTER CHECK
RIBBON
PAPER
MFU CHECK
NORMAL STOP
MASTER CHECK
RESET
CONVERTER SYSTEM
CONVERTER CHECK
PRINTER
MFU
START PRINT
STOP
ON
END OF LINE
END OF FORM
END OF RUN
MANUAL BOARD
MFU READY
TAPE IN MOTION
SEARCH DEMAND
SEARCH ENDED
END OF RUN
END OF TAPE
FIRST HALF TAPE
BEGIN RUN
MFU DEMAND
SEARCH
RUN SELECTOR
PRIMARY CATEGORY
OFF
SECONDARY CATEGORY
MFU MANUAL CONTROLS
FORWARD
ONE BLOCK
CONTINUOUS
FIRST HALF
REVERSE
SECOND HALF
REWIND

Figure 4-11

900-LINE-PER-MINUTE PRINTER

The general characteristics of the Printer have been described on page 4-1. The main elements of the Printer which are of concern to the programmer are the print drum, the paper-feeding mechanism, and the format controller.

PRINT DRUM: The print drum contains 56 axial rows of alphabetic, numeric, and other symbols. A row consists of 160 identical symbols, spaced ten to the inch, and each row contains a different symbol. The 56 available symbols include the 26 alphabetic characters, the ten numeric digits, and the following special symbols:

' # @ + CR . □ ( ) − = $ * ¢ & / : , % ½

A 17-inch-wide inked ribbon moves continuously over the print-drum housing when the main power is on, automatically reversing itself each time it reaches the end of its travel. This ribbon is capable of approximately 40 to 60 hours of printing.

PAPER FEED: Vertical single spacing is six lines per inch. Paper is advanced by two movable sprocket wheels mounted above the print drum. Each sprocket wheel is capable of independent sidewise motion to accommodate forms from 4 to 22 inches in width. The left-hand sprocket may be set so that print position 1 corresponds to the left edge of the printing form, with the necessary allowance for margin. The right-hand sprocket is then set according to the width of the form. For most economical use of the ribbon, the sprockets may be set to utilize any portion of the ribbon. Both sprockets move along a ruled horizontal bar. Once the proper positions of the sprockets are determined for a given form, they may be noted for future reference in terms of the rulings on the bar.

FORMAT CONTROL TAPE: The use of the format control tape has been described. This special 8-channel paper tape must be planned by the programmer as part of the programming for a printing run and prepared on a special punch, delivered with the system. Note that corresponding punch positions in adjacent channels are not aligned physically, but are offset one position between the odd and the even channels, as shown in Figure 4-10, page 4-21. The two ends of the tape are glued together to form a continuous loop. The length of the tape must be as long as the printing form, or an exact multiple of this length, plus enough overlap to allow for the gluing. At the junction, the start of the tape must lie on top (or outside) of the end of the tape. Since the punches are read photoelectrically, opaque tape must be used.

The format control tape is mounted on two wheels to the left of the paper-feed sprockets. The forward wheel contains the sprockets which drive the tape. It also has axial rulings to be aligned with the cross-tape rulings which appear at every third punch position on the tape. The rear wheel may be moved toward or away from the forward wheel and fixed at the position which just accommodates the length of the tape loop. The maximum length of tape which can be accommodated is 24 inches.

HEAD OF FORM: The format control tape and the printing form must be lined up so that printing will start at the desired position. The format control tape should contain a head-of-form punch in the channel 1 position corresponding to the first line of print. Pressing the PRINTER-HEAD OF FORM button advances the tape to this position. The paper is then brought through manually to the proper alignment with the sprocket wheels, as indicated by the head-of-form marking on the form. This mark is aligned with a ruling on the metal plate above the print drum. If a special form is used which is not an exact multiple of ½ inch in length, it may be necessary to remove the paper from the sprockets and press the HEAD OF FORM button one or more additional times to achieve this alignment. The head-of-form marking must be preprinted on the form and located 6½ inches above the location of the first line of print (i.e., the marking for a given page appears on the preceding page).

TIMING: The print drum rotates continuously at 1200 rpm. Thus one rotation, which is required to print one line, is equal to 50 milliseconds (ms). During this time, the Converter Control Unit reads the next channel selection from tape. Advancing paper one line takes 16 ms, for a total print cycle of 66 ms. The data is processed during the paper-feed time. Multiple-line spacing requires an added 8 ms of paper-feed time for each additional line advanced. Although the print drum continues to rotate during the processing and paper-feed time, it is not necessary to perform a complete rotation before the next line is begun, since the actual printing may start with the print drum in any angular position. In the steady-state condition (when printing on every line of the form), the system produces a continuous output of 900 lines per minute. (This is simply one minute divided by the 66-ms print cycle time.) In general, however, a printing run involves skipping a number of lines on the form for format arrangement and, in some cases, reading some information from tape which is not to be printed. Each of these instances has some effect on printing speed.

The number of forms printed per minute is usually of more direct concern than the number of lines per minute. In general, this number may be calculated in the following manner:

(1) The length of the form in inches, multiplied by 6, gives the maximum number of lines of print per page. Subtracting the actual number of lines to be printed per page leaves the number of lines to be skipped.

(2) The total time for actual printing equals the number of lines to be printed times 66 ms; the total spacing time equals the number of lines to be skipped times 8 ms.

(3) The sum of the actual printing time and the spacing time equals the total time in milliseconds to print one page. Dividing this number *into* 60,000 gives the number of forms printed per minute. This number times the number of lines printed per form equals the number of lines printed per minute for this particular form.

For example, an 11-inch form is to be printed with nine inches of single-spaced copy and 1-inch margins top and bottom. There will be 54 lines printed and 12 lines skipped. Thus, actual printing time is 54 × 66 or 3564 ms, spacing time is 12 × 8 or 96 ms, and the total time per page is 3660 ms. The number of forms per minute is 60,000 divided by 3660, or 16.4. Multiplying this number by 54 (the number of lines printed per page), the printer will run at an average of 885 lines per minute.

A second example is the case of a 4 1/6 inch form containing 15 lines of printing. The maximum number of lines in this example is 25, of which ten are skipped. The actual printing time is 15 × 66, or 990 ms, the spacing time is 10 × 8 or 80 ms, and the total time per page is 1070 ms. The number of forms per minute is 60,000 divided by 1070 or 56.0. The number of lines per minute for this form is 56.0 ×15, or 840.

Neither of these examples considers the case of blockettes read from tape and rejected by the Category switches, which is somewhat more complicated. Since this case may coincide with a case of multi-line spacing, it can be seen that there are a large number of variables which affect timing.

SAMPLE PROBLEM, CONTROL WORDS ON TAPE

To illustrate the use of the Model 1400 Output Printing System in the case where all control words appear on tape, an employee bond register will be printed from an output tape containing payroll records prepared by the DATAmatic 1000 Central Processor. A typical payroll record, as shown in Figure 4-12, consists of 28 information words covering one employee, a control word to govern processing by the Converter Control Unit, and a second control word which is not used in this problem. The tape layout consists of two such records per block.

The bond register consists of one line of print per employee, containing the employee's number, name, bond deduction, bond balance, bond purchase price, and number of bonds due. The preprinted form used to print the register is illustrated in Figure 4-13.

Tape Channel Selection Control Panel: To print the bond register, the following information is needed from the tape:

(1) Employee Number—least significant five digits of word 1 (or 31),

(2) Employee Name—words 2, 3, and 4 (or 32-34),

(3) Bond Deduction—least significant six digits of word 22 (or 52),

(4) Bond Balance—most significant five digits (excluding sign) of word 26 (or 56),

(5) Bond Purchase Price—most significant five digits (excluding sign) of word 28 (or 58),

(6) Number of Bonds Due—sixth least significant digit of word 28 (or 58).

