IBM

Reference Manual

705 Data Processing System

Reference Manual

IBM 705 Data Processing System

Preface

This manual is a comprehensive presentation of the operation and use of the IBM 705 Data Processing System. Its purpose is twofold: (1) to provide a reference and guide for those already familiar with the system, and (2) to be used as an instruction aid in the development and training of both operators and programmers. The material is presented in a direct manner assuming that the reader is familiar with the information contained in the IBM *705 General Information Manual,* Form D22-6509.

The manual has been divided into sections; each section is developed to include related machine or functional operations. For example, information about any input-output unit used with the system is located in the "Machine Components" section. In the "Data Transmission Operations" section can be found the instructions used to move information from memory to memory, memory to storage, and storage to memory. Sectionalizing permits a more effective use of the manual by making possible quick references and a logical association of related material. The various sections are independent and need not necessarily be used in the order in which they are presented.

Each instruction is presented with a dual purpose. One part briefly presents pertinent information about the instruction. The second part is a description of the instruction in more detail, supplied with illustrations and examples. Sample programs, illustrating the use of instructions, are shown in actual machine language rather than autocoder. However, all autocoder mnemonics and considerable information involved in the use of autocoder is supplied in the appendix section of the manual.

Instructions relating only to the 705 III are so noted at the right of the instruction title. Differences in operation with instructions used on the 705 I, II, and III are indicated in the description of the instruction. Appendix A contains an explanation of the coding symbols and formulas used for calculating instruction execution times (to be found with each instruction description).

Contents

IBM 705 Data Processing System

IBM 705 Data Processing System

The IBM 705 Data Processing System is a high-speed, large-scale electronic computer designed to serve the needs confronting business, government, and science.

The 705 system is composed of an integrated system of record reading and writing devices interconnected through a central processing and control unit (CPU). Input to the system can be from magnetic tape, or IBM cards. Output is in the form of magnetic tape, punched cards, or printed reports. Data entered into the system or processed by the system may be letters of the alphabet, decimal numbers, or any of eleven punctuation marks or symbols.

The central processing unit contains the magnetic core memory, core storage units, and the circuits for performing all arithmetic operations and logical decisions. The CPU is controlled through the highly flexible medium of the stored program and other automatic and manual control circuits.

The primary core storage unit of the CPU is memory. Instructions for the control of the system and all data to be processed by the various units of the 705 system must be placed in core memory. Once stored in a specific location, the instructions and data are retained or "remembered" until new instructions or data are inserted in the same location. A highly flexible system for addressing individual character positions in memory makes possible the rapid selection of any stored record, group of records, field or individual character within a record. The 705 I has a memory capacity of 20,000 characters while the 705 II and 705 III have a memory capacity of 40,000 characters. An IBM 739 Magnetic Core Storage of 40,000 character positions may be used with the 705 III to provide a memory storage capacity of 80,000 characters.

In addition to the basic core storage, the CPU contains one accumulator core storage unit of 256 positions, one auxiliary core storage unit of 32 positions, and 14 auxiliary core storage units of 16 positions each. These units serve as a temporary storage for information being acted upon either logically or arithmetically.

The arithmetic section of CPU contains an add and subtract unit, controls for effecting multiplication and division, and controls for comparing the contents of accumulator and auxiliary storage with memory and other circuits. These circuits do not store data, but operate on these data as they are passed through. Their function is to receive data from memory and storage and process them in accordance with instructions obtained from the stored program.

The system is provided with internal automatic checking devices and control checks. Checks are made upon the handling of data within the CPU and upon the reading and writing of data by input-output units. The programmer can use these devices entirely at his own discretion. The console control, a separate unit of the 705 system, is used to control the machine manually and to monitor all operations.

Code for the IBM 705

Data are stored and transferred within the 705 systems using a seven-part code for each character. These seven parts consist of seven positions of binary representation and are divided into the three following groups:

CHECK	ZONE	NUMERICAL
C	B A	8 4 2 1

The binary number system uses two symbols, zero (0) or one (1), to represent all quantities. Zero (0) represents the absence of a quantity and one (1) represents the presence of a quantity. This quantity in binary representation is known as a "bit."

The four numerical bits have the assigned decimal values of 8, 4, 2 and 1. Numbers are interpreted as the sum of the numerical bits present. Considering only the numerical bit positions, the decimal 6 would be represented as 0110, the decimal 7 as 0111 and the decimal 8 as 1000. Figure 1 is a chart showing the 705 code bit arrangement for the decimal digits 1 through 9.

	C	B	A	8	4	2	1
1	1	0	0	0	0	0	1
2	1	0	0	0	0	1	0
3	0	0	0	0	0	1	1
4	1	0	0	0	1	0	0
5	0	0	0	0	1	0	1
6	0	0	0	0	1	1	0
7	1	0	0	0	1	1	1
8	1	0	0	1	0	0	0
9	0	0	0	1	0	0	1

Figure 1. Numerical Digit Bit Representation

The code logic used in the IBM card code is also used in the 705 code to represent alphabetic and special characters. The two zone bit positions, B and A, are used in combination with the four numerical bit positions to represent alphabetic and special characters. The four possible zone patterns are compared with IBM card code.

CARD CODE	ZONE BITS B	ZONE BITS A	705 CODE
12 zone	1	1	plus (+) zone
11 zone	1	0	minus (—) zone
0 zone	0	1	zero (0) zone
no zone	0	0	no zone

A bit in both the B and A positions represents the plus zone, a bit in the B position and a no bit in the A position represents the minus zone. No bit in the B position and a bit in the A position represents the zero zone. No bit in the B and A positions represents the no-zone condition and would be used when only numerical numbers are indicated.

The C bit position, known as the check bit, is used for checking. The check bit will be used when the sum of the zone and numerical bits representing the character is odd. If the number of bits in a character is even, without the C bit, the C bit is not used. The sum of the bits used to represent a character must be an even number including zone, numerical, and check bits, or the character is considered to be invalid. Figure 2 shows all the characters (in their bit representation) used in the 705 code.

The code system shown in Figure 2 is used in all components of the 705, including memory, accumulator and auxiliary storage, and drum storage. The conversion of the 705 code system to or from IBM card code is automatic whenever a card reader, printer, or punch is used.

	CHAR.	C BA 8421	CHAR.	C BA 8421	CHAR.	C BA 8421	CH.	C BA 8421	
	&	0 11 0000	-	1 10 0000	Blank	1 01 0000	Storage Mark and Drum Mark 0	0 00 0000	
ALPHABETIC	A	1 11 0001	J	0 10 0001	/	0 01 0001	1	1 00 0001	NUMERICAL
	B	1 11 0010	K	0 10 0010	S	0 01 0010	2	1 00 0010	
	C	0 11 0011	L	1 10 0011	T	1 01 0011	3	0 00 0011	
	D	1 11 0100	M	0 10 0100	U	0 01 0100	4	1 00 0100	
	E	0 11 0101	N	1 10 0101	V	1 01 0101	5	0 00 0101	
	F	0 11 0110	O	1 10 0110	W	1 01 0110	6	0 00 0110	
	G	1 11 0111	P	0 10 0111	X	0 01 0111	7	1 00 0111	
	H	1 11 1000	Q	0 10 1000	Y	0 01 1000	8	1 00 1000	
	I	0 11 1001	R	1 10 1001	Z	1 01 1001	9	0 00 1001	
	Plus Zero 0	0 11 1010	Minus Zero 0	1 10 1010	Record Mark	1 01 1010	Numerical Zero 0	0 00 1010	
SPECIAL CHAR	.	1 11 1011	$	0 10 1011	,	0 01 1011	#	1 00 1011	
	¤	0 11 1100	*	1 10 1100	%	1 01 1100	@	0 00 1100	
	Group Mark	0 11 1111					Tape Mark	0 00 1111	

Figure 2. IBM 705 Character Code Chart

Instruction

The 705 uses a five-character instruction to control its operations. An instruction consists of two parts, the operation part (single character) and the address part (four characters).

The operation part tells the machine which function to perform, such as read, write, add, and subtract. To the machine, for example, the letter Y means read. Thus, the instruction format would be: Yxxxx. In a like manner, the letter R means write, with the following format: Rxxxx.

Depending upon the operation part, the address part of the instruction can indicate the memory location of data to be operated upon, the address of an input or output unit to be used or the address of a check indicator or alteration switch. An example would be G 2050, the letter G meaning add and the number 2050 designating what address in memory is to be added. In the case where the address part pertains to a machine component, an example would be 2 0201. Here, the number 2 means select and the number 0201 means that tape unit 0201 will be used.

The operation charts in the appendix section of the manual show the codes for all operations and the addresses for all components.

The only distinction between instructions and data which are both stored in memory, is the way in which they are interpreted by CPU. For example, an instruction address may be treated arithmetically as data. The instruction operation part may be modified or even changed entirely by inserting a new character. One instruction can thus call for the modification of another instruction. It can direct the machine to compute a new address part or to substitute another operation part. Conversely, if for any reason data were entered in the memory location of instructions, the data could be acted upon as an instruction.

Instructions may be stored on drum, tape or cards. However, at the time they are to be executed by the 705, they must be stored in memory. Each time an operation is performed, the 705 gets the instruction from memory, decodes it, executes it, and then goes back to memory for the next instruction. Both the operation and the address part of an instruction are interpreted at the same time.

Stored Program

The work accomplished by the 705 in solving a problem or processing data consists of executing many instructions at high speed. The entire set of instructions used in solving a problem forms a *program* for the computer. Because the 705 holds and executes its instructions internally, it is called a stored program computer.

Normally, instructions are taken from sequentially ascending locations. However, the execution of instructions does not necessarily have to occur sequentially. It is possible, through *control* or *transfer* instructions, to alter the process of sequential execution and to indicate some particular position in memory as the location of the next instruction to be executed. In this way, it is possible to modify the sequence in which any instruction or block of instructions is executed.

For some control instructions, whether the next instruction is taken from the next sequential location or from some other location may depend on the result of a test, such as: *Is a number positive or negative?* In this case, the control operations are said to be conditional. By providing a stored program with the ability to control its course of execution, these conditional operations increase immeasurably the scope of the system.

There are no special areas of memory reserved for the location of a program. The location of the program, constants, and other data is entirely at the discretion of the programmer.

Memory and Memory Addressing

Memory is a core storage device containing a specific number of character storage positions. The 705 I has a memory capacity of 20,000 characters while the 705 II and III have a memory capacity of 40,000 characters. An IBM 739 Magnetic Core Storage of 40,000 character positions may be used with the 705 III to provide a memory storage capacity of 80,000 characters.

Each position of memory can store one character of information and has a specific address (location). When it is desired to place information in memory or remove information from memory, the address of the memory position to be used must be known. Each of the storage positions of memory has an actual five-digit numerical address, numbered from 00000 to the end of the particular memory, such as 19,999, 39,999 or 79,999.

Four character positions are provided in the address part of an instruction for addressing memory; thus, the five-digit actual memory address must be converted to a four-digit instruction memory address. This is accomplished by placing zone bit coding over the high-order and low-order characters of the address portion of the instruction. The zone bits are interpreted by the CPU memory addressing circuitry as the fifth-order position of an actual memory address.

B and A zone bits over the high-order position of the instruction address have a decimal value of 2 and 1, respectively. The B zone bit over the low-order position of the instruction address has a decimal value of 4. By adding the decimal values of the zone bits in various combinations, it is possible to represent all the necessary decimal digits used in the fifth-order position of an actual memory address. For example, the first 10,000 positions of memory, 00000 to 09999, have a zero value in the fifth-order position; therefore, no zone bits are required in the high- or low-order positions of the instruction address. Memory positions 20,000 to 29,999 have a high-order address value of 2; therefore a B bit over the high-order position of the instruction address is required. Memory addresses referring to the upper 40,000 positions of a memory always have a B bit in the low-order position, plus the proper B and A bits in the high-order position of the instruction memory address. A chart of actual memory addresses and coded memory addresses follows:

ACTUAL MEMORY ADDRESS	CODED MEMORY ADDRESS	
00000 to 09999	0000	9999
10000 to 19999	$\overset{01}{0}000$	$\overset{01}{9}999$
20000 to 29999	$\overset{10}{0}000$	$\overset{10}{9}999$
30000 to 39999	$\overset{11}{0}000$	$\overset{11}{9}999$
40000 to 49999	$000\overset{10}{0}$	$999\overset{10}{9}$
50000 to 59999	$\overset{01}{0}00\overset{10}{0}$	$\overset{01}{9}99\overset{10}{9}$
60000 to 69999	$\overset{10}{0}00\overset{10}{0}$	$\overset{10}{9}99\overset{10}{9}$
70000 to 79999	$\overset{11}{0}00\overset{10}{0}$	$\overset{11}{9}99\overset{10}{9}$

The B bit over the units position of an instruction address has not been assigned for use on the 705 I and II. Therefore, programs for these models could be run with equal facility on an 80,000 character machine. However, it is possible that a particular user of the Model I or II system may have assigned this B bit of the units position for other purposes.

Because the greater portion of memory is expressed in five digits, it will be the practice hereafter to refer to all positions of memory as five-digit addresses. That is, memory address 3525 will be written 03525, and address 73525 as 73525. Exceptions to this are illustrations, where attention must be drawn to the zone bit structure of the address. In such cases, zones will be shown above the digits involved (binary fashion), the B bit followed by the A bit. Thus, 73525 would be shown $\overset{11}{3}$ 5 2 $\overset{10}{5}$.

Any specific memory position is addressable, but because of the design and operating logic of memory, information is removed from memory and entered into memory as groups of five characters. The first five memory positions are one group, the second five memory positions the second group, and so on, to the end of memory (Figure 3).

Addressing any one of five sequential memory positions with a units digit of 0 through 4 removes the same five characters from memory. For example, any one of the following addresses, 00010, 00011, 00012, 00013, and 00014 selects the five memory positions 00010 through 00014. Likewise, addressing any one of five sequential memory positions with a units digit of 4 through 9 removes the same five characters from memory. Any one of the following addresses, 00015, 00016, 00017, 00018, and 00019, selects the five memory positions 00015 through 00019.

Each machine cycle using memory is a complete operation in respect to memory. That is, data are removed from memory and replaced into memory in one cycle. When characters are to be processed serially, the addressed character of the five removed from memory is selected for processing. In this instance, four characters are unused. Thus, the four unused characters and a processed character are returned to memory. When characters are to be processed in parallel, the five characters removed from memory may be returned unaltered or may be replaced by five new characters.

Characters stored in memory may be letters of the alphabet, decimal numbers, or any of the eleven punctuation marks or symbols used in report printing. Characters may be stored in any sequence to form fields of various sizes. A group of related fields may, in turn, be combined in any convenient arrangement to form records. Groups of records may also be combined to form files. The record may be any length within the capacity of memory. Once stored in a specific location, the record is retained or "remembered" until a new record is inserted in the same location.

Storage and Storage Addressing

The accumulator (ACC) and auxiliary storage units (ASU) are small core storage units, storing information temporarily from memory. Operations may then be performed on this information without changing the original field or record that remains in memory. The various operations are not actually performed by these units, however, but are executed in the arithmetic and logical unit (Figure 4).

One accumulator storage and fifteen auxiliary storage units are provided. The number 00 identifies the accumulator, with a capacity of 256 characters. The fifteen auxiliary units are identified by the numbers 01 through 15. Units 01-14 have a capacity of 16 characters each; unit 15 has a capacity of 32 characters.

Instructions using or involving accumulator or auxiliary storage must indicate the storage unit to be used. Zone coding (B and A bits) of the characters located in the tens and hundreds positions of the address part of the instruction specify the particular unit. The B and A bits of the character in the tens position have an assigned decimal value of 2 and 1, respectively. The B and A bits of the character in the hundreds position have an assigned decimal value of 8 and 4, respectively. The decimal sum represented by the presence of B and A bits in these two character positions indicates the addressed storage unit (Figure 5).

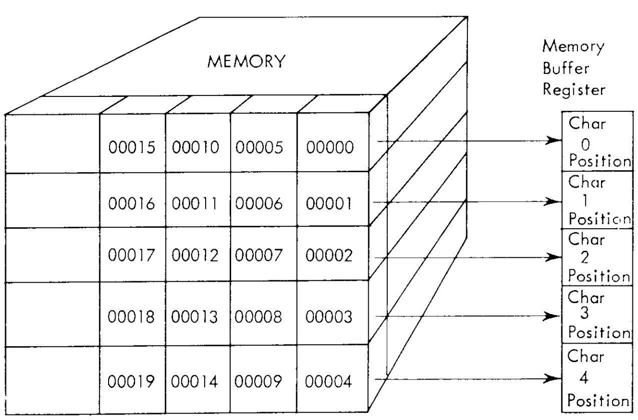


Figure 3. Schematic of Memory Address Assignment

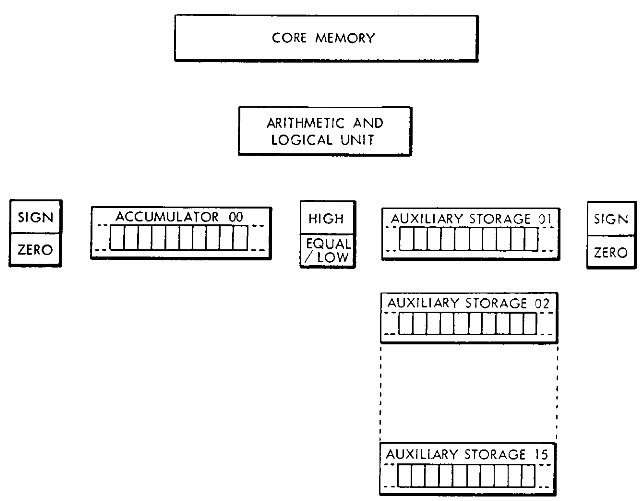


Figure 4. Schematic of Storage, Memory, and CPU

Hundreds Position Zone Bits (first B A pair); Tens Position Zone Bits (second B A pair)

B	A	B	A	
8	4	2	1	Assigned decimal value
0	0	0	0	Accumulator
0	0	0	1	ASU 1
0	0	1	0	ASU 2
0	0	1	1	ASU 3
0	1	0	0	ASU 4
0	1	0	1	ASU 5
0	1	1	0	ASU 6
0	1	1	1	ASU 7
1	0	0	0	ASU 8
1	0	0	1	ASU 9
1	0	1	0	ASU 10
1	0	1	1	ASU 11
1	1	0	0	ASU 12
1	1	0	1	ASU 13
1	1	1	0	ASU 14
1	1	1	1	ASU 15

Figure 5. Chart of Storage Coding

It will be the practice in this manual to show the ASU indication in parentheses following the address as in ADD 04759 (13), unless attention is being drawn to the bits themselves. In this case, the address will be shown as $0\ 4\ \overset{11}{7}\ \overset{01}{5}\ 9$. Where the zoning over a digit is (00), no zoning will be shown (i.e., $0\ 4\ \overset{00}{7}\ \overset{00}{5}\ 9$ will be shown) simply as 0 4 7 5 9.

There are instructions which, depending upon the operation part, use zone bit coding in the tens and hundreds positions of the address part of the instruction to indicate a particular type of operation but do not involve storage.

A special character called a *storage mark* normally occupies at least one position of accumulator and auxiliary storage. This character marks the left limit of the storage contents, and automatically appears in the proper position next to the highest-order character of the stored field. The mark is represented in text and programs by a letter "a" and internally in the 705 by 000 0000. The storage mark is peculiar to these storage units and normally does not appear in memory.

The starting point counter, containing the location of the right-hand character in the stored field, sets the right-hand limit to the field. All operations involving a field in either accumulator or auxiliary storage, therefore, operate only on those characters located between the storage mark and the starting point counter (Figure 6).

The field in accumulator or auxiliary storage can be shortened or extended to the left by shifting the storage mark. Only a field in accumulator storage can be shortened or extended to the right by shifting the starting point counter, for the position of the starting point in the auxiliary storage units is fixed. The operations of multiplication and division are also

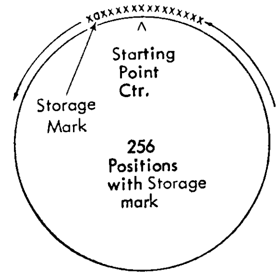


Figure 6. Schematic of Accumulator Storage

restricted to accumulator storage and cannot be performed in auxiliary storage.

Accumulator storage 00 can be represented in the form of a circle with 255 available positions and a storage mark. Auxiliary storage units 1-15 can also be represented as being grouped to form a circle with the units arranged in sequence around the circle. By certain operations, coupling of auxiliary storage units enlarges the capacity in one or more units up to the total capacity of the 256-position circle.

Figure 6 shows an accumulator field of six positions, marked off by the starting point counter and the storage mark. Figure 7 shows auxiliary storage units laid out using the circle method. A storage mark in each unit is automatically placed to the left of the highest-order character of a stored field.

Positive and negative fields are stored as true numbers in accumulator or auxiliary storage. Two sets of sign indicators register the sign of the fields; one set serves accumulator storage, the other set serves all auxiliary storage units. The sequence of operations within a procedure may be changed depending upon whether the sign of accumulator or auxiliary storage is plus or minus, or the result is zero (Figure 4).

A field in an ASU or the ACC may also be compared against another field in memory. Neon comparison indicators, on the 705 I and II, register a high or equal condition resulting from this comparison (Figure 4). A low comparison is not registered. One set of indicators serves the accumulator and all auxiliary

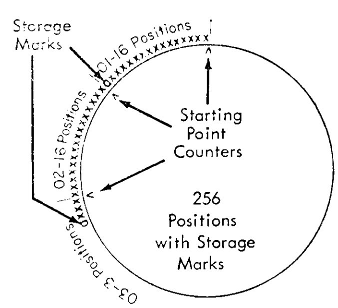


Figure 7. Schematic of Auxiliary Storage

storage units. On the 705 III, the comparison indicators register a high or low condition resulting from this comparison.

The sequence of operations within a program may be varied, depending upon whether the factor in the particular storage unit is higher, lower, or equal to a specified factor in memory.

When arithmetic operations are performed, the ACC or the ASU contains one of the two fields to be used in a calculation. The second field is in memory. To calculate $A + B = T$, the factor A is in a storage unit while factor B is in memory. After the addition operation is completed, the result T replaces factor A in the storage unit. The result of the calculation always replaces the original field in the ACC or the ASU, with the exception that a result may be added directly to a field in memory from a storage unit. In this instance, the ACC or the ASU remains unchanged.

Accumulator (ACC) and auxiliary storage unit (ASU) can be used to rearrange data in memory. Fields, records, or any portion of either, can be taken from one location in memory to a storage unit and from there can be relocated in another part of memory to form any desired arrangement. Data cannot be transferred directly from one storage unit to another but must first pass through memory.

Components and Component Addressing

Each component of the 705 system, input-output, check indicators and alteration switches have a specific address. The address provides the means of selecting, from all the components attached to the CPU, the desired component for a particular operation. A component address is distinguished from a memory address by the operation part of the instruction. When components are addressed, the data in memory and storage are not altered. The following addresses for component assignment are used for all models of the 705.

I-O ADDRESSES	CONTROL UNIT	ATTACHED UNITS
0100-0199	759 Card Reader Control	(1) 714 Card Reader
0200-0299	754 Tape Control	(10) 727 Magnetic Tape Units
0200-0299	760 I Control and Storage	(1) 720A Printer (2) 727 Magnetic Tape Units
0200-0299	760 II Control and Storage	(1) 730A Printer (2) 727 Magnetic Tape Units
0200-0299	767 Data Synchronizer	(5) 729 I Magnetic Tape Units (5) 729 III Magnetic Tape Units
0300-0399	758 Card Punch Control	(1) 722 Card Punch
0400-0499	757 Printer Control	(1) 717 Printer
0400-0499	774 Tape Data Selector	(1) 407/408 Printer or 519 Punch
0500		(1) Typewriter
0600-0699	777 Tape Record Coordinator	(8) 727 Magnetic Tape Units
1000-9999	734 Magnetic Drum	

CHECK INDICATOR ADDRESSES	INDICATOR FUNCTION
0900	Instruction Check Indicator
0901	Machine Check Indicator
0902	Read-Write Check Indicator
0903	Record Check Indicator
0904	Overflow Check Indicator
0905	Sign Check Indicator

ALTERATION SWITCH ADDRESSES
0911
0912
0913
0914
0915
0916

Input-Output Addressing

Data can be entered into the 705 system from both IBM cards and magnetic tape. Results can be printed, punched in IBM cards or written on magnetic tape. All input-output units are connected to the CPU through a control unit, with the exception of the typewriter. The control units synchronize and control the flow of data between CPU and the input-output unit. All data sent to or received from an input-output unit passes through the I-O (input-output) cable. This cable is common to all I-O devices in the 705 system (except the IBM 767 Data Synchronizer). Therefore, the addressed unit must be capable of determining when it is to be used (Figure 8).

The address part of an instruction, directed to an input-output unit, signifies a particular control unit and a specific functioning unit attached to it. In some instances, more addresses are provided for a particular type of equipment than will normally be used. The address circuitry for a unit can be adjusted to respond to any of the provided addresses. The address to which the unit will respond is normally set at the factory, but may be changed to respond to any one of the provided addresses. No two units of equipment should have the same address.

Each tape unit is provided with an address selection switch which may be positioned by the operator to indicate any one of the ten possible addresses for tape units. For a tape unit to be selected, the address selection switch setting must be identical to the digit con-

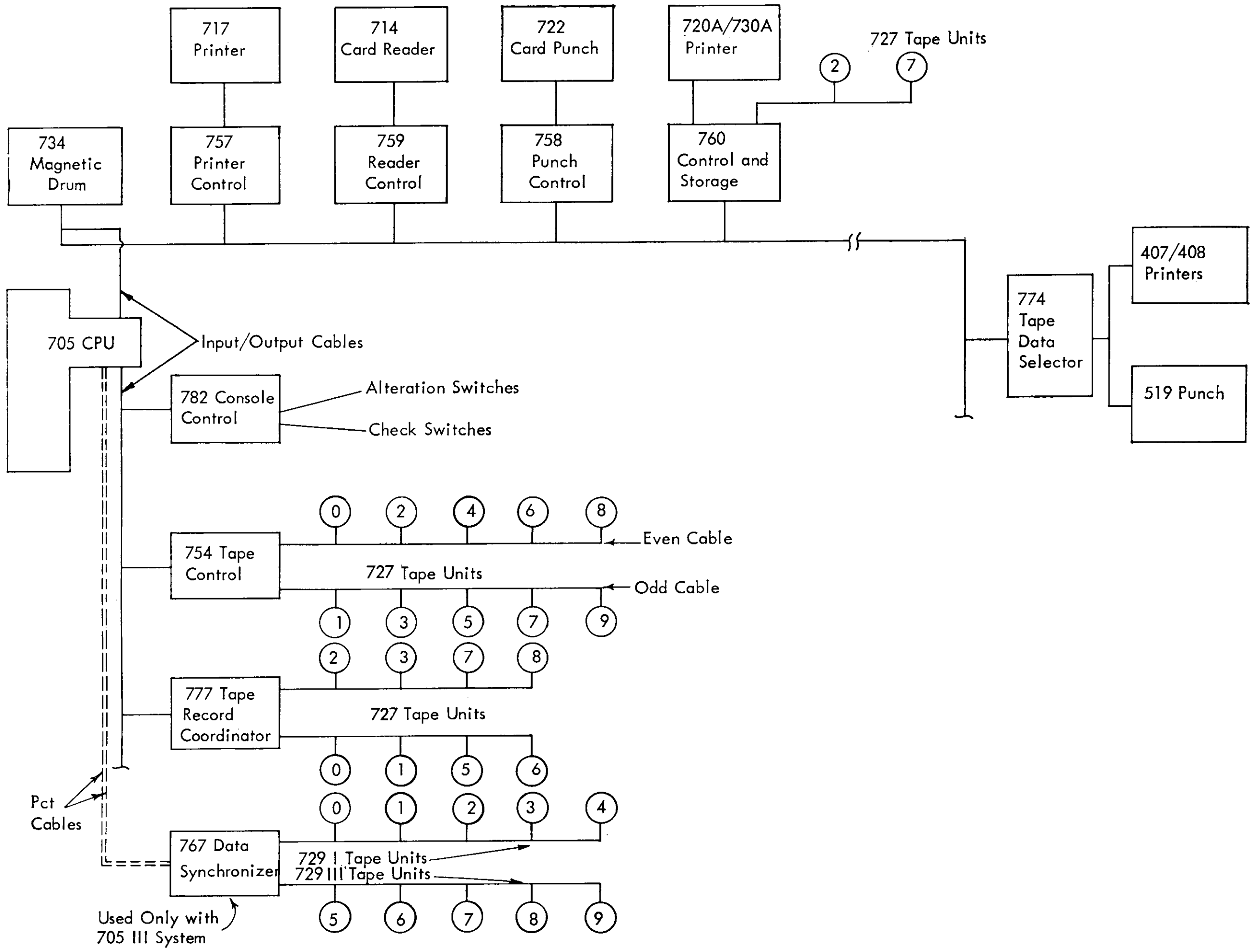


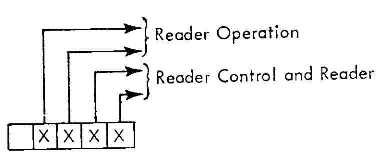
Figure 8. Schematic of IBM 705 Data Processing System with Components

tained in the units position of the address part of the instruction. The selector switches should not be set so that more than one tape unit has the same address on any one control unit.

The operating logic and use of the input-output equipment is discussed in the section entitled "Machine Components."

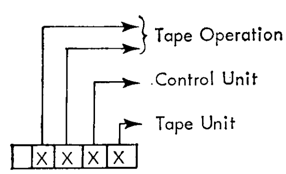
IBM 759 Reader Control and IBM 714 Card Reader (0100-0199)

Each reader control connects one card reader to the CPU. A typical address for a system using one card reader is 0100. A second reader could be 0101 or any of the other available addresses. The following illustration shows the basic address assignment.



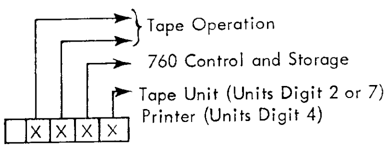
IBM 754 Tape Control (0200-0299)

The IBM 754 tape control (TC) connects as many as ten 727 tape units to the CPU. All tape units are connected to the TC by two signal cables referred to as "odd" or "even." As many as five tape units may be attached to each cable. Tape units attached to the "odd" cable must have odd numbered addresses. Tape units attached to the "even" cable must have even numbered addresses. A typical address for the TC would be 0200. This address specifies the TC numbered 0 and the tape unit 0 attached to the control unit. The following illustration shows the basic address assignment.



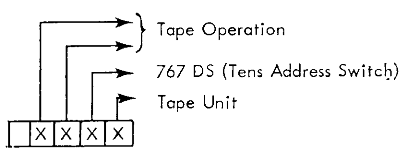
IBM 760 Control and Storage (0200-0299)

The 760 Control and Storage, models I and II, connect two 727 tape units and a 720A or 730A Printer, respectively, to CPU. Only one tape unit or printer may be selected at one time during any given operation. To address the tape units with the 760, the units digit of the address must be 2 or 7. To address the printer with the 760, the units digit of the address must be 4. Normally, the 760 address circuitry is set to respond to the digit 6 in the tens position of the address. If more than one 760 is to be used, another address must be assigned. A typical address for the 760 operation would be 0264. This address would indicate the 760 numbered 6 and the printer attached to it. The following illustration shows the basic address assignment.



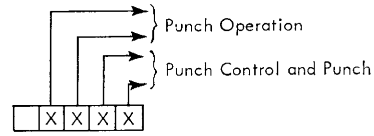
IBM 767 Data Synchronizer (0200-0299)

With the 705 III system, the IBM 767 Data Synchronizer (DS) connects as many as ten 729 tape units to CPU by the parallel character transmission cables (PCT). Each DS can accommodate as many as five 729 I and five 729 III tape units. A maximum of six data synchronizers may be in use if only 729 I tape units are used. With one or more 729 III tape units, only four data synchronizers may be in use. For the addressed DS to be selected, the tens address switch on the DS operator's panel must be set to indicate the digit in the tens position of the address. The following illustration shows the basic address assignment.



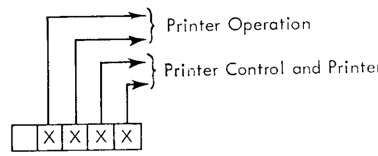
IBM 758 Punch Control and IBM 722 Card Punch (0300-0399)

The IBM 758 Punch Control connects one 722 Card Punch to the CPU. A typical address for a system using one card punch is 0300. A second punch could be 0301 or any of the other available addresses. The following illustration shows the basic address assignment.



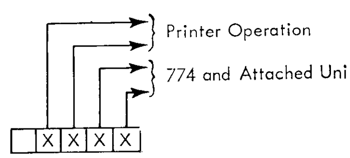
IBM 757 Printer Control and IBM 717 Printer (0400-0499)

The IBM 757 Printer Control connects one IBM 717 Printer to the CPU. A typical address for a system using one printer is 0400. A second printer could be 0401 or any of the other available addresses. The following illustration shows the basic address assignment.



IBM 774 Tape Data Selector (0400-0499)

The IBM 774 Tape Data Selector (TDS) connects either an IBM 407-408 Accounting Machine or IBM 519 Document Originating Machine to the CPU. The TDS uses the same address assignments as the 757 Printer Control. If both units are to be used on the same system, each must have a separate address. The operating logic and use of the TDS can be found in the IBM *774 Tape Data Selector System, Manual of Operation,* Form 222-6688. The following illustration shows the basic address assignment.

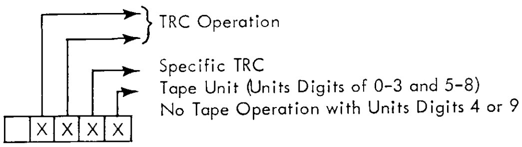


IBM Typewriter (0500)

One address, 0500, has been assigned for the typewriter.

IBM 777 Tape Record Coordinator (0600-0699)

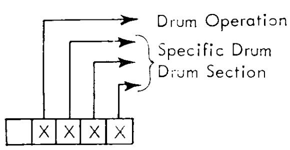
The IBM 777 Tape Record Coordinator (TRC) connects as many as eight 727 tape units to the CPU. The TRC may be addressed using the attached tape units as input or output units. The address may also designate an operation between the TRC and memory that does not involve a tape unit. A units digit of 4 or 9 in the address indicates to the TRC that no tape unit is involved in the operation. Units digits of 0-3 and 5-8 are used to address the tape units. The operating logic and use of the TRC can be found in the IBM *777 Tape Record Coordinator, Manual of Operation,* Form 222-6674. The following illustration shows the basic address assignment.



IBM 734 Magnetic Drum (1000-9999)

Each drum has 300 addressable sections; each section can store as many as 200 characters. Assuming that one drum is attached to the CPU, the addresses 1000 to 1299 would be used. The beginning section of the drum would have the address 1000; the next section, 1001; the following section, 1002, and so on.

Assuming that this 705 system has several drums, the following address scheme could exist. Addresses 1000-1299 would pertain to the sections on the first drum; addresses 1300-1599 would pertain to the sections on the second drum; addresses 1600-1899 would pertain to the sections on the third drum, and so on. The addressing system allows for the addressing of thirty drums. The following illustration shows the basic address assignment.



Check Indicator Addressing

The IBM 782 Console Control has six check indicators, each of which is associated with a separate switch. When a switch is set to PROGRAM, the corresponding check indicator may be interrogated during the program. The indicators are selected in the program in the same manner as any other component. Each indicator can be addressed by the address portion of a select instruction. In the 705 III, the interrogation of a check indicator can be accomplished with a transfer-on-signal instruction, without a previous select instruction. Check indicators and their assigned addresses are:

Instruction Check Indicator	0900
Machine Check Indicator	0901
Read-Write Check Indicator	0902
Record Check Indicator	0903
Overflow Check Indicator	0904
Sign Check Indicator	0905

Alteration Switch Addressing

The IBM 782 Console Control has six alteration switches. Each switch may be turned on or off by the operator and can be interrogated by the program. The alteration switches are selected in the program in the same manner as any other component. Each switch can be addressed by the address portion of a select instruction. In the 705 III, the interrogation of an alteration switch can be accomplished with a transfer-any instruction without a previous select instruction. Alteration switches and their assigned addresses are: 0911 to 0916.

CPU General Organization and Functional Logic

The purpose of this section of the manual is to provide a simplified version of the 705 machine logic. It is presented as an aid in developing a better concept of computer operation and to relate machine operation to memory and storage addressing, data flow and instructions.

Counters and Registers

Of the many and varied electronic devices in the computer, the register and counter are most important in this discussion. The register and counter are temporary storage devices. Registers hold numerical or alphabetic information while counters normally hold only numerical information. Also, counters differ from registers in that they have the ability to count, that is, the numerical value represented by the counter can be increased or decreased by certain specific amounts. This is known as stepping the counter. The number of character positions of a counter or register depends upon the particular function for which it is to be used.

A counter or register can be "set" to all the information or part of the information held by another counter or register. After being set, a counter or register has information identical to the counter or register to which it was set. Counters or registers can also be "reset." When a counter or register is reset, it is returned to some prescribed recognized condition. For example, a counter may be reset to indicate a zero condition.

The basic functioning unit of the register and counter is the trigger. The trigger is an electronic device, logically similar to a switch, that can indicate one of two conditions, on or off. Once it is set to one condition, it remains in that condition until it is changed. When used in a counter or register, it indicates the presence or absence of a bit of information. Because information within the computer is in binary form (bit or no bit), the trigger is an ideal storage device. The 705 uses a 7-bit code; therefore, seven triggers are required to represent the seven possible bits of a character. A five-character position register would require 35 triggers to represent all possible bits.

The trigger can also be used to indicate numerous conditions in the machine. For example, the fact that a field of data in storage is signed plus or minus can be indicated by one trigger. For illustration purposes, it could be assumed that if the trigger is on, the field is plus; if the trigger is off, the field is minus.

Instruction and Execution Time

To read, interpret, and execute an instruction, the central processing unit must operate in a prescribed logical sequence. Control of CPU is determined by the specific instruction to be performed and certain exact timing impulses provided by an electronic clock. The timing impulses are provided at the rate of one million a second, or, each pulse is one microsecond (μs) in duration. A series or group of these pulses constitutes a machine cycle, and represents the amount of time used to perform a specific machine operation. Machine cycles in the 705 I and II are 17 or 9 microseconds in duration while machine cycles in the 705 III are 13 or 9 microseconds in duration.

The first cycle of each 705 instruction to be performed is the instruction cycle, referred to as instruction time (I time). There is one I time cycle of 17 microseconds for each instruction to be performed using the 705 I or II. In the case of the 705 III, I time is 13 microseconds. During I time, the five characters composing a particular instruction are read from memory. The operation part of the instruction is interpreted and controlling circuits to be used in the operation are activated. The address portion of the instruction is set in certain registers and counters for reference. The instruction is then placed back in memory.

The instruction cycle may be followed by one or more execution cycles referred to as execution time (E time). During execution time, all operations necessary to perform the desired instruction are carried out. The 705 I and II have execution cycles which may be 17 or 9 microseconds in duration depending on the operation to be performed. All execution cycles in the 705 III are 9 microseconds in duration. The number of execution cycles comprising execution time for any one instruction depends upon the particular instruction being executed. The last execution cycle of one instruction will be followed by the instruction cycle of the next instruction.

Memory Address Controls

Use Figure 9 as a reference aid in the following discussion. Starting a typical program will be preceded by setting the *instruction counter* (IC) to the memory address of the first instruction to be executed. The IC can be reset to 00004 if the first instruction to be executed is located in the first five positions of memory. The IC can also be manually set from the IBM 782 Console Control to indicate the memory address of any instruction located in memory.

Just preceding the first instruction cycle, following the use of the start key, the *memory address counter 1* (MAC I) sets to the address held by the IC (705 I and II). At the beginning of the instruction cycle, the *memory address select register* (MASR) sets to the address held by MAC I (705 I and II). In the case of the 705 III, MASR sets directly to IC during the instruction cycle. The five-digit memory address held by MASR is decoded and selects the five memory positions, containing the instruction, as originally indicated by the IC.

The five characters comprising the instruction are removed from memory and are set in the *memory buffer register* (MBR). The characters are held by MBR during the instruction cycle and are then replaced in memory at the same location from which they were

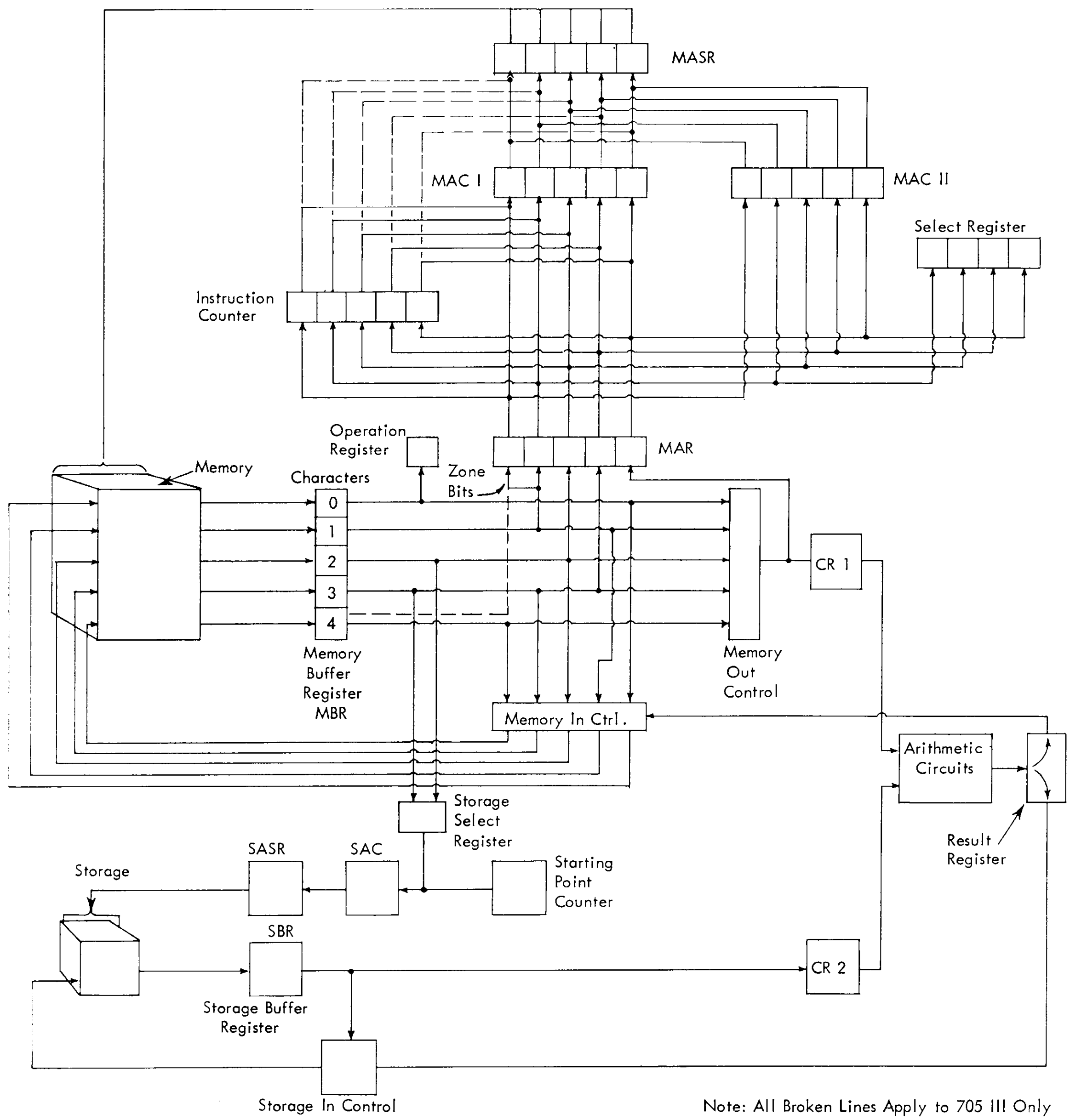


Figure 9. Schematic of Memory and Storage Addressing Controls

removed. The bits of the characters as held by MBR are made available to the *operation register* (OP REG), the *memory address register* (MAR), *storage select register* (SSR), *memory-out control* and the *memory-in control.*

Core memory design fixes a relationship between the five selected memory positions and each of the five character positions of MBR. Characters located at memory addresses ending in 0 or 5 are always placed in the character 0 position of MBR; addresses ending in 1 or 6 are placed in the character 1 position of MBR, and so on (Figure 3). The digit contained in the units position of MASR is decoded and determines, except in certain 705 III operations, which one of the five characters in MBR is the addressed character. The addressed character will be routed through the memory-out control to the character register 1 (CR 1). During instruction cycles, the units position of MASR should contain a 4 or 9. This results in routing the character in MBR character 4 position into CR 1.

During the instruction cycle only, the character contained in MBR character position 0 is placed in the OP REG. The character is decoded to determine the operation to be performed and to activate the necessary machine circuitry to perform the specific instruction. If the character placed in the OP REG is not a valid character, an instruction check error is indicated. The OP REG holds the operation part of the instruction from one instruction cycle to the next instruction cycle.

So that the operation part of the instruction will be placed in the OP REG during the instruction cycle, the units position of the instruction must be located in a memory address with a units digit of 4 or 9. This is necessary because the five characters that make up an instruction must be moved from memory to MBR as a block of five characters for correct interpretation by the machine.

The first four order positions of the memory address register (MAR) set to only the numerical portion of the address part of the instruction as held by MBR (Figure 10). The zone bits over the high-order position of the address portion of the instruction are set in the fifth-order position of MAR. In the case of the 705 III, which can have 80,000 positions of memory, the B bit of the MBR character 4 position is also set in the fifth-order position of MAR.

The five-digit address as held by MAR can be interpreted as a memory address, an input-output address, check indicator or alteration switch address, depending upon the operation part of the instruction set in the operation register.

In Figure 10, the address portion of an instruction held by MBR is JHK6, interpreted by MAR to be a memory address of 21826 (705 II or III). For ease of understanding, portions of the binary representation have been converted to decimal equivalents. The 705 I would interpret this address to be 11826 because any zone in the MBR character 1 position addresses only the second 10,000 positions of memory. See the following chart for the relationship of zone coding and the decimal equivalents for the fifth-order position of MAR.

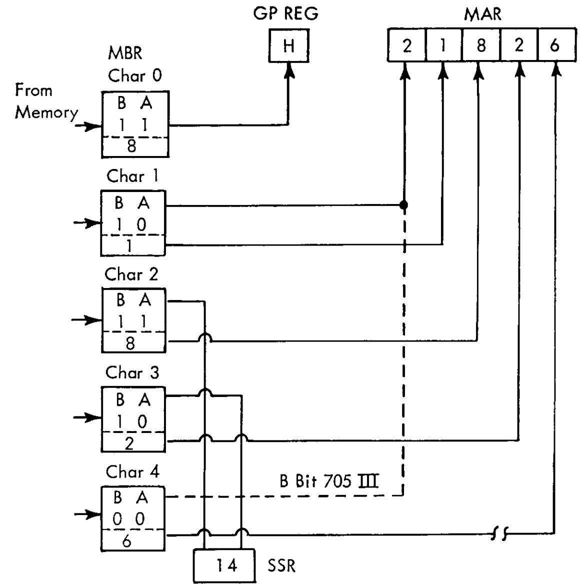
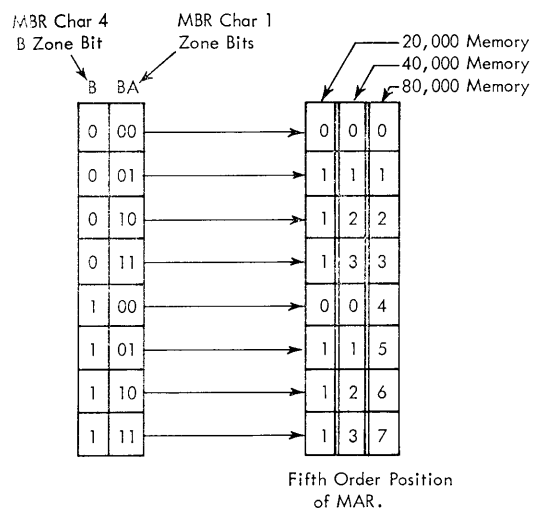


Figure 10. Schematic of MBR Operation



After the OP REG and MAR have been set, the IC is stepped by 5, or is set to MAR. In most instances, the address of the next instruction of the program is five memory positions higher than the address of the present instruction being analyzed. By automatically stepping the IC plus five during the instruction cycle (705 I or II) the memory address of the next instruction to be used is determined. The IC does not step until after the instruction cycle, in the 705 III.

If the next instruction to be executed is not sequentially the next instruction of the program, the IC sets to the memory address in MAR. Under these circumstances, the address in MAR is the memory address of the next instruction to be read from memory.

If the address placed in MAR during the instruction cycle is the address of an input-output unit, alteration or check indicator switch, the select register sets to MAR. The address in the select register makes possible the selection of the addressed unit. The address in the select register remains until it is changed by another instruction indicating the use of the select register.

Near the end of the instruction cycle, MAC I sets to the MAR, which contains the memory address for the next machine cycle. The five characters in MBR are returned to memory, through the memory-in control at the address indicated by MASR.

Some instructions require that the memory address counter 2 (MAC II), rather than MAC I, sets to MAR. MAC I and MAC II are similar in function in that each indicates the next memory address to be used. Two memory address counters make it possible to have two different memory addresses indicated. The operation to be performed selects the particular memory address counter to be used.

At the beginning of an execution cycle which involves removing information from memory for processing, MASR sets to MAC I or MAC II depending upon the instruction to be performed. During succeeding execution cycles, the memory address indicated by MAC I or MAC II can be increased or decreased (stepped) automatically under control of the instruction.

The five-digit address in MASR is decoded and the proper block of five characters is removed from memory and placed in MBR. The addressed character of the five held by MBR is indicated by the digit in the units position of MASR, except in certain 705 III operations. The addressed character is routed through the memory-out control and set in CR 1.

From CR 1, the selected character can be routed through the arithmetic circuits to the result register (RESULT REG). A character placed in the result register from memory or storage can be returned to memory or storage depending upon the operation to be performed. If the character in the result register is to be returned to memory, it is routed to the memory-in control. The memory-in control allows the four unselected characters from MBR and the character from the result register to be placed in memory. The character from the result register is placed in the memory position that originally held the addressed character. If the character in the result register is to be placed in storage, all five characters in MBR are returned to memory at the address from which they were removed.

During the next execution cycle, if the particular operation is to continue, MASR will set to MAC I or MAC II (which has stepped) and another execution cycle will be initiated.

Storage Address Controls

A storage unit is designated by the zone information in the tens and hundreds positions of the address portion of the instruction. During the instruction cycle, the zone bits in the character 2 and 3 positions of MBR are set in the storage select register (SSR).

Storage is not used during the instruction cycle, but the particular unit to be used during the following execution time is determined. If no zone bits are present in SSR, accumulator storage is indicated. The storage address counter (SAC) is, in this case, set to a storage position indicated by the *starting point counter* (SPC). Because accumulator storage functions as a circle of storage positions, it is only important that the starting position be determined and retained by SPC.

Zone bits set in SSR indicate the use of auxiliary storage. The zone bits are decoded and will designate which unit is to be used. Figure 5 illustrates the coding necessary for the selection of a particular auxiliary storage unit.

When auxiliary storage is indicated, the SAC sets to SSR and indicates the units position of the selected storage unit. Addressing any auxiliary storage unit always sets the SAC to the units position of the addressed storage unit (ASU).

Some instructions have zone bits in the tens and/or hundreds position of the address portion of the instruction, but do not indicate the use of auxiliary storage. In this case, SSR is used only as an indicator for the selection of one of several possible types of operation. For example, the read 00 and read 01 instructions are differentiated by the fact that the read 01 instruction has zone bits over the tens position of the address portion of the instruction. Instructions using SSR as an indicator do not affect the data in storage.

At the beginning of an execution cycle involving the use of storage, the *storage address select register* (SASR) sets to the storage address contained in SAC. The address in SASR is decoded and the addressed character is removed from storage, set in the *storage buffer register* (SBR) and routed to character register 2 (CR 2). The character in CR 2 can be routed through the arithmetic circuits to the result register. From the result register, the character can be routed to memory or storage, depending upon the operation to be performed.

If the character is to be returned to storage, it is routed from the result register to the storage-in control and placed in storage at the address indicated by SASR. If the character is to be returned to memory, the character in the SBR is returned to the storage position from which it was previously removed. During succeeding execution cycles, the storage address as indicated by SAC can be increased or decreased (stepped) automatically under control of the instruction being performed.

Information Flow

Figure 11 is a schematic of data flow through the 705 CPU. In general, data to be processed are removed from memory and/or storage, routed through the arithmetic circuits to the result register and returned to memory or storage. Under control of the instruction to be performed, the direction of data flow and the action of the various arithmetic circuits upon the data are determined.

Character Register 1 (CR 1)

Removing data from memory places five characters in the memory buffer register (MBR). The addressed character in MBR is routed through the memory-out control, set in CR 1, and code checked. An odd number of bits or the lack of bits in CR 1, turns on the CR 1 code check indicator, when required by the operation being performed. Character recognition is also accomplished in CR 1 by special decoding circuits. For example, there are circuits to recognize when CR 1 contains a zero, record mark, group mark, ampersand, and so on. Certain characters are used to modify action upon the control of the instruction being performed.

From CR 1, data can be routed to the input-output write buses or through the arithmetic circuits to the result register. Data to be written from memory during a write instruction, except when a DS is used, pass from CR 1, character by character, to the write buses in the input-output cable. The selected output unit then receives the characters from the write buses and performs the writing.

The zone portion, the numerical portion, and the check bit of a character in CR 1 to be processed have separate routes through the CPU circuitry to the result

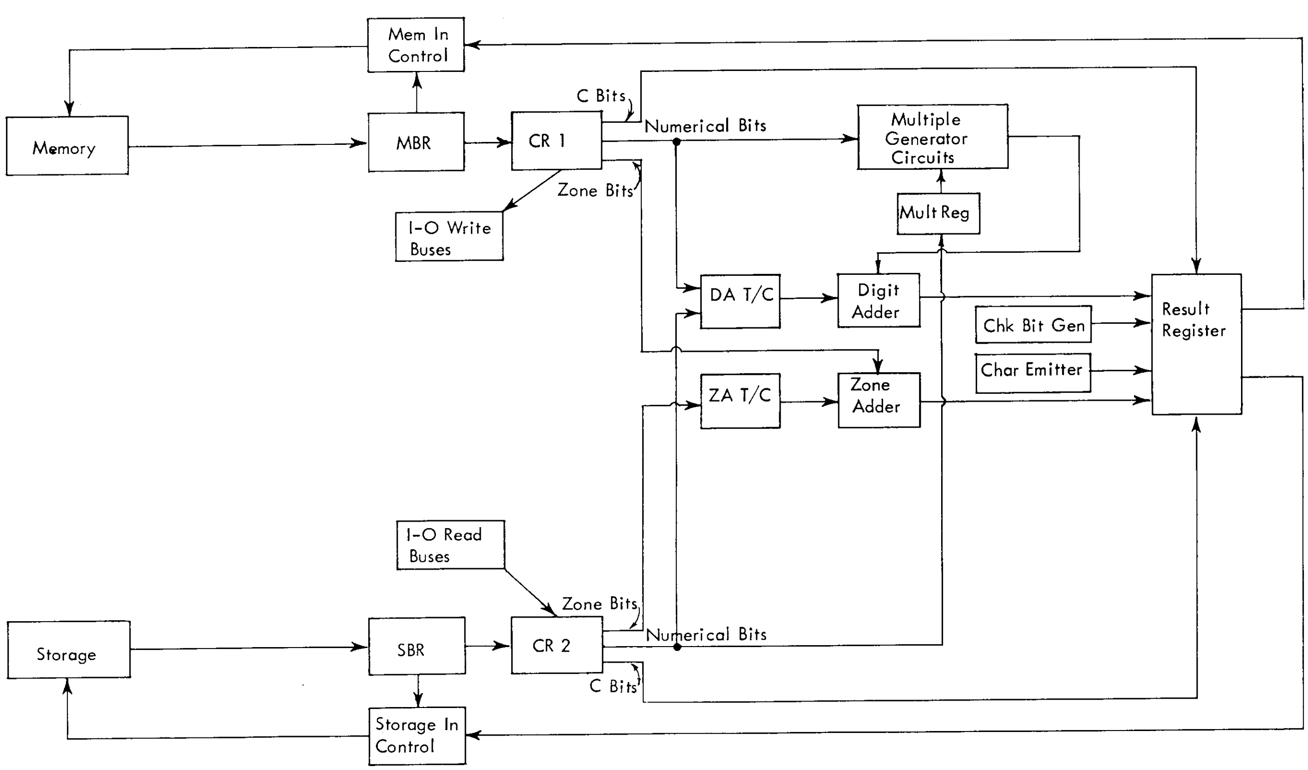


Figure 11. Schematic of CPU Information Flow

register. The particular instruction being executed determines what portion or portions of the character are to be routed into the result register.

Character Register 2 (CR 2)

Removing information from storage places a single character in the storage buffer register and routes the character into CR 2. Information can also be placed in CR 2 from the read buses of the input-output cable, during the execution of a read instruction, except when a DS is used.

The character, as held by CR 2, is code checked. An odd number of bits in any character indicates a CR 2 code check error, when required by the operation being performed. CR 2 is also used for character recognition to sense special characters used to modify and control the operation being performed.

From CR 2, information can be routed through the arithmetic circuits to the result register or to the multiple register during a multiply instruction. The zone portion, the numerical portion, and the check bit of a character in CR 2 have separate routes through the CPU circuitry to the result register. The particular instruction being executed determines what portion or portions of the character are to be routed to the result register.

Numerical Bit Transmission

The numerical portion of a character routed from CR 1 can pass through the multiply circuitry to the digit adder or through the true-complement circuitry to the digit adder. The numerical bits (except in the ADM instruction) pass through the multiply circuitry to the digit adder. During instructions, other than multiply, the multiply circuitry is set to multiply by 1. If the instruction is MPY, the numerical bits of the character in CR 2 are set in the multiple register where they act as the multiplier and condition the multiply circuitry to generate the proper multiple of the digit from CR 1.

The numerical bits of a character routed from CR 2 always pass through the digit adder, true-complement (T-C) circuits to the digit adder. The T-C circuits complement (effective tens complement) or pass (unaltered) the input numerical bits, depending on the operation to be performed.

The digit adder can add the numerical bits of a character from CR 1 and CR 2 or it can allow the numerical bits from either CR 1 or CR 2 to pass through unaltered. When sums with a numerical value greater than nine are produced, the significant digit can be routed to the result register and the carry retained and added to the sum produced in the next machine cycle.

Zone Bit Transmission

Zone bits are routed from the character registers to the result register, apart from the numerical bits. The zone bits from CR 1 are routed to the *zone adder* directly. The zone bits from CR 2 are routed through the zone adder true-complement T-C circuits to the zone adder. The T-C circuits complement or pass, unaltered, the zone bit input. Complementing the zone structure of a character is done binarily by reversing the status of the bits. For example, if the zone bits are 01, the complement is 10; if the bits are 11, the complement is 00.

Zone bits from CR 1 and CR 2 can be added binarily in the zone adder or can be passed, unaltered, to the result register.

Check Bit Transmission

A check bit in CR 1 or CR 2 may be routed directly to the result register or a check bit may be generated by the check bit generator. If the numerical and zone bits of a character are moved through the arithmetic circuits to the result register, unaltered, the check bit (if present) of that character is routed into the result register. This is done to insure a code check on the number of bits in the character, after transmission through the arithmetic circuits.

If, during any machine cycle, the numerical and zone bits have a possibility of changing legitimately within the arithmetic unit, a check bit is automatically generated and placed in the result register.

Character Emitter

Some 705 instructions require that certain characters or specific bits of characters be emitted (generated) within the machine. The circuitry associated with this type of operation is often referred to as the character emitter. The character emitter circuitry can emit a blank, hyphen (minus zone), ampersand (plus zone) or a numerical zero (8 and 2 bit). In the 705 III, the changing of bits of a character using the set bit instruction involves the character emitter circuitry. The emitted character or bits of a character are set in the result register.

Result Register

The three bit parts of a processed character are assembled in the result register. They are:

1. The numerical part from the digit adder.
2. The zone part from the zone adder.
3. The C bit from CR 1 or CR 2 or from the C bit generator.

A character in the result register can be routed to memory or storage. If the character is routed to memory, one of the five characters read from memory is replaced by the character from the result register. If the character is routed to storage, the original character read from storage is replaced by the character from the result register.

Read and Write Operations

This section explains those instructions used or involved in moving information from an input unit into memory and from memory to an output unit.

Any specific details in the operation and use of an input-output unit related to these instructions can be found in the section "Machine Components."

Types of possible errors and error correction procedures can be found in the sections, "End-of File and Checking Procedures" and "Machine Components."

In all reading and writing instructions (except through a DS), the operation is executed in a single period of time. During reading or writing through a DS, data transfer may occur between instructions and may also interrupt the processing of an instruction. The time formulas for read and write operations with the DS indicate only the time required of CPU.

Select (2—SEL)

Function. This operation selects, from all the components attached to the CPU, the desired component for a particular operation.

Address. The address portion of the instruction is the address of the component.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	51	-------
705 III	40	-------

Description. The address of the select instruction specifies the one component, from the many attached to CPU, to which future action is to be directed. Only one component can be selected at a time and the selected device remains selected until another select instruction is given.

The machine differentiates between a memory address and the address of a component by the operation part of the instruction.

The address 0100, for example, when preceded by an *add* operation part, instructs the machine to add data stored at memory location 0100. When the address is preceded by a select operation part, the instruction refers to card reader 0100.

The select instruction, when addressing all input-output units except the card punch and typewriter, selects the associated input-output indicator. (Refer to the section "End-of-File Procedures" for use of input-output indicators.)

Read 00 (Y—RD)

Function. This operation reads a record from a selected input unit and stores it in memory.

Address. The address specifies the memory position into which the first character of the input record is to be placed. During use of the DS, the addressed memory position must be one having a units digit of 0 or 5.

Limiting Factors. The operation is limited by indication from the input unit.

Tape unit ------- Inter-record gap
Card reader ----- Reader storage mark
Drum ----------- Drum mark

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$10000 + 67N$	From tape to mem.
	$68 + 33.5N$	Card reader to mem.
	$8000 + 40N$	Drum to mem.
705 III	$58 + 9\frac{N}{5}$	From tape to mem. through *DS*
	$40 + 33.5N$	From card reader to mem.
	$8000 + 40N$	From drum to mem.

Description. The read instruction with 00 coding is used to enter information into memory from a previously selected input unit. Information is placed in memory, starting at the addressed memory position and continuing to successively higher memory positions, until terminated by a proper end-of-record indication.

The end-of-record indications for the input units are as follows:

1. Tape ---- Inter-record gap
2. Reader --- Reader storage mark (RSM) or last position of the record storage unit (RSU)
3. Drum ---- Drum mark

The inter-record gap, reader storage mark or drum mark are not entered into memory as part of the input record.

NOTE: DS operation.

During use of the DS, characters are entered into memory in blocks of five characters, rather than character by character. If the last block of characters from the input record is not completely filled, group marks are inserted in the unfilled positions. Use of the DS requires that the address portion of the RD 00 instruction have a units digit of 0 or 5.

Refer to the section "Machine Components" for further details of the RD 00 instruction related to the DS.

Read 01 (Y—FSP)

Function. A record is read from an input unit and checked but not entered into memory.

Address. Any memory address can be used except with the DS. With the DS, the addressed memory position must be one having a units digit of 0 or 5.

Limiting Factors. The operation is limited by an indication from the input unit.

Tape ------- Inter-record gap
Reader ------ Reader storage mark
Drum ------- Drum mark

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$10000 + 67N$	From tape to mem.
	$68 + 33.5N$	Card reader to mem.
	$8000 + 40N$	Drum to mem.
705 III	58	From tape to mem. through *DS*
	$40 + 33.5N$	From card reader to mem.
	$8000 + 40N$	From drum to mem.

Description. The read instruction with 01 coding permits a record to be read from an input unit, and the data checked but not entered into memory. The instruction is normally used to space over unwanted tape records, but is also used to check the validity of a previously written tape.

Except for DS operation, coding other than 01, such as 08, will be interpreted as 01.

NOTE: DS operation.

Use of the DS requires that the address portion of the RD 01 instruction have a units digit of 0 or 5.

Refer to the section "Machine Components" for further details of the RD 01 instruction related to the DS.

Read 02 (Y—RMA) — *705 III*

Function. This operation places the contents of the output buffer (five characters) of the selected DS in memory.

Address. The address indicates the memory position into which the first character of a block of five characters is to be located. The address must have a units digit of 0 or 5.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 III	130	-------

Description. The read instruction with 02 coding (read memory address) places the contents of the output buffer (OB) of the selected DS into memory at the location specified by the address of the instruction. The address portion of the instruction must have a units digit of 0 or 5. Zone coding other than 01 or 02 results in a RD 00 operation.

At the end of each read or write instruction involving the DS, the contents of the *synchronizer memory address counter* (SMAC) are set in the output buffer. The contents of SMAC are a five-digit memory address, greater by five than the location of the last memory reference held by SMAC.

Refer to the section "Machine Components" for further details of the RD 02 instruction related to the DS.

Write 00 (R—WR)

Function. This operation transmits a record from memory to a selected output unit.

Address. The address specifies the memory position from which writing is to begin. With the DS, the addressed memory position must be one having a units digit of 0 or 5.

Limiting Factors. Group mark in memory.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$10000 + 67N$	Mem. to tape
	$68 + 33.5N$	Mem. to punch or printer
	$8000 + 40N$	Mem. to drum
705 III	$94 + 9\frac{N}{5}$	From mem. to tape through *DS*
	$49 + 33.5N$	From mem. to punch or printer
	$8000 + 40N$	From mem. to drum

Description. The write 00 instruction transmits a record from memory to the selected card punch, printer, tape unit, drum, or typewriter.

Information is written from memory, from left to right, starting at the memory position specified by the address part of the instruction and continuing until a group mark is reached. The write instruction does not affect the record in memory.

No write operation occurs if the write instruction is addressed to the memory address of a group mark. A no-operation occurs and the machine proceeds to the next instruction.

Sensing the group mark in memory stops the writing operation. The group mark causes an inter-record gap to be automatically placed on tape; if drum is used, the group mark is converted to a drum mark at the end of the drum record.

NOTE: If a DS is being used, the units position of the address of the first character to be written must be a 0 or 5. Refer to the section "Machine Components" for further details of the WR 00 instruction related to the DS.

Write 01 (R—DMP)

Function. This operation transmits a record from memory to a selected output unit.

Address. The address specifies the memory position from which writing is to begin. With use of the DS, the addressed memory position must be one having a units digit of 0 or 5.

Limiting Factors. The end of the memory block in which writing began.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$10000 + 67N$	Mem. to tape
	$68 + 33.5N$	Mem. to punch or printer
	$8000 + 40N$	Mem. to drum
705 III	$94 + 9\frac{N}{5}$	From mem. to tape through *DS*
	$49 + 33.5N$	From mem. to punch or printer
	$8000 + 40N$	From mem. to drum

Description. The write instruction with 01 coding (dump memory) transmits a record from memory to the selected card punch, printer, tape unit, drum, or typewriter.

Information is written from memory, starting at the addressed memory position and continuing to successively higher memory positions until the end of the memory block is reached.

Except for DS operation, the last character written is located at memory position 19,999, 39,999, 59,999 or 79,999. To write the last character of a memory block, a write (01) instruction addressed to the last character of the memory block must be used. For example, write (01) 39,999 causes only one character of a 40,000-character memory to be written.

Any zone coding from 01 to 15 causes a WR 01 operation, except with the DS.

NOTE: Use of the DS requires that the address portion of the WR 01 instruction have a units digit of 0 or 5. Also, writing continues until memory position 19,999, 39,999, 59,999 or 79,999 is reached. Therefore, four WR 01 instructions, with appropriate addresses, are required to write all of an 80,000-character memory.

Refer to the section "Machine Components" for further details of the WR 01 instruction related to the DS.

Write 02 (R—SRC) *705 III*

Function. This operation sets the record counter (RC) of the DS to the digits contained in the tens, hundreds and thousands positions of the address. The RC indicator is turned on to put the next RD (00-01), WR (00-01), WTM, SKP or BSP instruction under control of the RC.

Address. The digits in the tens, hundreds and thousands positions of the address should represent the number of operations to be under the control of the RC.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 III	85	- - - - - - -

Description. A write instruction with 02 ASU coding places the tens, hundreds, and thousands digits of the instruction address into a record counter located in the DS. The RC indicator is turned on to put the next RD (00-01), WR (00-01), WTM, SKP, or BSP instruction under control of the RC. Each time the instruction following the WR 02 instruction is executed, the contents of the RC are reduced by one. The operation is repeated until the contents of the RC have been reduced to 000.

The following routine reads 125 records into memory, starting at location 05000:

OPERATION	ADDRESS	ADDRESS COMMENTS
SEL	00200	Selects DS 0 and its attached tape unit 0.
WR (02)	01250	Puts 125 into the record counter and turns the RC indicator on.
RD (00)	05000	Starts reading from the tape into memory location 05000.

During reading or writing under RC control, the following conditions may occur:

1. An end of reel marker can be sensed during writing.
2. A tape mark can be sensed during reading.
3. An invalid character turns on the PCT data check indicator.

In the case of the end of tape marker or invalid character, the multiple write operation is first completed before stopping. The sensing of a tape mark on a read operation, however, causes an immediate stop.

Since the RC control indicator may be on at the start of a program, housekeeping instructions to insure the resetting of this indicator should be a part of a program which uses the DS. The use of a RD 01 and a RWD instruction accomplishes the reset.

A WR 02 instruction applies only to the DS. If this instruction is given to a serial I-O device, a WR 01 operation results. A WR 02 instruction following a WR 02 instruction merely resets the RC to the number indicated in the second address.

Refer to the section "Machine Components" for further details of the WR 02 instruction related to the DS.

Write and Erase 00 (Z—WRE)

Function. This operation transmits a record from memory to a selected output unit and replaces the written memory field with blank characters (cannot be used with the DS).

Address. The address specifies the memory position from which writing is to begin.

Limiting Factors. Group mark in memory.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	10000 + 67*N*	To tape
	68 + 33.5*N*	To punch or printer
	8000 + 40*N*	To drum
705 III	Not to be used	From mem. to tape through DS
	49 + 33.5*N*	From mem. to punch or printer
	8000 + 40*N*	From mem. to drum

Description. The WRE 00 instruction transmits a record from memory to a selected output unit (except tape with the DS) and replaces the written memory field with blanks.

Information is written from memory, starting at the addressed memory position and continuing to successively higher memory positions until a group mark is sensed. The group mark will be replaced by a blank.

Sensing the group mark in memory stops the WRE operation. The group mark causes an inter-record gap to be automatically placed on tape; when drum is used, the group mark is converted to a drum mark at the end of the drum record.

No operation occurs if the instruction is addressed to a memory address of a group mark. The machine automatically proceeds to the next instruction.

The instruction is normally used for printing successive lines of different field arrangement or for group indication when detail printing. Fields may also be arranged on the tape for future printing and repetitive information can be eliminated in successive records.

