IBM 704 Manual of Operation

EDW. STORM
BATSON

IBM

Electronic Data-Processing Machines

TYPE

704

MANUAL OF OPERATION: PRELIMINARY EDITION

MINOR REVISION (Date: January 1956)

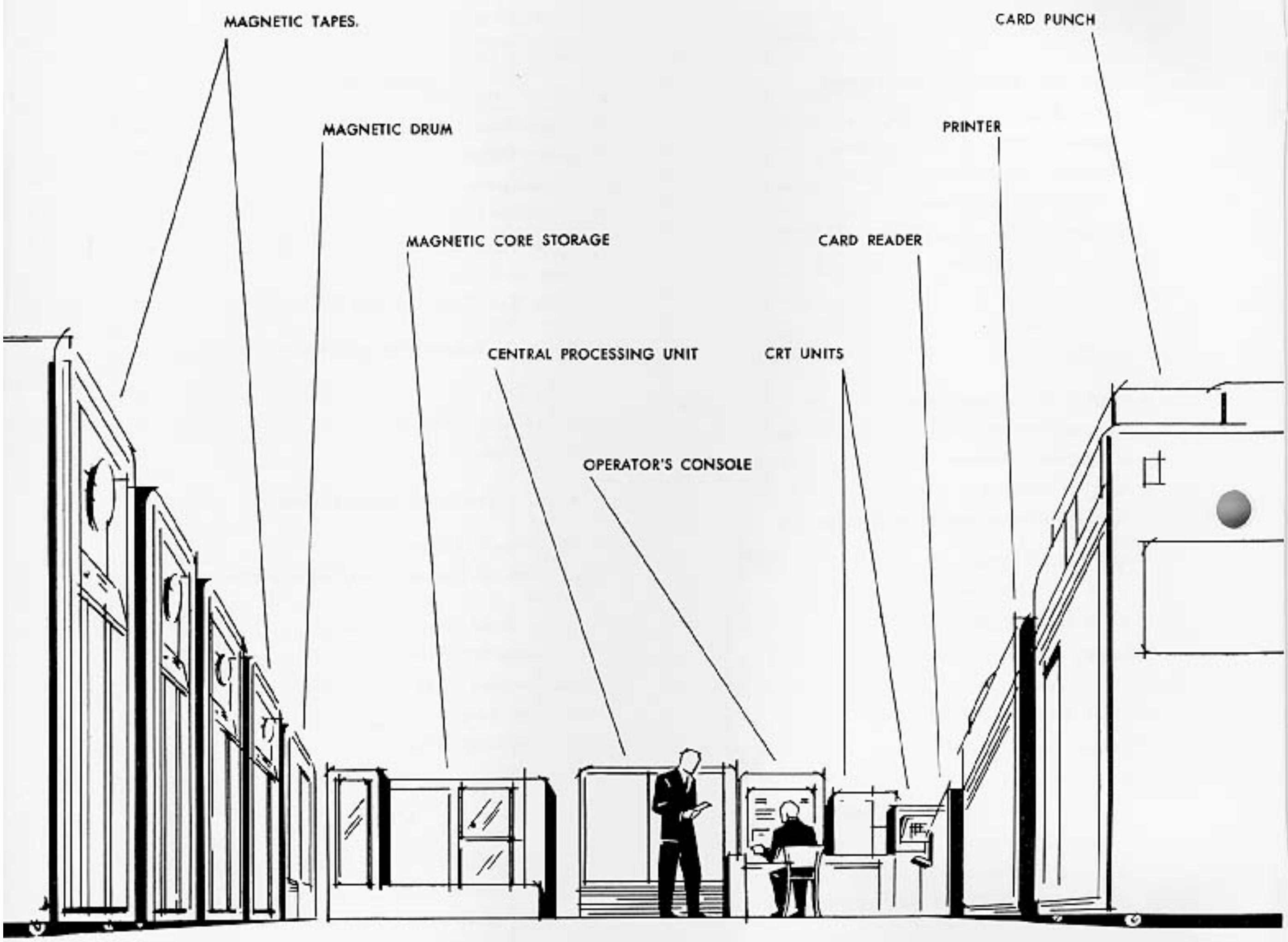
This edition, Form 24-6661-1, is a minor revision of the preceding edition but does not obsolete Form 24-6661-0. Principal changes in this edition are:

PAGE	SUBJECT
10, 11	Removal of fixed instructions ("Addenda").
19, 28, 88	Octal operation code for STP changed from +0623 to +0630.
26, 30, 31, 33	Input-output delay instruction changed in all programs to WRS 333_8.
32, 33	Change in specification of simultaneous tape writing.
33	Tape checking: "Incomplete Word on Tape."
35	Conditions turning on and off the tape indicator light on the tape unit ("Addenda").

Form 24-6661-1

Edward F. Storm

CONTENTS



IBM ELECTRONIC DATA-PROCESSING MACHINES
TYPE 704 AND ASSOCIATED EQUIPMENT

ELECTRONIC DATA-PROCESSING MACHINES
TYPE 704 AND ASSOCIATED EQUIPMENT

THE TYPE 704 Electronic Data-Processing Machine is a large-scale, high-speed electronic calculator controlled by an internally stored program with instructions of the single address type. This machine is designed for higher speeds and larger capacities required by problems of increasing complexity and size which confront business, industry, government and science. These problems include engineering development, scientific research, production scheduling and control, econometrics, logistics, procurement and supply, and many others.

In order to achieve maximum versatility, every function of the machine is under control of the stored program. This versatility allows the machine to execute instructions at the rate of about 40,000 per second on most problems. Also, the functions of getting data in and out of the calculator are controlled by the stored program, and hence, under the complete control of the operator. The great advantage of this system lies in the fact that a customer may build up a library of programs which will perform his special applications at peak machine efficiency.

To achieve greater computing efficiency, the 704 works internally in the binary number system. The input and output, however, may be accomplished directly on standard IBM cards in the familiar decimal number system by programming which does not interfere with maximum reading, punching, and printing speeds. Or the information on cards may be put on a tape on peripheral equipment and the tape will then be the primary input. Similarly, the results of a computation may be put on a tape and, at some later time, punched on cards or printed by peripheral equipment.