The control word for this operation is word 16 (or 46). All of the information needed to process the first item resides in channels 0, 1, 10, 12, 13, and 7. Since it is necessary to wire a full eight channels in each channel selection, channels 2 and 3 are added to complete channel selection 1. These four words will be discarded on the Conversion Control Panel. Channel 7 (containing the control word) must be the eighth channel of the selection. Similarly, channel selection 2 consists of channels 15, 16, 25, 27, and 28, containing the information needed for the second item; channels 17 and 18 to be discarded later; and channel 22, containing the control word. These two channel selections, shown in Figure 4-14, complete the wiring of this control panel. Figure 4-14 takes the form of a worksheet which is recommended for planning the wiring of the tape channel selection panel. Such worksheets will be shown in the sample problems to familiarize the reader with their use.

Control Words: The control words required to process the bond register are illustrated in Figure 4-15. All control words are of the form shown as "steady state

SAMPLE PROBLEM, CONTROL WORDS ON TAPE

Word	12	11 10 9 8 7 6 5 4 3 2 1	Sector	12	11 10 9 8 7 6 5 4 3 2 1	Word
1	+	Dept. No. / Employee No.	0		Employee Name	2
3		Employee Name	1		Employee Name	4
5		Street and Number	2		Street and Number	6
7		City and State	3		City and State	8
9		Net Pay	4	+	Total Deductions / Withholding Tax	10
11	+	FICA Wages to Date / FICA Deduction	5	+	FICA Tax to date / Total Federal Tax	12
13	+	Overtime Pay / Regular Pay	6	+	Night Bonus Pay / Gross Pay	14
15	+	Overtime Hours / Night Bonus Hours	7		Control Word	16
17	+	Total Hours	8	+	Check Number / Pay Period	18
19	+	Union Dues	9	+	Group Insurance	20
21	+	Hospitalization	10	+	Bond Deduction	22
23	+	Community Fund	11	+	Total Deductions	24
25	+	Marr Sing / Gross Pay to Date	12	+	Bond Balance / With. Tax to Date	26
27	+	Rate / Depend. / Union Loc.	13	+	Bond Price / No Bnds / Blue Cross	28
29	+	Social Security Number	14		Control Word	30
31			15			32
33			16			34
35			17			36
37			18			38
39		Same As	19		Above	40
41			20			42
43			21			44
45			22		Control Word	46
47			23			48
49			24			50
51			25			52
53			26			54
55			27			56
57			28			58
59			29		Control Word	60
61			30			62

Figure 4-12

BOND REGISTER

Employee Number (Field 2)	Employee Name (Field 3)	Bond Purchase Price (Field 6)	Bond Deduction (Field 4)	Number of Bonds Due (Field 7)	Bond Balance (Field 5)
XXXXX	XXXXXXXXXXXXXXXXXXXXXXXX	$XXX.XX	$XXXX.XX	X	$XXX.XX
30 34	38 61	66 72	75 82	86	90 96

Print Positions

Figure 4-13

TAPE CHANNEL SELECTION PANEL WORKSHEET

Positions \ Blockette	1	2	3	4	5	6	7	8
1	0	15						
2	1	16						
3	10	25						
4	12	27						
5	13	28						
6	2	17						
7	3	18						
8	7	22						

Positions \ Blockette	9	10	11	12	13	14	15
1							
2							
3							
4							
5							
6							
7							
8							

Figure 4-14

CONTROL WORD LAYOUT

MCB PROG. STEPS | ←— MEANINGLESS ON MCB —→

←— TRANSFER WEIGHT ZERO ON MCB —→

Bit Position	52 51 50 49	48 47 46 45	44 43 42 41	40 39 38 37	36 35 34 33	32 31 30 29	28 27 26 25	24 23 22 21	20 19 18 17	16 15 14 13	12 11 10 9	8 7 6 5	4 3 2 1	
Transfer Weight	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	
Control Word	Field Selection 20 19 18 17	Field Selection 16 15 14 13	Field Selection 12 11 10 9	Field Selection 8 7 6 5	Field Selection 4 3 2 1	Horizontal Format Bd; Converter Board	Vertical Format Channels 7 6 5 4	Vertical Format Channels 3 2 1; Odd-Even	Tape Channel Selection	Adv. Block; N-P Skip; Category 6 5	Category 4 3 2 1	End Run; Begin Run	Weight Count	
1. Steady State (1st Item)	0000	0000	0000	0000	0000	0101	0000	0100	0010	0000	0001	0000	0011	B
	0	0	0	0	0	5	0	4	2	0	1	0	3	H
2. Steady State (2nd Item)	0000	0000	0000	0000	0000	0101	0000	0100	0001	1000	0001	0000	0001	B
	0	0	0	0	0	5	0	4	1	8	1	0	1	H
3. No Printout (1st Item)	0000	0000	0000	0000	0000	0101	0000	0100	0010	0000	0000	0000	0010	B
	0	0	0	0	0	5	0	4	2	0	0	0	2	H
4. No Printout (2nd Item)	0000	0000	0000	0000	0000	0101	0000	0100	0001	1000	0000	0000	1001	B
	0	0	0	0	0	5	0	4	1	8	0	0	9	H
5. Head of Form (1st Item)	0000	0000	0000	0000	0001	0101	0000	0100	0010	0000	0001	0001	0101	B
	0	0	0	0	1	5	0	4	2	0	1	1	5	H
6. Head of Form (2nd Item)	0000	0000	0000	0000	0001	0101	0000	0100	0001	1000	0001	0001	0011	B
	0	0	0	0	1	5	0	4	1	8	1	1	3	H
7. End Run	0000	0000	0000	0000	0000	0101	0000	0100	0001	1000	0001	1000	1001	B
	0	0	0	0	0	5	0	4	1	8	1	8	9	H
														B
														H

Remarks:

BIT USE, HFB # 01

Field selection bits 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 — 1: Special Field Use ($)

Vertical format channels 7 6 5 4 3 2 1 — 1: Control Channel

Category 6 5 4 3 2 1 — 1: Print or Don't Print

Figure 4-15

(1st item)" or "steady state (2nd item)", depending upon the position of the item in the block, except for the first item to be printed on each form, the last item of the tape record, and those items which are not to be processed. The two "steady state" control words differ only with respect to the tape channel selection and the advance block bits. Those residing in the first item of a block are designed to read tape channel selection 2 of the same tape block, while those residing in the second item advance the tape and read tape channel selection 1 of the following block.

The "no printout (1st item)" or "no printout (2nd item)" control word is used in the payroll record of each employee who does not buy bonds, depending upon the position of such a record in the tape block. The tape channel selection and advance block bits of these two control words are the same as those of the corresponding "steady state" control words. Category bit 1 is a "zero" in these control words and a "one" in the "steady state" control words. The PRIMARY CATEGORY switch is set to 1 so that all items having "steady state" control words will be processed and items having "no printout" control words will not. The SECONDARY CATEGORY switches are all "off". The non-print skip bit is a "zero" in all of these control words because it is desired to read in each item in succession, whether it is to be processed or not.

The "head of form" control words differ from the "steady state" control words in the following two respects:

(1) The begin run bits contain the configuration $\overline{0}\overline{0}\overline{1}$, which is used during the begin run search to identify the start of the payroll record on tape. Since the first item in the record is also the first head-of-form item, these control words are used to perform a dual function.

(2) Field selection bit 1 is a "one". This field is used to print dollar signs in the first printed line of each page, as described under "Horizontal Format Control Panel".

The programmer will place a "head of form (1st item)" control word in the first item of the record. Thereafter, he will place one of the "head of form" control words in every twentieth item, depending upon the position of such an item in the tape block.