NOTE: The WRE instruction used with the DS is converted into a WR instruction. The ACC or ASU coding determines the mode of execution of the resulting WR instruction.

Write and Erase 01 (Z—WRE)

Function. This operation transmits a record from memory to a selected output unit and replaces the written memory field with blank characters. (It cannot be used with the DS.)

Address. The address specifies the memory position from which writing is to begin.

Limiting Factors. The end of the memory block in which writing began.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	10000 + 67*N*	To tape
	68 + 33.5*N*	To punch or printer
	8000 + 40*N*	To drum
705 III	Not to be used	From mem. to tape through DS
	49 + 33.5*N*	From mem. to punch or printer
	8000 + 40*N*	From mem. to drum

Description. The WRE 01 instruction transmits a record from memory to a selected output unit (except tape unit with the DS) and replaces the written memory field with blank characters.

Information is written from memory, starting at the addressed memory position and continuing to successively higher memory positions until the end of the memory block is reached. The last character written and erased is located in a memory position 19,998, 39,998, 59,998 or 79,998. The last position of the memory block is not written or erased but causes the proper end-of-record indication (drum mark, inter-record gap) to be recorded. Group marks have no effect upon the execution of the instruction and are written out on the selected unit with all other characters.

To write and erase the last character of a memory block, a WR 01 instruction addressed to the last character of the memory block must be used.

Any ASU coding (01-15) is interpreted as 01.

NOTE: The WRE instruction used with the DS is converted into a WR instruction. The ASU coding determines the mode of execution of the resulting WR instruction.

Read While Writing (S—RWW)

Function. This operation conditions a selected input tape unit to retain its selected status, prepares the selected tape unit to read, and sets MAC II to the address part of the instruction.

Address. The address specifies the memory position into which the first character of the input record is to be placed.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	34	-------
705 III	22	-------

Description. A record or group of records may be read into memory from tape (except DS) and at the same time another record or group of records may be written from memory on a selected output unit (except DS and drum). This operation is performed by the use of four instructions: SEL, RWW, SEL, and WR or WRE.

The RWW instruction conditions a selected *input* tape unit (only tape can be used as input) to retain its selected status. It prepares the tape unit to read, but reading is not actually accomplished until a subsequent WR or WRE instruction is given to a selected output unit.

MAC II sets to the address part of the RWW instruction and indicates the memory location into which input data are to be read. When reading is completed, the setting of MAC II is equal to the address of one memory position beyond the address of the final character read.

The address part of the subsequent WR or WRE instruction specifies the memory location from which output data are to be written. It is controlled by MAC I. The read-while-writing operation cannot be performed using overlapping areas in memory.

NOTE: A WR instruction directed to a tape unit on the DS, following a RWW instruction, causes the DS and CPU to idle in automatic status. Any subsequent RD instruction operates in a normal manner.

Data Conversion Operations

The following examples are offered only as illustrations of basic principles of operation and are not intended to be primary applications of an installation. The end-of-file, checking and error correction procedures have been omitted.

CARD-TO-TAPE CONVERSION THROUGH MEMORY

Figure 12 illustrates the flow of information for processing data from cards to magnetic tape. The procedure, when first reduced to basic machine steps, requires five instructions described as follows:

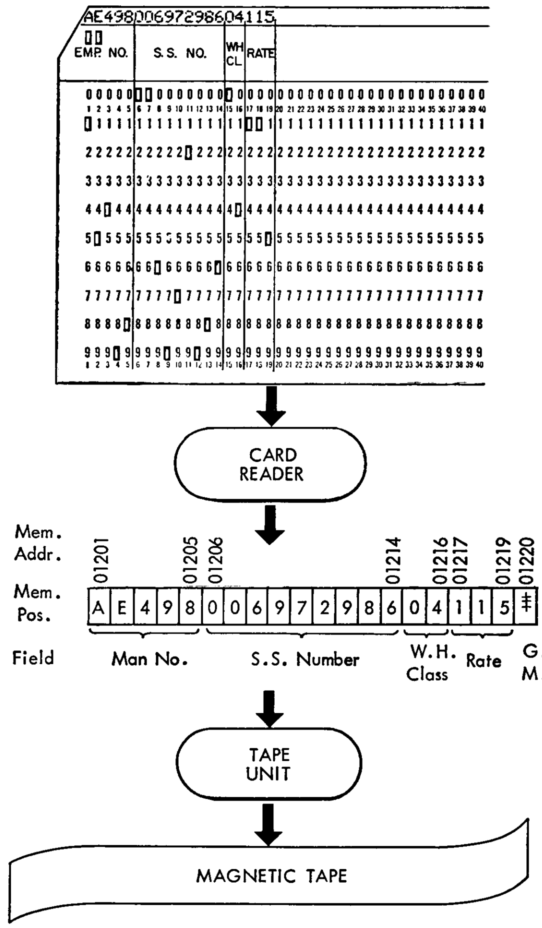


Figure 12. Data Conversion, Card to Tape

1. Select a card reader.
2. Read a card record from this selected unit into core memory.
3. Select a tape unit.
4. Write the record from memory on tape, using the selected unit.
5. Transfer to the first operation and repeat the program until all cards are processed.

The programmed instructions select two machine units, a card reader and a tape unit. Two operations are performed, reading the card record and writing on tape. The transfer operation is inserted to "loop" the program until all card records are processed.

Each of the above steps instructs the machine. As previously stated, an instruction consists of both an operation part and an address part. The select operation (step 1) must, therefore, specify the address of the unit to be used (in this case, card reader 00100).

Each character of the card record wired on the control panel to record storage must occupy one position of memory. The address part of the instruction specifies the memory position that stores the first character of the record. In step 2, this is address 01201.

Instruction		Description
Operation	Address	
SEL	00100	Card reader
RD	01201	Memory address
SEL	00200	Tape unit
WR	01201	Memory address
TR		Repeat program

Figure 13. Program, Card to Tape

The program, with abbreviated operations and proper addresses, is shown in Figure 13.

To be executed by the machine, the program must be stored in memory. The mnemonic instruction abbreviations are used merely for convenience in writing the program; *they are not stored in memory.* The operation code of one character and its related address make up five characters for each instruction. When stored in memory, beginning at address 00000, the program would appear as in Figure 14. Here, the address of instruction 1 is 00004, instruction 2 is 00009, and so on.

The transfer instruction may now be given the address of the memory location of the first instruction. The machine transfers to memory location 00004 and executes the instruction it finds there.

Tape-to-Card Conversion through Memory

Figure 15 illustrates the flow of information in processing data from tape to cards. The procedure, when first reduced to basic machine steps, requires five instructions described as follows:

1. Select a tape unit.
2. Read a record from the tape into memory.
3. Select a card punch.
4. Punch the record from memory into a card.
5. Transfer to repeat the instructions until all records are processed.

For this problem, assume that a group mark has been previously placed in memory location 17028 ($\overset{01}{7}$ 0 2 8).

Figure 16 shows the program steps. Memory addresses are assigned to each instruction.

00004. Select tape unit 00200.

00009. Read a unit record from the tape into memory, beginning at address 17001 ($\overset{01}{7}$ 0 0 1). Continue

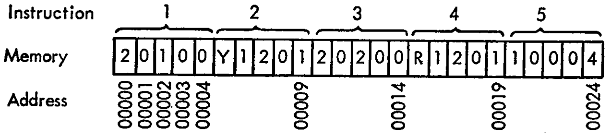


Figure 14. Program, Card to Tape in Memory

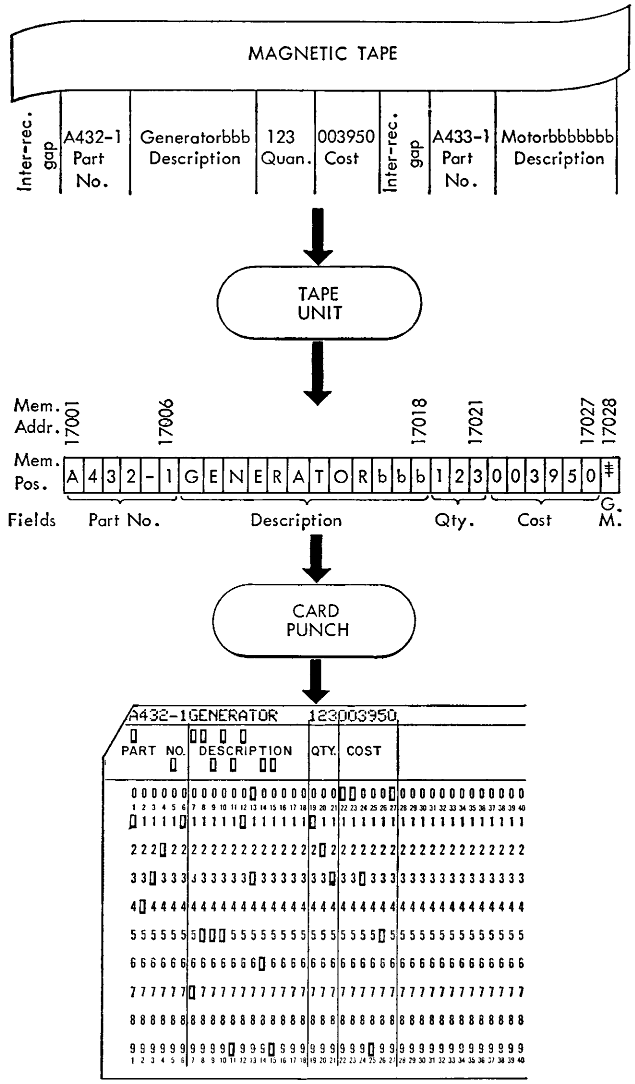


Figure 15. Data Conversion, Tape to Card

to read the tape into successively higher memory positions until the inter-record gap is reached.

00014. Select card punch 00300.

00019. Write the record stored in memory beginning with address 17001 and continue to write out successively higher-order memory positions until the group mark is reached.

00024. Transfer to the instruction located at memory address 00004 and repeat the program until all records are processed.

Instruction Location	Instruction		Address
	Operation Abbrev.	Operation Part	
00004	SEL	2	00200
00009	RD	Y	17001
00014	SEL	2	00300
00019	WR	R	17001
00024	TR	1	00004

Figure 16. Program, Tape to Card

Tape-to-Card and Printer through Memory

Information from tape can be punched in cards and also written on the printer during the same procedure by the program shown in Figure 17, with two additional instructions: (1) a select instruction to select the printer, and (2) a write instruction to transmit the record from memory to the printer.

Instruction Location	Instruction		Address
	Operation Abbrev.	Operation Part	
00004	SEL	2	00200
00009	RD	Y	17001
00014	SEL	2	00300
00019	WR	R	17001
00024	SEL	2	00400
00029	WR	R	17001
00034	TR	1	00004

Figure 17. Program, Tape to Card and Printer

Data Transmission Operations

Various methods are used to move data in the 705 from memory to memory, from memory to the storage units, and from storage units to memory. Data to be transmitted from one storage unit to another, however, must pass through memory. The transmission may affect an entire record in one operation or may specify particular fields, groups of fields, or individual characters.

Two or more records can be combined within the machine, either with or without calculation, to form any desired arrangement of output information for punching in cards, printing on report forms, or writing on tape. Conversely, single records can be split to form several records in any arrangement and transcribed by one or more of the output units. Any or all of the various input-output devices may be used during a single procedure to handle the record forms: cards, tape, or printed reports.

Memory-to-Memory Transmission

Receive (U—RCV)

Function. This operation sets the memory address counter (MAC II) to the address part of the instruction.

Address. The address designates the memory location at which information is to be placed by a transmit or send instruction.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	34	-------
705 III	22	--------

Description. The receive instruction is used to designate the memory location at which information is to be placed by a subsequent transmit or send instruction.

When blocks of five characters are to be received (TMT 00 and SND), the address part specifies the memory location of the fifth character to be received and must always have a units digit of 4 or 9.

When individual characters or fields are received one character at a time (TMT 01), the address specifies the memory location of the first character to be received. The address may then specify any memory address.

Any ASU coding 01-15 is interpreted as 00. The mnemonic code RCVS (RCV 01) is used to distinguish receive instructions that are used with the TMT 01 instruction from those used with the TMT 00 and SND instructions.

Transmit 00 (9—TMT)

Function. Data are transmitted, in blocks of five characters at a time, from a memory location specified by the address part of the instruction to a memory location specified by the address part of a previous RCV instruction.

Address. The address specifies the location in memory from which the record is to be transmitted. The addressed memory position must be one with a units digit of 4 or 9 and represents the location of the fifth character of the first block of characters to be transmitted.

Limiting Factors. A record mark in a memory location with a units digit of 4 or 9.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$17 + \left(\frac{N_m}{5} \times 18\right)$	--------
705 III	$13 + \left(\frac{N_m}{5} \times 18\right)$	---- --

Description. The transmit instruction specifies the location in memory from which the record is to be transmitted to the receiving area.

Blocks of five characters are transmitted with the address part specifying the memory location of the fifth character. This address part must have a units digit of 4 or 9.

Transmission is limited by the record mark which would normally be the last character of the record block. The address of this limiting mark must also have a units digit of 4 or 9. The total number of characters transmitted, including the record mark, must be evenly divisible by five. A record mark in other positions of memory is transmitted like any other character and does not limit the transmission.

The address of the transmit instruction is placed in the memory address counter I (MAC I). This counter advances five positions after each block of five characters has been transmitted. The original record in memory is not affected by the transmit instruction.

If a TMT instruction is used without a previous RCV instruction, the operation takes place using whatever

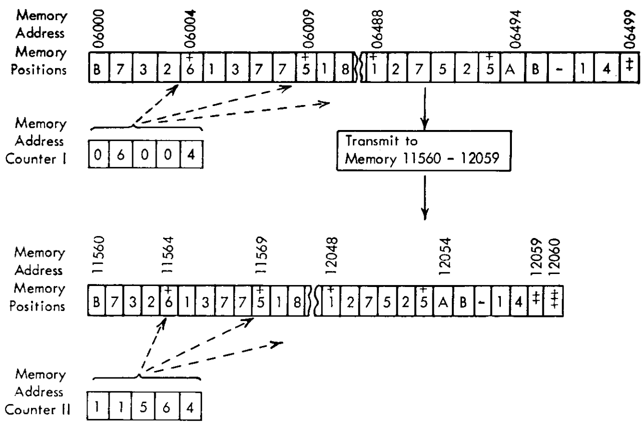


Figure 18. Memory-to-Memory Transmission (Five Character)

memory address is indicated by MAC II for the receiving area of memory.

Program, Transmitting Five Characters

The record shown in Figure 18 is read from tape unit 00200 into memory locations 06000-06499, inclusive. The record is to be transmitted to a work area starting with memory location 11560. After transmission and perhaps some processing, the record is written on output tape unit 00201.

The last character of the record is a *record mark.* This special character may be a part of the record on tape, or it may be placed in the proper memory position at the end of the record by instructions in the program. The character may also originate from the control panel of the card reader.

The record mark serves two purposes:

1. When placed in the *proper position* at the end of the record, it stops five-character transmission of the record from one memory location to another in much the same manner that a group mark limits a writing operation.
2. It may be written between records on tapes when records are grouped into blocks. In this case, each block is separated by the inter-record gap; records within the block are separated by record marks. This procedure is more fully explained in the section on grouping records.

Figure 19 is the program for transmitting five-character groups.

00004. Select tape unit 00200.

00009. Read the record into memory beginning at the address 06000.

00014. Set MAC II to address 11564 to prepare to receive the first five characters of the record.

00019. Set memory address counter I to address 06004, to transmit the first five characters to the location specified by the receive instruction. Because accumulator 00 is specified, transmission continues in five-character groups until the record mark is reached at address 06499. Both memory address counters advance five positions at a time.

INSTR. LOCATION	INSTRUCTION OPER.	ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00004	SEL	00200					
00009	RD	06000					
00014	RCV	11564					
00019	TMT	06004	00				
00024	RAD		04				
00494	SEL	00201					
00499	WR	11560	00				
00504	TR	00004					

Figure 19. Program, Memory-to-Memory Transmission (Five Character)

00024. Continue the main body of the program.

00494. Select tape unit 00201.

00499. Write, beginning at address 11560.

00504. Transfer to repeat the program.

Transmit 01-15 (9—TMTS)

Function. Data are transmitted, one character at a time, from a memory location specified by the address part of the instruction to a memory location specified by the address part of a previous RCV instruction.

Address. The address specifies the location in memory from which transmission is to begin.

Limiting Factors. Storage mark in the ASU indicated by the ASU coding of the address part of the instruction.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$34 + 18N_S$	-------
705 III	$31 + 18N_S$	-------

Description. Individual characters are transmitted, one character at a time, with the address part of the instruction specifying the memory location of the first character to be transmitted. The address may specify any memory address. The transmit instruction must be preceded by a RCV instruction to designate the receiving location.

The storage mark in the selected ASU limits the number of characters transmitted. During transmission, the storage unit is checked for a storage mark beginning at the right-hand position. This check is made for each character transmitted until the storage mark is sensed. Sensing the storage mark stops the operation.

Any auxiliary storage unit may be used. The storage mark is usually adjusted by a set left instruction

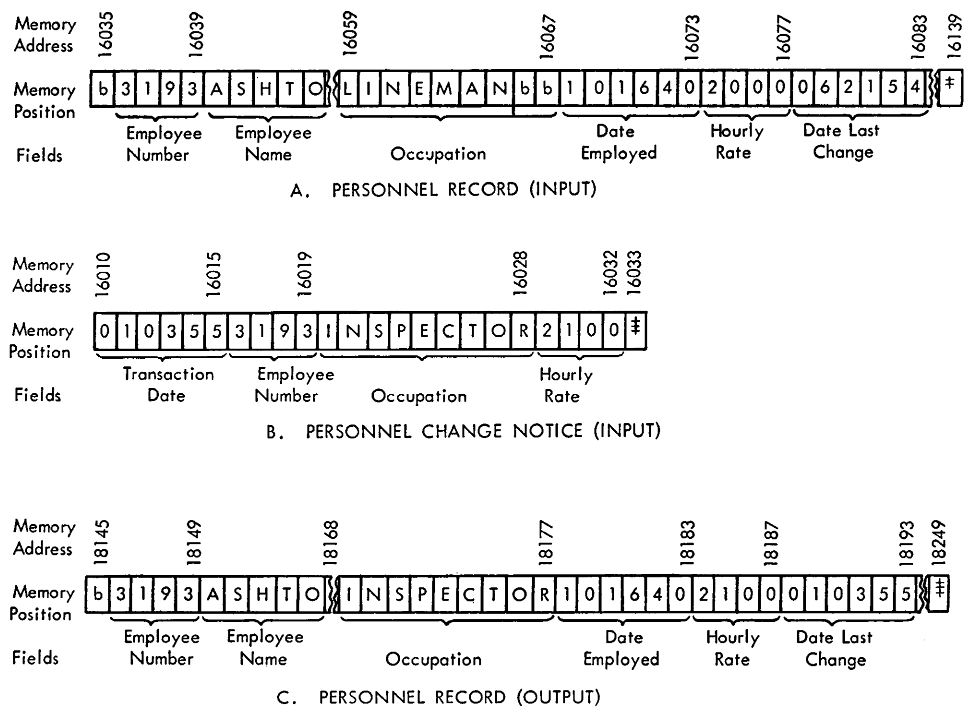


Figure 20. Memory-to-Memory Transmission (Single Character)

or properly positioned as a result of calculation. The transmit instruction must be preceded by a receive instruction to designate the receiving location.

The address of the transmit instruction is placed in memory address counter I (MAC I). This counter advances one position after each character has been transmitted. The original record in memory and the field in storage are not affected by the transmit operation.

If a TMT instruction is used without a previous RCV instruction, the operation takes place using whatever memory address is indicated by MAC II for the receiving area of memory.

Program, Transmitting Single Characters

Two records are read into memory as shown in Figure 20. Record A is a portion of a master payroll record on tape. Record B is an employee rate and occupation change notice on IBM cards.

A record mark is included as the last character of record A. A group mark is emitted from the card reader as the last character of record B. Record A is transmitted to work area C in blocks of five characters. Since the total number of characters in record A is not divisible by five, one blank position of memory at address 16035 is included in the transmission. The rate, occupation name, and date are placed in the work area by single-character transmission. Auxiliary storage units are adjusted to control the transmission as "housekeeping" program steps before the records are read into memory.

The matching of input tapes and card records by employee number is omitted from the problem.

Figure 21 is the program for single-character transmission from memory to memory.

00004, 00009, 00014, 00019. The storage marks in auxiliary storage units 01, 02, 03, and 04 are adjusted to control transmission of the group marks and the rate, occupation name, and date fields to

INSTR. LOCATION	INSTRUCTION OPER.	ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN
00004	SET	00001	01		
00009	SET	00004	02		
00014	SET	00006	03		
00019	SET	00009	04		
00104	SEL	00200			
00109	RD	16036			
00114	SEL	00100			
00119	RD	16010			
00124	RCV	18149			
00129	TMT	16039	00		
00134	RCV	18169			
00139	TMT	16020	04		
00144	RCV	18184			
00149	TMT	16029	02		
00154	TMT	16010	03		
00159	RCV	18249			
00164	TMT	16033	01		
00169	SEL	00201			
00174	WR	18146	00		
00179	TR	00104			

Figure 21. Program, Memory-to-Memory Transmission (Single Character)

record C. These preliminary steps are referred to as "housekeeping" instructions and need to be executed only once for all records to be processed. Once the storage marks are placed, they remain in their respective locations throughout the entire procedure, unless their position is changed by other instructions in the program.

00104. Select tape unit 00200.

00109. Read the tape record into memory beginning at address 16036. Location 16035 is assumed to be blank.

00114. Select card reader 00100.

00119. Read card into memory, starting at 16010.

00124. Set MAC II to memory address 18149 to receive the first five characters of record A.

00129. Set MAC I to memory address 16039 to transmit the first five characters of record A to the location specified by the receive instruction. Accumulator 00 is specified for five-character transmission. The transmission continues, in blocks of five characters, until the record mark is sensed.

00134. Set MAC II to 18169 to receive the first character of occupation name from the card record.

00139. Set MAC I to 16020 to transmit the first character of occupation name. Auxiliary storage unit 04 is specified to limit single-character transmission to the nine characters of the field.

00144. Set MAC II to 18184 to receive the first character of hourly rate from the card record.

00149. Set MAC I to 16029 to transmit hourly rate. Auxiliary storage unit 02 is specified to control the transmission of the four-character field.

00154. Set MAC I to 16010 to transmit transaction date from the card. MAC II has been properly positioned at 18188 during transmission of hourly rate. The counter remains set to this address until another receive instruction is given or until it is stepped to a new address by a transmit instruction, by a simultaneous read-write instruction, or by the 705 III SND or BLM instruction. The transmission of the six-character transaction date field is controlled by specifying auxiliary storage unit 03.

00159. Set MAC II to 18249 to receive a group mark.

00164. Transmit the group mark from the card record at memory address 16033. The group mark replaces the record mark transmitted by the instruction at memory location 00129.

00169. Select tape unit 00201.

00174. Write the record on tape, beginning at memory address 18146. The group mark placed at location 18249 stops the writing operation.

00179. Transfer to repeat the program from the instruction located in memory position 00104.

Send (/—SND) *(705 III)*

Function. Data are transmitted, in blocks of five characters at a time, from a memory location specified by the address part of the instruction to a memory location specified by the address part of a previous RCV instruction.

Address. The address specifies the location in memory from which the record is to be transmitted. The addressed memory position must be one with a units digit of 4 or 9 and represents the location of the fifth character of the first block of characters to be transmitted.

Limiting Factors. Storage mark in a designated ACC or ASU storage unit.

Timing

MODEL	TIME (μs)	REMARKS
705 III	$31 + \left(\frac{N}{5} \times 18\right)$	-------

Description. The send instruction causes a high-speed movement of data, in groups of five characters, from one location in memory to another. The address part of the instruction specifies the location in memory from which the record is to be moved. The SND must be preceded by a receive (RCV) instruction to set MAC II at the address of the receiving location. Both the RCV and SND instructions must designate memory locations with addresses ending in either 4 or 9. This address is the fifth character of the record or group of data to be sent.

The SND instruction must also be preceded at some time by an instruction setting the accumulator or auxiliary storage unit at a length to properly control the amount of information that is to be moved. This preset storage unit is then designated by the send instruction (ACC or ASU coding) and five characters are sent for each position to the right of the storage mark; for example, an ASU set at four places causes four groups of five characters (20 characters) to be moved; an ASU set at 20 causes 20 groups of five characters (100) to be moved. The contents of the designated storage unit are unaffected by the execution of the SND instruction.

If a SND instruction is used without a previous RCV instruction, the operation takes place using whatever memory address is indicated by MAC II for the receiving area of memory.

If the SND instruction is preceded by a RWW instruction, the SND instruction is not executed and a check memory operation is performed. The RWW-SND sequence of instructions performs a high-speed redundancy check (groups of five characters) on the contents of memory, starting at the address specified by the SND instruction and continuing to the end of the memory block being used. Any redundant characters detected in the memory block cause the 00901 check indicator to be turned on, but do not cause a stop regardless of the setting of the 00901 switch.

If the 705 III is used with an IBM 754 Tape Control, attached tape units should not be in a select status.

This operation is useful to determine whether a redundancy in memory is the cause of persistent PCT data check error indications during a WR instruction. Also, it can be used to check the validity of memory before taking a check point.

Blank Memory 00 ($—BLM) (705 III)

Function. This operation places blank characters, five at a time, in memory.

Address. The numerical value of the address indicates the number of five-character groups to be used for blanking.

Limiting Factors. Blanking of as many five-character groups as indicated by the numerical value of the address.

Timing

MODEL	TIME (μs)	REMARKS
705 III	$31 + \left(\frac{N_m}{5} \times 9\right)$	-------

Description. The number of five-character groups that are blanked is indicated by the numerical value of the instruction address of the BLM instruction. A preceding RCV instruction specifies the address of the five-character group where blanking will begin. With the high-speed mode, blanking always begins in a memory address units position of 0 or 5, meaning that the RCV instruction address must refer to an address ending in 4 or 9, respectively. Example:

RCV	44004
BLM 00	00015

These instructions cause the high-speed erasing or blanking of 15 groups of five characters (total of 75 characters), beginning with the character at location 44000 and ending, therefore, in location 44074.

If a BLM instruction is used without a previous RCV instruction, the operation takes place using whatever memory address is indicated by MAC II for the blanking area of memory.

Blank Memory 01 ($—BLMS)

Function. This operation places blank characters, one at a time, in memory.

Address. The numerical value of the address indicates the number of single characters to be blanked.

Limiting Factors. Blanking of as many characters as indicated by the numerical value of the address part of the instruction.

Timing

MODEL	TIME (μs)	REMARKS
705 III	$31 + 9N_m$	-------

Description. The number of individual characters that is to be blanked is indicated by the numerical value of the instruction address. A preceding RCV instruction specifies the address of the character where the blanking will begin. This may be any address in memory. Example:

RCV	44002
BLM 01	00015

These instructions cause the slow-speed erasing or blanking of a total of 15 characters, starting with location 44002 and ending in location 44016.

If a BLM instruction is used without a previous RCV instruction, the operation takes place using whatever memory address is indicated by MAC II for the blanking area of memory.

Memory-to-Storage Transmission

Reset and Add (H—RAD)

Function. This operation enters a numerical field from memory into ACC or ASU storage.

Address. The address signifies the units position of the memory field and the storage unit to be used.

Limiting Factors. First non-numerical character to the left of the addressed character.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$34 + 17N_m$	-------
705 III	$40 + 9.3N_m$	-------

Description. The reset-and-add instruction enters a numerical field from memory into accumulator or auxiliary storage. The address part of the instruction specifies the location of the field and the storage unit to be used.

Digits are entered into storage, starting with the specified right-hand digit of the memory field and continuing from *right* to *left* until a non-numerical character is sensed. This non-numerical character is not entered into storage. The zoning of the addressed digit of the memory field is not entered into storage.

The accumulator or auxiliary storage sign is set to plus when the addressed memory character has plus zoning and is set to minus when the character has minus zoning. If the addressed character has neither plus nor minus zoning, an error is indicated and the sign of the field is interpreted as plus. Refer to "Sign Check Indicator." The sign is always set to plus when the result in storage is zero.

The left-hand limit of the storage field is automatically set by a storage mark (a) stored next to the last digit entered from memory.

When the memory field exceeds the capacity of an auxiliary storage unit, the field automatically extends into the adjacent unit with proper positioning of the storage mark in this unit.

The reset and add instruction does not affect the field in memory.

Special case: When the numerical portion of the character addressed by the instruction is 0000 (& — blank), the character is entered into storage as a zero (0 00 1010). The accumulator or auxiliary storage sign and the sign check indicator are set according to the rules stated above.

EXAMPLES, RESET AND ADD

	STORAGE BEFORE		STORAGE AFTER		SIGN CHECK
MEMORY	STORAGE	SIGN	STORAGE	SIGN	IND.
$\overset{+}{4}45\bar{6}$	a23456	+	a456	—	
$\bar{4}45\overset{+}{6}$	a23456	—	a456	+	
$\bar{4}b5\overset{+}{6}$	a1234	+	a56	+	
$\overset{+}{4}456$	a1234	—	a456	+	On
$E45\bar{6}$	a2345	+	a456	—	
$T00\bar{0}$	a1111	+	a000	+	
4bbb	a111	+	a0	+	On
ST	a4567	+	a3	+	On

Reset and Subtract (Q—RSU)

Function. This operation enters a numerical field from memory into ACC or ASU storage.

Address. The address signifies the units position of the memory field and the storage unit to be used.

Limiting Factors. First non-numerical character to the left of the addressed character.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$34 + 17N_m$	-------
705 III	$40 + 9.3N_m$	-------

Description. The reset-and-subtract instruction enters a numerical field from memory into accumulator or auxiliary storage. The address part of the instruction specifies the field location and the storage unit used.

Digits are entered into storage, starting with the specified right-hand digit of the memory field and continuing from *right* to *left* until a non-numerical character is sensed. This non-numerical character is not entered into storage.

The zoning of the addressed digit of the memory field is not entered into storage.

The accumulator or auxiliary storage sign is set to minus when the addressed memory character has plus zoning and is set to plus when the character has minus zoning. If the addressed character has neither plus nor minus zoning, an error is indicated and the sign of the memory field is interpreted as plus, thus setting the storage sign minus. The sign is always set to plus when the result in storage is zero.

The left-hand limit of the storage field is automatically set by a storage mark (a) stored next to the last digit entered from memory.

When the memory field exceeds the capacity of an auxiliary storage unit, the field automatically extends into the adjacent unit with proper positioning of the storage mark in this unit.

The field in memory is not affected by the reset and subtract instruction.

Special case: When the numerical portion of the character addressed by the instruction is 0000 (& — blank), the character is entered into storage as a zero (0 00 1010). The accumulator and auxiliary storage sign and the sign check indicator are set according to the rules stated above.

EXAMPLES, RESET AND SUBTRACT

	STORAGE BEFORE		STORAGE AFTER		SIGN CHECK
MEMORY	STORAGE	SIGN	STORAGE	SIGN	IND.
$\overset{+}{4}45\bar{6}$	a11111	+	a456	+	
$\overset{+}{4}45\overset{+}{6}$	a1	—	a456	—	
$4b5\overset{+}{6}$	a11	+	a56	—	
T0000	a1	—	a0000	+	See note
4bbb	a111	—	a0	+	On
ST	a11	—	a3	—	On
A1	a1	—	a1	—	On

NOTE: If the resulting storage field is zero (regardless of memory sign), the sign is set to PLUS with sign check on.

Load Storage (8—LOD)

Function. This operation moves alphabetic or numerical information from memory to ACC or ASU storage.

Address. The address specifies the location of the right-hand character of the field in memory and the storage unit to be used.

Limiting Factors. Storage mark in the designated storage unit.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$34 + 17N_S$	-------
705 III	$31 + 9.3N_S$	-------

Description. The load instruction permits single characters, series of characters, or fields to be entered into accumulator or auxiliary storage from memory. The number of characters or length of the field loaded into storage is determined by the position of the storage mark relative to the starting point counter. The load instruction may be preceded by a set left instruction to adjust storage to the size of the field to be loaded.

The address part of the instruction specifies the location of the right-hand character of the field in memory and the storage unit to be used. Characters are loaded from right to left from memory until the specified storage space is filled.

The accumulator or auxiliary storage sign is always set to plus by a load instruction. The field, character, or series of characters in memory are not affected by load storage.

EXAMPLES, LOAD

	STORAGE BEFORE		STORAGE AFTER	
MEMORY	STORAGE	SIGN	STORAGE	SIGN
ABCbbb1234$\bar{5}$	a7310	+	a234$\bar{5}$	+
ABCbbb1234$\bar{5}$	a12345678901	—	aABCbbb1234$\bar{5}$	+
ABCbbb1234$\bar{5}$	a00	+	a4$\bar{5}$	+
ABCbbb1234$\overset{+}{5}$	a00000012	+	abbb1234$\overset{+}{5}$	+

Load Address (#—LDA) *(705 III)*

Function. This operation places the address portion of an instruction in memory into ACC or ASU storage as a five-digit field.

Address. The address specifies the memory address of the instruction whose address portion is to be moved, and the storage unit to be used.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 III	76	-------

Description. The load address instruction is used to load the address portion (four characters) of an instruction in memory into the accumulator or auxiliary storage as a five-digit field. For example, assume that in location 00009 of memory there is the instruction RD $\overset{01}{3}$ 4 4 $\overset{10}{2}$. By giving the instruction LDA 00009 (04), the four address positions of the RD instruction (00009, 00008, 00007, and 00006) are loaded into ASU 04 as 53442. In this completely numerical form, the instruction address is much easier to manipulate.

To execute this instruction, the machine performs the following:

1. Only the numerical portion of the four addressed characters in memory is loaded into storage; forming the four low-order digits there.
2. The fifth (high-order) digit in storage is formed from the zone bits over the units and thousands positions of the address in memory. Thus:
   a. The A bit of the thousands character in memory becomes the 1 bit of the fifth storage digit.
   b. The B bit of the thousands character in memory becomes the 2 bit of the fifth storage digit.
   c. The B bit of the units character in memory becomes the 4 bit of the fifth storage digit.

All other zoning in the memory field will be ignored. This includes any ASU zoning over the tens or hundreds positions and the A bit over the units position for indirect addressing.

A storage mark will be placed to the left of the high-order digit in storage. Therefore, the LDA instruction need not be preceded by a SET instruction. The following illustrates the effect of the load address instruction:

MEMORY	STORAGE BEFORE	STORAGE AFTER
$\overset{01}{3}$ 4 4 $\overset{10}{2}$	a	a53442
$\overset{01}{3}$ $\overset{10}{4}$ $\overset{11}{4}$ $\overset{11}{2}$	a	a53442
$\overset{01}{1}$ 2 2 8	a6722481	a11228
$\overset{11}{4}$ 8 7 $\overset{01}{7}$	a25	a34877
7 3 0 9	a	a07309

Sign (T—SGN)

Function. This operation removes zone bits from a memory character and places them in ACC or ASU storage as an ampersand or dash.

Address. The address specifies the character to be processed and the storage unit to be used.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	68	-------
705 III	49	-------

Description. The sign instruction is used to remove any zone from a memory character and place it in accumulator or auxiliary storage as an ampersand or dash. The character affected and the storage unit used are specified by the address part of the instruction.

When the zoning of the addressed character in memory is minus, a dash (minus zone) is placed in the storage unit and the storage sign is set to minus.

When the zoning of the addressed character in memory is other than minus, an ampersand (plus zone) is placed in storage and the storage field length is set to one position. The storage sign is set to plus.

The addressed character remains in memory with 00 zoning unless that character is an ampersand (&), a dash (—) or a blank. In these cases, the character remaining in memory is a blank.

The sign placed in accumulator or auxiliary storage as an ampersand (plus) or a dash (minus) may be given to any character in memory that is not already zoned. The add-to-memory is used for this purpose and the character to be signed is specified by the address part of the add-to-memory instruction.

EXAMPLE, SIGN

BEFORE			AFTER		
ACC. OR AUX. STORAGE	SIGN	MEMORY CHARACTER	ACC. OR AUX. STORAGE	SIGN	MEMORY CHARACTER
a123456	+	B	a&	+	2
aEDPM	+	R	a—	—	9
a&	+	$\overline{4}$	a—	—	4
a16AB	—	$\overset{+}{4}$	a&	+	4
a16AB	+	&	a&	+	b
a16AB	+	—	a—	—	b
a16AB	+	b	a&	+	b

Storage-to-Memory Transmission

Store (F—ST)

Function. A numerical field in ACC or ASU is placed in memory.

Address. The address specifies the memory position (right-hand) where the field from storage is to be placed and the storage unit to be used.

Limiting Factors. Storage marked in the designated storage unit.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$34 + 17N_S$	-------
705 III	$40 + 10.8N_S$	-------

Description. A numerical field in the accumulator or auxiliary storage is placed in memory by a store instruction.

The right-hand digit of the storage field is stored at the specified memory address. The remaining digits to the left are stored in successively lower memory positions until the storage mark is sensed. All digits from the position of the starting point counter to the storage mark are stored.

The sign of the accumulator or auxiliary storage is converted to plus or minus zoning and is placed over the units position of the field in memory.

When the character in the next lower memory position is numerical, this character is signed plus to define properly the stored field. A non-numerical character is not affected. The store instruction does not affect the field in accumulator or auxiliary storage.

EXAMPLES, STORE

ACC OR AUX STORAGE	STORAGE SIGN	MEMORY BEFORE	MEMORY AFTER
a48	—	$\overset{+}{1}72\overset{+}{9}$	$\overset{+}{1}\overset{+}{7}4\overline{8}$
a67	+	$\overset{+}{1}72\overset{+}{9}$	$\overset{+}{1}\overset{+}{7}6\overset{+}{7}$
a592	—	$\overset{+}{3}41\overset{+}{5}$	$\overset{+}{3}59\overline{2}$
a738	+	$\overset{+}{3}41\overline{5}$	$\overset{+}{3}73\overset{+}{8}$
a7468	—	FRAME	F746$\overline{8}$
a35	—	F16	F3$\overline{5}$

If the store instruction is used on non-numerical fields in storage, the zone information is removed and invalid characters can be placed in memory. Any character, other than the first character of the storage field, with 01 or 10 zoning produces an invalid character in memory. Characters with 11 zoning do not produce invalid characters in memory.

A 00901 check indication on data placed in memory using the 705 I or II does not occur until the data are used in subsequent operations. A redundant character placed in memory using the 705 III results in an immediate 00901 check indication.

Store for Print (5—SPR)

Function. A numerical or alphabetic field in ACC or ASU storage is placed in memory with discriminative action by the instruction.

Address. The address specifies the memory position (right-hand) where the field from storage is to be placed and the storage unit to be used.

Limiting Factors. Storage mark in the designated storage unit.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$51 + 17N_s$	------
705 III	$40 + 10.8N_s + 9N_p$	------

Description. The store-for-print instruction normally is used to transfer a numerical field from the accumulator or auxiliary storage to memory. However, this instruction can also be used to store alphabetic fields from accumulator or auxiliary storage to memory.

When the sign of the storage unit is plus, a blank is stored in the memory position specified by the address part of the instruction.

When the sign of the storage unit is minus, a dash is stored in the memory position specified by the address part of the instruction.

The numerical storage field is stored in the memory positions directly to the left of the sign position. The storage mark determines the left limit of the field to be stored.

When periods or commas are encountered in memory, these memory positions are skipped and the digits are stored in successively lower address positions. The characters b & — are stored as ‡ $\overset{+}{0}$ $\bar{0}$ respectively.

After the storage field has been placed in memory, the memory field is inspected from left to right, to the first significant character. Insignificant zeros, characters with a zero numerical part, and commas are replaced by blanks. Zeros to the right of a decimal point are not replaced.

The field in storage remains unchanged by this operation.

EXAMPLES, STORE FOR PRINT

STORAGE	SIGN	MEMORY BEFORE	MEMORY AFTER
a007638	+	$ 21 35.146	$ bb76.38b
a0071834	—	bb,bbb.bbb	bbb718.34—
a00000000	+	bbb,bbb.bbb	bbbbbbb.00b
a0473829	—	bb,bbb.bbb	b4,738.29—
aABCDE	+	bb.bbbb	AB.CDEb
a$\overset{+}{0}$461$\overset{+}{2}$	+	bb.bbbb	b4.61$\overset{+}{2}$b
aABCbD	+	bb.bbbb	AB.C‡Db

The SPR instruction is commonly used to place numerical data from storage into special memory areas for printing. These memory areas normally contain a prescribed number of positions, as well as predetermined periods and commas, for report printing. Under these circumstances, this instruction should be used with fields of known length. For example, to store in a ten-position (plus punctuation) memory field, the storage unit should contain ten digits. If it contains less, the resulting memory field may include remaining high-order digits from a previous field.

A 00901 check indication on data placed in memory using the 705 I or II does not occur until the data are used in subsequent operations. A redundant character placed in memory using the 705 II results in an immediate 00901 check indication.

Unload Storage (7—UNL)

Function. Numerical or alphabetic information in ACC or ASU is placed in memory.

Address. The address specifies the memory position (right-hand) where the field from storage is to be placed and the storage unit to be used.

Limiting Factors. Storage mark in the designated storage unit.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$34 + 17N_S$	-------
705 III	$31 + 10.8N_S$	-------

Description. The unload instruction is used to place the contents of accumulator or auxiliary storage in memory. The length of the field unloaded into memory is equal to the number of positions in the storage unit.

The right-hand character of the designated storage unit is unloaded into the memory position specified by the address part of the instruction. Remaining characters in storage are entered successively into memory from right to left until a storage mark is sensed.

The accumulator or auxiliary storage sign has no effect upon the data placed in memory. The contents of accumulator or auxiliary storage are not affected by unload storage.

EXAMPLES, UNLOAD

STORAGE	SIGN	MEMORY BEFORE	MEMORY AFTER
a374$\overset{+}{8}$	—	B0229	B374$\overset{+}{8}$
a450	+	1576	1450
aAB1$\bar{2}$	+	134CD	1AB1$\bar{2}$
ab$bb	+	0000	b$bb

Unload Address (—ULA)* *(705 III)*

Function. A five-digit field in storage is unloaded into memory as a four-character address field.

Address. The address specifies the instruction location address into which the storage field is to be placed.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 III	85	-------

Description. The unload address instruction causes a five-digit field in storage to be unloaded into memory as a four-character address field. For example, if ASU 04 contains the address a53442, the instruction ULA 00009 (04) causes the contents of ASU 04 to be unloaded into memory positions 00006, 00007, 00008 and 00009 as $\overset{01}{3}\ 4\ 4\ \overset{10}{2}$.

The numerical portion of the high-order character in storage is placed in memory as follows:

1. The 1 bit becomes the A bit of the thousands order character in the memory field.
2. The 2 bit becomes the B bit of the thousands order character in the memory field.
3. The 4 bit becomes the B bit of the units order character in the memory field.
4. The 8 bit, if any, is ignored.

If the storage field contain zones, they are ignored. The A bit of the units order, and the tens and hundreds order zone bits of the memory field are not affected by this instruction.

If, at the time of executing the ULA instruction, the field in storage is less than five digits, it is treated as five digits with zeros to the left of the significant digits. If the field in storage is greater than five digits, only the low-order five are recognized.

EXAMPLES, UNLOAD ADDRESS

STORAGE	MEMORY BEFORE	MEMORY AFTER
a 5 3 4 4 2	$\overset{10}{5}\ 6\ 4\ 7$	$\overset{01}{3}\ 4\ 4\ \overset{10}{2}$
a $\overset{11}{0}\ \overset{11}{0}\ \overset{11}{0}\ \overset{11}{2}\ \overset{11}{4}$	$0\ 0\ 1\ \overset{01}{4}$	$0\ 0\ 2\ \overset{01}{4}$
a $7\ 5\ \overset{01}{5}\ \overset{01}{5}\ 5$	$2\ \overset{10}{2}\ \overset{01}{2}\ 2$	$\overset{11}{5}\ \overset{10}{5}\ \overset{01}{5}\ \overset{10}{5}$
a 2 5	3 7 3 7	0 0 2 5
a 1 7 6 2 3 0 7	$\overset{01}{4}\ \overset{01}{3}\ \overset{11}{2}\ \overset{01}{1}$	$\overset{10}{2}\ \overset{01}{3}\ \overset{11}{0}\ \overset{11}{7}$
a 1 7 9 2 3 0 7	4 3 2 1	$\overset{01}{2}\ 3\ 0\ 7$

Arithmetic and Shift Operations

Arithmetic Operations

The 705 adds, subtracts, multiplies, and divides when given arithmetic instructions. These instructions can be applied to data stored in accumulator storage, auxiliary storage (except multiply and divide) or in memory. They are normally applied to specific numerical factors or fields, such as factors developed during calculation or fields selected from records.

To select a field from memory to be acted upon by an arithmetic instruction, the field is always addressed by the memory location of its units digit. The remaining digits of the field are automatically read from right to left until a non-numerical character is reached. All characters, including blanks, are considered non-numerical except the digits 0-9. Thus, a numerical field in memory is defined as beginning with the address of its units digit and extending to, but not including, the next left non-numerical character.

Arithmetic instructions should always be addressed to "signed" fields. Both positive and negative fields in memory should be signed. Numerical fields are signed by placing a plus or minus sign indication over the units digit of the field. The absence of a zone or the presence of a zero zone does not satisfy the requirements for a signed field.

Add (G—ADD)

Function. A numerical field in memory is added to a factor in accumulator or auxiliary storage.

Address. The address specifies the units position of the memory field and the storage unit to be used.

Limiting Factors. The first non-numerical character to the left of the addressed memory character.

Timing

MODEL	TIME (µs)	REMARKS
705 I, II	$34 + 17N_m$	All cases except that below:
	$51 + 34N_m$	Signs unlike and $\|N_m\| < \|N_s\|$
705 III	$40 + 9.3N_m$*	Mem. and stor. signs alike
		Mem. and stor. signs unlike
	$40 + 9.3N_m$	$\|N_m\| \geqq \|N_s\|$
	$67 + 18.5N_m$	$\|N_m\| < \|N_s\|$

* When addition is being done with like signs, the time to execute the instruction will be lengthened by 9.3 microseconds for each carry propagated beyond the length of the memory field. This is included in the time shown.

Description. Digits are added into storage starting with the specified right-hand digit of the memory field and continuing from right to left until a non-numerical character is sensed. This non-numerical character is not added to the storage factor.

The result in the storage unit is the sum of the storage factor and the specified memory field. The result replaces the original storage factor. The field in memory is not affected.

The accumulator or auxiliary storage sign is set according to the rules of algebra for addition. When the address character has neither plus nor minus zoning, a sign check error is indicated and the sign of the memory field is interpreted as plus. Refer to "Sign Check Indicator." The sign is always set to plus when the result in storage is plus.

The left-hand limit of the result is automatically set by a storage mark stored next to the highest-order digit. The length of the result is equal to the longer of the two fields being added, unless a carry is made out of the high-order position. In this case, the result is extended one position to include the carry as its most significant digit, the storage mark is positioned to the left of this digit, and the overflow check indicator is turned on. Refer to "Overflow Check Indicator."

When the overflow exceeds the capacity of an auxiliary storage unit, the carry is made into the adjacent unit with proper positioning of the storage mark in this unit.

EXAMPLES, ADD

	STORAGE BEFORE		STORAGE AFTER		CHECK
MEMORY	STORAGE	SIGN	STORAGE	SIGN	IND.
$\overset{+}{6}23\overset{+}{4}$	a23	+	a257	+	
$\overset{+}{6}23\overset{+}{5}$	a23	—	a212	+	
$\overset{+}{6}23\overset{-}{5}$	a23	+	a212	—	
$\overset{+}{6}235$	a23	+	a258	+	Sign check on
$\overset{+}{1}2345$	a15	+	a2360	+	Sign check on
$b8\overset{-}{9}$	a20	—	a109	—	Overflow check on
$b8\overset{-}{9}$	a89	+	a00	+	

Execution of the add instruction with the 705 I or II is not completed until the end of both the memory and storage fields is reached, plus any machine cycles necessary to handle carries propagated beyond the length of the two fields.

Execution of the instruction with the 705 III will be terminated at the end of the memory field plus any machine cycles necessary to handle carries propagated beyond the length of the memory field.

With the 705 I or II, it is possible to remove zoning from an entire storage field by adding a + or — zero. For example:

Add:	Storage	ABCD+
	Memory	b$\overset{+}{0}$
	Result	1234+

To remove zoning from the entire storage field on the 705 III, it is necessary to add a field of zeros equal in length to the storage field. For example:

Add:	Storage	ABCD+
	Memory	b000$\overset{+}{0}$
	Result	1234+

PROGRAM, ADDITION

Field A in Figure 22 is defined by an arithmetic instruction as containing the digits stored in the address 07005 through 07001, while field B contains the digits in addresses 07012 through 07006. The addresses of fields A and B are 07005 and 07012, respectively.

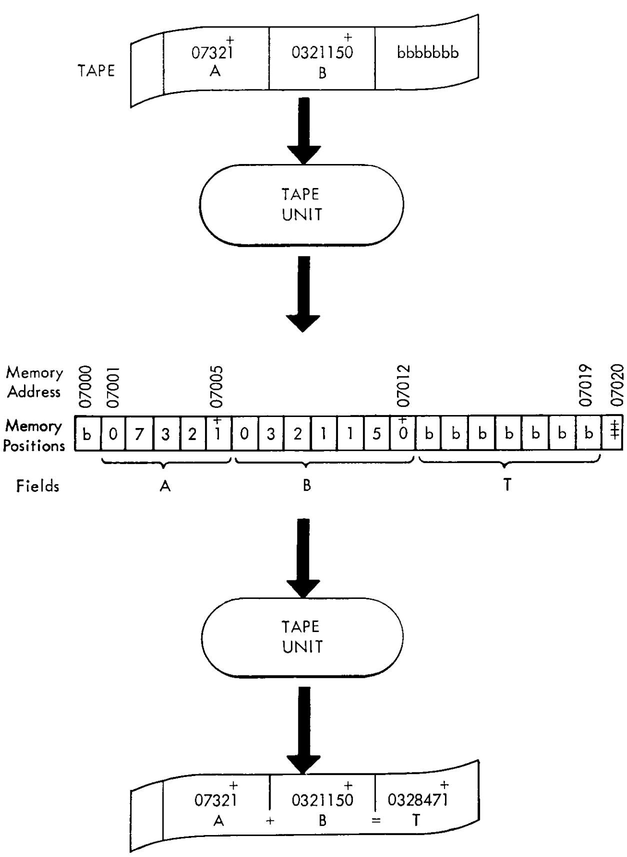


Figure 22. Addition, $A + B = T$

INSTR. LOCATION	INSTRUCTION OPER.	ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00004	SEL	00200					
00009	RD	07001					
00014	RAD	07005	00	a07321	+		
00019	ADD	07012	00	a0328471	+		
00024	ST	07019	00	a0328471	+		
00029	SEL	00201					
00034	WR	07001	00				
00039	TR	00004					

Figure 23. Program, $A + B = T$

Figure 23 shows the program for $A + B = T$.

00004. Tape unit 00200 is selected.

00009. The tape record, including blank characters, is read into memory from left to right, beginning at address 07001 and continuing to address 07019. It is assumed that a blank has been stored previously in address 07000 and a group mark at address 07020.

00014. The proper positions of accumulator storage are reset and field A is entered. Field A is defined in memory as beginning with the signed digit 1 in memory address 07005 and continuing successively to the left until the non-numerical or blank character (b) is sensed (Figure 22).

00019. Field B is added to field A. The result is placed in accumulator storage. The sign of storage is plus.

00024. The result T is stored in memory at address 07019. The sign of the accumulator (plus) is placed over the units digit of the field T at address 07019.

The character at address 07012, the first position of the next left-hand field, is already signed. Therefore, field T is limited in memory to the characters found at addresses 07019 to 07013, inclusive. The factor in the accumulator is unaffected by the store instruction.

00029. Tape unit 00201 is selected.

00034. The unit record stored in memory, beginning at address 07001, is written from left to right up to the group mark in 07020. The record in memory is unaffected by this instruction.

00039. A transfer instruction is given to repeat the program for successive records in the tape unit.

Add to Memory (6—ADM)

Function. A field in accumulator or auxiliary storage is added to a field in memory. The field from storage may be added to memory in two ways depending upon whether the memory field is signed or unsigned. The result replaces the original memory field and the field in storage is unchanged.

Address. The address specifies the units position of the memory field and the storage unit to be used.

Limiting Factors. (1) Signed memory field—first non-numerical character to the left of the addressed memory character, (2) Unsigned memory field—storage mark in the storage field.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$34 + 17N$	All cases except that below:
	$51 + 34N$	Signed field, signs unlike $\lvert N_m \rvert < \lvert N_s \rvert$
705 III	$40 + 10.8N_h$	Signed mem. field Mem. and stor. signs alike
	$40 + 10.8N_h$	Mem. and stor. signs opposite except when
	$67 + 21.6N_m$	$N_s \geqq N_m$ and $\lvert N_s \rvert > \lvert N_m \rvert$
	$49 + 10.8N_s$	Unsigned mem. field

Description.

SIGNED MEMORY FIELD. The addition follows the rules of algebra. The addressed field in memory starts with its right-hand signed digit and continues to the left until a non-numerical character is reached. Any carry-over of the result into the position of the next non-numerical character in memory is ignored and the overflow check indicator is *not* turned on. Only numerical portions of the character in storage are added. Both plus and minus fields in storage may be added. The proper sign of the result is placed over the addressed character in memory.

EXAMPLES, ADD TO MEMORY (SIGNED FIELDS)

ACC. OR AUX. STORAGE	SIGN	MEMORY BEFORE	MEMORY AFTER
a33	+	$\overset{+}{5}66\overset{+}{3}$	$\overset{+}{5}69\overset{+}{6}$
a25	—	$\overset{+}{5}42\overset{+}{5}$	$\overset{+}{5}40\overset{+}{0}$
a625	+	$\overset{-}{4}67\overset{+}{6}$	$\overset{-}{4}30\overset{+}{1}$
a676	—	$\overset{-}{4}62\overset{+}{5}$	$\overset{-}{4}05\overset{-}{1}$
a12676	—	$\overset{-}{4}62\overset{+}{5}$	$\overset{-}{4}05\overset{-}{1}$
a12121	—	$B45\overset{+}{6}$	$B33\overset{+}{5}$
a3	+	A&	$A\overset{+}{3}$

Execution of an ADM instruction (signed memory field) is completed at the end of the memory field in the 705 I and II. With the 705 III, execution terminates no later than at the end of the memory field but could terminate sooner, depending upon the length of the storage field.

Through the use of the ADM instruction (signed field), any given memory location can become an accumulative counter into which a number may be added with this one instruction.

UNSIGNED MEMORY FIELD. The addition is not algebraic. It begins with the right-hand digit of accumulator or auxiliary storage and the addressed character in memory and continues from right to left until the storage mark is reached.

When non-numerical characters, including blanks, are encountered in the memory or storage fields, both zones and digits are added. The zone positions of the characters are added separately as binary numbers. Any carry-over is added to the next high-order zone position, except a carry beyond the last character of the storage field into the position of the storage mark, which is disregarded.

The numerical parts of the characters are added decimally. Any carry-over is added only to the numerical part of the next high-order position, except a carry-over beyond the last character of the storage field into the accumulator or auxiliary storage mark.

Any carry from the numerical portion of the last character of the field is binarily added to the zone of that character. Any carry from the zone of the last character is disregarded.

The three possible zones are indicated in the following example by placing binary notation over the numerical portion of the character. The zero zone is numbered 01, the eleven zone 10, and the twelve zone 11. In binary, this corresponds to zones 1, 2, and 3, respectively.

EXAMPLES, ADD TO MEMORY (UNSIGNED FIELD)

ACC. OR AUX. STORAGE	SIGN	MEMORY BEFORE	MEMORY AFTER
$a\,\overset{00}{4}\,\overset{00}{0}$	+	$W\,\overset{00}{1}\,\overset{00}{2}\,\overset{00}{3}\,\overset{00}{4}$	$W\,\overset{00}{1}\,\overset{00}{2}\,\overset{00}{7}\,\overset{00}{4}$
$a\,\overset{00}{0}\,\overset{00}{9}\,\overset{00}{0}\,\overset{00}{0}$	+	$R\,\overset{00}{0}\,\overset{00}{1}\,\overset{00}{1}\,\overset{00}{1}\,\overset{00}{1}$	$R\,\overset{00}{0}\,\overset{00}{2}\,\overset{00}{0}\,\overset{00}{1}\,\overset{00}{1}$
$a\,\overset{00}{9}\,\overset{00}{9}\,\overset{00}{0}\,\overset{00}{0}$	+	$R\,\overset{00}{0}\,\overset{00}{1}\,\overset{00}{1}\,\overset{00}{1}$	$R\,\overset{01}{0}\,\overset{00}{0}\,\overset{00}{1}\,\overset{00}{1}$
a &	+	$\overset{00}{2}\,\overset{00}{6}$	$\overset{00}{2}\,\overset{11}{6}$
a —	—	$\overset{00}{2}\,\overset{00}{6}$	$\overset{00}{2}\,\overset{10}{6}$
$a\,\overset{01}{9}\,\overset{00}{9}\,\overset{00}{0}\,\overset{00}{0}$	+	$R\,\overset{11}{0}\,\overset{00}{1}\,\overset{00}{1}\,\overset{00}{1}$	$R\,\overset{01}{0}\,\overset{00}{0}\,\overset{00}{1}\,\overset{00}{1}$

Through the use of the ADM instruction (unsigned field), a method is provided for incrementing any instruction address in memory, within a memory block or from memory block to memory block. The instruction also can be used to change accumulator or storage unit designations of instructions located in memory.

Note that the add-to-memory (ADM) instruction cannot be used on the 705 III to increment any instruction address which refers to the upper 40,000 positions of memory. Obviously, the 1 in the B-bit position would cause the field to be considered as a signed field, and algebraic addition would result, affecting only those characters to the right of the next non-numerical character in memory. The unsigned ADM instruction causes a wrap-around at 40,000. If a memory field is standing at 39,999 and is increased by 1 with the ADM instruction, the field reverts to 00,000. Programs written for 705 I or II, using the ADM instruction, work in the same manner on the 705 III.

Add Address to Memory (@—AAM) (705 III)

Function. A five-digit field in accumulator or auxiliary storage is added to a four-digit field in memory.

Address. The address specifies the units position of the memory field and the storage unit to be used.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 III	85	- - - - - - -

Description. To execute this instruction, the first four low-order digits of the storage field are added to the numerical portion of the four characters of the memory field. The numerical bits of the digit in the fifth-order position of the storage field are added to the memory field in the following way. Bits in the 2- and 1-bit positions are added to the B and A zone bits, respectively, of the character in the thousands-order position of the memory field. The bit in the 4-bit position is added to the B zone bit position of the character in the units position of the memory field. The 8 bit is ignored unless there is an 8-and-2 bit combination, which is interpreted as zero.

A carry developing from the A-bit position adds to the B-bit position of the character in the thousands-order memory position and from there to the B bit of the character in the units order memory position. Any carry from the B bit of the units position is disregarded.

For example, in memory location 15004 is the instruction RAD 40850, appearing actually as RAD 0 8 5 $\overset{10}{0}$. The address portion of this instruction (40850) is to be incremented by 25000 so that it will read RAD 65850, actually RAD $\overset{10}{5}$ 8 5 $\overset{10}{0}$. With the 705 III, this can be accomplished by placing the addend 25000 in the accumulator or auxiliary storage such as ASU 05, and executing the instruction AAM 15004 (05).

All addition with the AAM instruction is non-algebraic, always adding the absolute value of the amount in storage regardless of the sign. The zone bits over the tens and hundreds positions in memory (the ASU designation) add in binary fashion with any zoning over the corresponding positions in storage. In this regard, the AAM instruction works in the same way as the existing add-to-memory (ADM) instruction for an unsigned field in memory. However, it is unlike ADM in that the addition of the zones over these two digits is independent of zone addition of any other positions, and any carry from the hundreds position zone is disregarded.

A 1 in the A-bit position of the units character in memory, indicating an indirect address, is unaffected by the AAM instruction.

Thus, the AAM instruction may be used to increment any instruction address in memory to any value within the range of 00000 to 79999. The AAM instruction is said to cause a wrap-around at 80,000. That is, if a memory field were standing at 79999 and were incremented by 1 with the AAM instruction, it would revert to 00000; or, if 10001 were added to 79999, the result would be 10000.

Because addition with the AAM instruction is non-algebraic, subtraction must be accomplished by taking advantage of this wrap-around. A memory field standing at 60,000 can be reduced to 30,000 by adding 50,000. That is, 20,000 added to 60,000 gives 00,000; 30,000 more makes 30,000. Or, to state it differently: to accomplish subtraction, add the 80,000's complement of the number to be subtracted. Thus, to subtract 30,000 from 60,000, add the 80,000's complement of 30,000 or 50,000.

Use of the AAM instruction causes a wrap-around at the upper end of memory.

If the field in storage that is to be added to a field in memory is greater than five digits, the AAM instruction adds only the first (low order) five digits and ignores the remainder. The field in storage may also be less than five digits. (It will be treated as a five-digit field with 0's to the left of significant digits.) This makes the presetting of the ASU to a specific length unnecessary.

The following example demonstrates the effect of the AAM instruction:

STORAGE	MEMORY BEFORE	MEMORY AFTER
a 2 5 0 0 0	0 0 5 5	$\overset{10}{5}$ 0 5 5
a 5 0 0 0 0	$\overset{10}{5}$ 0 5 5	$\overset{11}{5}$ 0 5 $\overset{10}{5}$
a 4 9 4 5	$\overset{11}{5}$ 0 5 $\overset{10}{5}$	0 0 0 0
a 3 2 $\overset{10}{0}$ $\overset{10}{0}$ 5	$\overset{11}{2}$ $\overset{10}{0}$ $\overset{10}{0}$ $\overset{01}{4}$	$\overset{10}{4}$ $\overset{01}{0}$ 0 $\overset{11}{9}$
a 4 4 $\overset{11}{4}$ 4 4 $\overset{10}{4}$ $\overset{01}{4}$	$\overset{10}{2}$ 2 $\overset{10}{2}$ 2	$\overset{10}{6}$ $\overset{01}{6}$ 6 $\overset{10}{6}$
a 3 0 0 0 0	$\overset{11}{0}$ 0 0 $\overset{10}{0}$	$\overset{10}{0}$ 0 0 0
a $\overset{01}{1}$ $\overset{10}{1}$ $\overset{01}{1}$	1 $\overset{11}{2}$ 3 $\overset{01}{4}$	1 3 $\overset{10}{4}$ $\overset{01}{5}$

Subtract (P—SUB)

Function. A numerical field in memory is subtracted from a factor in accumulator or auxiliary storage.

Address. The address specifies the units position of the memory field and the storage unit to be used.

Limiting Factors. The first non-numerical character to the left of the addressed memory character.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$34 + 17N_m$	All cases except that below:
	$51 + 34N_m$	Signs alike and $\lvert N_m\rvert < \lvert N_s\rvert$
705 III	$40 + 9.3N_m$*	Mem. and stor. signs opposite
		Mem. and stor. signs alike
	$40 + 9.3N_m$	$\lvert N_m\rvert \geqq \lvert N_s\rvert$
	$67 + 18.5N_m$	$\lvert N_m\rvert < \lvert N_s\rvert$

* When addition is being done in the subtract instruction (subtract with unlike signs), the time to execute the instruction will be lengthened by 9.3 microseconds for each carry progagated beyond the length of the memory field. This is included in the time shown.

Description. Digits in memory are subtracted from storage starting with the specified right-hand digit of the memory field and continuing from right to left until a non-numerical character is sensed. This non-numerical character is not subtracted from storage.

The result in accumulator or auxiliary storage is the difference between the storage factor and the specified memory field. The result replaces the original storage factor. The field in memory is not affected.

The accumulator or auxiliary storage sign is set according to the rules of algebra for subtraction. When the addressed character has neither plus nor minus zoning, a sign check error is indicated and the sign of the field is interpreted as plus. Refer to "Sign Check Indicator." The sign is always set to plus when the result in storage is zero.

The left-hand limit of the result is automatically set by a storage mark stored next to the highest-order digit.

The length of the result equals the longer of the two fields being subtracted, unless a carry is made out of the highest-order position. In this case, the result is extended one position to include the carry as its most significant digit, the storage mark is positioned to the left of this digit, and the overflow check indicator is turned on. Refer to "Overflow Check Indicator."

When the overflow exceeds the capacity of an auxiliary storage unit, the carry is made into the adjacent unit with proper positioning of the storage mark in this unit.

Execution of the subtract instruction, with the 705 I or II, is not completed until the end of both the memory and storage field is reached, plus any machine cycles necessary to handle carries propagated beyond the length of the two fields.

Execution of the instruction, with the 705 III, is terminated at the end of the memory field plus any machine cycles necessary to handle carries propagated beyond the length of the memory field.

Program, Crossfooting

The tape record shown in Figure 24 is to be read into memory at addresses 13988 to 14014, inclusive. A

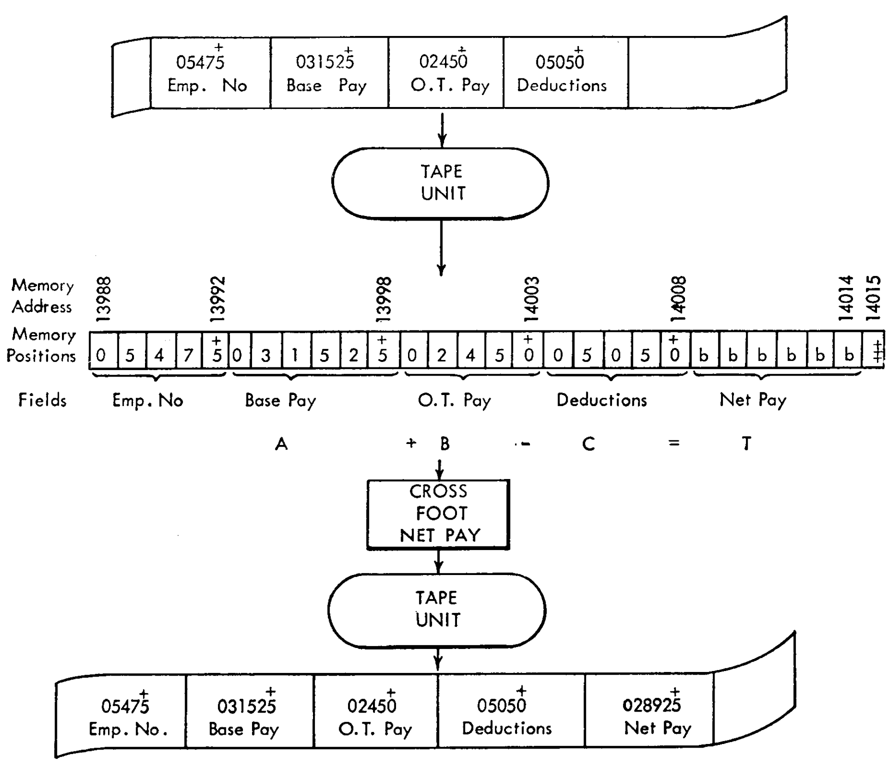


Figure 24. Crossfooting

group mark is stored at address 14015. Fields A, B, and C are signed plus. The entire record is to be written on tape after calculation.

Figure 25 shows the program for $A + B - C = T$.

00004. Select tape unit 00200.

00009. Read the tape record into memory, beginning with address 13988.

00014. Reset and add field A into auxiliary storage unit 01.

00019. Add field B from memory to the contents of auxiliary storage 01 to obtain the result of A + B.

00024. Subtract field C in auxiliary storage 01 to obtain the result T.

00029. Store the result T in memory at address 14014.

00034. Select tape unit 00201.

00039. Write out the completed record, beginning at address 13988. It is assumed that a group mark has been placed at memory address 14015.

00044. Transfer to repeat the program for succeeding records.

INSTR. LOCATION	INSTRUCTION OPER.	INSTRUCTION ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00004	SEL	00200					
00009	RD	13988					
00014	RAD	13998	01			a031525	+
00019	ADD	14003	01			a033975	+
00024	SUB	14008	01			a028925	+
00029	ST	14014	01			a028925	+
00034	SEL	00201					
00039	WR	13988	00				
00044	TR	00004					

Figure 25. Program, $A + B - C = T$

Multiply (V—MPY)

Function. A factor in memory is multiplied by a factor in accumulator storage. The field in memory is the multiplicand and the storage factor is the multiplier. The product is developed in storage.

Address. The address specifies the units position of the memory field.

Limiting Factors. First non-numerical character to the left of the addressed memory character.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$17\ [N_s\ (N_m + 4) + 2]$	-------
705 III	$58 + N_s\ (63 + 9.3N_m)$	-------

Description. The multiply instruction causes a field in memory (multiplicand) to be multiplied by a field in accumulator storage (multiplier). The product is developed in accumulator storage 128 positions away from the units position of the multiplier.

The number of digits in the product is equal to the sum of the number of digits in the multiplier and multiplicand. A maximum product of 128 digits can be obtained.

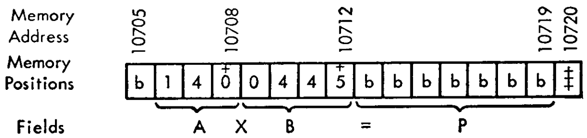


Figure 26. Multiplication, $A \times B = P$

The accumulator sign is set to plus if both multiplier and multiplicand have like signs, and minus if they have unlike signs. When the addressed character of the field in memory has neither plus nor minus zoning, an error is indicated and the sign of the field is interpreted as plus. Refer to "Sign Check Indicator."

Only numerical fields can be used in multiplication. The use of non-numerical fields produces inconsistent results.

The multiplier may be recovered in accumulator storage by executing the instruction SHR 00128. Refer to "Shorten." The product may be half adjusted by use of the round instruction.

Examples, Multiply

MEMORY	ACC. 00 BEFORE STORAGE	SIGN	ACC. 00 AFTER STORAGE	SIGN	CHECK INDICATOR
$\overset{+}{2}8\overset{+}{0}$	a7	—	a560	—	
$b\overset{+}{3}$	a2	+	a06	+	
$\$2\bar{5}$	a31	—	a0775	+	
$\overset{+}{6}\bar{5}$	a007	+	a0035	—	
b5	a007	+	a0035	+	Sign check on

Program, Multiplication

The record shown in Figure 26 is stored in memory from tape at addresses 10706 through 10719, inclusive. The field P is blank and is to be calculated. The entire record is to be written on tape after calculation.

Figure 27 shows the program for $A \times B = P$.

00004. Select tape unit 00203.

00009. Read the record into memory beginning at address 10706.

00014. Reset and add A into accumulator storage.

INSTR. LOCATION	INSTRUCTION OPER.	INSTRUCTION ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00004	SEL	00203					
00009	RD	10706					
00014	RAD	10708	00	a140	+		
00019	MPY	10712	00	a0062300	+		
00024	ST	10719	00	a0062300	+		
00029	SEL	00200					
00034	WR	10706	00				
00039	TR	00004					

Figure 27. Program, Multiplication, $A \times B = P$

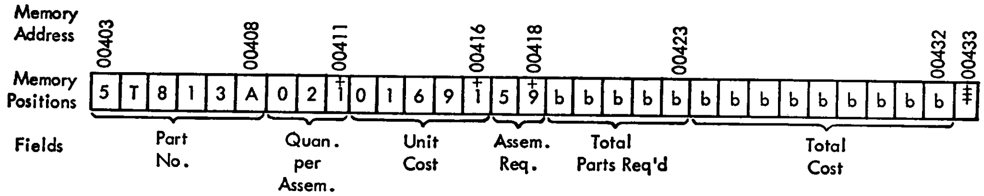


Figure 28. Multiply and Round

00019. Multiply A by B. The product is developed in accumulator 00. The number of digits in the product is equal to the sum of the digits in A and B, including zeros.

