File No. 1620-00
Form A26-5718-0

1

IBM Systems Reference Library

IBM 1620 Systems Summary

Brief descriptions are provided for the system concepts, the system units and special features, and the programming systems and available programs. The purpose of this publication is to help the user achieve a basic understanding of the system and the interrelationship of its many parts.

Publications providing detailed information on the subjects discussed in this summary are listed in the IBM *1620 Bibliography* (Form A26-5692).

Copies of this and other IBM publications can be obtained through IBM Branch Offices. Address comments concerning the content of this publication to: IBM, Product Publications Department, San Jose, California

Contents

IBM 1620 Data Processing System

The IBM 1620 Data Processing System is an electronic computer system designed for scientific and technological applications. The use of solid-state circuit components and the availability of from 20,000 to 60,000 positions of core storage provide the 1620 system with the capacity, reliability, and speed to solve problems that in the past have required the use of larger and more expensive data processing systems.

Preparation of programs for the 1620 is simplified by IBM advanced programming systems and the IBM library of utility routines. Similar programs are used with the IBM 700, 1400, and 7000 series Data Processing Systems. The Symbolic Programming System (SPS) assembles a program written in mnemonic and symbolic notation by converting mnemonics to machine language and assigning core storage locations to both data and instructions, thus reducing the clerical work involved. The library of utility routines provides a series of thoroughly tested programs that perform most of the more standardized computations and routine tasks that occur in many compution problems.

Six units are available with the IBM 1620 Data Processing System.

The IBM 1620 Model 1 Central Processing Unit contains the computer, 20,000 positions of core storage, a console panel, and an input/output (I/O) typewriter. Paper tape I/O operations are permitted by the IBM 1621 Paper Tape Reader unit, which also includes the paper tape controls and the IBM 1624 Tape Punch.

The IBM 1622 Card Read-Punch is available for card I/O operations.

The IBM 1311 Disk Storage Drive provides virtually unlimited random and sequential access storage. A disk pack containing two million digits of information can be removed from the 1311 and another pack put into its place in one to two minutes.

The ease of mobility of a disk pack (the weight of the pack is less than 10 lb) and the simplicity of its removal from the drive means that two million digits of data can be placed in the system within seconds. Up to four disk drives, each equipped with one disk pack, can be placed "on-line" to make available at one time eight million numerical characters (equivalent to 100,000 80-column IBM cards).

The IBM 1623 Core Storage unit expands the 20,000 core storage positions in the Central Processing Unit of the 1620-1 to 40,000 or 60,000 positions. The 1623 Model 1 contains 20,000 additional positions and the 1623 Model 2 contains 40,000 additional positions.

The IBM 1443 Printer offers printing speeds from 150 to 600 lines per minute with 120 to 144 characters per line. An interchangeable reciprocating type bar allows the operator to change type styles and character sets for specific applications.

The IBM 1620 Model 2 Central Processing Unit can be used in the 1620 Data Processing System instead of the Model 1 Central Processing Unit. The 1620-2 system consists of the 1625 Core Storage unit, the 1620-2 Central Processing Unit, and the 731 typewriter (IBM SELECTRIC®) for input/output.

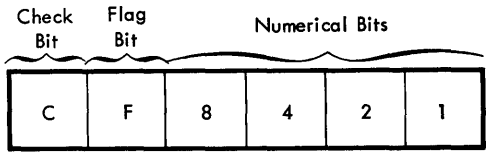
The 1620-2 offers increased computing and processing speeds – up to four times that of the 1620 – and programming compatibility for systems with initial requirements within the range of the 1620. In addition, expanding work loads can usually be transferred from the 1620 to the 1620-2 without reprogramming.

Data Representation

Data can be classified as digits, fields, or records, depending upon the operation in which the data is addressed.

Digits

BCD Bit Array. Each core storage position is addressable and can store one digit of information in binary-coded-decimal (BCD) form (C, F, 8, 4, 2, and 1). Four numerical bits, one flag (F) bit, and one check (C) bit comprise the bit positions of each digit.



The value of a decimal digit is the sum represented by the bits present in the 8, 4, 2, and 1 numerical bit positions. Only bit combinations with sums of nine or less are used. A negative numerical expression has a sign flag in the units position of its field.

Check Bit (C). Each digit position within the computer must contain an odd number of coded bits, including a flag bit, if there is one, for correct parity. To create this odd-bit number, a C bit is automatically added, when required, to each digit position as data enters

core storage. Thereafter, during processing, a digit position with an even number of bits causes the machine to signal a parity error. A C bit alone represents a plus zero.

Flag Bit (F). Depending on its location and the operation performed, the flag bit is used in four ways:

1. Sign Control
2. Field Mark
3. Carries (add table)
4. Indirect Addressing (special feature)

A flag bit over the units position of an instruction address (P or Q) indicates an indirect address.

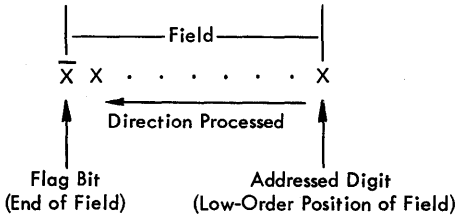
Record Mark (≠). The record mark is a nondecimal machine digit coded C-8-2. It is used primarily in input/output operations and in record transmission within the 1620; it cannot be used as a significant digit in an arithmetic or compare operation.