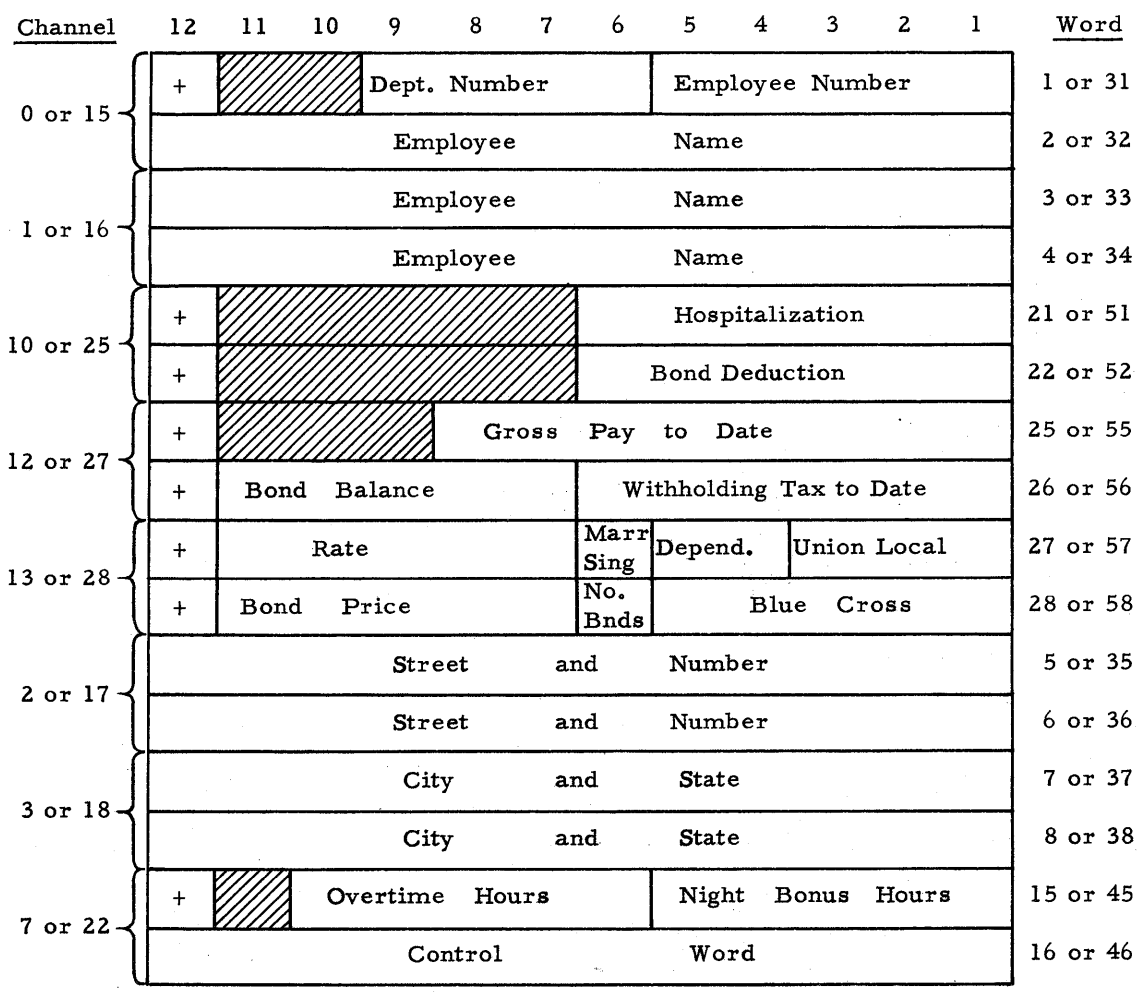


Figure 4-16

CONVERSION CONTROL PANEL WORKSHEET

Panel $\bar{0}\bar{1}$

Step	Order	W	Repetitions	W	Bits	Blockette Word(s)	Digits
1	BD	-	28	-	28	1	7
2	4H	-	5	-	20	1	5
3	6A	-	24	-	144	2-4	36
4	WD	-	1	-	48	5	12
5	BD	1	24	3	24	6	6
6	4H	2	6	-	24	6	6
7	WD	4	1	4	48	7	12
8	BD	5	4	-	4	8	1
9	4H	6	5	2	20	8	5
10	BD	8	24	5	24	8	6
11	WD	7	1	7	48	9	12
12	BD	10	4	8	4	10	1
13	4H	9	6	6	24	10	6
14	BD	12	20	-	20	10	5
15	EC	-	-	-	288	11-16	72
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							

6A - 6-bit Alphanumeric
4H - 4-bit Hexadecimal
1N - 1-bit Numeric
1S - 1-bit Sign
4S - 4-bit Sign
BD - 1-bit Discard
WD - Word Discard
EC - End Conversion

Figure 4-17

The "end run" control word differs from a "steady state" control word only in that the end run bit is a "one". The contents of the tape channel selection and advance block bits are irrelevant. If this is a "no printout" item, the first category bit of its control word is a "zero".

CONVERSION CONTROL PANEL: The layout of the information in Tape Data Storage, shown in Figure 4-16, is the same regardless of which channel selection was read in. Since the format of all lines of printout is the same, only one conversion panel is required in this example. This panel, which is identified in control word bits 29-30 as panel $\bar{0}\bar{1}$, is wired according to the worksheet shown in Figure 4-17. Channel selection 1, for example, fills Tape Data Storage with words 1, 2, 3, 4, 21, 22, 25, 26, 27, 28, 5, 6, 7, 8, 15, and 16. Step 1 on the conversion panel discards the high-order 28 bits of word 1. Step 2 converts the remaining 20 bits to the five hexadecimal digits of employee number. The remaining steps apply the conversion and discard rules to extract the six items of information which are required

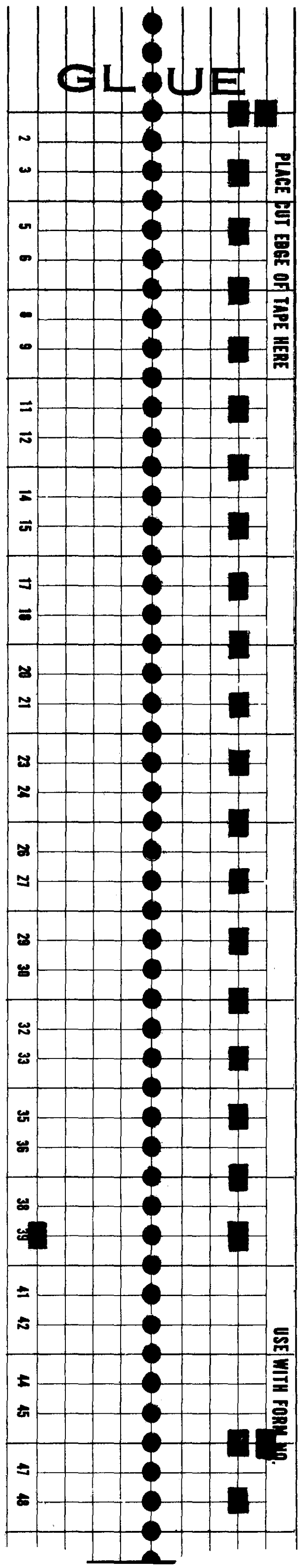


Figure 4 3

to print the bond register. In Figure 4-17, which shows the recommended format for conversion panel worksheets, the columns labelled "W" are used when chain wiring to specify the last previous step at which the same orders hub or repetitions hub was wired. For example, opposite step 5, which executes a 1-bit discard, the "W" column contains a "1" which means that this hub is wired from conversion step 1, the last previous step to execute a 1-bit discard. Both the wiring and the checking of control panels will be expedited by the use of such worksheets.