00024. The result P of seven digits is stored in memory at address 10719.

00029. Select tape unit 00200.

00034. Write the completed record on tape from memory, beginning at address 10706.

00039. Transfer to instruction 00004 to repeat the program.

Program, Multiplication with Half Adjustment

The product, P, may be half adjusted in accumulator 00 at any position by use of the round instruction. The record shown in Figure 28 is stored in memory from tape. Computations to be made in the record are as follows:

a. Quantity of parts per assembly × assemblies required = total quantity of parts required.
b. Total quantity of parts required × unit cost (3 decimals) = total cost.
c. Adjust total cost to the nearest cent.

Figure 29 shows the program for completion of the record shown in Figure 28.

00004. Select tape unit 00200.

00009. Read the record into memory, beginning at memory address 00403.

00014. Reset and add the quantity of assemblies required at address 00418.

INSTR. LOCATION	INSTRUCTION OPER.	INSTRUCTION ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00004	SEL	00200					
00009	RD	00403					
00014	RAD	00418	00	a59	+		
00019	MPY	00411	00	a01239	+		
00024	ST	00423	00				
00029	MPY	00416	00	a0002095149	+		
00034	RND	00001	00	a000209515	+		
00039	ST	00432	00				
00044	SEL	00201					
00049	WR	00403	00				
00054	TR	00004					

Figure 29. Program, $A \times B = P$ (Rounded)

00019. Multiply: Quantity of assemblies × parts per assembly = total quantity of parts required.

00024. Store total quantity of parts required at address 00423.

00029. Multiply: Parts required × unit cost = total cost. Unit cost contains three decimals.

00034. Adjust total cost to nearest cent by executing round 00001.

00039. Store total cost at address 00432.

00044. Select tape unit 00201.

00049. Write the record on tape. Assume that a group mark has been placed in memory at address 00433.

00054. Transfer to repeat the program.

Divide (W—DIV)

Function. A factor in accumulator storage is divided by a field in memory. The memory field is the divisor and the accumulator factor the dividend. The quotient is developed in accumulator storage.

Address. The address specifies the units position of the memory field.

Limiting Factors. First non-numerical character to the left of the addressed memory position.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$17\,[11 + N_s + (N_s - N_m)(7.5N_m + 15)]$	Average formula
705 III	$90 + 9N_s + (N_s - N_m)(6.7N_m + 37)\,9$	....

Description. The divide instruction causes a factor in accumulator storage 00 to be divided by the field in memory specified by the address part of the instruction. Thus, the memory field is the divisor and the accumulator factor is the dividend. The quotient is developed in accumulator storage.

The number of digits in the quotient is equal to the number of digits in the dividend less the number of digits in the divisor, including insignificant zeros. A maximum dividend length of 128 digits can be used.

1. The dividend must contain a greater number of digits than the divisor. Otherwise, the division is ignored, the zero indicator is turned on, and the machine proceeds to the next instruction.

2. The divisor must have a greater absolute value than an equal number of digits taken from the left end of the dividend. For example, 7 ÷ 2 cannot be performed because the divisor, 2, is less than the dividend 7; 07 ÷ 2 can be performed because the divisor 2 is of greater value than the high-order digit of the dividend, 0; 1234 ÷ 13 can be performed because the divisor 13 is of greater value than the two high-order digits of the dividend, 12. This rule can be satisfied by inserting zeros in the high-order positions of the dividends as required. If this rule is not satisfied:

a. The overflow check indicator and the zero indicator turn on.

b. The division is not completed.

c. A single zero replaces the contents of accumulator storage (quotient).

d. The machine proceeds to the next instruction.

e. The accumulator sign remains the same as that of the replaced dividend.

The accumulator sign is plus if the divisor and dividend have like signs, and minus if they have unlike signs. When the right-hand character of the field in memory has neither plus nor minus zoning, an error is indicated and the sign of the field is interpreted as plus. Refer to "Sign Check Indicator."

Only numerical fields may be used in division. Nonnumerical fields produce inconsistent results.

After division, the remainder may be recovered in either of two ways:

a. By multiplying the quotient by the divisor and subtracting the result from the dividend.

b. By performing a shorten operation with an address part equal to 00128 minus the length of the divisor. Refer to "Shorten."

The quotient is developed to the right of the storage mark which is positioned 128 positions to the left of the storage mark defining the dividend. The remainder always occupies the same accumulator positions as the original dividend. For example, after dividing 02333 by 111, accumulator storage would appear as in Figure 37A.

If a shorten 00128 instruction is given, the starting point counter moves 128 positions to the left and stops under the position containing the second zero to the right of the remainder storage mark (Figure 37B). However, the starting point counter is to be placed under the 2. This position is three positions (number of digits in divisor) to the right. Therefore, shorten 00125 places the starting point counter properly (Figure 37C). The storage mark to the right of the remainder is placed there automatically at the beginning of the division operation and is used to stop the division.

The number of significant digits in the remainder can never exceed the number of digits in the divisor. Therefore, it is usually advantageous to set left the number of digits in the divisor before storing or using the remainder.

Examples, Division

MEMORY	ACCUMULATOR BEFORE STORAGE	SIGN	ACCUMULATOR AFTER STORAGE	SIGN	CHECK INDICATOR
A5$\overset{+}{0}$	a2501	+	a50	+	
b5$\overset{+}{0}$	a511	+	a0	+	Zero on Overflow on
b5$\overset{+}{0}$	a55	+	a55	+	Zero on
b5$\overset{-}{0}$	a12700	+	a254	—	
b50	a12700	+	a254	+	Sign check on
b5$\overset{+}{0}$	a604	—	a0	—	Zero on Overflow on

Division by Zero

Division by zero always violates the rule that the divisor must be greater than the value of an equal number of digits taken from the left end of the dividend. However, when zero is divided by zero, the violation of this rule does not turn on the overflow check indicator. Division by zero results in a quotient of a single zero, ones and a zero, or all ones.

In most divisions, it is necessary to know a great deal about the divisor and its relationship to the dividend. Where this is not the case, it is recommended that a transfer-on-zero instruction be inserted to determine whether the dividend is a zero. If it is a zero, a transfer may be made to examine the divisor or take other corrective action.

Program, Division

The fields A and B are stored in memory at addresses 16026 and 16028, respectively, as shown in Figure 30. The quotient Q is to be stored at address 16031. Field

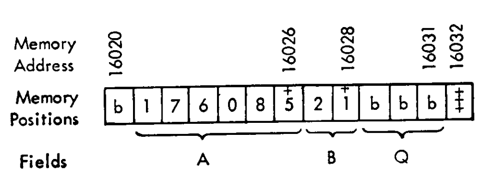


Figure 30. Division, $A \div B = Q$

INSTR. LOCATION	INSTRUCTION OPER.	ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00004	SEL	00200					
00009	RD	16021					
00014	RAD	16026	00	a176085	+		
00019	DIV	16028	00	a8385	+		
00024	RND	00001	00	a839	+		
00029	ST	16031	00				
00034	SEL	00201					
00039	WR	16021	00				
00044	TR	00004					

Figure 31. Program, Division, $A \div B = Q$

A contains six positions; field B, two positions; field Q, three positions.

Figure 31 shows the program for $A \div B = Q$.

00004. Select tape unit 00200.

00009. Read the record into memory, beginning at address 16021.

00014. Reset and add the dividend A into accumulator storage.

00019. Divide A by B to produce the quotient Q in accumulator storage.

00024. Half adjust Q one position.

00029. Store Q at memory address 16031.

00034. Select tape unit 00201.

00039. Write the record on tape. Assume a group mark has been placed at address 16032.

00044. Transfer to repeat the program.

Program, Gross Pay Calculation

Programming a section of a payroll problem further demonstrates the combined use of the previously discussed arithmetic instructions.

The payroll data are received on tape and are stored in memory as shown in Figure 32. Fields are: man number, incentive earnings, hourly earnings, overtime allowance hours, and regular hours. The calculations to be performed are:

$$\text{Average rate} = \frac{\text{regular earnings} + \text{incentive earnings}}{\text{regular hours}}$$

$$\text{Overtime amt.} = \text{average rate} \times \text{overtime allowance hours}$$

$$\text{Gross pay} = \text{regular earnings} + \text{incentive earnings} + \text{overtime amount}$$

Figure 33 is the program for gross pay calculation.

00004. Select tape unit 00200.

00009. Read the record into memory beginning at address 00901.

INSTR. LOCATION	INSTRUCTION OPER.	ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00004	SEL	00200					
00009	RD	00901					
00014	RAD	00929	00	a05500	+		
00019	ADD	00933	00	a06756	+		
00024	SET	00006	00	a006756	+		
00029	LNG	00002	00	a00675600	+		
00034	DIV	00922	00	a01535	+		
00039	RND	00001	00	a0154	+		
00044	SET	00003	00	a154	+		
00049	ST	00936	00				
00054	MPY	00924	00	a03080	+		
00059	RND	00001	00	a0308	+		
00064	ST	00940	00				
00069	ADD	00929	00	a05808	+		
00074	ADD	00933	00	a07064	+		
00079	ST	00945	00				
00084	SEL	00201					
00089	WR	00901	00				
00094	TR	00004					

Figure 33. Program, Gross Pay Calculation

00014. Reset and add regular earnings into accumulator storage.

00019. Add incentive earnings to regular earnings. The result is the dividend to be divided by regular hours.

00024. One zero is added to the left of the dividend to insure that the absolute value of the divisor is always greater than an equal number of digits taken from the left end of the dividend.

00029. The divisor (regular hours) has one decimal, the quotient (average rate) has two decimals, and the quotient is to be half adjusted, making a total of four decimals to point off in the dividend. Because the dividend now has only two decimals, it must be lengthened two positions by adding zeros. The number of decimals in the quotient, plus one for half adjustment, equals the number of positions to the right of the decimal point in the dividend.

00034. Divide:

$$\text{Average rate} = \frac{\text{regular earnings} + \text{incentive earnings}}{\text{regular hours}}$$

00039. Half adjust the average rate to the nearest cent.

00044. Adjust average rate to three positions.

00049. Store average rate.

00054. Multiply: Average rate $\times$ overtime allowance hours $=$ overtime earnings.

00059. Adjust overtime earnings to the nearest cent.

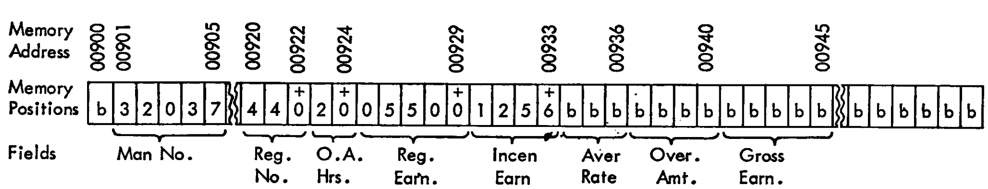


Figure 32. Gross Pay Calculation

00064. Store overtime earnings.
00069. Add regular earnings + overtime earnings.
00074. Add incentive earnings + gross pay.
00079. Store gross earnings.
00084. Select tape unit 00201.
00089. Write the record on tape.
00094. Transfer to repeat the program.

Shift Operations

The 705 can perform shift instructions to adjust field length in accumulator or auxiliary storage units. Fields may be lengthened to the right or left with the addition of zeros, or may be shortened by moving the storage mark or the starting point counter as required. These instructions are useful for eliminating insignificant zeros from calculated results, coupling auxiliary storage units, adjusting the dividend and placing the decimal in division operations, preparing storage units for load operations, and so on.

The shorten, lengthen, and round instructions may be used only by specifying accumulator storage. The set left instruction may specify accumulator or auxiliary storage units.

Set Left (B—SET)

Function. This operation adjusts the length of the accumulator or auxiliary storage field, by moving the storage mark the number of positions indicated by the address part of the instruction.

Address. The address indicates the number of positions the storage mark is to be moved.

Limiting Factors. When the storage mark has been moved the specified number of positions, the operation stops.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$51 + 17N_a$	-------
705 III	$40 + 9.3N_a$	-------

Description. The set left instruction, by moving the storage mark, adjusts the length of the accumulator or auxiliary storage field to the number of characters specified by the address part of the instruction. The operation starts with the character in the starting point counter position and continues to the left, character by character, until the number of storage positions specified by the address have been examined.

When a storage mark is sensed, it is replaced by a zero and zeros are placed to the left in all remaining high-order positions of the adjusted field (Figure 34A). When the storage mark is not sensed, a storage mark is inserted to the left of the high-order character of the adjusted field and no characters are replaced with zeros (Figure 34B). The set left instruction may, therefore, either decrease the number of characters in the field by moving the storage mark toward the right to the starting point counter, or add zeros by moving the storage mark toward the left, as required.

A group of adjacent auxiliary storage units may be coupled by a set left instruction. Set left 00035 specifying ASU 01 uses auxiliary storage 01, through 02, to the fourth position of storage unit 03. The length of the resulting field is 35 positions (Figure 35). Zeros are added in the normal manner when the storage mark is moved to the left, or, when a storage mark is entered before the one which pre-

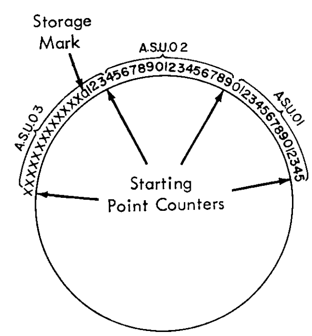


Figure 35. Auxiliary Storage Coupling

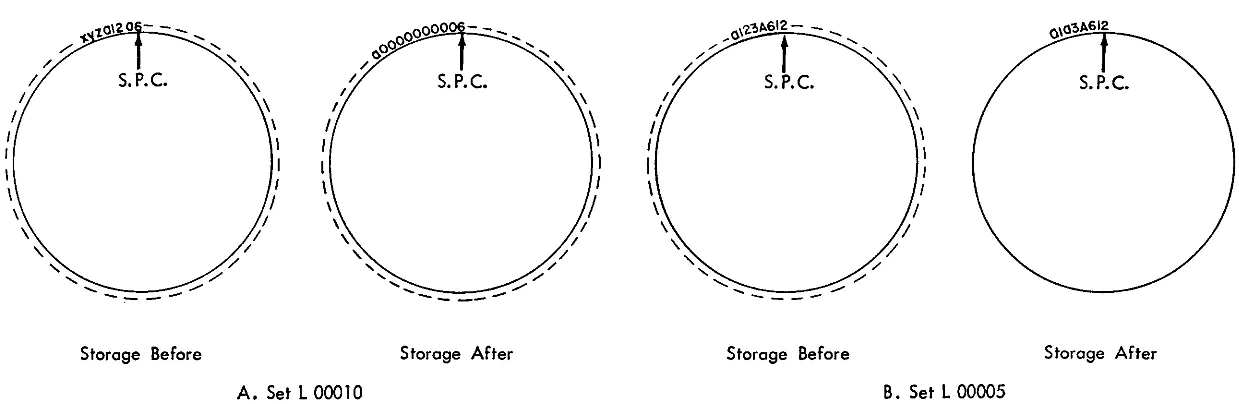


Figure 34. Schematic, Set Left Instruction

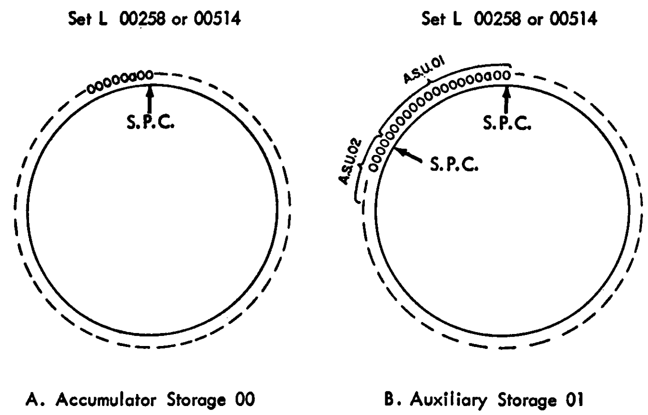


Figure 36. Schematic, Set Left Instruction

viously limited one field, the number of characters in the field is decreased.

A subsequent store instruction, when addressed to auxiliary storage 01, stores all 35 positions of the field. The store instruction, addressed to storage 03, stores only the three digits from the starting point counter position of 03 to the storage mark.

Because the accumulator and auxiliary storage units operate as a circle, an address part greater than 00256 causes the machine to examine the entire circle as many times as the number 256 can be completely subtracted from the address. The remainder following the last subtraction is the number of positions from the starting point to the storage mark (Figure 36). All accumulator or auxiliary storage positions contain zeros following an instruction of this sort, with the exception of the position containing the storage mark. For example, set left 00258 or set left 00514 places the storage mark in accumulator storage 00 in the same position as set left 00002. All positions of the resulting storage field, except the second position to the left of the starting point field, contain zeros.

Examples, set left

INSTRUCTION	STORAGE BEFORE	SIGN	STORAGE AFTER	SIGN
SET 00004	a52	—	a0052	—
SET 00002	a0052	+	a52	+
SET 00001	a52	+	a2	+
SET 00003	a2000	—	a000	+

The set 00000 instruction places a storage mark at the position of the starting point of the field, turns on the zero indicator, and sets the sign of accumulator or auxiliary storage units to plus.

Shorten (C—SHR)

Function. This operation shifts the starting point counter of accumulator storage to the left the number of positions specified by the address part of the instruction.

Address. The address indicates the number of positions the starting point counter is to be moved.

Limiting Factors. Movement of the starting point counter the specified number of positions.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$68 + 17N_a$	-------
705 III	$50 + 9N_a$	-------

Description. The shorten instruction shifts the starting point counter of accumulator storage to the left. The number of positions moved is specified by the address part of the instruction. Because the field in storage consists of those characters between the position of the starting point counter and the storage mark, the movement of the starting point counter to the left has the effect of removing characters from the right end of the storage field (Figure 37).

The shorten 00000 instruction has no effect. The position of the starting point counter, when using address parts greater than 00256, can be found by con-

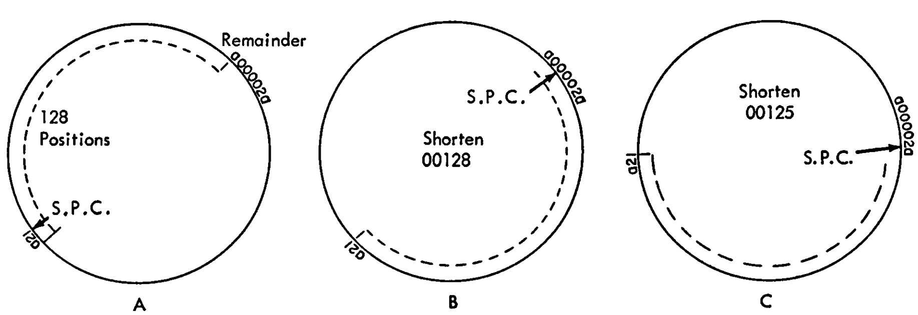


Figure 37. Schematic, Shorten Instruction

sidering the circular nature of accumulator storage and its total capacity of 256 positions. Thus, shorten 00769, 00257, and 00001 all have the same effect.

When, as a result of a shorten instruction, the field in accumulator storage is zero, the accumulator sign is always set to plus.

EXAMPLES, SHORTEN

INSTRUCTION	ACCUMULATOR BEFORE STORAGE	ACCUMULATOR BEFORE SIGN	ACCUMULATOR AFTER STORAGE	ACCUMULATOR AFTER SIGN
SHR 00002	a1246	+	a12	+
SHR 00001	a1246	—	a124	—
SHR 00000	a1246	—	a1246	—
SHR 00002	a0046	—	a00	+

Lengthen (D—LNG)

Function. This operation shifts the starting point counter of accumulator storage to the right the number of positions specified by the address part of the instruction.

Address. The address indicates the number of positions the starting point counter is to be moved.

Limiting Factors. Movement of the starting point counter the specified number of positions.

Timing

MODEL	TIME (µs)	REMARKS
705 I, II	$51 + 17N_a$	-------
705 III	$31 + 9N_a$	-------

Description. The lengthen instruction shifts the starting point counter of accumulator storage to the right. The address part of the instruction specifies the number of positions to be moved.

A zero is inserted to the right of the field in storage for each position moved by the starting point counter. The number of zeros inserted is designated by the address part of the instruction.

A storage mark is always placed in the position to the right of the starting point counter. This occurs even in the case of LNG 00000, which performs no lengthening function. Because of the circular nature of accumulator storage, an address part greater than 00254 fills accumulator storage with 255 zeros and a storage mark. The final position of the starting point counter can be found by repetitive subtraction of 256 from the address part. The remainder, following the last subtraction, indicates the position of the counter.

EXAMPLES, LENGTHEN

INSTRUCTION	ACCUMULATOR BEFORE	ACCUMULATOR AFTER
LNG 00002	a5723	a572300
LNG 00005	a1	a100000
LNG 00001	a4689	a46890

Round (E—RND)

Function. This operation shifts the starting point counter of accumulator storage to the left the number of positions specified by the address part of the instruction. A 5 is added to the digit to the right of the final position of the starting point counter.

Address. The address indicates the number of positions the starting point counter is to be moved.

Limiting Factors. Movement of the starting point counter the specified number of positions, addition of the numerical 5, and any resulting carries.

Timing

MODEL	TIME (µs)	REMARKS
705 I, II	$85 + 17N_a$	-------
705 III	$68 + 9N_a$	-------

Description. The round instruction moves the starting point counter of accumulator 00 to the left the number of positions specified by the address part of the instruction. Only accumulator 00 can be specified. The field remaining in storage is limited to those digits between the accumulator mark and the new position of the starting point counter.

A 5 is added to the digit to the right of the final position of the starting point counter. Any resulting carry is added to the units digit of the remaining storage field.

When a carry is made out of the high-order position of the original field, the result is extended one position to the left to include the carry, and the overflow check indicator is turned on.

The instruction round 00000 has no effect.

When the result in accumulator storage is zero, the sign is always set to plus.

When the position where the 5 is to be added happens to contain a storage mark, the 5 is placed in the accumulator at this point, but a storage mark is placed at the position of the starting point counter. Thus, the result has a zero field length. In this case, the overflow check indicator is turned on. NOTE: In the 705 I and II, the zero indicator is turned on and the storage sign is not changed. In the 705 III, the zero indicator is *not* turned on and the storage sign is not changed with the overflow switch in program position.

EXAMPLES, ROUND

INSTRUCTION	ACC. 00 BEFORE STORAGE	ACC. 00 BEFORE SIGN	ACC. 00 AFTER STORAGE	ACC. 00 AFTER SIGN	CHECK IND.
RND 00002	a5653	+	a57	+	
RND 00002	a5653	—	a57	—	
RND 00004	a98912	+	a10	+	O'flow on
RND 00001	a349	+	a35	+	
RND 00002	a0049	—	a00	+	
RND 00003	a146a41	+	a14a	+	O'flow on

Decision and Control Operations

Decision Operations

The decision operations provide the stored program with the ability to control its course of action by modifying the sequence in which any instruction or block of instructions is executed.

The results of interrogating or "looking at" a designated indicator, switch, or some other condition within the machine can be used to direct the program to one of several alternate program routines. This branch or subroutine is made up of instructions to calculate, rearrange the record, select specific input-output units, check results, perform end-of-file routines, and so on.

Compare (4—CMP)

Function. This operation compares the contents of accumulator or auxiliary storage with a field in memory.

Address. The address specifies the units position of the memory field and the storage unit to be used.

Limiting Factors. Storage mark in the storage field to be compared.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$34 + 17N_s$	-------
705 III	$31 + 9N_s$	-------

Description. The compare instruction compares the contents of accumulator or auxiliary storage with the portion of memory specified by the address part of the instruction. The particular storage unit to be used is also specified by the address.

The comparison begins between the specified character in memory and the right-hand character of the designated storage unit. It proceeds from right to left, character by character, between storage and memory until a storage mark is sensed. The results of the comparison are determined in the usual way; that is, the most significant characters are those on the left. The number of characters compared is equal to the number of positions in the storage field.

All characters that can appear in memory may be compared. The tape mark, drum mark, and storage mark do not normally appear in memory. The record mark and group mark can be compared. The ascending sequence of characters is as follows:

blank . ⌑ ⧧ & $ * - / , % # @ $\overset{+}{0}$ A through I
$\bar{0}$ J through R ⧧ S through Z 0 through 9

The results of a comparison can be "interrograted" or tested by two special transfer instructions, transfer on high and transfer on equal. When the storage field is higher than the field in memory, a following transfer-on-high instruction transfers to the location of the program step specified by the address of the transfer instruction. When the two fields are equal, a transfer-on-equal instruction accomplishes the same result. If neither of these conditions exists, the storage field is lower than the field in memory. In this way, supplementary routines may be programmed to handle the results of comparisons.

Any number of tests may be made between comparisons. The result of a comparison is not changed until another comparison is made.

Either the results of calculation or data loaded into storage may be compared against a portion of memory. However, a calculated result appears in storage without a sign indication over the right-hand digit while a numerical field in memory usually appears with a signed right-hand digit. A comparison between such fields would be unequal because of the sign.

On the 705 III, it is possible to have a 00901 check indication when more than one of the three possible indications (high, equal, low) exists after the execution of the compare instruction.

EXAMPLES, COMPARE

		CONSOLE INDICATOR NEONS		
STORAGE	MEMORY	HIGH	LOW (705 III)	EQUAL (705 I/II)
a123$\overset{+}{4}$	$\$123\overset{+}{4}$			on
a1234	$\$123\overset{+}{4}$	on		
a123$\overset{+}{4}$	$\$234\overset{+}{5}$		on	
aTYPEb705	TYPEb704	on		
aSMITHbbb	WARFIELD		on	
aSMITHbbb	SMITHbbb			on

Transfer on High (K—TRH)

Function. This operation interrogates the high comparison indicator and effects a transfer if on.

Address. The address indicates the instruction location address to which the machine transfers if the high comparison trigger is on.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	34	--------
705 III	22	--------

Description.

1. The transfer-on-high instruction interrogates the high comparison indicator. When the indicator is on, the machine transfers to the instruction specified by the address part of the transfer-on-high instruction.

2. If the high indicator is not on, a transfer is not made and the machine proceeds to the next sequential instruction.

3. The instruction can be used during any program step after a comparison.

4. The instruction can be used any number of times between comparisons without turning the indicator off. The indicator is turned off on the next compare instruction.

Transfer on Equal (L—TRE)

Function. This operation effects a transfer if an equal condition exists from a comparison of a memory and storage field.

Address. The address indicates the instruction location address to which the machine transfers if an equal condition exists.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	34	--------
705 III	22	--------

Description.

1. The transfer-on-equal operation effects a transfer if an equal condition exists from a comparison of a memory and storage field. With an equal condition, the machine transfers to the instruction specified by the address part of the transfer-on-equal instruction.

2. If an equal condition is not indicated, a transfer is not made and the machine proceeds to the next instruction.

3. The instruction can be used during any program step following a comparison.

4. The instruction can be used any number of times between comparisons without removing the indication.

Normalize and Transfer (X—NTR)

Function. This operation interrogates the left-hand character of a storage field. If the numerical part of the character is zero, the character is removed and a transfer is effected.

Address. The address specifies the memory address to which the machine transfers if transfer is effected and specifies the storage unit to be used.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	$51 + 17N_s$	Not normalized
	$68 + 17N_s$	Normalized
705 III	$58 + 9N_s$	--------

Description. The normalize-and-transfer instruction removes the left-hand character of the storage field if the numerical part of that character is a zero.

A transfer is made to the location specified by the address part of the instruction when a zero is deleted. If the numerical part of the left-hand character is not a zero, the storage field is not changed, a transfer is not made, and the machine proceeds to the next instruction.

When the storage field consists of a single zero character, the zero is not deleted and a transfer is not made.

The normalize-and-transfer instruction is useful in removing zeros, one at a time, from the left end of a factor in accumulator and auxiliary storage. A program routine may then be inserted to count the number of zeros removed, a necessary function in floating decimal or floating dollar sign operations.

Program, Normalize and Transfer

Figure 38 illustrates the problem of printing asterisks in place of insignificant zeros in an amount field. Assume the following:

A six-position amount field in ACC 00 is to be stored for printing at memory address 05040. The dollar

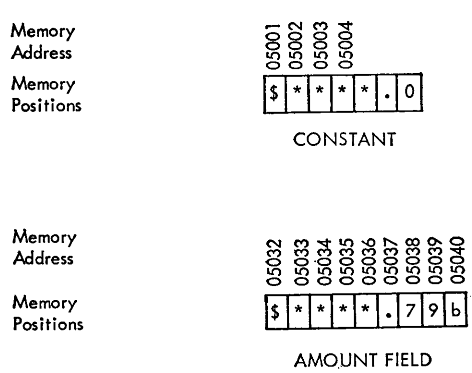


Figure 38. Normalize and Transfer

INSTR. LOCATION	INSTRUCTION OPER.	ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00004	SET	00007	07			axxxxxxx	+
00009	LOD	05007	07			a$****.0	+
	Main Program						
00014	SEL						
	RD						
00099							
	Prepare for Printing						
00104	UNL	05038	07				
00109	NTR	00109	00	a00079	+		
00114	SPR	05040	00	a79	+		
00119	Write Record						
00124							
00144	TR	00014					

Figure 39. Program, Normalize and Transfer

sign, asterisks to replace the insignificant zeros to the left of the decimal point, and the decimal point are to be placed in memory for proper printing of the amount. Constant factors include a dollar sign, asterisks, a decimal point, and the memory address where they are to be placed.

Figure 39 is the program for normalize and transfer.

Housekeeping

00004 and 00009. Load the print pattern into ASU 07.

Main program

00014 to 00099. Read in records and calculate.

Prepare for printing

00104. Reset the print area.

00109. Remove the insignificant zeros to the left of the decimal point from the amount field in ACC 00.

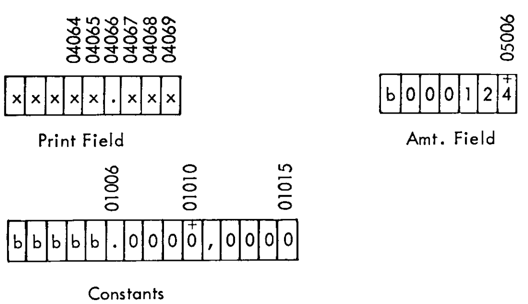
00114. Store amount for printing.

Print records and transfer to main program

00119 to 00144. Write record and repeat process.

Program, Floating Dollar Sign

A method of performing a floating dollar sign operation is shown in Figure 40. This method shows an-



INSTR. LOCATION	INSTRUCTION OPER.	ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN	EXPLANATION
Housekeeping								
	If WR is used to print total							
00004	SET	00006	07			axxxxxx		Clear print area
00009	LOD	01006	07			abbbbb.	+	and place
00014	UNL	04066	07					decimal point
	If WRE is used to print total							
00004	SET	00001	07			ax		Clear print area
00009	LOD	01006	07			a.	+	and place
00014	UNL	04066	07					decimal point
Main Program								
	Calculate Amount							
00504	SET	00005	00			axxxxx		Set No. pos left of dec in field
00509	LOD	01015	00			a,0000	+	
00514	RAD	01010	02			a0000	+	To limit NTR left of dec. pt.
00519	RAD	05006	01	a000124	+			Get amount field
00524	NTR	00534	01	a00124	+			Remove left zeros
00529	TR	00544						
00534	SHR	00001	00			a,000	+	Position comma
00539	NTR	00524	02			a000	+	Limit operation to left of dec. pt.
00544	SPR	04069	01					Put amt in print field
00549	ADM	04065	00					Comma + blank in memory = $ Sign
	NOTE: ADM in 00549 is to unsigned field. Both numerical and zone bits are added.							

Figure 40. Program, Floating Dollar Sign

other use of the NTR instruction that may be useful in some applications.

HOUSEKEEPING

00004, 00009, and 00014. Used to clear the print area and place the decimal point.

MAIN PROGRAM

00019 to 00499. Calculate amount.

PREPARE FOR PRINT

00504, 00509. Set the number of positions to the left of the decimal in amount field.

00514. To limit the NTR to the left of the decimal point.

00519. Get the amount field.

00524. Remove the left zeros.

00529. Transfer to store for print.

00534. Position the comma.

00539. Limit the operation to the left of the decimal point.

00544. Put the amount in the print field.

00549. (Comma and blank in memory = dollar sign.)

NOTE: The ADM at 00549 is to an unsigned field. Both numerical and zone bits are added.

Transfer on Plus (M—TRP)

Function. This operation interrogates the sign indicator of either accumulator or auxiliary storage and effects a transfer if the interrogated indicator is on.

Address. The address indicates which sign indicator is to be interrogated and the instruction location address to which the machine will transfer if the interrogated sign indicator is on.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	34	-------
705 III	22	-------

Description. The transfer-on-plus instruction causes a program transfer when the sign of accumulator storage or the sign of the auxiliary storage units is plus. The address part of the instruction specifies the memory location of the next instruction to be executed after the transfer. The address must also specify either accumulator storage (00) or any of the auxiliary storage units (01-15).

When a storage field consists of characters having zero numerical portions, the sign indicator is set to plus. Therefore, if a distinction is to be made between zero and plus, the transfer on zero must precede the transfer on plus.

NOTE: As a result of an incompleted division operation, the accumulator contents may be zero with the minus sign of the replaced dividend. See "Divide."

Transfer on Zero (N—TRZ)

Function. This operation interrogates the zero indicator of either accumulator or auxiliary storage and effects a transfer if the interrogated indicator is on.

Address. The address indicates which zero indicator is to be interrogated and the instruction location address to which the machine will transfer if the interrogated zero indicator is on.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	34	--------
705 III	22	--------

Description. The transfer-on-zero instruction causes a program transfer when the zero indicator of accumulator storage or of the auxiliary storage units is turned on. The accumulator zero indicator is turned on when the contents of accumulator storage consist of characters having zero numerical portions. The auxiliary storage unit's zero indicator is turned on when the contents of the last used unit consists of characters having zero numerical portions (except for a lengthen or round instruction of more than 254 positions). These characters are zero, plus or minus signed zero, and the record mark.

The address part of the instruction specifies the memory location of the next instruction to be executed after the transfer. The address must also specify either accumulator storage (00) or any of the auxiliary storage units (01-15).

Transfer (1—TR)

Function. This operation is used to unconditionally transfer to the next instruction to be executed.

Address. The address indicates the instruction location address to which the machine transfers.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	34	------
705 III	22	------

Description. The transfer instruction is used to change the sequence in which instructions of a program are executed. The address part of the instruction specifies the memory address of the right-hand digit of the next instruction to be executed. The instruction referred to by the address can be located at any point within the program. ASU coding 00-15 used with this instruction on a 705 I or II does not modify the operation. See the instruction transfer 01 in relation to the 705 III.

Transfer 01 (1—TSL) *(705 III)*

Function. The memory address of the next sequential instruction is placed in memory at an address indicated by MAC II and a transfer is made to a memory position as indicated by the address part of the instruction.

Address. The address indicates the instruction location address to which the machine transfers.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 III	67	-------

Description. A transfer instruction coded for ASU 01 will be interpreted as "transfer and store location" (TSL). When this instruction is executed, the five-digit instruction location address of the instruction counter is stepped by 5, converted to a four-digit memory address and placed into consecutive memory locations starting at the location specified by MAC II. The TSL instruction then effects a transfer to the location specified by the address part of the instruction.

For the TSL instruction to be executed properly, MAC II must indicate a memory address ending in 1 or 6. (See the instruction "receive" for the setting of MAC II). ASU coding other than 01 is interpreted as a transfer (1—TR) instruction.

With this instruction, a subroutine may be called into the program without making any changes in the subroutine instructions. Only two instructions are required in the main program, the receive and TSL (transfer and store location) instructions.

Assume that an error correction subroutine is located in memory locations 08004 to 08114, with a transfer instruction at 08114. As the result of a test routine, the main program requires the correction subroutine. The following steps would take place:

00994. Test instruction. Transfer to 01004.
01004. RCV 08111
01009. TSL 08004
01014. Proceed.
08004. First instruction. Continue to 08114.
08114. TR 01014

00994. As a result of the test routine, the program transfers to 01004.

01004. The RCV instruction places its address into MAC II. *This address must always end in a 1 or a 6 when used with a* TSL.

01009. During execution of the TSL instruction, the instruction counter (IC) is automatically increased by 5, making it 01014. The contents of the IC (01014) are then stored, in address form, at the memory location of MAC II (08111). The TSL instruction then transfers to the memory location specified by its address (08004).

08004. This is the first instruction of the subroutine. The rest of the routine is executed normally and, when 08114 is reached, this transfer instruction now has an address which is the re-entry point to the main program.

Transfer Any (1—TRA)

Function. This operation interrogates the transfer-any indicator and effects a transfer if the indicator is on.

Address. The address indicates the instruction location address to which the machine will transfer if the interrogated transfer-any indicator is on.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	34	-------
705 III	22	-------

Description. The transfer-any indicator is turned on whenever an input-output or check indicator is turned on. When the indicator is on, a transfer is made to the memory location specified by the address part of the instruction. The transfer-any indicator is turned off by the transfer itself.

The TRA instruction used with the 705 I or II may have ACC or ASU coding. When used with the 705 III, 00 coding must be used.

Transfer Any 01 (1—TAA) 02 (1—TAB) 03 (1—TAC) 04 (1—TAD) 05 (1—TAE) 06 (1—TAF) *(705 III)*

Function. This operation interrogates an addressed alteration switch and effects a transfer if the switch is on.

Address. The address indicates the instruction location to which the machine transfers if the interrogated alteration switch is on. The ASU coding 01-06 indicates which alteration switch is to be interrogated.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 III	22	-------

Description. The TRA instruction with ASU codes 01 through 06 makes possible the selection and interrogation of an alteration switch with only one instruction. The program transfers according to the setting of the various alteration switches as follows:

ASU CODE	ALTERATION SWITCH	MNEMONIC
01	00911	TAA
02	00912	TAB
03	00913	TAC
04	00914	TAD
05	00915	TAE
06	00916	TAF

If an interrogated alteration switch is off or ASU coding 07 through 15 are used, a NOP occurs. The transfer is not made and the machine proceeds to the next sequential instruction.

This instruction does not require the use of the select register; thus, it retains the previous selection whatever it may have been. For example, the following sequence of instructions can be given:

SEL	00201	Select a tape unit.
TAB	- - - - (02)	Transfer on 00912.
TRS	- - - -	Transfer on signal.

Transfer on Signal (O—TRS)

Function. This operation interrogates the last previously selected input-output indicator, alteration switch, or check indicator and effects a transfer if the selected indicator is on.

Address. The address indicates the memory address to which the machine transfers if the selected indicator is on.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	34	-------
705 III	22	-------

Description. When the TRS instruction is executed, the selected indicator is automatically turned off and the machine transfers to an instruction location specified by the address part of the TRS instruction. When the selected indicator is off, a TRS instruction has no effect and the machine continues to the next sequential instruction.

The TRS instruction with 00 zoning operates the same way in the 705 I, II, and III.

Transfer on Signal 01 (O—TRR)

Function. This operation interrogates the ready indicator of the selected IBM 777 Tape Record Coordinator, IBM 754 Tape Control or IBM 767 Data Synchronizer. A transfer is effected if the interrogated ready indicator is on.

Address. The address indicates the instruction location address to which the machine transfers if the interrogated ready indicator is on.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	34	-------
705 III	22	-------

Description. The TRS instruction with ASU 01 coding becomes a transfer-on-ready operation for use with the 777, 754, and 767. The TRR operation interrogates the ready indicator of the selected unit.

The ready indicator is on and a transfer effected if the selected tape unit has the ready light on and is not in the process of rewinding.

If a selected tape unit does not exist or has not been placed in ready status by the operator or is ready but in the process of rewinding, a transfer is not made and the machine proceeds to the next sequential instruction.

The 705 I and II may have any ASU coding to perform this operation. The 705 III must have 01 coding.

Transfer on Signal 02 (0–TTC) *(705 III)*

03 (0–TSA)
10 (0–TIC)
11 (0–TMC)
12 (0–TRC)
13 (0–TEC)
14 (0–TOC)
15 (0–TSC)

Function. This operation interrogates the particular indicator designated by the ASU coding and effects a transfer if the interrogated indicator is on.

Address. The address indicates the instruction location address to which the machine transfers if the interrogated indicator is on. ASU coding is used to indicate the particular indicator to be interrogated.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 III	22	-------

Description. The various indicators that can be interrogated by this operation are:

ASU		MNEMONIC	INDICATOR
02	(TTC)	Transfer Transmission Check	PCT Data Check
03	(TSA)	Transfer Synchronizer Any	PCT or I-O (Tape)
10	(TIC)	Transfer Instruction Check	Instruction Check (00900)
11	(TMC)	Transfer Machine Check	Machine Check (00901)
12	(TRC)	Transfer Read-Write Check	Read-Write Check (00902)
13	(TEC)	Transfer Echo Check	Record Check (00903)
14	(TOC)	Transfer Overflow Check	Overflow Check (00904)
15	(TSC)	Transfer Sign Check	Sign Check (00905)

When codes 02 and 03 are used, the DS must be previously selected. The TRS 03 has a built-in ready check; thus, it is not necessary to use a transfer ready routine to determine if the DS is ready.

When the TRS 10-15 operations are used, the interrogated indicator, if on, will be turned off and a transfer effected. These operations may be used without the use of a prior selection. Thus, the select register retains the previous selection whatever it may have been.

For example, where a tape had been previously selected, the synchronizer check indicator or the read-write check indicator could be interrogated and a transfer made to a common error routine without the necessity of remembering, through programming, which tape had been selected.

The new modes of operation for both the TRS and TRA instructions make possible the wider use of common and simple subroutines for error correction, end of file, and so on. For example, the main routine of a program might make the following checks at one of several places in the program.

SEL	00200	Select tape unit.
TSA	- - - (03)	See if record has completely read in.
TTC	- - - (02)	Check accuracy.
TRS	- - - (00)	Check end of file.

If the check indicator is on, the program transfers to a common check routine where the instructions can act upon the last selected tape unit, which in this case is 00200. If the transfer had been from a different place in the program where 00203 had been used, these instructions would refer to 00203.

ASU coding of 04 or 09 used with this instruction results in a NOP.

Set Bit 01-06 (%–SBZ) *(705 III)*

07 (%–SBA)
08 (%–SBR)
09-14 (%–SBN)

Function. This operation changes any bit in an addressed character to the 0 or 1 state or reverses the status of the A bit or C bit.

Address. The address indicates the character location. The ASU coding identifies the particular bit and the type of operation.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 III	31	-------

Description. The set-bit-zero (SBZ) operation sets any bit position of an addressed character (except C bit) to a zero bit status. ASU zoning for the various bits of the character are as follows:

	ASU	ZONING
	01	Set 1 bit to 0
	02	Set 2 bit to 0
SBZ	03	Set 4 bit to 0
	04	Set 8 bit to 0
	05	Set A-bit to 0
	06	Set B-bit to 0

If the addressed character bit position is already in a zero bit status, no change is effected. The C bit of the character will be automatically corrected for consistency with the character code. For example:

	Before							After						
	C	B	A	8	4	2	1	C	B	A	8	4	2	1
SBZ 03	1	1	1	0	1	0	0	0	1	1	0	0	0	0
SBZ 02	1	1	1	0	1	0	0	1	1	1	0	1	0	0

The set-bit-alternate (SBA) operation reverses the bit status in the A bit position of the addressed character. The ASU coding used for this operation is 07. The C bit is automatically adjusted for consistency with the character code. For example:

	Before							After						
	C	B	A	8	4	2	1	C	B	A	8	4	2	1
SBA (07)	1	1	1	0	1	1	1	0	1	0	0	1	1	1
SBA (07)	0	1	0	1	0	1	1	1	1	1	1	0	1	1

The set-bit-redundant (SBR) operation reverses the bit status in the C bit position of the addressed character. The ASU coding used for this operation is 08. The C bit is not adjusted for consistency with the character code. This instruction may be used to develop an invalid character in memory. For example:

	Before							After						
	C	B	A	8	4	2	1	C	B	A	8	4	2	1
SBR (08)	1	0	1	0	0	0	0	0	0	1	0	0	0	0
SBR (08)	0	0	1	0	1	1	1	1	0	1	0	1	1	1

Set bit to one (SBN) sets any bit position of an addressed character (except C bit) to a 1 bit status. ASU zoning for the various bits of the character are as follows:

	09	Set 1 bit to 1
	10	Set 2 bit to 1
SBN	11	Set 4 bit to 1
	12	Set 8 bit to 1
	13	Set A-bit to 1
	14	Set B-bit to 1

If the addressed character bit position is already in a 1 bit status, no change is effected. The C bit of the character is automatically corrected for consistency with the character code. For example:

	Before							After						
	C	B	A	8	4	2	1	C	B	A	8	4	2	1
SBN (10)	1	0	0	0	0	0	1	0	0	0	0	0	1	1
SBN (12)	0	0	1	0	1	0	0	1	0	1	1	1	0	0

When using the set bit instruction in setting up code characters or a number of variables, there are three things to consider:

1. If six separate variables are to be recorded in a single character, there must not be the possibility of their all being zero at the same time. This would result in placing a storage mark in memory. Storage mark is an invalid character in memory. If this possibility exists, only five bits of the character may be used for the recording of variables and the sixth is set permanently to 1.
2. If the code character is to be written out of memory by means of the WR 00 instruction, there must not be the possibility of all the bits being at 1 since this would constitute a group mark and would terminate a WR 00 instruction. To prevent this, the programmer could use the WR 01 instruction, or plan to use only four bits for the recording of variables, with the fifth set permanently at 1 (to insure against all 0's) and the sixth set permanently at 0 (to insure against all 1's).
3. If the code character is to be written out onto a printer, the programmer must insure that the combination of bits does not constitute some character that is not printable.

If the set bit instruction is coded for ACC or ASU 15, a NOP occurs.

Transfer on Zero Bit 01-07 (.—TZB) (705 III)

Function. This operation interrogates any bit position of an addressed character and effects a transfer if the interrogated bit is zero.

Address. The address indicates the instruction location to which the machine transfers if a transfer is effected.

The ASU coding of the instruction indicates the specific bit of the character to be tested.

The memory address of the character to be interrogated is determined by a RCV instruction which must precede the TZB.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 III	31	-------

Description. This instruction is preceded by a receive instruction which sets MAC II at the address of the subject character in memory. The address portion of the TZB instruction indicates where the program should transfer if the bit tested is a 0. The ASU zoning of the instruction indicates the specific bit of the character that is to be tested; thus:

BIT	ASU ZONING
1	01
2	02
4	03
8	04
A	05
B	06
C	07

ACC coding and ASU codes other than 01-07 are ignored and a NOP occurs.

If the bit tested is 0, the program will transfer; if the bit is 1, the program will proceed to the next instruction. Therefore, to test the A bit of the character in location 34660 of memory, the programming would be:

```
10004 RCV 34660
10009 TZB 20004 (05)
10014 ---
```

If the A bit of character 34660 is 0, the program transfers to location 20004 for its next instruction. If the A bit is 1, the program proceeds to 10014 for its next instruction.

During the execution of the TZB instruction, the character designated by MAC II (the RCV instruction) is placed in CR 1 where the bit indicated by the ASU zoning is interrogated. MAC II remains set until another RCV or a RWW instruction is given.

By testing a bit individually, it is possible to determine the sign of any value stored in memory without first moving it to the ACC or ASU, as follows:

```
RCB --           Units position of field in question.
TZB --- (05)     Transfer if field is minus.
    --- --       No transfer if field is plus.
```

The TZB need only look at the A bit, because the B bit must always be a 1 if the field is always signed.

If the A bit is 0, the field is minus; if it is 1, the field is plus.

If it is necessary to determine which, if any, zone bits a character has, it can be done in the following manner:

```
00104  RCV ---          Character in question
00109  TZB --- (06)     00124
00114  TZB --- (05)     Zone 10
00119  TR  --- --       Zone 11
00124  TZB --- (05)     Zone 00
       --- --- --       Zone 01
```

To reverse a bit from one state to another, regardless of its previous state (as might be done to alternate between two numbers or addresses) the following routine could be used.

```
00104  RCV        01004   Designate character.
00109  TZB  (01)  00204   Transfer if the bit is 0.
00114  SB   (01)  01004   If bit is not 0, set it to 0.
00119  TR         00304   Continue with program.
00204  SB   (09)  01004   If bit is 0, set it to 1.
00304  ---         ----   Continue with program.
```

No Operation (A—NOP)

Function. This operation causes the machine to proceed to the next instruction in the program.

Address. The address can be any address and has no effect for the 705 I and II. In the 705 III, an indirect-zoned address part that does not end in a 4 or 9 causes an instruction check indication (00900).

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	34	-------
705 III	22	-------

Stop (J—HLT)

Function. Execution of this instruction stops the CPU operation.

Address. The address can be any address and has no effect.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS
705 I, II	34	-------
705 III	22	-------

Description. Several stops may be included in a program for the convenience of the operator. An error in reading or writing, an end-of-file condition, or various other situations may be programmed to stop or "halt" operation. The address part of the stop instruction can be read from the console when a stop occurs.

The address may be coded to indicate to the operator why the machine operation has been interrupted. Depressing the start key on the 782 Console causes the machine to read and execute the next instruction.

Control Operations

The control instructions control various features of the input-output units and turn on and off the input-output indicators. The control instructions related to the IBM 760 Control and Storage and the IBM 777 Tape Record Coordinator are listed in the Appendix of this manual. These operations are discussed in the manuals related to the particular machine.

Control 00000 (3—IOF)
Control 00001 (3—WTM)
Control 00002 (3—RWD)
Control 00003 (3—ION)
Control 00004 (3—BSP)

Control 00004 (01) (3–BSF) 705 III
Control 00005 (3–SUP)
Control 00009 (3–SKP) 705 III

Function. These operations control various features of the selected input-output units and turn on or off the input-output indicator. Use of these instructions with the DS requires that the DS be selected and ready.

NOTE: Refer to the section "Machine Components" for further details of these instructions related to the DS.

Address. The address specifies the feature to be controlled.

Limiting Factors. None.

Timing

MODEL	TIME (μs)	REMARKS	
705 I, II	68	Ctrl 00000	IOF
	10000	Ctrl 00001	WTM
	51	Ctrl 00002	RWD
	51	Ctrl 00003	ION
	60000 + 67*N*	Ctrl 00004	BSP
	6000	Ctrl 00005	SUP
705 III	49	Ctrl 00000	IOF
	49	Ctrl 00001	WTM
	33000	Ctrl 00002	RWD
	49	Ctrl 00003	ION
	49	Ctrl 00004	BSP
	49	Ctrl 00004 (01)	BSF
	6000	Ctrl 00005	SUP
	49	Ctrl 00009	SKP

Control 00000 (3—IOF)

The input-output indicator of the unit previously selected, if on, is turned off. This instruction refers to printers, tape units, drums, and card readers.

Control 00001 (3—WTM)

A tape mark is written on tape by the last selected unit. The writing of this special character is checked in the same way as the writing of characters from memory.

Control 00002 (3—RWD)

The tape on the last unit selected is rewound.

Control 00003 (3—ION)

The input-output indicator on the *tape unit* last selected, if off, is turned on. The instruction may be used for tape units only.

Control 00004 (3—BSP)

The tape on the unit last selected is backspaced to the previous inter-record gap. Backspacing tape at load point turns on the tape indicator.

Control 00004 01 (3—BSF) 705 III (DS)

Tape is backspaced until a tape mark or load point is reached. The tape indicator is not turned on when a tape mark is read, unless the load point is reached. This instruction cannot be used under RC control and must have ASU coding of 01. ASU coding other than 01 is interpreted as 00.

Control 00005 (3—SUP)

This instruction applies to printers and punches only. It prevents printing or punching of information from record storage for one cycle. The instruction is normally used to prevent printing or punching when a read-write error has occurred from memory to record storage. Under program control, the record storage can be reloaded from memory after an error condition has been recognized and the corrected record printed or punched.

Control 00009 (3—SKP) (705 III)

When preceded by a select instruction which selects a tape unit, the skip instruction causes the tape to skip forward approximately five or six inches. During the skip, the tape is erased as it passes over the read-write head.

The instruction is intended for use with the 729 tape unit with the two-gap head. If a write error persists after two or three attempts, the tape may be backspaced once more and skipped over what can be presumed to be an imperfection in the tape itself.

End-of-File and Checking Procedures

End-of-File Procedures

Each input or output unit in the 705 system, except the card punch and typewriter, is equipped with an indicator to signal an end-of-file condition. Whenever a unit is selected by a program instruction, the input-output indicator associated with that unit is also automatically selected. The instruction SEL 00200, therefore, not only selects a tape unit, but also selects the input-output indicator for that unit.

An indicator is either on or off. It may be turned on by any of the conditions listed under "End-of-File and Checking Procedures." Once an indicator is turned on, it remains on until it is turned off either by the program or by a manual operation.

The status of an indicator is tested or interrogated by a transfer-on-signal instruction in the program. This instruction usually follows immediately after reading or writing operations. When an indicator is turned on by an end-of-card-file, end-of-tape, or other condition, a transfer is made to an instruction location specified by the transfer-on-signal instruction. End-of-file instructions are normally included as subroutines in the program. When the indicator is off, a transfer on signal has no effect and the machine continues to the next instruction of the main program.

The end-of-file subroutine or branch program may be arranged in a variety of ways, depending upon operating conditions. For example, the typewriter may be used to notify the operator that a tape unit is in end-of-file condition. The machine may then be programmed to stop while reels are changed, or to select automatically an alternate unit and continue operation. Other control instructions may automatically rewind a completed reel.

By prearranged manual switch settings, the operator can, after putting in the last group of cards, make the machine automatically select program instructions to continue operation after an end of file has been signaled. Such operation might include final total calculation and printing.

When two or more tape units are being used to read or write a single file on multiple reels, an end-of-file signal on the first reel can change the select instruction address to specify the second tape unit. Reading or writing can continue on the second reel without loss of operating time while the first reel is automatically rewound. Reading can continue alternately between two units until the installation of the last reel is noted by alteration switch setting.

When several related records are processed in or out of the system during the same procedure, an end of one input record file can cause program modification to consider only those records remaining to be processed. Calculation or reading steps for the completely processed records are then ignored.

Tape Unit Indicator

Each tape unit is provided with an input-output indicator to indicate the end of a reel or file. The indicator can be turned on by any of the following:

1. Sensing the end-of-reel marker while writing.
2. Sensing the tape mark as a unit record while reading.
3. Using an ION instruction in the program when a tape unit was last selected.

The indicator can be turned off by:

1. Depressing the unload key on the tape unit to remove the reel.
2. An IOF instruction in the program when a tape unit was last selected.

Card Reader Indicator

Each reader is provided with an input-output indicator. This indicator is turned on when a read instruction involving this particular reader is given after the last card has been read from record storage. The indicator is turned off by:

1. Loading record storage by feeding cards.
2. An IOF instruction when the reader was last selected.

Printer Indicator

Each printer is provided with an input-output indicator to indicate the end of a printed page. It is turned on by the overflow signal obtained from channel 12 of the carriage tape. The indicator is turned on after the next write instruction involving that printer is given. The indicator is turned off by:

1. Using an IOF instruction in the program when a printer was last selected.
2. Depressing the printer start key.

Drum Indicator

The input-output indicator of a drum is turned on if an attempt is made to read or write beyond the limits of the drum. The indicator is turned off by using an IOF instruction in the program when a drum section was last selected.

Program, End of File, Input and Output Tapes

The problem illustrated in Figure 41 represents a simple end-of-file procedure using one input tape unit and one output tape unit. The input records are read into memory in the area shown and the total cost field is computed. When an end-of-file condition is signaled on either tape, a transfer to a branch program is made where the proper reel is rewound, the tape mark is written, the operator is notified by typed instructions, and the machine is stopped to permit a reel change. A constant area is shown where the typed messages with group marks have been stored during program loading. It is assumed that there is only one reel of input tape. However, because the output record is longer than the input, the output tape might require two reels.

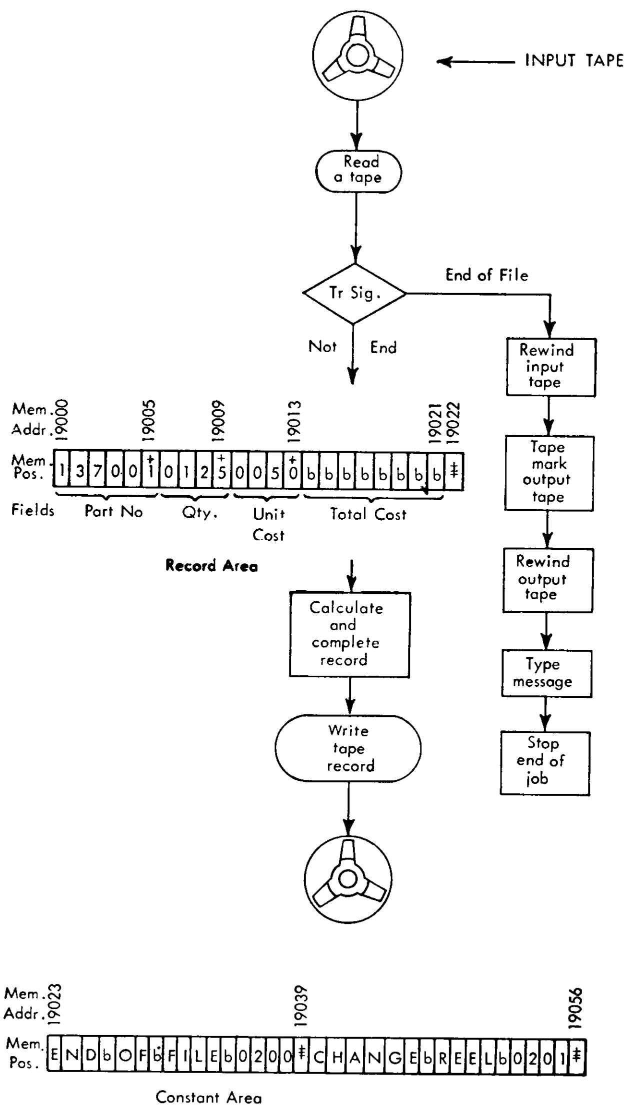


Figure 41. Tape, End of File

Figure 42 is the program for end-of-file on input and output tapes.

00004. Adjust ASU 01 to one position.
00009. Get group mark.
00014. Put group mark at end of record.
00019. Select input tape 00200.
00024. Read the tape record.
00029. Transfer on signal. A transfer is effective when a tape mark is read as the last record from the input tape. The machine transfers to the end-of-file subroutine, beginning at location 00069.
00034. Reset and add the quantity field into accumulator 00.
00039. Multiply quantity by unit cost to get total cost.
00044. Store total cost in the record at address 19021.
00049. Select output tape 00201.
00054. Write out the record on tape.
00059. Transfer on signal. A transfer is effective when a reflective spot is sensed during a write operation. This may happen when the output unit runs out of tape before the input unit is at end of file.
00064. Transfer to repeat the program.
00069. Rewind the input tape. This instruction is executed when the transfer on signal has been activated by a tape mark on the input tape. In this case, all records in the file have been read.
00074. Select the output tape.

INSTR. LOCATION	INSTRUCTION OPER.	INSTRUCTION ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00004	SET	00001	01			ax	+
00009	LOD	19039	01			a‡	+
00014	UNL	19022	01				
00019	SEL	00200					
00024	RD	19000					
00029	TRS	00069					
00034	RAD	19009	00	a0125	+		
00039	MPY	19013	00	a00006250	+		
00044	ST	19021	00				
00049	SEL	00201					
00054	WR	19000	00				
00059	TRS	00104					
00064	TR	00019					
00069	RWD						
00074	SEL	00201					
00079	WTM	00001					
00084	RWD						
00089	SEL	00500					
00094	WR	19023	00				
00099	HLT	00001					
00104	WTM						
00109	RWD						
00114	SEL	00500					
00119	WR	19040	00				
00124	HLT	00002					
00129	TR	00019					

Figure 42. Program, End of File

00079. Write a tape mark to record the end of file on the output tape.

00084. Rewind the output tape.

00089. Select the typewriter.

00094. Write out the message that the input tape is at end of file.

00099. Stop the machine for the end of the job.

00104. Write a tape mark on the output tape when a reflective spot has been sensed while writing the output record.

00109. Rewind the output tape.

00114. Select the typewriter.

00119. Write a message to the operator to change reels on the output tape unit after machine stops.

00124. Stop the machine to permit the operator to change reels.

00129. After the reel change, depressing the start key transfers the machine to repeat the program and continue operation.

PROGRAM, END OF FILE, READ WHILE WRITING

Figure 43 is a flow chart of the end-of-file routines related to a balance forward program. It illustrates the end-of-file procedure for read-while-writing operations.

Assume that the input-output units used are one card reader, one input and one output tape unit. The input tape file may be on more than one reel; therefore, the output tape is also on multiple reels.

When either input or output tape units sense the end of file, the operator is properly notified, the machine is stopped for reel change, and normal operation continues after the change. When the operator loads the last input reel, an alteration switch is set accordingly, and the machine transfers to an end-of-file routine and stops for the end of job.

Figure 44 is the program for end-of-file shown in Figure 43. The program is written as added instructions to a balance forward program.

00164. Select the input tape.

00169. Prepare to read while writing.

00174. Select the output tape.

00179. Write the output tape; read the input tape.

00184. Transfer on end-of-file signal from the output tape. The transfer is effective only when the end of file is sensed on the output tape because 00201 was the last unit selected.

00189. To test for end of file on input tape, unit 00200 is reselected.

00194. Transfer if an end-of-file signal was sensed from input tape during the read-while-writing operation.

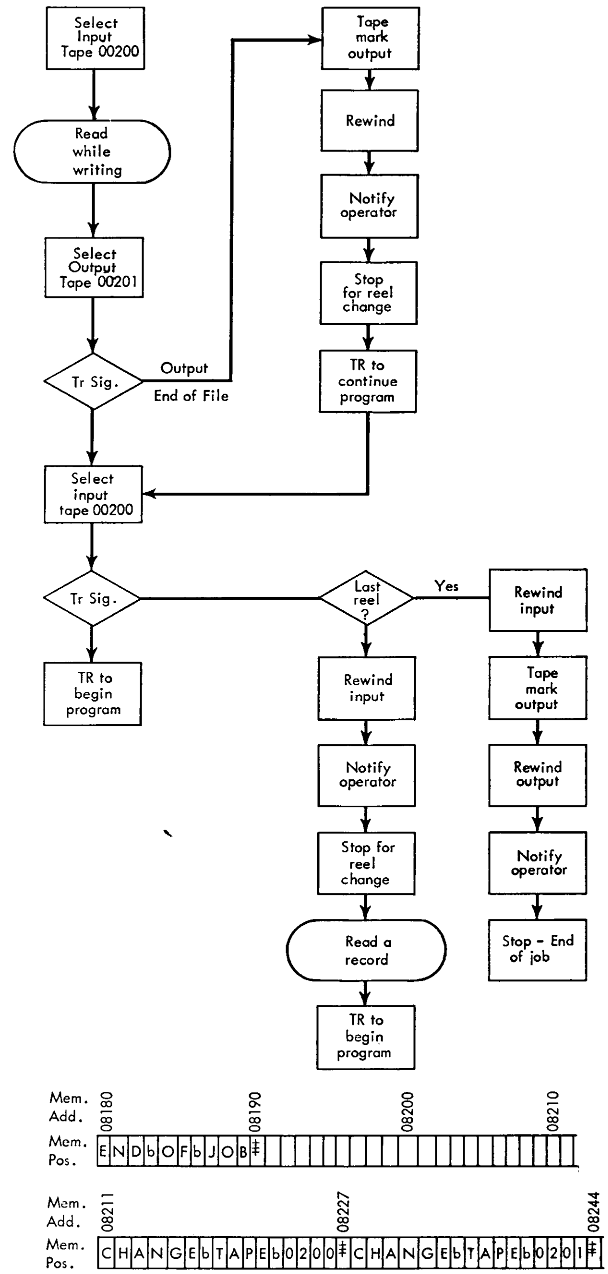


Figure 43. Flow Chart, End of File

00199. No end of file; transfer to repeat the program.

END OF OUTPUT FILE

00204. Write a tape mark on the output tape.

00209. Rewind the output tape.

00214. Select the typewriter.

00219. Write a message to the operator; change the reel.

00224. Stop while the reel change is made.

00229. Transfer to continue the program (Figure 44).

INSTR. LOCATION	INSTRUCTION OPER.	ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00164	SEL	00200					
00169	RWW	08047					
00174	SEL	00201					
00179	WR	08112					
00184	TRS	00204					
00189	SEL	00200					
00194	TRS	00234					
00199	TR	xxxxx	(Transfer to repeat program.)				
	End of Output File						
00204	WTM						
00209	RWD						
00214	SEL	00500					
00219	WR	08228					
00224	HLT	00001					
00229	TR	00189					
	End of Input File						
00234	SEL	00911					
00239	TRS	00284					
00244	SEL	00200					
00249	RWD						
00254	SEL	00500					
00259	WR	08211					
00264	HLT	00002					
00269	SEL	00200					
00274	RD	08047					
00279	TR	xxxxx	(Transfer to repeat program.)				
	End of Job						
00284	SEL	00200					
00289	RWD						
00294	SEL	00201					
00299	WTM						
00304	RWD						
00309	SEL	00500					
00314	WR	08180					
00319	HLT	00003					

Figure 44. Program, End of File

END OF INPUT FILE

00234. Select alteration switch 00911.

00239. Transfer if the switch is on, indicating that this is the end of the last reel of input.

00244. Select the input tape.

00249. Rewind the input tape.

00254. Select the typewriter.

00259. Write a message; change the reel.

00264. Stop while the reel change is made.

00269. Select the input tape.

00274. When the end of file is sensed, a tape mark is read as a unit record. A single read instruction must be given to read in the next record for processing.

00279. Transfer to repeat the program.

END OF JOB

00284. Select input tape unit 00200.

00289. Rewind the input tape.

00294. Select output tape unit 00201.

00299. Write a tape mark.

00304. Rewind the output tape.

00309. Select the typewriter.

00314. Write the end-of-job message.

00319. Stop for end of job.

Checking Procedures

Check Indicators

Accuracy is an essential requirement of the data processing system. To meet this requirement, the 705 provides various checking devices. The programmer can use these devices at his own discretion. Two types of checks may be made:

1. Checks upon the reading and writing of data by input-output units. (Refer to the particular machine component under consideration.)
2. Checks upon the handling of data within the machine, including the check for legitimate instructions, overflow, sign and character coding.

The 705 has six check indicators to provide the operator with a check on the accuracy of data being processed. These indicators are associated with separate switches on the operator's console.

In many cases, it is not necessary to interrupt machine operation when an error condition is detected. The programmer can include special branch programs to handle certain types of errors as exceptions. An error in reading a record from tape, for example, may be programmed to backspace the tape and re-read the record. If a correct reading is obtained the second time, normal machine operation continues. If the error persists, machine operation can be interrupted or the incorrect record can be noted and operation continued.

The indicator switch setting on the operator's console gives the choice of programming around an error or stopping the machine. When a switch is set to AUTOMATIC, the error detected by the corresponding check indicator causes an automatic machine stop. To resume operation, the indicator may be turned off by depressing the start key on the console. When a switch is set to PROGRAM, the corresponding check indicator may be interrogated during the program and an error does not automatically stop the machine. The particular instruction during which the error is detected is executed and the machine proceeds to the next instruction.

A check indicator is interrogated by two instructions: select, followed by transfer on signal. The select instruction specifies the proper indicator. The transfer-on-signal address transfers the program to the first instruction of a subroutine which is to be followed if an error is detected. The transfer is made only when the indicator has been turned on by an error condition. Machine operation is not interrupted when the error is corrected by the branch program. The transfer-on-signal instruction turns the indicator off.

Check indicators and their assigned addresses are:

Instruction Check Indicator	00900
Machine Check Indicator	00901
Read-Write Check Indicator	00902
Record Check Indicator	00903
Overflow Check Indicator	00904
Sign Check Indicator	00905

Instruction Check Indicator 00900

The instruction check indicator turns on when the following conditions occur:

1. A character code error is detected during instruction time.
2. An invalid operation part is encountered in the operation register.
3. The operation part is incorrectly interpreted.
4. The units position of the address part of any transfer instruction, or a transmit instruction specifying accumulator 00, is not 4 or 9. The send instruction is also checked on the 705 III.
5. The field addressed by an indirect address coded instruction is not a position ending in 4 or 9. With the switch set to AUTOMATIC, the machine stops during the character cycle in which the error occurred.

Machine Check Indicator 00901

The machine check indicator is turned on when a character code error is detected during the execution of all instructions (except read) in which data are transferred from accummulator or auxiliary storage or memory. These instructions include:

ADD	MPY	ST	WR	SET
SUB	DIV	SPR	WRE	SHR
RAD	LOD	ADM	SGN	RND
RSU	UNL	TMT	CMP	TZB
SB	ULA	LDA	AAM	NTR
				SND

When the indicator switch is turned to AUTOMATIC, the machine stops during the character cycle in which the error occurred except if an error occurs during the execution of write or write and erase. In this case, the indicator will be turned on but no automatic stop will occur. Such an error may be detected by programming, as the read-write check indicator will also be turned on.

Read-Write Check Indicator 00902

The read-write indicator turns on when a character code error is detected during the execution of a read, write, read-while-writing, or write-and-erase instruction except when using the DS. The indicator also turns on when an error is detected in reading the holes in the card or by the longitudinal check in tape reading. The indicator, therefore, checks the transmission of data from all input units to memory. It also checks the transmission of all output data from memory to the drum, tape unit, card punch record storage, printer record storage, and typewriter. The indicator turns on if an attempt is made to read or write beyond the limits of the drum or if an error occurs in recording a tape mark.

When the indicator switch is turned to AUTOMATIC, an error stops the machine after the instruction is executed.

Record Check Indicator 00903

The record check indicator turns on when an error is detected by the brush-compare method on the punch and by the echo-check method on the printer. An error in card punching is detected as the card passes a brush station after it has been punched. If an error occurs, the record check indicator turns on during the execution of the next write or write-and-erase instruction to that card punch.

An error in printing is detected by sensing the position of each print wheel during the print cycle. If an error occurs, the indicator turns on during the execution of the next write or write-and-erase instruction involving that printer.

In both cases, when the switch for this indicator is on AUTOMATIC, an error stops the machine at the end of the punching or printing cycle during which the indicator was turned on. At this time, the error card is the last card to go into the punch stacker. The incorrect line of printing immediately precedes the last printed line.

Overflow Check Indicator 00904

The overflow check indicator is turned on during an add or subtract operation when the number of digits in the result is greater than the number of digits in the longer of the two fields. An overflow is indicated as a result of a round operation, if a carry-over is made out of the high-order position of the accumulator storage field.

The indicator is turned on by a divide instruction when the divisor does not have a greater absolute value than an equal number of digits taken from the left end of the dividend. When the error switch for this indicator is turned to AUTOMATIC, an error stops the machine during the execution of the instruction.