The internal high-speed storage on the 704 is magnetic core storage. When the amount of storage available in magnetic core storage is not large enough, magnetic drums are used to store and supply large blocks of information for ready access at frequent intervals. When the amount of storage needed is in excess of the capacities of both core storage and magnetic drums, then magnetic tapes are used. Also, information may be stored on tapes and the tapes may then be removed from the calculator. In this way, large amounts of information can be filed for future reference in a very compact and convenient form. Magnetic tape is a storage and input-output medium that allows rapid reading and writing and can be reused many times.

The stored programs may be written and introduced into the calculator in many ways. Usually the instructions are key punched on cards in their original form and read into the machine. If the program is to be preserved for future use, it can be punched on cards in the binary number system for compactness or recorded on tape and filed away. To prepare the machine for calculation the appropriate magnetic tapes are inserted in the tape units, cards are placed in the punch hopper, if necessary, and the cards containing the instructions and data of the problem are placed in the hopper of the card reader. By pressing one key the calculator may be made to store the program and data of the problem and start computing. From then on operation of the calculator is fully automatic, with all the components being under the complete control of the program, although it is possible for the operator to interrupt the calculation manually at any time.

All of the real work is done in the central processing unit; that is, all additions, subtractions, multiplications, etc. are done in the special registers of the central processing unit. In addition to standard arithmetic, the 704 has instructions which will perform logical arithmetic for increased flexibility in doing complex problems. Also in the central processing unit are three index registers for automatic counting and effective address modification.

An optional feature on the 704 is a complete set of instructions which will perform floating-point arithmetic. This manual includes a complete description of floating-point numbers and the special floating-point instructions (such as floating add, subtract, multiply, divide or halt, and divide or proceed) needed to manipulate data in this form.

STORAGE AND INPUT-OUTPUT UNITS

Access Time

The fundamental machine cycle of the 704 is 12 microseconds. One cycle is the core storage access time, that is, the time required by the central processing unit to transmit or receive a word of information to or from core storage. The time required to transmit information between core storage and any of the input-output units is given in the description of the unit.

Address System

Individual locations (or registers) in magnetic core storage, together with magnetic drums, magnetic tapes, and all input-output units are identified by a system of numerical addresses. By means of a number contained in the *address* part of an instruction, it is possible to refer to the information contained in any register in magnetic core storage or any component of the machine.

Magnetic Core Storage

Information is stored in the primary storage unit by the use of magnetic cores. Each core is a ring of ferromagnetic material. The cores can retain information indefinitely, and recall it in a few millionths of a second. When a wire is inserted through the hollow center of a core, a current passed along the wire sets up a magnetic field around the wire. This magnetizes the core. When the current is removed, the core remains magnetized. If the current is sent along the wire in the opposite direction, the magnetic field set up around the wire is reversed. If the current is again removed, the core will again remain magnetized but its magnetic state will be opposite to that which remained after the first current was removed (Figure 1).

If the first magnetic state can be called positive, the second can be called negative. The positive state can be used to represent a 1; the negative, zero. A group of 36 cores constitutes one register in storage. There are 4096 core storage registers in each unit. Either one or two magnetic core storage units are available.

The principal advantage of magnetic core storage over other types is the very small time necessary to extract information from any given location and send it to the central processing unit. Also the program

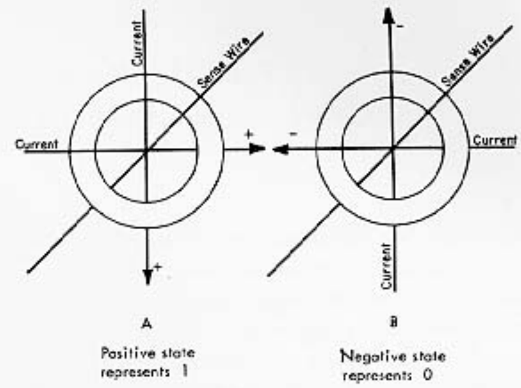


FIGURE 1

has random access to any core storage location. Information is not retained when the power is off.

Magnetic Drum Storage

Additional storage capacity is provided by eight magnetic drums in two drum units. These drums are rotating cylinders surfaced with a material that can be magnetized locally. Binary digits are stored on a drum through the presence or absence of small magnetized areas at cerain locations on the surface of the drum. Each drum has a storage capacity of 2048 words. The location of a word on a drum is identified by a system of addresses analagous to the system used for core storage.

Any part of the information on a drum can be selectively altered at any time. Because access to individual words on a drum is slow in relation to core storage access, it is more efficient to use the drums for storing large blocks of information. After the first word of such a block has been located, the remaining words are transmitted at the rate of 10,000 words per second. Magnetic drums will retain information when the power is turned off.

Magnetic Tapes

For greater internal working storage as well as their input-output function, ten magnetic tape units are available on the 704. Each unit contains one reel of tape which may be 2400 feet long. The tape itself is a plastic, oxide-coated band one-half inch wide. Binary information is recorded on a tape by means of magnetized spots. A block of words recorded consecutively on a tape is called a *record*. The amount of information contained on each tape depends on the lengths of the individual records since there is a certain amount of space between each record to allow for starting and stopping the tape. It is possible to store as many as 900,000 words on each tape. After the tape is in motion, information can be transmitted at the rate of 2500 words per second.

Flow of Information

The magnetic core storage is always connected to the central processing unit; also, it is the site of the stored program which controls the entire calculator. The auxiliary storage media and the input-output devices, on the other hand, are normally disconnected; they become connected only by the execution of certain stored program instructions. The contents of these units may control the calculator only after being copied into core storage. Thus, information flows between input-output components and magnetic core storage through the central processing unit (Figure 2).

WORDS

IN THE 704 the word, or basic unit of information, consists of 36 binary digits (36 bits). Words may be stored in 4096 distinct word locations in each magnetic core storage unit, magnetic drums (8192 words per drum unit), on magnetic tapes (33⅓ words per

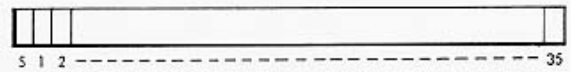


FIGURE 3

inch of tape), or on punched cards (24 words per card).

A word may be an instruction, a fixed-point number, a floating-point number, or any pattern of 36 bits desired by the programmer for any reason. The 36 positions of a word are shown schematically in Figure 3. S refers to the sign position, *1* refers to bit position 1, 2 refers to bit position 2, and so on.