FORMAT CONTROL TAPE: The bond register is to be printed upon a 7½-inch-long, preprinted form with three lines to the inch and 20 lines per form. This means that the paper is to be advanced two lines after each line is printed. All of the control words to be used contain the same configuration of vertical format bits, namely a "zero" in bit 28 to designate the normal or channel control technique, a "one" in bit 23 which specifies channel 2 as the controlling channel, and "zeros" in all of the other vertical format bits. The format control tape, shown in Figure 4-18, is the same length as the printing form and contains 20 punches in the alternate spaces of channel 2. The punch in channel 1 is the head-of-form punch which enables the operator to line up the form at the first line of print by means of the PRINTER-HEAD OF FORM switch. On the actual tape, this punch will be offset one space below the position indicated in the layout. The punch in channel 8 is an end-of-form punch which may be used with the STOP-END OF FORM switch to check the vertical spacing prior to the start of the run.

HORIZONTAL FORMAT CONTROL PANEL: The Horizontal Format Control Panel governs the layout of each line on the printed form. The Tape Channel Selection Control Panel arranges the sequence of channels (2-word groups) before they enter Tape Data Storage; the Conversion Control Panel selectively converts and discards information from Tape Data Storage but does not change its sequence; the Output Format Control Panel inserts spaces to position the line of print over the 160 available print positions. However, only the Horizontal Format Control Panel can arrange one line differently from the others on the same form and insert special symbols not recorded on the tape.

The bond register is a very simple example which utilizes very few of the capabilities of the horizontal format panel. Converted data is rearranged in a different sequence and divided into six fields (see Figure 4-13). The use of control word bit 33 for special field control purposes has been introduced as a simple example of the flexibility of this panel. Only one horizontal format panel, specified by the configuration $\overline{0}\overline{1}$ in bits 31-32, is required to print the bond register. Figure 4-19 shows the worksheet for wiring this panel, while the wiring diagram is illustrated in Figure 4-20. This wiring is developed as follows:

HORIZONTAL FORMAT PANEL WORKSHEET

Field	Hub Type	Function
1	1	Set Flip-flop A1 to 1
	U	Special End
	0	Set Flip-flop A1 to 0
2	U	Begin field in PCA 30
	PCB 34	End field unconditionally
3	U	Begin field in PCA 38
	PCB 61	End field unconditionally
4	U	Begin field in PCA 75
	PCB 75 + (A1=1)	Emit $ unconditionally
	PCB 75 + (A1=0)	Emit Space unconditionally
	PCB 80	Emit Period unconditionally
	PCB 82	End field unconditionally
5	U	Begin field in PCA 90
	PCB 90 + (A1=1)	Emit $ unconditionally
	PCB 90 + (A1=0)	Emit Space unconditionally
	PCB 94	Emit Period unconditionally
	PCB 96	End field unconditionally

Field	Hub Type	Function
6	U	Begin field in PCA 66
	PCB 66 + (A1=1)	Emit $ unconditionally
	PCB 66 + (A1=0)	Emit Space unconditionally
	PCB 70	Emit Period unconditionally
	PCB 72	End field unconditionally
7	U	Begin field in PCA 86
	PCB 86	End horizontal format unconditionally

Notes: PCA(B) indicates Print Control A (B).

(Ax=y) indicates setting of specified flip-flop.

Each + sign indicates use of an AND gate.

Each "function" wired more than once requires use of an OR gate.

Each "hub type" wired more than once requires use of a commoning set.

Figure 4-19

(1) As mentioned, field 1 is used for special control purposes; fields 2 through 7 contain printed information. The use of field 1 is explained in paragraph (4). The unconditional hubs of fields 2-7 are wired to print control A hubs 30, 38, 75, 90, 66, and 86, respectively. This places the beginnings of the fields in these positions in Print Storage.

(2) Print control B hubs 34, 61, 82, 96, and 72 are wired through an OR gate to an unconditional end-of-field hub, defining the print storage positions where fields 2 through 6 end. Note that field 2, for example, is wired to occupy print storage positions 30 through 34. However, due to the automatic suppression of nonsignificant zeros, a given field may not occupy all of the positions defined by the print control A and B hubs.

(3) Print control B hub 86 is wired to the end-horizontal-format hub to reset the two stepping switches after field 7 is positioned. Note that each field to be printed must have its beginning and ending defined on the horizontal format panel, even though it contains but one character.

(4) The unconditional hub of field 1 is wired to the special end hub in order that bit 33 of the control word may govern the selective printing of dollar signs to the left of dollar amounts in the bond register. In this case, it is desired to print dollar signs only in the first printed line of each form with corresponding spaces in all other lines. Thus only the "head-of-form" control words have "ones" in bit position 33 and all other control words have "zeros" in this bit position. The selective printing of dollar signs is accomplished using the first auxiliary flip-flop. One side of the flip-flop is wired from the "one" hub of field 1 and, via an AND gate, to the unconditional dollar sign emitter. The other side is wired from the "zero" hub of field 1, and, via another AND gate, to the unconditional space emitter. The emitted dollar signs (or spaces) are sent to print storage positions 66, 75, and 90. These hubs under print control B are wired through commoning hubs and OR gates to the AND gates previously mentioned. (The reader may find it helpful at this point to review the functions of OR gates, AND gates, and commoning hubs.)

(5) Since the dollar amounts do not contain decimal points on the tape, these points must be emitted and sent to print storage positions 70, 80, and 94. This is accomplished by wiring from print control B hubs 70, 80, and 94 to the unconditional decimal point emitter, via an OR gate.