Sign Check Indicator 00905

The sign check indicator turns on if a field addressed by a reset and add, add, reset and subtract, subtract, multiply, or divide instruction does not have plus or minus zoning over the right-hand digit.

When the switch for this indicator is on AUTOMATIC, an error stops the 705 I or II during the character cycle following the one in which the error was detected. For 705 III operation, the sign check occurs and stops the machine in the same cycle in which the error is detected.

Program, Error Correction

The record shown in Figure 45 is read from tape. After each read operation, a check is made for a read-write error. If an error is discovered, the tape is backspaced and the record is reread. If the error persists, the tape is backspaced and read a third time. The 705 stops if the error occurs a third time.

Salesmen's commissions are computed when records are read correctly and the completed record is written on tape. Assume that constants are in memory as shown. Figure 46 is the program.

00004. Adjust ASU 01 to one position.
00009. Get group mark.
00014. Put group mark at end of record.

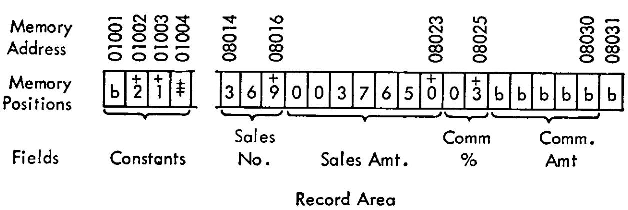


Figure 45. Error Correction

INSTR. LOCATION	INSTRUCTION OPER.	ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00004	SET	00001	01				
00009	LOD	01004	01			aǂ	
00014	UNL	08031	01				
00019	RAD	01002	02			a2	+
00024	SEL	00200					
00029	RD	08014					
00034	SEL	00902					
00039	TRS	00084					
00044	RAD	08025	00	a03	+		
00049	MPY	08023	00	a000112950	+		
00054	RND	00002	00	a0001130	+		
00059	SET	00005	00	a01130	+		
00064	ST	08030	00				
00069	SEL	00201					
00074	WR	08014	00				
00079	TR	00019					
		Correction Routine					
00084	SEL	00200					
00089	BSP						
00094	SUB	01003	02			a1	+
00099	TRP	00029	02				
00104	HLT	00001					

Figure 46. Program, Error Correction

00019. Reset and add a 2 in ASU 02.
00024. Select input tape 00200.
00029. Read the tape record into memory beginning at address 08014.
00034. Select the read-write check indicator.
00039. Test the condition of the indicator by a transfer-on-signal instruction. If the indicator is on, an error in reading has occurred and a transfer is made to the correction routine in the program.
00044. Reset and add commission percentage.
00049. Multiply percent by sales amount to get commission amount.
00054. Round to adjust the commission amount to the nearest cent.
00059. Adjust the commission amount to five positions.
00064. Store commission amount.
00069. Select output tape 00201.
00074. Write the completed record.
00079. Transfer to repeat the program.

Correction routine

00084. Select input tape 00200.
00089. Backspace the input tape.
00094. Subtract 1 from ASU 02 to count the reread.
00099. Transfer to reread when ASU 02 is plus.
00104. ASU 02 is not plus. The record has been read three times. Stop the machine.

Program, Error Correction with End of File

Figure 47 illustrates a program written to provide for end-of-file condition and read-write checking when processing either input or output records.

Assume that a tape record is read into memory locations 19000-19105 inclusive (Figure 48). A check is made for both end of file and read-write error. When the end of file is sensed, the input tape is rewound and the machine stops. When an error occurs in reading, the record is read twice more. When the error persists, machine operation stops.

After calculation (omitted from the program), the completed record is written on the printer. Writing is checked for a read-write error when loading record storage and for a record error from record storage to the print wheels. Machine operation is stopped after three errors have occurred while loading record storage or it is stopped for any record error.

00004. Adjust ASU 01 to one position.
00009. Get a group mark.
00014. Put a group mark in the record.
00019. Get +2 in ASU 01.
00024. Get +2 in ASU 02.
00029. Select the input tape.

INSTR. LOCATION	INSTRUCTION OPER.	ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00004	SET	00001	01				
00009	LOD	17003	01			a‡	+
00014	UNL	19106	01				
00019	RAD	17002	01			a2	+
00024	RAD	17002	02			a2	+
00029	SEL	00200					
00034	RD	19000					
00039	TRA	00429					
00044	↓						
00404	Main	Routine					
00409	SEL	00400					
00414	WR	19000					
00419	TRA	00489					
00424	Main	Routine					
		Input End of File and Error Correction					
00429	SEL	00902					
00434	TRS	00449					
00439	SEL	00200					
00444	TRS	00474					
00449	SEL	00200					
00454	BSP						
00459	SUB	17001	01			a1	+
00464	TRP	00034	01				
00469	HLT	00001					
00474	RWD						
00479	HLT	00002					
00484	TR						
		Output End of File and Error Correction					
00489	SEL	00901					
00494	TRS	00529					
00499	SEL	00902					
00504	TRS	00539					
00509	SEL	00903					
00514	TRS	00564					
00519	SEL	00400					
00524	TRS	To end-of-page routine					
00529	HLT	00003					
00534	TR	00409					
00539	SEL	00400					
00544	SUP						
00549	SUB	17001	02			a1	+
00554	TRP	00414	02				
00559	HLT	00004					
00564	HLT	00005					

Figure 47. Program, End of File and Error Correction

00034. Read the record.

00039. Transfer when any indicator has been turned on during a read operation.

00044. Continue with main routine of program.

00404. This is the last step of the main routine.

00409. Select the printer.

00414. Write the completed record on the printer.

00419. Transfer when any indicator has been turned on during a write operation.

00424. Continue with program.

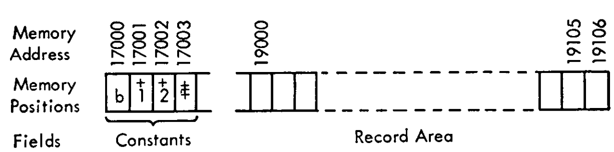


Figure 48. End of File and Error Correction

Input end of file with error correction

00429. Test for read-write error.

00434. With 00902 on, transfer to reread input record.

00439. Reselect input tape unit.

00444. Test for end of file and transfer if input-output indicator is on.

00449. Reselect input tape.

00454. Backspace one record.

00459. Count record backspaced.

00464. Reread record if counter is plus.

00469. Stop if counter is minus.

00474. Rewind input tape.

00479. Stop for reel change or end of job.

00484. Transfer to read first record of new reel or to end-of-job routine.

Output end of file with error correction

00489. Test for machine error.

00494. With 00901 on, transfer to stop for correction.

00499. Test for read-write error.

00504. When 00902 is on, transfer to rewrite output record.

00509. Test for record error.

00514. With 00903 on, transfer to stop for correction.

00519. Reselect printer.

00524. Test for end of page. If input-output indicator is on, transfer to end-of-page routine.

00529. Machine error. Stop for manual control.

00534. If error is corrected, transfer to rewrite. If record cannot be corrected, transfer to reread and recompute.

00539. Reselect the printer.

00544. Supress printing.

00549. Count one error.

00554. Transfer to rewrite if counter is plus.

00559. Stop if counter is minus.

00564. Stop for record error.

Systems Checking

Systems checks can be defined as any checks other than those made by the built-in check circuits in the 705. This is a broad category and includes such programming checks as record counts, hash totals, control totals, proof figures, limit checks, and cross-footing balance checks. All of these checks can be programmed and are useful tools for program checking.

The systems checks which are to be incorporated in a program should be designed during the original planning phase. What kind of systems check to use depends upon the program to be checked. Systems checks designed for a specific program generally are unique to that program. Some general techniques applicable to any program are described here.

Magnetic Label

File identification placed at the beginning of a reel of tape is referred to as a magnetic label. The label may specify the job title and/or number, date of last processing, number of the reel and so on. A label may also be placed at the end of the reel to designate the end of the file. The labels are read into memory at the beginning and end of the program as an added control that the proper records have been processed. The label may also insure a true end of file or end of job.

A counter may be included as part of the label to record the number of passes to which the tape has been subjected. Old tape may consequently be retired from active files before excessive wear has occurred.

Record Count

A record count is simply a count of the number of records in a file. This count is made each time the file is written and is carried as an additional record at the end of the file. The count is made again when the file is being read for processing to see that all records in the file have been read in.

Hash Total

A hash total is a total of an important numerical or alphamerical field (such as part number) for all records. It checks that all of the records written on the last processing run have been read in during the present cycle. It is similar to a record count, except that the hash total gives an additional check that all part numbers have been read in correctly. The hash total is carried as an additional record at the end of the file. This total may be computed as the original tape is written, or during a subsequent machine run.

The hash total may also be computed for certain vital fields in a single record. This total is carried as an additional field in each record and can be checked whenever that record is read into memory. Hash totals must be accumulated by use of the add-to-memory instructions if alphamerical fields are part of the total.

Control Total

A control total is a predetermined total of some amount or quantity field in a file of records. During the processing, a sum of this field is accumulated and checked against the control total. The control total can be in the form of a grand total for all input data, or an intermediate or minor total for each control group in the file. An example of the use of control totals is a simple payroll where a predetermined total is made of the employee hours per pay period. During the processing of the payroll, a total of hours per employee is accumulated and at the end of the program, the two totals are compared.

Proof Figures

Proof figures are sometimes used to check an important multiplication in a program. The proof figure is usually additional information carried in the record. An example of this is the multiplication of quantity by cost required in grocery billing. The check is based on a relationship between cost and a so-called proof cost. An arbitrary fixed figure Z, larger than any normal cost, is set up. Then the proof cost is expressed by the formula: Cost + proof cost = Z.

When quantity is multiplied by cost, it is also multiplied by proof cost. Normally, two of the totals needed for the check, quantity and quantity times cost, are accumulated during the program. The other factor needed for the check (quantity times proof cost) is also accumulated in the program. Now it is possible at any point to check as follows:

$$\Sigma(\text{Quantity} \times \text{Cost}) + \Sigma(\text{Quantity} \times \text{Proof Cost}) = \Sigma(\text{Quantity} \times Z)$$

The left side of the equation can be calculated by a single addition of the two progressive totals accumulated during the program. The right side of the equation can be calculated by a multiplication of the accumulated quantity and the factor Z. This check insures that each particular multiplication was performed correctly. This type of check applies to other applications by the same general approach, that of adding check information.

Limit Check

A limit check is the test of a field in a record or a total in the program to see if certain predetermined limits have been exceeded. An example of this would be a transaction code which is known to include only numbers 0 through 5. In the program, a check should be made to see that the code does not exceed 5.

Another limit check applies to reasonableness. For example, certain totals are known to vary not more than 10 per cent between processing cycles. This check can be easily programmed.

A further use of this check is in a table look-up operation. If a value is known to be in a given table, the modified table address may be checked against the address of the upper table value to verify correctness of the search. If the search begins to exceed the limits of the table, an error has occurred and corrective action should be taken.

Crossfooting Balance Checks

Crossfooting balance checks are useful in many programs. An example is in payroll calculation. During the processing of each record in a payroll, independent totals are accumulated of gross pay, taxes, social insurance, deductions, and net pay. These totals can be crossfooted and checked at any point in the program. For example, the total gross pay at any point should equal total net pay, plus total deductions, social insurance, and taxes.

Program Checks

Program checks are normally designed to examine and verify the execution of certain routines or particular instructions in the program which may not be satisfactorily covered by systems or machine checks. Like systems checks, they are accomplished by programming. Many types of program checks may be used. They include checks of arithmetic operations when amounts or quantities are being originated in the procedure and no previous control total exists. For example, the payroll calculation of gross pay for each individual employee cannot be verified by control totals.

A check can be made of comparisons or upon the sequence in which instructions are executed. Checks can be devised which check recognition of plus or minus balances, reading or writing into proper locations, selection of proper drum sections or other units, backspacing, and others.

The programmer should always first use the checks which are readily available, including machine and systems checks. However, if these checks do not satisfy the requirements of procedure quality control, the characteristics of the problem itself may be considered for additional checks. A problem often contains self-checking or limiting factors which may furnish suitable systems or program checks. When necessary, the problem should be modified during early stages of planning to include program checks.

Check-Point Procedure

A check-point procedure is a programmed checking routine performed at specific processing intervals or check-points. Its purpose is to check that the program has been performed correctly to a predetermined point. If it has, then the status of the machine is written out or "remembered" periodically. The program is then continued until the next check-point is reached.

If an error is detected at a check-point, or during the processing interval between check-points, the procedure is backed up to the previous check-point. The machine is then restored to its exact status as recorded on the check-point tape or on the drum. The entire program between check points is rerun.

The operation of "backing up" requires that the record of machine status, as originally written on the check-point tape or on the drum, be restored in memory. Also, it requires that all input and output tape units must be returned to their positions at the time the last check-point was recorded.

However, check-point can also be used in procedures involving printing or card punching. When restarting, proper identification of the output group of cards or printed records containing the error must be made so that these records can be removed at the end of the job. When using the card reader, manual intervention is necessary to restart.

Restart Procedure

A restart procedure is a programmed routine designed to return the machine automatically to some predetermined point in the problem. This is normally the preceding check-point.

The restart accomplishes two things:

1. It "backs up" the entire machine system to the predetermined point in the problem. Tape units are backspaced automatically; card units and printer are adjusted manually.
2. It restores the memory of the machine to its status preceding check-point. Only certain selected portions of memory may be restored, if desired.

The purpose of the restart procedure is to reduce the need for manual intervention in case of error. The routine may be executed automatically when an error is discovered at check-point or it may be entered from any point in the program when an error condition is recognized.

If a stop occurs while a program is running, one or more of the following factors will have caused it:

1. Data errors.
2. Operator errors.
3. Random machine errors.
4. Emergency repairs or power failure.
5. End of shift or job interruption.

The operator has a choice of procedures to put the machine back in operation. He may make the correction, rerun all or a portion of the program, adjust the machine (maintenance), or shut the system down. It is a great advantage to include programming to automatically correct steps 2, 3, and 5 above.

The program may instruct the machine to "try again" in the case of a random machine error or to check the problem at regular intervals so that, if a rerun is necessary, only a portion of the program needs to be rerun.

The first approach to a restart is to establish some point in the program to which the program will return in the event of a stop. The simplest procedure is to merely return to the original starting point of the job. The program includes a transfer to "housekeeping" instructions to rewind all tapes, turn off I-O indicators, set switches, check tape reel labels, and so on. In such a restart procedure, no record of memory is needed.

A second approach is to establish a break or recovery point as the end of a reel. Checking for errors, storing of accumulated totals, checking reel labels, or recording of other pertinent information is done before or after each reel change. The end of a reel then signals a break point and all other reels are changed at the same time, whether at end of file or not. If a restart is necessary, the system recovers to the beginning of all files.

This second method, however, is practical only when both input and output reels are of approximately equal length. The effect is to divide a multireel application into a number of single-reel jobs. Figure 49 shows variations of this method. In this case, assume that files consist of four reels of master input, four reels of master output, and one reel of detail transactions.

Detail records are merged with masters and do not add records to the master input file. Recovery points are established at the end of each master input reel, dividing the application into four smaller, independent runs. Control totals and other restart information are written on the output tape at the beginning of each master reel as a special record. This record might be an addition to the reel label.

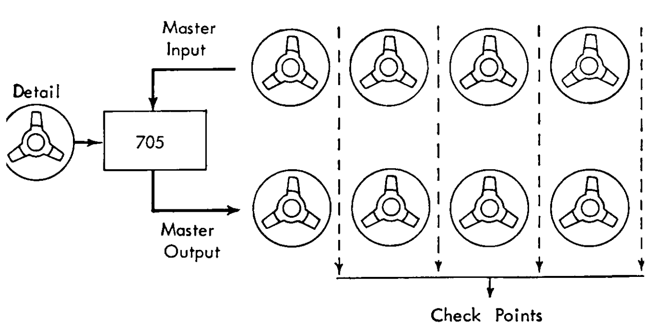


Figure 49. Schematic, Recovery Point at End of Reel

If a restart is necessary, the program rewinds the master input, output, and the detail reels. The totals are reestablished and the detail reel is then read forward by the restart routine until a record matching the first master on the input reel is located. The job is then continued in the normal manner. Maximum time lost by the restarting procedure is the time required to process and rewind one full reel of tape. If a power failure or other interruption occurs, the same procedure can be used, except that the program must also be reloaded.

General Consideration of Check Point and Restart

The problems of an application must be viewed when planning for restart procedures.

1. Checks may be taken, before printing, of totals when group control occurs. These checks include 00901, 00904, and 00905. In the event of an error, a transfer to a restart procedure is made which returns the entire machine to the last error-free check point. A sign check usually indicates an error in input data. Therefore, a programmed stop when this error is detected may be advisable. Indicator 00900 is normally set to AUTOMATIC. The operator then has an option to make corrections when an instruction error occurs or to transfer to a restart without continuing to the next check point. The program may be reloaded or read in automatically as a part of the restart procedure.

2. Systems checks may be included at check point. Restart may not always be desirable, if a systems error is detected. For example, if a control total is out of balance and the record count shows a record is missing from an input file, a rerun of this section of the file cannot locate the missing record. In this case, the check point should indicate the cause of error and stop the machine with operator option either to continue or to take corrective action.

3. Normally, the check point routine is modified depending upon the point at which it is taken. For example, when systems checks are also taken at check point, a more complete check occurs than if check point is forced at periodic intervals.

4. Read-write errors are not tested at check points. They should be reprocessed as they occur and operation stopped if they persist.

5. Jobs running 15 minutes or over should normally include check-points with automatic restart.

Jobs without check-point might extend for longer periods of time if they are to be run infrequently. Validity of results must be checked for any job, regardless of its length. However, for jobs of less than 15 minutes' running time, it is generally more economical merely to rerun the job in case of error (not R-W errors), rather than to devote additional programming time to comprehensive checking.

Locating Character Code Errors in Memory

When a character code error occurs with 00901 check switch at AUTOMATIC, its location in most cases is indicated by MAC I. During the writing of such a record from memory or reading one from tape or drum, the location of the error record in memory is not known. If the record consists of a large number of characters, manual display will be exceedingly inefficient. Therefore, a useful technique for locating character code errors is the following set of instructions (00901 machine check indicator set to AUTOMATIC):

1.	LNG	00256
2.	SET	00256
3.	MPY	00256
4.	LOD	00256

With this procedure, the reset key is not used to reset the accumulator. Therefore, the first instruction removes all but one storage mark and replaces all other positions of the accumulator with numerical zeros, thus assuring the removal of all code errors and/or non-numerical characters. The second instruction places the storage mark over the starting point counter. The MPY instruction removes this storage mark, which allows the LOD instruction to load continuously all of the characters of memory into the accumulator until character coding errors cause a 00901 machine check stop; (MAC contains the location of the character and CR 1 contains the character).

The character in error should be replaced by a manual store operation, and then the same above set of four instructions should be executed in order to locate additional character code errors, if any.

The above routine should be repeated until memory circulates freely through the accumulator. The continuous LOD instruction can be stopped only by depressing the machine stop key. This method may be used on 705 I and II systems.

The following method may be used on all 705 systems:

1.	LNG	00256
2.	SET	00256
3.	MPY	00256
4.	CMP	00256

The function of the first three instructions is the same as in the preceding routine, and the CMP instruction locates a code error in memory in the same way as the LOD instruction in the first program.

The advantage of the CMP instruction is that no character from memory will be moved into the accumulator. Therefore, all that is necessary to locate the second and subsequent code errors is to repeat the CMP instruction.

When it is necessary to use one of these routines, remember that the contents of the accumulator are destroyed, and, in the case of the second routine, the status of the equal and high indicators may be changed.

Programming Techniques and Examples

Indirect Addressing

The feature of indirect addressing in the 705 III produces simplified programming and faster processing times, especially in the jobs requiring a considerable amount of address modification.

All instructions previously illustrated have been shown using direct addresses. Such an address always refers to the location of data in memory, RAD 16004, or to a selected machine component, SEL 00204. A direct address may also control other machine functions of shifting, SET 00010, rounding, RND 00002, and so on.

Any instruction executed by the 705 III, however, can use an *indirect* address. This type of address does not refer directly to the data to be processed or to a machine component. It only refers to memory locations which *contain the address* of the data, device, or control function. In memory, the indirect address is identified by a 1 in the A-bit position of the low-order character. In the following text, indirect addresses are shown with an asterisk over the units position.

Assume that memory contains the following:

LOCATION	CONTENTS	DESCRIPTION
01049	H $\overset{01}{6}$ 0 0 $\overset{01}{4}$	Instruction RAD
16004	$\overset{01}{8}$ 0 0 3	Address of data
18003	b 7 $\overset{+}{2}$	Data

The address part of the RAD instruction, located at 01049, is indirect since it contains a 1 in the A-bit position of the low-order character. The *effective* address of the data (18003) is found at location 16004. The contents of location 18003 (+72) are the amount placed in the accumulator.

Method of Operation

The 705 III executes an instruction with an indirect address in the following manner. During the instruction cycle (I-time), the operation code appears in the operation register and the indirect address appears in the memory address register. The machine then notes that the address is indirect. At this point, the equivalent of a second instruction cycle is taken, during which the indirect address is replaced in the memory address register by the effective address. Following this, the normal execution cycles are taken according to the operation code of the original or indirect address instruction.

In executing the instruction RAD 1600$\overset{*}{4}$, the 705 III refers to locations 16004, 16003, 16002, and 16001 for the effective address. All ASU zoning and the A-bit position of the low-order character are ignored.

Any number of indirect addresses throughout a program may refer to a single effective address. In the above example, a second instruction, SUB 1600$\overset{*}{4}$, causes the number +72 to be subtracted from the accumulator.

If the indirect address does not end in 4 or 9, the 4 or 9 indicator is turned on.

1. The ASU zoning of an effective address is ignored when an indirectly addressed instruction refers to it. For example, the routine below adds some constant, located in 01004, into ASU 01, and then subtracts the same constant from ASU's 02, 03, and 04.

00104	ADD	(01)	01004
00109	SUB	(02)	0010$\overset{*}{4}$
00114	SUB	(03)	0010$\overset{*}{4}$
00119	SUB	(04)	0010$\overset{*}{4}$

2. If an effective address has a 1 in the A-bit position of the low-order character (indicating that it also is an indirect address), it is interpreted as an effective address only when being executed as such. In other words, the 705 could not be made to step through a series of indirect addresses, searching for a bona-fide effective address, before completing the execution of an instruction. For example, in the illustration below, the address 01004 is both the indirect address of instruction 00104 and the effective address of instruction 00109.

00104	LOD	(01)	0100$\overset{*}{4}$
00109	LOD	(02)	0010$\overset{*}{4}$

In this case, the 01004 would be treated as an indirect address only when executing instruction 00104 and not when executing instruction 00109. Thus, these two instructions would load different amounts into ASU 01 and ASU 02. Or, to restate the rule: An indirect address cannot refer to another indirect address and have the second interpreted as indirect.

Any of the intruction operation codes that are operative on the 705 III may be used with indirect addressing, including the control (operation code 3) instructions. For example, one could give CTRL 1200$\overset{*}{9}$

and in location 12009 might be found 00004, meaning backspace.

A 1 in the A-bit position of the units character of an instruction address is interpreted by the 705 III as has been described. This does not interfere with the running of programs written for Model I and II machines, because on those machines the zone of the units position was unassigned. However, note that, if a particular user has assigned this zone position for other purposes, the program obviously would not run as on the model I or II, when used on the 705 III.

Assume that a particular instruction needs to be modified several times during the course of a program. Assume further that there are ten different instructions at different places in the program, all having the same instruction address. Without indirect addressing, all ten instructions would have to be modified whenever one was modified. However, if these ten were indirect addresses referring to a single effective address, only the effective address need be changed. This eliminates keeping track of the ten individual instructions and modifying each of them separately.

Perhaps the most obvious example of the need for this is in modifying the instruction that selects a tape unit whenever tape units are switched on an end-of-file condition. Consider a program starting as follows:

00004	SEL 00200
00009	RD 10000

The SEL 00200 instruction also appears several other places in the program, as in the testing of error and ready indicators, in error-correction routines and in end-of-file routines. Whenever the first input reel is completed, the program switches to tape unit 00202 where the second input reel is ready to start. To make this switch of tape units, all of the SEL 00200 instructions have to be changed to SEL 00202; however, if all these other SEL instructions were written SEL 0000$\overset{*}{4}$, only the instruction at location 00004 would have to be changed. In this example, the normal address of the first SEL instruction is also the effective address for the others (which are indirect).

With the possibility of using a machine with a memory capacity of up to 80,000 characters, many, if not most, of the instructions in a program refer to locations over 10,000; thus, they have some zoning over the thousands and/or units position of the instruction address. The feature of indirect addressing also makes use of zoning over the units position of the instruction. This increased use of zoning not only prevents considering an address always as an unsigned field in memory (as when using the add-to-memory instruction) but also may tend to complicate the manipulation, modification, and comparison of instruction addresses with other existing operation codes.

To provide greater ease in the manipulation of instruction addresses, three instructions are available on the 705 III.

INSTRUCTION	MNEMONIC CODE
Load address	LDA
Unload address	ULA
Add to address in memory	AAM

Because these instructions are designed to operate on instruction addresses, rather than general data, they must be addressed to a memory field ending in 4 or 9. If the instruction address of any of these three instructions is other than one ending in 4 or 9, the instruction check indicator (4 or 9 check) is turned on.

Program Switch

NOP-TR Switch

Several types of switches can be built into the program. For example, an instruction can be cancelled entirely to become "no operation" by substituting the no-operation code in the operation part of the instruction. Or, by inserting different operation codes into instructions located in memory, the machine can change a no-operation to some other specific operation.

A flow diagram for a portion of a balance-forward procedure is illustrated in Figure 50. A master inventory record is read as one input record. Detail transactions affecting the master record are read as a second input record.

The detail is compared to the master record by product number. The results of the comparison lead to one of three possible program paths.

Equal. The master and the matching detail are written out on tape. Assume that there may be more than one detail to match one master. Therefore, after a detail is written, a new detail must be read into memory and compared against the same master. If a second equality results, only the detail is written, without rewriting the master record. To accomplish this, a switch is inserted before the write instruction, to direct the program around the writing of the master record.

The switch is a no-operation instruction with the location in memory of the detail write instruction for its address part. The operation code of the no-operation instruction is A. To change the no-operation instruction to transfer, the 11 (one, one) zoning is removed from the A by a sign instruction. The result-

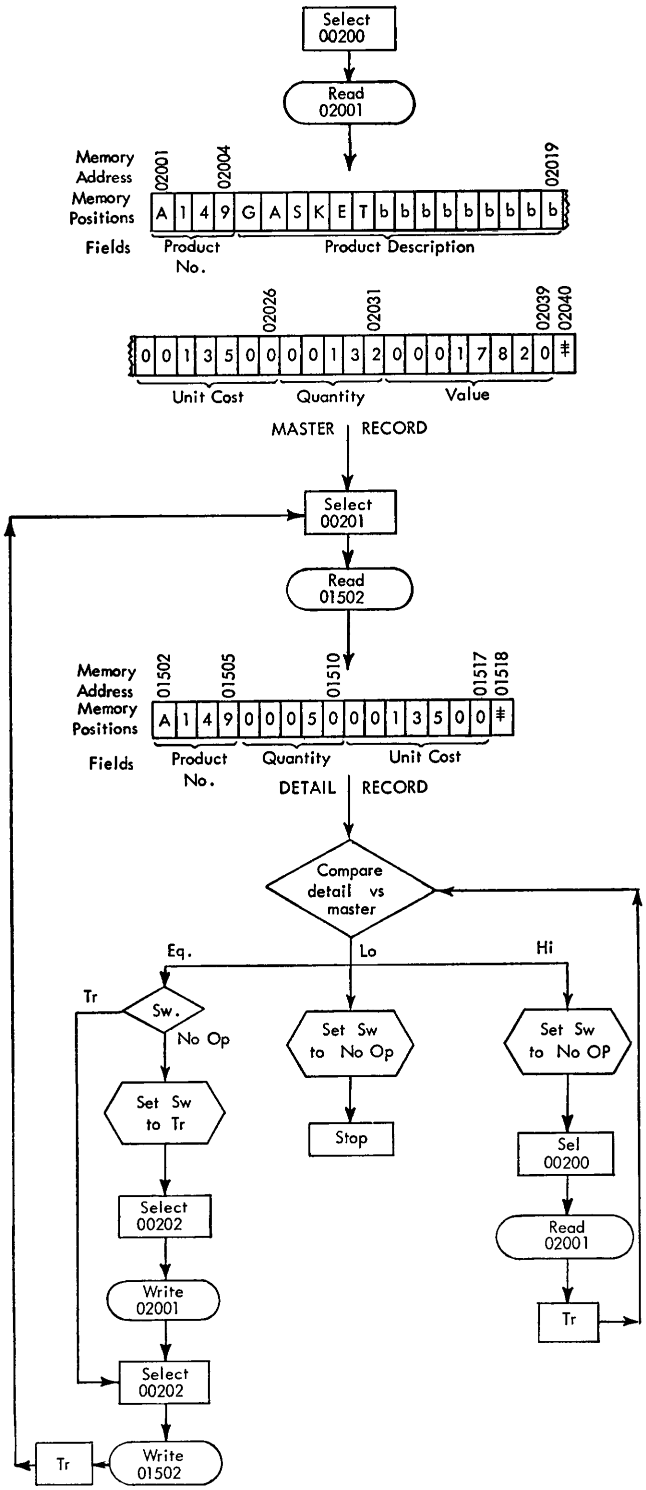


Figure 50. Balance Forward

ing character is 1, the operation code of the transfer instruction. The switch instruction now becomes a transfer whose address part specifies the location of the detail write instruction. Once the switch has been set to TR, the instructions for writing the master record are bypassed.

Low. A low detail is unmatched. The switch is set to NOP to permit writing the next master. A stop is indicated because an unmatched detail is assumed to be an error condition. Further instructions could be inserted here to type or punch out an error record and continue automatically. These instructions have been omitted to simplify the problem.

High. A high condition means that the master record is unmatched. This record is ignored and a new master is read into memory to be compared with the detail record previously read. The switch is set to NOP to permit the printing of the next master record.

Program, NOP-TR Switch

Figure 51 is the program for the switch shown in Figure 50. Assume that a group mark has been properly placed in the writing areas. Housekeeping or other preliminary routines are not illustrated.

00004. Select the input master record.
00009. Read the master record.
00014. Select the input detail record.
00019. Read the detail record.
00024. Adjust ASU 01 to four positions.
00029. Load product number into ASU 01 from the detail record.
00034. Compare the detail with the master product number.
00039. Transfer to the equal routine when the detail matches the master.

INSTR. LOCATION	INSTRUCTION OPER.	INSTRUCTION ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
Read Records and Compare							
00004	SEL	00200					
00009	RD	02001					
00014	SEL	00201					
00019	RD	01502					
00024	SET	00004	01			axxxx	
00029	LOD	01505	01			aA149	+
00034	CMP	02004	01			aA149	+
00039	TRE	00064					
00044	TRH	00099					
Lo Detail							
00049	SGN	00060	02			a&	+
00054	ADM	00060	02				
00059	HLT	00001					
Eq Detail							
00064	(NOP)	00084					
00069	SGN	00060	02			a&	+
00074	SEL	00202					
00079	WR	02001	00				
00084	SEL	00202					
00089	WR	01502	00				
00094	TR	00014					
Hi Detail							
00099	SGN	00060	02			a&	+
00104	ADM	00060	02				
00109	SEL	00200					
00114	RD	02001					
00119	TR	00034					

Figure 51. Program, Balance Forward

00044. Transfer to the high routine when the master is unmatched.

LOW DETAIL

00049. Sign the operation part of the switch instruction to change the character A (NOP) to 1 (TR). The sign instruction places a 11 (one, one) zone in ASU 02 when the address operation part is either NOP or TR (A or 1). The sign instruction may be placed in a housekeeping routine, provided ASU 02 is not used for any other operation in the program. In this case, ADM 02 will always change the switch to NOP.

00054. Add the 11 zone to the operation part of the switch instruction. If the character is A, it will remain A. If the character is 1, it will be changed to A. The sign and add-to-memory instructions, therefore, always set the switch to NOP regardless of its previous setting.

00059. Stop. Unmatched detail.

EQUAL DETAIL

00064. Switch instruction. When the operation part is the character A, the switch is set to NOP. The machine ignores this instruction and the address part is not effective. When the operation part is the character 1, the switch is set to TR. A transfer is made to the instruction specified by the address of the switch.

00069. Set the switch to TR by removing the 11 (one, one) zone from the operation part of the switch instruction. The sign instruction always sets the switch to TR regardless of its previous setting.

00074. Select the output tape unit.
00079. Write the master record.
00084. Select the output tape record.
00089. Write the detail record.
00094. Transfer to read another detail record.

HIGH DETAIL

00099, 00104. Set the switch to NOP.
00109. Select the input master tape.
00114. Read the next master record.
00119. Transfer to compare the detail against the master.

Bit Switches (705 III)

With bit manipulation, it is possible, by programming, to examine or change any individual binary bit within a character in any position of memory. Two instructions, transfer on zero bit and set bit, expand program control of data and save memory space. Bit manipulation also provides a rapid, simple method of program switching. Any practical number of switches can be set or tested without using an ASU or placing a constant in memory.

The ability to test the bits of a character in memory and to transfer offers the equivalent of an unlimited number of program alteration switches. Using the bits as switches, the program, demonstrating the equivalent of using three alteration switches, would be written as follows:

```
RCV - - -
TZB - - - (01)
TZB - - - (02)
TZB - - - (03)
```

These switches can be individually set either from the console or from the program. Moreover, when the switches are to be grouped five per character, all switches can be set in a single operation by unloading predetermined characters into these positions.

Where NOP-TR switches are used, they can be set easily, positively, and, in some cases, more quickly by bit manipulation, and without tying up an ASU or a character in memory as would be the case when SGN and/or ADM instructions are used:

```
SB - - - (05)     Set A bit to 0.
SB - - - (06)     Set B bit to 0.
```

The switch would be reversed in the same manner with SB (13) and SB (14).

Grouping Records

To take full advantage of the high internal speed of the 705 and to conserve reading and writing time, it is frequently desirable to process a block of records at a time. Records can be grouped in any desired number within the capacity of memory.

Figure 52 illustrates the time saved by grouping records in blocks of one to ten records using the IBM 727/729 I Magnetic Tape Units. The figure shows that for 10,000 records of various sizes, an appreciable saving per record is realized by grouping at least four records per block. Although larger blocks continue to show savings, the increase in total savings for the job becomes negligible. It is doubtful that grouping records in blocks greater than ten per block for this number of records would be particularly advantageous. The time that is saved by grouping records is the 10 milliseconds of tape unit start time. As the number of records to be processed becomes larger, more advantage is obtained by grouping records in larger blocks.

The grouped records are usually separated by a record mark. If the record size including the record mark is divisible by 5, the high-speed RCV and TMT instructions can be used to move the individual rec-

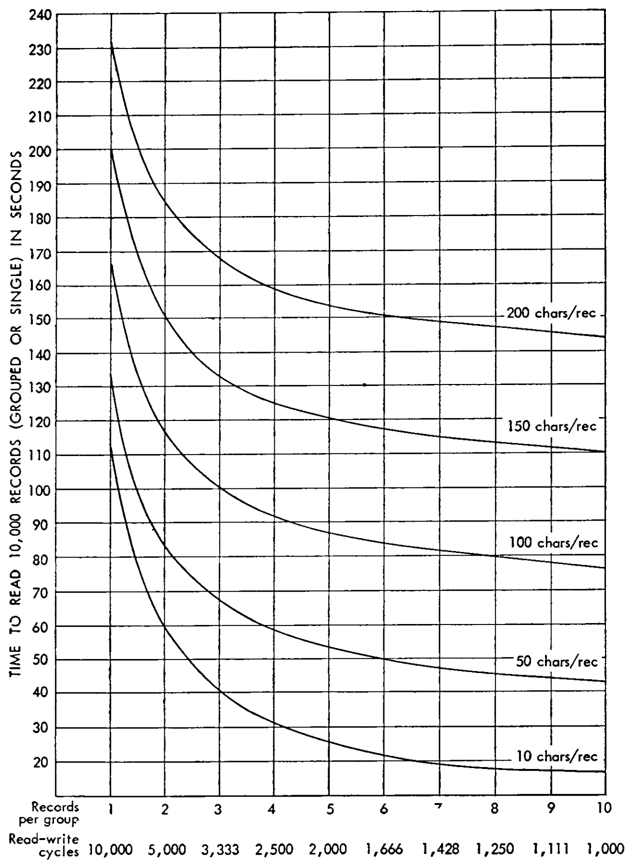


Figure 52. Timing Chart, Grouped Records

ords within memory for processing. A primary use of the record mark is to separate one record from another when records are blocked.

A sample file maintenance flow chart is illustrated in Figure 54. In this sample application, the records are grouped in blocks of four records per block as shown in Figure 53. Each record, including a record mark, is 100 characters in length. Each block of master records is separated on tape by an inter-record gap and is read into memory each time a read instruction is executed.

Detail records, representing activity against the master file, are read into memory singly. To process the detail against the master records, each master record is moved into a temporary work area so that the same instructions can be used regardless of where the master record was originally read into memory.

The adjusted master records are assembled into two blocks of four records each for writing. The purpose of the two blocks is to make as much use as possible of the simultaneous RWW instruction. Because it is possible for a detail record to delete a master record or to be used to create a new master record, the number of records in the input block and the output block can get out of step. By using two output blocks, usually at least one of the output blocks will be full and ready to write when the records in the input block have been processed.

This situation is controlled by placing a 5 into auxiliary storage unit 01 at the beginning of the program (Figure 56). Whenever the input area is exhausted, 3 is subtracted from auxiliary storage 01. Whenever a new block is read into the input area, 3 is added to storage unit 01. Whenever one of the output blocks is full, a 1 is subtracted from auxiliary storage 01, and, when an output block is written, a 1 is added to auxiliary storage 01.

Each time a master record is moved into one of the output blocks (Figure 55), auxiliary storage 01 is checked and a decision is made as shown in Figure 55.

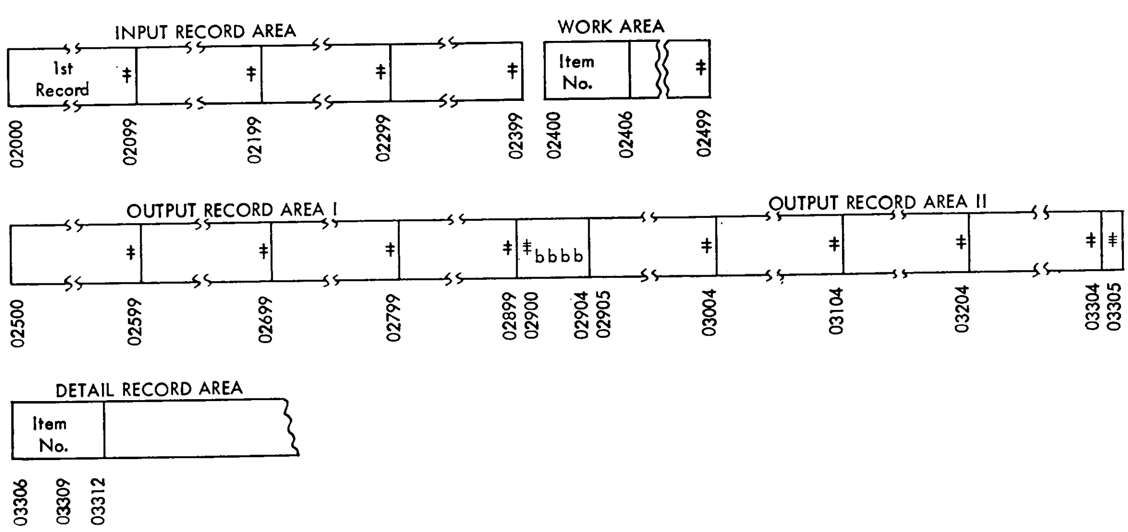


Figure 53. Record Groupings, File Maintenance

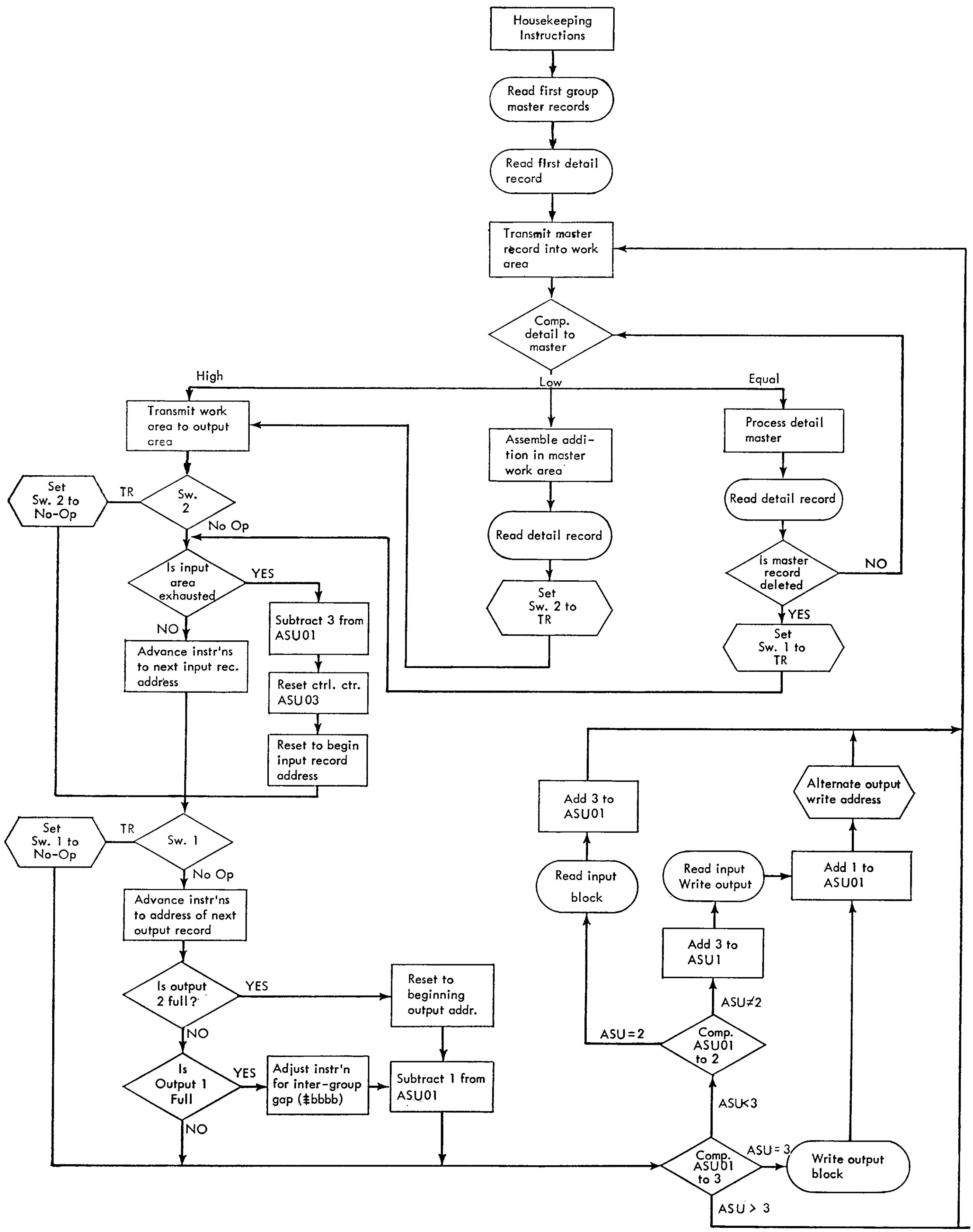


Figure 54. Flow Chart, File Maintenance

INPUT BLOCK				OUTPUT BLOCK I				OUTPUT BLOCK II			
1	2	3	4	5	6	7	8	9	10	11	12
Record				Record				Record			

Figure 55. Input-Output Decisions

The two output areas are always filled in sequence starting with output area I, proceeding through II, and then back to I, and so on. An alternator switch is set up to govern which output area is to be written next. It causes output area I to be written, then II,

b	b	b	b	b	0	+5	2	5	0	+4	2	0	0	+4	+1	+1	+3	+5	0	1	0	+0	3	3	0	+9	2	9	0	+4	2	5	0	+0	2	9	0	+5	‡
01960						01966				01970				01974	01975	01976	01977	01978				01982				01986				01990				01994				01998	01999

(Constants)

a5	a0100	a0000	a3309	a2904	Store Sign	a‡	axxxxxxx
01	02	03	04	05	06	07	08
a2004	a2504	a2500	a2905				
09	10	11	12	13	14	15	

INSTR. LOCATION	INSTRUCTION OPER.	ADDRESS	STOR. CODE	EXPLANATION
Housekeeping Routine				
00004	RAD	01978	01	Place 5 in ASU 01
00009	RAD	01982	02	Place 0100 in ASU 02 (Rec. length)
00014	SET	00004	03	Place 4 zeros in ASU 03
00019	RAD	01986	04	Place 3309 in ASU 04
00024	RAD	01990	05	Place 2904 in ASU 05
00029	SET	00001	07	
00034	LOD	01999	07	Load group mark in ASU 07
00039	UNL	02900	07	Unload behind Output Area I
00044	UNL	03305	07	Unload behind Output Area II
00049	SET	00007	08	For comparing detail master
00054	RAD	01974	09	Beg. Input Address 2004 (TSMT)
00059	RAD	01970	10	Beg. Output Address 2504 (RCV)
00064	RAD	01994	11	Beg. Output Address Block I (Write)
00069	RAD	01998	12	Beg. Output Address Block II (Write)
00074	SEL	00200		
00079	RD	02000		Read first block of master records
00084	SEL	00201		
00089	RD	03306		Read first detail record
00094	TRA			
Main Routine				
00099	RCV	02404		
00104	TMT	(02004)	00	Move input record to work area
00109	LOD	03312	08	Load detail item number
00114	CMP	02406	08	Comp. to master item number
00119	TRH	00309		
00124	TRE	00229		
Addition to Master File				
Assemble detail record in work area				
00204	SEL	00201		
00209	RD	03306		Read next detail record
00214	TRA			
00219	SGN	00315	06	Set switch 2 to Tr
00224	TR	00309		
Activity against Master File				
00229	Process detail record. If transaction does not result in deletion, transfer to location 0114 for next instruction. If transaction results in deletion:			
00304	SGN	00350	06	Set switch 1 to Tr
No Activity against Master File				
00309	RCV	(02504)		
00314	TMT	02404	00	Move master to output area
00319	NOP	00594		Switch 2
00324	NTR	00349	03	Does this record exhaust input area?
00329	SUB	01977	01	Yes, sub 3 from ASU 01
00334	SET	00004	03	Restore input ctrl. counter
00339	UNL	00104	09	Reset TMT instr. to beg. input address
00344	TR	00354		
00349	ADM	00104	02	Advance
00354	NOP	00579		Switch 1
00359	ADM	00309	02	Advance address of next output rec.
00364	CMP	00309	04	Is output area II full?
00369	TRE	00549		Yes
00374	CMP	00309	05	Is output area I full?
00379	TRE	00559		Yes

INSTR. LOCATION	INSTRUCTION OPER.	ADDRESS	STOR. CODE	EXPLANATION
00384	CMP	01983	01	Compare ASU 01 to 3
00389	TRH	00099		ASU 03
00394	TRE	00449		ASU=3
00399	CMP	01987	01	Compare ASU 01 to 2
00404	TRE	00519		ASU=2
Input Block Exhausted/Output Block or Blocks Full				
00409	ADD	01977	01	Add 3 to ASU 01
00414	TRA			
00419	SEL	00200		Alert input tape unit
00424	RWW	02000		Keep input tape unit alerted
00429	SEL	00202		Alert output tape
00434	WR	(02500)	00	Read input block, write output block
00439	TRA			
00444	TR	00469		
Both Output Blocks Full				
00449	TRA			
00454	SEL	00202		
00459	WR	(02500)	00	Write output block
00464	TRA			
00469	ADD	01976	01	Add 1 to ASU 01
00474	RSU	01975	00	Alternator Switch for
00479	ST	01975	00	alternating write instructions
00484	TRP	00504	00	for output areas
00489	UNL	00434	12	Adjust write instructions address
00494	UNL	00459	12	for output block II
00499	TR	00099		
00504	UNL	00434	11	Adjust write instructions address
00509	UNL	00459	11	for output block I
00514	TR	00099		
Input Block Exhausted				
00519	TRA			
00524	SEL	00200		
00529	RD	02000		Read next block of master records
00534	TRA			
00539	ADD	01977	01	Add 3 into ASU 01
00544	TR	00099		
Output Area II is Full				
00549	UNL	00309	10	
00554	TR	00569		
Output Area I is Full				
00559	RAD	01966	00	Place 05 in accumulator
00564	ADM	00309	00	
00569	SUB	01976	01	Subtract 1 from ASU 01
00574	TR	00384		
00579	SGN	00350	06	
00584	ADM	00350	06	Set switch 1 to No Op
00589	TR	00384		
00594	SGN	00315	06	
00599	ADM	00315	06	Set switch 2 to No Op
00604	TR	00354		

Figure 56. Program, Grouped Records

and then back to I, and so on, regardless of whether the operation is write only or simultaneous read-while-writing. Figure 56 is the program for file maintenance with grouped records.

NO. IN ASU 01	SITUATION	DECISION
5	Record 4 not processed. Records 8 and 12 not loaded.	Continue processing
4	Record 4 not processed. Record 8 or 12 loaded.	Continue processing
3	Record 4 not processed. Records 8 and 12 loaded.	Write one output block
2	Record 4 processed. Records 8 and 12 not loaded.	Read input block
1	Record 4 processed. Record 8 or 12 loaded.	Read and write
0	Record 4 processed. Records 8 and 12 loaded.	Read and write

NOTES: When record 4 is processed, all input records on the input block have been processed.

When record 8 is loaded, output block I is full.

When record 12 is loaded, output block II is full.

Sequence Checking

The accumulator and auxiliary storage units of the 705 can store either alphabetic or numerical data with equal ease. Single characters, fields, or complete records can be placed in storage as required.

Once stored, the data can be compared against other data in memory or transferred to one or more locations in memory. Addresses, operation codes, or complete instructions may be placed in a storage unit and transferred to modify or replace other instructions in the program. Unsigned numerical fields may be placed in storage for comparison or for rearrangement of records without the use of arithmetic instructions. These operations are accomplished by using load, unload, and compare instructions.

Figure 57 illustrates a problem of sequence checking. An inventory record to be stored in memory is now on tape. The file of records is to be checked to determine if the records are in ascending numerical sequence by part number. If any out-of-sequence or equal records are discovered, those records are to be written on the typewriter for checking by the operator. When the sequence is correct, unit cost is developed, and the record is transmitted to an output area and written on tape. A work area of memory is provided for storage of part number from the previous record. A group mark has been placed in memory location 15865 and must be moved to both input and output areas.

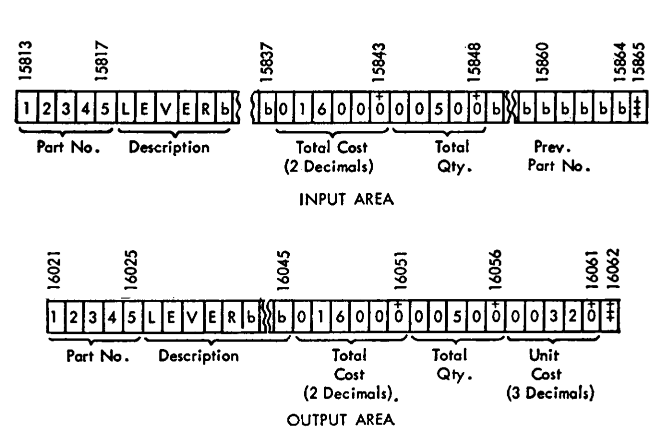


Figure 57. Sequence Checking

PROGRAM, SEQUENCE CHECKING

Figure 58 is the program for the problem shown in Figure 57.

00004. Adjust auxiliary storage unit 01 to one position.

00009. Load the group mark from location 15865.

00014. Unload group mark at end of input record. If records are out of sequence, the input area is written on the typewriter.

00019. Unload group mark at end of output record. Records in sequence are written out from this area.

00024. Adjust auxiliary storage unit 02 to five positions. The five-position part number from each rec-

INSTR. LOCATION	INSTRUCTION OPER.	INSTRUCTION ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00004	SET	00001	01				
00009	LOD	15865	01				
00014	UNL	15849	01				
00019	UNL	16062	01				
00024	SET	00005	02				
00029	SET	00020	03				
00034	SEL	00200					
00039	RD	15813					
00044	LOD	15817	02			a12345	+
00049	CMP	15864	02				
00054	TRH	00074					
00059	SEL	00500					
00064	WR	15813	00				
00069	TR	00034					
00074	UNL	15864	02				
00079	UNL	16025	02				
00084	RCV	16026					
00089	TMT	15818	03				
00094	RAD	15843	00	a016000	+		
00099	ST	16051	00				
00104	SET	00009	00	a000016000	+		
00109	LNG	00002	00	a00001600000	+		
00114	DIV	15848	00	a003200	+		
00119	RND	00001	00	a00320	+		
00124	ST	16061	00				
00129	RCV	16052					
00134	TMT	15844	02				
00139	SEL	00200					
00144	RWW	15813					
00149	SEL	00201					
00154	WR	16021	00				
00159	TR	00044					

Figure 58. Program, Sequence Checking

ord is loaded into this unit for comparison with the preceding record.

00029. Adjust auxiliary storage unit 03 to 20 positions. This unit is used to monitor the transmission of part name from input to output area. Units 03 and 04 are coupled by this instruction.

00034. Select input tape unit.

00039. Read in the first record, beginning at memory address 15813.

00044. Load part number into ASU 02.

00049. Compare the input record part number with the part number from the previous record. The comparison for the first record is made against blanks and, therefore, is always high.

00054. When the input record is higher than the previous record, the records are in ascending sequence. The high indicator is turned on and a transfer is made to the instruction located in memory address 00074. When the input record is lower than, or equal to, the previous record, the high indicator is not turned on and the next instruction is executed without a transfer.

00059. A low or equal condition selects the typewriter.

00064. Write the input record, beginning at memory address 15813.

00069. Transfer to read in another record.

00074. Unload the part number from ASU 02 into the working area. This instruction is reached only when records are in sequence.

00079. Unload the part number into the output area.

00084. Set MAC II to prepare to receive part name into the output area.

00089. Transmit part name to output area. ASU 03 is specified to limit transmission to 20 characters.

00094. Reset and add total cost into accumulator storage 00.

00099. Store total cost in output area.

00104. Adjust the accumulator storage 00 to nine positions to place three zeros to the left of the dividend.

00109. Lengthen the dividend by two zeros. The dividend is now four decimals to permit a three-decimal quotient with half adjustment.

00114. Divide total cost by quantity to get unit cost.

00119. Round unit cost.

00124. Store unit cost in output area.

00129. Set MAC II to prepare to receive quantity into the output area.

00134. Transmit quantity to output area. ASU 02 is specified to limit transmission to five characters. The output record is now complete.

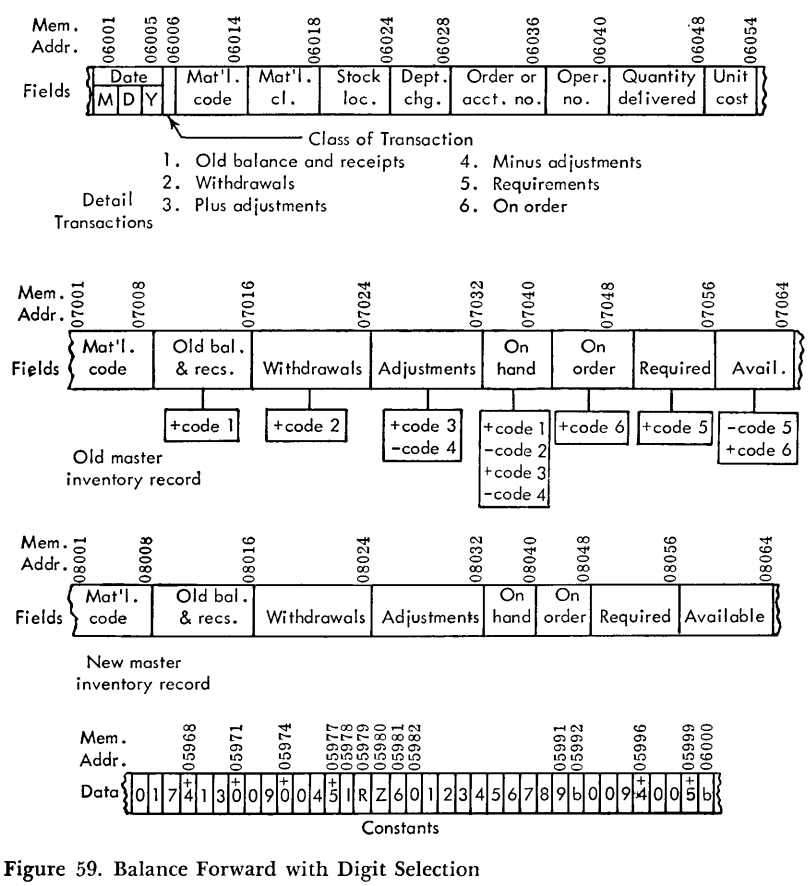


Figure 59. Balance Forward with Digit Selection

00139. Select input tape unit.

00144. Prepare to read while writing.

00149. Select output tape unit.

00154. Write output tape beginning at memory address 16021. Read in a new record beginning at address 15813.

00159. Transfer to load part number. Instructions in locations 00004 through 00029 are housekeeping instructions. Instructions in locations 00034 and 00039 are needed only for the first record, or to replace a record which is out of sequence.

Digit Selection and Decoding

When a single type of record is used for recording amounts or other quantitative data for different classes of transactions, each transaction must be distinguished from the others by some identifying code. A single digit, an alphabetic character or a multiple character code may be used. The choice of codes, single or multiple character, normally depends upon the number of different types of transactions which must be identified.

In Figure 59, the quantity, identification, and descriptive data, unit cost, and other information concerning detail inventory transactions have been transcribed to a tape record. Each type of transaction is identified by a single digit code. Code 1 indicates a receipt, code 2, a withdrawal; code 3, a plus adjustment; and so on. The transaction records are read into memory from tape unit 00202 beginning at address 06001.

The master inventory balance record is also on tape and is read into memory from tape unit 00200 beginning at address 07001. The master record balances are to be brought up to current status by processing against the detail transactions. Before the proper balances are adjusted, the type of transaction must be identified by its digit code. The program instructions cause the digit code to transfer the machine to the proper routine to adjust the particular balances affected by that code. Calculations for other codes are ignored.

For example, the instructions for handling receipts add the quantity received to the receipts and on-hand fields of the master record. Instructions for processing withdrawals add the quantity withdrawn to the withdrawal field and subtract it from the quantity on hand.

The flow chart in Figure 60 outlines the main steps to be programmed to adjust the master inventory record balances. The output area indicated is shown in Figure 59 with assigned memory addresses.

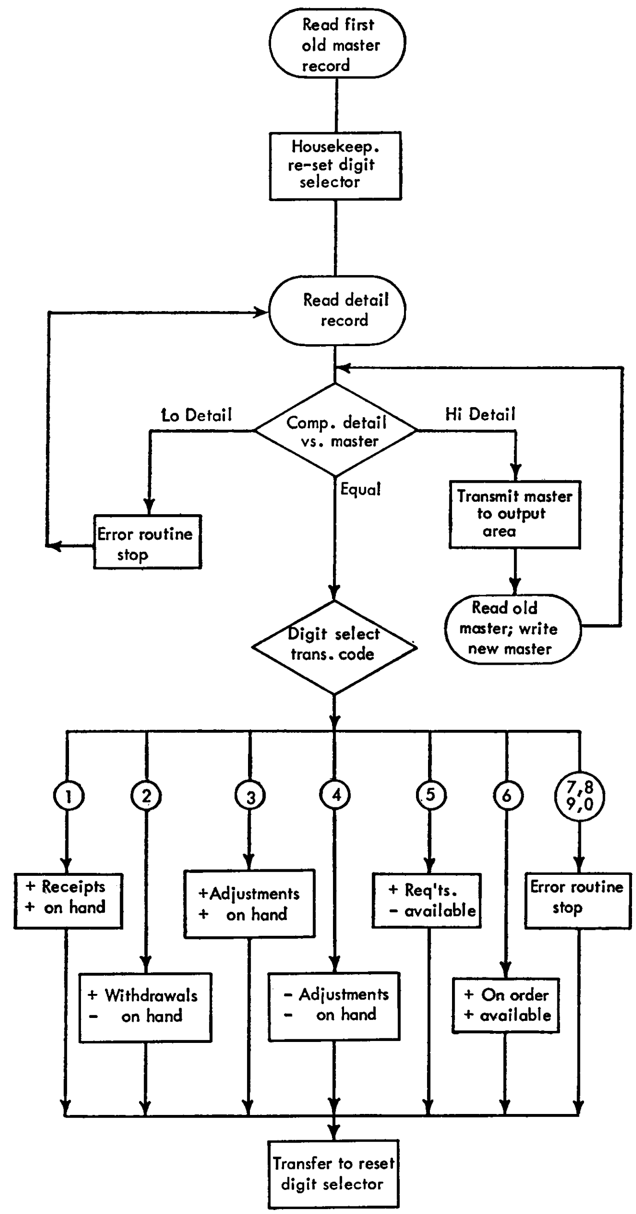


Figure 60. Flow Chart, Balance Forward with Digit Selection

Program, Selection of Single-Digit Code

Figure 61 is the program illustrating the method of digit selection by using address modification of a transfer instruction.

00004, 00009. Select and read the first old master inventory record.

00014. Adjust ASU 01 to eight positions.

00019. Get the initial address of the digit selector transfer instruction.

00024. Reset the digit selector transfer instruction.

00029, 00034. Select input tape unit and read a record.

00039, 00044. Load the material code from the detail and compare against the master record. It is as-

INSTR. LOCATION	INSTRUCTION OPER.	ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00004	SEL	00200					
00009	RD	07001					
00014	SET	00008	01			axxxxxxxx	
00019	RAD	05996	02			a0094	+
00024	UNL	00089	02				
00029	SEL	00202					
00034	RD	06001					
00039	LOD	06014	01				
00044	CMP	07008	01				
00049	TRE	00069					
00054	TRH	01249					
00059	HLT	00001					
00064	TR	to error routine					
00069	RAD	06048	03				
00074	RAD	06006	00				
00079	MPY	05999	00				
00084	ADM	00089	00				
00089	TR	( )					
00094	TR	to error routine					
00099	TR	01144					
00104	TR	01159					
00109	TR	01179					
00114	TR	01194					
00119	TR	01214					
00124	TR	01234					
00129	TR	to error routine					
00134	TR	to error routine					
00139	TR	to error routine					
	Code	1					
01144	ADM	07016	03				
01149	ADM	07040	03				
01154	TR	00024					
	Code	2					
01159	ADM	07024	03				
01164	RSU	06048	03				
01169	ADM	07040	03				
01174	TR	00024					
	Code	3					
01179	ADM	07032	03				
01184	ADM	07040	03				
01189	TR	00024					
	Code	4					
01194	RSU	06048	03				
01199	ADM	07032	03				
01204	ADM	07040	03				
01209	TR	00024					
	Code	5					
01214	ADM	07056	03				
01219	RSU	06048	03				
01224	ADM	07064	03				
01229	TR	00024					
	Code	6					
01234	ADM	07048	03				
01239	ADM	07064	03				
01244	TR	00024					
	Read & write master records						
01249	RCV	08004					
01254	TMT	07004	00				
01259	SEL	00200					
01264	RWW	07001					
01269	SEL	00201					
01274	WR	08001					
01279	TR	00044					

Figure 61. Program, Balance Forward with Digit Selection

sumed that the two records are in ascending sequence by material code.

00049. The detail material record matches the master record. Transfer to identify the class of transaction.

00054. High detail record. Transfer to transmit the master record to the output area.

00059, 00064. Unmatched detail record. Stop the machine and transfer to the error routine (not shown in program).

00069. Reset and add the quantity field from the detail record into ASU 03. When the type of transaction is identified, the quantity can be added to the inventory record in memory. If the quantity is subtracted from the master record, a reset and subtract instruction is given in the subroutine.

00074. Reset and add the digit code from the detail record.

00079. Multiply the digit code by 005.

00084. Add the result to the initial digit selector transfer address at location 00089. For example, when the transaction code is 1: (a) the result obtained by instruction 00079 is 00005, and (b) 00005 plus the initial address 00094 equals 00099. The address of the transfer instruction at 00089 is therefore 00099. A transfer is made to the instruction located at 00099, which is a transfer to the program for the code 1 transaction.

00089. After each detail record is processed, the digit selector address is reset to 00094 by the instruction 00024. The instruction 00084 always modifies this address to transfer to another transfer instruction which selects the proper transaction routine.

00094. Transfer to code 0. No code 0 is given; therefore, all codes except 1, 2, 3, 4, 5 and 6 are errors.

00099. Transfer to the code 1 routine.

00104. Transfer to the code 2 routine.

00109. Transfer to the code 3 routine.

00114. Transfer to the code 4 routine.

00119. Transfer to the code 5 routine.

00124. Transfer to the code 6 routine.

00129, 00134, 00139. Codes 7, 8 and 9 are errors.

CODE 1, OLD BALANCE AND RECEIPTS

01144, 01149. Add the detail quantity to receipts and on-hand.

01154. Transfer to reset the digit selector.

CODE 2, WITHDRAWALS

01159. Add the detail quantity to withdrawals.

01164, 01169. Subtract the detail quantity from on-hand quantity.

01174. Transfer to reset the digit selector.

CODE 3, PLUS ADJUSTMENTS

01179, 01184. Add the detail quantity to adjustments and on-hand quantity.

01189. Transfer to reset the digit selector.

CODE 4, MINUS ADJUSTMENTS

01194, 01199, 01204. Subtract the detail quantity from adjustments and on-hand quantity.

01209. Transfer to reset the digit selector.

CODE 5, REQUIREMENTS

01214. Add the detail quantity to requirements.

01219, 01224. Subtract the detail quantity from available amounts.

01229. Transfer to reset the digit selector.

CODE 6, ON ORDER

01234, 01239. Add the detail quantity to on-order and available amounts.

01244. Transfer to reset the digit selector.

READ AND WRITE MASTER CARDS

01249, 01254. Transmit the master record to the output area.

01259, 01264, 01269, 01274. Select and read the old master record; select and write the new master record.

01279. Transfer to compare the detail against the master record.

Digit Selection by Comparison

Digit or code selection can also be accomplished by comparing a transaction code against a series of constants in memory. One constant is normally established for each of the possible codes as shown in Figure 59, memory addresses 05982 through 05991. In this case, a transfer-on-equal instruction selects the proper program routine.

This method is particularly useful with multiple character codes when the calculation method is impractical. The instructions required to convert Figure 61 to selection by comparison are show in Figure 62.

PROGRAM, DIGIT SELECTION BY COMPARISON

00074. Get the transaction code.

00079. Compare against 0.

00084. Code is equal to 0; transfer to the error routine.

00089. Compare against 1.

00094. Code is equal to 1. Transfer to code 1 routine.

00099. Compare against 2.

INSTR. LOCATION	INSTRUCTION OPER.	ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00074	RAD	06006	00				
00079	CMP	05982	00				
00084	TRE	to error	routine				
00089	CMP	05983	00				
00094	TRE	01144					
00099	CMP	05984	00				
00104	TRE	01159					
00109	CMP	05985	00				
00114	TRE	01179					
00119	CMP	05986	00				
00124	TRE	01194					
00129	CMP	05987	00				
00134	TRE	01214					
00139	CMP	05988	00				
00144	TRE	01234					
00149	TRH	to error	routine				

Figure 62. Program, Digit Selection by Compare Method

00104. Code is equal to 2. Transfer to code 2 routine.

00109, 00114, 00119, 00124, 00129, 00134, 00139, 00144. Compare code against 3, 4, 5 and 6. Transfer to proper routine when equal.

00149. Code is higher than 6. Transfer to the error routine.

Decoding, Single-Character Code

A transfer switch can be set up in the program when the single-character code is either alphabetic or numerical. In this case, the switch can select any of 36 possible subroutines corresponding to the 26 letters of the alphabet and the 10 numerical digits. Figure 63 is a flow diagram showing the program arrangement of the switch.

The calculation of the switch address is basically the same as illustrated for numerical digit selection. However, a comparison of the code character is first made against constants Z, I, and R. This comparison places the code character either in a section of the alphabet (A-I, J-R, or S-Z) or determines that the

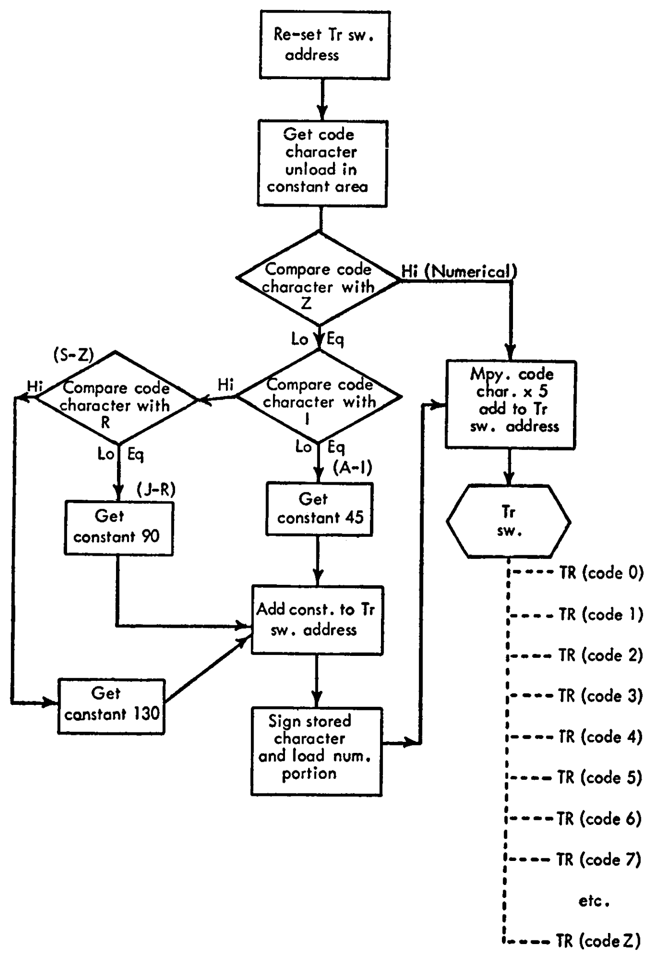


Figure 63. Flow Chart, Decoding Single Characters

code is numerical. A constant is added to the initial address of the switch when the character is alphabetic; 45 for A-I, 90 for J-R, and 130 for S-Z. The alphabetic character is then addressed by a sign instruction which converts it to a digit. The digit is next multiplied by five and the result is also added to the transfer switch address. The final modification of the switch selects the proper transfer instruction.

Program, Decoding Single-Character Code

Figure 64 is the program for decoding single-character codes used with Figure 63.

00019, 00024. Reset the transfer switch at 00139 to address 00174.

00074, 00079. Get the code character from the detail transaction record.