When a word is interpreted as numerical data, the zero position acts as the sign (position S in the diagram) of the word. If the sign position contains a 0, the word is positive; if it contains a 1, the word is negative. When a *logical* operation is performed on a word, the word is interpreted as a 36-bit signless number. As an algebraic (signed) binary number, a word is equivalent to an algebraic decimal number of slightly more than ten digits. Three binary digits are exactly equal to one octal digit, and, therefore, a signless word consists of twelve octal digits.

Instructions

The two principal classes of instructions are referred to as Types A and B. Figure 4 shows the form

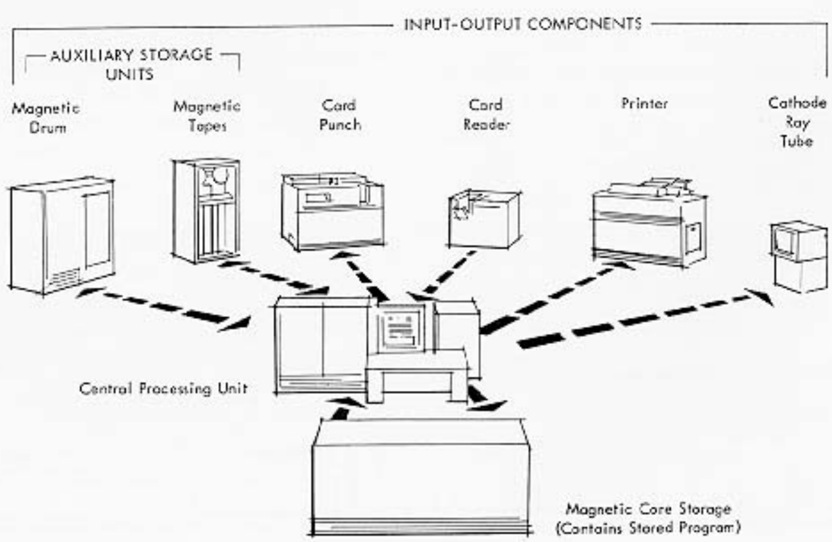


FIGURE 2

TYPE A INSTRUCTION

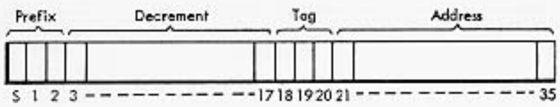


FIGURE 4

of a Type A instruction. Type A instructions use two 15-bit fields (decrement and address) containing numbers in the octal range 00000 to 77777. The prefix contains the operation part while the contents of the tag field select the index register used by the instruction. Positions *1* and *2* of Type A instructions are not both zero.

Bits 21-35 are called the *address* part of an instruction because their principal function is to indicate the storage address of the operand used by the instruction. Bits 3-17 are called the *decrement* part of an instruction because they may represent a number subtracted from the contents of an index register.

Figure 5 shows the form of a Type B instruction. Positions S, *1*, *2*, ...*11*, contain the operation part of Type B instructions, with the exception of the sense-type instructions. These are defined by the code ±0760, and the address part, since they do not refer to a location in storage. Positions 1 and 2 of all Type B instructions are both zero.

Numbers

Numbers are often referred to as data.

Fixed Point. Fixed-point numbers have a sign bit and a magnitude of 35 bits, as illustrated in Figure 6. (Example: The octal fixed-point number + 001367457632 appears in storage as 0 00 000 001 011 110 111 100 101 111 110 011 010.) Theoretically, assume the binary point to be to the right of position 35. However, by proper scale-factoring, the binary point may be placed anywhere in the

TYPE B INSTRUCTION (not sense-type)

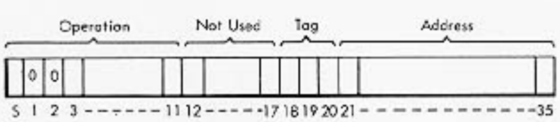


FIGURE 5

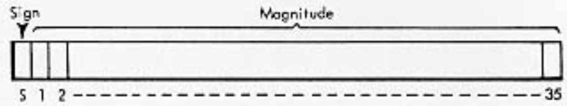


FIGURE 6

number. For example, 0 00 000 . . . 000 010 is equivalent to $1 \times 2^{+1}$.

Floating Point. A floating-point decimal number X may be expressed as a signed proper fraction N times some integral power of 10, or $N \times 10^n$. In the normalized case, the power of ten is chosen so that the decimal point is positioned to the left of the most significant digit of N. Examples:

±	X	=	±	N	×	$10^{\pm n}$
—	.010	=	—	.10	×	10^{-1}
+	.140	=	+	.14	×	10^{0}
+	4.600	=	+	.46	×	10^{+1}
—	88.000	=	—	.88	×	10^{+2}

Similarly, a floating-point binary number X may be expressed as a signed proper fraction B times 2^b where b is an integer. In the normalized case the binary point is positioned to the left of the most significant digit of B. Examples:

±	X	=	±	B	×	$2^{\pm b}$
—	.001	=	—	.100	×	2^{-2}
+	.100	=	+	.100	×	2^{0}
—	1.100	=	—	.110	×	2^{+1}
+	110.000	=	+	.110	×	2^{+3}

In the 704, a floating-point binary number is stored in a register as shown in Figure 7.

1. The magnitude of B is in bit positions *9-35*. A floating-point binary number having a 1 in position *9* is said to be normalized, (i.e., $1/2 \leq |B| < 1$).
2. The sign of B is in the *S* position of the word.
3. Since the sign bit indicates the algebraic sign of the fraction and since signed exponents are desirable, the characteristic, C, of the number, instead of the exponent, is stored in positions *1-8*. The characteristic of the fraction is formed

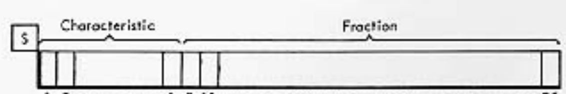


FIGURE 7

by adding +128 to the exponent. Thus, the range of the exponent is $-128 \leq b \leq 127$, while the range of the characteristic is $0 \leq C \leq 255$. (Examples: An exponent of -32 would be represented by a characteristic of $-32 + 128 = +96$. An exponent of $+100$ would be represented by a characteristic of $+100 + 128 = +228$).

CENTRAL PROCESSING UNIT

THE CENTRAL processing unit accomplishes all *arithmetic* and *control* functions. For any given instruction, the time used by the central processing unit to interpret the operation part of the instruction is called the *interpretation* time. The time required to execute an instruction is called the *execution* time. There is some time-sharing between consecutive intructions; that is, while one instruction is being executed, the next instruction is being interpreted, but this rarely concerns the programmer.