Group Mark (≠). The group mark (coded 0-8-4-2-1) is used in disk storage operations to verify the correct length of records written on or read from disk storage. The group mark may be used in input/output operations and in record transmission operations within the 1620; it cannot be used as a significant digit in an arithmetic or compare operation.

Numerical Blank. The numerical blank (coded C-8-4) is used for format control of blank columns in card punching, and cannot be used in arithmetic or compare operations. The I/O instructions, Read Numerically and Write Numerically, further detail the use of the numerical blank.

Field

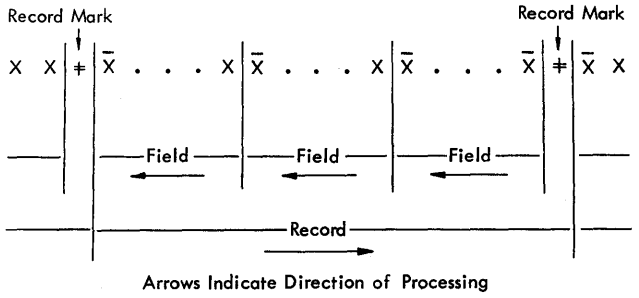
A field is composed of related digits that are treated as a unit of information (temperature, flow rate, etc.); the digits of a field are consecutively addressed. A field is addressed by its rightmost (low-order) digit which occupies the highest-numbered core storage position of the field. Fields are processed from right to left into successively lower-numbered core storage positions until a digit with a flag bit is sensed. The shortest admissible field consists of two digits: the addressed digit which may or may not contain a flag (negative or positive), and the high-order digit containing the flag bit or field mark.



Record

A record consists of a field or fields of related data normally grouped for input/output operations and internal data transmission. A record is addressed at the leftmost (high-order) digit, which occupies the lowest-numbered core storage position of the record. Records are processed serially from left to right into successively higher-numbered core storage positions.

Output and internal record transmission are terminated when a record mark or group mark is sensed, except for card output which is terminated only after 80 columns are transferred to 1622 buffer storage.



Arrows Indicate Direction of Processing

Character Representation

The 1620 can be programmed to read and write numerical and alphameric data. The input/output instruction (numerical or alphameric) determines whether data is read and/or written numerically or alphamerically.

Numerical Representation

One decimal digit is required in core storage to represent a numerical character. No alphabetic or special characters except the record mark, group mark, and numerical blank can be represented in the numerical mode.

Alphameric Representation

Two decimal digits are required in core storage to represent an alphameric character; i.e., an alphabetic character, a special character, or a numerical character. A 2-digit alphameric representation of numerical characters is provided to permit reading of mixed alphabetic, special, and numerical characters without changing from an alphameric to a numerical instruction. The two alphameric digits of a character must occupy adjacent core storage positions, and the zone digit must occupy the even address. This storage requirement is satisfied by programming – alphameric read/write instructions must contain an odd-numbered P address.

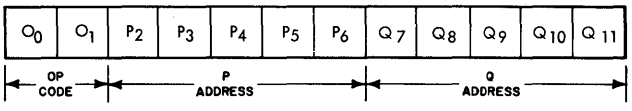
Stored Program Concept

The 1620 Central Processing Unit is a stored program computer, that is, it stores and executes its instructions internally. The computer can perform distinct operations such as adding, subtracting, multiplying, comparing, branching, and so on. It is directed by an instruction placed in core storage to perform a specific operation. To solve a problem or to process data, the programmer selects from various computer operations those necessary to do the desired work. The group of instructions representing the operations to be performed is called a program.

Once the program is placed in core storage, the computer can be directed to execute automatically the instructions composing the program. The program normally is executed in a sequential manner, that is, the computer starts with the first instruction and progresses serially through the program, interpreting and executing each instruction. However, this sequence of operations can be altered by the use of instructions that may direct the computer to an instruction located at other than the next sequential position.

Instruction Characteristics

The 1620 uses a 12-digit machine language instruction divided into three parts: a 2-digit operation (Op) code, a 5-digit P address, and a 5-digit Q address. An instruction as it appears in core storage may be divided into O, P, and Q subscripted numbers, as follows:



In contrast to a data field, which is addressed at its rightmost (low-order) digit and read from right to left, instructions are addressed at O_0, the leftmost (high-order) digit, and read from left to right.

Op Code

Upon initiation of an instruction, the Op code is placed in a 2-digit Op register and is analyzed to determine the operation to be performed. The address of an instruction must always be even, i.e., the O_0 digit of an operation code must be stored in an even-numbered address so that the Op register can receive both digits.

P Address

The P address can specify: (1) the location to which data is transmitted, (2) the location to which the program branches, (3) the location from which data is transmitted (output instructions), or (4) the location of the alphameric field in the Transfer Numerical Strip and Transfer Numerical Fill (1620-1 special feature) instructions.

Q Address

The Q address can specify: (1) the location from which data is transmitted, (2) the indicator being interrogated, (3) the input/output device being used, or (4) the location of the numerical field in the Transfer Numerical Strip and Transfer Numerical Fill (1620-1 special feature) instruction. Also, instruction modifier digits are placed in the Q address for those instructions with the same Op code number.

Instruction Execution Time

The time required to execute an instruction is divided into two parts: the Instruction (I) time and the Execution (E) time.

I Time. The instruction is read out of core storage into instruction registers for decoding. Since all instructions have the same length (12 digits), I time is the same for all instructions — 160 microseconds. There are eight 20-microsecond I cycles in each I time. For the 1620-2, I time is variable.

