CPB 98A

GE 225

SYSTEM MANUAL

GENERAL ELECTRIC

GE 225

GE 225

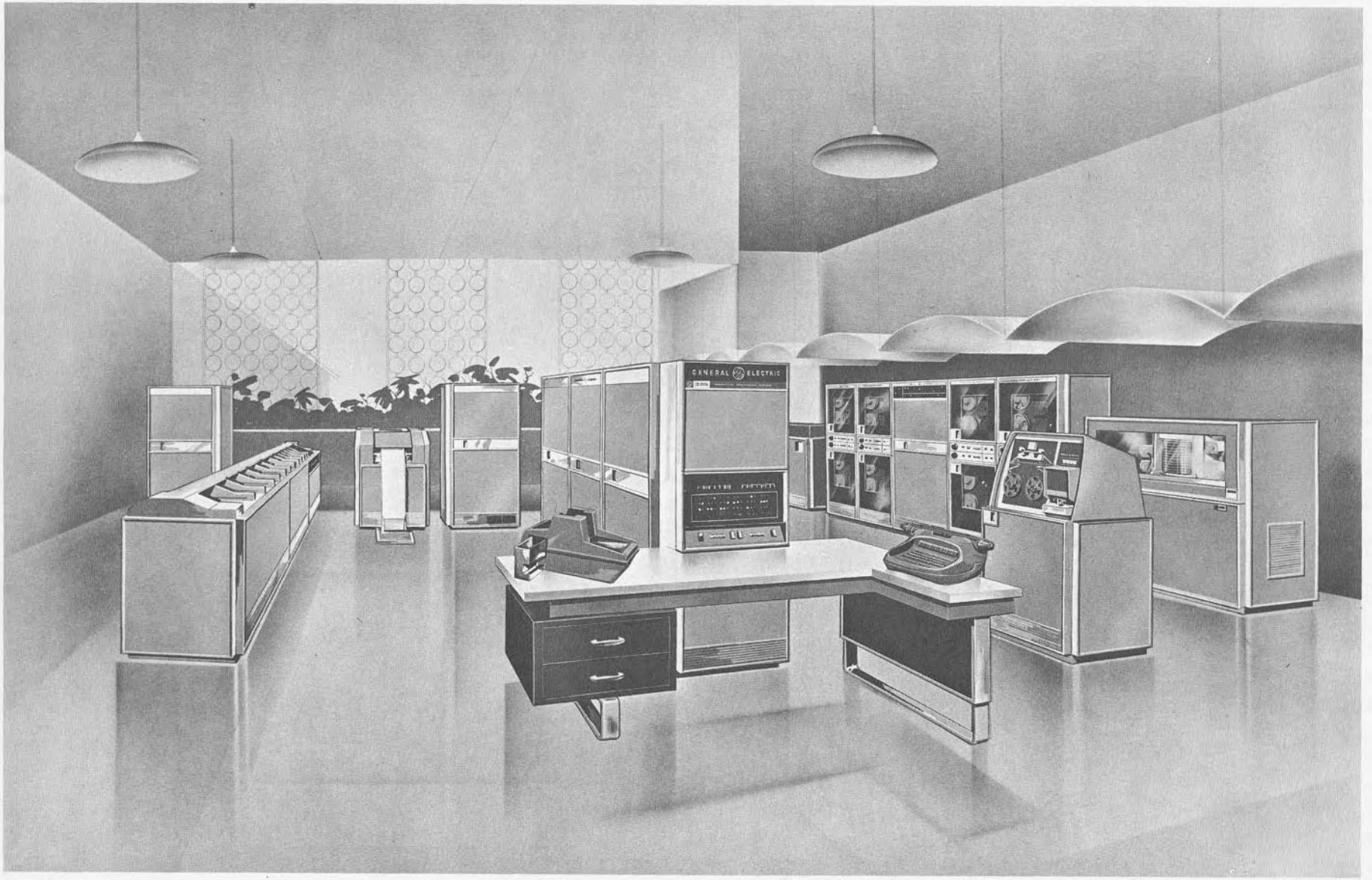
SYSTEM MANUAL

GENERAL ELECTRIC
COMPUTER DEPARTMENT • PHOENIX, ARIZONA

GENERAL ELECTRIC

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)