00084. Unload the code into memory at address 05981.

00089. Compare the code against the constant Z.

00094. The code is numerical. Transfer to multiply by 005.

00099. Compare code against the constant I.

00104. The code is between letters J and Z. Transfer to compare against R.

00109. The code is equal to or lower than I. Get the constant 045.

00114. Add the constant 045 to the address of the transfer switch.

INSTR. LOCATION	INSTRUCTION OPER.	INSTRUCTION ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00019	RAD	05968	02			a0174	+
00024	UNL	00139	02				
00074	SET	00001	00				
00079	LOD	06006	00				
00084	UNL	05981	00				
00089	CMP	05980	00				
00094	TRH	00129					
00099	CMP	05978	00				
00104	TRH	00144					
00109	RAD	05977	00				
00114	ADM	00139	00				
00119	SGN	05981	00				
00124	LOD	05981	00				
00129	MPY	05999	00				
00134	ADM	00139	00				
00139	TR	( )					
00144	CMP	05979	00				
00149	TRH	00164					
00154	RAD	05974	00				
00159	TR	00114					
00164	RAD	05971	00				
00169	TR	00114					
00174	TR	to code	0				
00179	TR	to code	1				
00184	TR	to code	2				
00224	TR	to code	A				
00269	TR	to code	J				
00314	TR	to code	S				
00349	TR	to code	Z				

Figure 64. Program, Decoding Single Characters

00119. Remove the zone from the transaction code which was stored at 05981. The numerical position remains in memory.

00124. Load the numerical position of the transaction code character.

00129. Multiply by 005.

00134. Add the result to the transfer switch.

00139. Transfer switch.

00144. Compare the transaction code against the letter R.

00149. The code is between S and Z. Transfer to get the constant 130.

00154. Get the constant 090.

00159. Transfer to add the constant to the transfer switch.

00164. Get the constant 130.

00169. Transfer to add the constant to the transfer switch.

00174-00349. Transfer to the routine for transaction codes.

Binary Search

The following problem illustrates a method of searching which approximates the identical filing procedures. In such a filing or searching operation, the search through any given file or table usually begins at some index point nearest to a particular item. From this point, a trial search is made at the center of a group between index points.

For example, in manually searching for an item in a file cabinet of punched cards, the first search is in the drawer indexed to include the item. The second step would be to examine a particular card at about the center of the drawer in relation to the item being looked for. If the card examined in the drawer is higher than the item looked for, it may be assumed that the desired item is in the lower section of the drawer (assuming that the cards in the drawer are in ascending sequence by the item control field). Conversely, if the card examined is lower than the item looked for, the desired item is in the upper section of the drawer.

In the 705, the index point may specify a drum section or sections. The desired portion of the table may then be "pulled out" into memory to make the detailed search. However, instead of beginning the search with the first item of the table in memory, the program may direct the machine to proceed directly to the item at the center. A comparison is then made to determine whether the desired item is in the upper or lower half of the table.

When it has been determined which half of the file contains the item being searched for, this portion of the file is again divided in half and the process is repeated. The machine successively reduces the area of search by halves until the item is located.

The following formulas express this method of search.

N_0 is the number of items in the table. N_1, N_2, N_3, and so on, represent the location in the table of the successive search operations. To adapt the formula for machine use, several facts about the table must be established.

1. The reference portion of the items in the table must be in ascending sequence.
2. The length of all the items in the table, including the reference facts, should be equal. The search in memory is conducted by modifying the addresses of various instructions. This modification must be by a fixed number of memory positions.
3. The number of factors in the table must be known in order to reserve the proper space in memory for a given portion of the table.
4. The address in memory of both the first and last items in the table must be known in order to limit the search to the specific portion of memory in which the table is located.

Figure 65 shows the arrangement of eight items in memory as a simple table. Using the binary search

Table

Mem. Add.	00399		00401		00403		00405		00407		00409		00411		00413		00415
Mem. Pos.	b	0	+8	1	+2	1	+4	1	+6	2	+1	2	+6	3	+2	3	+4

Constants

Mem. Add.		00500	00501		00503				00507		00509		00511		00513				00517				00521
Mem. Pos.	b	+9	+5	0	+2	0	3	9	+9	1	+4	0	−1	0	1	0	4	0	+1	0	4	1	+5

1. $\frac{N+1}{2} = N_1$ (Rounded)
   $\frac{8+1}{2} = 4.5$ or 5
   Length = −2
   Calculated Address = 00409

2. $\frac{N_1}{2} = N_2$ (Rounded)
   $\frac{5}{2} = 2.5$ or 3
   Length = +2
   Calculated Address = 00403

3. $\frac{N_2}{2} = N_3$ (Rounded)
   $\frac{3}{2} = 1.5$ or 2
   Length = −2
   Calculated Address = 00407

4. $\frac{N_3}{2} = N_4$ (Rounded)
   $\frac{2}{2} = 1$
   Length = +2
   Calculated Address = 00405

Instr. Location	1. Oper.	1. Address	1. Stor. Code	1. Accumulator 00	1. Sign	2. Oper.	2. Address	2. Stor. Code	2. Accumulator 00	2. Sign	3. Oper.	3. Address	3. Stor. Code	3. Accumulator 00	3. Sign	4. Oper.	4. Address	4. Stor. Code	4. Accumulator 00	4. Sign
00004	SET	00004	01	axxxx																
00009	RAD	00500	00	a9	+	RAD	00500	00	a5	+	RAD	00500	00	a3	+	RAD	00500	00	a2	+
00014	CMP	00513	00			CMP	00513	00			CMP	00513	00			CMP	00513	00		
00019	TRH	00029				TRH	00029				TRH	00029				TRH	00029			
00024																				
00029	MPY	00501	00	a45	+	MPY	00501	00	a25	+	MPY	00501	00	a15	+	MPY	00501	00	a10	+
00034	RND	00001	00	a5	+	RND	00001	00	a3	+	RND	00001	00	a2	+	RND	00001	00	a1	+
00039	ST	00500	00			ST	00500	00			ST	00500	00			ST	00500	00		
00044	MPY	00503	00	a010	+	MPY	00503	00	a006	−	MPY	00503	00	a004	+	MPY	00503	00	a002	−
00049	ADM	00507	00			ADM	00507	00			ADM	00507	00			ADM	00507	00		
00054	LOD	00507	01	a00409	+	LOD	00507	01	a00403	+	LOD	00507	01	a00407	+	LOD	00507	01	a00405	+
00059	CMP	00517	01			CMP	00517	01			CMP	00517	01			CMP	00517	01		
00064	TRH	00094				TRH	00094				TRH	00094				TRH	00094			
00069																				
00074						SGN	00503	02	a−	−										
00079						SGN	00513	02	a&	+										
00084						ADM	00503	02												
00089						TR	00009													
00094	CMP	00521	01			CMP	00521	01			CMP	00521	01			CMP	00521	01		
00099																				
00104	UNL	00114	01			UNL	00114	01			UNL	00114	01			UNL	00114	01		
00109	SET	00002	00		+	SET	00002	00			SET	00002	00			SET	00002	00		
00114	LOD	(00409)	00	a21	+	LOD	(00403)	00	a12		LOD	(00407)	00	a16	+	LOD	(00405)	00	a14	+
00119	CMP	00509	00			CMP	00509	00			CMP	00509	00			CMP	00509	00		
00124	TRH	00139									TRH	00139								
00129																TRE	00164			
00134						TR	00074													
00139	SGN	00503	02	a&	+						SGN	00503	02	a&	+					
00144	SGN	00511	02	a−	−						SGN	00511	02	a−	−					
00149	ADM	00511	02								ADM	00511	02							
00154	ADM	00503	02								ADM	00503	02							
00159	TR	00009									TR	00009								
00164																				

Figure 65. Binary Search

formula, locate the number 14. The necessary constants are as follows:

CONSTANT DATA	EXAMPLE	MEMORY ADDRESS
Number items in table plus one ($N_0 + 1$)	$b\overset{+}{9}$	00500
Constant .5	$\overset{+}{5}$	00501
Length of items	$0\overset{+}{2}$	00503
Calculating address for searching the table	$039\overset{+}{9}$	00507
Number to be located in the table	$1\overset{+}{4}$	00509
Minus one	$0\overset{-}{1}$	00511
Constant one	01	00513
Address of first item	$040\overset{+}{1}$	00517
Address of last item	$041\overset{+}{5}$	00521

Program, Binary Search

Figure 65 is the program to locate the number 14 in the table shown stored in memory between locations 00400 and 00415, inclusive. Four searches are necessary to locate the number. The contents of the various storage units are shown during each calculation of N and the comparison of 14 with the number at location N.

Figure 66 is the complete program written in the normal manner.

INSTR. LOCATION	INSTRUCTION OPER.	ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
00004	SET	00004	01				
00009	RAD	00500	00				
00014	CMP	00513	00				
00019	TRH	00029	00				
00024	HLT	00001					
00029	MPY	00501	00				
00034	RND	00001	00				
00039	ST	00500	00				
00044	MPY	00503	00				
00049	ADM	00507	00				
00054	LOD	00507	01				
00059	CMP	00517	01				
00064	TRH	00094					
00069	TRE	00104					
00074	SGN	00503	02				
00079	SGN	00513	02				
00084	ADM	00503	02				
00089•	TR	00009					
00094	CMP	00521	01				
00099	TRH	00139					
00104	UNL	00114	01				
00109	SET	00002	00				
00114	LOD	( )	00				
00119	CMP	00509	00				
00124	TRH	00139					
00129	TRE	00164					
00134	TR	00074					
00139	SGN	00503	02				
00144	SGN	00511	02				
00149	ADM	00511	02				
00154	ADM	00503	02				
00159	TR	00009					
00164							

Figure 66. Program, Binary Search

00004. Adjust ASU 01 to four positions.

00009. Reset and add N.

00014. Compare N to 1.

00019. N is higher than 1; continue the program.

00024. N is equal to 1. This condition indicates that the number being searched for is not in the table.

00029. Get one half of N.

00034. Round N one position.

00039. Store N for next calculation.

00044. Multiply N by the length factor, in this case 02. The result is the number of memory positions from the beginning address of the table when N is located.

00049. Add to beginning address. The result is the actual address in memory where the search is to be made. For the first search, this is 00409; for the second, 00403 and so on.

00054. Get the calculated address in ASU 01.

00059. Compare the calculated address against the lower limit of the table. The search cannot be made beyond the limts of the table in memory.

00064. Calculated address is higher than lower limit. Transfer to compare against upper limit.

00069. The calculated address is equal to the first address in the table. Transfer to compare the number.

00074, 00079, 00084. Change the length field to plus. The sign of the length field controls the search up or down the table.

00089. Transfer to recalculate N.

00094. Compare the calculated address against the upper limit of the table.

00099. The calculated address is beyond the upper limit of the table.

00104. Unload the calculated address in a load instruction.

00109. Adjust accumulator storage to two positions.

00114. Get the number at the calculated address.

00119. Compare the number at the calculated address against 14.

00124. The number is higher than 14. Transfer to change the length field to minus and search lower in the table.

00129. Number 14 is located in the table. Transfer to continue the program.

00134. The number is lower than 14. Transfer to change the length field to plus and search higher in the table.

00139, 00144, 00149, 00154. Change the sign of the length field to minus.

00159. Transfer to recalculate N.

00164. (Start of routine when number is located.)

Variable Field Lengths within Records on Tape

In many business applications using sequential master files, the time spent in computing is appreciably outweighed by the time spent reading and writing tapes. This is especially true in those applications in which there are comparatively few transactions for a voluminous master file. Where 705 systems are used in such applications, they function as fixed field-length (for corresponding fields in a set of records), variable record-length systems. These can, however, be made to function as variable field-length, as well as variable record-length, systems at the expense of the execution of a few extra instructions on those records which are active and do have to be processed. The purpose of using such a method is to decrease considerably tape reading and writing times in the main program and also in any related programs and auxiliary operations. The following method will make this change.

Each record on tape consists of a fixed-length identifying field (such as a serial number, part number, and so on), a fixed-length control field, and then the variable length fields which make up the record. These variable fields are stored one after another with no blank spaces necessary. In most cases, some fields will be of fixed length by the nature of their contents. If certain fields do not exist in some record, they may simply be omitted but must be indicated in the control field by $\overset{+}{0}$ or $0\overset{+}{0}$, depending on the range of field length. The identifying field would be used to identify the record and sequence-check it, and to determine whether or not the record is active.

The fixed-length control field would specify the number of digits in each variable field. For example, a control field of $\overset{+}{8}1\overset{+}{3}0\overset{+}{0}0\overset{+}{4}$ would show that there are eight characters in the first field, 13 in the second, none in the next, and four in the last. These digits are used to set a storage unit, through which to transfer the compressed data to the applicable work area. If the control field specifies a $\overset{+}{0}$ or $0\overset{+}{0}$ for some particular field, then the field in question has been omitted from the record currently being processed. To process a file by this system, three areas are set up in memory:

a. A read-in area to receive the compressed data.
b. A work area which allows for all fields, each at maximum length.
c. A write-out area to contain the updated compressed data.

The work area is filled with zeros at the start of the program and re-initialized after processing of each active record.

Setting storage for the exact number of characters in the field insures that only pertinent data are transferred to the work area.

Before writing the processed record on tape, the procedure needs to be reversed to accommodate the field lengths which have been changed during processing. This reversal of procedure is accomplished by transferring each maximum-length field from the work area to a storage unit and then using a normalize and transfer instruction to count the number of "insignificant" zeros in the field. By subtracting the total number of these zeros from the known maximum field length, the number of "significant" digits in the field can be determined. This procedure will also permit the calculation of the address in the write-out area in which to place the processed item, so that it follows immediately upon the preceding item with no blank spaces being left between them.

If any changes in field lengths are made during the processing of an active record, then it is also necessary to make the related changes in the control field.

In handling a tape with the characteristics mentioned earlier, i.e., very few transactions to a voluminous master file, it is readily apparent that the tape reading and writing time which can be saved by elimination of excess record space will frequently substantially outweigh the extra processing time required on the active records. In other applications (if, for instance, the processing is computer-bound because of a preponderance of computing time), it is equally apparent that the savings in time realized through compression of data on tape files may not outweigh the time lost through the extra processing required, apart from the additional programming effort necessary.

To determine the feasibility of using this system on any application, certain points should be taken into consideration:

1. The reading and writing time saved on the average per record and the computing time saved, or lost, on the average per record. (Net computing time saved or lost must be considered, which is the time needed for the additional instructions less the saving in time due to the processing of shorter fields).
2. The programming effort involved. Whether programming is done in symbolic notation or in autocoder, a fixed routine to handle variable length fields may be developed and incorporated in all applicable programs.
3. Other programs using the same tape file. Before a decision can be made whether or not this is a feasible and time-saving feature, all programs using the tape-file in question have to be evaluated.

4. The effect on associated auxiliary operations, i.e., tape-to-printer, tape-to-card, and card-to-tape. Particularly in printed reports inclusion of the control field and lack of columnar consistency may not be desirable.

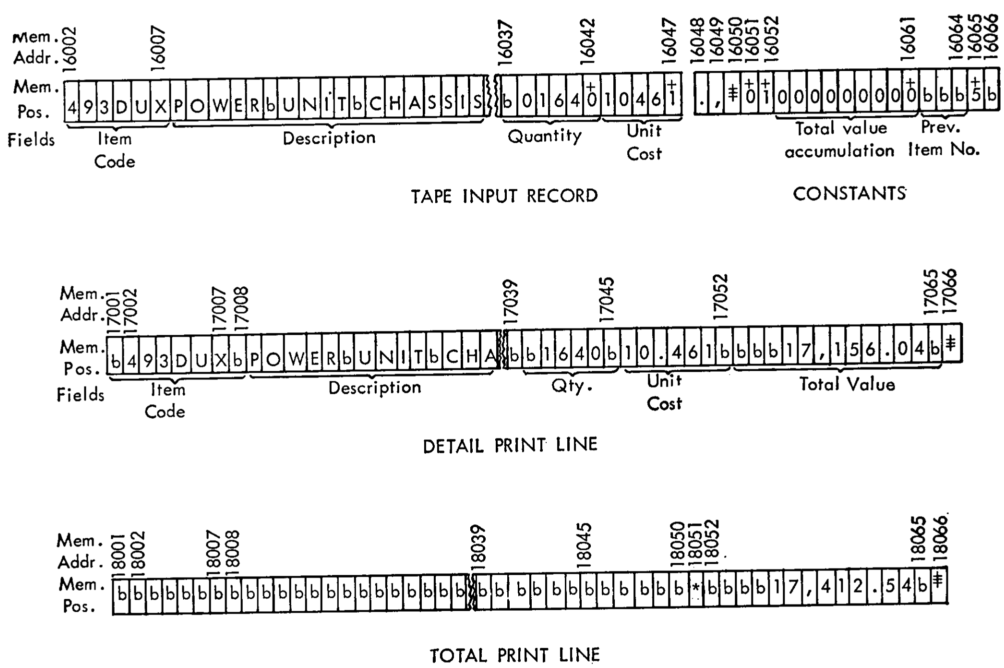
Certain intangible benefits of using this system are also evident, e.g., a smaller amount of physical tape handling and tape storing, as a given file can be written on few tape reels. On the other hand, blocking of records may be more difficult and may require extra computer instructions.

Record Arrangement for Printing

In normal programming, the report form must be considered and fields arranged in memory before a write instruction is given. One position in the record is allowed for each print wheel space of the printed line. Insignificant zeros to the left of digits in arithmetic fields are normally changed to blanks before printing by use of the store-for-print instruction. Indicative fields, descriptions, or other portions of a record can be shifted in memory to conform to the printing arrangement by use of the load and unload instructions. Amount fields from accumulator storage are stored in memory for printing by the store-for-print instruction.

Store-for-Print Problem

The tape record shown in Figure 67 is read into memory from tape unit 00209. The record is prepared for printing in a print line area with one space between each field to fit a predetermined form layout. Each tape record is listed on the report form. Commas and decimal points from the constant data area are placed in the print line to punctuate properly the value and unit cost fields. Commas must be restored after each record has been printed since they are replaced by blanks if there are no significant digits to the left.



COST REPORT				
Item	Description	Qty.	Unit Cost	Value
493DUX	POWER UNIT CHASSIS	1640	10.461	17,156.04
493DUR	MOTOR GENERATOR	10	25.650	256.50
			*	17,412.54
494DUX	POWER CABLE ASSEMBLY	2419	1.168	2,825.29

Figure 67. Store for Print

The first three digits of item number from the tape record are checked for sequence. A change in item number indicates that a total value should be printed. A total print line is indicated with the character * inserted to identify this line on the report form. Total value is accumulated in a separate area of memory until ready for printing.

It is assumed that all memory positions have been cleared to blanks before the program is loaded into the machine. It is also assumed that all necessary signs, symbols and constants are placed in memory by the program loading routine. The asterisk for designating a total can be placed in any location.

Program, Store for Print

Figure 68 is the program for the store-for-print problem.

Housekeeping

00004, 00009, 00014, 00019. Place the group mark in the print line areas.

00024, 00029, 00034, 00039. Place the decimal points in the print line areas.

00044. Adjust ASU 02 to three positions.

00049. Adjust ASU 03 to six positions.

00054. Adjust ASU 04 to 30 positions.

00059. Place a comma in ASU 01.

00064. Place a zero in ASU 06.

00069. Place nine zeros in ASU 06.

00074. Reset the total accumulation area to zero.

00079, 00084, 00089, 00094. Place commas in the print line areas.

00099. Select the input tape unit.

00104. Read the tape record into memory.

00109. Reset and add quantity into accumulator 00.

00114. Store quantity in the detail print line.

00119. Multiply: Quantity × unit cost = total value.

00124. Adjust total value to the nearest cent.

00129. Store total value in the detail print line.

00134. Add the total value to the accumulated total at memory location 16061.

00139. Reset and add unit cost in accumulator 00.

00144. Store unit cost in the detail print line.

00149, 00154. Place item code in the detail print line.

00159, 00164. Place description in the detail print line.

Test for total line

00169. Load the first three digits of item code in ASU 02.

00174. Compare the item code against the previous item code.

00179. Test for equal item code. Transfer to restore commas in the detail print line and repeat the program.

00184. Test for high item code. Transfer to print the total line.

00189. Low item code, error stop.

Print detail line

00194. Select the printer.

00199. Write the detail line.

00204. Unload item code.

00209. Transfer to replace commas.

Print total line

00214. Unload the item code for comparison with the next record.

INSTR. LOCATION	INSTRUCTION OPER.	INSTRUCTION ADDRESS	STOR. CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
	Housekeeping						
00004	SET	00001	01			ax	
00009	LOD	16050	01			a‡	
00014	UNL	17066	01				
00019	UNL	18066	01				
00024	LOD	16048	01			a.	+
00029	UNL	17048	01				
00034	UNL	17062	01				
00039	UNL	18062	01				
00044	SET	00003	02			axxx	
00049	SET	00006	03			axxxxxx	
00054	SET	00030	04			axx....x	
00059	LOD	16049	01			a,	+
00064	RAD	16051	06			a0	+
00069	SET	00009	06			a000000000	+
00074	ST	16061	06				
00079	UNL	18054	01				
00084	UNL	18058	01				
00089	UNL	17054	01				
00094	UNL	17058	01				
	Read, Calculate and Assemble						
00099	SEL	00209					
00104	RD	16002					
00109	RAD	16042	00	a01640	+		
00114	SPR	17045	00				
00119	MPY	16047	00	a0017156040	+		
00124	RND	00001	00				
00129	SPR	17065	00				
00134	ADM	16061	00				
00139	RAD	16047	00				
00144	SPR	17052	00				
00149	RCV	17002					
00154	TMT	16002	03				
00159	RCV	17009					
00164	TMT	16008	04				
	Test for Total Program after First Record						
00169	LOD	17004	02			a493	+
00174	CMP	16064	02				
00179	TRE	00194					
00184	TRH	00214					
00189	HLT	00001					
	Print Detail Line						
00194	SEL	00400					
00199	WR	17002	00				
00204	UNL	16064	02				
00209	TR	00089					
	Print Total Line						
00214	UNL	16064	02				
00219	RAD	16061	00	a001741254	+		
00224	SPR	18065					
00229	SEL	00400					
00234	WR	18002					
00239	TR	00074					

Figure 68. Program, Store for Print

00219. Reset and add the accumulated total value in accumulator 00.

00224. Store the total value in the total print line.

00229. Select the printer.

00234. Write the total line.

00239. Transfer to reset the accumulated total area and restore commas in both the detail and total lines. Repeat the program.

Sign Storage for Auxiliary Storage Units

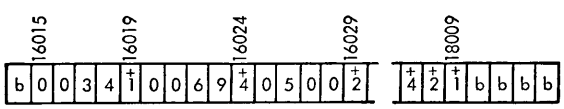
Because all auxiliary storage units share the same sign indicator, the sign of a result developed in one unit may be changed by subsequent calculation involving other units.

Figure 69 illustrates a method of storing the sign of an ASU in memory where it can be referred to at any point later in the program. Memory location 18009 is reserved for the storage of the sign of ASU 07 after the execution of a subtract instruction. Any available location can be used to store the sign as a plus or minus over any single digit. In Figure 69, the operation is begun with a plus one in memory position 18009.

00604. Reset and add the plus one in ASU 15. The sign trigger for all auxiliary storage units is now plus, including ASU 07.

00609, 00614, 00619. Obtain a result in ASU 07 by adding and subtracting the factors shown in memory. The sign trigger is now set to minus because there is a minus factor in ASU 07.

00024. Store the one in ASU 15 in memory location 18009 with the minus sign developed by the subtract operation in ASU 07. The condition of the sign trigger is now stored for use later in the program.



INSTR LOCATION	INSTRUCTION		STOR CODE	ACCUMULATOR 00	SIGN	AUXILIARY STORAGE 01-15	SIGN
	OPER	ADDRESS					
00604	RAD	18009	15			a1	+
00609	RAD	16019	07			a00341	+
00614	ADD	16024	07			a01035	+
00619	SUB	16029	07			a03967	-
00624	ST	18009	15			a1	-

Figure 69. Sign Storage, Auxiliary Storage Units

IBM 705 Autocoder System

The IBM 705 Autocoder system is a body of instructions to the 705 which makes possible a greatly simplified technique for instructing the machine.

Before there were any simplified programming techniques, a programmer had to write his instructions in *actual* machine language. This method of programming is difficult, time-consuming, and subject to clerical error. Moreover, it requires monumental effort to keep track of the proper address portion of the instruction. A program of, say, 3000 instructions written in this manner is almost impossible to read or to analyze for corrections and additions.

Any method of programming other than actual programming requires an *assembly* program. This is a program that converts, or assembles, the programmer's symbolic instructions into actual instructions. In other words, the programmer writes a program in a convenient way, using meaningful *symbols,* and then writes an assembly program to convert these symbols into actual (machine language) instructions.

The IBM 705 Autocoder System is actually an improved method of symbolic programming. The basis of this system is the autocoder assembly program. Because of its features, the autocoder is often referred to as an *automatic* coding system.

The most distinguishing feature of the autocoder system is its ability to handle "macro-instructions."

A macro-instruction (special autocoder instruction) permits one instruction to take the place of several 705 instructions. The macro-instruction incorporates into the program a group of 705 instructions previously arranged to perform such routines as end-of-file, error correction, and so on.

This feature is valuable because of the frequency with which programmers want to use a particular series of instructions in one program or from one program to the next. The autocoder system makes it possible for them to write it once, enter it into their autocoder "library" and then add the routine to their programs whenever they want to, by use of a single macro-instruction.

A most important feature of the autocoder system is the reduction in coding or clerical errors. It would not be unusual for an autocoder programmer to obtain an assembled program (in machine language) of 2000 instructions as the result of writing only 500 instructions. The autocoder provides 1,500 instructions that are free from clerical errors. These 1,500 instructions were completely checked out before they became a part of the autocoder library. The autocoder also systematically checks the instructions written by the programmer. Upon detection of errors, the programmer is notified by typed messages during the assembly process. In some cases, the autocoder diagnoses the *intention* of the programmer and *corrects* the error.

When fully used, the 705 autocoder system makes an important contribution to commercial programming in reducing coding time and effort. Programming experience and ingenuity may eventually bring forth other more sophisticated systems, placing more emphasis on pseudo-instructions and subroutines. The IBM 705 autocoder system, however, is a proven system in use at the present time. It represents an invaluable contribution which must not be overlooked in planning for a 705 data processing system.

IBM 782 Console Control

IBM 705 I or II Console

The operator's console (Figure 70), a separate unit of the 705 system, may be placed conveniently in the installation within input-output cable length restrictions. The console is used to:

1. Control the machine manually.
2. Correct errors.
3. Determine the status of 705 circuits, registers, and counters.
4. Determine the contents of memory and accumulator or auxiliary storage.
5. Revise the contents of memory.

The lights and keys on the console are shown in Figure 71 and their operation or function is explained in the following section.

1. Operation Decoder

There are 36 neon lights in this section. Thirty-five of the lights represent the 35 machine instructions and are turned on as each instruction is executed. One light is not used. Both the instruction abbreviation and the operation part are printed beside each light. The binary code decimal notation on the left side represents the digits 1 through 9. The binary code across the top represents the zones. For example: The operation part for NOP is A. The binary code for A is zone 11 and numerical 0001.

2. Operation Register

The operation part of an instruction being executed is displayed in the 705 character code by neon lights in the operation register.

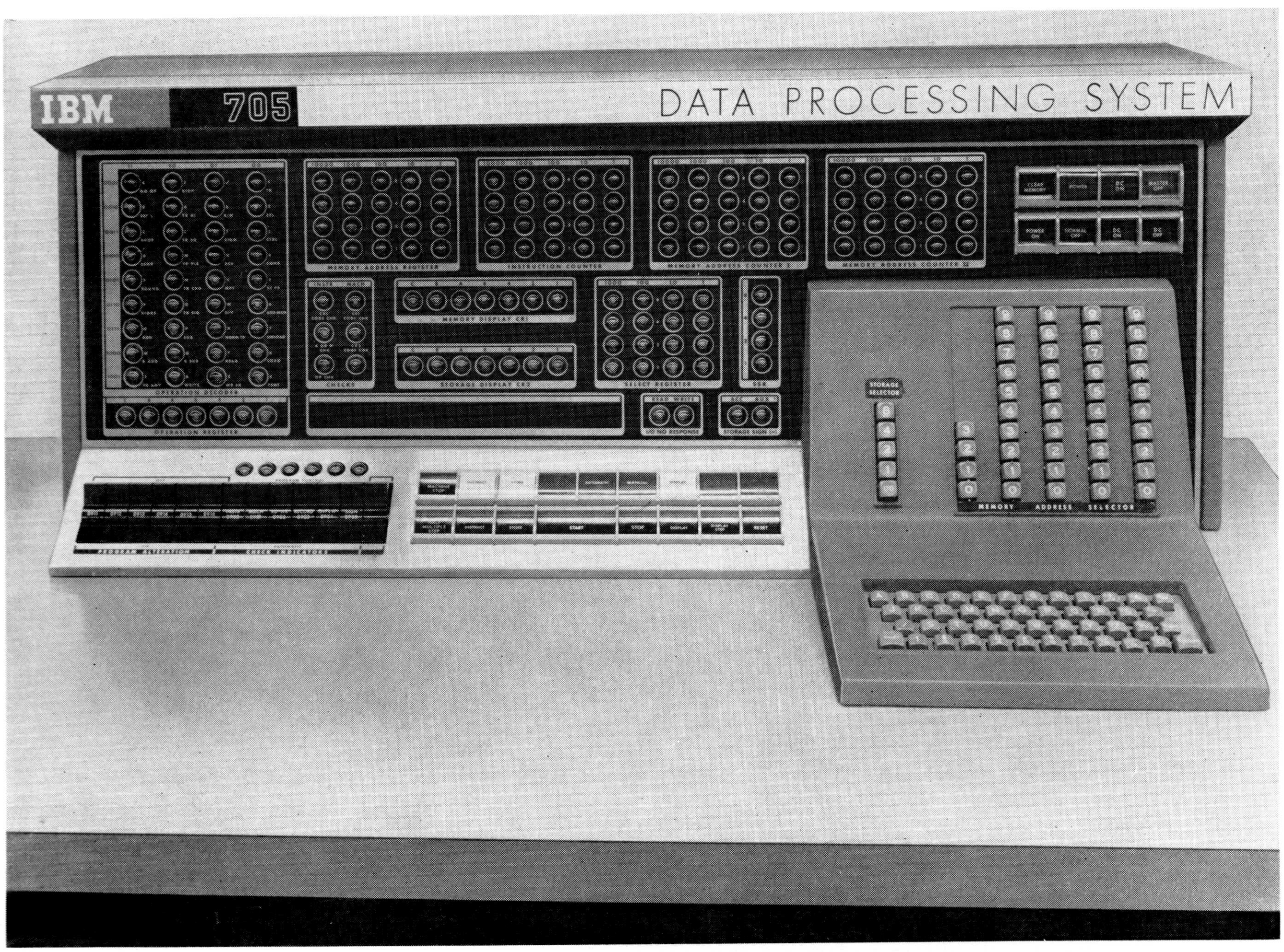


Figure 70. Operator's Console, IBM 705 II

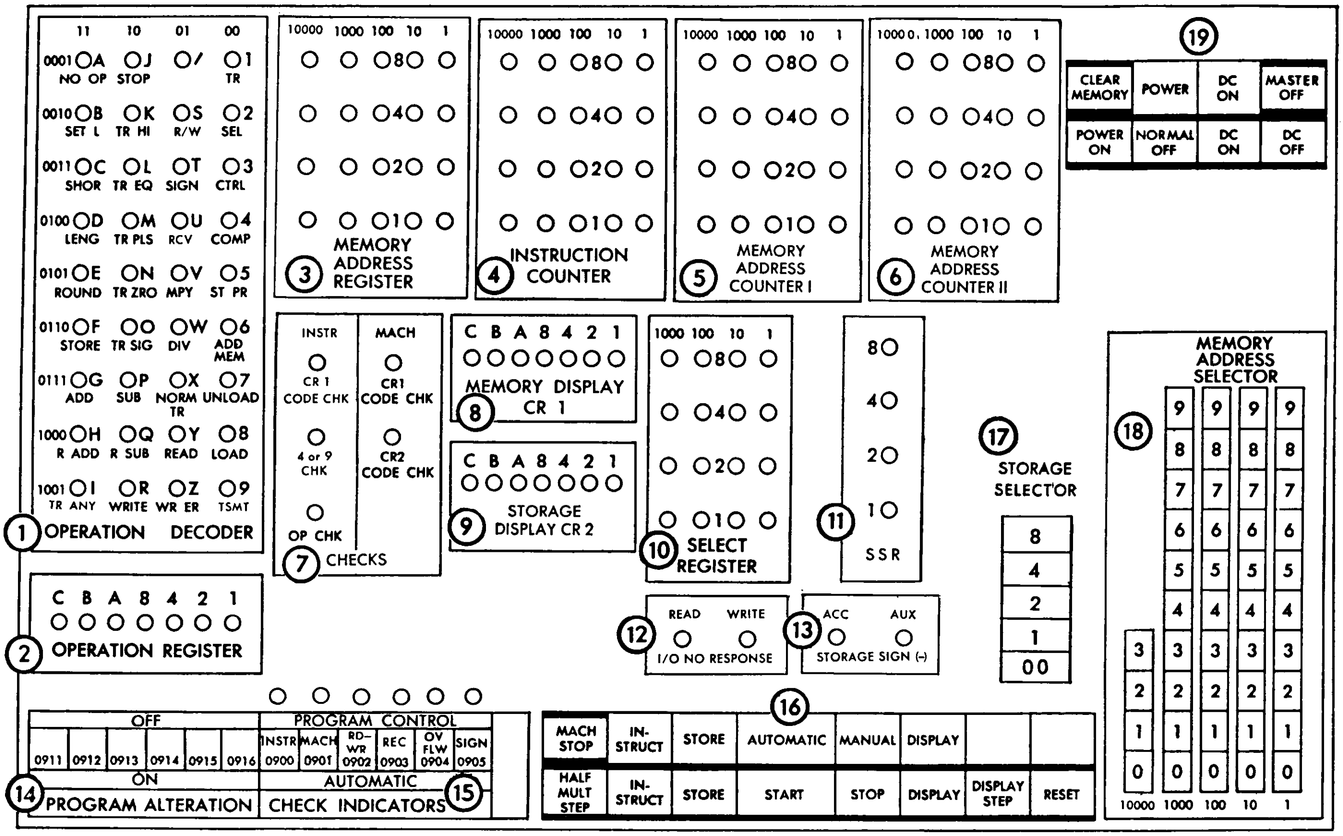


Figure 71. Schematic of Operator's Console, IBM 705 I or II

3. Memory Address Register (MAR)

The address part of the instruction is displayed by the memory address register neon lights. Twenty neon lights are used to display the address part of the instruction being executed.

4. Instruction Counter (IC)

The twenty neons in this section show the memory address contained in the instruction counter.

5. Memory Address Counter I (MAC I)

The twenty memory address counter neons display the address of the next character position to be operated upon in memory. The operation of the machine, while executing an instruction, is divided into two parts: (1) the reading and interpreting of the instruction (instruction time), and (2) the actual execution of the instruction (execution time).

The instruction counter and MAC I are both involved in the reading of an instruction. At the beginning of instruction time, the instruction counter is already set to the memory address of the right-hand digit of the instruction. At the end of the execution of the preceding instruction, memory address counter I is set to the address indicated by IC. During instruction time the address part of the instruction is read into MAR. The operation part of the instruction is read into the operation register.

At the end of instruction time, the instruction counter is stepped five addresses higher to the address of the right-hand digit of the next program instruction. It remains in this position during execution time.

During the normal preparation for executing an entire program, the reset key is depressed, setting the instruction counter to memory address 00004. When the start key is depressed, the sequence of operations just described occurs in reading the instruction.

The instruction counter may be manually transferred to the address of any instruction in memory. When this is done, the reading operations are the same except that they begin at the address of the right-hand digit of the instruction transferred to.

The machine automatically executes the instruction during execution time, performing different series of actions for different instructions. During execution time, MAC I is changed to the different addresses in memory as necessary to control the information read out of or stored in memory.

When a simultaneous reading and writing operation is performed, MAC I controls the information being written out of memory. When a receive and transmit operation is performed, MAC I controls the information being transmitted.

At the end of execution time, MAC I is set to the address position of the instruction counter, which has been positioned at the next instruction. The same operations described for instruction time are repeated for that instruction. The counter continues to count, instruction by instruction, to successively higher memory addresses until it reaches the end of the program or a transfer to a different series of instructions.

6. Memory Address Counter II (MAC II)

Twenty neon lights display the contents of memory address counter II. This counter operates in conjunction with MAC I to control information read into memory during a read-while-writing operation or to control information being received during a receive-and-transmit operation.

7. Checks

The check stop neon lights display the cause of a stop occurring because of an instruction check or a machine check.

INSTRUCTION CHECK LIGHTS

CR 1 Code Check. This light turns on when there is a character code error in the units position of the address part of the instruction or in any of the five characters in the memory buffer register.

4 or 9. This neon light turns on when the units position of the address part of any transfer instruction, or the transmit instruction specifying accumulator 00, is not 4 or 9.

Operation Check. This neon light turns on when the operation part of an instruction is any character which is not a valid code construction, or when the operation decoder does not function properly.

MACHINE CHECK LIGHTS

CR 1 Code Check. This light turns on where there is a character code error in the character last read from memory or in any one of the characters affected by a five-character transmit instruction.

CR 2 Code Check. This light turns on when there is a character code error in the character last read from storage.

8. Memory Display CR 1 (Character Register)

The seven lights in this unit indicate in 705 code form the character most recently read from memory. The address of the next character to be read is displayed in MAC I.

9. Storage Display CR 2 (Character Register)

The seven neon lights in this unit indicate in 705 code form the character most recently read out of storage.

10. Select Register (SR)

The sixteen neon lights on the select register display in 705 code form the number of the device last selected. The number is shown for that device until another device is selected.

11. Storage Select Register (SSR)

Four neon lights display the contents of the storage select register and indicate, in straight binary form, which one of the storage units has been selected for use while executing the current instruction.

12. I-O (Input-Output) No Response

Two neon lights, one for read and one for write, indicate that a no-response signal has been received from an input or output unit after it has been selected and has been told to read or write. The proper light goes on when the selected unit does not exist, when an addressed unit is not in "ready" status, or for conflicting instructions such as select 00100 and write, or select 00400 and read.

The machine stops in automatic status when any of these conditions exists. Depressing the machine stop key causes the machine to go from automatic to manual status.

13. Storage Sign (—)

Aux. The auxiliary storage sign neon lights when the sign of the field stored in the *last-used auxiliary storage unit is minus.* The use of other storage units may subsequently change the auxiliary storage sign to plus. In this case, the minus field in one or more auxiliary storage units is not indicated by the neon when the sign of the field in the last-used unit is plus.

Acc. The accumulator storage sign neon lights when the sign of the field stored in the accumulator is minus.

14. Alteration Switches

Each alteration switch has a specific address. If a switch is on when interrogated by the program, a transfer is made. If the switch is off, no action takes place. Six alteration switches are provided.

15. Check Indicators (Neons and Switches)

Six switches are provided, one for each of the check indicators. When a switch is turned to AUTOMATIC, the machine stops when the corresponding error occurs. When the switch is turned to PROGRAM, the error condition will be indicated, but the course of action is determined by the program. Each switch is associated with a light above it to indicate when the indicator is on.

00900—Instruction Check. If the switch is turned to AUTOMATIC and an error occurs, the 705 stops with MAC I indicating the address of the instruction in error.

00901—Machine Check. If the switch is set to AUTOMATIC, and an error occurs, the 705 stops with MAC I indicating the address of the error character.

00902—Read-Write Check and 00903—Record Check. If the switch is set to AUTOMATIC and an error occurs, the 705 stops after a complete record has been read or written.

00904—Overflow Check and 00905—Sign Check. If the switch is set to AUTOMATIC and an error occurs, the 705 stops with MAC I indicating the address of the character one position from that which was in error.

16. Operation Controls

Machine Stop. When an internal operation is being performed, depressing the machine stop key stops the 705 immediately. The stop can occur during the execution of an instruction such as an arithmetic or writing operation and, therefore, should be used as a last resort.

The machine stop key may be used to interrupt input-output operations or to stop the 705 when the manual stop key is not effective.

The machine stop key may also be used when a machine stop occurs and the 705 remains in an automatic status. The instruction causing this condition may be bypassed by depressing the machine stop key and then the start key.

Instruction (Key and Light). To instruct the machine manually, the procedure is:

1. Depress the instruction key. If the machine is in automatic operation, it stops after the current instruction is executed exactly as if the manual stop key had been depressed.
2. Key the address part of the instruction in the memory address selector (either before or after depressing the instruction key).
3. Key the desired storage unit in the storage selector.
4. Key the operation part of the instruction in the keyboard.

The instruction and manual lights go on upon completion of the instruction being executed. The instruction status continues until superseded by another mode of operation. Subsequent instructions may be keyed in without redepressing the instruction key.

Depressing the start key causes the machine to continue from the point in the program at which it was interrupted unless one or more transfer instructions are executed under manual control. If any check indicators (with the associated switch set to PROGRAM) were on, the error indication would be reset.

Store (Key and Light). To store information in memory manually, the procedure is:

1. Depress the store key. If the machine is in automatic operation, it stops after the current instruction is executed, exactly as if the manual stop key had been depressed.
2. Key the memory address where the first character of information is to be stored in the memory address selector.
3. Key the characters to be stored in the keyboard. The first character is entered into the memory position specified by the memory address selector. Subsequent characters are entered into successively higher memory positions.

The store and manual lights go on when the store key is depressed. After one or more characters have been stored, the operator may select another memory address on the memory address selector. Depressing the store key a second time causes the next character to be stored to go into the memory location specified by the second setting of the address selector.

The store status continues until superseded by another mode of operation.

Display (Key and Lights). Depressing the display key when the machine is running causes the machine to stop as though the manual stop key had been depressed. In addition, it prepares the machine for reading a character from memory and from storage, upon subsequent depression of the display step key.

The character read from memory is displayed on the memory display (CR 1) neons. If the character code check indicates an error in the memory character being displayed, the CR 1 code check neon lights but does not turn on the machine check indicator.

The first memory character displayed is the one located at the address specified by the memory address selector. Subsequent characters are displayed from successively lower memory positions.

The character read from storage is displayed in the storage display (CR 2) neons. If the character code check indicates an error in the storage character being displayed, the CR 2 code check neon lights but does not turn on the machine check indicator.

The first storage character displayed is the one located at the right-hand position of the storage unit selected by the storage selector switches. Successive depressions of the display step key display successive storage characters to the left.

The position of the next memory character to be displayed in the CR 1 neons is indicated by the MAC I neons.

Depressing the display key after the display step key has been depressed causes the next characters to be displayed from the addresses specified by the memory address and storage selector switches.

When the display key is depressed, the display light and the manual light go on.

Manual Stop (Key and Light). Depressing the stop key causes the machine to stop after executing the current instruction. The manual light goes on when the machine is stopped. This is the normal way to stop operation. After the stop key is depressed the machine is left in manual status and is prepared to respond to any manual function or to start automatic operation if the start key is depressed.

Certain other keys also cause the machine to stop before their special functions begin. These are:

Store key	Display key
Instruction key	Half-multiple step key

Half Multiple Step. Depressing this key causes the machine to operate in half-steps. One depression causes the machine to read an instruction. A second depression causes the machine to execute the instruction. If the key is held depressed for more than ¾ of one second, the machine alternately reads and executes instructions at the rate of about ten per second, as long as the key is held. If the machine is running when the key is depressed, it stops after executing the current instruction. Depressing the start key causes the machine to run at its normal speed, starting at the point at which is stopped after the half-multiple step key was last depressed.

Automatic Start (Key and Light). Depressing the start key, if the machine is stopped, causes it to run at its normal high-speed rate, starting with the instruction at the address indicated by the instruction counter neons. Operation continues as indicated by the automatic light until a programmed stop or an automatic check stop occurs, or until the machine is made to stop by depressing one of the control keys provided for this purpose.

Depressing the start key resets all of the check indicators off. If an instruction error has occurred (00900) and the switch associated with the indicator is set to AUTOMATIC, depressing the start key causes the machine to continue execution of the improper instruction. The instruction check light is then turned on again immediately. The operator should manually correct the instruction or transfer to another instruction before restarting the machine.

If a machine error has occurred (00901) with the machine check switch set to AUTOMATIC, the machine may stop before completely executing the instruction involving the error. Depressing the start key turns off the check indicator, but an attempt to complete the instruction turns it on again at once. A correction should be made before operation can continue.

If check indicator 00902, 00903, 00904, or 00905 is on with its associated switch set to AUTOMATIC, depressing the start key causes the machine to execute the next instruction. The indicator is reset off.

Depressing the start key when the machine is in automatic operation has no effect.

Reset. The reset key restores all checking circuits to normal and sets the instruction counter to 00004. These functions are also performed automatically whenever the power is turned on and the machine voltages have reached their required level. The reset key is operative only when the machine is stopped. It has no effect if depressed when the machine is in automatic operation.

17. Storage Selector

The storage selector keys are used to select the storage unit to be used while executing a manual instruction, or when displaying it. It consists of one group of five keys. Four of the keys represent four binary ones valued 1, 2, 4 and 8, and are so labeled. These four keys are used to select the auxiliary storage unit to be used. The selection is done in straight binary coding and a depressed key indicates a binary one in that position. Each of the four keys will remain down when depressed. The fifth key (labeled "00") is depressed to select the accumulator storage unit. Depressing this key causes the other four to release. Depressing any of the five keys while the machine is in automatic operation will have no effect.

18. Memory Address Selector

The memory address selector consists of four groups of ten keys each numbered from 0 through 9, and one group of four keys numbered 0 through 3. The first four groups are used to select the units, tens, hundreds, and thousands positions of the address and the last group is used to select the ten-thousands position. The memory address selector is used to set up the address in memory when performing the manual operations instruct, store, or display.

During manual instruct, the memory address selector is set up to the address portion of the instruction to be executed. During manual store, the address selector is set up to the address of the first character to be stored in memory. During manual display, the address selector is set up to the address of the first character in memory to be displayed.

Depressing the address selector keys while the machine is running in automatic operation has no effect.

A modified card punch keyboard (Figure 72) is used to enter data or instructions into memory while the machine is in store status. The operation part of an instruction is keyed on the keyboard while the machine is in instruction status. (Refer to instruction and store keys.)

Alphabetic characters on the keyboard are arranged in sequence from left to right in the top three rows of keys. The bottom row has numerical digits in the lower case position, and special characters in the upper case position. An upper case key and a test key are also provided.

19. Power Controls

Clear Memory. Depressing the clear memory key if the machine is stopped, causes memory to be filled with blanks (1 01 0000) and storage to be filled with storage marks (0 00 0000). All check indicators are turned off and the instruction counter is set to 00004. Depressing this key while the machine is in automatic status has no effect. These functions are also performed automatically whenever the power is turned on and the machine voltages have reached their required level.

Figure 72. IBM 705 II or III Keyboard

Power-On (Key and Light). Depressing the power-on key turns on the AC and DC voltages sequentially in the machine and turns on the power-on light. When the voltages are properly stabilized, the machine is automatically reset. The instruction counter is set to address 00004, all positions of storage units are reset to storage marks, all memory positions are reset to blanks, and all check indicators are turned off. The manual light is then turned on, indicating that the machine is ready for operation.

DC On-Off (Keys and Light). Depressing the DC-on key supplies DC power to the machine and lights the DC-on light. It also accomplishes a reset in the same manner as the power-on key. Depressing the DC-off key cuts off DC power to the machine. These keys are normally used for maintenance purposes only.

Master-Off. Depressing this key immediately cuts off all power to the entire machine, including the cooling system.

Normal-Off. Depressing this key turns off the AC-DC voltages sequentially in the machine. The machine cooling system continues to operate several minutes.

IBM 705 III Console

The 705 III console is shown in Figure 73 and the schematic in Figure 74. Only the differences between 705 I and II and the 705 III consoles are explained in this section. If a feature is not mentioned, its operation is the same for all 705 models.

1. Operation Decoder

There are 48 neon lights in this section, representing the machine instructions. The mnemonic code and operation part are printed next to each neon used.

Operation Register

These register neons are not used on the 705 III console.

2. Memory Address Counter I (MAC I)

MAC I operation is the same except that MAC I is not set to the address of the instruction counter.

3. Memory Address Selector

This selector consists of five groups of keys. By a combination of these keys, any address may be set

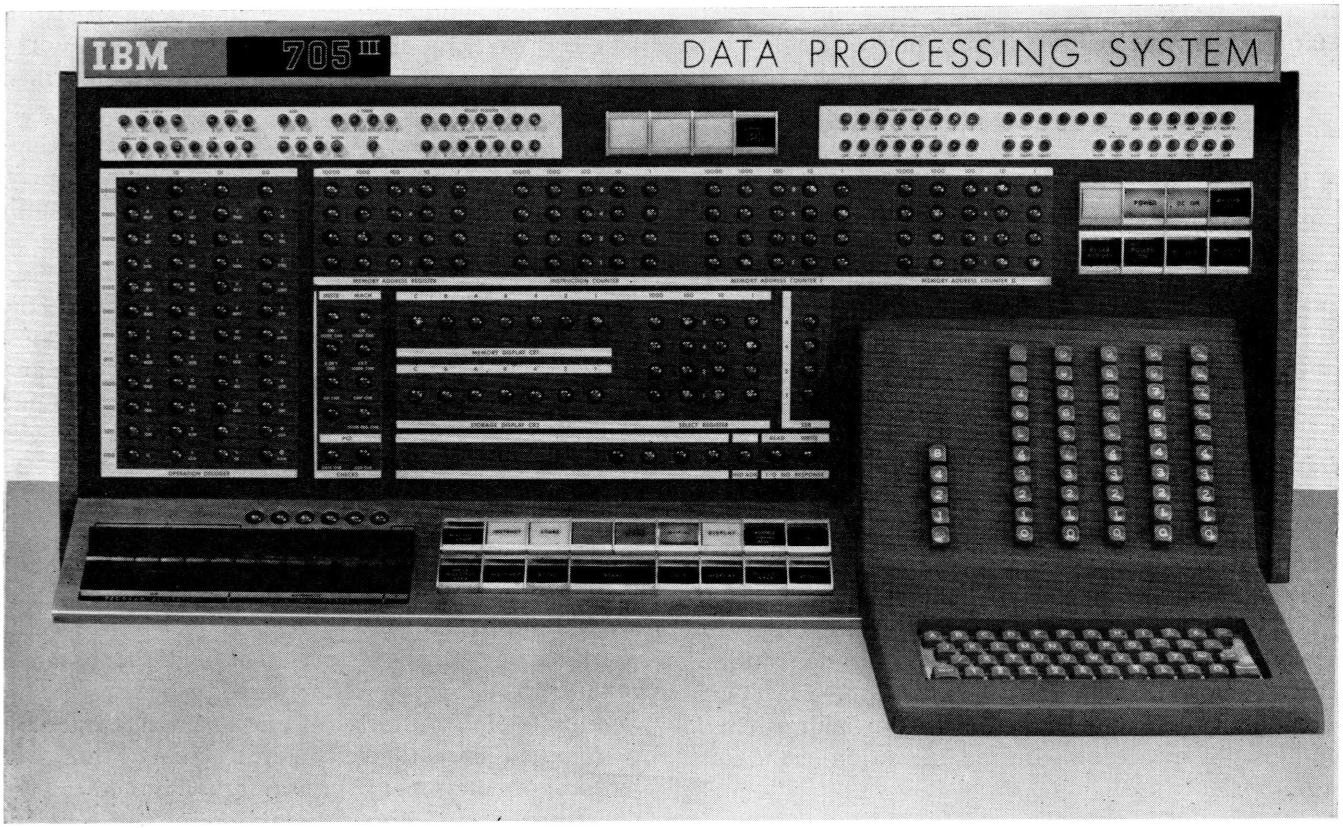


Figure 73. Operator's Console, IBM 705 III

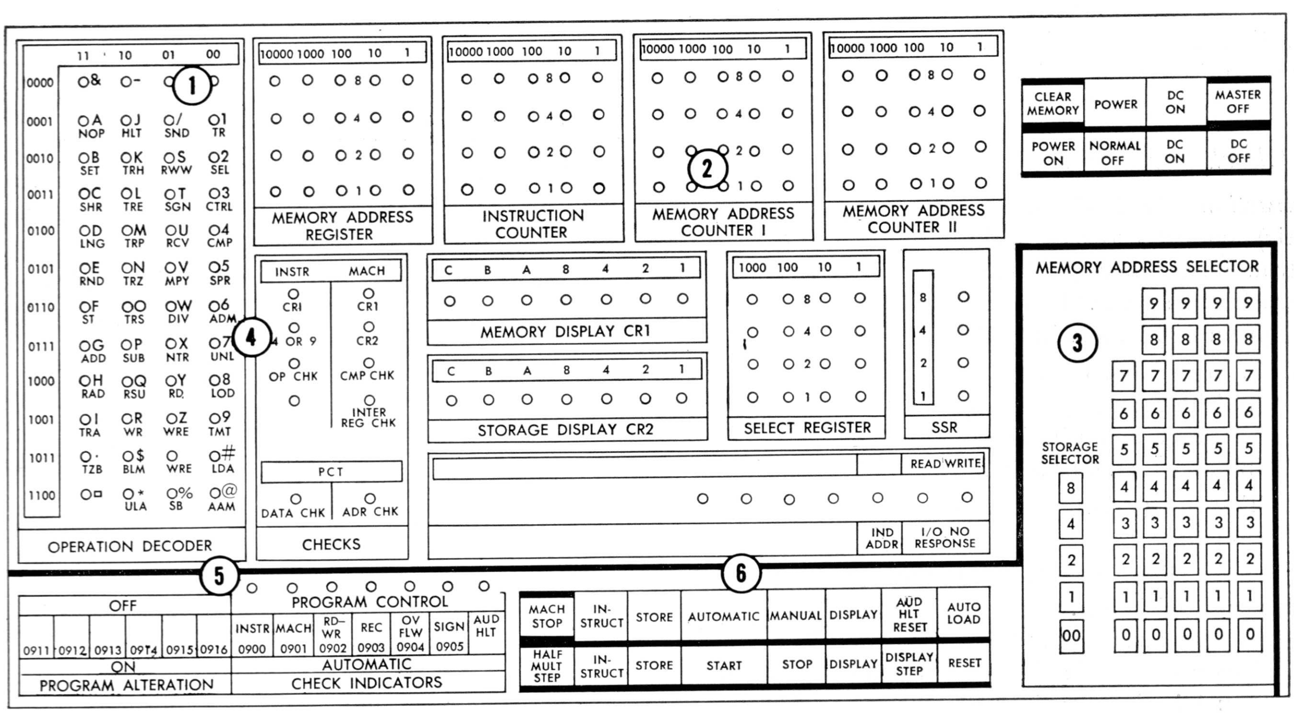


Figure 74. Schematic of Operator's Console, IBM 705 III

into the memory address selector when in a manual status in the same manner as with the 705 II.

4. Checks

4 or 9 Indicator. This neon turns on as with 705 I or II operation. In addition, the instructions SND, LDA, ULA, AAM, and the field addressed by an indirect address coded instruction must be located at memory positions ending in 4 or 9; otherwise, the indicator is turned on.

Compare Check. This indicator is turned on if, after a compare instruction execution, only one of the high, low or auxiliary indicators is on, or if all three indicators are on.

Inter-register Indicator. This indicator is turned on if any bits are changed during transmission of data between the character register and the result register.

PCT Address Check. The parallel character transmission address check indicator is turned on if any read address failure is sensed in the DS to CPU addressing. The CPU is subject to a manual stop. The operator must initiate a restart and the PCT address check indicator is turned off with the depression of the start key.

PCT Data Check Indicator. This indicator is turned on when any data error occurs in the data synchronizer, and will also be on with the PCT address check neon.

Storage Sign. This neon is placed in the top portion (customer engineering section) of the console but its operation remains the same as before with the 705 II.

Indirect Address Indicator. This neon is on during the execution of an indirect addressed instruction.

5. Check Indicators

Instruction Check 00900. If the switch is on (automatic) and an error occurs, the 705 stops, with the instruction counter (IC) indicating the address of the instruction in error. It operates in the same manner as with the 705 II, and in addition:

1. Is turned on, together with the indirect address indicator and the 4-or-9 indicator, if the field addressed by an indirect address instruction ends in other than 4 or 9 memory positions.
2. The instruction counter is stepped during execution time.

Machine Check 00901. The machine check conditions are listed below.

WITH THE SWITCH SET TO AUTOMATIC, if an error caused by a CR 1 code check occurs, the 705 will stop with MAC I indicating the address or the address less one of the error character. To find the error address, the operator should display the address set in MAC I. If it is correct, the next depression of the display key will give the next lower address, which should then be compared for error.

IF A STOP OCCURS FROM A CR 2 CODE CHECK, the character in error may be examined by manually displaying the proper ASU or accumulator storage unit until the error character is reached.

WITH A STOP CAUSED BY A COMPARE CHECK, depression of the start key continues operation. Again, the error may be ignored or examined by the operator. After depression of the start key, subsequent TRH and TRE instructions may not function properly because the respective indicators have not necessarily been set correctly.

WHEN AN INTER-REGISTER CHECK CAUSES A 00901 STOP, the instruction being executed determines where the error occurred. The operator should display the fields of data involved, comparing, character by character, until the error character is recognized. This character may then be corrected and the program restarted by depression of the start key.

Audible Halt Feature. Whenever the machine fails to complete execution of an instruction within four seconds, or enters manual status for reasons other than depression of the store, display, or instruct keys, an audible tone will be sounded at timed intervals.

Audible Halt Switch. When this switch is on, the audible halt feature is operative.

Audible Halt Reset Key. Depression of this key turns off the audible tone, if on. The tone will not sound again unless the start key is depressed and another audible halt condition occurs.

6. Operation Controls

Machine Stop. Be careful about depressing the machine stop key, because this key causes an immediate halt of DS operation, resulting in:

1. An incomplete record on tape while writing.
2. An incomplete record in memory while reading.
3. Interruption of the instruction being executed.
4. A reset of the PCT address neon and the data and record counter indicators.

Clear Memory. This key is operative whether the 705 is in a manual or automatic status. Depressing it does the following:

1. Clears memory to blanks and clears all storage units to storage marks.
2. Returns all check indicators to their normal operating status.
3. Sets the instruction counter to 00004.

Reset. Zero and comparison indicators are not reset.

Load Key. When depressed, the load key performs the following operations:

1. Selecting the input unit addressed by the keys of the memory address selector.
2. Performing the reset function (reset all check indicators and set the instruction counter at 00004).
3. Reading one record into memory, starting at location 00000.
4. Starting automatic operation.

Use of this key simplifies and speeds up the loading of a program into memory.

Stop Key. The stop key (not machine stop) does not operate on a RD, WR, WRE, TMT, or SND *instruction.* If the stop key is depressed during execution of one of these instructions, the 705 continues to operate until execution of an instruction (other than those mentioned) has been completed.

IBM 727 or 729 I and IBM 729 III Magnetic Tape Units

Functional Description

The magnetic tape units are used to record data on magnetic tape or to read data previously recorded on tape. The magnetic tape units used with the 705 systems are the 727, 729 I (Figure 75) and the 729 III (Figure 76). The 727 can be used in all operations requiring a tape unit except that of the 767 Data Synchronizer. The 729 I (when attached to the 705 system) and the 729 III can be used only with the 767 Data Synchronizer.

Except for a few milliseconds during starting or stopping, the magnetic tape on a tape unit is driven at a constant speed while reading, writing, or backspacing. The magnetic tape on the 727/729 I tape units is driven at the constant rate of approximately 75 inches per second. Data are recorded on tape at a density of 200 characters per inch. The character rate, a factor of tape speed and character density, is 15,000 characters per second. The magnetic tape on the 729 III tape unit is driven at the constant rate of approximately 112.5 inches per second, with a character density of 556 per inch and a character rate of 62,500 characters per second.

Figure 77 shows schematically the physical location of the tape when it is mounted on a tape unit. During reading or writing, tape is moved from the file reel through two vacuum columns, located on either side of the read-write head, and to the machine reel.

Reading or writing takes place as the tape is moved across the magnetic gap in the read-write head. The 727 tape unit has one magnetic gap in the read-write head which is used for reading or writing. The 729

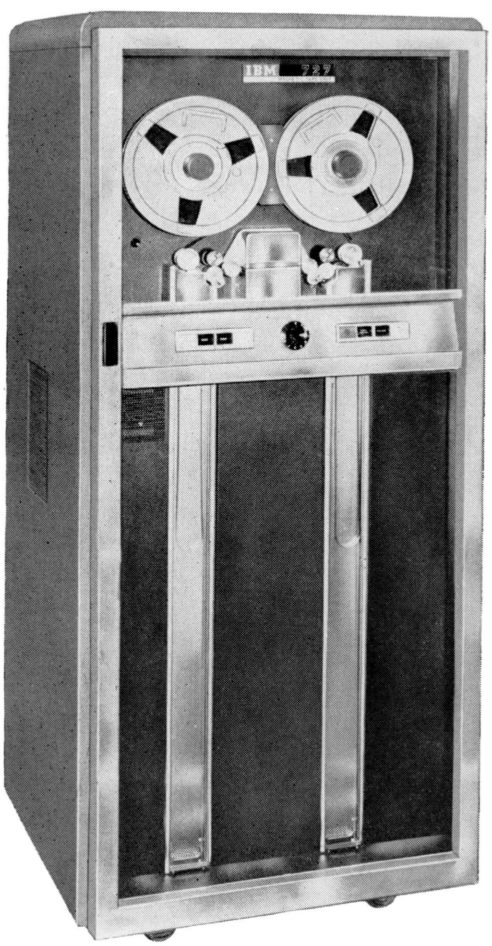


Figure 75. IBM 727 Magnetic Tape Unit

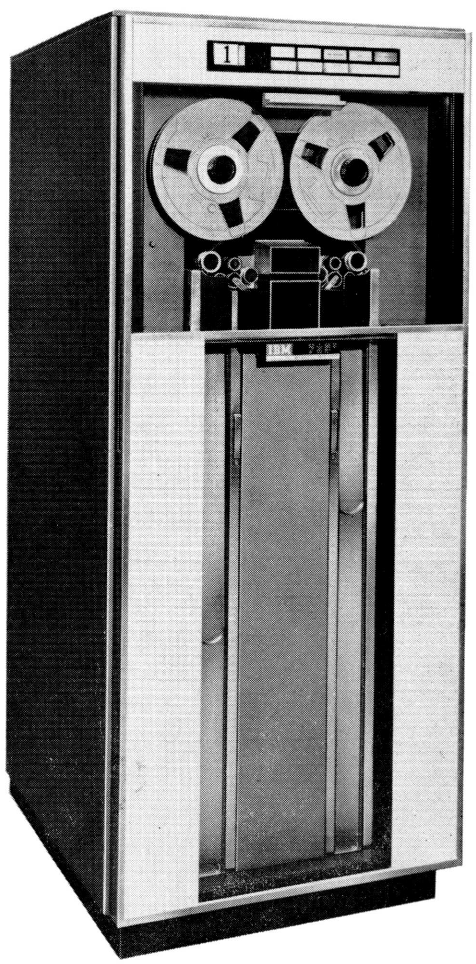


Figure 76. IBM 729 III Magnetic Tape Unit

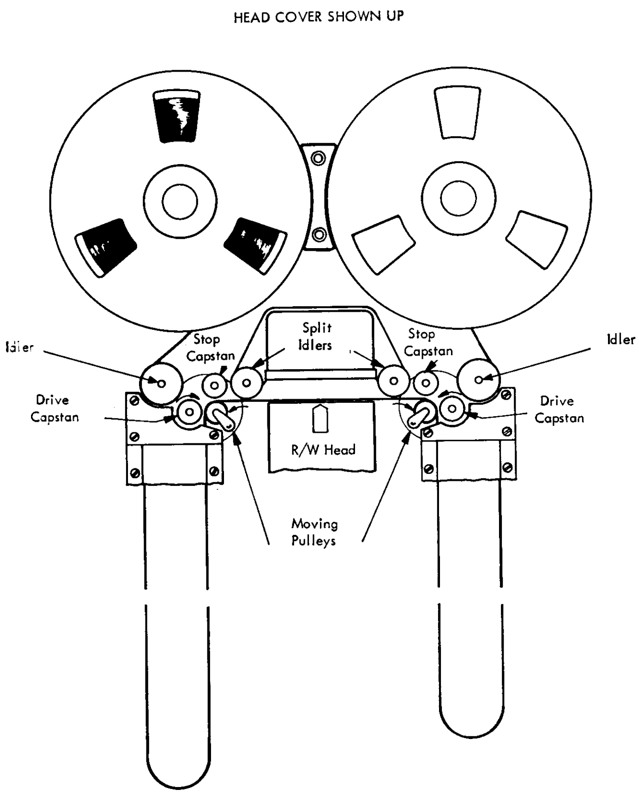


Figure 77. Schematic, Tape Unit Feed

I and 729 III tape units have two magnetic gaps in the read-write head. One gap is used for writing and the other for reading (Figure 78). This makes possible the reading of information immediately following the writing of information.

The head assembly located between the vacuum column is built in two sections. The lower section is stationary and the upper section can be moved up and down. When the upper section is up, it allows the operator to thread tape. When down, it causes the read-write head to be in close contact with the tape for reading or writing.

The vacuum columns permit constant speed of the tape without waiting for the tape to accelerate or decelerate when starting or stopping.

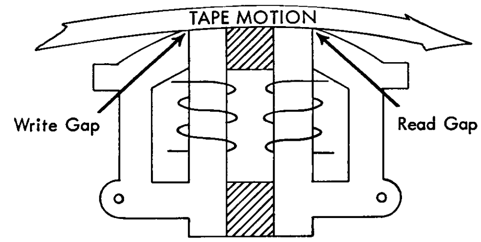


Figure 78. Two-Gap Head

Tape may be backspaced over a record or rewound to the beginning of the reel. When backspacing, tape movement is from the machine reel to the file reel. No reading or writing may take place while tape is moving in a backward direction.

Data are recorded on magnetic tape in the form of magnetized spots or bits located in seven longitudinal tracks (Figure 79). These tracks represent the character bit writing positions using the 705 code. A character is represented transversely across the width of the tape by seven bit positions. Figure 79 illustrates 705 coded characters on a section of tape. The characters are shown schematically, as actual recording is not visible to the eye. Magnetic tape may be used for repeated processing because information is automatically erased before new information is written.

Inter-record Gap

The end of a block of recorded information on tape, which may be composed of a record or a group of records, is indicated by an inter-record gap. The inter-record gap is an area of tape approximately ¾ inch long on which no information is written (Figure 80). Tape movement while reading is such that all information between inter-record gaps is read as a unit. The tape never stops between inter-record gaps. Inter-record gaps are automatically placed on the tape during the writing operation, when the end of a block of information in CPU is sensed.

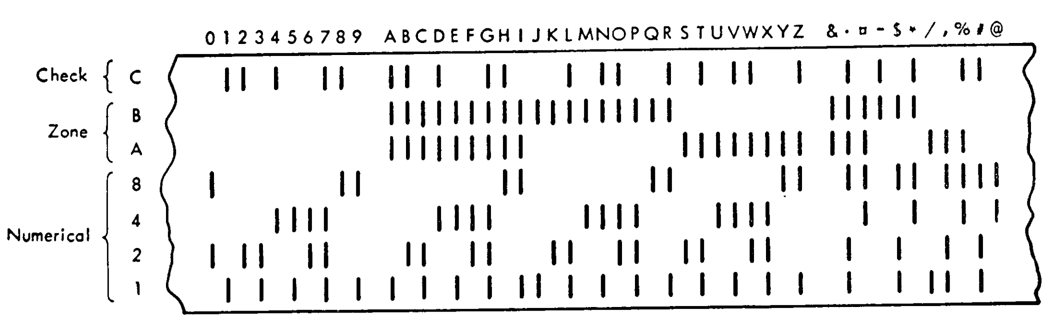


Figure 79. Tape Character Coding

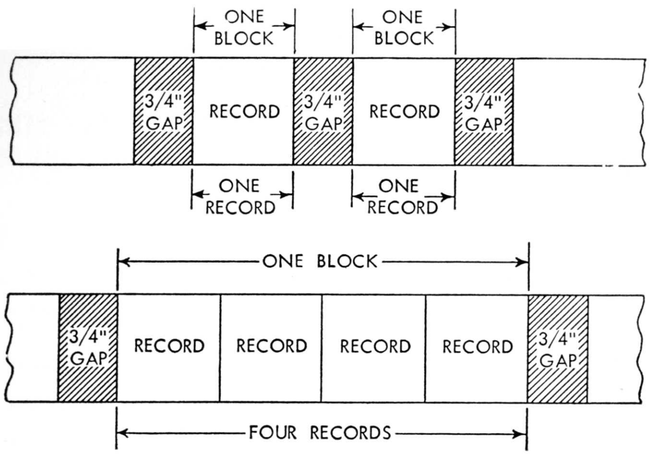


Figure 80. Single and Multiple Record Blocks

Tape Mark

The tape mark (special character) is used to indicate the end of a file of information on tape when writing. The tape mark is not automatically generated but can be written on tape with the control 00001 instruction. The tape mark is a unit record of one character and appears after the inter-record gap that follows the last record.

The tape mark serves either as an end-of-file or end-of-reel indication when the tape is in read status.

Photo-Sensing Markers

The beginning and ending of the useful portion of a reel of tape are indicated by photo-sensing markers which are manually placed on the tape. The photo-sensing markers are small pieces of plastic, one inch by $\frac{3}{16}$ inch coated with vaporized aluminum on one side and with adhesive on the other. The marker indicating the beginning of the useful portion of tape is known as the load point marker. The marker indicating the effective end of a tape reel is known as the end-of-reel marker.

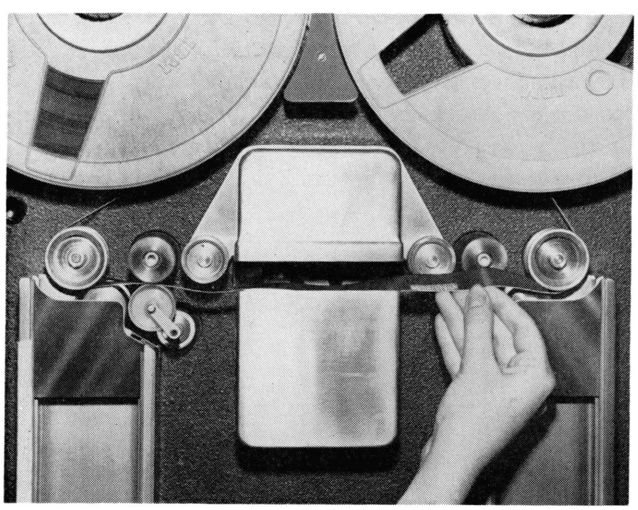
At least ten feet of tape must be allowed between the beginning of the reel and the load point marker as a leader for threading the tape over the feed rolls and the read-write head. More than ten feet may be allowed by placing the marker any desired length from the beginning of the reel. Information may not be stored in this area of the tape. To indicate the load point, the one inch dimension of the marker must be parallel to, but no more than, $\frac{1}{32}$ inch from the channel 1 edge of the tape—the edge nearest to the operator when the tape reel is mounted (Figure 81A).

Approximately 18 feet of tape are normally reserved between the end-of-reel marker and the physical end of the tape.