E Time. The instruction is executed as specified by its Op code and P and Q parts. E time, which is the actual time required to execute the instruction after it has been read into the instruction registers, requires a number of 20-microsecond E cycles. For the 1620-2, an E cycle is 10 microseconds. The number of E cycles varies with individual instructions, size of data fields, speed of I/O devices, etc. The last E cycle of an instruction is followed by the first I cycle of the next instruction.

Immediate Instructions

Certain internal data transmissions, arithmetic, compare, and branch instructions are labeled "immediate." Immediate instructions use the Q_7, Q_8, Q_9, Q_{10}, and Q_{11} digits of the instruction as data instead of as a core storage address. Thus, the Q data is located immediately within the instruction.

System Units and Special Features

This section describes the basic units, additional units, and special features of the IBM 1620, Models 1 and 2. The additional units available with the system are:

1621 Paper Tape Reader
1624 Tape Punch
1622 Card Read-Punch
1311 Disk Storage Drive
1623 Core Storage Unit (for 1620-1 only)
1625 Core Storage Unit, Models 2 and 3 (for 1620-2 only)
1443 Printer, Models 1 and 2

1620 Central Processing Unit, Model 1

The 1620 (Figure 1) is designed for scientific and technological applications. Its standard features include modular construction, solid-state circuitry, high-speed operation, internal self-checking, 20,000 positions of core storage, and a console typewriter for input/output. Core storage capacity may be increased to 40,000 or 60,000 positions with the addition of an IBM 1623 Core Storage unit.

The arithmetic and logic section of the computer is directed by the stored program. The computer performs its operations using a 2-address instruction format. The 2-address instruction format and automatic sequential execution of the programmed instructions simplifies programming and reduces the number of instructions required to solve a problem. The sequence of operations may be altered at any point in the program by unconditional or conditional branch instructions. Conditional branch instructions provide logical decisions through tests performed on a system of indicators and switches set by the computer or the operator.

Addition, subtraction, and multiplication operations are accomplished by a table look-up method in which Add and Multiply tables, located in specific areas of core storage, are referred to automatically when arithmetic operations are being performed. Division is accomplished by a division subroutine or by the Automatic Divide feature. Available special features include:

Automatic Divide
Automatic Floating-Point Operations
Additional Instructions (Transfer Numerical Strip, Transfer Numerical Fill, Move Flag)
Indirect Addressing

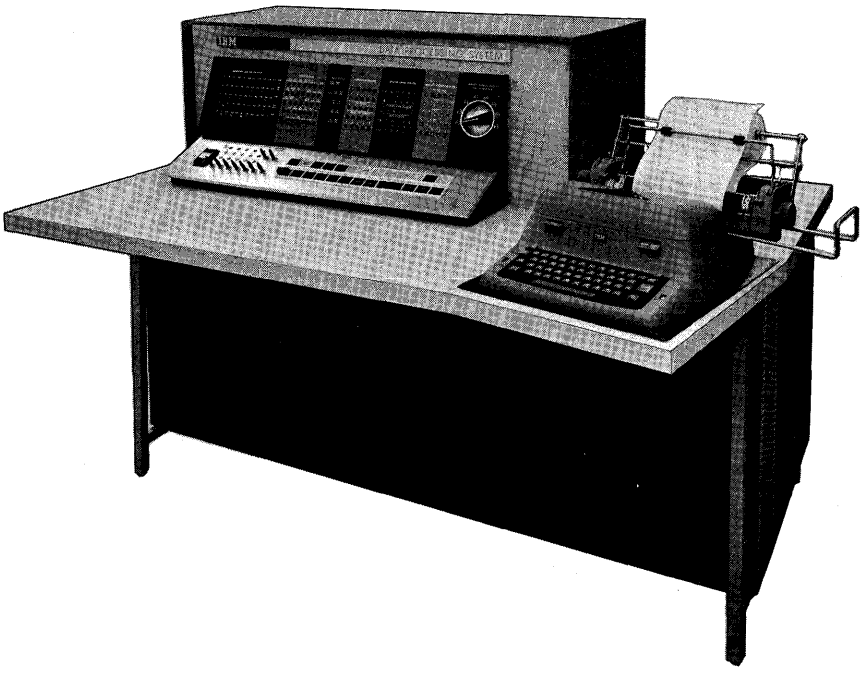


Figure 1. IBM 1620-1 Central Processing Unit with I/O Typewriter

1620 Console

The 1620 Console (Figure 2) is an integral part of the Central Processing Unit and provides for manual or automatic control of the 1620 system. The console lights, keys, switches, and typewriter are used to:

Instruct the machine manually
Display machine and program status indicators
Display the contents of core storage and registers
Place data and instructions in core storage
Alter the contents of core storage
Alter machine functions

Console Typewriter

The 1620 console typewriter is used for both input and output.

TYPEWRITER INPUT

The typewriter is used to enter data and instructions directly into core storage. Off-line use is not possible because the keyboard is locked except when entering data. Depression of the console Insert key unlocks the keyboard and permits data to be entered into core storage starting at location 00000.

TYPEWRITER OUTPUT

The typewriter prints data from core storage when it is programmed to do so. When the right-hand margin is reached, the carriage returns automatically and typing continues until a record mark is sensed or until the Release key is depressed.