Storage Register (SR)

One special register, which will be referred to as the SR, is used for both arithmetic and control functions. Its operation is entirely automatic and will rarely concern the programmer. The SR has a capacity of 36 bits (one word) and serves as a buffer between core storage and the central processing unit. Some of the interpretation of an instruction is performed in the SR. It is also used in the execution of floating-point instructions.

Arithmetic Element

Accumulator (*AC*). The accumulator is a register with a capacity of 37 bits and a sign. See Figure 8.

Nearly every arithmetic operation involves the accumulator. In some operations (for instance, addition, shifting left) it is possible that the contents of the accumulator will overflow positions 1-35. When an overflow occurs, with the exception of overflow caused by the ACL instruction, the AC OVERFLOW

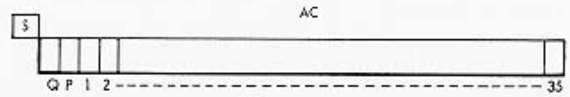


FIGURE 8

indicator is turned on. Certain instructions permit the program to sense the condition of the overflow indicator while the program is being performed. The programmer may preserve some of the overflow information if he wishes. For this purpose, two extra bit positions, or overflow positions, are provided. These are designated the *P* and *Q* positions.

When two numbers having different signs but the same magnitude are added algebraically in the AC, it is important to know if the result is $+0$ or -0, since $+0$ is considered larger than -0. In this case, the sign of the result is identical to the sign of the number in the AC before the addition took place.

Examples: $+6 - (+6) = +0.$
$-6 + (+6) = -0.$

Multiplier-Quotient Register (*MQ*). The MQ is a register with a capacity of 35 bits plus sign. It has five major uses:

1. During the execution of every CPY instruction, the MQ is used as a buffer between core storage and any of the other storage media or input-ouput devices.
2. The multiplier must be placed in the MQ before the execution of a multiplication instruction.
3. After a division instruction is executed, the quotient appears in the MQ (the remainder appears in the AC). In fixed point division, the MQ contains the least significant half of the dividend.
4. After a multiplication instruction is executed, the MQ contains the less significant half of the product. In this connection, the MQ may be regarded as the right-hand extension of the AC; see Figure 9.
5. The least significant 35 bits of the results of FAD, UFA, FSB, and UFS instructions are in the MQ.

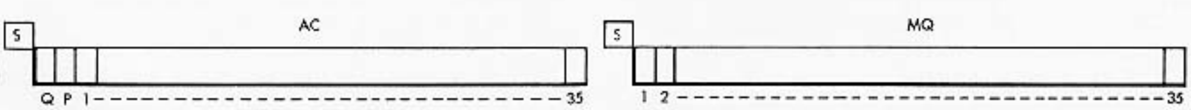


FIGURE 9

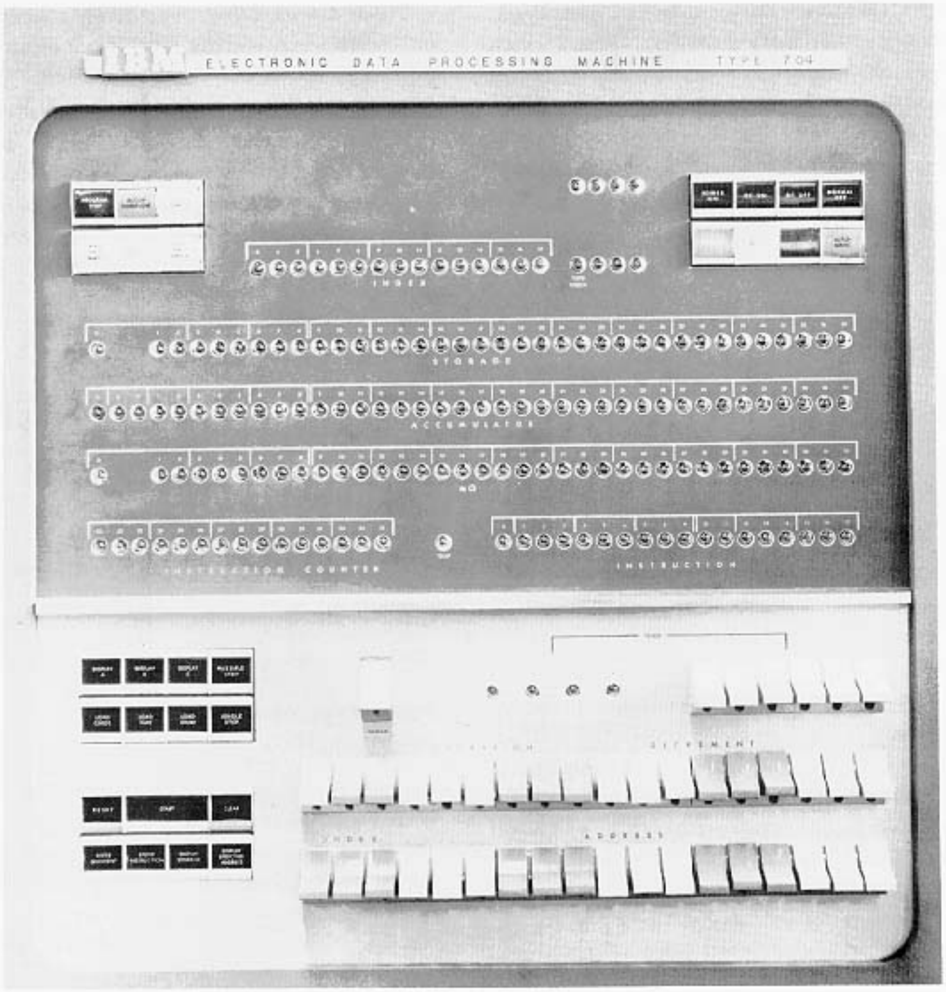


FIGURE 12

points in the program, giving sense instructions (explained under "Instructions") with the addresses of the sense switches causes the calculator to follow one of two courses, depending on which sense switches are depressed. The sense switches are also effective while the calculator is on MANUAL.

Panel Input Switches. There are 36 panel input switches, enabling the operator to insert a word of information into various registers of the calculator while it is on MANUAL. When a panel input switch is down, it represents a 1; when up, it represents a 0.