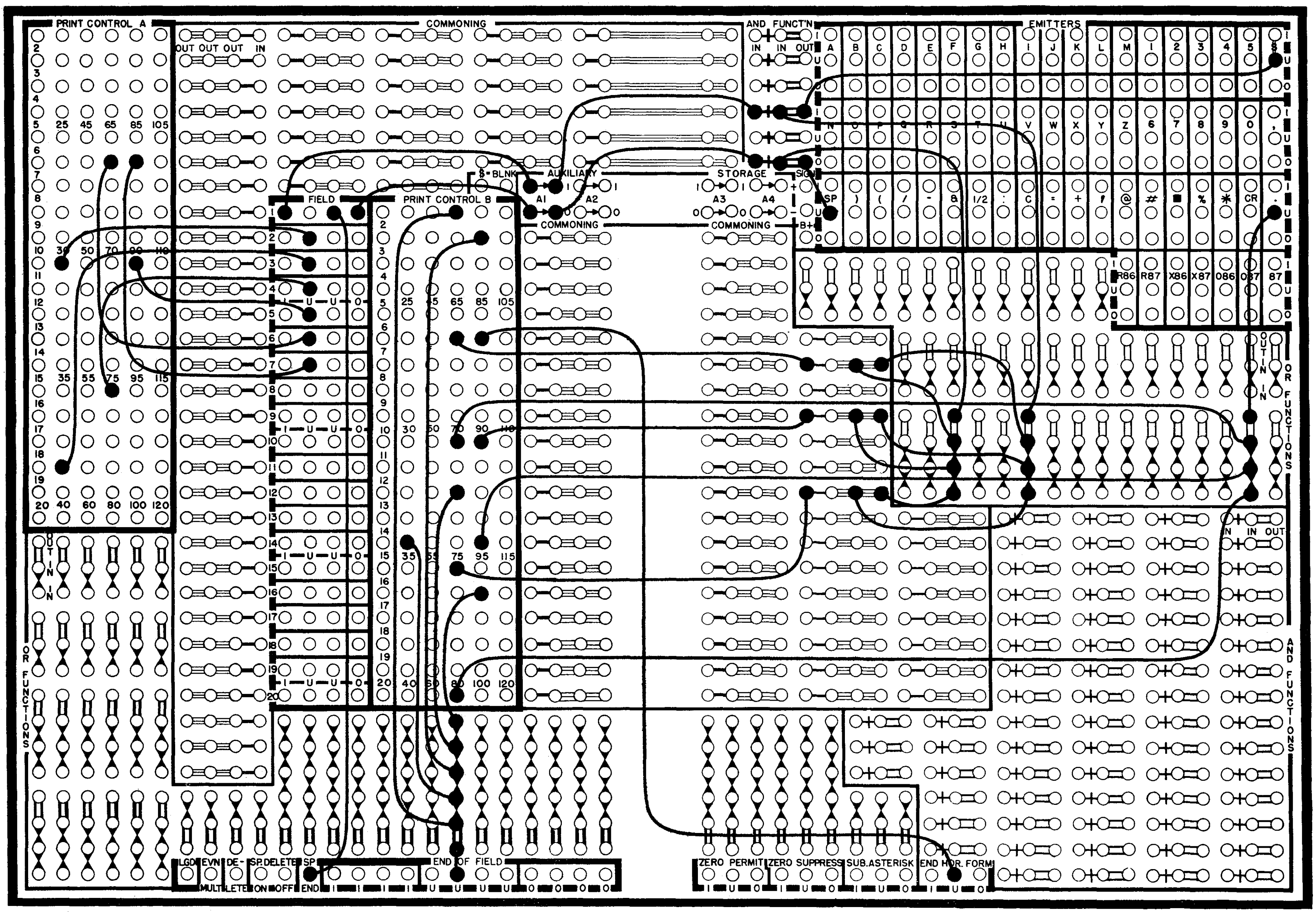

PRINT CONTROL A
COMMONING
AND FUNCT'N
EMITTERS
FIELD
PRINT CONTROL B
$-BLNK
AUXILIARY
STORAGE
SIGN
COMMONING
OR FUNCTIONS
AND FUNCTIONS
LGD
EVN
DE-
SP.DELETE
SP.
MULTI-
LETE
ON
OFF
END
END OF FIELD
ZERO PERMIT
ZERO SUPPRESS
SUB.ASTERISK
END HOR. FORM

Figure 4-20

Output Format Control Panel: In this example, all information to be printed is positioned by the horizontal format panel in the first 120 print storage positions. Since no further spacing is required, the wiring of the Output Format Control Panel is extremely simple. It is only necessary to wire print storage hubs 30 through 96 to the corresponding output column hubs in both halves of the panel. Note that if the programmer prepares one output format panel with all 120 print storage hubs wired to their corresponding output columns hubs, such a panel can be used for all printing runs such as the present example, in which all spacing and output positioning is accomplished on the horizontal format panel.

Operation: This completes the coding of the control words, wiring of the control panels, and preparation of the output format tape to print the bond register. Assuming the necessary Magnetic File Unit connections, the operator may now perform the begin run search. The run selector switch is set on 1 and begin run search is pressed. The system searches the tape, reading the even word of the eighth channel in channel selection 1 until it encounters the configuration $\bar{0}0\bar{1}$ in the begin run bits of such a word. Before starting the printing run, the operator must check the following switch settings:

(1) One of the three stop switches must be set to on to indicate at what point the operator wishes to stop.

(2) The manual board switch is set to off.

(3) The primary category switch is set to 1 and all of the secondary category switches are set to off, as stated under "Control Words".

(4) The mfu manual controls switch must be set to off to activate the start print button.

SAMPLE PROBLEM, MANUAL AND TAPE CONTROL WORDS

To demonstrate the use of the Model 1400 Output Printing System when only a portion of the required control words are recorded on tape, a set of W-2 Federal Withholding Tax Statements will be printed from the same payroll tape used in the bond register problem. The reader will recall that the format of the payroll tape (see Figure 4-12, page 4-26) contains two employees' records per tape block. Figure 4-21 is a W-2 Withholding Tax Statement, showing the format of the five lines of print required to complete the form.

To print the statement, the following information is required from the tape:

(1) Total FICA Wages—most significant six digits (excluding sign) of word 11 (or 41),
(2) Total FICA Tax—most significant four digits (excluding sign) of word 12 (or 42),
(3) Total Wages—least significant seven digits of word 25 (or 55),
(4) Total Withholding Tax—least significant six digits of word 26 (or 56),

WITHHOLDING TAX STATEMENT **1957** **Copy D - For Employer**
Federal Taxes Withheld From Wages

SOCIAL SECURITY INFORMATION		INCOME TAX INFORMATION		S-SINGLE M-MARRIED
$ 0000.00 (14 18 20)	$ 00.00 (27 29 31)	$ 00000.00 (38 43 45)	$ 0000.00 (51 55 57)	X (61)
Total F.I.C.A. Wages* Paid in 1957.	F.I.C.A. employee tax withheld, if any.	Total Wages* Paid in 1957	Federal Income Tax withheld, if any.	

Type or print EMPLOYER'S identification number, name, and address below.

Minneapolis Honeywell Regulator Company
Datamatic Division
151 Needham Street, Newton Highlands 61, Mass.
Reporting Unit 21-7 ***41-0415010***

Type or print EMPLOYEE'S social security account no., name, and address below.

000-00-0000
AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
13 16 19 23 28 36

EMPLOYER: This copy is provided for your convenience in keeping your withholding records.

APP. I.R.S. 12-11-56

FORM W-2 - U. S. Treasury Department, Internal Revenue Service *Before Payroll deductions

Figure 4-21

(5) Marital Status—one alphanumeric character to right of sign digit of word 25 (or 55),

(6) Social Security Number—least significant nine digits of word 29 (or 59),

(7) Employee Name—words 2, 3, and 4 (or 32-34),

(8) Street Address—words 5 and 6 (or 35 and 36),

(9) City and State—words 7 and 8 (or 37 and 38).

Owing to the differences in format among the various lines of printout, four basic types of control words are required to print the W-2. (Lines 4 and 5 can be printed by the same manual control word as their formats are identical.) The printing of W-2 forms is performed only once each year and consequently, it is impractical to insert four different control words in each payroll item. The programmed solution utilizes one control word from tape (word 30 or 60) to print the employee's name and prints the balance of the form under control of the Manual Control Panel.

Tape Channel Selection Control Panel: A total of four tape channel selections are wired on this control panel, two of which extract the pertinent information from the first item of the tape block and two which read the corresponding channels from the second item. Channel selection 1, which reads the information to print lines 1, 2, and 4, comprises channels 5 and 12 (to print line 1); channel 2 (to print line 4); channels 4, 6, 7, and 8 as fillers to complete the selection; and channel 14 (to print line 2). Channel 14 is the eighth channel of the array because it contains the control word which is used during the begin run search to identify the payroll record. Channel selection 2 consists of channels 0 and 1 (to print line 3); channel 3 (to print line 5); channels 4, 6, 7, and 8 as fillers; and channel 14 (to control the printing of line 3). Again channel 14 must be in the eighth channel position since one of its words is used as a control word. Channel 3 must occupy the same position in the array as did channel 2 in selection 1 in order to process lines 4 and 5 with the same Conversion Control Panel. Channel selections 3 and 4 correspond to selections 1 and 2, respectively, except that they read information from the second item in the tape block. The wiring of the Tape Channel Selection Control Panel is shown in the worksheet of Figure 4-22.

Conversion Control Panels: This problem uses all four conversion panels. Panel $\overline{01}$ processes the information in channel selection 1 (or 3), extracting the pertinent information from the first and second channels selected to print line 1 and discarding the remainder of the array. Panel $\overline{10}$ processes the information from the same channel selections but discards the first 7 channels and converts part of the eighth channel to print line 2. Panel $\overline{11}$ processes the information from selection 2 (or 4), printing the employee's name from the pertinent portions of the first two channels and discarding the remainder of the array. Panel $\overline{00}$ operates on all four channel selections, converting the third channel in each case and discarding the balance of the array. The third channel of selection 1 (or 3) yields line 4, the street and number, while the corresponding channel of selection 2 (or 4) yields line 5, the city and state. The worksheets used in wiring the conversion panels are shown in Figure 4-23.