1623 Core Storage Unit

The 1623 Core Storage unit (Figure 3), available with the 1620-1, provides the additional program and data storage needed for applications that require more than 20,000 storage positions. Two 20,000-position modules of core storage are available. These expand the storage capacity of the 1620 from 20,000 positions to 40,000 or 60,000 positions, depending on the number of modules installed.

1620 Central Processing Unit, Model 2

The description of the 1620-2 Central Processing unit is similar to the 1620-1 with few exceptions. All core storage is contained in the 1625 Core Storage unit in a 1620-2 system. The outward appearance of the 1620-2 CPU and console, with the exception of the console typewriter (SELECTRIC 731), is the same as the 1620-1 (see Figures 1 and 2). Programming and operating compatibility is maintained even though computing and processing speeds are increased.

The computing and processing speeds of the 1620-2 CPU are derived from the shorter instruction time, improved addition time, ability to handle zero multiplier digits in five memory cycles, and improved data transfer. The 1620-1 transfers one digit every two memory cycles, whereas the 1620-2 transfers two digits every three memory cycles.

The 1620-1 special features – Automatic Division, Indirect Addressing, Transmit Floating and Branch

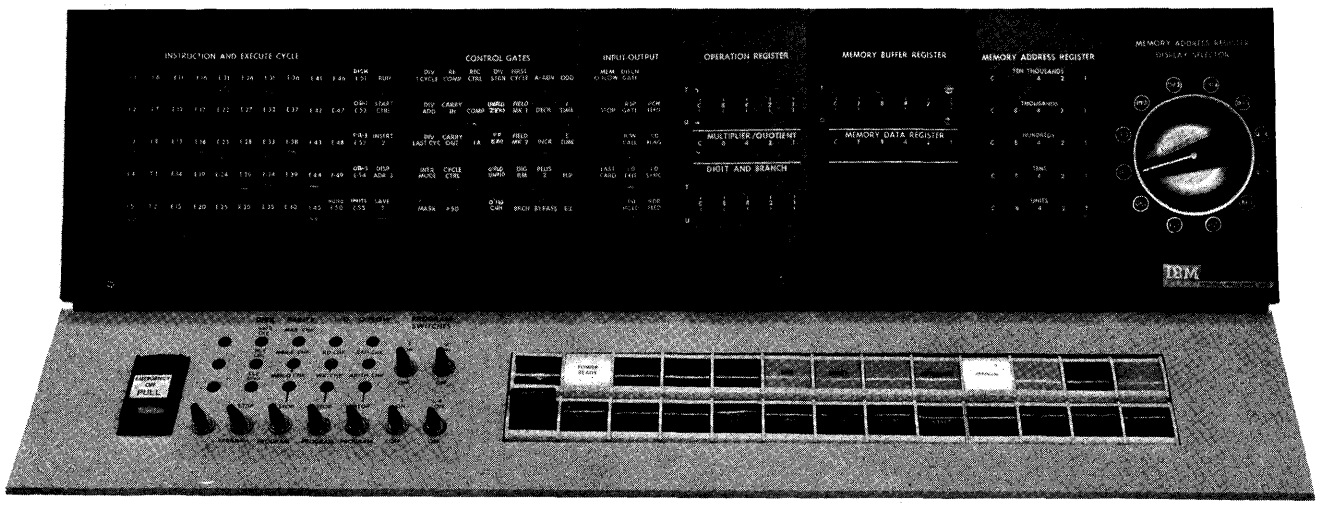


Figure 2. IBM 1620 Console

and Transmit Floating instructions, and Additional Instructions – are included as 1620-2 standard features. Also available with the 1620-2 are:

1621 Paper Tape Reader
1624 Tape Punch
1622 Card Read-Punch
1311 Disk Storage
Automatic Floating-Point Operations (special feature)

1625 Core Storage

Three models of the 1625 Core Storage unit are available with the 1620-2 system: the 1625-1 with 20,000 positions of core storage; the 1625-2 with 40,000 positions of core storage; and the 1625-3 with 60,000 positions of core storage. One model of the 1625 is required for the system to operate. All 1625 models have a core storage access time of 10 microseconds. The outward appearance of the 1625 is similar to the 1623 (Figure 3).

1621 Paper Tape Reader

The 1621 Paper Tape Reader (Figure 4) reads coded alphameric characters from 8-track paper tape at the rate of 150 characters per second. The characters are photoelectrically sensed, converted to 1620 binary-coded decimal (BCD), and placed in core storage.

1624 Tape Punch

The tape punch is housed below the tape reader in the IBM 1621, and punches data from core storage into paper tape at the rate of 15 characters per second. The characters are sent serially from core storage, starting with the location addressed by an output instruction. Each character is translated into an 8-track code before being punched.

1622 Card Read-Punch

The 1622 Card Read-Punch (Figure 5) provides punched card input and output for the 1620. The reader and punch feeds are separate and functionally independent, with individual switches, lights, checking circuits, buffer storage, and instruction codes. Under program control, up to 250 cards per minute can be read and 125 cards per minute punched. Reading, punching, and processing can occur simultaneously because of individual buffer storage.