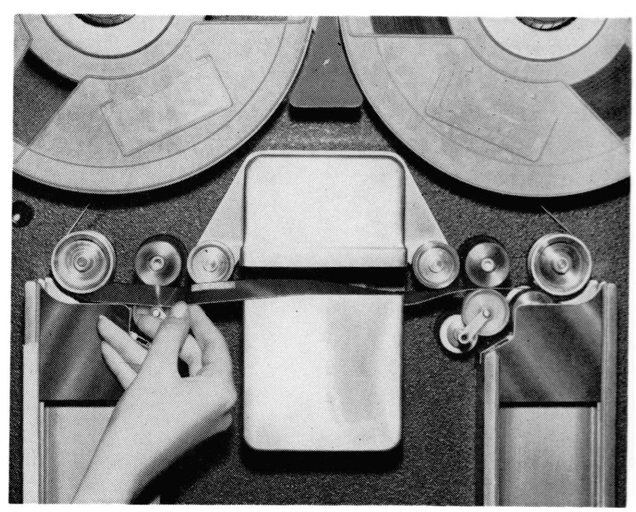
This space includes at least ten feet of leader and enough tape to hold a record of 20,000 characters after the end-of-reel marker is sensed. To indicate end of reel, the marker must be placed parallel to, but no more than, $\frac{1}{32}$ inch from the C track edge of the tape—the edge nearest the tape unit when the reel is mounted (Figure 81B). Place load point and end-of-reel markers on tape with care; they should be properly aligned and pressed tightly onto the tape with the back of the fingernail.

Tape Unit Indicator

Each unit is provided with an input-output indicator to indicate the end of a reel or file. The indi-

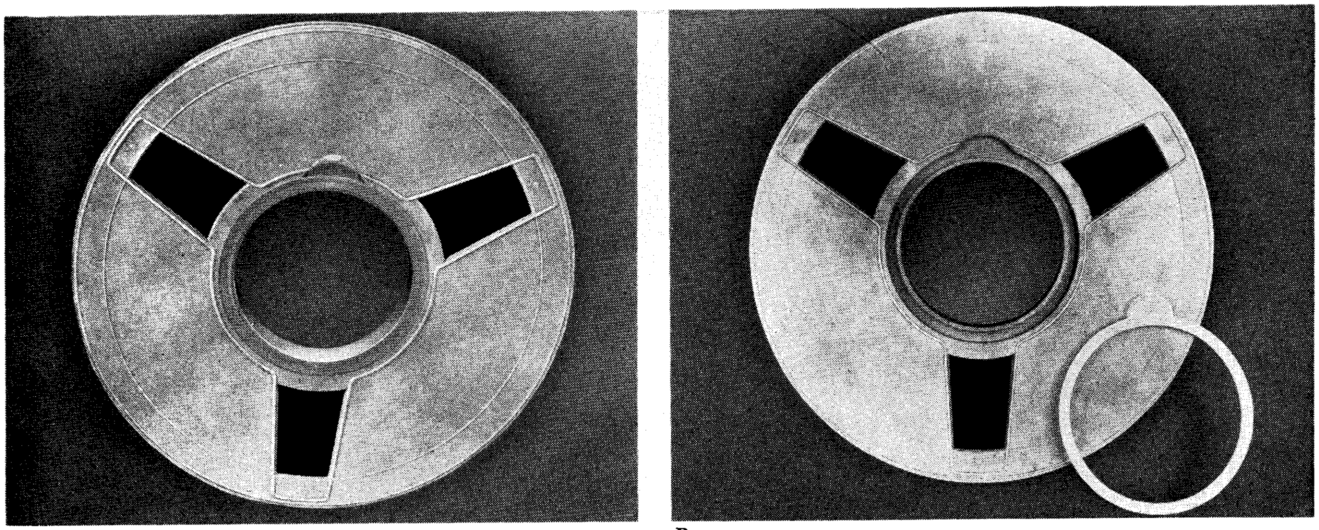


(a) Load Point



(b) Physical End

Figure 81. Load Point and End-of-Reel Markers



A B

Figure 82. File Protection Device

cator can be turned on by any of the following:

1. Sensing the end-of-reel marker while writing.
2. Sensing the tape mark as a unit record while reading.
3. Using an ION instruction in the program directed to a selected tape unit.

The indicator can be turned off by:

1. Depressing the unload key on the tape unit.
2. An IOF instruction in the program directed to a selected tape unit.

File Protection

The back of the tape reel (machine side) has a circular groove in which a plastic ring may or may not be inserted. To write on tape, the plastic ring (file protection ring) must be placed in the groove of the file tape reel (Figure 82A). A tape may be read with the file protection ring inserted or removed. The file protection ring should be removed from the tape reel after writing on tape is completed (Figure 82B). Doing this prevents accidental writing and resultant loss of valuable tape records. Do not remove the file protection ring while tape is loaded in the vacuum columns of the tape unit. This could result in a broken or damaged tape.

NOTE. To prevent possible lost system time from tape breakage near the load point region of acetate tape, it is wise to periodically remove a section of tape (about ten feet) from the leading edge of reels prior to rewriting. This procedure should be performed following about every 500 reel mounting operations. The frequency for performing this procedure should be modified, depending upon operation procedures, to insure that tape breakage at load point resulting from frequent handling does not occur.

Tape Validity Checking

Information written on tape or read from tape is checked to insure accuracy of data transmission and recording. The actual tape checking circuitry is located in the particular control to which the tape unit is attached.

As a record is being written on tape, an odd or even indication of the number of bits written in each of the seven bit tracks is made. At the end of every record, an extra bit is written in those tracks having an odd number of bits. The extra bits written produce a check character which follows the record. After the check character has been written, the number of bits in any bit track, including the check character bits, should be even (Figure 83). If not, a read-write check error is indicated.

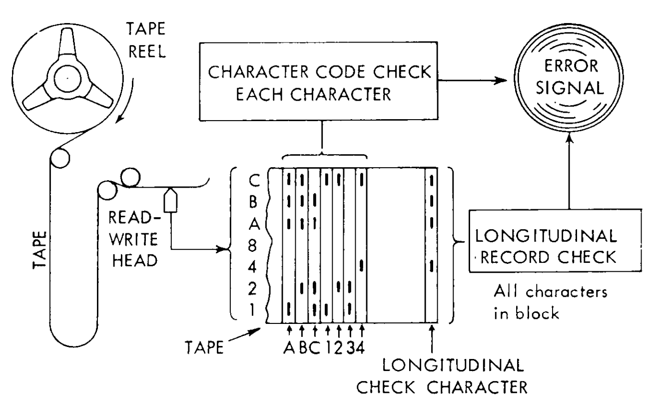


Figure 83. Tape Reading Checks

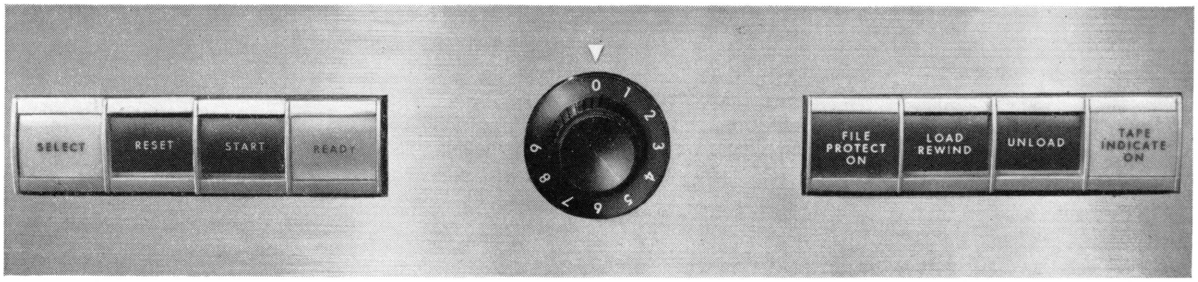


Figure 84. IBM 727/729 I Operating Keys and Lights

Tape is checked in two ways as it is read:

1. A character code check (vertical redundancy check VRC) is made to insure that an even number of bits exists for each character read. If an odd number of bits is detected for any character, a read-write check error is indicated.
2. A longitudinal record check (LRC) is made by developing an odd or even indication of the number of bits read in each of the seven bit tracks of the record including the bits of the check character. If any bit track of the record indicates an odd number of bits after it is read, a read-write check error is indicated.

   Dual level energy checking which is a feature with the DS only is discussed in the IBM 767 Data Synchronizer section of "Machine Components."

Operating Keys and Lights

The operator's panel of the magnetic tape units is provided with keys and lights to control the unit manually and to determine the status of the unit. The functional use of identical keys and lights on all three tape units is the same. However, the operating keys and lights of the 729 III (Figure 84) tape unit are different in appearance and location than those of the 727 and 729 I tape units (Figure 85).

Address Selection Switch. A rotary switch is provided to permit the operator to set a tape unit to any of the ten possible addresses. This switch should not be turned during any tape operation.

Select Light. This light is turned on when a tape unit is selected by an instruction and the address selection switch corresponds to the addressed tape unit.

Start Key. Use of this key places the tape unit in ready status and turns on the ready light if:

1. Tape has been loaded into the vacuum columns.
2. The reel door is closed.
3. The tape unit is not in the process of rewinding.

Ready Light. This light, when on, indicates that the tape unit is ready for operation or is actually in the process of being used.

File Protection Light. This light automatically turns on if the tape unit is loaded with a file reel which does not have the file protection ring inserted in the back of the reel. Tape cannot be written as long as the file protection light is on.

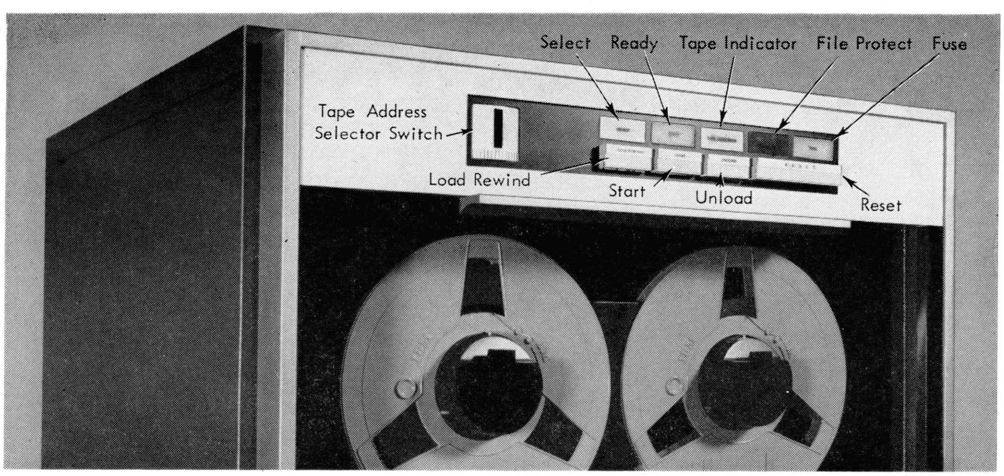


Figure 85. IBM 729 III Operating Keys and Lights

Tape Indicate On Light. This input-output indicator is turned on by the following:

1. Sensing the end-of-reel marker while writing on tape.
2. Sensing a tape mark while reading from tape.
3. By an ION instruction to a selected tape unit.

It is turned off by an IOF instruction to the selected tape unit or by the depression of the unload key.

Load Rewind Key. This key is operative only when the reel door is closed and the ready light is off. Three types of operations are performed by this key depending upon the status of the tape unit.

1. Use of the key after tape has been properly mounted in the magnetic tape unit lowers tape into the vacuum columns, lowers the head assembly, and moves tape in the rewind direction until the load point marker is sensed.

2. Use of the key with tape loaded and the machine reel containing more than one-half inch of wound tape, initiates a high-speed rewind operation. The tape is removed from the vacuum columns, the head assembly is raised and tape is rewound at high speed until approximately one-half inch of wound tape remains on the machine reel. Tape is then lowered into the vacuum columns, the head assembly is lowered and tape is rewound at low speed, until the load point marker is sensed.

3. Use of the key with tape loaded and the machine reel containing less than one-half inch of wound tape rewinds the tape, at low speed, until the load point marker is sensed.

CAUTION: Do not open the reel door during rewind or load point searching.

Unload Key. This key is operative only when the ready light is off, tape is in the vacuum columns, and the reel door is closed. Use of this key removes the tape from the vacuum columns and raises the head assembly, regardless of the distribution of tape on the two reels. If tape is not at load point when the operator wishes to change it, a load point search should be initiated first by depression of the load-rewind key. Depression of the unload key also turns off the tape-indicate-on light, if on.

Reset Key. Depressing this key resets the tape unit to manual control (except the input-output indicator). It changes a high-speed rewind to a low-speed rewind, stops a low-speed rewind, and in general stops any tape operation that has been initiated previously.

Reel Release Key. When depressed, the reel release key (located just below the file reel) releases the reel clutches so that reels may be turned manually. This is necessary when tape is to be loaded into or removed from the tape unit.

Reel Door Interlock. When the reel door is open, this interlock prevents any normal operation of the tape unit. On the 729 III, the reel door interlock is operated by either the opening of the reel door or the opening of the reel door window.

Fuse Light (IBM 729 III). When on, this light indicates that a fuse (protective device) has burned out. Notify a customer engineer of this condition.

Tape Load Procedure

Before the tape load procedure is initiated, the magnetic tape unit should be in an unload condition and tape removed from the machine.

1. Check the reel to be loaded to determine if it should have the file protection ring inserted or removed. Mount the reel to be loaded on the left mounting hub and tighten the hub knob (Figure 86). The hub contains a rubber rim that grips the reel tightly when the knob is tightened. When loading, push the reels firmly against the stop on the mounting hub to insure proper alignment. Always be careful that the hub knobs have been tightened. However, do not use excessive force when tightening hub knobs, for this tends to strip the threads.

2. Hold the reel release key depressed and rotate the file reel in a clockwise direction, unwinding about four feet of tape.

3. Place the tape over the left roller through the read-write head assembly and around the outside edge of the right roller (Figure 86). Place and hold the end of the tape between the index finger and the hub of the machine reel. Depressing the reel release key, wind the tape on the machine reel in a clockwise

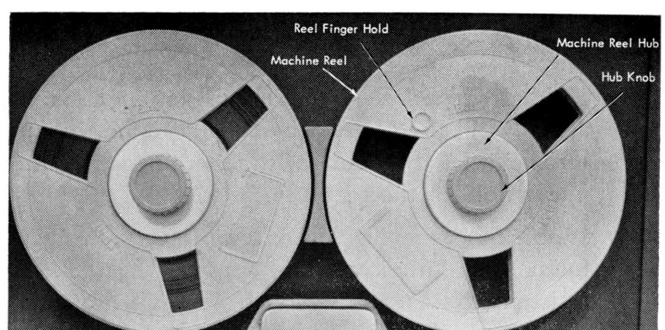


Figure 86. Tape Feeding

direction for at least two full turns beyond the load point marker. When placing tape on the machine reel, align it carefully to prevent damage to the edge on the first few turns. When winding the tape to load point, rotate the machine reel with the finger in the reel finger hold or near the hub and on the reel. Rotating the reel with the finger in the cutout can result in nicking or curling the edge of the tape.

4. Close the reel door. Make sure that the door interlock switch is closed.

5. Depress the load rewind key to:
   a. Load tape into the vacuum columns.
   b. Lower the head assembly.
   c. Rewind tape to the load point.

6. Depress the start key. This places the unit under automatic control and turns on the ready light.

NOTE: Do not turn power off with the tape unit in a load status because the head assembly must be up for removal of tape. If power is turned off after leaving load point, it will be necessary to begin a new start procedure to resume operation.

Tape Unload Procedure

To unload tape, use the following procedure:

1. Depress the reset key (tape unit) to turn off the ready light. Depressing the reset key is necessary only if the ready light is on.
2. Depress the load rewind key to rewind the tape.
3. When the load point has been reached, depress the unload key.
4. Open the reel door when the head cover is fully raised; the tape is out of the columns. Do not open the door of the tape unit until the tape drive mechanism has completed the unloading sequence.
5. Hold depressed the reel release key and manually rewind the file reel by turning it in a counterclockwise direction with the finger in the finger hold of the reel.
6. When the tape has been completely rewound, loosen the hub knob and remove the reel. If resistance is encountered in removing a reel, exert pressure from the rear of the reel with the hands as near the hub as possible. Never rock a reel by grasping it near the outer periphery in such a way as to pinch the edges of the outer turns of tape.
7. Check the removed reel to determine whether it is to be file protected and whether it has been labeled correctly. Place the reel in the container.

Magnetic Tape Handling

Foreign particles on tape can reduce the intensity of reading and recording pulses by increasing the distance between the tape and the read-write head. In all operating procedures, be extremely careful to protect magnetic tape from dust and dirt. Keep the tape in a dust proof container whenever it is not in use on a tape unit. When a reel of tape is removed from a tape unit, immediately place it in a container. Always place sponge rubber grommets or special clips on the reel as they are stored, to prevent the free end from unwinding in the container.

Information is recorded witin .020 inch of the edge of the tape. Proper operation requires that the edge of the tape be free from nicks and kinks. Handle reels near the hub whenever possible. Gripping the reel so as to compress its outer edges pinches the few turns of the tape near the outer edge. Dropping a reel of tape can easily damage both the reel and the tape. Never throw or mishandle reels even while they are protected in their containers.

To prevent possible lost systems time from tape breakage near the load point region of *acetate* tape, it is recommended that periodically about ten feet of tape be removed from the leading edge of reels, prior to rewriting. This procedure should be performed following approximately every 500 reel mounting operations. Modify the frequency of this procedure depending upon operation techniques to insure that tape breakage at load point resulting from frequent handling does not occur.

IBM 754 Tape Control

The IBM 754 Tape Control connects as many as ten IBM 727 Magnetic Tape Units to the IBM 705 Central Processing Unit. All tape units are connected to the control by two signal cables referred to as odd and even. Five tape units can be connected to each cable. Tape units attached to even cable must have even numbered addresses (address selection switch). Tape units attached to the odd cable must have odd numbered addresses (address selection switch).

The tape control interrogates the I-O (input-output) cable at all times and determines when it is selected to control the operation of one of the attached tape units. The selected tape control coordinates the flow of information between the CPU and the attached tape units. In both the read and write operations, validity checking of tape records occurs in the tape control.

When using one 754 Tape Control and two of the attached tape units for a read-while-write operation, one of the tape units must have an odd numbered address and the other an even numbered address. If separate control units are used, one controlling the input tape and one controlling the output tape, there is no restriction on the address of the input or output tape units.

Operating Keys and Lights

The keys and lights on the operator's panel of the IBM 754 Tape Control are show in Figure 87.

CPU-Tape Switch. The two positions are CPU and tape. In all operations with CPU, this switch must be set to CPU (the up position).

Power-On Key. When the power-on key is depressed, the various power supplies in the control unit are made operative in a sequential process that takes approximately four minutes to complete.

Power-On Light. The power-on light is on whenever the power-on key has been depressed and stays on as long as power is on.

DC-On Light. The DC (direct current) on light automatically lights after the power-on key is depressed and the power-on sequence is completed. The light remains on until a power-off condition is present.

Power-Off Key. Depression of the power-off key removes all electrical power (with the exception of 110 volts AC) from the control unit and the attached tape units.

Fuse Light. This light, when on, indicates that a fuse (protective device) has burned out. This condition forces a power-off condition. The customer engineer should be notified if this light is on.

Thermal Light. This light, when on, indicates that the operating temperature of the machine is too high. This condition forces a power-off condition. The customer engineer should be notified if this light is on.

Reset Key. This key resets the LRCR register and all registers necessary to have a normal select operation.

LRCR Neons. These neons reflect the contents of the longitudinal redundancy check register. If an LRC or VRC tape error is detected, the neon of the tape bit track in which the error occurred will be on.

IBM 767 Data Synchronizer

The IBM 767 Data Synchronizer (DS) is a component designed to control and synchronize the flow of data between the 705 III and the 729 I and the 729 III Magnetic Tape Units. Use of the DS makes possible the overlapping of processing with reading or writing operations. This is done by controlling the insertion and removal of data into or from memory at intervals throughout the processing cycles of the CPU.

The DS is provided with two tape buses to accommodate a total of ten tape units. As many as five 729 I tape units can be attached to one bus and as many as five 729 III tape units can be attached to the other bus.

A 705 III system can accommodate six DS's while using only 729 I tape units. If one or more 729 III tape units are to be used, a maximum of four DS's can be accommodated.

All existing card input-output, printers and tape controls may be used in a 705 III system along with the DS (Figure 88). The IBM 748 Data Synchronizer Power Supply is used with the 767.

Reading a Record into Memory

As shown in Figure 89, the DS operates in the following manner to read a record into memory.

A particular tape unit is selected by the program and instructed to read. Then, without waiting, the CPU proceeds to the next instruction and continues to process the previous record already in memory.

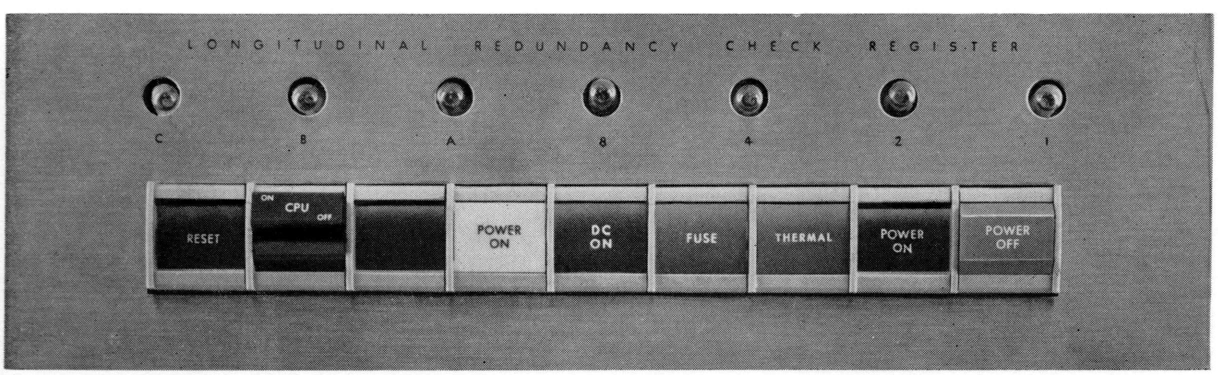


Figure 87. IBM 754 Operating Keys and Lights

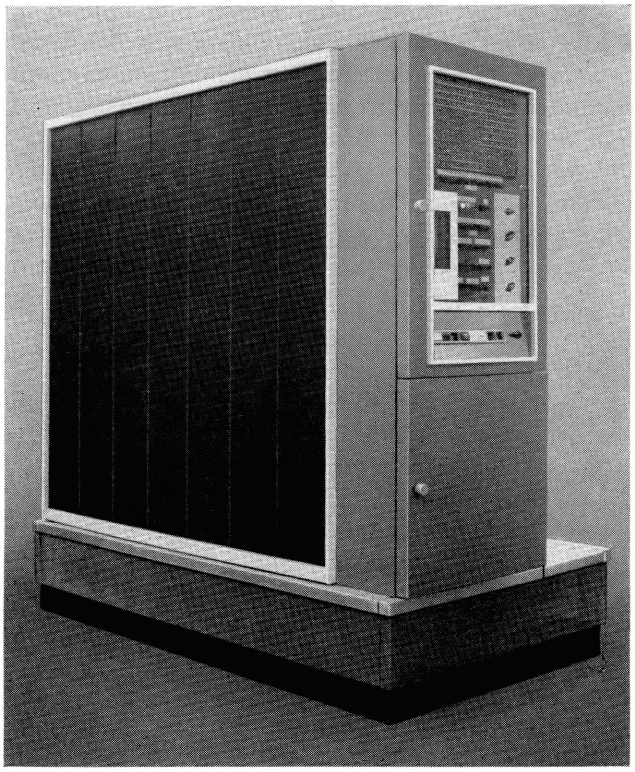


Figure 88. IBM 767 Data Synchronizer

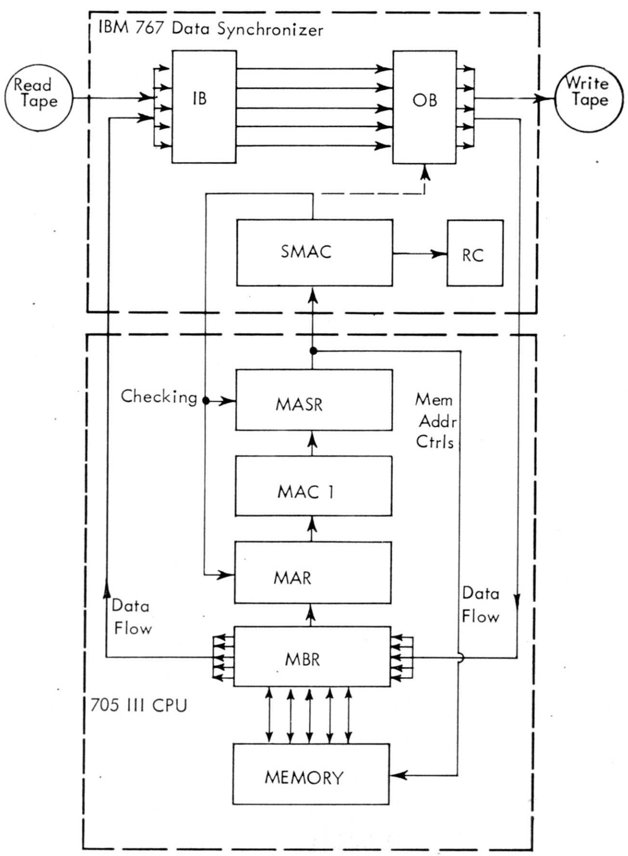


Figure 89. Schematic of Data Flow between CPU and the DS

While the CPU is processing other data, characters of the incoming record are read from the tape, one at a time, and placed in the *input buffer* of the DS. The capacity of the input buffer is five characters. As soon as it is filled, its contents are deposited into the *output buffer*. It does this in time to receive the first of the next five characters from the tape. The tape unit continues at full speed until the entire record is read.

As soon as the output buffer has received its five characters from the input buffer, it takes the next available machine cycle of the CPU to deposit its contents into the appropriate area in memory. This step interrupts the CPU for a total of only nine microseconds. Having been interrupted briefly, the CPU returns to the processing of other data until interrupted again by the DS with five more characters.

While the record is reading in this fashion, the *synchronizer memory address counter* (SMAC) controls the location in memory where data will be placed. It does this by starting at the address specified by the read instruction and increasing by five as each group is read in. It thus takes over the function of MAC I in the CPU so that the MAC I may be used for other processing, independently of a read or write operation.

In a read operation, from one to four group marks may be placed in memory automatically at the end of the record. This eliminates the programmed placing of these marks. The number of group marks placed depends upon where in memory the last character of the incoming record is placed. That is, if the record is not an even multiple of five characters, enough group marks are added so that the number of positions occupied by the record in memory is a multiple of five (ending in a 4 or 9 position) and with a group mark in the last position. If the record is an even multiple of five, group marks are not stored. For example, with the instruction RD 10000, the tape record 14833AbBbDUNN$1.25 would be written into memory as in line 1 of Figure 90. The tape record 13621JbRbADDAMS$1.50 would be written into memory as in line 2 of Figure 90.

Once the entire record has been read into memory, the ready indicator is set, indicating that the record is

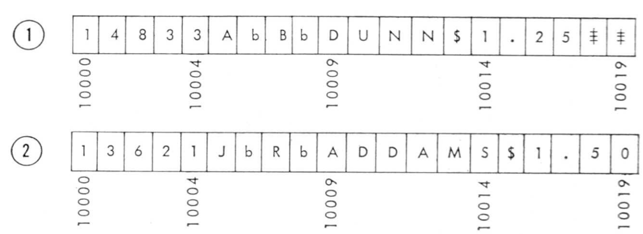


Figure 90. Placement of Group Marks by a DS

now in memory and the DS is available for further use. This indicator may be interrogated at any time thereafter by a TRS 01 instruction.

The parallel character transmission (PCT) data check (IBM 782 Console Control) indicator is turned on if there is a read error. The indicator may be interrogated by a TRS 02 or TRS 03 instruction. The PCT data check indicator replaces the 00902 indicator for tapes attached to the DS.

A write operation would work the same way, in reverse. Information would be removed from memory, five characters at a time, and placed in the DS input buffer. From there, it would be transferred to the output buffer which would then write the characters serially onto the tape. In writing, as in reading, the processing of other data may continue in the CPU, being interrupted periodically for nine microseconds as a group of five characters is transmitted from memory to the DS input buffer.

Operating Keys and Lights

The operating keys and indicator lights on the operator's panel of the 767 are shown in Figure 91.

Power-On Key. When the power-on key is depressed, the various power supplies in the control unit are made operative in a sequential process that takes approximately five minutes to complete.

Power-On Light. The power-on light comes on whenever the power-on key has been depressed and stays on as long as power is on.

DC-On Light. The DC (direct current) on light automatically lights after the power-on key is depressed and the power-on sequence is completed. The light remains on until the power-off key is depressed.

Power-Off Key. Depression of the power-off key removes all electrical power from the 767 and attached tape units, except for the blower which runs for approximately five minutes after power is turned off.

Fuse Light. This light when on indicates that a DC circuit protector has tripped and results in a power-off condition. The customer engineer should be notified of this condition.

Thermal Light. This light, when on, indicates that the operating temperature of the machine is too high. This condition forces a power-off condition. The customer engineer should be notified if this light is on.

Reset Key. This key resets the machine and prepares it for a normal operation.

Select Light. This light is on whenever the DS has been selected for use by CPU.

Ready Light. This light is on when the DS is selected and ready for CPU operation.

Tens Address Switch. This switch allows the tens position of the DS tape address to be changed. For example, if the CPU tape address is xx240, the tens address switch must be set on the 40 position before the addressed tape can be selected.

Addressing for Read and Write

As with all 705 models, the address portion of a write or read instruction indicates the position in memory where the first character of the record is to be withdrawn or placed. However, with use of a DS, the address of this first position must be an address with a units digit of 0 or 5. For example, instructions such as RD 10030 and WR 30085 have proper addresses while instructions such as RD 10031 and WR 30088 have improper addresses. These improper addresses will *not* turn on the instruction check indicator, but will turn on the PCT data check indicator. If a read operation is being performed, the PCT address-check indicator is also turned on, and CPU stops.

The number of DS's to be used for a particular application depends upon the application itself. A single DS enables the 705 III to process or calculate at the same time that it is either reading or writing. Two

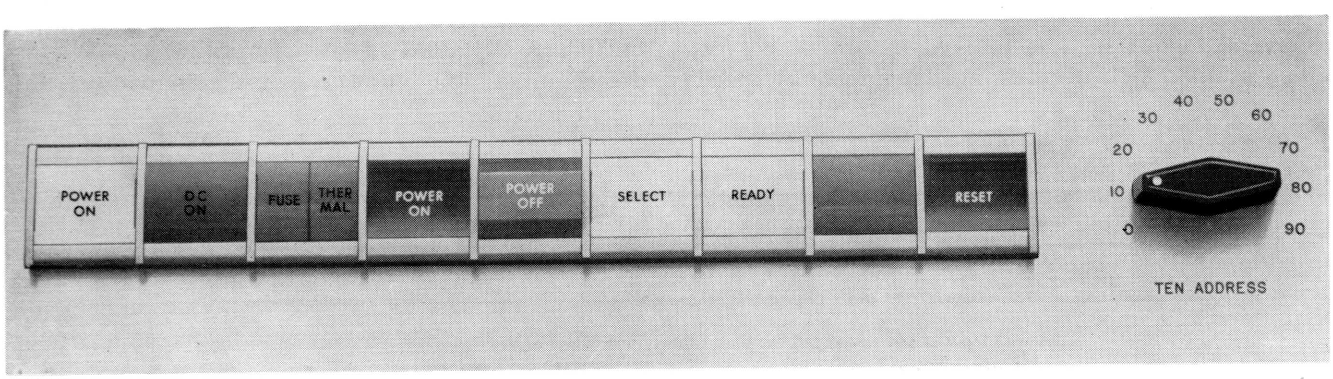


Figure 91. Data Synchronizer Operating Keys and Lights

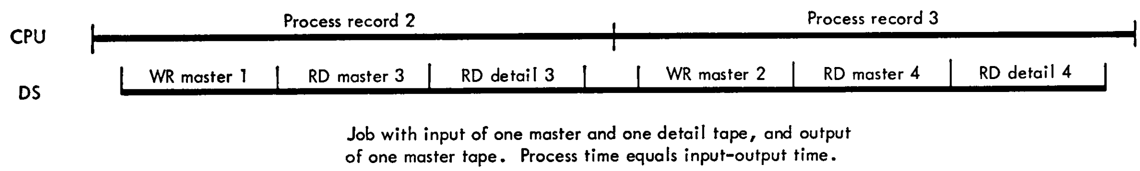


Figure 92. Efficient Use of a Single Data Synchronizer

DS's permit simultaneously reading and/or writing, and calculation (i.e., reading or writing on two tapes simultaneously or writing on one and reading from another tape simultaneously) as desired. Each additional DS, up to a maximum of six (if only 729 I tape units are used) or four (if 729 III tape units are used) permits that many more tape units to be operated simultaneously. Optimum use of the 705 CPU and the DS is realized when tape time is completely overlapped by processing time. Multiprogramming may be used to help in achieving this goal.

The diagram in Figure 92 shows how a single DS can be used to advantage where processing time is equal to, or greater than, input-output time. Figure 93 shows two DS's being used where processing time is about equal to one half the total input-output time.

Data Handling Methods

Each DS represents one channel to the CPU; tape files on the same DS compete with each other for use of the channel. Tapes should be so assigned as to engage the available channels simultaneously and continually as much as possible. Files that tend to be used together should be on different data synchronizers. When only one of several files is used at a time, those files may well be placed on the same DS. The number of high-speed and low-speed tape units available on each DS naturally affects the assignment of tape files.

When tape assignments have been established, a particular method of operation for the various files should be decided. Use of any method is determined by several factors such as blocked or unblocked records, variable length records, available memory space, number of DS's and the amount of time to be saved by overlapping. Three logical approaches to the problem are illustrated in Figure 94. These methods are provided only as an aid in understanding the problem.

Method 1 (Figure 94A). During housekeeping, the DS input area is initialized by reading the first tape record from the input file. The first set of instructions in the main line program moves the record in the input area to the compute area. As soon as the move has been completed, instructions are given to read the next record into the input area. Since input and computing are taking place in separate areas, they can proceed simultaneously without interfering with each other.

Upon completion of computation, it is necessary to clear the computation area to make way for the next input tape record. This is done by moving the contents of the computation area to the DS output area and giving an instruction to the output tape to commence writing from the DS output area. The cycle is now ready to repeat.

Method 2 (Figure 94B). Memory is set up to contain four areas, two for input and two for output. On the input side, the DS alternately uses first one and then the other of the twin input areas as the DS input area. The input area which is not being used by the DS is available to the program as the computation input area. In the same way, the twin output areas alternate with each other. First, one serves as computation output area and the other as DS output area, then vice versa.

To follow the sequence of events in a simple case, use Figure 94B, and assume that no other work areas are required. In housekeeping, the input area A is

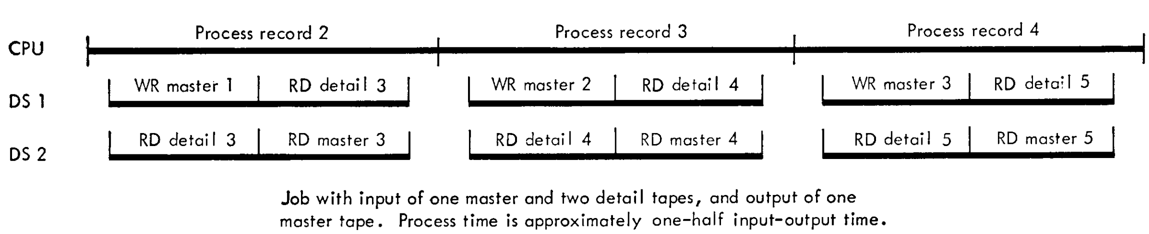


Figure 93. Efficient Use of Two Data Synchronizers

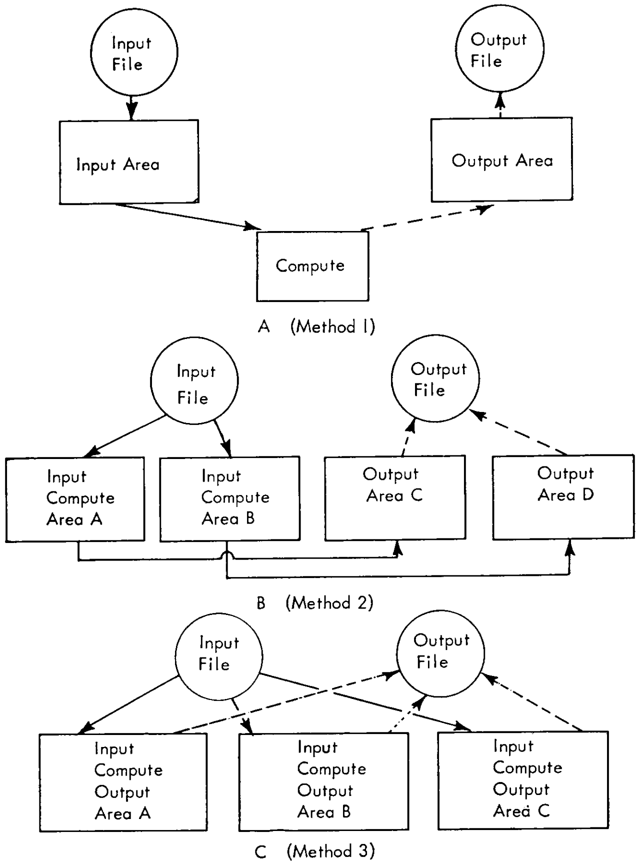


Figure 94. Data Handling Methods

initialized by reading into it the first tape record from the input file.

1. An instruction is given to start a DS reading into area B.
2. Computation goes ahead in input area A.
3. The processed contents of input area A are moved to output area C.
4. An instruction is given to start a DS writing output area C.

The cycle is now ready to repeat, interchanging A with B and C with D.

1. An instruction is given to start a DS reading into input area A.
2. Computation goes ahead in input area B.
3. The processed contents of input area B are moved to output area D.
4. An instruction is given to start a DS writing from output area D.

The alternation of reading, writing, and computing between the two input areas and between the two output areas is accomplished by address modification (or, in some cases, by using two routines).

Method 3 (Figure 94C). This method illustrates the use of three separate areas of memory. Each area acts as an input, compute, and output area for a record. The various functions of each area can be overlapped with other areas depending upon the number of DS's to be used. With one DS, only two memory areas would be required, to permit the overlapping of read-compute and write-compute operations. Using two DS's and three working areas makes possible the overlapping of reading and writing with computing.

Programming and Checking

The following is a simplified illustration of how to read and check programming with the DS.

00104	SEL	00200	Select DS 0 and tape unit 0.
00109	RD	05000	Start reading record 10 into input area of memory.
00114	- - -		Process record 9.
01004	- - -		Process record 9.
02004	SEL	00200	This instruction is needed only if another select instruction has intervened between steps 00104 and 02004.
02009	TRS 01	02019	Test ready indicator of the DS to see if record 10 has been completely read and placed in memory.
02014	TR	02009	Loop around until record 10 has been read in; or, do a few more steps of the processing.
02019	TRS 02	05004	Check for error on read operation and transfer to error routine if an error exists.
02024	TRS 00	05504	Because the tape unit has already been selected, this instruction transfers on end of file.
02029	- - -		If no error exists, record 10 may now be processed and the DS may then be used either to write out record 9 or to read in record 11.

This example represents the most logical sequence for the program, that is, instructing the machine to read, then later testing to see whether the entire record has been read in and, if it has, to check the record and process it.

When programming for the data synchronizer, the flow of data into and out of memory must be coordinated with the rate of processing the data. That is, before giving a read instruction, the programmer must be sure that the area in memory into which the data will be read is free to accept data, or will be free within 7.3 milliseconds for the 729 III, or within 10.8 milliseconds for the 729 I (average tape starting time), after the read instruction is given.

For example:

SEL	00201	Select DS 0 and tape unit 1.
RD	01000	Start reading.
RCV	02004	Move previous data from read-in area to work area.
TMT	01004	

Although the read operation is started before the previous data are moved out of the read-in area, there is ample time for the RCV and TMT instructions to be executed before the new record starts reading in, if the record to be transmitted is not too large.

To illustrate further an advantage of the DS, as well as the importance of timing, consider a file maintenance application where the control fields are at the beginning of long master records. The master record must be read in and compared with a detail transaction record to see if the master is active. In many cases it is not active and, therefore, is written out without change. This master record could be read in through DS 1. As soon as the first portion of the record is in memory, the control field could be compared against the detail to determine whether the record is inactive. If so, the instruction is given to write the record out by way of DS 2. All of this might take place before the record is entirely read in through DS 1.

To check this operation, the PCT data check indicator can be tested at any time during the read-in. It reflects the character validity of the operation prior to that time. A check can also be programmed to verify that the first part of the record (control field) has read in before the comparison is made.

When an EOF is encountered by any tape unit attached to a DS, only the I-O indicator (tested by a TRS 00 or TRS 03) is turned on. If the TRS 03 instruction is used, the computer is held up until the DS is ready. If the I-O indicator of a DS tape unit should be interrogated by a write-group-mark operation (which should be used only on TRC and 760 operations), it would be treated as a NOP (no operation).

While writing with a DS, the 00901, 00902, and the ANY indicators are not affected by a character code error; only the PCT data check indicator is turned on. It may be tested by a TRS 02 or TRS 03 instruction.

When a select instruction is executed, the CPU select register is set to MAR; processing continues in the CPU regardless of the status of the DS. With initiation of a read, write, or control instruction, CPU processing is stopped until the DS becomes free or available to execute the given instruction. Then the selection and specified action are initiated thus releasing the CPU for further processing. Once a tape unit is up to speed, the DS makes a request for a CPU interruption and transmission of the record data starts.

Functional Description

The DS consists mainly of the following features: input buffer (IB), output buffer (OB), synchronizer memory address counter (SMAC), record counter (RC), select register, ready indicator and PCT check indicators (Figuge 89).

Input Buffer (IB)

The IB is a five-character register that receives data either serially from a tape unit or five characters in parallel from memory, depending on the mode of operation.

In the read mode, the IB accepts data, one character at a time (serially), from the selected tape unit. When all five character positions have been filled, the contents of the IB are transferred, as a block of five characters (parallel) to the output buffer.

In the write mode, the IB accepts a block of five characters from memory, stores it until the OB has sequentially written out the preceding record block, and then transfers the block of five characters to the OB. In this manner, the IB is cleared to accept the next block of five characters from memory.

Output Buffer (OB)

The OB is a five-character buffer register. The OB receives data as a block of five characters from the IB on one side and transmits data either serially to a tape unit or in parallel to memory, depending on the mode of operation.

In the read mode, the OB accepts the block of five characters from the IB and stores it until there is an available memory cycle. The data are transmitted in parallel to memory. In this manner, the IB is ready to begin receiving the next five characters serially from tape.

In the write mode, the OB accepts in parallel the five characters to be written out from the IB and transmits them serially to the selected tape unit.

Synchronizer Memory Address Counter (SMAC)

SMAC stores the address of the initial memory location into which data are to be placed during a *read* operation or from which data are to be taken during a *write* operation.

During a *read* operation, SMAC sets to MAR by way of MASR to contain the initial address of the instruction. When data transfer begins, MASR sets to SMAC and OB unloads its first block of five characters starting at this address. SMAC is stepped plus five following each DS memory cycle and continues to control the *read* operation.

During a *write* operation, SMAC sets to MAR by way of MASR to contain the initial address of the instruction. When data transfer begins, MASR sets to SMAC and the first block of five characters loaded into the IB comes from this address. SMAC is stepped plus five following each DS memory cycle and continues to control the write operation.

At the end of a *read* or *write* operation, the contents of the synchronizer memory address counter (SMAC) contain a memory address five positions higher than the last memory reference. "Last memory reference" means the 0 or 5 address of the last group of five memory characters handled by the instruction. For a WR instruction, the last group handled is the group after the group containing the group mark. The contents of SMAC are placed in the four low-order positions of the OB. A zero is placed in the high-order position of the OB, unless there is an LRCR error. In this case, the bits in error in the LRCR are replaced in the high-order position.

Record Counter (RC)

The RC is a three-digit decimal counter used to accomplish multiple record operations. Multiple record operation permits a series of operations to be performed with one instruction. The WR 02 instruction sets the RC and RC indicator, putting the following instructions under RC control:

RD 00	WTM
RD 01	SKIP
WR 00	BSP 00
WR 01	

The WR 02 instruction sets the RC to the tens, hundreds, and thousands order positions of SMAC, which has been set to MASR and contains the address of the WR 02 instruction. The RC operation continues until the contents of the RC have been reduced to 000.

The following routine reads 125 records into memory, starting at location 05000.

OPERATION	ADDRESS	ADDRESS COMMENTS
SEL	00200	Select DS 0 and its attached tape unit 0.
WR (02)	01250	Put 125 into the RC and turn on the RC indicator.
RD (00)	05000	Start reading from the tape into memory location 05000.

Unit Select Register

The DS *unit select register* is used to store the number of the selected tape unit. This allows a tape unit to remain in select status during a DS operation, independently of the CPU select register.

Ready Indicator

The ready indicator is used to indicate the availability of the DS. The ready indicator is on when the DS is selected and is not busy (not executing a RD, WR, BSP, WTM or SKP instruction) and the selected tape unit is in ready status. If the DS is executing a RWD instruction, it can become ready for all tape units except the one that is rewinding.

The status of the ready indicator is interrogated by the TRS 01 instruction.

PCT Data Check Indicator. The neon for this indicator is located on the 782 Console Control. The PCT data check indicator designates several types of error conditions, depending on the prevailing operating mode of the DS.

When the selected DS is in the read mode, the PCT check indicator is turned on by any of the following:

1. Incorrect instruction address (0 or 5 check).
2. *Memory buffer register* code check.
3. Vertical redundancy check (VRC).
4. Longitudinal redundancy check (LRC).
5. Incorrect SMAC-MAR or SMAC-MASR exchange.

When the selected DS is in the write mode, the PCT check indicator is turned on by any of the following:

1. Incorrect instruction address (0 or 5 check).
2. Memory buffer register code check.
3. VRC, LRC, or match check.
4. Writing a redundant tape mark.
5. Incorrect SMAC-MAR or SMAC-MASR exchange.

The status of the PCT data check indicator is interrogated by the TRS 02 and TRS 03 instructions. The indicator is reset off, if on, at the start of each RD, WR, or WTM operation.

PCT Address Check Indicator. The neon for this indicator is located on the 782 Console Control. An address check (0 or 5 check) or incorrect SMAC-MAR or SMAC-MASR exchange in any DS in active read status results in a CPU manual stop and the turning on of this indicator. (The PCT data check indicator is also turned on in this event.) The address check indicator is turned off by depression of the start or load keys (IBM 782 Console Control).

Checking of Data Flow

Tape Checks

The use of the two-gap read-write head of the 729 I and 729 III tape units offers increased checking while writing. All characters read from tape are checked for strength of bit signals at two energy levels. The bits of a character read at the read head are sent to the DS and set in both a *high sensitivity* register and a *low sensitivity* register. A stronger signal is required to set the low sensitivity register than the high one. The high and low registers are used as follows:

READ CHECKING. The output of the *low register* is checked for vertical redundancy (VRC). If the VRC check fails, bit positions containing ones in the *high register* are placed in corresponding positions of the low register. The low register is again checked for vertical redundancy and placed in the IB. If the second VRC fails, the PCT data check indicator is turned on. A longitudinal redundancy check (LRC) is made at the conclusion of the read operation. An error here also turns on the PCT data check indicator.

WRITE CHECKING. During writing, the bits of the characters written are read at the read head, sent to the DS and placed in the high and low sensitivity registers. The output of the low register is checked for vertical redundancy (VRC). In addition, a bit-for-bit match check is made against the contents of the high register. If a redundancy is detected or the two registers do not contain the same information, the PCT data check indicator is turned on. An LRC check is also performed and the PCT data check indicator is turned on if an error is indicated.

The programmer can interrogate the PCT data check indicator and make additional attempts to rewrite a record. If the error persists, it is possible to backspace and then skip this assumed portion of defective tape with a control 00009 (SKIP) instruction. Thus, five or six inches of tape are skipped over and erased before the record can be rewritten. In this manner, the writing of records on defective portions of tape is prevented without discarding the entire tape. When the same portion of tape is read, the section is automatically skipped. As a result, nothing is read from the defective portion of the tape. This depends upon the nature of the defect. For example, a pin hole will be detected and cause an error.

MEMORY BUFFER REGISTER (MBR) CODE CHECK

Any time a block of five characters is placed in MBR, either from memory or from the DS, a vertical redundancy check is performed on the contents of the MBR. If an error is detected, the PCT data check indicator is turned on.

Operation of the DS

SELECT

A select instruction addressed to a DS sets only the CPU select register. The DS select register does not set to the units digit of this register until a subsequent instruction is issued to the DS. If the DS is not in ready status at this time, its unit select register will not be set until a ready condition has been established.

A tens-position switch on the DS permits manual setting of the DS select address (tens order digit).

READ OPERATIONS

RD 00. When a RD 00 is to be executed in conjunction with a DS, the outline of events is as follows:

IF THE DS IS NOT AVAILABLE OR NOT READY, the execution of the instruction is held up. No disconnect signal is returned to the CPU.

IF THE DS IS READY or when it becomes ready, the PCT data check indicator is reset. SMAC is set to MAR by way of MASR and a SMAC-MAR check made to insure that the address of the instruction ends in 0 or 5 and that the SMAC-MAR exchange is correct. The unit select register is set and the selected TU put into read status and started. The ready indicator is turned off and a disconnect signal sent to CPU to signify execution of the above.

WHEN THE TAPE REACHES FULL SPEED, characters are read serially into the IB. Characters enter the register, one at a time, and are switched to their successive locations until all five positions have been filled.

WHEN THE IB HAS BEEN FILLED, the five characters are transferred, in parallel, to the OB. The DS then signals CPU that it is prepared to place this data into memory; it requests a machine cycle to perform the write-in. IB is free to receive the next five characters from tape.

AS A RESULT OF THIS REQUEST, CPU interrupts its current processing task. MASR is set to SMAC and a SMAC-MASR check made to insure a correct exchange of address. The contents of OB are placed into memory and SMAC is stepped plus five. CPU resumes its interrupted processing task. The read sequence is now established.

WHEN THE END-OF-RECORD IS SENSED, a LRC check is taken and the tape stops in the inter-record gap. The buffer portion of the DS continues to operate until the characters in the IB have been passed through OB to memory.

IF THE INSTRUCTION IS UNDER RC CONTROL:

1. If RC equals zero initially, the PCT data check indicator and RC control indicator are reset and the DS disconnects. The tape unit will not be started.
2. If RC is not zero initially, the PCT data check indicator is reset, a SMAC-MAR check performed, and the tape unit started. When the tape reaches full speed, RC is stepped minus one and reading begins. (RC is stepped at this point so that it equals zero when the last record has been read.)
3. When the end of record is sensed, a LRC check is taken. If the RC equals zero or a tape mark has been sensed, the tape stops in the inter-record gap and the RC control indicator is turned

off. (The buffer portion of the DS continues to operate until the characters in the IB have passed through OB to memory.) If RC is not at zero and a tape mark has not been sensed, the tape continues to move. Rc is stepped minus one and the normal sequence of operations is resumed.

IF THE IB IS NOT COMPLETELY FILLED AT THE END OF A TAPE RECORD, the DS causes group marks to be inserted in the empty character positions.

AT THE TERMINATION OF THE READ OPERATION, the contents of the four lower positions of SMAC are placed in the lower four positions of the OB. The 1, 2 and 4 bits in the high-order position of SMAC appear as A and B bits in the thousands-order and the B bit in the units-order of OB, respectively. An 8-and-2 bit combination (zero) is placed in the high-order position of OB unless there is an LRCR error, in which case the contents of the LRCR are placed in the high-order position of the OB.

AN ERROR ON A SMAC-MAR OR SMAC-MASR CHECK turns on the PCT data check indicator, stops CPU at the end of execution time, and turns on the console PCT data check and address check indicators. The instruction is converted to a RD 01. (With a SMAC-MAR error, no data are placed into memory; with a SMAC-MASR error, some data are placed into memory since the error indication does not occur until after OB has been unloaded.) Tape continues to move to the inter-record gap unless the instruction is under RC control, in which case tape moves until the RC count has been completed.

AN ERROR ON A MBR CODE CHECK OR TAPE CHECK (VRC or LRC) turns on the PCT data check indicator. The CPU does not stop, and tape continues to be read.

DS OVERFLOW (WRAP-AROUND). When a record that is long enough to exceed the upper end of memory is read through the DS into memory, the following events occur:

1. No characters are placed in memory after the uppermost position is filled (39,999 or 79,999).
2. The tape unit continues to read until the end of record indication, but the information is not placed in memory. SMAC continues to advance.
3. The PCT data check indicator is turned on as the end of memory is exceeded. It is necessary, through a PCT routine (subtracting the read address from the SMAC address) to determine that an overflow situation exists rather than other possible types of PCT errors.
4. If reading is under RC control and the end of memory is exceeded within the RC group, the operation converts to a RD 01, turns on the PCT data check indicator, and proceeds until RC is equal to 0.

RD 01. The RD 01 instruction causes the selected tape to be moved, read, and checked as in a RD 00 operation. Data are *not* placed in memory, however. A RD 01 operation is equivalent to forward spacing over a record.

If the instruction is under control of the RC, multiple records may be passed over. The operation is as described above under RD 00.

SMAC steps in the normal fashion. Checking is performed as described under RD 00.

RD 02—Read Memory Address (RMA). RD 02 applies only to the DS. If this instruction is given to a serial I-O device, a RD 01 results.

If the DS is not available or not ready when the RD 02 instruction occurs, the execution of the instruction is held up. No disconnect signal is returned to CPU.

If the DS is ready or when it becomes ready, the PCT data check indicator is reset, the RC is reset to zero, and the RC control indicactor is reset off. The contents of OB are placed into memory at the location specified by the address of the instruction. The address placed into memory from the OB is the setting of SMAC at the termination of the previous operation. When this has been done, a disconnect signal is returned to CPU.

This instruction may *not* be executed under RC control. Checking is performed during its execution as described under RD 00.

WRITE OPERATIONS

WR 00. When a WR 00 is to be executed in conjunction with a DS, the outline of events is as follows:

IF THE DS IS NOT AVAILABLE or not ready, the execution of the instruction is held up. No disconnect signal is returned to the CPU.

IF THE DS IS READY or when it becomes ready, the PCT data check indicator is reset. SMAC is set to MAR and a SMAC-MAR check is made to insure that the address of the instruction ends in 0 or 5 and that the SMAC-MAR exchange is correct. The unit select register is set and the selected tape unit put into write status and started. The ready indicator is turned off and a disconnect signal sent to CPU to signify execution of the above.

WHEN THE TAPE REACHES FULL SPEED, the DS signals CPU that it is ready to begin writing. CPU interrupts its current processing task and causes MASR to be set to SMAC. A SMAC-MASR check will be made to insure a correct exchange of address, and a block of five characters will be read from memory into the IB. SMAC will be stepped plus five, and the contents of IB transferred in parallel to the OB.

AT THIS POINT, the IB is empty again and at once demands a machine cycle. MASR is set to SMAC, a SMAC-MASR check is performed and another block of five characters is read into the IB. SMAC is again stepped, whereupon CPU resumes its interrupted processing task.

THE OB WRITES the first block of characters serially onto the tape. As soon as OB is empty, the contents of IB are transferred to fill it, thus permitting IB to demand the next machine cycle. The write sequence is now established and continues until a whole group in memory has been written out.

WHEN A GROUP MARK IS RECOGNIZED on the output bus to the tape unit, the exit of characters from OB is stopped at once. The group mark is not written, but a check character is generated and written on tape.

IF THE INSTRUCTION IS UNDER RC CONTROL:

1. If RC equals zero initially, the PCT data check indicator and RC control indicator are reset off. The DS disconnects and tape unit is not started.
2. If RC is not zero initially, the PCT data check indicator is reset off, a SMAC-MAR check is performed, and the tape unit is started. When the tape reaches full speed, RC is stepped minus one and writing begins. (RC is stepped at this point so that it equals zero when the last record has been written.)
3. When a group mark is sensed in OB, a check character is generated and written as described above. If RC equals zero, the tape stops and the RC control indicator is reset off. If RC is not at zero, the tape continues to move; the DS suspends data transmission long enough to produce an inter-record gap on tape, and RC is stepped minus one. Characters remaining in the OB after the gap has been formed are then written out and the normal sequence of operations resumed. (Group marks present in the OB for the purpose of filling out an incomplete block of five characters are skipped over when writing is resumed.)

AT THE TERMINATION OF THE OPERATION, the contents of the lower four positions of SMAC are placed in the lower four positions of the OB. The 1, 2, and 4 bits in the high-order position of SMAC appear as A and B bits in the thousands-order and B bits in the units-order of OB, respectively. An 8-and-2 bit combination (zero) is placed in the high-order position of OB unless there is an LRCR error, in which case the contents of the LRCR are placed in the high-order position of OB.

THE READY INDICATOR IS TURNED ON.

AN ERROR on a SMAC-MASR or SMAC-MAR check, MBR code check, tape check (VRC, LRC, or match), or the writing of a redundant tape mark turns on the PCT data check indicator. The PCT address check indicator is not turned on and the machine does not stop.

DS OVERFLOW (WRAP-AROUND). When writing a record from memory through the DS and the end of memory (39,999 or 79,999) is exceeded, the following takes place:

1. The full record is written from both upper and lower memory onto tape, with no stoppage or cutoff occurring. SMAC steps in a normal manner.
2. The PCT data check indicator is turned on as the end of memory is exceeded. It is necessary, through a PCT routine, to determine that an overflow situation exists, rather than other possible types of PCT errors.
3. Writing under RC control operates as described under 1 and 2 above.

WR 01. Data flow via the DS in conjunction with a WR 01 instruction is essentially identical to the procedure described above under WR 00.

The DS remembers that it has been given a WR 01 and simply ignores the group mark character. It continues to write out information from memory, including group marks, until SMAC reaches 00000, 20000, 40000 or 60000. In other words, writing continues through the end of a particular memory, i.e., through position 19999, 39999, 59999, or 79999.

If the instruction is under control of the RC, multiple 20,000-character blocks of memory may be written out. If RC is not at zero when the end of memory is reached, a wrap-around occurs. The PCT data check indicator is turned on and writing continues.

Checking is performed as described under WR 00.

WR 02—Set Record Counter (SRC). WR 02 applies only to the DS. If this instruction is given to a serial I-O device, a WR 01 results.

If the DS is not available or not ready when the WR 02 instruction occurs, the execution of the instruction is held up. No disconnect signal is returned to CPU.

If the DS is ready or when it becomes ready, the PCT data check indicator is reset. The RC is set to the three-digit number contained in the tens-, hundreds-, and thousands-order of the instruction address; the RC control indicator is set to indicate that the next RD 00, RD 01, WR 00, WR 01, BSP 00, WTM or SKP is to be repeated the number of times it is set in the RC. When this has been done, a disconnect signal is sent to CPU.

This instruction may not be executed under RC control. A WR 02 following a WR 02 merely resets the RC to the number indicated in the second address.

Checking is performed as described under WR 00.

CONTROL FUNCTIONS

Control 00000 (IOF). This instruction is not executed until the selected DS is in ready status.

If the tape indicator of the selected tape unit is on, control 00000 causes it to be turned off. Once the indicator has been turned off, a disconnect signal is sent to CPU. If the indicator is off initially, an immediate disconnect occurs.

This instruction does not reset the PCT data check indicator, if on from a previous operation. It cannot be repeated under RC control and has no effect on the RC or the RC control indicator.

Control 00001 (WTM). This instruction is not executed until the selected DS is in ready status.

The control 00001 instruction turns off the ready indicator in the selected DS and causes a disconnect signal to be sent to CPU. Following disconnect, the tape mark character is emitted on the write bus, and the write circuits run for one character cycle. Finally, the DS writes the check character, i.e., another tape mark, and the ready indicator is turned on.

If the tape mark is written incorrectly, the PCT data check indicator is turned on as the character is sensed by the read gap.

This instruction resets the PCT data check indicator, if on from a previous operation. It may be repeated under RC control.

Control 00002 (RWD). This instruction is not executed until the selected DS is in ready status.

The control 00002 instruction puts the selected tape unit in read status and generates a rewind signal. Following this, a disconnect signal is sent to CPU. If the tape unit is already at its load point, an immediate disconnect occurs.

This instruction does not reset the PCT data check indicator, if on from a previous operation. It cannot be repeated under RC control and has no effect on the RC or the RC control indicator.

Control 00003 (ION). This instruction is not executed until the selected DS is in ready status.

If the tape indicator of the selected tape unit is off, control 00003 causes it to be turned on. Once the indicator has been turned on, a disconnect signal is sent to CPU. If the indicator is on initially, an immediate disconnect occurs.

This instruction does not reset the PCT check trigger, if on from a previous operation. It cannot be repeated under RC control and it has no effect on the RC or the RC control indicator.

Control 00004 (BSP). This instruction is not executed until the selected DS is in ready status.

There are two modes of control 00004, distinguished by accumulator coding or ASU 01 coding. The BSP instruction turns off the ready indicator in the selected DS and generates a BSP signal. Following this, a disconnect signal is sent to CPU. The PCT data check indicator (if on from a previous operation) is not turned off.

BSP 00—BACKSPACE (BSP)

The tape moves backward until an inter-record gap is detected at the read gap. It then stops with the head in the inter-record gap. When the load point is reached, or if already at load point, the tape indicator is turned on.

BSP 00 may be repeated under RC control. If the instruction is under control of the RC and RC equals zero initially, an immediate disconnect occurs; otherwise, the tape continues to move backward until RC reaches zero, and the beginning of an inter-record gap is detected at the read gap. Each time an inter-record gap is detected, RC is stepped minus one. (RC is also stepped initially so that it equals zero while the last record is backspaced.) When RC is at zero and the tape reaches the beginning of an inter-record gap, the tape stops with the head in the inter-record gap.

If the load point is reached or a tape mark is read while under RC control and RC has not yet reached zero, the RC control indicator is reset off and the multiple operation terminated. If the load point was reached, the tape indicator is turned on.

BSP 01—BACKSPACE FILE (BSF)

The tape moves backward until a tape mark is sensed (or until the load point is reached). The tape indicator is not turned on when a tape mark is read, but is turned on if load point is reached. This operation cannot be used under RC control and must have ASU coding of 01. ASU coding other than 01 is interpreted as 00.

Control 00009 (SKP). This instruction is not executed until the selected DS is in ready status.

The control 00009 instruction turns off the ready indicator in the selected DS and causes a disconnect signal to be sent to CPU. Following disconnect, the tape is skipped forward a small interval (about 47 milliseconds). During this time, the tape is erased. The ready indicator is then turned on.

This instruction is used with the two-gap head writing check; it allows a suspected bad spot on tape to be skipped over. The instruction may be repeated under control of the RC.

The PCT data check indicator, if on, is not reset by this instruction.

TRANSFER-ON-SIGNAL OPERATIONS

TRS 00—Transfer on Signal. The TRS 00 instruction interrogates the tape indicator in a selected tape unit. A program transfer occurs under the following conditions:

1. The selected DS is ready and the tape indicator in the selected tape unit is on. A NOP results if the selected tape unit itself is not ready, i.e., if tape is rewinding.
2. The selected DS is not ready, but the tape indicator is on in the tape unit, causing the DS to be busy.

The PCT data check indicator in the selected DS, if on, is not reset by this instruction.

TRS 01—Transfer on Ready (TRR). The TRS 01 instruction interrogates the ready indicator in the previously selected DS. If the indicator is on, a program transfer occurs.

The PCT data check indicator, if on, is not reset by this instruction.

TRS 02—Transfer on Transmission Check (TTC). The TRS 02 instruction refers only to the DS. If a serial I-O device is selected and TRS 02 is issued, a NOP results.

The instruction interrogates the PCT check trigger in the selected DS. The DS need not be in ready status when this occurs. If the trigger is on, a program transfer takes place.

The PCT data check indicator, if on, is not reset.

TRS 03—Transfer on Synchronizer Any (TSA). The TRS 03 instruction refers only to the DS. If a serial I-O device is selected and TRS 03 is issued, a NOP results.

The instruction interrogates both the PCT check trigger in the selected DS and the tape indicator in the selected tape unit. If either is on, a program transfer occurs.

TRS 03 is not executed until the DS is in ready status. The PCT data check indicator, if on, is not reset.

IBM 714 Card Reader and IBM 759 Card Reader Control

Functional Description

Data punched in standard IBM cards may be entered into the 705 system by the use of the IBM 714 Card Reader and IBM 759 Reader Control (Figure 95).

The card reader reads cards at the rate of 250 cards per minute. The selection arrangement and control of information to be read from cards is made by the

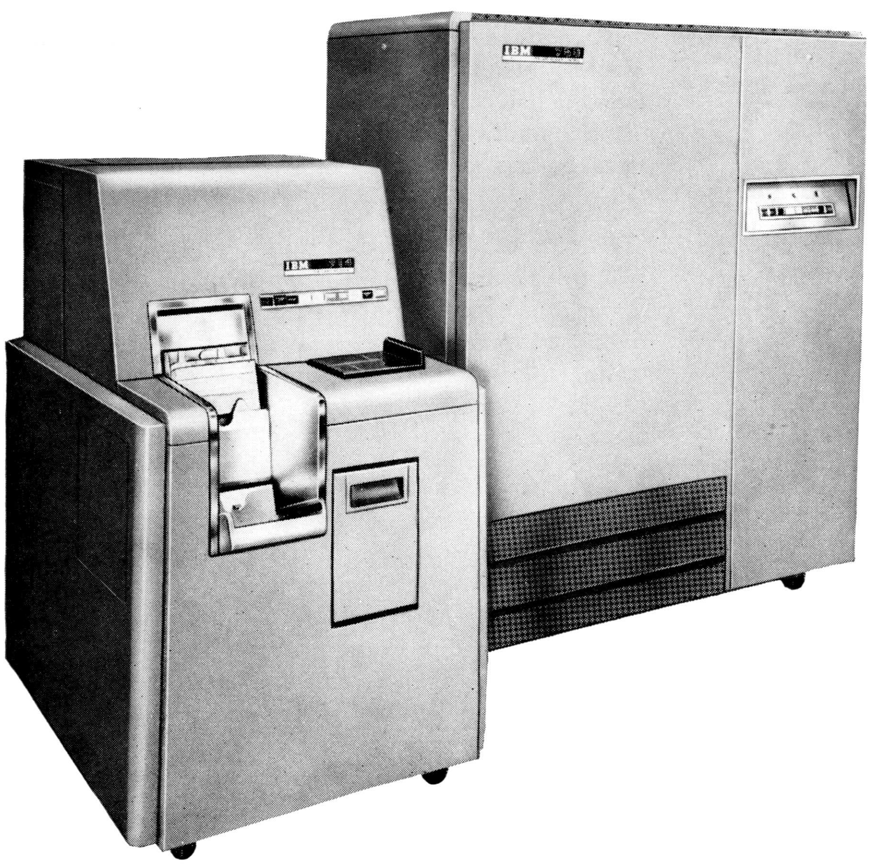


Figure 95. IBM 714 Card Reader and IBM 759 Card Reader Control

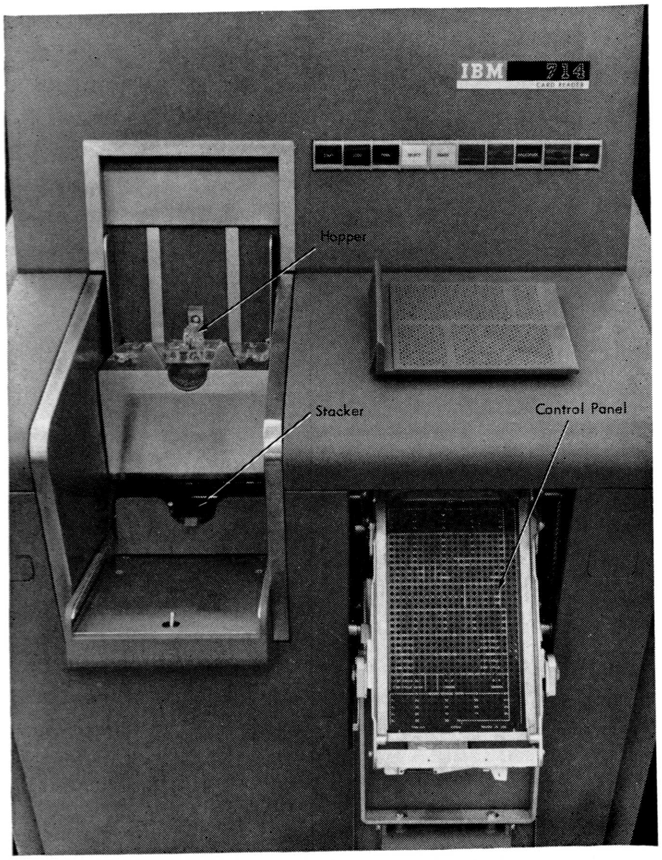


Figure 96. IBM 714 Card Reader

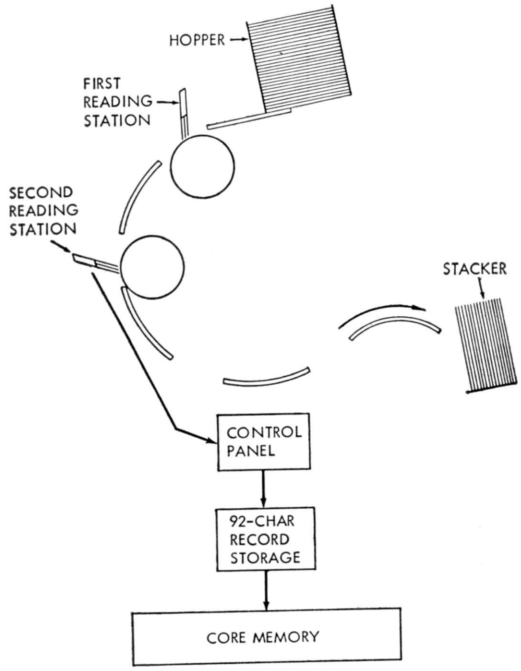


Figure 97. Schematic, Card Reader Feed

control panel which may be altered by use of external wires (Figure 96).

The cards to be processed are placed in the card hopper, 9 edge first, face down. The cards are mechanically moved through the card feed unit under reading brushes located at the first and second reading stations, and placed in the stacker (Figure 97). Information punched in the cards is sensed as the cards pass under the reading brushes.

The card data, as electrical impulses, are transferred from the brushes at the lower reading station, through the control panel (the wired positions), and are placed in the 92-character record storage unit (RSU) of the card reader. The information read from the card is transferred to the RSU, 80 columns in parallel, 9 row, 8 row . . . 0 row, 11 row, 12 row. Information may also be placed in the RSU, in addition to the card record, by emitting data from the control panel. The record format as it appears in the RSU resembles the record format of an imaginary 12-row 92-column card.

When a read instruction is directed to the card reader, the record is read from the RSU, character by character, converted to 705 code and transmitted to memory.