TAPE CHANNEL SELECTION PANEL WORKSHEET

Blockette * / Positions	1	2	3	4	5	6	7	8
1	5	0	20	15				
2	12	1	27	16				
3	2	3	17	18				
4	4	4	19	19				
5	6	6	21	21				
6	7	7	22	22				
7	8	8	23	23				
8	14	14	29	29				

* Blockettes 9-15 not used.

Figure 4-22

CONVERSION CONTROL PANEL WORKSHEET

Panel $\bar{0}\bar{1}$

Step	Order	W	Repetitions	W	Bits	Blockette Word(s)	Digits
1	BD	-	4	-	4	1	1
2	4H	-	6	-	24	1	6
3	BD	1	24	-	24	1, 2	6
4	4H	2	4	1	16	2	4
5	BD	3	32	-	32	2, 3	8
6	6A	-	1	-	6	3	1 1/2
7	BD	5	10	-	10	3	2 1/2
8	4H	4	7	-	28	3	7
9	BD	7	24	3	24	4	6
10	4H	8	6	2	24	4	6
11	EC	-	-	-	576	5-16	144

Panel $\bar{1}0$

1	WD	-	14	-	672	1-14	168
2	BD	-	12	-	12	15	3
3	4H	-	9	-	36	15	9
4	EC	-	-	-	48	16	12

Panel $\bar{1}\bar{1}$

1	WD	-	1	-	48	1	12
2	6A	-	24	-	144	2-4	36
3	EC	-	-	-	576	5-16	144

Panel $\bar{0}\bar{0}$

1	WD	-	4	-	192	1-4	48
2	6A	-	16	-	96	5, 6	24
3	EC	-	-	-	480	7-16	120

6A - 6-bit Alphanumeric
4H - 4-bit Hexadecimal
1N - 1-bit Numeric
1S - 1-bit Sign
4S - 4-bit Sign
BD - 1-bit Discard
WD - Word Discard
EC - End Conversion

Figure 4-23

CONTROL WORDS: It was previously noted that this problem requires four basic types of control words and that three of these are wired on the Manual Control Panel, the other being recorded on the payroll tape. The only other control word used is the end-run variation of the tape control word, which the programmer places in the last item of the payroll record. The bit structure of these five control words is shown in Figure 4-24, the control word worksheet. Several points may be noted from this figure:

(1) No use is made of the Category feature, so that the PRIMARY CATEGORY switch should be set to OFF in this example,

CONTROL WORD LAYOUT

MCB PROG. STEPS (bits 21–16) | ← MEANINGLESS ON MCB → (bits 15–5) | TRANSFER WEIGHT ZERO ON MCB (bits 20–5)

Bit Position	52 51 50 49	48 47 46 45	44 43 42 41	40 39 38 37	36 35 34 33	32 31 30 29	28 27 26 25	24 23 22 21	20 19 18 17	16 15 14 13	12 11 10 9	8 7 6 5	4 3 2 1	
Transfer Weight	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	
Control Word	Field Selection 20 19 18 17	Field Selection 16 15 14 13	Field Selection 12 11 10 9	Field Selection 8 7 6 5	Field Selection 4 3 2 1	Horizontal Format Bd, Converter Board	Vertical Format Channels 7 6 5 4	Vertical Format Channels 3 2 1, Odd-Even	Tape Channel Selection	Adv. Block, N-P Skip, Category 6 5	Category 4 3 2 1	End Run, Begin Run	Weight Count	
Tax Information (line 1)	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1	1 0 0 0	0 1 0 1	0 0 0 0	0 1 0					0 1 0 0	B
1. Manual Control Word No. 1	0	0	0	G	8	5	0	4					4	H
Social Security Number (line 2)	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	0 1 1 0	0 0 0 0	0 1 0					0 0 0 1	B
2. Manual Control Word No. 2	0	0	0	0	1	6	0	4					1	H
Name - Steady State (line 3)	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	0 1 1 1	0 0 0 0	0 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	0 1 0 1	B
3. Tape Word 30 (or 60)	0	0	0	0	2	7	0	4	0	0	0	2	5	H
Street & No., City & State (lines 4 5)	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 0	0 1 0 0	0 0 0 0	0 1 0					0 1 1 0	B
4. Manual Control Word No. 3	0	0	0	0	4	4	0	4					6	H
Name - End Run (line 3)	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	0 1 1 1	0 0 0 0	0 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0	0 0 1 0	B
5. Tape Word 30 (or 60)	0	0	0	0	2	7	0	4	0	0	0	8	2	H
														B
														H
														B
														H
														B
														H

Remarks:

BIT USE, HFB # 01

Field Selection 8	Start field 8, line 1
Field Selection 7	Start field 7, line 1
Field Selection 6	Start field 6, line 1
Field Selection 5	Start field 5, line 1
Field Selection 4	Start field 4, line 1
Field Selection 3	Identify lines 4 and 5
Field Selection 2	Identify line 3
Field Selection 1	Identify line 2
Vertical Format Channel 2	Control channel

Figure 4-24

(2) The channel control mode of vertical format control is used exclusively, with channel 2 as the controlling channel,

(3) All five control words designate horizontal format panel 01 for further processing.

Manual Control Word 1 governs the printing of the first line of information, derived from channel selection 1 (or 3). Conversion panel 01 is designated. The five fields of information in this line are controlled on the horizontal format panel by the "ones" in field selection bits 4-8. (Fields 1-3 are used for special control functions in this example.)

Manual Control Word 2 governs the printing of line 2, the Social Security Number. Conversion panel 10 is designated. Field selection bit 1 is a "one" to specify the horizontal format and emission parameters of this line. These parameters are discussed under "Horizontal Format Control Panel".

Tape Word 30 (or 60)—steady state: This word is used to print line 3, the employee name. The tape channel selection bits of this word (0000) return control to the Manual Control Panel after this line is printed. Conversion panel 11 is designated. Field selection bit 2 is a "one" to specify the horizontal format parameters of this line. The begin run bits contain the configuration 010, used to identify the items used in the withholding tax run.

Manual Control Word 3 governs the printing of employee's address, lines 4 and 5. Conversion panel 00 is designated and converts the information to print both of these lines. Field selection bit 3 is a "one" to specify the horizontal format parameters of this line.

Tape Word 30 (or 60)—end run: This word is identical to the steady state form except that the end run bit is a "one". The programmer places this word in the last item to be processed to indicate the end of the printing run.

FORMAT CONTROL TAPE: The channel control technique of vertical format control is used to print the W-2 forms, with channel 2 as the controlling channel. The tape is shown in Figure 4-25, together with a portion of a W-2 form to show the correspondence of punches with the printout lines. The tape is 22 punch positions (about 3⅝ inches) long. Channel 2 is punched in positions 5, 15, 16, 17, and 18. Channel 1 is punched in position 1 to designate the head of the form and channel 8 in the 18th position to designate the end of the form.