GE 225 Information Processing System

THE GENERAL ELECTRIC 225 INFORMATION PROCESSING SYSTEM

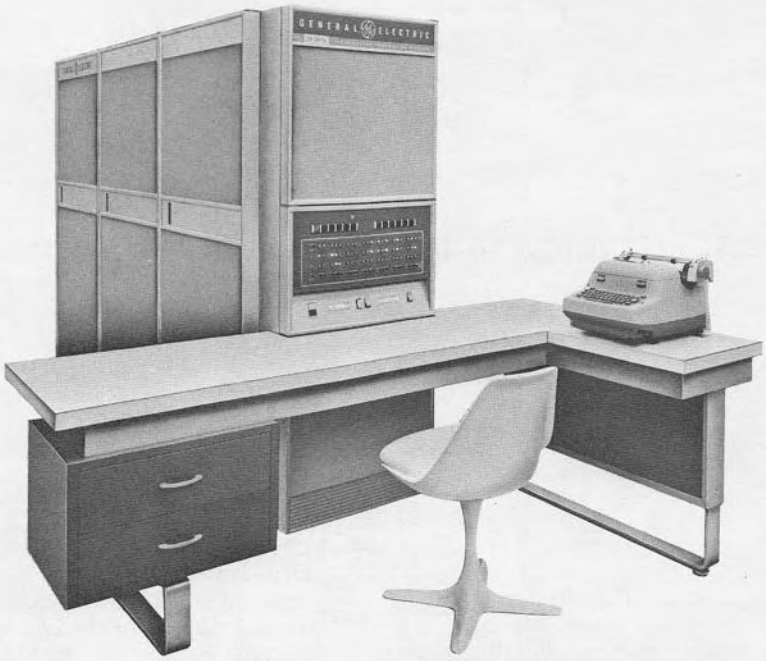
INTRODUCTION

The GE 225 is a new approach to the design of electronic data processing systems. Emphasis is placed on the total system concept and on flexibility of computer hardware organization as an answer to the increasing complexity of today's applications.

The total system concept considers a company to be an integrated system; the engineering, manufacturing, marketing and financial functions are not viewed as separate operations, but rather as parts of the overall operation. The data processing objective is to attain a balanced flow of information throughout the company according to the needs of each function.

To complement the new systems concept of the GE 225 hardware, General Electric is providing a General Compiler specifically designed to reduce significantly the traditionally high programming costs associated with computer installations. This compiler, which is equally applicable to business and scientific use, is an automatic coding technique that allows the program to be prepared with little or no knowledge of the intricacies or internal language of modern computers. The language of the General Compiler is one of convenience based upon English words and decimal numbers; the burden of translation is left to the computer.

Another programming facility is the General Assembly Program. It is desirable for those who wish to retain the individual instruction format in their programming effort. However, extensive clerical effort is eliminated through the use of significant mnemonic codes that are translated into computer language by the assembler. Addresses are assigned using either decimal or symbolic notations. Some advantages inherent in this method of programming are: ready reference between the machine program and the flow chart to aid in debugging; ease in the incorporation of future program modification; and efficient coding of special routines. The General Assembly Program includes the ability to incorporate many library routines such as input-output and mathematical packages.



CENTRAL PROCESSOR (showing Control Console and Console Typewriter)

- accepts and processes information from punched cards, magnetic tape, punched paper tape, magnetic ink character recognition equipment, mass random access memory, and other peripheral devices
- provides output on punched cards, magnetic tape, punched paper tape, high speed printer, and mass random access memory

CONTROL CONSOLE

- displays contents of the accumulator and other registers, as well as control and signal lights
- provides complete operator control and communication with the system

CONSOLE TYPEWRITER

- monitors system operations
- prints direct output from central processor
- prepares short summary reports
- gives operator instructions

GENERAL DESCRIPTION OF THE GE 225 SYSTEM

The GE 225 system meets requirements extending from conventional punched card arrangements to flexible systems using high speed input and output devices - it demonstrates high performance on business data processing, and engineering or scientific calculations.