Index Display Keys. The three index display keys let the operator display the contents of any of the index registers, while the calculator is on MANUAL, by pressing the key marked with the letter corresponding to the index register in question. For example, to display the contents of index register A, the operator presses the key marked DISPLAY A; the contents of index register A then appears in the index lights. The index registers are automatically displayed until the calculator is returned to automatic operation.

Load Keys. The load keys let the operator initiate the loading of a self-loading program stored on binary cards, a drum, or on a tape. If a self-loading program is stored on the tape whose logical identifica-

COMPONENTS

A DETAILED description of each of the Type 704 components will be found in this section.

MAGNETIC TAPE UNITS

IN ADDITION to magnetic core and magnetic drum storage, ten Type 727 tape units with an associated control unit are available on the 704. These tape units are compatible with the tape units used on the Types 702 and 705 EDPM.

Each tape unit may contain a half-inch-wide oxide-coated plastic tape up to 2400 feet long on which information is stored as bits in the form of magnetized spots. The mechanism (read-write head) that reads or writes information on the tape is preceded by an erase head which erases the tape prior to writing, but *not* while reading. Hence, the same tape may be re-used many times by writing new information on it.

The reading, writing, and backspacing speed of the tapes is 75 inches per second. The longitudinal density of the tapes is 200 bits per inch. Reading or writing is done at the rate of 2500 words per second after the tapes are placed in motion. Tapes are read or written in a forward direction only; but the same tape may be written, backspaced, read, backspaced and written again in that order. Thus a record may be written and then read for checking purposes before writing the succeeding record.

The normal rewinding speed of the tapes is 75 inches per second if the length of tape to be rewound does not exceed 450 feet. The tape unit automatically measures the length of tape to be rewound. The time for a high-speed rewind of a reel of tape of any length from 450 to 2400 feet is nearly constant (about 1.2 minutes, allowing for acceleration and deceleration time).

Reflective spots on the tape, made of adhesive aluminum stripping, are photo-electrically sensed to indicate the load point and the physical end of the tape, as indicated in Figure 18.

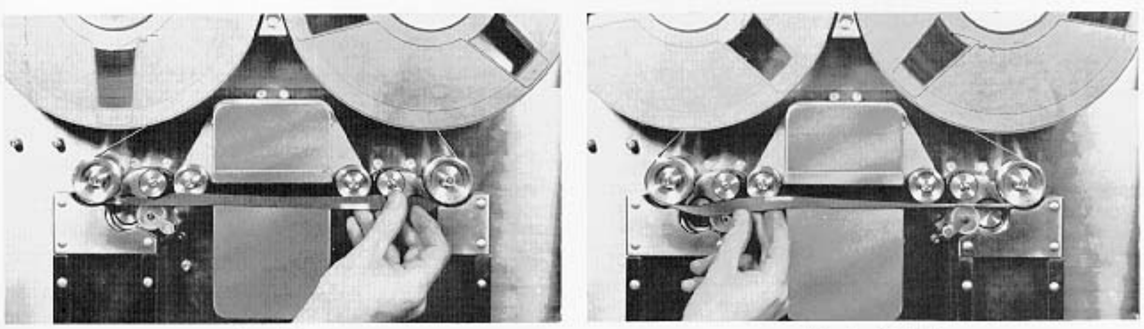
Operating Modes

Peripheral equipment (card-to-tape, tape-to-card, and tape-to-printer) requires information to be stored as binary-coded decimal (BCD) characters. Therefore, the 704 operates in two distinct modes, depending on the address used to select the tape unit:

MODE	OCTAL ADDRESS	DECIMAL ADDRESS
BCD	201-212	129-138
Binary	221-232	145-154

When operating in the binary mode, the calculator reads or writes words without altering the bit pattern during transmission. When reading or writing in the BCD mode, the calculator alters the form of some of the BCD characters during transmission from or to the tape. See "Character Alteration in BCD Mode."

Six bits make up one BCD character. Hence six BCD characters, comprising 36 bits, are transmitted with one copy and skip (CPY) instruction.



Load Point *Physical End*

FIGURE 18

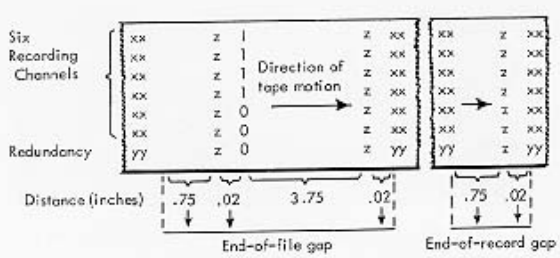
Physical Arrangement of Information on Tape

A ¾-inch blank space on the tape defines the *end of a record* of information. A 3.75-inch blank space, a tape mark followed by its redundancy character, and an end-of-record gap define the *end-of-file* of information (Figure 19). A tape may contain more than one file, and a file may contain any number of records. Each record contains an arbitrary number of words.

During a write operation, six bits and a redundancy check bit are recorded laterally across the tape. The lateral redundancy check bits are automatically placed on the tape to cause an even or odd number of binary 1's in each lateral row of tape for the BCD or binary mode, respectively. Also, at the end of each record written, a longitudinal redundancy check bit is placed automatically in each of the seven channels to cause an even number of binary 1's in each channel of that particular record. The longitudinal check is *always* an even check.

If information stored on a tape in the BCD mode is read in the binary mode, the tape-check indicator and corresponding light on the operator's console go on (because the lateral check bits are different), and the information is transmitted to storage *in unaltered form*. If a binary tape is read in the BCD mode, the tape-check indicator and light turn on, and the information is transmitted to storage in *altered form*.

In Figure 19, the tape is moving in the direction indicated by the arrows. Each *y* corresponds to the redundant bit for each six bits (*x*'s) stored laterally, and each *z* corresponds to the redundant bit for each channel of the preceding record. The tape mark in the end-of-file gap has its own longitudinal check bits .020 inch beyond the tape mark. These check bits are identical to the tape mark—the special character 0001111_2.



Writing

The programming needed to write a record is write select (WRS) Y (Y denotes the tape unit and mode of checking), followed by a CPY instruction, to be repeated as many times as there are words in the record. This iterative procedure is known as a copy loop. After interpreting the first CPY, the calculator automatically delays its execution if the tape is not yet positioned to transmit the first word.

The WRS Y instruction starts in motion the tape designated by Y and selects the checking mode. If the copy loop is terminated, i.e., the calculator fails to receive a CPY within 336 microseconds (μs) of the preceding CPY, the calculator writes the longitudinal check bits and end-of-record gap and disconnects the tape unit. If another CPY is given after the tape is disconnected, the calculator will stop with the read-write check light turned on.