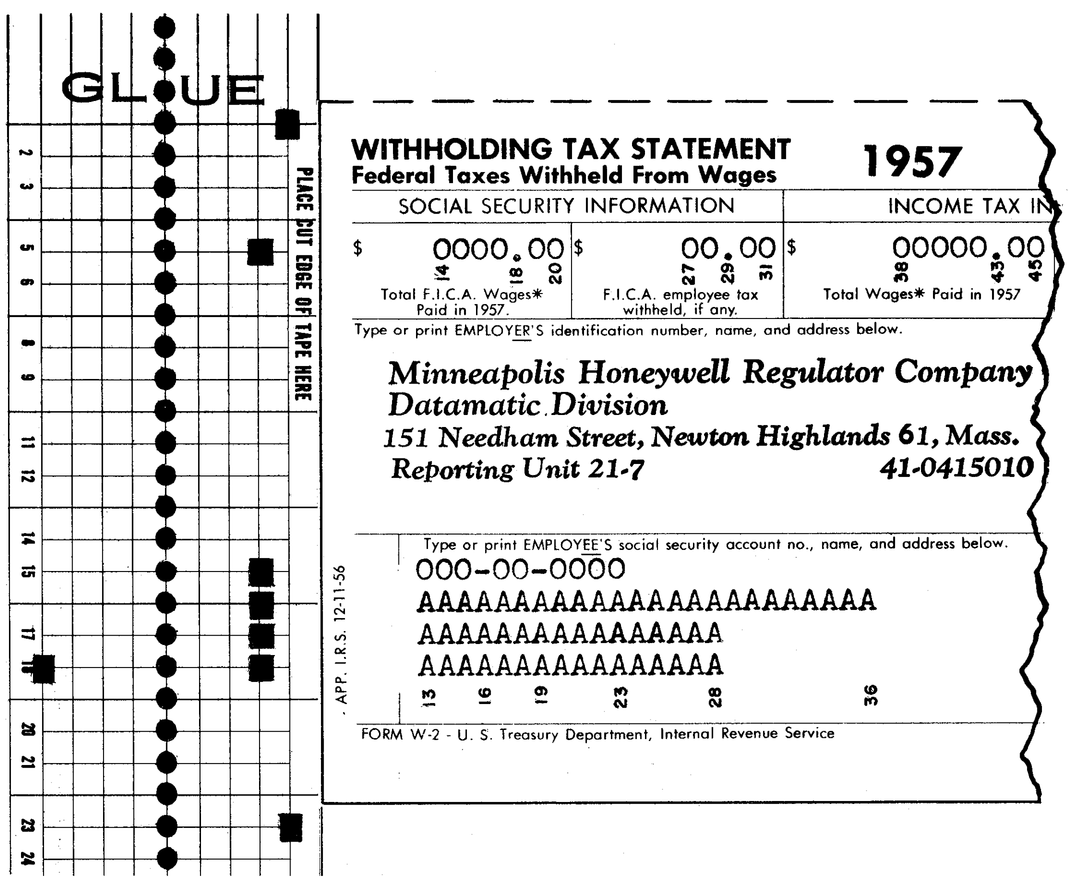


Figure 4-25

MANUAL CONTROL PANEL WORKSHEET

Step	Control Word	Blockette	Advance Block	Last Step
1	Manual No. 1	1	✓	
2	Manual No. 2	1		
3	Even	2		
4	Manual No. 3	1		
5	Manual No. 3	2		
6	Manual No. 1	3		
7	Manual No. 2	3		
8	Even	4		
9	Manual No. 3	3		
10	Manual No. 3	4		✓
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				

MCW	1	2	3	4	5	6	7	8
TWC	4	1	6					
VF	2	2	2					
CV	01	10	00					
HF	01	01	01					
F1	0	1	0					
F2	0	0	0					
F3	0	0	1					
F4	1	0	0					
F5	1	0	0					
F6	1	0	0					
F7	1	0	0					
F8	1	0	0					
F9	1	0	0					
F10	0	0	0					
F11	0	0	0					
F12	0	0	0					
F13	0	0	0					
F14	0	0	0					
F15	0	0	0					
F16	0	0	0					
F17	0	0	0					
F18	0	0	0					
F19	0	0	0					
F20	0	0	0					

(Binary bars omitted for simplicity)

Figure 4-26

Manual Control Panel: Figure 4-26 is the worksheet for wiring the Manual Control Panel. The lower portion of the panel contains the wiring for the pertinent bits of the three manual control words, plus the weight count values of these bits. The upper portion of this panel contains the wiring which directs the system through the control cycle to process the two items in a tape block. This includes the designation of the control word and the channel selection to be used, the last step in the cycle, and the step prior to which the tape is advanced. The control cycle in this example consists of ten manual panel steps. The tape is advanced prior to step 1, except when beginning a run. Thus the last step hub is wired to step 10 and the advance block hub to step 1. The first step also selects manual control word 1 and tape channel selection 1 to process the first line of printout. Step 2 reads the same channel selection and processes it under control of manual control word 2 to print the second line. Step 3 specifies channel selection 2 and passes control to the even word of the eighth channel in this selection. This word has the configuration $\overline{0000}$ in the tape channel selection bits (see Figure 4-24) and thus returns control to the next manual control step. Step 4 reads channel selection 1 and processes it under control of manual control word 3, while step 5 operates under the same control word upon channel selection 2. Steps 6 through 10 repeat the same cycle of control words, but read channel selections 3 and 4 to process the second item in the tape block. At the conclusion of step 10, the stepping switch resets to 1, the tape advances, and the cycle is repeated. Figure 4-27 is a summary table of the ten-step control cycle which prints two W-2 tax forms.

Horizontal Format Control Panel: In order to demonstrate some of the features of the horizontal format panel, all five lines of printing in the withholding tax problem are wired on a single such panel. This wiring would be somewhat simpler if all four horizontal format panels were used, but by using only one it is possible to illustrate the use of certain special control functions.

Printout Line	Contents	Tape Words	Channel Selec.	Conv. Panel	Control Word
Item 1, line 1	Tax Information	11, 12, 25, 26	No. 1	01	Manual No. 1
Item 1, line 2	Soc. Security No.	29	No. 1	10	Manual No. 2
Item 1, line 3	Employee Name	2, 3, and 4	No. 2	11	Tape No. 30
Item 1, line 4	Street Address	5 and 6	No. 1	00	Manual No. 3
Item 1, line 5	City and State	7 and 8	No. 2	00	Manual No. 3
Item 2, line 1	Tax Information	41, 42, 55, 56	No. 3	01	Manual No. 1
Item 2, line 2	Soc. Security No.	59	No. 3	10	Manual No. 2
Item 2, line 3	Employee Name	32, 33, and 34	No. 4	11	Tape No. 60
Item 2, line 4	Street Address	35 and 36	No. 3	00	Manual No. 3
Item 2, line 5	City and State	37 and 38	No. 4	00	Manual No. 3

Figure 4-27

Figure 4-28 is the worksheet for wiring the horizontal format panel. Note the following:

(1) Fields 1-3 are used for control purposes. Thus, the unconditional hubs for these fields are wired through an OR gate to special end. This signifies that no information is printed in these fields, but that the value of the corresponding field selection bits (33-35) is consulted to determine the settings of the auxiliary flip-flops to which these fields are wired. Fields 1-3 are wired to auxiliary flip-flops 1-3, respectively.

(2) The first line of printing has five separate fields, which are controlled by field selection bits 4-8. To distinguish this line from the others, these bit positions contain "ones" in the first manual control word and "zeros" in all other control words. The "one" hubs of fields 4-8 are wired to print control A hubs 14, 27, 61, 38, and 51, the respective starting positions of these fields in print storage (see Figure 4-21).

(3) The remaining lines of printing all start at print storage position 13 and consist of a single field (field 4). The "zero" hub of field 4 is wired to print control A hub 13 and a "zero" in the fourth field selection bit signifies that one of the lines 2 through 5 is to be printed. Any differences among these lines are signified by the first three field selection bits.

(4) The first four fields of line 1 end at positions 20, 31, 61, and 45, respectively. Since the "one" hubs are active during the printing of this line, the corresponding print control B hubs are wired through an OR gate to a "one" end-of-field hub. The last field ends at position 57; therefore print control B hub 57 is wired to the "one" hub of end horizontal format.