The RSU then refills with information from the next card, so that this record will be available when required. The 705 is free to execute subsequent instructions while the record storage unit is being refilled.

The sensing of a reader storage mark (a special character) during the reading of a record from RSU, stops the information flow from RSU to memory. For a reader storage mark to be placed in the RSU, it may be punched in the input record card or be emitted from the control panel. If a reader storage mark is not sensed in the record storage unit, all 92 positions of the RSU are read out and sent to CPU. The positions of RSU (between the first position and the reader storage mark) that do not contain information produce blank characters in memory. Normally, information is placed in RSU, beginning at the first position and continuing sequentially for as many positions as are needed. The reader storage mark then follows the last character of the record in RSU.

Read Check

As the card is read at the first reading station, an odd or even indication of the number of bits for each of the 12 horizontal rows of the card is developed. At the second reading station, the same record is placed in the record storage unit. When a read instruction is given, the card record in RSU is converted to the 705 code and sent to memory. The converted record is also reconverted to the IBM card code and an odd or even indication of the number of bits for each of the 12 horizontal rows of the reconverted record is developed (Figure 98).

The odd or even indications for each row of the record, developed at the first and second read stations, are compared. A difference in comparison turns on the read check indicator (reader and reader control operator's panels) and the read-write check indicator (00902) located on the 705 console.

If the 00902 indictor switch is turned to automatic, a read-write error indication automatically stops CPU operation after the read instruction is executed. When the 00902 indicator switch is set to PROGRAM, the read-write check indicator may be interrogated during the program and an error does not automatically stop the CPU. Refer to the section "End of file and Checking Procedures" for a detailed explanation.

As each character enters CPU from the card reader record storage unit, it is code checked in character register 2 before entering memory. A character code check error turns on the read-write check indicator (00902) on the 705 console.

Feed Check

If a card fails to feed from the hopper into the card feed unit or a card jam occurs in the card feed unit, a feed check error is indicated by the feed check light of the card reader. A feed check error removes ready condition and turns off the ready light. A subsequent read instruction to the reader results in a no-response indication (705 console); the CPU operation is suspended until the error condition is removed or another operation is initiated.

If a card jam occurs other than at the throat of the feed unit, access to the card feed unit can be gained by removing the upper front cover of the card reader. The reading brushes located at the upper and lower reading stations can be removed for visual inspection or removal of cards from the card feed unit. Before attempting to remove a card from the card feed unit, the operator should be fully instructed by an IBM customer engineer. If the feed check light remains on after the machine is apparently clear, notify a customer engineer.

Grouping

The grouping feature (alternator operation) provides a method by which data from two cards can be combined, rearranged, and selected to produce one record in the record storage unit. This makes possible the reading of information from two input cards and placing it in memory as one record.

Grouping is done by the alternator, serving as a selection device. The alternator makes possible the reading of as many as 46 identical columns of two cards. The selected columns to be read may be in any order or sequence desired. The alternator operates only when the group switch of the control panel is wired. The control panel has 46 alternator positions, each position separate and identical in operation (Figure 99A). Each alternator position has three hubs—common, first cycle, and second cycle.

During the first reading cycle (reading the first card at the second reading station), the common and first cycle hubs of each alternator position are internally

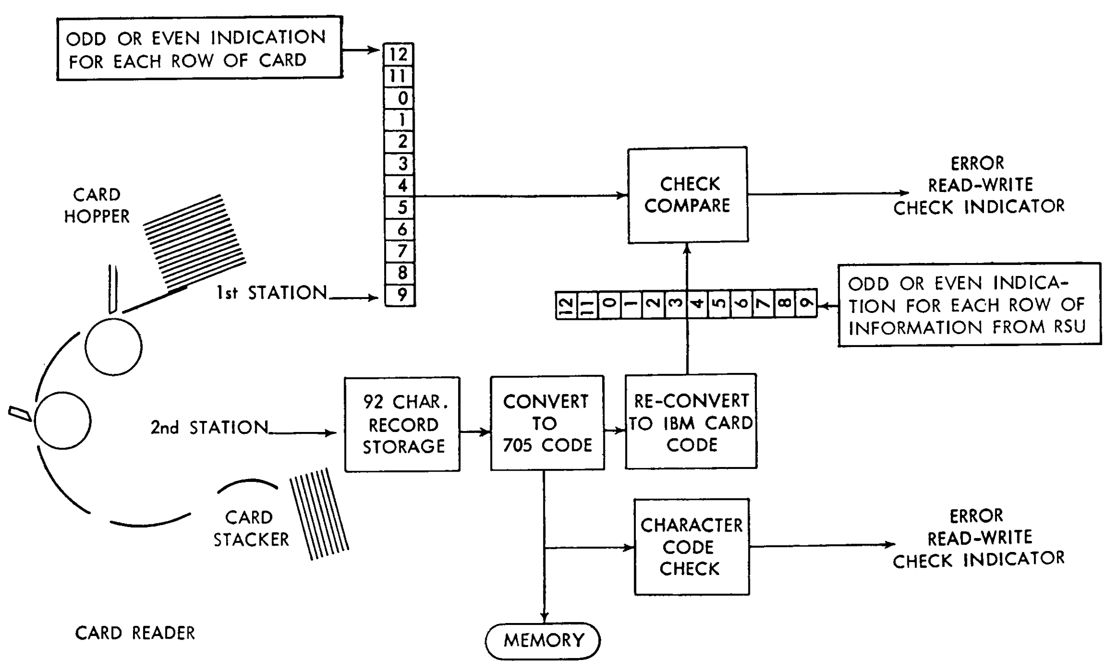


Figure 98. Schematic, Reader Checking

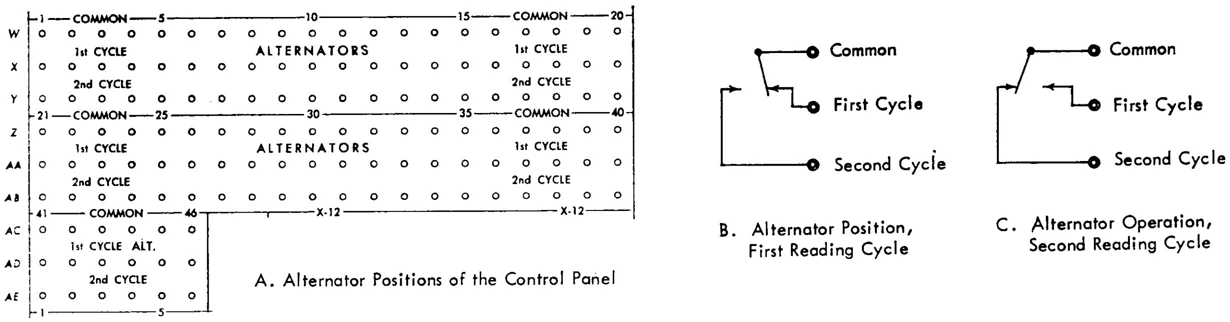


Figure 99. Alternator Operation

connected (Figure 99B). Data read from any one of the 80 columns of the card or a character emitted from the control panel may be entered into the common hub of any alternator position by wiring. The data are transferred through the internal connection to a related first cycle hub. The data from the first cycle hub may be transferred to any one of the 92 positions of the RSU by wiring.

During the second reading cycle (reading the second card at the second reading station), the common and second cycle hubs of each alternator position are internally connected (Figure 99C). The data entered in the common hub are transferred through the internal connection to a related second cycle hub. The information from the second cycle hub may be transferred to any of the 92 positions of the RSU not previously wired from the first cycle hub.

If blank columns of one card of the two-card group occupy identical punched positions of the other card, wiring from these positions can be direct to RSU without the use of the alternator.

Operating Procedures

Start. Before the reader can be used under program control, it must be in ready condition.

1. Place cards in the hopper, 9 edge first, face down.
2. Press the start key once. This feeds three cards into the card reader. (Group operation causes four cards to be run in.) At this time, the record storage unit contains the first record.
3. Press the start key again. This places the card reader in ready condition and turns on the ready light.

A read instruction directed to a selected card reader in ready condition places the record from RSU in memory and causes a card feed cycle to take place. If the hopper becomes empty or the stacker full, the ready light is turned off and the reader is removed from ready condition. Under these circumstances, a read instruction to the card reader results in a no response indication (705 console) and the CPU waits for the ready condition. Placing more cards in the hopper or removing cards from the stacker and depressing the start key restores the ready condition and the operation resumes.

End of File. When the last card of a file feeds from the hopper, the ready light is turned off and the card reader is removed from ready condition. Depressing the start key (card reader) allows the 705 to resume operation; the remaining cards are read as called for by program instructions. The input-output indicator for the card reader is turned on when the 705 executes the read instruction *following* the one that reads the last card from record storage to memory. The input-output is turned off by the IOF instruction or by running in the cards when starting an operation.

Error Correction. A read-write check (00902) or a feed check error using the card reader requires that the error condition be removed before the card reader can be used.

The following general restart procedure is suggested. It should be used with a composite chart (Figure 100) for error indication and procedures. It is important to perform work systematically when using error restart procedures.

1. In all error conditions, remove the cards from the hopper and stacker.
2. Use the feed key to run out the cards remaining in the card feed unit.

	(00902) Rd/Wr Check Error	Feed Check Error	Even Cycle Light	Good Cards	Rerun Cards	Stacked Rerun Card
Group Switch Not Wired	On	Off	Off	1	4	
	Off	On	Off	2	2	
	On	On	Off	1	3	
Group Switch Wired	On	Off	Off	0	5	1
	Off	On	Off	2	2	
	Off	On	On	1	3	
	On	On	Off	0	4	
	On	On	On	0	4	1

Figure 100. Error Correction Restart Procedure

a. Five cards will be in the card feed unit if a read check error occurs.

b. Normally four cards will be in the card feed unit if a feed check error occurs.

3. Check and process the run out cards as follows:

   a. The cards labeled "good cards" are the first cards to emerge from the feed unit and have been entered into memory correctly. Place these cards behind the cards removed from the stacker.

   b. The cards labeled "rerun cards" should be processed again. Place them ahead of the cards previously removed from the hopper.

   c. The cards labeled "stacked card" refer to the last card of the group removed from the stacker (before the run-out procedure). This card should be processed again by being placed ahead of the rerun cards.

   d. (Read Check with Group Switch Off). The first card of the "rerun group" should be checked for possible punching errors. If a read check occurs before any cards appear in the stacker, only four cards are run out in the restart procedure. All four cards are "rerun cards."

   e. (Read Check with Group Switch On). The first two cards of the rerun group should be checked for possible punching errors.

   f. (Feed Check). Repunch the card or cards that caused the feeding error. The usual feed check error is caused by the failure of a card to feed from the hopper.

4. Reload the rerun cards into the hopper.

5. Turn off error indicators.

   a. Use the reset key to turn off the read check light.

   b. Use the stop key to turn off the feed check light.

6. Follow normal start procedure.

Control Panel

The brush readings of the card feed unit are connected by external wires through the control panel to the RSU and the checking circuits. By use of control panel wiring, selection and arrangement of information to be read from the cards may be made. Also provided are such machine control features as column splits, digit selectors, digit emitters and alternators (grouping) (Figure 101).

The control panel has two kinds of hubs—exit and entry. An exit hub is one that emits an impulse. Some exits emit under control of the punched hole in

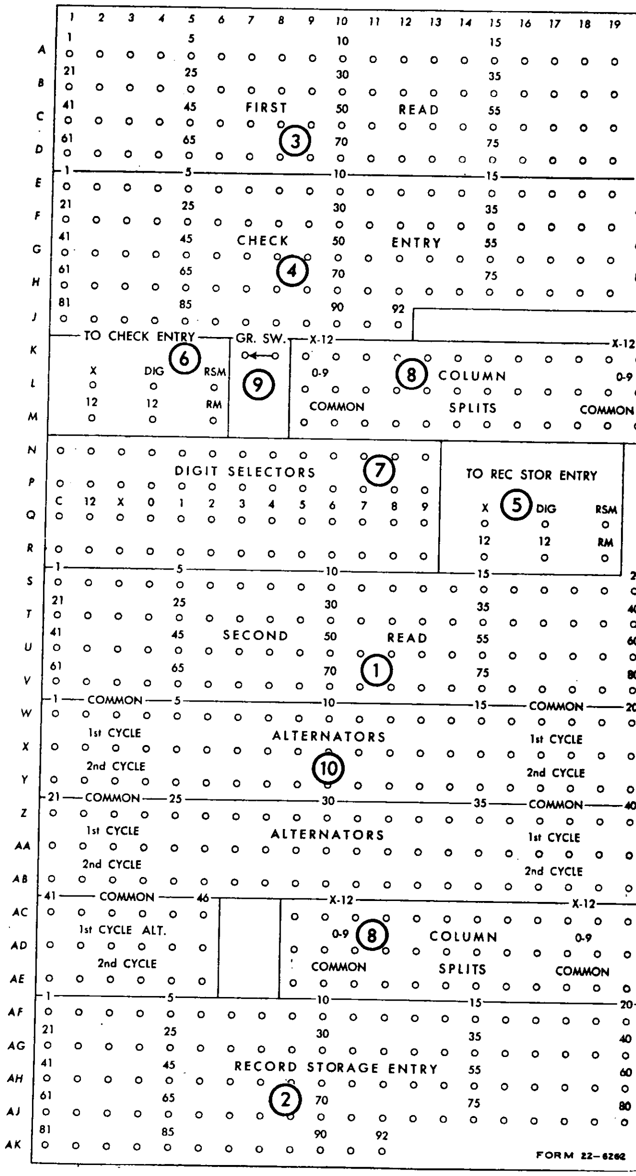


Figure 101. Control Panel Diagram

the card; some emit automatically during machine cycles. An entry hub is one that accepts an impulse wired to it.

A connection must always be made from an exit to an entry by inserting one end of a wire into an exit hub and the other end into an entry hub. Which exit and entry hubs are used depends upon the card columns being read, the record storage positions being filled, or what other functions of the card reader are being used for a particular job.

To easily refer to specific hubs on the control panel, the rows are numbered *1* to *20* horizontally and lettered *A* to *AK* vertically (Figure 101). Each of the following numbered paragraphs has a corresponding encircled number on the control panel diagram in Figure 101. Explanation of the control panel follows a sequence related to actual wiring practice.

1. Second Read (S-V, 1-20). Eighty second-read hubs, one for each of the 80 columns of the IBM card, emit impulses corresponding to the card punching sensed at the second reading station in the card reader. They are normally wired to the record storage entry hubs.

2. Record Storage Entry (AF-AJ, 1-20; AK, 1-12). Ninety-two record storage entry hubs corresponding to the 92 positions of the RSU receive impulses from the second read hubs, the record storage entry character emitter, column splits, alternators or the digit selectors. The selected impulses received may be wired in any desired arrangement into the record storage entry. However, to prevent writing blank characters (C and A bits) on tape, record storage entry should start in position 1 and continue sequentially for as many positions as required. The RSM should follow the last used position of record storage entry unless all positions of record storage entry are being used.

3. First Read (A-D, 1-20). The first-read hubs represent the 80 columns of the card. These hubs emit impulses corresponding to card holes sensed at the first reading station in the reader. They are normally wired to the check entry hubs.

4. Check Entry (E-H, 1-20; J, 1-12). The 92 check entry hubs receive impulses emitted from the first-read hubs, the check entry character emitter, column splits, alternators or digit emitters. The check entry hubs represent 92 possible data checking positions comparable to the 92 record storage entry positions. An odd or even indication of the number of bits in each row of the record entered in check entry is developed and is compared with the odd or even indication of the number of bits for each row of the record as read from RSU. All data wired into check entry should be identical to the data wired into record storage entry; if not, a read check error occurs.

NOTE: If a single reader storage mark is punched in the card, the column of the card containing the RSM must not be wired from the first read hubs to the check entry hubs; otherwise, a read check error occurs. If two reader storage marks are punched in the card and wired from the first read hubs to the check entry hubs, normal operation results.

5. Record Storage Entry Emitters (Q, R, 15, 17, 19). The record storage emitters permit the reading of impulses directly into the record storage. The impulses are generated within the reader during each card cycle and are not the result of holes punched in the card. The X hub emits an impulse corresponding to the 11 hole in the IBM card. The 12 hub impulse corresponds to the 12 hole in the card. Because of their wide usage, two 12 hubs are provided. The digit hub emits a series of impulses, 12 through 9, for each card cycle. The impulses may be selected by wiring into the common hub of a digit selector and from any of the digit hubs to check entry. The RSM hub emits a reader storage mark (12-1-4-7) and the RM hub emits a record mark.

Frequently, it is necessary to write information on tape that is not punched in a card. With a device called an emitter, the machines can manufacture impulses just like those obtained when a card column with every position punched is read by a brush.

The digit emitter is basically a switch with 12 timed positions (Figure 102). Each position is wired to the control panel and numbered to correspond with a punching position.

The switch is operated by the machine so that the 12 positions are connected to the common, one at a time. Thus, the 12 hub emits an impulse at 12 time, the 9 hub at 9 time, and so on. The common is the source of power and is internally connected.

6. Check Entry Emitters (L-M, 2, 4, 6). These hubs emit the same impulses as the record storage entry emitters except that they are wired to the check entry.

7. Digit Selectors (N-R, 1-13). Four digit selectors are provided. Each digit selector consists of a common hub and 12 hubs labeled for the 12 punching positions of the card. On every machine cycle, the C hub is internally connected successively to the 9, 8, 7 . . . 12 hubs. When a first or second read hub representing a card column is connected to the C hub, a specific punching position can be read from the corresponding hub of the digit selector.

When the digit emitter hub of either the check entry or record storage entry emitters is connected to the C hub, the digit selector becomes a digit emitter and may be used to generate numbers, letters or special characters. These special characters may be entered in both check entry and storage entry hubs, depending upon which entry emitter originates the digit impulse.

The emitter shown in Figure 102 has the source of power (common) permanently wired. The digit se-

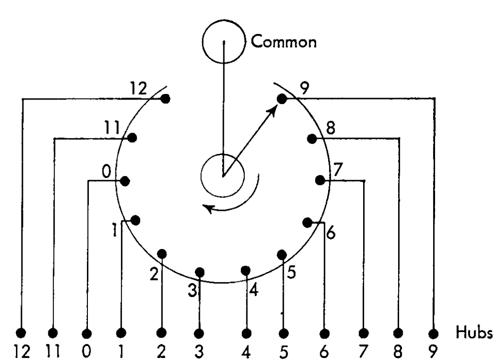


Figure 102. Schematic, Digit Emitter

lector operates in exactly the same manner as the digit emitter except that the common is not internally connected. This gives great flexibility in allowing the operator to wire the source of power (common) into the emitter.

8. Column Splits (K-M, AC-AE, 9-20). Each of the 24 column splits has three hubs: C (common), 0-9, and X-12. They separate X and 12 impulses from 0-9 impulses obtained from the card or as emitted impulses from the control panel.

Often, it is necessary to divide the punches of a column into two separate groups, 0-9 and X-12. This may be done by use of a column split. The column split on the control panel consists of common, 0-9, and X-12 hubs (Figure 103). From 0 to 9 time, the common hub is connected to the 0-9 hub. From X-12 time, the common hub is connected to the X-12 hub. Any of the hubs of the column split may be used for an exit or an entry hub. When using a column split, be careful to use the common, 0-9, and X-12 hubs belonging to the same column split position.

9. Group Switch (K, 7-8). To use the group feature of the reader when two cards are to be read as one record, the two hubs of the group switch are wired together.

10. Alternator (W-AB, 1-20; AC-AE, 1, 6). The alternator (used during group operation) serves as a selection device, reading the first card of the group into one section of the RSU, and the second card into a second designated section of the RSU. Three rows of hubs are provided—common, first cycle, and second cycle. The card columns to be read are wired from second read to common. During the first card reading cycle, the common and first cycle hubs are internally connected. Therefore, card impulses wired into the common hubs are emitted from the corresponding first cycle hubs and from there may be wired to the position of record storage, where the first card record is to be stored.

During the second reading cycle, the common and second cycle hubs are connected internally. Card impulses wired into the common hubs are emitted from the corresponding second cycle hubs and from there are wired to the position of record storage where the second record is to be stored.

Fundamentally, the alternators are like the column splits. The main difference between the two is that the column splits are activated automatically while the alternators are activated only from control panel wiring (group switch).

Control Panel Wiring

Wiring for Card Reading

Card columns 1 through 20 are to be read into record storage. A group mark is emitted as a special character and the reader storage mark is placed in the 22nd position of record storage. Each of the following numbered paragraphs describes the corresponding wires on the diagram in Figure 104.

1. Columns 1 through 20 are wired from first read to hubs 1 through 20 of check entry.

2. Columns 1 through 20 are wired from second read to hubs 1 through 20 of record storage entry.

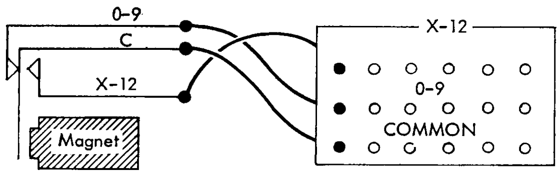


Figure 103. Schematic, Column Split Assembly

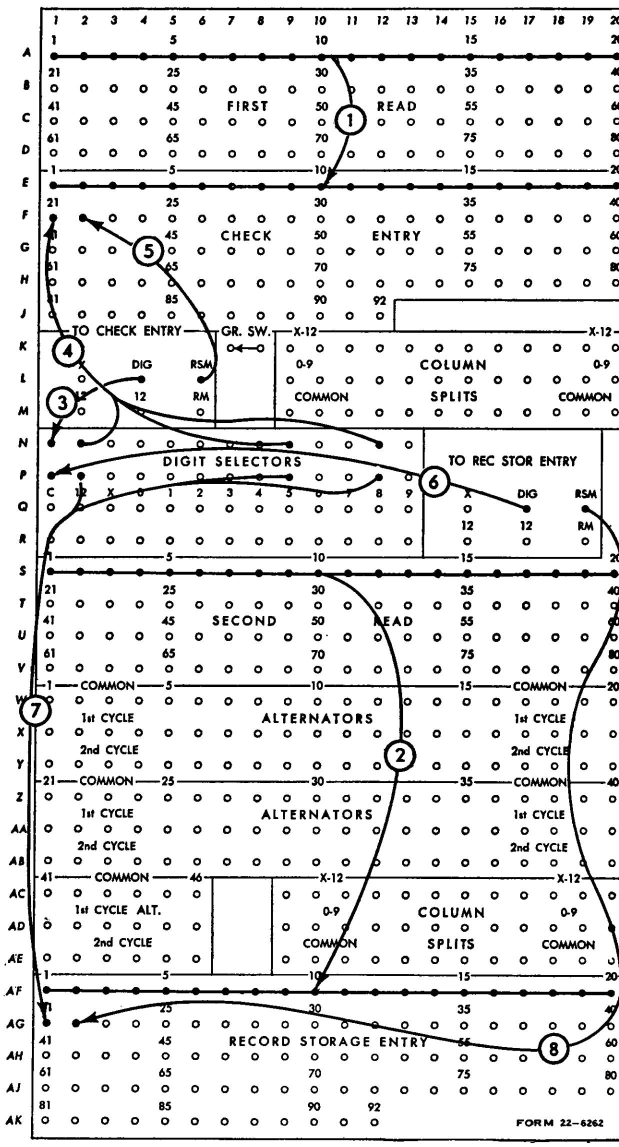


Figure 104. Wiring, Card Reading

3. The digit hub of the check entry emitter is wired to the C hub of the digit selector.

4. The 12, 5 and 8 hubs of the digit selector are wired to the 21st position of check entry using a split wire (wire with more than two ends). This emits the group mark as a special character into the check entry.

5. The RSM hub from the check entry emitter is wired to the 22nd position of check entry.

6. The digit hub of the record storage entry emitter is wired to the C hub of a digit selector.

7. The 12, 5, and 8 hubs of the digit selector are wired to the 21st position of the record storage entry.

8. The RSM hub from the record storage entry emitter is wired to the 22nd position of the record storage entry.

Wiring, Card Reading, Signing Fields (Figure 105)

Field A is read from card columns 71-74. The field is to be signed plus. Field B is read from card columns 76-79 and is to be signed minus.

1. Field A is wired from first read to positions 1-4 of the check entry. Column 75 from first read is wired to the 0-9 hub of a column split. The common hub of the column split is wired to the fifth position of check entry.

2. A 12 impulse (plus) is wired to the X-12 hub of the same column split. The character read into position 5 of the check entry is the numerical punching from the card with the 12 zone included to sign the field plus in the check entry.

3. Field B is wired from first read to positions 6-9 of the check entry. Column 80 from first read is wired to the 0-9 hub of a column split. The common hub is wired to the tenth position of check entry.

4. An X impulse (minus) is wired to the X-12 hub of the same column split. The character read into position 10 of the check entry is the numerical punching from the card with the X zone included to sign the field minus in the check entry.

5. The RSM hub of the check entry emitter is wired to a position of check entry.

6. Field A is wired from second read to positions 1-4 in the record storage entry. Column 75 from second read is wired to the 0-9 hub of a column split. The common hub of the column split is wired to the fifth position of the record storage entry.

7. A 12 impulse (plus) is wired to the X-12 hub of the same column split. The character read into position 5 of the record storage entry is the numerical punching from the card with the 12 zone included to sign the field plus memory.

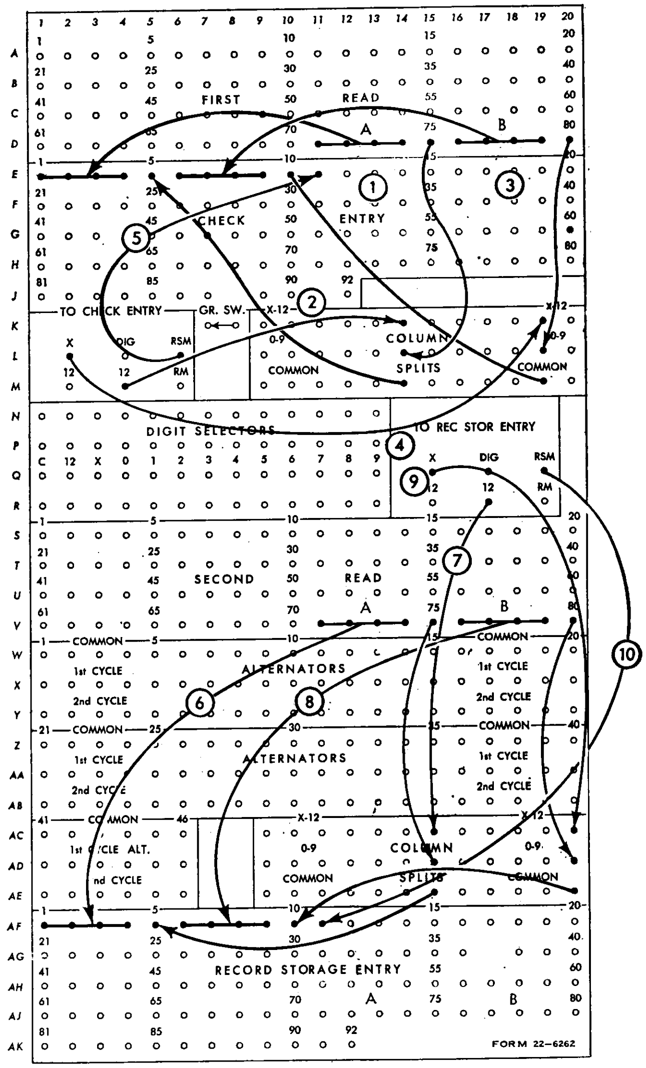


Figure 105. Wiring, Card Reading, Signing Fields

8. Field B is read from second read to positions 6-9 in the record storage entry. Column 80 from second read is wired to the 0-9 hub of a column split. The common hub is wired to the tenth position of record storage entry.

9. An X impulse (minus) is wired to the X-12 hub of the same column split. The character read into position 10 of the record storage entry is a combination of the numerical punching from the card and the X zone emitted from the character emitter. The field is, therefore, always signed minus.

10. The RSM hub of the record storage emitter is wired to the eleventh position of record storage entry.

Wiring, Card Reading, Grouped Records

Two card records are read into record storage and transferred to tape as one record. Identical columns are read from each card of the group.

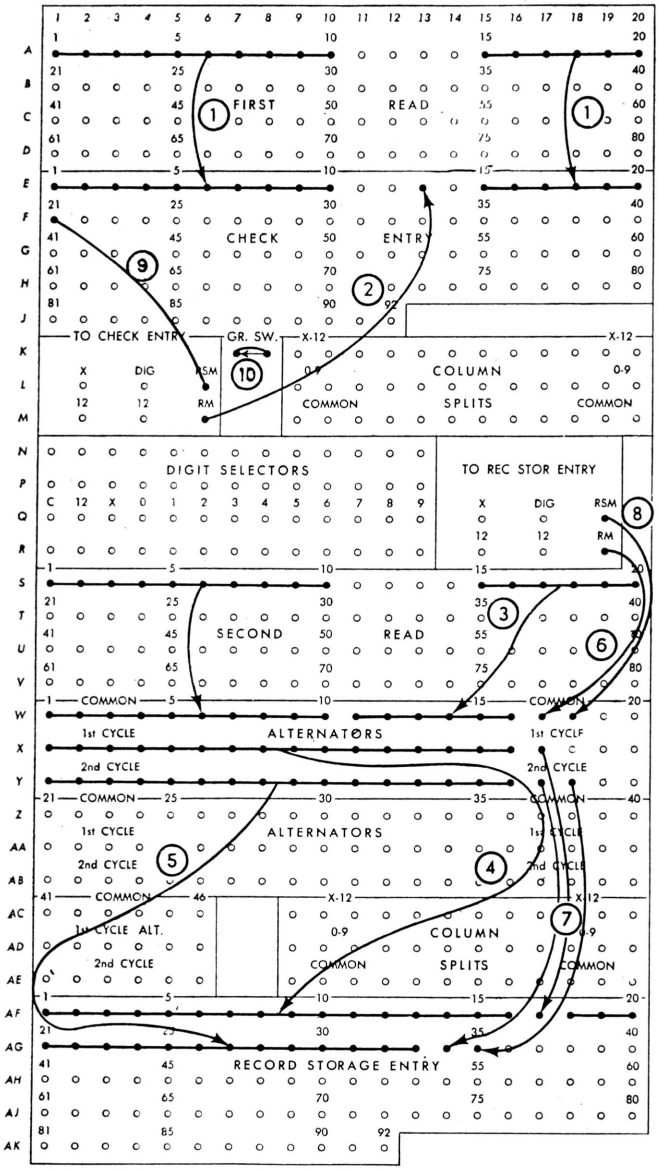


Figure 106. Wiring, Card Reading, Grouped Records

Columns 1-10 and 15-20 are to be grouped from two card records into RSU, by the wiring shown in Figure 106.

1. Columns 1-10 and 15-20 are wired from first read to comparable positions of check entry.

2. A check entry record mark is wired to hub 13 of check entry.

3. Columns 1-10 and 15-20 are wired from second read to the common hubs of the alternator.

4. The sixteen hubs of first cycle, corresponding to the 16 common hubs wired from second read, are wired to positions 1-16 of record storage entry.

5. The sixteen hubs of second cycle, also corresponding to the sixteen common hubs wired from second read, are wired to positions 18 through 33 of record storage entry.

6. A record mark is wired from the record storage emitter to a common hub of position 17 of the alternator.

7. A record mark is wired to positions 17 and 34 of record storage entry from the alternator first and second cycle hubs.

8. The record-storage-entry RSM hub is wired to a common hub of the alternator. From the second cycle of the corresponding alternator hub, the RSM is wired to the 35th position of record storage entry.

9. The check-entry RSM hub is wired to position 21 of check entry. The RSM is placed only once in both check and record storage entries.

10. The two hubs of the group switch are wired together.

IBM 714 Operating Keys and Indicator Lights

The operating keys and indicator lights on the operator's panel of the card reader are shown in Figure 107.

Start Key. This key has two distinct functions.

1. When the card reader is first prepared for operation, it is necessary to depress the start key twice, once to feed three cards and once to turn on the ready light, making the card reader available for operation under 705 control. Depressing this key has no effect if the ready light is already turned on.

2. When the last card has been fed from the hopper, it is necessary for the operator to depress the start key before the operation can proceed under 705 control. If the card file is not at an end, the operator replenishes the supply of cards in the hopper before depressing the start key. If the card file is at an end, depressing the start key allows the 705 to resume operation.

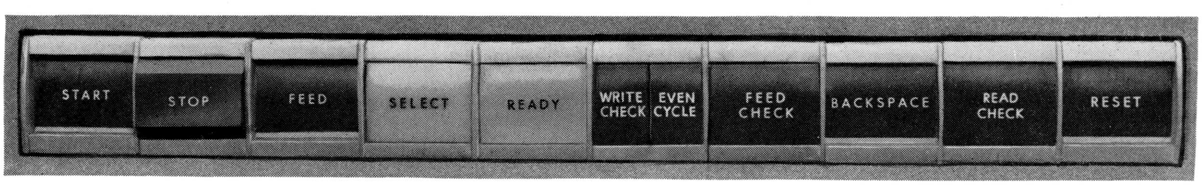


Figure 107. IBM 714 Operator's Panel

Stop Key. Depressing this key turns off the ready light and causes the card reader to stop operating. After a feed error, it can be used to turn off the feed check light after all the cards have been removed from the feed and hopper.

Feed Key. This key provides a manual feed without reading the cards in the machine. It is operative only when the ready light is not on.

Select Light. This light goes on when the card reader is selected by the 705 and remains on until another input-output unit, check indicator, or alteration switch is selected.

Ready Light. The ready light is on whenever the calculator interlock (ready condition) is turned on. It indicates that the card reader is ready for operation under control of the 705.

Write Check Light (Auxiliary Operation Only). The write check light turns on when a writing error is detected. A write check light on the IBM 759 Card Reader Control performs the same function.

Even-Cycle Light. This light is used with the feed-check light on alternator operation. On alternator operation, the feed-check light alone indicates a feed error on the odd card. With both the feed check and the even-cycle lights on, the feed error must have occurred on the even card.

Feed Check Light. A card jam or failure to feed cards turns on this light and turns off the ready light. The light is reset by removing the cards from the hopper, depressing the feed key to clear the machine, and depressing the stop key.

Backspace Key (Auxiliary Operation Only). The backspace key backspaces the tape one record for each depression. Records which have been recorded incorrectly may be backspaced and rewritten. A backspace key on the control unit performs the same function.

Read Check Light. The read check light turns on when a card reading error is detected. A read check light on the control unit performs the same function.

Reset Key (Auxiliary Operation Only). Depressing the reset key turns off the read or write check indicator lights. Depressing the reset key also turns off the tape indicator light (tape unit).

IBM 759 Reader Control

The control unit contains the necessary decoding, checking, and timing circuits for the transfer of data from the card reader to the CPU. The control unit is attached to the CPU by the I-O cable. The control unit also supplies the necessary electrical power for the operation of the control and the attached reader.

Each card reader has a power and signal cable that must be attached to the control unit, the signal cable carries the data and coordinating signals. The power cable carries the electrical power required.

The operating keys and indicator lights on the operator's panel of the reader control are shown in Figure 108.

CPU-Tape Switch. The two positions are CPU and TAPE. In all operations with CPU, this switch must be set to CPU (the up position).

Power-On Key. When the power-on key is depressed, the various power supplies in the control unit are made operative in a sequential process that takes approximately four minutes to complete.

Power-On Light. The power-on light is on whenever the power-on key has been depressed and stays on as long as power is on.

DC-On Light. The DC (direct current) on light automatically lights after the power-on key is depressed and the power-on sequence is completed. The light remains on until a power-off condition is present.

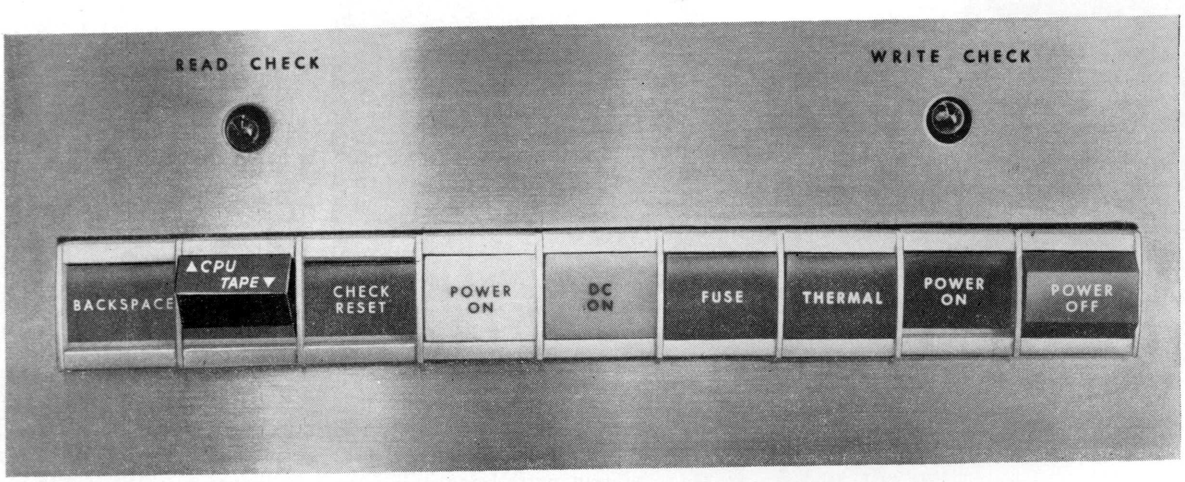


Figure 108. IBM 759 Operator's Panel

Power-Off Key. Depression of the power-off key removes all electrical power from the control unit and the reader.

Fuse Light. This light, when on, indicates that a fuse (protective device) has burned out. This condition forces a power-off condition. The customer engineer should be notified if this light is on.

Thermal Light. This light, when on, indicates that the operating temperature of the machine is too high. This condition forces a power-off condition. The customer engineer should be notified if this light is on.

The read check and write check indicators, the backspace, and check reset keys are similar in function to those of the 714 operator's panel.

IBM 722 Card Punch and IBM 758 Card Punch Control

Functional Description

The IBM 722 Card Punch and IBM 758 Card Punch Control are used to prepare IBM cards as output records from 705 memory (Figure 109). The information is punched in standard 80-column cards at the rate of 100 cards a minute.

When a write or write-erase instruction is directed to the card punch, the record in memory is transmitted, character by character, to the 80-character record storage unit (RSU) of the punch. The record received from memory is translated from the 705 code to the IBM card code before it is placed in the RSU. As soon as the end of a record in memory is sensed (group mark), CPU is free to execute subsequent instructions while the record in RSU is punched in a card. Information is punched in the card in the same order in which it is sent from memory.

The maximum record length that can be punched on a single card is 80 characters. Records that are longer than 80 characters are punched in successive cards with a single write or write-erase instruction. In this case, CPU is not free to continue with the program until the last block of characters (of the complete record) have been entered into RSU as indicated by the sensing of a group mark in memory.

Cards are placed in the card hopper 12 edge first, face down and move through the card feed unit to the punching station (Figure 110). At the punching station, the information from RSU is punched in the card, a row at a time (12, 11, 10, 1 . . . 9). After the card is punched, it is read at the checking station for checking purposes and is placed in the card stacker.

Figure 109. IBM 722 Card Punch and IBM 758 Card Punch Control

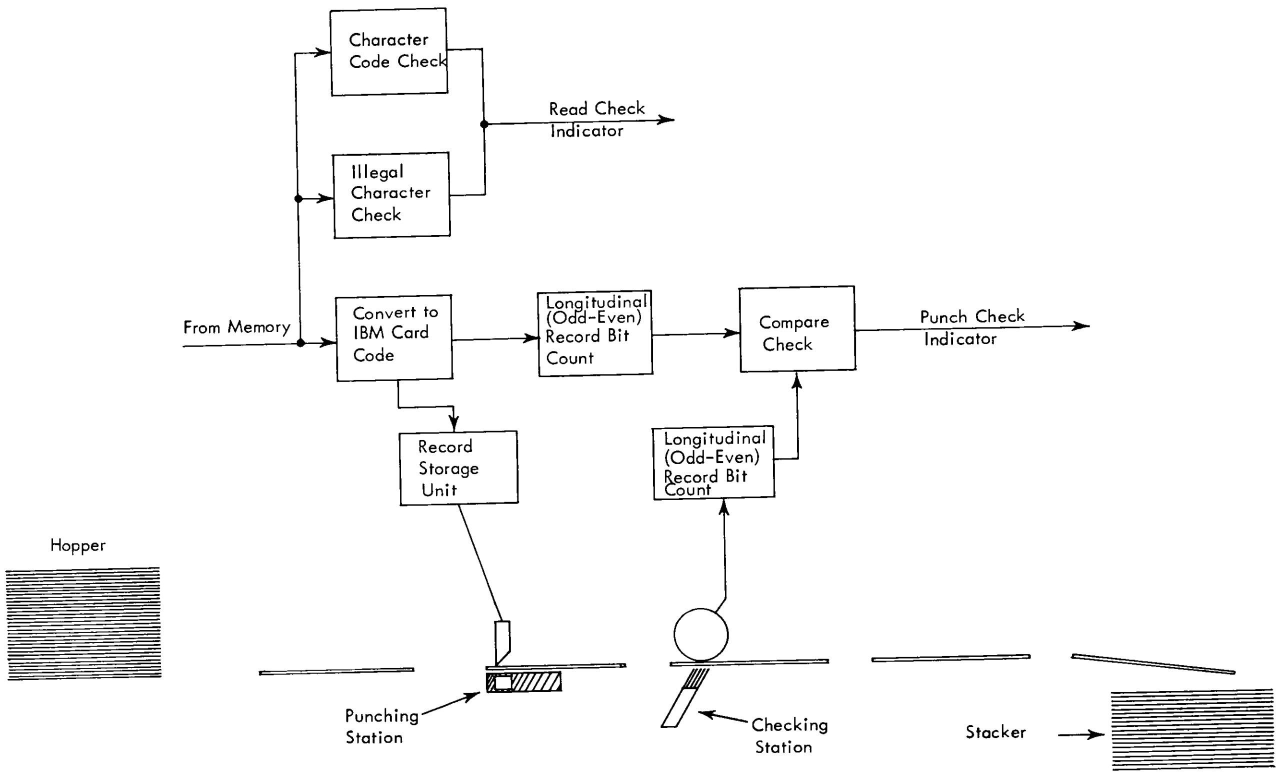


Figure 110. Schematic, Tape-to-Card Feeding and Checking

Non-valid characters received from CPU are entered into the RSU as blanks and result in blank columns on the punched card. The card punch has no signal device corresponding to the input-output indicator of other input-output units. If the card punch runs out of cards, a subsequent write or write erase instruction directed to the punch results in a no response indication (705 console); CPU suspends operation until the condition is removed or another operation is initiated.

Read Check

Each character received by the punch control from memory is given a character code check and an illegal character check. An illegal character or a character code check is indicated by the read check indicator (punch and punch control operator's panels). A read check error automatically turns on the read-write check indicator (00902) on the 705 console, which may be interrogated by programming to detect the error.

Punch Check

An odd or even indication of the number of bits for each row of the translated record placed into RSU is compared with an odd or even indication of the number of bits for each row of the card read at the checking station. A difference in comparison indicates a punch check error and turns on the punch check indicator. A punch check error automatically turns on the record check indicator (00903) on the 705 console.

The actual checking of a punched card takes place during the cycle following the one in which the card is punched. For example, at the time the punch check error is detected for record one, record two has been punched and record three has been placed in the RSU.

Feed Check

If a card fails to feed from the hopper into the card feed unit or a card jam occurs in the card feed unit, a feed check error is indicated by the feed check indicator (punch and punch control operator's panels). A feed check error removes the ready condition and turns off the ready light of the punch. A subsequent write or write erase instruction to the punch results in a no response indication (705 console); the CPU operation is suspended until the error condition is removed or another operation initiated.

Operating Procedures

Start. Before the punch can be used under program control, it must be in ready condition.

1. Place blank cards in the hopper, 12 edge first, face down.
2. Press the start key once. This feeds two cards into the card feed unit.
3. Press the start key again. This turns on the ready light, indicating that the punch is available for operation under CPU control.

If the hopper becomes empty or the stacker full, the ready light is turned off and the punch is removed from ready condition. Under these circumstances, a write or write erase instruction to the card punch results in a no response indication (705 console); CPU operation is suspended until ready condition is restored. Placing more cards in the hopper or removing cards from the stacker and depressing the start key restores the punch ready condition and the operation resumes.

End of File. After the last record has been punched:

1. Remove excess cards from the card hopper.
2. Use the feed key to run out three cards remaining in the card feed unit. Two blank cards will follow one punched card. The punched card is checked during the run-out.

Error Correction

READ CHECK. The 00902 read-write check indicator of CPU is turned on as a result of a read check error in the punch. The indicator is turned on before the record containing the error is punched. If the read-write check indicator is interrogated by a TRS instruction immediately following the write instruction, punching can be prevented by a SUP instruction. Record storage can then be reloaded with the same record and another trial for error can be carried out. Punching can be prevented until record storage is correctly loaded.

Punch suppression for long records (records requiring more than one card) cannot be initiated until the complete record has been received by the punch. If a read check error occurred on the first card of a two-card record, only the characters for the second card can be punch suppressed. The first card will be punched in error. The punch suppress operation does not alter the normal operation of the punch.

If the read-write indicator switch is turned to AUTOMATIC and a read check error occurs, the record in error is punched before the operation is stopped. This results in a subsequent 00903 check indication.

PUNCH CHECK. The 00903 record check indicator of CPU is turned on as a result of a punch check error in the punch. If the 00903 indicator switch is set to AUTOMATIC, CPU stops after the write or write erase instruction in which a punch check error is detected. If the switch has been set to program control, any action taken must be written in the program. In any case, at the time the error is detected, the record following the card in error has been punched and the record of the immediate write or write-erase instruction is in the punch RSU.

FEED CHECK. A feed check error in the punch results in a no-response indication in CPU as a subsequent write or write erase instruction directed to the punch. The following procedure is suggested in case of a feed check error. For feed check errors:

1. Remove cards from the hopper and stacker.
2. Depress feed key until cards are cleared from the feed unit. The usual feed check error is caused by the failure of a card to feed from the hopper. With this condition, two cards (one punched card followed by a blank card) are placed in the stacker.
3. Reload blank cards in the hopper (remove any nicked cards).
4. Use stop key to turn off the feed check indicator.
5. Follow normal start procedure.

If a card jam occurs other than at the throat of the feed unit, access to the card feed unit can be gained by removing the punch die and checking brushes. Before attempting to remove cards from the card feed unit, the operator should be fully instructed by an IBM customer engineer. If the feed check light remains on after the machine is apparently clear, notify a customer engineer.

IBM 722 Operating Keys and Lights

The operating keys and indicator lights on the operator's panel of the punch are shown in Figure 111.

Start Key. This key is provided for the run-in of cards. When the hopper is first filled, depressing the start key once causes one card to be fed to the punching station. A second depression turns on the ready light, making the card punch available for operation under 705 control.

Stop Key. Depressing this key causes the ready light to be turned off and the card punch to stop operating. After a feed error, it is also used to turn off the feed check light after all the cards have been removed from the feed and hopper.

Feed Key. This key provides a manual feed without punching the cards in the machine and is operative only when the ready light is turned off.

Select Light. This light is on whenever the card punch is selected and remains on until another input-output unit, check indicator, or alteration switch is selected.

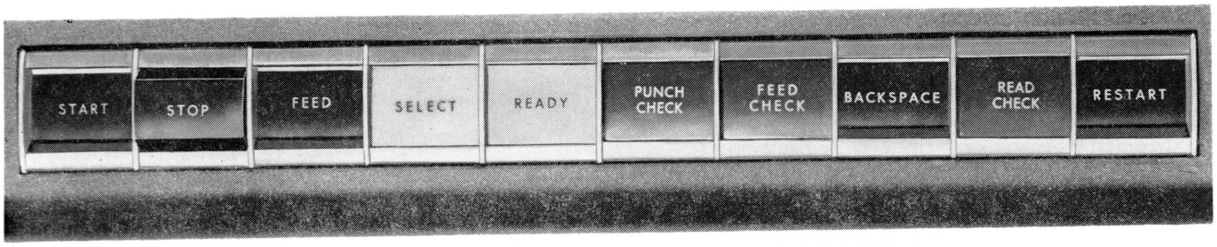


Figure 111. IBM 722 Operating Keys and Lights

Ready Light. The ready light is on whenever the card punch is ready for operation under control of the 705.

Punch Check Light. This light is turned on when an error is revealed by the check performed on punching. The punch check light reveals an error between the record punched and the record as it was stored in record storage.

Backspace Key (Auxiliary Operation Only). Depressing this key backspaces the tape one record. Records that have been read erroneously may be backspaced and reread.

Read Check Light. A read check error turns on the read check light and turns off the ready light.

Restart Key (Auxiliary Operation Only). This key is used to restart operations stopped by a read check or a print check error. One depression of the restart key turns off the error indicator light and restarts the operation. The restart key is also used to turn off the tape-indicate-on light, if on. The start key must be used to resume operation after turning off the tape-indicate-on light.

IBM 758 Punch Control

The control unit contains the necessary decoding, checking, and timing circuits for the transfer of data from the CPU to the punch. The control unit is attached to the CPU by the I-O (input-output) cable. The control unit also supplies the necessary electrical power for the operation of the punch control and the attached punch.

Each punch has a power and signal cable that must be attached to the control unit; the signal cable carries the data and coordinating signals. The power cable carries the electrical power required.

The operating keys and indicator lights on the operator's panel of the punch control are shown in Figure 112.

CPU-Tape Switch. The two positions are CPU and TAPE. In all operations with CPU, this switch must be set to CPU (the up position).

Power-On Key. When the power-on key is depressed, the various power supplies in the control unit are made operative in a sequential process that takes approximately four minutes to complete.

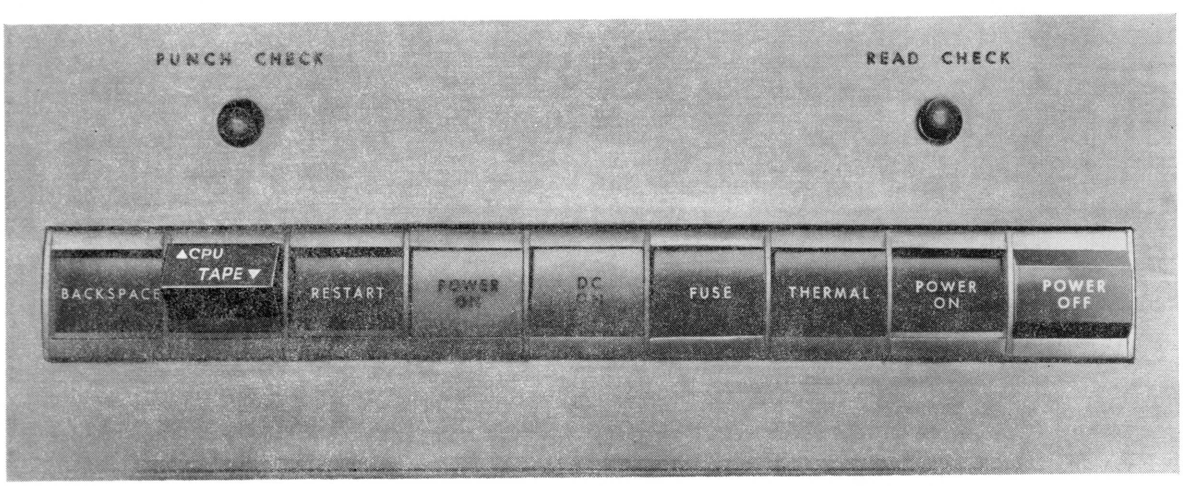


Figure 112. IBM 758 Operator's Panel

Power-On Light. The power-on light is on whenever the power-on key has been depressed and stays on as long as power is on.

DC-On Light. The DC (direct current) on light automatically lights after the power-on key is depressed and the power-on sequence is completed. The light remains on until a power-off condition is present.

Power-Off Key. Depression of the power-off key removes all electrical power from the control unit and the punch.

Fuse Light. This light, when on, indicates that a fuse (protective device) has burned out. This condition forces a power-off condition. The customer engineer should be notified if this light is on.

Thermal Light. This light, when on, indicates that the operating temperature of the machine is too high. This condition forces a power-off condition. The customer engineer should be notified if this light is on.

The punch check and read check indicators, backspace, and restart keys are similar in function to those of the 722 operator's panel.

IBM 717 Printer and IBM 757 Printer Control

Functional Description

The IBM 717 Printer and IBM 757 Printer Control are used for the direct printing of information from memory (Figure 113). The printer prints as many as 120 characters per line at the rate of 150 lines per minute. A paper tape controlled carriage associated with the printer automatically controls the feeding and spacing of forms while documents or reports are being prepared on the printer.

During the writing operation with the printer, data are transmitted, one character at a time, from memory to the 120-character record storage unit (RSU) of the printer. Data are converted from 705 code to IBM card code before being placed in the RSU. The information is read from the RSU in parallel, setting the type wheels to the indicator character and printing as a 120-character record. One print wheel (Figure 114) is related

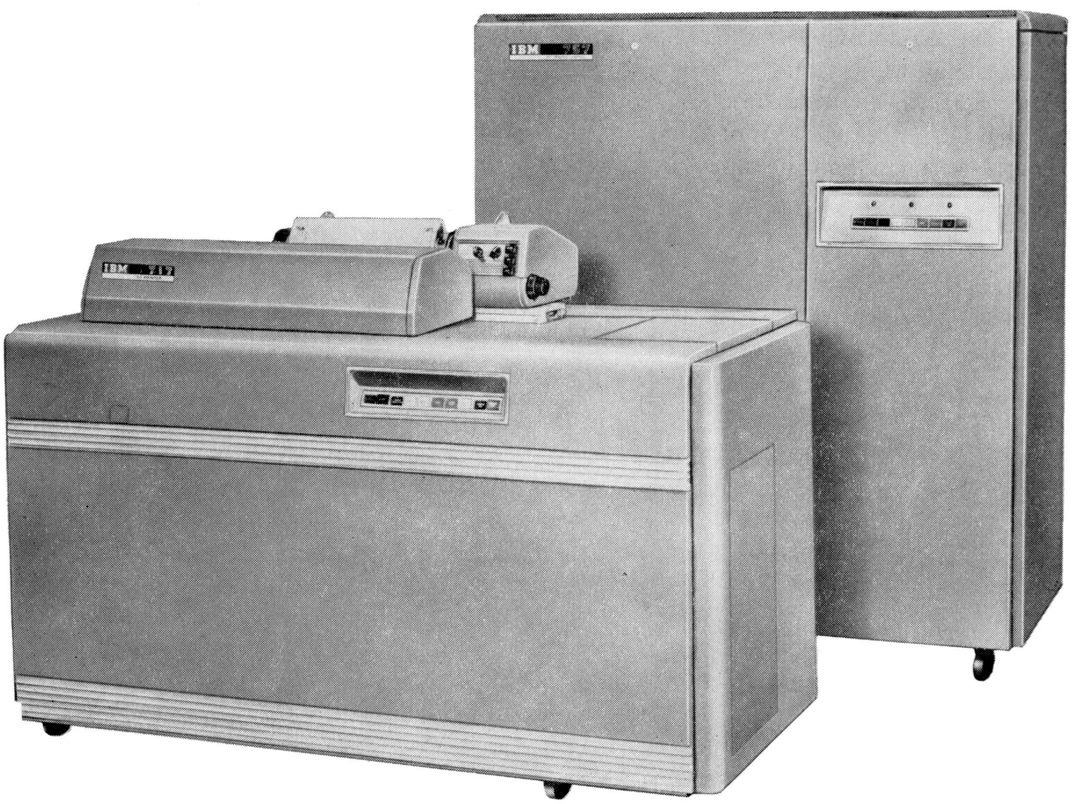


Figure 113. IBM 717 Printer and IBM 757 Printer Control

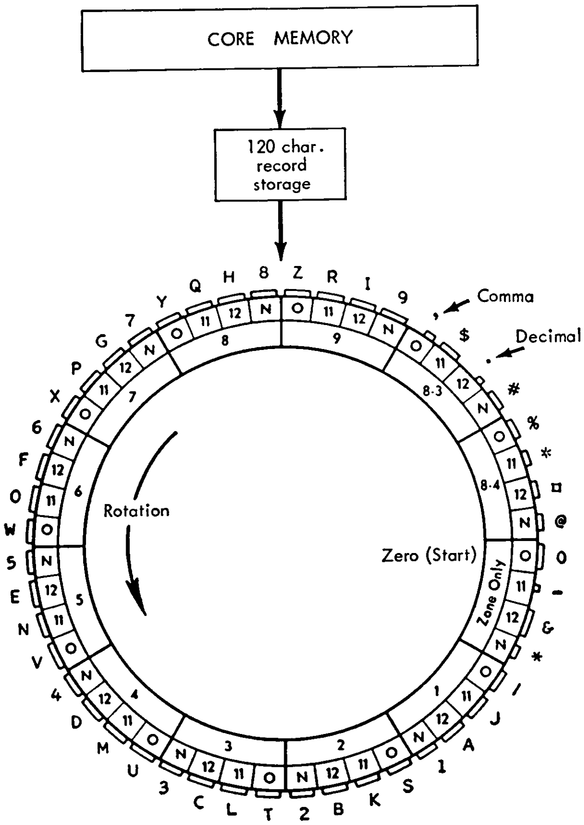


Figure 114. Schematic, Type Wheel

to each storage position of RSU. Any arrangement of data to fit report forms must be made in memory before the writing operation is performed.

When records, 120 characters or less in length, have been placed in RSU and the group mark in memory has been sensed, the CPU is free to continue with other operations while the record in RSU is printed. Records longer than 120 characters in length are printed on successive lines. In this case, CPU is not free for other operations until the last group of characters has been entered into record storage and the group mark sensed.

Read Check

Each character received from memory is checked for an illegal character structure and redundant character code. Any illegal character or redundant character indicates an error and is entered into RSU as "no bits." The read check indicator (printer and printer control operator's panels) is turned on after the complete record has been entered into RSU, but before the information is printed (Figure 115).

A read check error also turns on the read-write check indicator (00902) located on the 705 console. If the indicator is interrogated by a transfer-on-signal instruction immediately following the write instruction, a SUP instruction can suppress printing. Record storage can be reloaded for a second try at printing. Printing can be prevented until record storage is correctly loaded. If a character code error exists in the data being transmitted from memory to record storage, the machine check indicator (00901) is also turned on.

Print Check

As the converted record is entered into RSU, an odd or even indication of the number of bits in each of the horizontal numerical bit rows of the record is developed. During printing, an odd or even indication of the number of numerical bits in each row of the printed record is determined by sensing (echo impulses) the position of each print wheel during the print cycle. The odd or even indications of the record entered into RSU are compared with the odd or even indications of the printed record (Figure 115). If they do not compare a print check error is indicated by the print check indicator on the printer and printer control operator's panels. A print check error also turns on the record check indicator 00903 of CPU during the execution of the next write instruction involving the printer.

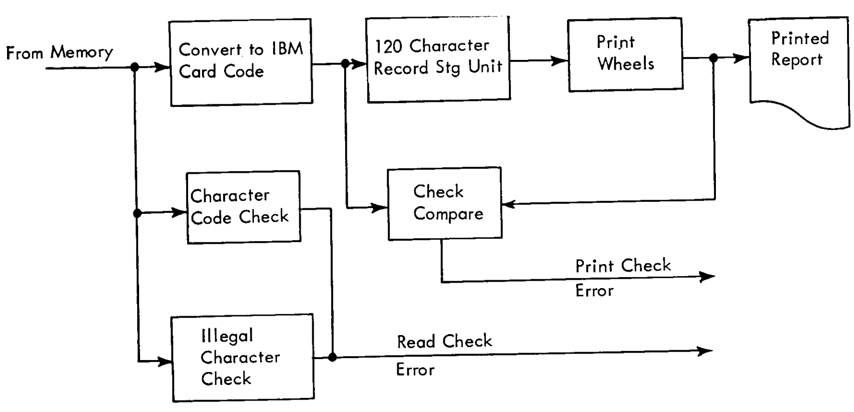


Figure 115. Schematic, Printer Checking

Operating Procedures

Start. Before the printer can be used under programmed control, it must be in ready condition.

1. Prepare the carriage for operation. (See "IBM 922 Tape-Controlled Carriage," in the "Machine Components" section.

2. Depress the start key. This causes the printer to go through a clearing cycle. The input-output indicator is turned off (if on) during this cycle.

3. Depress the start key a second time. This establishes a ready condition and turns on the ready light. The printer is now available for operation under program control.

If the carriage runs out of paper, the ready light is turned off and the form stop light is turned on. A subsequent write or write erase instruction to the printer results in a no-response indication (705 console) and CPU operation is suspended until ready condition is restored or another operation is initiated. (See "Form Control Key" under "IBM 717 Operating Keys and Indicator Lights.")

Error Correction

READ CHECK. The 00902 read-write check indicator of CPU is turned on as a result of a read check error. The indicator is turned on before the record containing the error is printed. If the indicator is interrogated by a transfer on signal instruction immediately following the write instruction, a SUP instruction can suppress printing. Record storage can be reloaded for a second try at printing. Printing can be prevented until record storage is correctly loaded. If the character code error exists in the data being transmitted from memory to record storage, the machine check indicator (00901) is also turned on.

PRINTER CHECK. The 00903 record check indicator of CPU is turned on as a result of a print check error. The record check indicator is turned on during the execution of the next write instruction directed to the printer. If the program switch has been set to program control, any action taken must be written in the program. If the 00903 switch has been set to AUTOMATIC, CPU stops following the execution of the write instruction in which the indicator is turned on.

IBM 717 Operating Keys and Indicator Lights

The operating keys and indicator lights on the operator's panel of the printer are shown in Figure 116.

Start Key. Depressing the start key once causes the printer to go through a clearing cycle. The input-output indicator is turned off during this cycle. Depressing this key again turns on the ready light and makes the printer available for operation under 705 control. When the ready light is on, the start key is no longer operative.

Stop Key. Depression of this key causes the ready light to be turned off and the printer to stop operating.

Form Control Key. Reaching the end of paper in the printer closes a form stop contact and provides one of two modes of operation, depending on whether or not a carriage tape is being used with the operation. When a tape is not being used, the printer stops when the form stop contact closes and the form stop light goes on. Depressing the form control key allows the printer to operate for one more print cycle. As long as the form control key is held down, the printer remains available and continues to print.

When a tape is used in the carriage, channel 1 must be punched as the first printing line. The printer does not stop when the form stop contact closes. After the form stop contact has closed and a channel 1 skip is sensed, the printer stops before the next form is printed and the form stop light goes on. This indicates that the carriage has moved from one form to the next. The operator now has two options:

1. Manually, to feed new forms of paper into the carriage, thereby opening the form stop contact. (In doing this, depress the form control key to turn off the form stop light and allow the printer to continue.)
2. To depress the form control key and thereby turn off the form stop light and allow the printer to continue until the closed form contact and the next channel 1 skip again stops the printer.

This arrangement provides for the printing of one or several forms before finally stopping the printer and inserting additional paper forms.

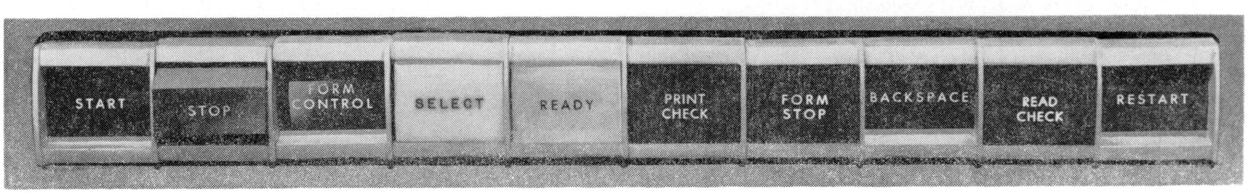


Figure 116. IBM 717 Operating Keys and Lights

Select Light. This light goes on when the printer is selected by the select instruction and remains on until another input-output unit, check indicator, or alteration switch is selected.

Ready Light. The ready light indicates that the printer is ready for operation under control of the 705.

Print Check Light. A print check error turns on the print check light.

Form Stop Light. This light indicates that the printer is running out of paper.

IBM 757 Printer Control

The control unit contains the necessary decoding, checking, and timing circuits for the transfer of data from the CPU to the printer. The control unit is attached to the CPU by the I-O (input-output) cable. The control unit also supplies the necessary electrical power for the operation of the printer control and the attached printer.

Each printer has a power and signal cable which must be attached to the control unit; the signal cable carries the data and coordinating signals. The power cable carries the electrical power required.

The operating keys and indicator lights on the operator's panel of the printer control are shown in Figure 117.

CPU-Tape Switch. The two positions are CPU and TAPE. In all operations with CPU, this switch must be set to CPU (the up position).

Power-On Key. When the power-on key is depressed, the various power supplies in the control unit are made operative in a sequential process that takes approximately four minutes to complete.

Power-On Light. The power-on light is on whenever the power-on key has been depressed and stays on as long as power is on.

DC-On Light. The DC (direct current) on light automatically lights after the power-on key is depressed and the power-on sequence is completed. The light remains on until a power-off condition is present.

Power-Off Key. Depression of the power-off key removes all electrical power from the control unit and the printer.

Fuse Light. This light, when on, indicates that a fuse (protective device) has burned out. This condition forces a power-off condition. The customer engineer should be notified if this light is on.

Thermal Light. This light, when on, indicates that the operating temperature of the machine is too high. This condition forces a power-off condition. The customer engineer should be notified if this light is on.

The printer check and read check indicators, backspace, and restart keys are similar in function to those of the 717 operator's panel.

IBM 922 Tape-Controlled Carriage

The carriage used with the 717 Printer is a 922 Tape-Controlled Carriage. The tape-controlled carriage controls the feeding and spacing of forms at high speed while documents or reports are being prepared on the printer (Figures 118, 119). The carriage accommodates continuous forms to maximum sheet dimensions of 22 inches in length and 16¾ inches in width, including punched margins. Spacing and skipping for form control are performed under control of the carriage control switch and carriage control tape.

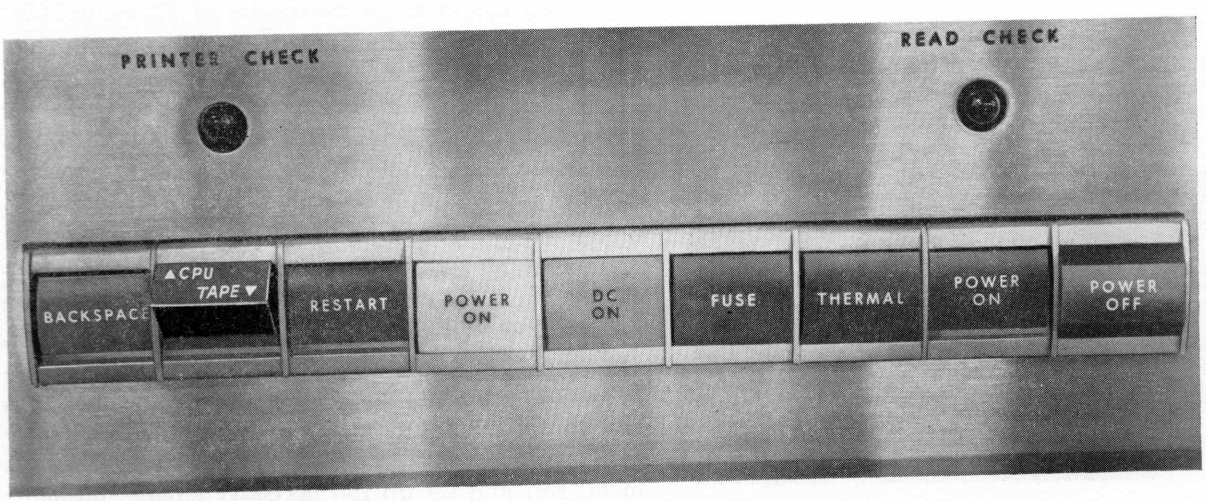


Figure 117. IBM 757 Operator's Panel

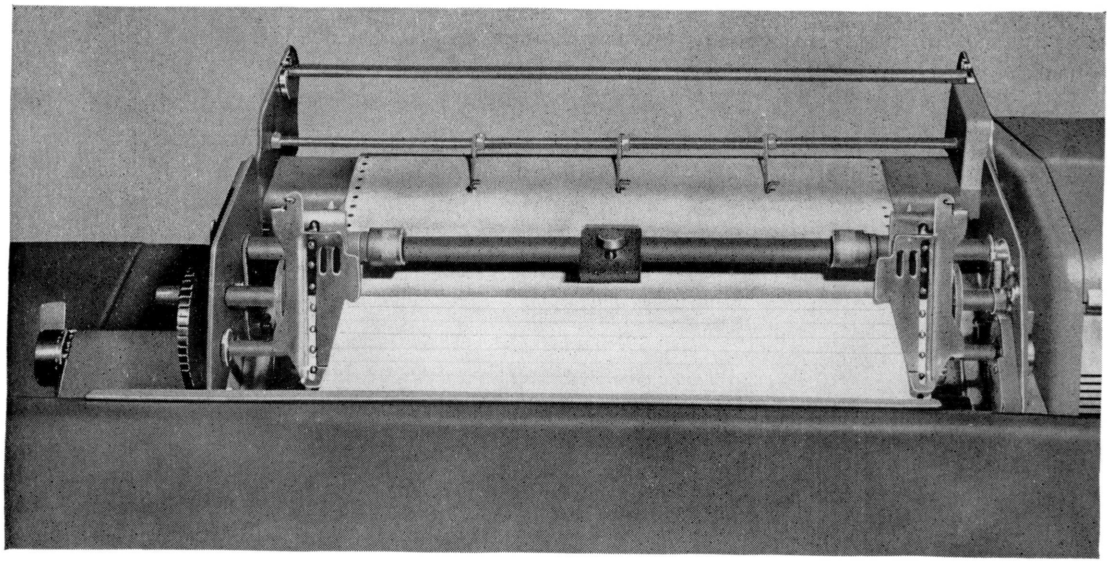


Figure 118. Form Feeding Carriage

Carriage Control Switch

The carriage control switch has three positions: SINGLE space, DOUBLE space and PROGRAM (Figure 123). In the SINGLE or DOUBLE (space) position, the spacing of the form is automatic and will be single or double depending upon the setting of the switch. When the switch is set to SINGLE, form spacing is six lines to the inch. When the switch is set to DOUBLE, spacing is three lines to the inch. Under either setting, print wheel 1 prints the first character in the record, print wheel 2 prints the second character, and so on.

With the switch set to PROGRAM, the spacing and skipping operations are governed by the first character of each memory record and are not automatic. The first character of each memory record is the control character and is sent through the carriage control circuits to signal operations instead of going into RSU. The carriage control character is not printed. Print wheel 1 prints the second character of the record and successive characters are printed by print wheels 2, 3, and so on.