When a tape is written, the MQ cannot be used for computing between successive CPY instructions, or for 500 μs after the final CPY execution. The delay instruction, WRS 333_8 delays any instruction execution until the MQ is free.

Write End of File

The write end of file (WEF) causes the tape to erase an end-of-file gap and write a tape mark plus the corresponding longitudinal check bits. The calculator disconnects the tape immediately upon interpretation of the WEF instruction; hence, the MQ is free for computing while this instruction is executed. No tape instruction may be executed for 50 milliseconds following a WEF instruction.

To write more than one file of information, it is only necessary to write an end of file after writing the first file of information. At any later time, the first record of the second file of information can be written.

Reading

The execution of an RDS instruction starts the tape in motion, selects the checking mode, and clears the MQ. If the MQ is used for computing between the RDS and the first CPY, it must be cleared by the program before the first CPY instruction is given. After a CPY Y, during the reading loop, the word read into

The select light is turned on only when the calculator selects the tape unit. The tape unit is in ready status (the ready light is on), provided the tape is loaded into the columns, the reel door interlock is closed, and the tape unit is not in the process of finding the load point (rewind or load operation). Manual control is indicated when the ready light is off, provided the tape unit is not rewinding or loading and the reel door is closed.

Pressing the start key places the tape unit under control of the tape control unit (and, indirectly, the calculator) and causes the ready light to be turned on, provided the tape unit is in ready status. Pressing the reset key removes the tape unit from the calculator's control. It turns off the ready light, and resets all controls to their normal positions. It also stops any tape operation which has been initiated (except high-speed rewind, which will revert to low-speed rewind). After the tape is loaded into the vacuum columns and low-speed rewind is in progress, the reset key may be pressed again to stop the low-speed rewind.

When the door is open, the reel door interlock prevents operation of the reel drive motors. If the reel door is closed and the ready light is off, pressing the load-rewind key causes a fast rewind at the end of which the tape is loaded into the vacuum columns and searched in a backward direction for the load point. Pressing the unload key causes the tape unit to remove the tape from the vacuum columns and raise the head cover, regardless of the distribution of the tape on the two reels. If the tape is not at the load point when the operator wishes to change it, he starts a load point search by pressing the load-rewind key.

The tape indicator light is turned on when the tape breaks or when the physical end of the tape is reached during a writing operation. The program is allowed to complete the writing operation when the end of the tape is reached. If the program selects the tape for reading or writing after the tape indicator light is turned on, all calculation is stopped and cannot be resumed until the reset key on the operator's console is pressed.

The tape indicator light is turned off by pressing the reset key on the tape unit and then pressing the unload key on the tape unit.

The plastic tape reels are 10½ inches in diameter.

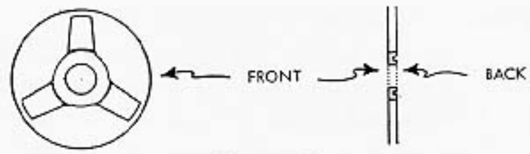


FIGURE 21

They are designed so that the front and back sides of the reel are different (Figure 21). In normal operation, a special ring is inserted in a groove in the back side of the reel to depress a pin which is then under spring tension. If the special ring is removed from the reel, the pin rides freely in this groove and a writing interlock is automatically set. Also, the file protection light is turned on to inform the program that it is impossible for the program to write on the tape. However, this tape may be read, backspaced, or rewound freely when the file protection light is on.

MAGNETIC DRUMS

IN ADDITION to magnetic core and magnetic tape storage, two Type 733 magnetic drum units are available for the 704. Each magnetic drum unit has a storage capacity of 8192 words, each word consisting of 36 bits. A drum unit contains two distinct physical drums, each with a storage capacity of 4096 words.

Each physical drum consists of two logical drums whose octal addresses are indicated in Figure 22. Each logical drum has a storage capacity of 2048 words.

A logical drum is selected by giving the appropriate address 193-200 or 301-310 octal.

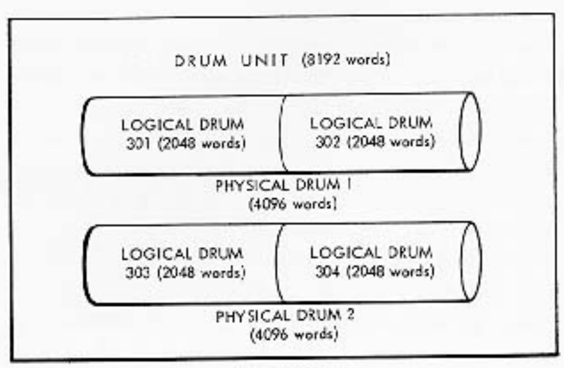


FIGURE 22

Physical Arrangement of Words on Drum

The 2048 locations on each logical drum can be individually addressed by integers in the range 0000-2047 decimal (0000-3777 octal). A record (block) of words is normally stored on a drum in sequentially numbered locations. The programmer must indicate the drum address where the first word of the core storage record is to be written on or read from the drum. The number of CPY instructions executed in the copy loop determines the number of words in the record.

Figure 23 illustrates the physical arrangement of words on a logical drum. The addresses are numbered octally. Observe that, when reading or writing a continuous record, the calculator refers to every eighth word of the drum for consecutive addresses.

Each logical drum has 256 sectors. Therefore, it must make eight complete revolutions for all 2048 words to be read or written as a continuous record.

NOTE: Drum sectors are numbered from 000 to 255 (000 to 377 octal). The eight least significant binary positions of the drum address of a word determine the number of a drum sector where a word is stored.

Reading and Writing

Because 96 μs are needed to read or write one word, successive words are written on or read from the drum at the rate of 10,000 words per second. A drum read select (RDS) Y or write select (WRS) Y selects one of the eight logical drums indicated by the address Y and connects it to the calculator. The drum then remains indefinitely selected waiting for a locate drum address (LDA) instruction. (If an LDA is not given, then the drum remains selected waiting for the first CPY.

The 11 least significant bits of the address part of the C(Y) in the LDA Y instruction specify the initial drum location of the record. If an LDA Y is not given, the first CPY refers to the drum address 0000. The automatic address counter for the drum has only 11 binary positions. Hence, if a 38-word record begins at 2040, the last word of the record is found at location 0029.