(5) Normally, decimal points are not recorded on tape but are emitted on the horizontal format panel. In the first printout line, decimal points are required in positions 18, 29, 43, and 55. These hubs under print control B are wired through an OR gate to the "one" hub of the decimal point emitter.

(6) The last four lines of printing all start at the same position; however, line 2 ends at position 23, line 3 at position 36, and lines 4 and 5 at position 28. These three hubs under print control B are wired to AND gates and from there through an OR gate to the "zero" hub of end horizontal format. The other entries to these AND gates are wired from the appropriate auxiliary flip-flops. Auxiliary flip-flop 1, set by the "one" in field selection bit 1 of the second manual control word, controls the emitters and the end of horizontal format for the Social Security Number. Since two functions are controlled, the exit from the flip-flop is wired through a commoning set to two different AND gates, one of which controls each function. Auxiliary flip-flop 2, set by the "one" in field selection bit 2 of tape word 30 (or 60), is wired to the AND gate which controls the end of the third line. Auxiliary flip-flop 3, set by the "ones" in field selection bit 3 of manual control words 3 and 4, is wired to the AND gate which governs the end of the fourth and fifth lines.

(7) The format of the Social Security Number in line 2 requires hyphens in positions 16 and 19. These hubs under print control B are wired through an OR gate to one of the AND gates controlled by auxiliary flip-flop 1, which is in turn wired to the hyphen emitter.

HORIZONTAL FORMAT PANEL WORKSHEET

Field	Hub Type	Function
1	1 U 0	Set Flip-flop A1 to 1 Special End Set Flip-flop A1 to 0
2	1 U 0	Set Flip-flop A2 to 1 Special End Set Flip-flop A2 to 0
3	1 U 0	Set Flip-flop A3 to 1 Special End Set Flip-flop A3 to 0
4	1 PCB 18 PCB 20 0 PCB 16 + (A1=1) PCB 19 + (A1=1) PCB 23 + (A1=1) PCB 28 + (A3=1) PCB 36 + (A2=1)	Begin field in PCA 14 Emit Period (1) End field (1) Begin field in PCA 13 Emit Hyphen unconditionally Emit Hyphen unconditionally End horizontal format (0) End horizontal format (0) End horizontal format (0)

Field	Hub Type	Function
5	1 PCB 29 PCB 31	Begin field in PCA 27 Emit Period (1) End field (1)
6	1 PCB 61	Begin field in PCA 61 End field (1)
7	1 PCB 43 PCB 45	Begin field in PCA 38 Emit Period (1) End field (1)
8	1 PCB 55 PCB 57	Begin field in PCA 51 Emit Period (1) End horizontal format (1)

Notes: PCA(B) indicates Print Control A (B).

(Ax=y) indicates setting of specified flip-flop.

Each + sign indicates use of an AND gate.

Each "function" wired more than once requires use of an OR gate.

Each "hub type" wired more than once requires use of a commoning set.

Figure 4-28

OUTPUT FORMAT CONTROL PANEL: As in the case of the bond register example, all printed information is laid out by the horizontal format panel in the first 120 print storage positions. Since the desired output format is achieved in the print storage section, all that is required on the output format panel is a one-to-one connection of hubs 13 through 61 from the Print Storage section to the Output Column section in both halves of the panel. The withholding tax example is another instance which could utilize an output format panel whose 120 print storage hubs are wired to the corresponding 120 output column hubs.

OPERATION: To perform the begin run search, set the RUN SELECTOR switch to 2 and press the BEGIN RUN SEARCH button. The tape will be searched until the configuration $\overline{0}1\overline{0}$ is encountered in the even word of the eighth channel in channel selection 1. The following switch settings must be established before the printing run is started:

(1) One of the three STOP switches must be "on" to indicate at what point the operator wishes to stop,

(2) The PRIMARY CATEGORY and all SECONDARY CATEGORY switches are set to OFF since no category functions are performed in this example,

(3) The MFU MANUAL CONTROLS switch must be set to OFF to activate the START PRINT button.

(4) The MANUAL BOARD switch must be "on" to start the run on the Manual Control Panel. Two methods of utilizing this switch are described under "Converter Console Panel" on page 4-22. The reader is advised to review the paragraph pertaining to the MANUAL BOARD switch on this page and to note that either of the methods described will be satisfactory for the tax form problem.

When the last item of the printing run is reached, with the STOP-END OF RUN switch set to ON, the system will stop on the end-run control word, which contains a "one" in bit position 8. Since this control word governs the processing of the third printout line, the system will stop after printing only three lines of the last form. This form is then completed by setting the STOP-END OF FORM switch to ON and pressing the START PRINT button.

MODEL 1400 CONVERSION CODES

Tape Code	Model 1400 Code	Printed Character	Effect on Zero Suppression
1-bit Numeric Conversion			
0	000000	0	*
1	000001	1	*
1-bit Sign Conversion			
0	100000	-	*
1	001101	+	*
4-bit Hexadecimal Conversion			
0000	000000	0 (zero)	-
0001	000001	1	stops
0010	000010	2	"
0011	000011	3	"
0100	000100	4	"
0101	000101	5	"
0110	000110	6	"
0111	000111	7	"
1000	001000	8	"
1001	001001	9	"
1010	010010	B	"
1011	010011	C	"
1100	010100	D	"
1101	010101	E	"
1110	010110	F	"
1111	010111	G	"
6-bit Alphanumeric, 1st Quadrant			
000000	000000	0 (zero)	-
000001	000001	1	stops
000010	000010	2	"
000011	000011	3	"
000100	000100	4	"
000101	000101	5	"
000110	000110	6	"
000111	000111	7	"
001000	001000	8	"
001001	001001	9	"
001010	001010	' (apos.)	"
001011	001011	#	"
001100	001100	@	"
001101	001101	+	"
001110	001110	space	"
001111	001111	space	"

(Binary bars omitted for simplicity.)

* The 1-bit Numeric and 1-bit Sign rules have no effect on zero suppression. Each character converted under these rules will be printed but the zero condition previously in effect will remain in effect.

Table 4-1, page 1

Tape Code	Model 1400 Code	Printed Character	Effect on Zero Suppression
6-bit Alphanumeric, 2nd Quadrant			
010000	010000	space	starts
010001	010001	A	"
010010	010010	B	"
010011	010011	C	"
010100	010100	D	"
010101	010101	E	"
010110	010110	F	"
010111	010111	G	"
011000	011000	H	"
011001	011001	I	"
011010	011010	CR	"
011011	011011	. (period)	"
011100	011100	□	"
011101	011101	(	"
011110	011110	)	"
011111	011111	space	"
6-bit Alphanumeric, 3rd Quadrant			
100000	100000	- (minus)	starts
100001	100001	J	"
100010	100010	K	"
100011	100011	L	"
100100	100100	M	"
100101	100101	N	"
100110	100110	O (letter)	"
100111	100111	P	"
101000	101000	Q	"
101001	101001	R	"
101010	101010	=	"
101011	101011	$	"
101100	101100	*	"
101101	101101	¢	"
101110	101110	space	"
101111	101111	space	"
6-bit Alphanumeric, 4th Quadrant			
110000	110000	&	starts
110001	110001	/	"
110010	110010	S	"
110011	110011	T	"
110100	110100	U	"
110101	110101	V	"
110110	110110	W	"
110111	110111	X	"
111000	111000	Y	"
111001	111001	Z	"
111010	111010	: (colon)	"
111011	111011	, (comma)	"
111100	111100	%	"
111101	111101	1/2	"
111110	111110	space	"
111111	111111	space	"

Table 4-1, page 2