Figure 3. IBM 1623 Core Storage Unit

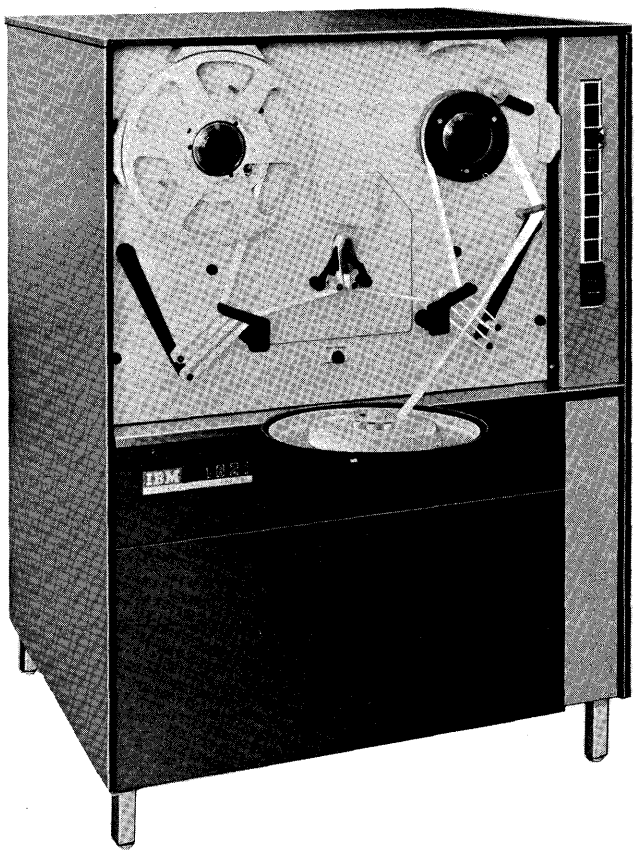


Figure 4. IBM 1621 Paper Tape Reader

1311 Disk Storage Drive, Model 3

The 1311 Disk Storage Drive, Model 3 (Figure 6), provides for random-access storage of two million numerical characters on-line at any one time. One to three additional units, designated Model 2, can be connected for a total on-line random-access storage capacity of eight million numerical characters. In addition, practically unlimited data storage is possible because the disk pack on each drive unit can be easily removed and replaced with another pack.

1443 Printer (Figure 7)

The IBM 1443 Printer is an on-line output unit for both models of the 1620 Data Processing System. Standard features include a tape-controlled carriage for transporting continuous paper forms, an interchangeable reciprocating type bar, and buffer storage.

The easy interchangeability of the reciprocating type bar enables the operator to select the type style and character set for specific applications.

The printer buffer enables the computer to transfer core storage data to the buffer, and then continue processing during the relatively slow printing operation. The 1443 Model 1 prints from 150 to 430 lines a minute; the Model 2, from 240 to 600 lines a minute. A standard print line consists of 120 characters. An additional 24 characters are available as a special feature for each line.

1620 Special Features

Automatic Division

Automatic Division simplifies programming and increases the processing speed of division problems by two to four times that of programmed routines. Only one command needs to be given. Four commands are provided, however, to facilitate positioning of the dividend and divisor in core storage. There are no practical limitations placed upon the size of the dividend, divisor, or quotient.

Indirect Addressing

Indirect Addressing saves program steps and computer time by providing a direct method of address modification. It is primarily used in programs where multiple instructions have the same address, and this address is to be modified by the program. Indirect Addressing may also be used for linking subroutines. Normally, the P or Q address of an instruction specifies the location of the data used during execution of the instruction. An indirect address, however, is the address of a second address, and this second address is the location of the data, or another indirect address. In effect, the address at the indirect address location is substituted for the address specified in the instruction.