Following an LDA, the first CPY must be given within 36 μs (three cycles); otherwise, the drum may disconnect. The LDA is an indexable instruction.

When information is written on a drum, the execution of a CPY Y instruction causes the word at location Y in core storage to be loaded into the MQ from which it is transmitted to the drum. During a reading operation, the execution of a CPY Y instruction causes the word from the drum to be loaded into the MQ from which it is transmitted to location Y in core storage. The MQ cannot be used for computing during the copy loop.

Between successive CPY's, three cycles (36 μs) are available for programming (excluding the CPY itself). When a CPY is not given within three cycles of the preceding CPY, the drum disconnects. If a CPY is given after the drum has disconnected, the calculator stops with the read-write check light on.

Table IV shows the minimum time T between the execution of the RDS or WRS and the LDA (or first CPY if no LDA is given). During this entire time T, the calculator is available for computing. However, if any portion of time T is not used for calculating, the calculator delays the amount of time which is the difference between T and the time used for calculating during this period.

Drum Motion Time

The minimum time between the execution of the last copy of record x and the execution of the first CPY of record $x + 1$ is A, D, or $\bar{A}$ (Table IV). The average access time A is 12.29 ms, although it may be as high as 24 ms.

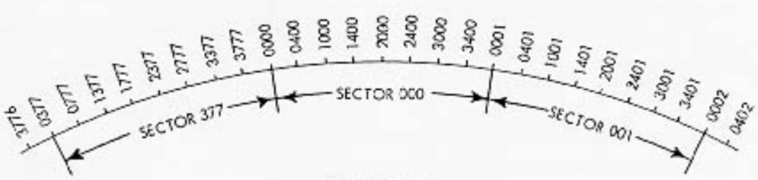


FIGURE 23

PREVIOUS INSTR.	CURRENT INSTR.	T (in ms)	ROTATION TIME (in ms)
RDS 301 or 302	RDS 301 or 302	0.5	D (or A if insufficient information is available to use the formula for D)
RDS 301 or 302	WRS 301 or 302	15.0	
WRS 301 or 302	RDS 301 or 302	15.0	
WRS 301 or 302	WRS 301 or 302	15.0	
RDS 301 or 302	RDS 303-310	0.5	$\overline{A} = A + D'$
RDS 301 or 302	WRS 303-310	15.0	$\overline{A} = A + D'$
WRS 301 or 302	RDS 303-310	15.0	$\overline{A} = A + D'$
WRS 301 or 302	WRS 303-310	15.0	$\overline{A} = A + D'$

TABLE IV

To compute D, divide the difference between the final drum address in the preceding record and the initial drum address in the current record by 256; the quotient Q and remainder R appear in the formula $D = .012Q + .096R$. (This formula is used in computing the rotation time when the physical drum selected is the same as the last one used.)

To compute $\overline{A}$, divide the initial drum address by 256; the quotient Q' and remainder R' appear in the formula $D' = .012Q' + .096R'$. (This formula is used in computing the rotation time when the physical drum selected is different from the last one used.)

The computed rotation time is valid only when A, D, or $\overline{A} > T + .12$ ms.

Multiple Record

Because one drum revolution requires 24 ms, it is possible to read multiple records during a single revolution if the words to be read are stored on one physical drum in an optimal way. If the last CPY of the first record is followed immediately (within three cycles) by an RDS selecting the same physical drum, the LDA may be given for a drum address that is at least eight sectors beyond the drum address of the last word in the preceding record. Six sectors are passed over during execution of RDS and one sector is passed over during execution of LDA. An additional sector must be added for each 84 μs, or portion thereof, beyond the allowable 500 μs of programming between the RDS and the LDA instructions.

For an example, assume that a record is written on a drum where the last word of the record is stored in location 0200_{10}. We wish to know the earliest sector in which to place the first word of the next record so that both records can be read during the same drum revolution when there are (a) 700 μs of programming between records, (b) 840 μs, (c) 500 μs or less.

(a) 700 ÷ 84 = 8 +
= 9 sectors for computing
1 sector to execute LDA
10 sectors to be skipped
Next record can begin at 0211_{10}.

(b) 840 ÷ 84 = 10 sectors for computing
1 sector to execute LDA
11 sectors to be skipped
Next record can begin at 0212_{10}.

(c) The minimum sector allowance is seven sectors (this includes the sector necessary for the LDA).
Next record can begin at 0208_{10}.

PUNCHED CARDS

IN THIS MACHINE, IBM cards are intended to be the primary input medium because of their great flexibility and because of the availability of apparatus for key-punching, verifying, and duplicating. Errors are easily detected and corrected, input data may be readily prepared on several key-punches simultaneously, and the cards may be collected before entry into the computer. Cards are particularly desirable when one wants to have manual access to a file. They can be easily separated. Their contents may be printed on them. It should be emphasized that the punched card input and output may represent any alphabetic character or special symbol, provided only that a program exists to recognize the IBM code for this information. A program may also provide for quantities to be represented in any number system and read or punched accordingly.

Entering a program on cards may be done in such a way that instructions are punched, one to a card, in the form most desirable to the programmer (e.g., in decimal notation). The computer can then be supplied with a standard program to assemble the instructions in the desired order. Then, if errors are detected or if changes must be made, the wrong cards are removed, the correct ones (not necessarily the same number of cards) are added, and the computer prepares the new program. Note that there is no need to repunch any but the cards in question.

The card-feeding mechanism in the card reader is similar to that in the Type 402 Accounting Machine and includes two sets of 80 reading brushes. Correspondingly, there are 80 punching magnets and 80 punching brushes in the card punch. Only 72 columns of the standard IBM card, however, can be read into core storage, and only 72 columns can be punched from core storage (unless split-column wiring is used). Any 72 columns of the card can be selected through control panel wiring. For simplicity in the following discussion, assume that columns 1 to 72 of the card are used for both reading and punching.

Binary information is represented on a card as follows: each of the 12 rows of the card is split into two parts, the left half consisting of columns 1 to 36 and the right half of columns 37 to 72; each half row can be treated as a 36-bit word and read into a location in core storage.