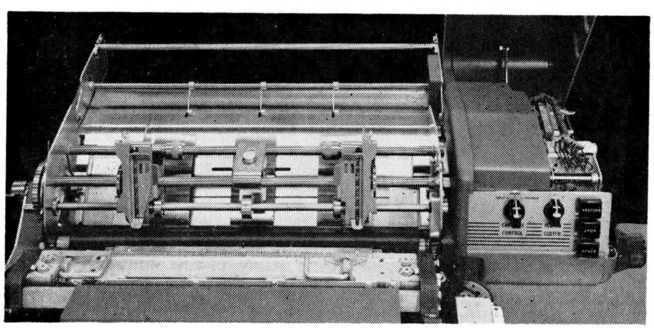


Figure 119. IBM 922 Tape-Controlled Carriage

The following characters are used to control skipping and spacing:

Suppress space	& (ampersand)
Single space	b (blank)
Double space	0
Skip to channel 1-9	1-9
Short skip to channels 1-9	J-R

Control Tape

The carriage is controlled by punched holes in a narrow paper tape, exactly the length of one or more forms to be used. When the tape is installed in the carriage and the carriage is in motion, the control tape is advanced in synchronous movement with the paper forms in the carriage (Figure 120).

The control tape (Figure 120) has twelve columnar positions indicated by vertical lines. The positions are called channels. A maximum of 22 inches (132 lines) can be used for control of a form, although, for convenience, the tape blanks are slightly longer. Horizontal lines are spaced six to the inch for the entire length of the tape representing possible printing lines. Round holes in the center of the tape are prepunched for a pin feed drive in the tape sensing mechanism (Figure 122).

Twelve brushes, one for each channel, are positioned over the tape for sensing the holes that are punched. As viewed from the front of the machine, they are numbered 1 through 12 from left to right. Brush 1 rests on channel 1, brush 2 on channel 2, and so on. A hole punched in a channel of the control tape allows the brush to make contact with a metal roll and set up the necessary circuits normally used to stop skipping or to initiate an overflow.

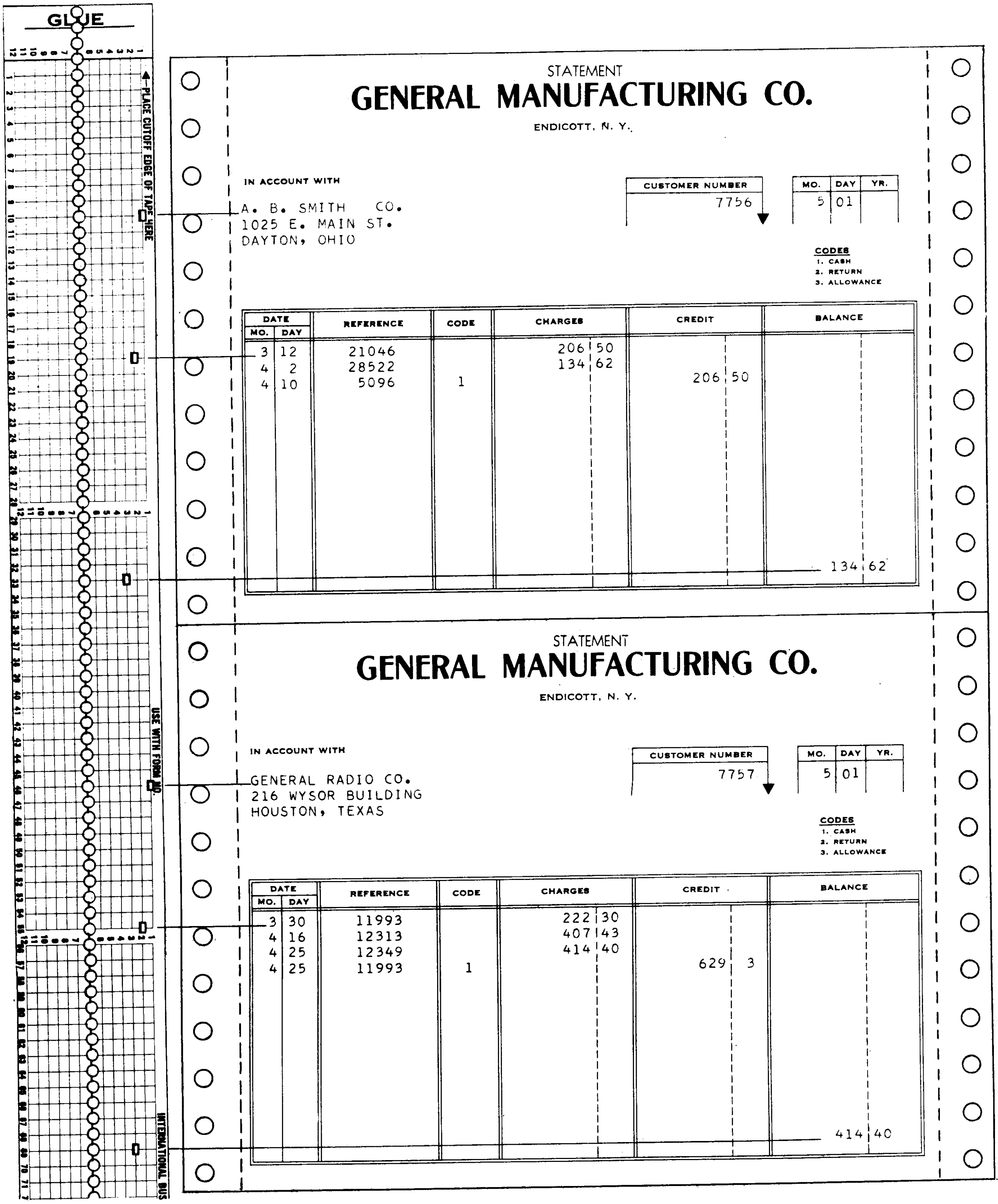


Figure 120. Control Tape (Carriage Control Switch Set to PROGRAM)

With the carriage control switch set to SINGLE or DOUBLE (space), spacing is dependent upon the setting of the carriage control switch and is automatic. The overflow or ejection of the form from the last printing line of one form to the first printing line of the following form is controlled by punches in the carriage control tape. The following control tape punches are effective:

1. Channel 1 is punched to indicate the first printing line of a form. The channel 1 punch is sensed after an overflow and stops the form at the first printing line.
2. Channel 12 is punched to eject or overflow the form to channel 1. Channel 12 must be punched in a control tape position corresponding to the next-to-the-last printing line of a form.

With the carriage control switch set to PROGRAM, single or double spacing is not automatic and is controlled by the carriage control character. Spacing can vary between lines. For example, the heading section of a form may be single spaced and the body double spaced. Skipping up to two inches (12 line spaces) between successive print lines can be accomplished while maintaining the normal printing speed of 150 lines a minutes. When using carriage control characters indicating skips of two inches or less, the carriage control tape skip distance must be two inches or less or printing in flight will occur. Skipping lengths greater than two inches decreases the normal printing speed by causing print delay cycles. The following tape channel punches are effective:

1. Channels 1-9 (skip stops) are punched to cause a skip stop. Skipping is started by the control character of the record and stopped by an appropriate punch in channels 1-9. Skipping in any sequence can be controlled to eight different sections of the form including the first printing line.
2. A channel 10 punch is used to change a long skip (greater than two inches) to a short skip with a possible saving of one print delay cycle. The channel 10 punch is punched two inches (12 line spaces) before the following (indicated) print line.
3. A channel 12 punch turns on the printer input-output indicator which can be interrogated by the transfer-on-signal instruction. The indicator will not be turned on when the carriage is in automatic control (S or D) and a 12 hole in the tape is sensed. The resulting transfer is normally made to a subroutine where the input-output indicator is turned off. The indicator can be turned off by:
   1. Using the IOF instruction.
   2. Depressing the printer start key.

Control Tape Punching

A small, compact tape punch (Figure 121) is provided for punching the tape. Punch the tape to conform to printing line positions on the form; then cut the tape to the length of the form to be used, or, in the case of an extremely short form (such as a payroll statement), allow the tape to represent more than one form. After punching, glue the tape into a loop to provide for a repetitive operation.

When marking the tape, first mark the channels in which the holes are to be punched. The easiest way to do this is to lay the tape beside the left edge of the paper form to be used, with the top line (immediately under the glue portion) even with the top edge of the form. Make a mark in the first channel on the line corresponding to the first printing line of the form. Make additional marks in the appropriate channels for each of the other skip stops and for the overflow signal required for the form.

Repeat the marking for one form as many times as the usable length of the tape (22 inches) allows. Having the tape control several forms in one revolution (through the sensing mechanism) increases the life of the tape. Finally, mark the line corresponding to the bottom edge of the last form for cutting after the tape is punched.

To insert the tape into the punch: (1) place the line to be punched over a guide line on the base of the punch, (2) place the center feed holes of the tape over the pins projecting from the base, (3) turn the dial until the arrow points to the number of the channel to be punched, and (4) press the top of the punch, toward the back, to cut a rectangular hole in the intersection of a vertical and horizontal line in the required channel of the tape.

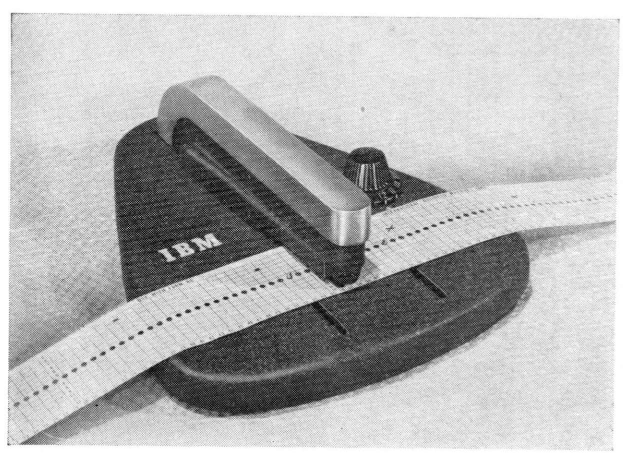


Figure 121. Carriage Tape Punch

The tape may be punched with holes in more than one channel on the same line. This is a help whenever several skip impulses are directed to the same skip stop.

After the tape is punched, cut it and loop it into a belt. Glue the bottom line to the top line by using the section marked "glue" but first remove the glaze from the glue section with an ink eraser. If the glaze is not removed, the tape ends may come apart. The center feed holes should coincide when the two ends of the tape are glued together.

Do not punch the last hole in the tape closer than four lines from the cut edge, as about the last half inch of the tape overlaps the glue section when the two ends are spliced. If it is necessary to punch a hole lower than four lines from the bottom of the form, place the tape with the top line (immediately under the glue portion) four lines lower than the top edge of the form before marking the channels. To compensate for the loss, cut the tape four lines lower than the bottom edge of the form.

Inserting Tape in Carriage

Tilt back the cover of the carriage to gain access to the tape reading mechanism. Turn the platen clutch to a disengaged position, and raise the brushes, after releasing the brush holder frame latch on the side of the brush holder (Figure 122). With the tape held so that the printed captions can be read, place one end of the loop over the pin-feed drive wheel so that the pins engage the center drive holes. Place the opposite end of the loop over the nearest half-circle guide piece. Remove the excess slack from the tape by lifting the adjusting lever away from the notched bar and by moving the guide piece unit to the right. The tape should be just tight enough to give slightly when the top and bottom portions of the loop are pressed together as shown in Figure 122. If it fits too tightly, the pin feed holes will be damaged.

After the tape is in position, press down the brushes and close the cover. Depress the restore key to bring the tape to its home position and turn the platen clutch back to its engaged position. The carriage is then ready to operate.

Tapes can be changed readily and used repeatedly over a considerable period of time.

Carriage Operating Features

Platen Clutch. When the arrow on the platen clutch is pointed upward, as shown in Figure 123, the platen is engaged and can be turned manually only by the vernier knob. To disengage the platen from the machine control, turn the platen clutch to the right;

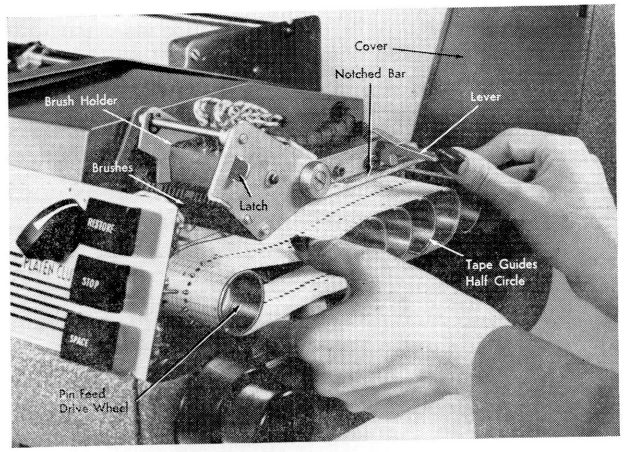


Figure 122. Inserting Tape in Carriage

then it is possible to turn the platen manually, using the platen knob.

Restore Key. Use this key to restore the carriage tape to channel 1 or to the home position. If the platen clutch is engaged, the platen moves with the carriage; if the platen is disengaged, it does not move with the carriage. After the carriage is restored to channel 1, the forms should be aligned to the first print line and the platen clutch engaged.

Stop Key. Depress this key to stop the printer operation. Also, use it to stop undesired carriage skipping.

Space Key. When the printer is stopped, the form can be advanced one or two spaces at a time depending upon the setting of the carriage control switch. With the carriage control switch set to SINGLE, spacing is one line at a time. With the carriage control switch set to DOUBLE or PROGRAM, spacing is two lines at a time. To feed the first form into position, depress the space key if the platen clutch is engaged; disengage

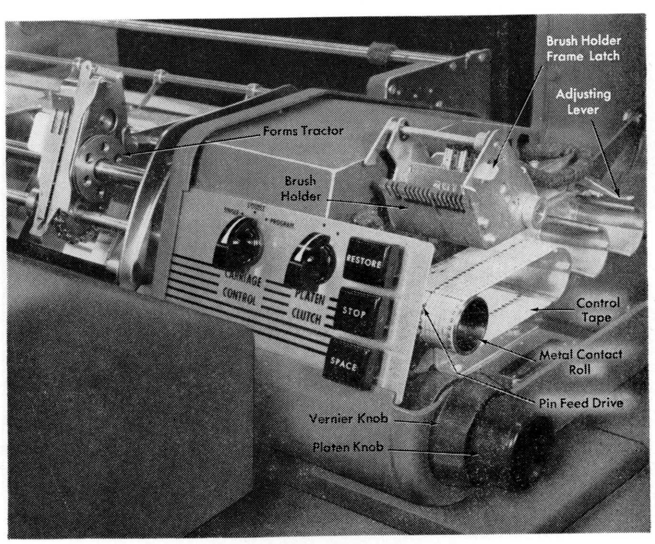


Figure 123. Carriage Operating Features

the platen clutch to permit restoring the tape without advancing the form.

Platen Knob. Use the vernier knob to obtain exact registrations in relation to the horizontal lines of the forms. Turning the knob counterclockwise advances the platen, thus lowering the position of the printed line on the form. Turning the knob in a clockwise direction causes the position of the printed line to be higher on the form. In either case, the carriage tape is not affected and adjustments can be made while the platen is engaged or while the printer is in operation.

Form Thickness Adjustment Device. Use this device to adjust the distance between the print wheels and the platen, for varying thicknesses of paper stock or for varying numbers of copies. This device is located on the left side of the carriage (Figure 124). It contains seven notches numbered from 0 through 6. When the dial is in the 0 notch, the print wheels are one-eighth inch from the platen. Each of the remaining six notches adds to the distance between the print wheels and the platen. When the dial is set to 6, the distance is increased to about one-sixth inch. The dial should be set wherever the best results are obtained. To adjust for varying thicknesses: (1) pull out the dial lock, (2) turn the dial counterclockwise to increase the distance between the platen and the print wheels, or turn the dial clockwise to decrease this distance.

Pressure Release Lever. Push back this lever to release the feed rolls so that the paper can be moved freely around the platen. Always release pressure when the above-platen feed (forms tractor) is being used. Apply pressure when the above-platen feed is not in use.

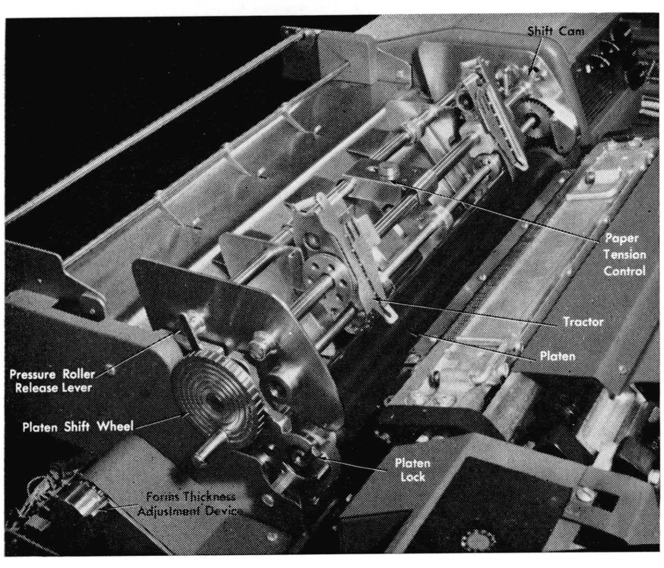


Figure 124. Left Side of Carriage

Platen Shift Wheel. The platen may be shifted laterally to a total of four inches by turning the platen shift wheel. For example, with the carriage in the extreme left position, the carriage may be moved four inches to the right. Do not make this adjustment while the machine is in operation (Figure 124).

End-of-Form Stop. The end-of-form stop, located in the center of the carriage, stops the machine when the carriage runs out of paper. The forms feed under the form stop; when the bottom edge of the last form passes the form stop, the machine stops. The distance between the end-of-form stop and the printing line is about 13⅜ inches.

Platen. The carriage is equipped with an easily removable solid platen. Remove platen by raising the platen lock on the left side and lifting the platen from the bearing housing. To insert the platen, place the gear-wheel end of the platen in the slot on the right end of the carriage; drop the left end into the platen bearing housing; then move the platen to the right, turning it back and forth to fit the platen drive key into the carriage drive mechanism; finally, close the platen lock.

Forms Tractors

Two IBM forms tractors are available for the printer. The F2 is standard but the F4 may be specified in place of, or in addition to, the F2. Each of these devices is used for feeding marginally punched continuous forms and each has two adjustable tractor-type pin-feed units, one for each side of the form.

The F2 provides the choice of spacing either six or eight lines to the inch; the F4, either four or six lines to the inch. Make this adjujstment on the F2 by moving the shift cam until its pointer is positioned between the two scribed lines at either the 6 or 8 on the side frame (Figure 125). Adjust the F4 in the same way for spacing of four or six lines per inch. Variations that must be considered for spacing of four or eight lines per inch are illustrated in Figure 126.

Procedure for Using the Forms Tractor

1. After the forms tractor is in position, make sure that the platen and the forms tractor can be freely moved by hand.

2. If a narrow form is to be used, remove the center paper guide supporting the form in the center.

3. Move the left lower paper guide and tractor slightly to the left of the first printing position. Place the first form between the left and right lower paper guides and move the right guide in against the edge of the form. Allow a slight clearance so that the form

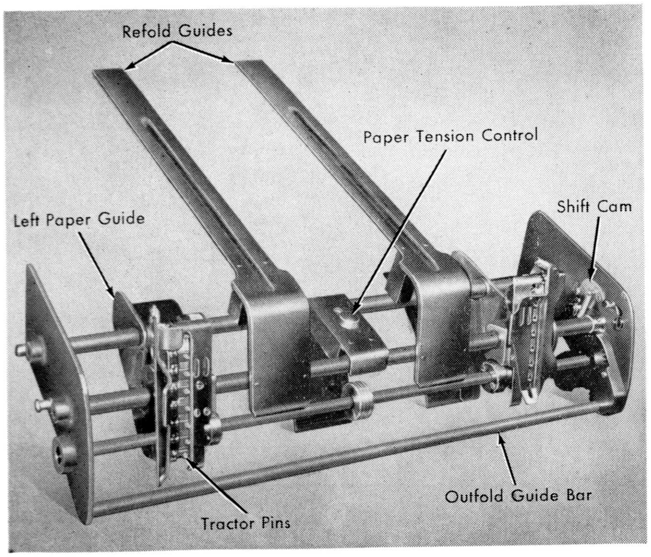


Figure 125. IBM Forms Tractor

slides freely between both guides. Tighten both lock assemblies to hold the guides in place.

4. With the pressure rolls engaged, insert the form under the carriage between the round rod and the platen, and then into the pressure rolls. Turn the platen by hand (platen clutch disengaged) until the end of the form can be grasped.

5. Raise the pressure plates away from the pins.

6. Release the pressure roll, draw up the form, and attach the pin feed holes to the tractor pins.

7. Lower the pressure plates.

8. Set the form so that the first printing line is even with the first printing line indicator mark on the lower part of the pressure plates.

Tractor Adjustments. The tractor adjustment wheels may be turned to provide a one-eighth inch lateral movement of the tractors. The wheels are used to make the chain pins line up exactly with the center of the marginal holes in the paper after the paper guides have been set.

IBM 734 Magnetic Drum Storage

Magnetic drum storage provides additional storage capacity for the 705 system (Figure 127). Drum storage consists of one or more 734 Magnetic Drum Storage units, each unit with a storage capacity of 60,000 characters. Information can be removed from memory and placed on the drum or removed from the drum and placed in memory. Recording is permanent and may be retained after the power is turned off. Each drum is a metal cylinder coated with a material that can be easily magnetized (Figure 128). Information is written on the cylindrical surface in the form of magnetized spots while the drum is rotating at high speed. This writing is placed along a number of parallel channels located side by side across the width of the drum. The drum is divided into 300 addressable sections and each section can store up to 200 characters.

The address part of the SEL instruction specifies the drum and the drum section to be used. Assuming that a 705 system has several drums, the following system would exist: Addresses 01000 to 01299 would select the sections on the first drum; 01300 to 01599 would select sections on the second drum; 01600 to 01899, the third drum, and so on. The thousands and hun-

	6 LINES PER INCH		8 LINES PER INCH		4 LINES PER INCH	
	F-2 and F-4		F-2		F-4	
	Carr. Tape Lines	Distance on Form	Carr. Tape Lines	Distance on Form	Carr. Tape Lines	Distance on Form
Maximum Length of Form	132 Lines	22 Inches	132 Lines	16 1/2 Inches	132 Lines	33 Inches
Length of Form Compared with Length of Tape	———	Same	———	Form is 3/4 as long as one tape revolution	———	Form is 1 1/2 times as long as one tape revolution
Short Skip	12 Lines Maximum	2 Inches Maximum	12 Lines Maximum	1 1/2 Inches Maximum	12 Lines Maximum	3 Inches Maximum
Distance to Move First Printing Line Back from Indicator to Print	14 Lines	2 1/3 Inches	19 Lines	Approximately 2 1/3 Inches *	9 Lines	Approximately 2 1/3 Inches *

* Vernier adjustment may be necessary

Figure 126. Line Space Variations

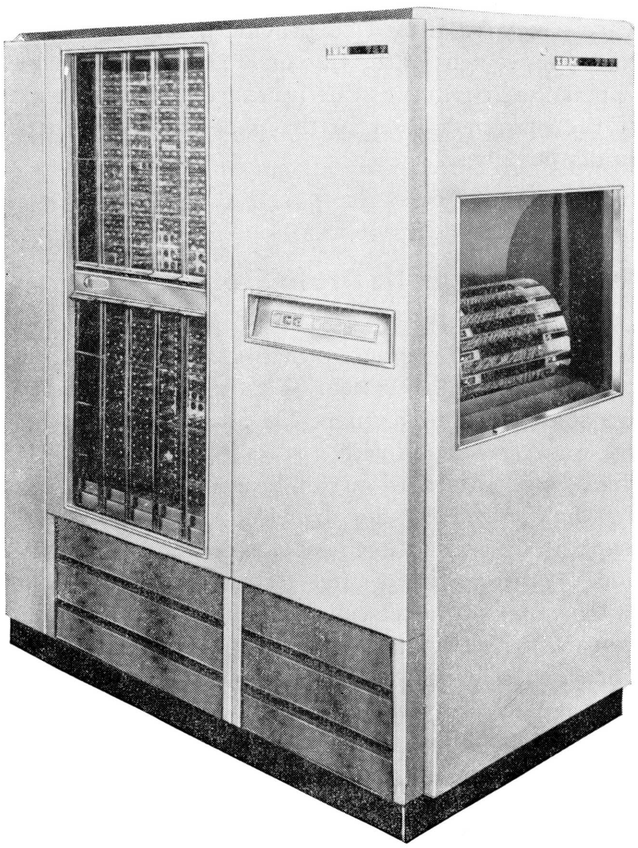


Figure 127. IBM 734 Magnetic Drum Storage

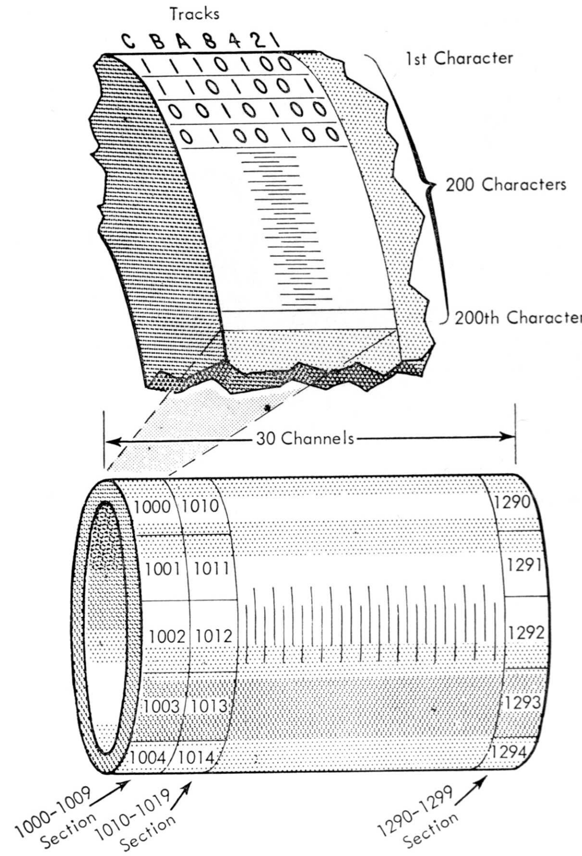


Figure 128. Schematic, Magnetic Drum Storage

dreds positions of the select address indicate the drum. The hundreds, tens and units positions indicate the section. For example, the select address 01245 selects drum 1 and section 245.

The average time required to locate the first character position of a drum section for a read or write operation is eight milliseconds. This is termed access time. Thereafter, characters can be read or written consecutively on the drum at a rate of one character every 40 microseconds.

Reading or writing can pass from one drum section to another until a drum or group mark is sensed. An attempt to read or write beyond the limit of the drum turns on the read-write check indicator and also the drum input-output indicator used especially to detect this error. The input-output indicator is turned off by a subsequent read, write or write-and-erase instruction for the drum, or it may be turned off by an IOF instruction. A select and transfer-on-signal instruction turns off the read-write indicator.

Writing on the Drum

During a write or write-and-erase operation, information is written from memory, starting at the address specified by the instruction and continuing to higher memory addresses until a group mark is sensed. Sensing the group mark terminates the operation and causes a drum mark to be emitted onto the drum after the last written character.

The characters from memory are written on the drum starting in the first position of the drum section specified by the address part of the select instruction. The following positions of the section will be filled sequentially until the group mark in memory is sensed. If the record to be written is greater than 200 characters in length, writing will automatically continue into the next higher section of the drum.

Writing over a section erases any information previously written. An error detected in writing information on the drum is indicated by a 00902 read-write check indication.

Reading from the Drum

During a read operation, information read from the drum starts with the first character of the section previously selected. This character is read into the memory location specified by the address part of the read instruction. Reading continues from successively

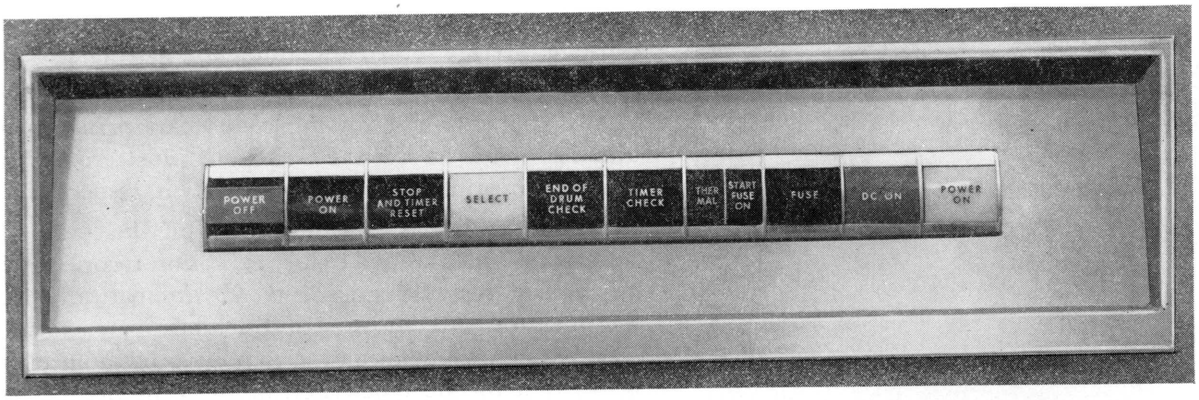


Figure 129, IBM 734 Operator's Panel

higher drum locations into successively higher memory addresses until the drum mark is sensed.

An error detected in reading information from the drum is indicated by a 00902 read-write check indication.

IBM 734 Operating Keys and Indicator Lights

The operating keys and indicator lights on the operator's panel of the 734 drum are shown in Figure 129.

Power-On Key. Use of this key sequentially activates the various power supplies and starts the magnetic drum rotating. Approximately five minutes are required before the correct DC voltages are established and the drum has attained maximum speed.

Power-On Light. The power-on light is on whenever the power-on key has been depressed and stays on as long as power is on.

DC-On Light. The DC (direct current) on light automatically lights after the power-on key is depressed and the power-on sequence is completed. The light remains on until a power-off condition is present.

Power-Off Key. Depression of the power-off key removes all electrical power from the drum.

Thermal Light. This light, when on, indicates that the operating temperature of the machine is too high. Normal operation is not possible with this light on. The customer engineer should be notified if this light is on.

Start Fuse On Light. The start fuse on light is on while the drum is attaining rotating speed. If the light remains on, notify a customer engineer.

Fuse Light. This light, when on, indicates that a fuse (protective device) has burned out. Normal operation is not possible with this light on. The customer engineer should be notified if this light is on.

Select Light. This light goes on when the drum is selected by the select instruction and remains on until another component is selected.

Timer Check Light. An internal timing error within the machine automatically turns on the timer check light. Normal operation is not possible with this light on. Use the stop and timer reset key to turn off the timer check light.

End-of-Drum Check Light. This light is turned on when a read or write operation is allowed to go beyond the end of the drum. The light may be turned off by use of the timer and reset key and the control instruction.

Stop and Timer Reset Key. Use of this key resets the machine to manual control and resets all error indications. Normal start procedures are necessary to begin operation after the use of this key.

IBM Typewriter

A typewriter is supplied for the 705 console (Figure 130). It can be used to print a portion of memory, one character at a time. The speed of typing is approximately 600 characters per minute. The typewriter has no record storage unit and prints directly from memory. All other operations of the machine are held up during the use of the typewriter, with the exception of previously initiated DS operations.

After the typewriter is selected and a write or write erase instruction is given, information is typed starting at the addressed memory position and continuing to sequentially higher memory positions, until a group mark is sensed. The group mark prints as a ⌑ when the write 01 instruction is used.

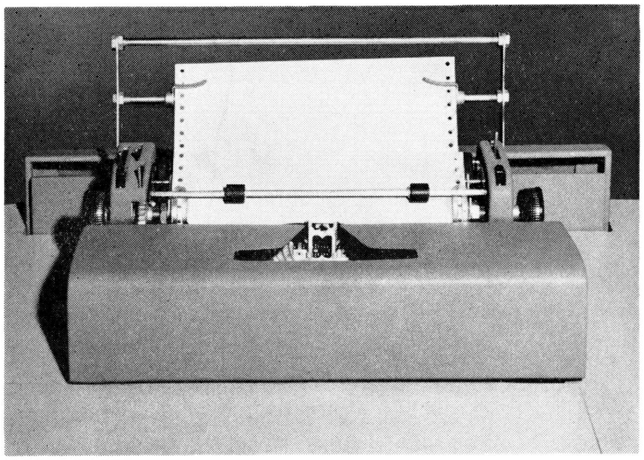


Figure 130. Typewriter

Sensing the right-hand margin or the end of record causes a carriage return and automatic spacing according to the setting of the space control on the carriage.

Any character not on the code chart prints as a question mark on the typewriter. A plus zero and a minus zero print as a plus sign and dash, respectively.

Special conditions encountered during the course of a program (such as end-of-file, exception records, and unmatched records) can be typed automatically for the operator. Accounting control totals, batch totals, or other control information may be programmed to be written by the typewriter, or the contents of any portion of memory may be examined by manually selecting the typewriter and using the write instruction in the usual way.

Appendix A

List of Abbreviations

ACC	accumulator
ASU	auxiliary storage unit
AUX	auxiliary
char	character
CPU	central processing unit
CR	character register
DS	data synchronizer
I	instruction
IB	input buffer
IC	instruction counter
I-O	input-output
LRC	longitudinal redundancy check
LRCR	longitudinal redundancy check register
MAC	memory address counter
MAR	memory address register
MASR	memory address select register
MBR	memory buffer register
OB	output buffer
OP	operation
PCT	parallel character transmission
RC	record counter
REG	register
SAC	storage address counter
SASR	storage address select register
SBR	storage buffer register
SMAC	synchronizer memory address counter
SPC	starting point counter
SSR	storage select register
TC	tape control
TDS	tape data selector
TRC	tape record coordinator
μs	microsecond
VRC	vertical redundancy check

Appendix B

Execution Times

Formulas for computing the times required to execute the instructions on the 705 are presented below.

For the 705 I and II, the formulas represent minimum time; for the 705 III, they are weighted for certain variables, e.g., carries and rounding.

All timings are shown in microseconds. One second = 1,000 milliseconds = 1,000,000 microseconds; one millisecond = 1,000 microseconds.

The symbols used in the formulas are defined as follows:

N = Number of characters
N_s = Number of characters in storage field
N_m = Number of characters in memory field
$|N_s|$ = Absolute value of quantity in storage field (magnitude without regard to sign)
$|N_m|$ = Absolute value of quantity in memory field (magnitude without regard to sign)
N_a = Address of instruction
N_p = The number of periods or commas skipped plus the number of zeros or commas replaced by blanks (used only in the SPR instruction)
N_h = Shorter of the two fields.

IBM 705 III Execution Times / IBM 705 I and II Execution Times

		IBM 705 III		IBM 705 I and II		
INSTRUCTION	CODE	TIME (μs)	REMARKS	TIME (μs)	REMARKS	PAGE
Add	ADD	$40 + 9.3N_m$*	Mem. and stor. signs alike	$34 + 17N_m$	All cases except that below:	40
			Mem. and stor. signs unlike	$51 + 34N_m$	Signs unlike and $\|N_m\| < \|N_s\|$	
		$40 + 9.3N_m$	$\|N_m\| \geqq \|N_s\|$			
		$67 + 18.5N_m$	$\|N_m\| < \|N_s\|$			
Add Addr to Mem	AAM	85	———			43
Add to Mem	ADM	$40 + 10.8N_h$	Signed mem. field Mem. and stor. signs alike	$34 + 17N$	All cases except that below:	41
		$40 + 10.8N_h$	Mem. and stor. signs opposite except when	$51 + 34N$	Signed field, signs unlike $\|N_m\| < \|N_s\|$	
		$67 + 21.6N_m$	$N_s \geqq N_m$ *and* $\|N_s\| > \|N_m\|$			
		$49 + 10.8N_s$	Unsigned mem. field			
Blank Mem	BLM 00	$31 + \left(\frac{N_m}{5} \times 9\right)$	Stor. sel. reg. = 0 (ACC)			34
	BLM 01	$31 + 9N_m$	Stor. sel. reg. ≠ 0 (ASU)			34
Compare	CMP	$31 + 9N_s$		$34 + 17N_s$		52
Ctrl 00000	IOF	49		68		60
Ctrl 00001	WTM	49		10000		61
Ctrl 00002	RWD	33000		33000		61
Ctrl 00003	ION	49		68		61
Ctrl 00004	BSP	49		$60000 + 67N$		61
Ctrl 00004 (01)	BSF	49				61
Ctrl 00005	SUP	6000		6000		61
Ctrl 00009	SKP	49				61
Divide	DIV	$90 + 9N_s + (N_s - N_m)(6.7N_m + 37)\,9$		$17[11 + N_s + (N_s - N_m)(7.5N_m + 15)]$	Average formula	46
Lengthen	LNG	$31 + 9N_a$		$51 + 17N_a$		51
Load	LOD	$31 + 9.3N_s$		$34 + 17N_s$		35
Load address	LDA	76				36
Multiply	MPY	$58 + N_s(63 + 9.3N_m)$		$17[N_s(N_m + 4) + 2]$		45
No operation	NOP	22		34		60
Norm and Tr	NTR	$58 + 9N_s$		$51 + 17N_s$ $68 + 17N_s$	Not normalized Normalized	53
Read	RD 00	$58 + 9\frac{N}{5}$**	From tape to mem. through *DS*	$10000 + 67N$ $68 + 33.5N$	From tape to mem. Card reader to mem.	23
		$40 + 33.5N$	From card reader to mem.	$8000 + 40N$	Drum to mem.	
		$8000 + 40N$	From drum to mem.			
	RD 01	58	Stor. sel. reg. = 1			24
	RD 02	130	Stor. sel. reg. = 2			24

INSTRUCTION	CODE	TIME (μs)	REMARKS	TIME (μs)	REMARKS	PAGE
Read while Writing	RWW	22		34		26
Receive	RCV	22		34		30
Reset add	RAD	$40 + 9.3N_m$		$34 + 17N_m$		34
Reset Subt	RSU	$40 + 9.3N_m$		$34 + 17N_m$		35
Round	RND	$68 + 9N_a$		$85 + 17N_a$		51
Select	SEL	40		51		23
Send	SND	$31 + \left(\frac{N}{5} \times 18\right)$				33
Set left	SET	$40 + 9.3N_a$		$51 + 17N_a$		49
Set bit	SB	31				58
Sign	SGN	49		68		36
Shorten	SHR	$50 + 9N_a$		$68 + 17N_a$		50
Stop	HLT	22		34		60
Store	ST	$40 + 10.8N_s$		$34 + 17N_s$		37
Store for print	SPR	$40 + 10.8N_s + 9N_p$		$51 + 17N_s$		37
Subtract	SUB	$40 + 9.3N_m$*	Mem. and stor. signs opposite	$34 + 17N_m$	All cases except that below	43
			Mem. and stor. signs alike	$51 + 34N_m$	Signs alike and $\lvert N_m\rvert < \lvert N_s\rvert$	
		$40 + 9.3N_m$	$\lvert N_m\rvert \geqq \lvert N_s\rvert$			
		$67 + 18.5N_m$	$\lvert N_m\rvert < \lvert N_s\rvert$			
Transfer	TR 00	22		34		55
	TR 01	67				56
Transfer any	TRA 00	22		34		56
	TRA 01-06	22				57
Transfer equal	TRE	22		34		53
Transfer high	TRH	22		34		52
Transfer plus	TRP	22		34		55
Transfer on signal	TRS 00	22		34		57
	TRS 01	22		34		57
	TRS 02-15	22				58
Transfer zero	TRZ	22		34		55
Transfer zero bit	TZB	31				59
Transmit	TMT 00	$13 + \left(\frac{N_m}{5} \times 18\right)$	Stor. sel. reg. = 0 (ACC)	$17 + \left(\frac{N_m}{5} \times 18\right)$		30
	TMT 01-15	$31 + 18N_s$	Stor. sel. reg. ≠ 0 (ASU)	$34 + 18N_s$		31
Unload	UNL	$31 + 10.8N_s$		$34 + 17N_s$		38
Unload address	ULA	85				38
Write	WR 00	$94 + 9\frac{N}{5}$**	From mem. to tape through DS	$10000 + 67N$	Mem. to tape	24
		$49 + 33.5N$	From mem. to punch or printer	$68 + 33.5N$	Mem. to punch or printer	
		$8000 + 40N$	From mem. to drum	$8000 + 40N$	Mem. to drum	
	WR 01		Stor. sel. reg. = 1		(Same as above)	25
		$94 + 9\frac{N}{5}$	From mem. to tape through DS			
		$49 + 33.5N$	From mem. to punch or printer			
		$8000 + 40N$	From mem. to drum			
	WR 02	85	Stor. sel. reg. = 2			25
Write erase	WRE 00	Not to be used	From mem. to tape through DS	$10000 + 67N$	To tape	26
		$49 + 33.5N$	From mem. to punch or printer	$68 + 33.5N$	To punch or printer	
		$8000 + 40N$	From mem. to drum	$8000 + 40N$	To drum	
	WRE 01	(Same as above)			(Same as above)	26

*When addition is being done in the add and subtract instructions (i.e., add with like signs or subtract with unlike signs), the time to execute the instruction will be lengthened by 9.3 microseconds for each carry propagated beyond the length of the memory field. This is included in the time shown.

**In all operations including reading from card reader or drum and writing to card punch, printer or drum (except reading and writing through a DS), the instructions are executed in a single period of time. When reading or writing through a DS, individual transfers of groups of five characters requiring nine microseconds transfer time take place while processing. These may occur between instructions and may also interrupt the processing of an instruction. The time required for these transfers between memory and the DS is included in the formulas for reading and writing through a DS. The formulas for read and write through a DS give central processing unit time only.

Appendix C

IBM 705 Autocoder Mnemonic Codes

INSTRUCTION	705-I, II	705-III	CODE	PAGE
Add	ADD	ADD	G	40
Add Addr to Mem		AAM	@	43
Add to Memory	ADM	ADM	6	41
Blank Memory (BLM 00)		BLM	$	34
Blank Memory serial (BLM 01)		BLMS	$	34
Compare	CMP	CMP	4	52
Ctrl 00000 (Turn off I-O indicator)	IOF	IOF	3	60
Ctrl 00001 (Write tape mark)	WTM	WTM	3	60
Ctrl 00002 (Rewind)	RWD	RWD	3	60
Ctrl 00003 (Turn on I-O indicator)	ION	ION	3	60
Ctrl 00004 (Backspace)	BSP	BSP	3	60
Ctrl 00004 (01)		BSF	3	60
Ctrl 00005 (Suppress printing or punching)	SUP	SUP	3	60
Ctrl 00009 (Skip tape)		SKP	3	60
Divide	DIV	DIV	W	46
Lengthen	LNG	LNG	D	51
Load	LOD	LOD	8	35
Load address		LDA	#	36
Multiply	MPY	MPY	V	45
No Operation	NOP	NOP	A	60
Norm and Tr	NTR	NTR	X	53
Read 00	RD	RD	Y	23
Read 01 (Forward space)	RD	FSP	Y	24
Read 02 (Read memory address)		RMA	Y	24
Read while Writing	RWW	RWW	S	26
Receive	RCV	RCV	U	30
Reset Add	RAD	RAD	H	34
Reset Subtract	RSU	RSU	Q	35
Round	RND	RND	E	51
Select	SEL	SEL	2	23
Send		SND	/	33
Set Bit Alternate (SB 07)		SBA	%	58
Set Bit 1 (SB 09-14)		SBN	%	58
Set Bit Redundant (SB 08)		SBR	%	58
Set Bit 0 (SB 01-06)		SBZ	%	58
Set Left	SET	SET	B	49
Shorten	SHR	SHR	C	50
Sign	SGN	SGN	T	36
Stop	HLT	HLT	J	60
Store	ST	ST	F	37
Store for Print	SPR	SPR	5	37
Subtract	SUB	SUB	P	43
Transfer	TR	TR	I	55
Tr Sw A On (00911)		TAA	I	57
Tr Sw B On (00912)		TAB	I	57
Tr Sw C On (00913)		TAC	I	57
Tr Sw D On (00914)		TAD	I	57
Tr Sw E On (00915)		TAE	I	57
Tr Sw F On (00916)		TAF	I	57
Tr and Store Location Ctr (TR 01)		TSL	1	56
Tr Any	TRA	TRA	I	56
Tr Read-Write Check (TRS 12)		TRC	O	58
Tr Equal	TRE	TRE	L	53
Tr High	TRH	TRH	K	52
Tr Instr Check (TRS 10)		TIC	O	58
Tr Machine Check (TRS 11)		TMC	O	58
Tr O'flow Check (TRS 14)		TOC	O	58
Tr Plus	TRP	TRP	M	55
Tr Ready (TRS 01)		TRR	O	57
Tr Echo Check (TRS 13)		TEC	O	58
Tr Sign Check (TRS 15)		TSC	O	58
Tr Signal	TRS	TRS	O	57
Tr Sync Any (TRS 03)		TSA	O	58
Tr Transmission Check (TRS 02)		TTC	O	58
Tr Zero	TRZ	TRZ	N	55
Tr Zero Bit		TZB	.	59
Transmit	TMT	TMT	9	30
Transmit Serial (TMT 01-15)		TMTS	9	31
Unload	UNL	UNL	7	38
Unload Address		ULA	*	38
Write 00	WR	WR	R	24
Write 01 (Dump memory)	WR	DMP	R	25
Write 02 (Set record counter)		SRC	R	25
Write and Erase 00	WRE	WRE	Z	26
Write and Erase 01	WRE	WRE	Z	26

IBM 760 Operations

INSTRUCTION	705-I, II	705-III	CODE
Ctrl 00026 (Read or write tape, write on printer)	RWS	RWS	3
Ctrl 00027 (Read or write tape, early start)	RWT	RWT	3
Ctrl 00028 (Reset 760 counter)	RST	RST	3
Ctrl 00029 (Write on printer and magnetic tape)	PTW	PTW	3

IBM 777 Operations

INSTRUCTION	705-I, II	705-III	CODE
Ctrl 00015 (Prepare to read while writing)	PRW	PRW	3
Ctrl 00016 (Read tape to TRC)	RTS	RTS	3
Ctrl 00017 (Write TRC to tape)	WST	WST	3
Ctrl 00018 (Bypass TRC)	BPC	BPC	3

Appendix D

IBM 705 III AUTOCODER INSTRUCTION TRANSLATION

THE FOLLOWING is a key to codes and notes in the table below:

(A) Requires 4 or 9 address.

(B) Processed as macro-instruction in the interim system.

(C) The bit addressed is designated by a 1, 2, 4, 8, A, B, or C in the numerical column of the autocoder coding form.

(D) Accumulator or auxiliary storage unit.

(E) Any instruction which can accept a descriptive operand can take an indirect address.

(F) Autocoder III only.

N_a The address of the instruction.

N_m The number of characters in the memory field.

N_s The number of characters in the storage field.

N_t N_s plus the number of periods or commas skipped.

	Programmer Writes		Permissible Autocoder Entries							Actual Machine Code Produced				
Instruction	Mnemonic	Numerical	Blank	Actual	(E) Descriptive	Literal	(F) Location C'tr.	Character Addressed	Presently Known as	Operation	Stor. Unit	Addr.	Code	Comments
Add	ADD	SU(D)	Yes	Yes	Yes	Yes	Yes	Low Order		G	00-15			
Add Addr to Mem	AAM	SU	Yes	Yes	Yes	No	Yes	Low Order		@	00-15		A	
Add Mem	ADM	SU	Yes	Yes	Yes	Yes	Yes	Low Order		6	00-15			
Backspace	BSP		Yes	No	No	No	No	None	CTRL 4	3	00	00004		
Backspace File	BSF		Yes	No	No	No	No	None	BSP 01	3	01	00004	B	
Blank Mem	BLM		Yes	Yes	Yes	Yes	Yes	(F) Length/5		$	00			Assumes presetting of MAC II to 4 or 9.
Blank Mem Serial	BLMS		Yes	Yes	Yes	Yes	Yes	Length	BLM 01	$	01		B	Assumes presetting of MAC II.
Bypass	BPC		Yes	No	No	No	No	None	CTRL 18	3		00018		
Compare	CMP	SU	Yes	Yes	Yes	Yes	Yes	Low Order		4	00-15			
Divide	DIV		Yes	Yes	Yes	Yes	Yes	Low Order		W	00			
Dump	DMP		Yes	Yes	Yes	Yes	Yes	High Order	WR 01	R	01		B	Requires 0 or 5 address if DS.
Forward Space	FSP		Yes	Yes	Yes	Yes	Yes	High Order	RD 01	Y	01		B	
Halt	HLT		Yes	Yes	Yes	Yes	No	None		J				
Indicator Off	IOF		Yes	No	No	No	No	None	CTRL 0	3		00000		
Indicator On	ION		Yes	No	No	No	No	None	CTRL 3	3		00003		
Lengthen	LNG		Yes	Yes	Yes	Yes	No	None		D	00			
Load	LOD	SU	Yes	Yes	Yes	Yes	Yes	Low Order		8	00-15			
Load Address	LDA	SU	Yes	Yes	Yes	No	Yes	Low Order		#	00-15		A	
Multiply	MPY		Yes	Yes	Yes	Yes	Yes	Low Order		V	00			
No Operation	NOP		Yes	Yes	Yes	Yes	Yes	Low Order		A	00			
Norm Tr	NTR	SU	Yes	Yes	Yes	No	Yes	Low Order		X	00-15		A	

Instruction	Programmer Writes: Mnemonic	Programmer Writes: Numerical	Permissible Autocoder Entries: Blank	Actual	(E) Descriptive	Literal	(F) Location Ctr.	Character Addressed	Presently Known as	Actual Machine Code Produced: Operation	Stor. Unit	Addr.	Code	Comments
Prep to Rd while Wr	PRW		Yes	No	No	No	No	None	CTRL 15	3		00015		
Print Tape Write	PTW		Yes	No	No	No	No	None	CTRL 29	3		00029		
Read	RD		Yes	Yes	Yes	Yes	Yes	High Order		Y				Requires 0 or 5 address if DS.
Rd Mem Addr	RMA		Yes	Yes	Yes	No	Yes	High Order	RD 02	Y	02		B	Requires 0 or 5 address if DS.
Rd Tape to TRC	RTS		Yes	No	No	No	No	None	CTRL 16	3		00016		
Rd while Wr	RWW		Yes	Yes	Yes	Yes	Yes	High Order		S				
Rd-Wr Start	RWS		Yes	No	No	No	No	None	CTRL 26	3		00026		
Rd-Wr Tape	RWT		Yes	No	No	No	No	None	CTRL 27	3		00027		
Receive	RCV		Yes	Yes	Yes	Yes	Yes	High Order +4		U	00		A	Literal only if divisible by 5.
Receive Serial	RCVS		Yes	Yes	Yes	Yes	Yes	High Order	RCV 01	U	01		B	
Reset Add	RAD	SU	Yes	Yes	Yes	Yes	Yes	Low Order		H	00-15			
Reset Counter	RST		Yes	No	No	No	No	None	CTRL 28	3		00028		
Reset Subt	RSU	SU	Yes	Yes	Yes	Yes	Yes	Low Order		Q	00-15			
Rewind	RWD		Yes	No	No	No	No	None	CTRL 2	3		00002		
Round	RND		Yes	Yes	Yes	Yes	No	None		E	00			
Select	SEL		Yes	Yes	Yes	Yes	No	None		2				
Send	SND	SU	Yes	Yes	Yes	Yes	Yes	High Order +4		/	00-15		A	Assumes presetting of MAC II to 4 or 9. Literal only if divisible by 5.
Set Bit Alt	SBA		Yes	Yes	Yes	Yes	Yes	Low Order	SB 07	%	07			
Set Bit 1	SBN	BIT	Yes	Yes	Yes	Yes	Yes	Low Order	SB 09-14	%	09-14		B, C	
Set Bit Redundant	SBR		Yes	Yes	Yes	Yes	Yes	Low Order	SB 08	%	08		B	
Set Bit 0	SBZ	BIT	Yes	Yes	Yes	Yes	Yes	Low Order	SB 01-06	%	01-06		B, C	
Set Left	SET	SU	Yes	Yes	Yes	Yes	No	Length		B	00-15			
Set Record Counter	SRC		Yes	Yes	Yes	No	Yes	Length $\times 10^{(F)}$	WR 02	R	02		B	The units digit is ignored but must be 0 or 5, e.g., coder must write SRC @ 40 to set counter to 4.
Shorten	SHR		Yes	Yes	Yes	Yes	No	None		C	00			
Sign	SGN	SU	Yes	Yes	Yes	Yes	Yes	Low Order		T	00-15			
Skip	SKP		Yes	No	No	No	No	None	CTRL 9	3		00009		
Store	ST	SU	Yes	Yes	Yes	Yes	Yes	Low Order		F	00-15			
Store for Print	SPR	SU	Yes	Yes	Yes	Yes	Yes	Low Order		5	00-15			
Subtract	SUB	SU	Yes	Yes	Yes	Yes	Yes	Low Order		P	00-15			
Suppress	SUP		Yes	No	No	No	No	None	CTRL 5	3		00005		
Transfer	TR		Yes	Yes	Yes	No	Yes	Low Order		1			A	
Tr Alt Sw A On (00911)	TAA		Yes	Yes	Yes	No	Yes	Low Order	TRA 01	1	01		A, B	

Instruction	Programmer Writes: Mnemonic	Programmer Writes: Numerical	Permissible Autocoder Entries: Blank	Actual	(E) Descriptive	Literal	(F) Location Ctr.	Character Addressed	Presently Known as	Actual Machine Code Produced: Operation	Stor. Unit	Addr.	Code	Comments
Tr Alt Sw B On (00912)	TAB		Yes	Yes	Yes	No	Yes	Low Order	TRA 02	I	02		A, B	
Tr Alt Sw C On (00913)	TAC		Yes	Yes	Yes	No	Yes	Low Order	TRA 03	I	03		A, B	
Tr Alt Sw D On (00914)	TAD		Yes	Yes	Yes	No	Yes	Low Order	TRA 04	I	04		A, B	
Tr Alt Sw E On (00915)	TAE		Yes	Yes	Yes	No	Yes	Low Order	TRA 05	I	05		A, B	
Tr Alt Sw F On (00916)	TAF		Yes	Yes	Yes	No	Yes	Low Order	TRA 06	I	06		A, B	
Tr and Store Location	TSL		Yes	Yes	Yes	No	Yes	Low Order	TR 01	I	01		A, B	Assumes presetting of MAC II.
Tr Any	TRA		Yes	Yes	Yes	No	Yes	Low Order		I			A	
Tr 0 Bit	TZB	BIT	Yes	Yes	Yes	No	Yes	Low Order	TZB 01-07	.	01-07		A, B, C	Assumes presetting of MAC II.
Tr Echo Check (00903)	TEC		Yes	Yes	Yes	No	Yes	Low Order	TRS 13	O	13		A, B	
Tr Equal	TRE		Yes	Yes	Yes	No	Yes	Low Order		L			A	
Tr High	TRH		Yes	Yes	Yes	No	Yes	Low Order		K			A	
Tr Instr Check (00900)	TIC		Yes	Yes	Yes	No	Yes	Low Order	TRS 10	O	10		A, B	
Tr Mach Check (00901)	TMC		Yes	Yes	Yes	No	Yes	Low Order	TRS 11	O	11		A, B	
Tr O'flow Check (00904)	TOC		Yes	Yes	Yes	No	Yes	Low Order	TRS 14	O	14		A, B	
Tr Plus	TRP	SU	Yes	Yes	Yes	No	Yes	Low Order		M	00-15		A	
Tr Ready	TRR		Yes	Yes	Yes	No	Yes	Low Order	TRS 01	O	01		A, B	
Tr Rd-Wr Check (00902)	TRC		Yes	Yes	Yes	No	Yes	Low Order	TRS 12	O	12		A, B	
Tr Sign Check (00905)	TSC		Yes	Yes	Yes	No	Yes	Low Order	TRS 15	O	15		A, B	
Tr Signal	TRS		Yes	Yes	Yes	No	Yes	Low Order		O			A, B	
Tr Sync Any	TSA		Yes	Yes	Yes	No	Yes	Low Order	TRS 03	O	03		A, B	
Tr Transm'n Check	TTC		Yes	Yes	Yes	No	Yes	Low Order	TRS 02	O	02		A, B	
Tr 0	TRZ	SU	Yes	Yes	Yes	No	Yes	Low Order		N	00-15		A	
Transmit	TMT		Yes	Yes	Yes	Yes	Yes	High Order +4		9			A	Literal only if divisible by 5.(F) Assumes presetting of MAC II to 4 or 9.
Transmit Serial	TMTS	ASU	Yes	Yes	Yes	Yes	Yes	High Order	TMT 01-15	9	01-15		B	Assumes presetting of MAC II.
Unload	UNL	SU	Yes	Yes	Yes	Yes	Yes	Low Order		7	00-15			
Unload Address	ULA	SU	Yes	Yes	Yes	No	Yes	Low Order		*	00-15		A	
Write	WR		Yes	Yes	Yes	Yes	Yes	High Order		R				Requires 0 or 5 address if DS.
Wr Erase	WRE	SU	Yes	Yes	Yes	Yes	Yes	High Order		Z	00-01			
Wr TRC to Tape	WST		Yes	No	No	No	No	None	CTRL 17	3		00017		
Wr Tape Mark	WTM		Yes	No	No	No	No	None	CTRL 01	3		00001		

Appendix E

Chart 1. Instruction Operation Differences Between the Central Processing Units of 705 I, II and 705 III

INSTR.	CODE	705 I AND II	705 III	PROGRAMMER'S NOTES
All Instr.		Ignores zoning in units position.	B-bit in units position indicates upper 40K (except LNG, RND, SEL, SET, SHR). A-bit in units position indicates indirect addressing.	Programmer must check all instructions that have units position zoning. All instructions including NOP must have a 4 or 9 in the numerical portion of the units position if the A bit is present.
ADD	G	Execution not completed until end of both memory and storage field is reached.	Execution terminates no later than at the end of the storage or memory field, whichever is longer, but could terminate sooner, depending on carry requirements, at or after the end of the memory field.	On the 705 I and II, it is possible to strip zoning from an entire storage field by adding $\overset{+}{0}$ or $\bar{0}$. On the 705 III, the following would result: ADD: STORAGE aABCD + MEMORY b$\bar{0}$ RESULT aABC4+ In order to strip zoning from the entire storage field on the Model III, it will be necessary to add a *field* of zeros equal in length to the storage field. For example: ADD: STORAGE aABCD + MEMORY b000$\bar{0}$ RESULT a1234+
ADM	6	Execution of ADM to signed memory field is completed at end of *memory* field.	Execution of ADM to a signed memory field will terminate no later than at the end of the memory field, but could terminate sooner, depending on carry requirements, at or after the end of a shorter storage field.	
CMP	4		Internal checking of the compare indicators may cause a 00901 check.	
NOP	A			When using a NOP instruction as a constant, observe that if the A bit is present in the units position, the numerical portion must be a 4 or 9.
RD	Y		See Chart 2.	
RWW	S		See Chart 2.	
ST	F	Alphabetic characters may cause redundancies upon being stored but will not cause 00901 check indication until used in subsequent operations. The units position normally does not cause a redundancy because a new C bit is generated for it.	The character in the units position is handled as in 705 II. In other positions, alphabetic characters may cause redundancies upon being stored and in any case will cause an immediate 00901 check.	The storing of alphabetic characters is sometimes used during 705 I and II program testing to create redundancies. For example: LOD 00 (JK) ST 00 2001 SEL 0200 WR 2000 (00901 and 00902 check indication) On the 705 III, an immediate 00901 occurs when storing the alphabetic character J, but not the K. On the 705 III, SBR (set bit redundant) should be used to obtain a redundancy.
SUB	P	Same as ADD.	Same as ADD.	
TR	1	TR(00-15) = unconditional transfer.	TR00 = unconditional transfer TR01 = (transfer store location) TSL TR02-15 = unconditional transfer	Programmer need modify only TR instructions with ASU 01 zoning.
TRA	I	TRA(00-15) will transfer if ANY Trigger is on.	TRA(00) = TR if ANY Trigger is on 01 = TAA = SEL 00911 TRS- - - - 02 = TAB = SEL 00912 TRS- - - - 03 = TAC = SEL 00913 TRS- - - - 04 = TAD = SEL 00914 TRS- - - - 05 = TAE = SEL 00915 TRS- - - - 06 = TAF = SEL 00916 TRS- - - - (07-15) = NOP	Programmer must check for TRA with any ASU zoning.

INSTR.	CODE	705 I AND II	705 III	PROGRAMMER'S NOTES
TRS	O	TRS 00 = TR if selected unit I/O indicator is on. TRS 01-15 = TR if selected tape exists on the 754/777, if the ready light is on, and if tape is not rewinding. TRS 01-15 = NOP if the selected tape does not exist on the 754/777, and the unit has not been placed in ready status or the unit is in the process of rewinding.	TRS 00 = TR if I/O indicator is on TRS 01 = TR Ready (TRC or DS) TRS 02 = TR if DS data check indicator (PCT) is on TRS 03 = TSA (see Appendix D) TRS 04-09 = NOP TRS 10 = TIC = SEL 00900 TRS- - - - TRS 11 = TMC = SEL 00901 TRS- - - - TRS 12 = TRC = SEL 00902 TRS- - - - TRS 13 = TEC = SEL 00903 TRS- - - - TRS 14 = TOC = SEL 00904 TRS- - - - TRS 15 = TSC = SEL 00905 TRS- - - -	See Chart 2 for details.
WR	R		See Chart 2.	
WRE	Z		See Chart 2.	

Chart 2. Data Synchronizer Input/Output Modes of Operation and Differences Between Instruction Operations in the 754, 760, 767 and 777

INSTR.	CODE	754 Tape Control and 760 Control and Storage		777 Tape Record Coordinator		767 Data Synchronizer
		705 I and II	705 III	705 I and II	705 III	
BSP (CTR)	3	BSP00-15 = BSP00	No change	BSP00-15 = BSP00	No change	BSP 00 = BSP 00 BSP 01 = BSF (backspace file) BSP (02-15) = BSP 00
RD	Y	RD00 = RD00 RD(01-15) = RD01 00902 indicates error when reading	No change No change	RD00 = RD00 RD(01-15) = RD01 00902 indicates error when reading	No change No change	RD00 = RD00 RD01 = FSP (forward space) RD02 = RMA (read memory address) RD(03-15) = RD00 1. Units position of the read address must be 0 or 5. If not, the instruction is automatically converted to a FSP (RD01). The DS check indicators are turned on and the machine will stop at end of execution time. 2. Programmer must check all RD instructions zoned 02-15 since these instructions indicate to the DS either RMA or RD00, instead of RD01. 3. Does not turn on 00902 or the ANY indicator. 4. Characters enter memory in blocks of five. If there are insufficient characters on tape to fill the last block, group marks will be generated to fill the block. 5. A RD instruction which has not been completed on reaching memory position 39,999 on a 40,000-position memory (or 79,999 on an 80,000-position memory) is automatically converted to a FSP (RD 01) for the remainder of the operation and turns on the PCT data check indicator.
RWW	S	No change		No change		RWW will initialize MAC II and turn on RWW trigger. (a) Any subsequent WR will cause DS and CPU to hang up in automatic status. (b) Any subsequent RD will operate as normal RD, ignoring RWW trigger.
TRA	I	TRA(00-15) = TR if ANY indicator is on.	TRA00 = TR if ANY indicator is on. TRA(01-15) See Chart 1	TRA(00-15) Same as 754 and 760.	TRA00 Same as 754 and 760. TRA(01-15) See Chart 1	The ANY indicator is not affected by DS operations. See the TRS instructions below.
TRS	O	TRS00 = TR if selected unit I/O indicator is on. TRS01-15 = TRR. (See Chart 1)	TRS00 = TR if selected I/O indicator is on. TRS 01 = NOP on 760 TRR on 754/777 TRS 02 = NOP TRS 03 = NOP TRS(04-09) = NOP TRS 10 = TIC = SEL 00900 TRS- - - - TRS 11 = TMC = SEL 00901 TRS- - - -	TRS00 = TRS00 TRS01 = TRR TRS(02-15) = TRR	TRS00 = TRS00 TRS01 = TRR TRS02 = NOP TRS03 = NOP ready TRS(04-09) = NOP TRS(10-15) = (same as for 754 and 760)	TRS00 = TRS00 TRS01 = TRR = TR *if* ready TRS02 = TTC = TR if Data Check (PCT) is on TRS03 = TSA = TTC and TRS00 *when* ready TRS (04-09) = NOP TRS (10-15) = (Same as for 754—760) A series of 705 I and II instructions such as: 00204 SEL 00604 00209 TRR 00219 00214 TR - - - - 00219 WR (ǂ) 00224 TRA - - - -

INSTR.	CODE	754 Tape Control and 760 Control and Storage 705 I and II	 705 III	777 Tape Record Coordinator 705 I and II	 705 III	767 Data Synchronizer
			TRS 12 = TRC = SEL 00902 TRS- - - - TRS 13 = TEC = SEL 00903 TRS- - - - TRS 14 = TOC = SEL 00904 TRS- - - - TRS 15 = TSC = SEL 00905 TRS- - - -			May be replaced in 705 III DS system by using TSA. 00204 SEL 00200 00209 TSA - - - The TSA holds up all CPU operations until the DS is ready, then a TR is executed only if the tape unit I/O indicator is on or the Data Check Indicator is on. The TRR, TTC and TSA instructions do not turn off any of the DS's indicators. The ANY indicator is never involved in DS operations. The TRS and TTC instructions, if given while the DS is reading or writing, will test the appropriate indicator as it stands at that moment. These instructions are normally given after a TRR loop or a TSA.
WR	R	WR(00) = WR(00)	WR(00) = WR(00)	Same as 754.	Same as 754.	WR(00) = WR00 WR(01) = DMP (Dump) WR(02) = SRC (Set Record Counter) WR(03-15) = WR00 For DS, the units position of a WR address must be 0 or 5. If not, the Data Check (PCT) indicator will be turned on and an address ending in a 4, 5, 6, or 7 will be treated as ending in a 5. Addresses ending in 0, 1, 2, 3, 8 or 9 will be treated as ending with 0. A WR instruction that has not been completed on reaching memory position 39,999 on a 40,000-position memory (or 79,999 on an 80,000-position memory) turns on the PCT data check indicator, but continues until the group mark is sensed.
		WR(01) Writes through the respective memory block, i.e., 19998 or 39998. If addressed to 19999 on 20k machine or 39999 on 40k machine, only one character is written.	WR(01) Writes through the respective memory block, i.e., 19998, 39998, 59998, or 79998. On a 40k, WR(01) 39999 causes one character to be written. With an 80k, WR (01) 39999 causes writing up to 59998, while WR(01) 79998 causes one character to be written.			WR(01) writes through 19999 or 39999 or 59999 or 79999, whichever quadrant is addressed. If a WR(01) is given following a SRC instruction, as many blocks of memory will be written as the number set in the record counter of the selected DS. If the RC is not zero when the end of memory is reached, writing will continue at the beginning of memory until the RC is zero.
		WR(02-15) = WR01 WR(ǂ) = NOP	No change No change	WR(ǂ) = NOP but turns on appropriate 00902, 00903 or any indicators if the TRC has detected an error or EOF in its last operation.	No change.	WR(ǂ) = NOP. The DS Data Check indicator (PCT) is not affected. The 00902, 00903 and any indicators are never affected by DS operations.
		00901, 00902, 00903 turned on as appropriate		00901, 00902, 00903 turned on as appropriate.	No change.	00901, 00902, 00903 are not turned on, but the Data Check (PCT) indicator is turned on instead. No stop will occur.
WRE	Z	WRE(00) = WRE00 WRE(01-15) = WRE01	No change	WRE(00) = WRE00 WRE(01-15) = WRE01	No change.	No WRE Mode. In DS the WRE will be converted into a WR instruction. WRE(00) = WR00 WRE(01) = WR01 WRE(02) = WR02 WRE(03-15) = WR00

Appendix F

Chart of DS Operations

Operation	Overlap With Computation	When Executed if DS Is Busy[2]	Effect on Data Check Indicator[1]	Effect on Address Check Indicator[1]	Memory Address Obtained by Subsequent RMA[5]	Record Counter Control Possible	Effect on Record Counter Control
RD	Yes	When DS is ready	Resets, then may turn on	Resets, then may turn on	Last memory reference + 5	Yes	Turns off at end of count or at TM
FSP = RD 01	Yes	When DS is ready	Resets, then may turn on	Resets, then may turn on	Last memory reference + 5	Yes	Turns off at end of count or at TM
RMA = RD 02	No	When DS is ready	Resets, then may turn on	Resets, then may turn on	Last memory reference + 5	No	Turns off, does one operation
WR	Yes	When DS is ready	Resets, then may turn on	Resets	Last memory reference + 5	Yes	Turns off at end of count
DMP = WR 01	Yes	When DS is ready	Resets, then may turn on	Resets	20000, 40000, 60000, or 00000	Yes	Turns off at end of count
SRC = WR 02	No	When DS is ready	Resets, then may turn on	Resets	Record counter setting	No	Turns on
WTM	Yes	When DS is ready	Resets, then may turn on	Resets	WTM address minus 1	Yes	Turns off at end of count
SKIP	Yes	When DS is ready	No effect	Resets	None. RMA becomes NOP	Yes	Turns off at end of count
BSP	Yes	When DS is ready	No effect	Resets	None. RMA becomes NOP	Yes	Turns off at end of count, at load point, or at TM
BSF = BSP 01	Yes	When DS is ready	No effect	Resets	None. RMA becomes NOP	No	Turns off, does one operation
RWD	Yes after 33 ms.[4]	When DS is ready	No effect	Resets	No effect	No	No Effect
IOF	No	When DS is ready	No effect	Resets	No effect	No	No Effect
ION	No	When DS is ready	No effect	Resets	No effect	No	No Effect
TSA = TRS 03	No	When DS is ready	No effect	Resets	No effect	No	No Effect
TTC = TRS 02	No	Immediately	No effect	Resets	No effect	No	No Effect
TRR = TRS 01	No	Immediately as a NOP	No effect	Resets	No effect	No	No Effect
TRS = TRS 00	No	Immediately[3]	No effect	Resets	No effect	No	No Effect
SEL	No	Immediately[3]	No effect	Resets	No effect	No	No Effect

Note 1. The load key completely resets all DS's including the data check indicator and address check indicator and turns off the record counter control. The start key resets address check indicator.

Note 2. The DS is busy when it is doing a RD, WR, FSP, DMP, WTM, SKIP, BSP or BSF independently of the CPU. But when it is doing a RWD independently of the CPU, it is ready for those tapes that are not rewinding.

Note 3. The select instruction sets the select register in the CPU immediately but has no immediate effect on the unit select register in the DS. When an instruction is given to a DS that is not busy, the unit select register is set to whatever tape unit is designated by the units digit in the CPU select register. Therefore, a TRS 00 addressed to a busy DS interrogates the unit in operation without regard to which unit is selected in the CPU select register.

Note 4. If the RWD instruction is given when tape is already at load point or is already rewinding, an immediate NOP takes place.

Note 5. "Last memory reference" means the 0 or 5 address of the last group of five memory characters handled by the instruction. In the case of FSP = RD 01, it refers to what would have been the last memory reference if the instruction had been a RD 00. In the case of the WR 00, the last group handled is the group *after* the group containing the group mark, although no characters after the group mark go on tape. An error in any character handled, even after the group mark, turns on the data check indicator. RMA becomes NOP if the selected DS has not been used since being manually reset.

Index

Form A22-6506-0 (5/59:9M-ME:L-1)

IBM®

International Business Machines Corporation
590 Madison Avenue, New York 22, N.Y.

Printed in U.S.A. A22-6506-0