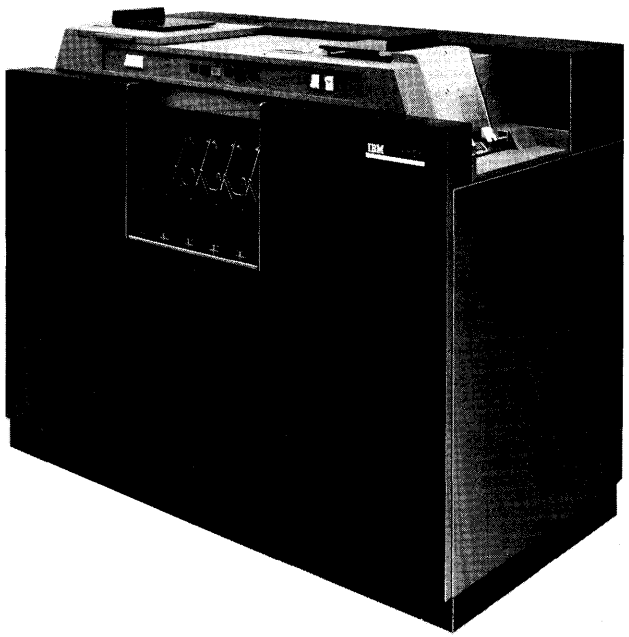


Figure 5. IBM 1622 Card Read-Punch

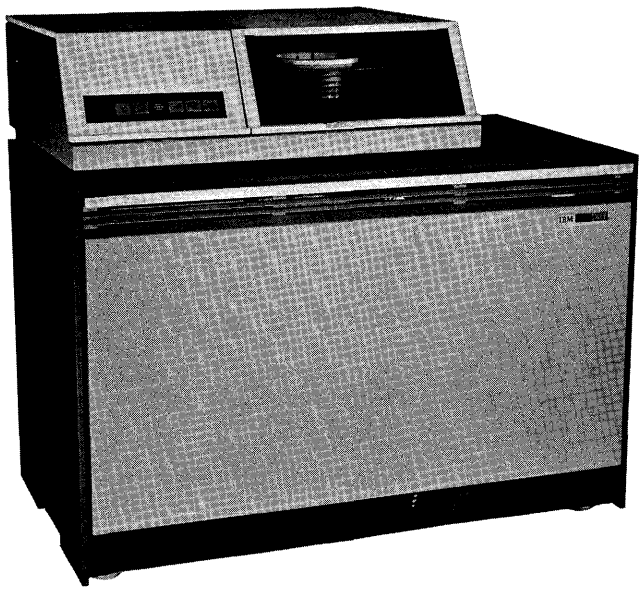


Figure 6. IBM 1311 Disk Storage Drive, Model 3

Additional Instructions

MOVE FLAG

The Move Flag (MF-71) instruction moves a sign or a field definition flag from the core storage location specified by the Q address to the location specified by the P address.

TRANSFER NUMERICAL STRIP

The Transfer Numerical Strip (TNS-72) instruction converts numerical data in the 2-digit alphameric mode into single-digit numerical data, with sign.

TRANSFER NUMERICAL FILL

The Transfer Numerical Fill (TNF-73) instruction moves and expands single-digit numerical data, with sign, into 2-digit alphameric data.

Automatic Floating-Point Operations

Automatic floating-point operations enable the 1620 to compute in floating-point arithmetic using floating-point instructions instead of program subroutines.

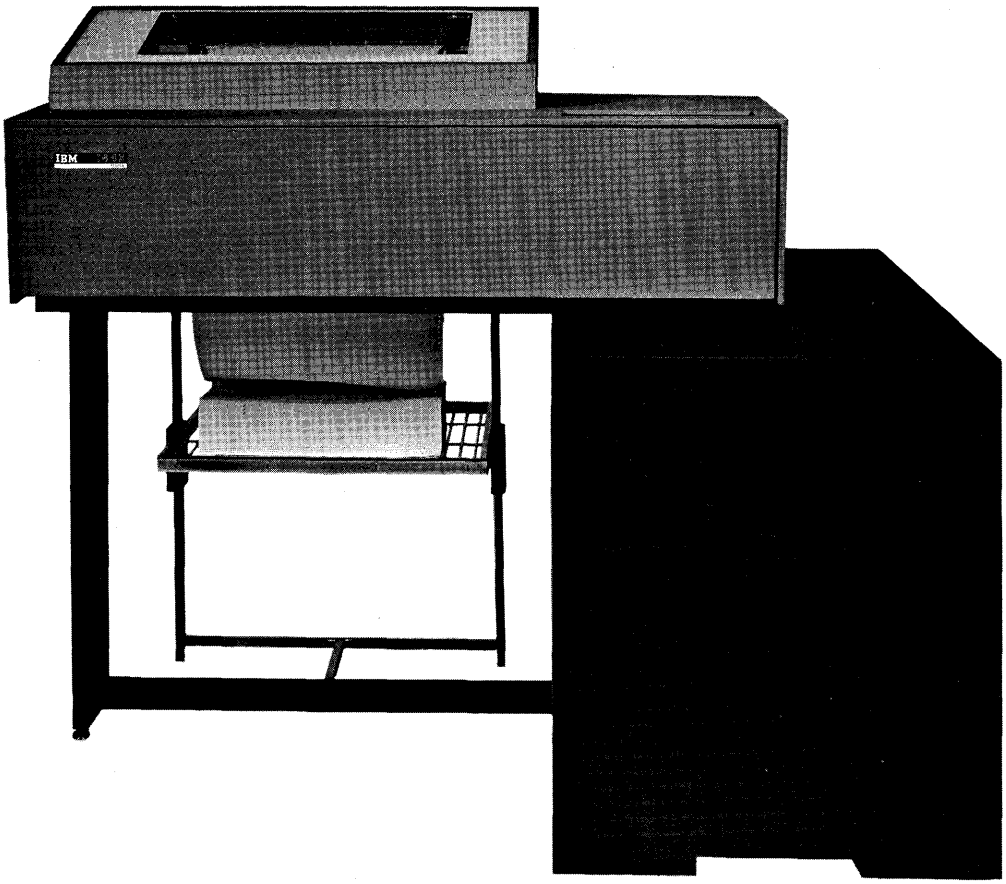


Figure 7. IBM 1443 Printer

A set of generalized programs is available for the 1620 Data Processing System. These programs are finished products that perform specific functions under the guidance of control information supplied by the user of the program. Included in the 1620 generalized program set are:

Flow Trace program
Selective Trace program
Hash Total program
Numerical Tape Duplicator and Corrector
Disk Utility Program for 1620/1710-1311
1620 Monitor I system

Available programming systems are:

1620/1710 Symbolic Programming System
1620 Symbolic Programming System
7090 Processor for Assembling 1620/1710 SPS Programs
1620 FORTRAN (with FORMAT)
1620 FORTRAN II
1620 COTRAN

These systems permit the programmer to represent machine locations and instructions by easily recognizable names and symbols. In addition, the need to preassign storage locations is eliminated.

The effect of all of the programs and programming systems is to provide the IBM 1620 user with a powerful set of tools. These tools enable him to adapt the system to his applications more easily and in less time.

Flow Trace Program

This program traces the logic flow of an object program during program testing by typing the instruction location and the address to which the branch has occurred for those instructions that actually result in a branch. This routine cannot be used on systems with the Indirect Addressing special feature; however, it may be used on a system with a minimum of 20,000 positions of storage and a 1621 Paper Tape Reader.