Figure 24 shows how the card is divided. In this particular example, the first 72 columns of the card are used. Each of the rows is split into half-rows of 36 columns each. Thus, the half-row identified by the circled 9 is named the 5-row left. Similarly, the row identified by the circled 10 is named the 5-row right. Thus, there are 24 half-rows in the card. One full word of binary information can be punched in any half-row (including sign). The machine regards any punched hole as a binary 1. "No punch" indicates a binary 0. Thus, an 8-punch in column 36 of the card is regarded by the machine as a binary 1 in the least significant position of the binary word punched in the 8-row left. The leftmost position of each half-row is reserved for the sign bit of the word. A binary 1 represents a negative sign, while a binary 0 represents a positive sign.

NOTE: The exact position of a word that each column represents is completely arbitrary according to how the particular control panel is wired.

Observe that this card representation of 24 binary words does not mean that the cards must always be punched with true binary information. The holes in the card can just as well be numerical punching in the standard decimal card code, alphabetic punching,

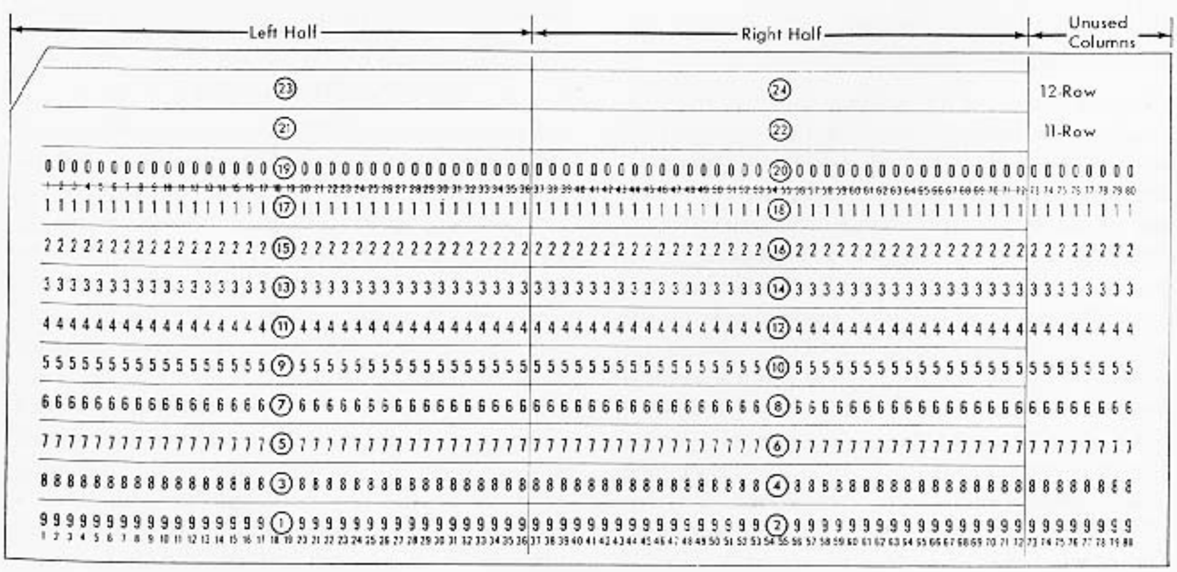


FIGURE 24

or control punching. It is necessary only to provide a suitable program for the computer to translate between the binary code in which it operates and the particular code used on the card. The translation to and from the decimal numerical code, for instance, can proceed simultaneously with reading and punching so that the over-all card-handling speed is not reduced below the standard rates of 150 or 250 cards per minute for reading and 100 cards per minute for punching.

Feed cards face down, 9's edge first, in *both* the card reader and card punch. The internal card circuits are arranged so that the 24 half-rows of the card are read or punched in the sequence indicated by the circled numbers in Figure 24. The sequence of reading or punching full words is then as follows: 9-row left, 9-row right, 8-row left, 8-row right, and so on to 12-row left, 12-row right.

For reading and punching cards, a unit record is defined as the information contained in *one* card. A file consists of any number of unit records. It takes the form of a deck of cards. Note that definitions of unit records and files are usually different, depending on the particular input or output component being discussed.

CARD READER

EITHER one of the two Type 711 card readers, model 1 or model 2, can be used on the 704. The model 1 reads cards at the rate of 150 cards per minute, the model 2 reads cards at the rate of 250 cards a minute. The principal difference is found in the timing section.

For a program to cause the calculator to read all of the information punched on a card into core storage, it is necessary to give an RDS instruction with an address of 209 (card-reader identification) followed by 24 CPY instructions.

The RDS instruction causes the card-feeding mechanism to start in motion. The program then is free to continue any operations until the 9-row of the card appears under the reading brushes. At this time, the program must provide a CPY Y which causes the word punched in the 9-row left to be read and stored in core storage location Y. The program can then resume until the calculator is prepared to read information punched in the 9-row right. The program now must supply another CPY instruction to read this word into core storage. This procedure continues until all 24 half-rows have been read. Because of their functions, these CPY instructions are called 9 left CPY, 9 right CPY, and so on. Another RDS must be given to read another unit record (card).

The RDS instruction can be given, followed immediately by the 24 CPY instructions in succession, without any other operations being done between instructions. In such a case the calculator waits automatically until a half-row is in position to be read before executing the CPY instruction.

The intervals of time between these instructions which may be used for useful calculating are definitely limited and are completely specified below. If a CPY is given *after* the card reader is in position to read a given half-row, the machine stops, and the read-write check light turns on at the operator's console. The amount of calculating time available between the last CPY instruction for a given card and the RDS instruction that initiates the reading of a succeeding card is unlimited. But if an RDS instruction does not occur within a definite time limit, the card reader stops. It will start up only after the new RDS instruction has been received. To keep the card reader in continuous motion and operating at its full speed of 150 or 250 cards per minute, the time limits discussed below must be observed.

Calculator operation is such that during execution of a CPY, the word read from a half-row of the card first enters the MQ before being sent to core storage. This, of course, destroys any information previously stored in the MQ.

If a 25th CPY instruction is given after an RDS instruction, the card reader will already have set up an end-of-record condition (denoting that all 24 half-rows of the card have been read). Under this condition, the 25th CPY is not executed, and the program skips to the *third* instruction after the CPY. In this way the program may transfer control to a section that will cause the succeeding card to be read.

When the hopper of the card reader becomes empty, the calculator stops. Depress the start key on the card reader to allow the cards remaining ahead of the reading station to be read under control of the program. After the last card has been read in this way, and if another RDS instruction followed by a CPY is given, the card reader sets up an end-of-file condition. Under this condition the CPY instruction is not executed,