Selective Trace Program

The Selective Trace program aids in debugging an object program by typing out selected program steps and their results. The output consists of the instruction location, the instruction, and the fields affected. The status of the arithmetic indicators is typed on branch (B, BI, and BNI) instructions. This program is useful when an error continues to appear in an object program, and when desk checking and the Flow Trace program fail to locate the error. Under these conditions, the programmer may wish to execute his program and print out specific instructions and the fields affected. This program may be used to print the desired results under program control.

Hash Total Program

The Hash Total program is used to verify that a duplicated paper tape is an exact copy of its original paper tape. This check is made by taking an arithmetic "hash total" of all characters on a given paper tape.

Numerical Tape Duplicator and Corrector

This program is used to duplicate or correct 1620 paper tapes which contain only numerical records.

Disk Utility Program for IBM 1620/1710-1311

The Disk Utility Program (DUP) may be employed to implement the transfer of programs, routines, and data between the 1311-3 Disk Storage Drive and the 1620 Data Processing System. Designed to operate with or without the 1710 Executive Program, the Disk Utility Program permits the user to load and change programs, write addresses on the disk pack in an efficient manner, and read data into core storage and punch it in reloadable form. The user may assign the placement of programs on the disk pack or direct DUP to do so. If DUP assigns the programs, the locations are optimized, i.e., the best possible use is made of available disk storage areas. To prevent a cylinder overflow or an overlap of programs, the associated map is checked before the programs are placed on the disk pack. If an error occurs during any facet of these operations, the user is notified and allowed to specify the action to be taken. The Disk Utility Program is designed to operate on a 1620 system with 20,000 positions of core storage, a 1621 Paper Tape Reader and 1624 Tape Punch, or a 1622 Card Read-Punch and a 1311 Disk Storage Drive, Model 3.

The routines that make up the Disk Utility Program are:

Alter Sector Data. This routine allows the user to change data in disk storage from the typewriter. In most cases, only the digits to be changed need to be typed.

Update Programs. This routine implements the changes or additions necessary to update a program on disk storage.

Load Programs Optimally. This routine selects disk storage addresses so that all programs are loaded at optimum locations, no programs overlap each other, and no programs are split and stored on two cylinders.

Disk-to-Output. This routine dumps disk data on any one of the output units: card punch, paper tape punch, or console typewriter.

Move Programs. This routine transfers programs from one disk storage location to another. The user may specify both "from" and "to" locations.

Load Programs to Specified Addresses. This routine loads one or more programs on disk storage at a specified address or addresses and checks for both a cylinder overflow and an overlap of previously stored programs.

Update Maps. This routine facilitates the alteration of the maps that identify the programs on disk storage.

Write Addresses. This routine writes new addresses on the disk pack within the limits specified by the user.

1620 Monitor I System

The IBM 1620 Monitor I system is comprised of four separate programs:

Supervisor Program
Disk Utility Program
SPS II-D
FORTRAN II-D

The Supervisor program provides control for the IBM 1620 Monitor I system as well as for input/output routines. The nine routines provided with the Disk Utility Program are:

Write Addresses
Alter Sector
Disk-to-Output
Load Programs
Replace Programs
Move Programs
Disk-to-Disk
Delete Programs
Define Parameters

Monitor I, a disk-oriented system, allows the user to execute several programs without operator intervention. Jobs to be performed are stacked and separated by control cards that identify the jobs. Jobs may be processed in any order; i.e., FORTRAN compiling jobs, user jobs, and SPS assembly jobs may be entered as input in any sequence. Input may be from cards, paper tape, or typewriter.

Use of the Monitor I system reduces the amount of operator supervision time required of the programmer. The operator may perform the set-up operations for each job, independent of the programmer, because operating instructions are provided by Monitor Job cards. Substantial savings in set-up time are achieved when jobs are performed using the Monitor I system.

The Monitor I system operates on a 1620 system which has a minimum of 20,000 positions of core storage, a 1311 Disk Storage Drive, Model 3, and the special features, Indirect Addressing and Automatic Divide. Automatic Divide is required only for the execution of FORTRAN II-D and SPS II-D programs which use the Arithmetic and Functional subroutines.

1620/1710 Symbolic Programming System

SPS permits the programmer to code in a symbolic language that is more meaningful and easier to handle than numerical machine language. SPS automatically assigns and keeps a record of storage locations and checks for coding errors. By relieving the programmer of these burdensome tasks, SPS significantly reduces the amount of programming time and effort required.

The 1620/1710 Symbolic Programming System may be divided into the symbolic language used in writing a program, the library containing the subroutines and linkage instructions (macro-instructions) that may be incorporated into the program, and the processor program that is used to assemble the user's program.

Symbolic language is the notation used by the programmer to write (code) the program. This language provides the programmer with mnemonic operation codes, special characters, and other necessary symbols. The use of symbolic names (labels) makes a program independent of actual machine locations. Routines within a program can be written independently with no loss of efficiency in the final program. Symbolic instructions may be added or deleted without reassigning storage addresses.

Seventeen subroutines are available; they fall into three general categories: arithmetic, data transmission, and functional.

Arithmetic subroutines
Floating-Point Add
Floating-Point Subtract
Floating-Point Multiply
Floating-Point Divide
Fixed-Point Divide

Data transmission subroutines
Floating Shift Right
Floating Shift Left
Transmit Floating
Branch and Transmit Floating

Functional subroutines (those that evaluate)
Floating-Point Square Root
Floating-Point Sine
Floating-Point Cosine
Floating-Point Arctangent
Floating-Point Exponential (natural)
Floating-Point Exponential (base 10)
Floating-Point Logarithm (natural)
Floating-Point Logarithm (base 10)

The subroutines are provided in card or paper tape form for floating-point numbers with either a fixed-length or variable-length mantissa.

The macro-instructions that are written in a source program are commands to the processor to generate the necessary linkage instructions. Linkage instructions provide the path to a subroutine and a return path to the user's program. These subroutines may be any of the seventeen IBM Library subroutines or special subroutines prepared by the user. The ability to process macro-instructions simplifies programming and further reduces the time required to write a program.

The source program is punched into an input tape or into cards if the system is equipped with the IBM 1622 Card Read-Punch unit. The source program, after it is punched, together with the Library subroutines that are required, is assembled into a finished machine language program known as the "object program." The program is self-loading (contains its own loader program).

The SPS processor programs and the Library subroutines are available through your IBM sales representative. For distribution, the card and tape version of the 1620/1710 SPS are numbered 1620-SP-020 and 1620-SP-021, respectively.

1620 Symbolic Programming System

This program, an earlier version of SPS, contains many of the features of 1620/1710 SPS, but lacks some of its flexibility. Available subroutines include:

Arithmetic	Floating-Point Functional
Floating Add	Square Root
Floating Subtract	Sine
Floating Multiply	Cosine
Floating Divide	Arctangent
Divide	Exponential (natural)
	Exponential (base 10)
	Logarithm (natural)
	Logarithm (base 10)

The distribution numbers for this SPS program are:
1620-SP-008 paper tape version
1620-SP-009 card version

This programming system is designed to operate on a 1620 with a minimum of 20,000 positions of core storage. With minimum core storage, the Label table will accommodate up to 312 labels. The Label table can be enlarged on systems that have 40,000 or 60,000 positions of core storage.

7090 Processor for Assembling 1620/1710 Programs

1620 programs, written in the SPS language for the 1620/1710 processor, can be assembled by the 7090 processor. The 7090 processor (program number 1710-SP-001) has been designed to operate on a 709 or 7090 with a minimum of 32,768 storage positions, two data channels, and ten tape units. The processor runs under control of the IB SOS Monitor.

1620 FORTRAN (with FORMAT)

FORTRAN (FORmula TRANslation) is a language that allows the engineer and scientist to utilize a computer for problem solving with only a slight knowledge of the computer and a short period of training.

FORTRAN is written in a language that is a compromise between the language of the computer and the language of the engineer and scientist. To satisfy the computer, FORTRAN statements are converted to machine language. To satisfy the engineer and scientist, as many of the detailed computer control operations as possible are eliminated from the job of writing programs, and a problem statement format close to that of the mathematical equation is used.

The types of FORTRAN statements included in this version of FORTRAN are:

ACCEPT	IF
ACCEPT TAPE	IF (SENSE SWITCH)
CONTINUE	PAUSE
DIMENSION	PRINT
DO	PUNCH
END	PUNCH TAPE
FORMAT	READ
GO TO	TYPE
Computed GO TO	STOP

This FORTRAN compiler (1620-FO-003 paper tape version, 1620-FO-004 card version) operates on a 1620 with a minimum of 20,000 positions of core storage.

1620 FORTRAN II

FORTRAN II, an extension of FORTRAN (with FORMAT), has 23 different types of FORTRAN statements which offer added programming flexibility for the FORTRAN user. The FORTRAN II (1620-FO-019) system is designed to be used with an IBM 1620 equipped with Automatic Division and Indirect Addressing features, and to operate in conjunction with a 1622 Card Read-Punch and a 1623 Core Storage unit containing one 20,000-position module of core storage. To permit greater facility in the handling of data, a 1621 Paper Tape Reader and a 1624 Tape Punch may be added to the 1620 system.

1620 GOTRAN

The 1620 GOTRAN system uses a language that is a subset of the 1620 FORTRAN language. It converts GOTRAN statements to pseudo-instructions during the compilation phase and executes them interpretively during the execution phase. For this reason, the user is relieved of any consideration of the actual characteristics of the computer and thus is permitted to concentrate on the problem itself.

The GOTRAN language consists of thirteen different kinds of statements which may be grouped into four classifications: arithmetic statements, control statements, input/output statements, and a specification statement.

One great advantage of the GOTRAN system is that the compiler, once loaded, remains in the machine; it is an effective part of the computer until removed. Therefore, any number of different problems may be presented to the computer, singly or in batches, with solutions forthcoming at once. The time-consuming tedium of loading the system, creating the object deck or tape, loading the object program, etc., for each job, is eliminated. All that the user must be concerned with is the GOTRAN language because conversion to the interpretive language and subsequent execution of the instructions is entirely automatic. There is, naturally, a price exacted for this: that price is a somewhat reduced speed of operation as well as sacrifice of the portion of core storage that must be occupied by the compiler and interpreter.

The paper tape and card versions of GOTRAN are identified by the numbers 1620-PR-010 and 1620-PR-011, respectively. These programs are designed to operate on a 1620 system with a minimum of 20,000 position of core storage.

IBM 1620 Printed in U.S.A. A26-5718-0

IBM®
International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, New York