Starting with an initial punched card arrangement, tailored to basic requirements, the GE 225 system may be easily expanded to meet growing data processing demands. Other equipment available includes high speed magnetic tape units, high speed printers, high speed perforated paper tape readers and punches, mass random access memories, and magnetic document handlers. In combination with paper tape input and output, console typewriter, and automatic floating point arithmetic, the GE 225 also becomes a powerful engineering and scientific computer because of fast internal operation.

The outstanding ability of the GE 225 to incorporate a variety of peripheral devices is accomplished by means of a common connecting device - the Controller Selector. An immediate advantage of this component is the interconnection of input and output devices, which results in simplified programming, installation, and maintenance. This unit permits the easy addition of peripheral equipment as the requirements of an installation grow, as well as additions of new or improved devices to the system with little, if any, logic or wiring changes.

Presented below are brief descriptions of the computer and its peripheral components to acquaint you with the capabilities of the GE 225 system.

The Central Processor is a transistorized, single address, general purpose digital computer.

The Memory of the GE 225 is composed of magnetic cores. The magnetic core memory furnishes an extremely fast and accurate method for storing data and for performing calculations. The unit of storage in computer memory is known as a "word". Available memory sizes are 4096, 8192, and 16,384 words. Each word in memory may be addressed individually and access time is identical for all locations.

A word in memory consists of 21 bits - 19 data bits, a sign bit, and a parity bit to insure accurate transference of information. Each magnetic core stores one bit of information. The GE 225 word represents information in either alphanumeric or binary form. Any word in memory may be addressed, transferred to a register where action is to be taken upon it, and regenerated in its original location in one 18 microsecond cycle.

The Console of the GE 225 provides direct manual and visual communication with the Central Processor. It displays the contents of:

The A register which serves as the accumulator where words are acted upon.

The I register which indicates the specific computer instruction being executed.

The P counter or register which indicates the memory address of the next instruction to be executed.

The Q register (which is an extension of the accumulator to contain double words) may be displayed by pressing a pushbutton which causes an exchange of the contents of the A and Q registers.

Twenty external control switches are provided, as well as other manual controls, which are discussed in detail in a later section. Registers and memory cells may be loaded or examined through the use of console controls.

The typewriter is another form of control. It prints messages to the computer operator under program control; and it is an on-line output device for obtaining answers to lengthy computations, for short reports, or for exception conditions. It types at a rate of 10 alphanumeric characters per second.

The GE 225 system peripheral equipment includes:

Card Readers and Card Punches. Card readers are on-line input devices. One transfers data to the computer at the rate of 1000 cards per minute and has dual reading; the second transfers data at the rate of 400 cards per minute. Normally, data are read from 80 column cards in either Hollerith alphanumeric code or in 10-row or 12-row column binary form. The card reader is an asynchronous device which is ready to feed as any time up to the maximum rate of speed. Thus reading rates are controlled by the program rather than by mechanical synchronization.

Punched cards may also be obtained as a result of computer output during processing operations. The Card Punches are on-line output units; one punches

cards at a speed of 250 cards per minute, and the second at the rate of 100 cards per minute. Information may be punched in Hollerith code or in 10-row or 12-row column binary code.

Both card reading and card punching can proceed concurrently with the operation of other peripheral equipment and the Central Processor.

The High Speed Printer is an output unit for applications where large quantities of data are developed for presentation in printed format. The High Speed Printer prints alphanumeric output, up to 120 characters per line, at the rate of 900 lines per minute. Printing format is governed by the Printer Controller, which contains logic circuitry to edit the print line independently of the Central Processor. Editing includes zero suppression, deletion of data, and insertion of special symbols, constants, and spaces. Many program steps are eliminated because of the editing ability of the GE 225 Printer Controller. Data to be printed and format masks are transferred from memory to the Printer Controller. After this transfer is complete the printer works off-line. Thus, under priority control, the printer operates concurrently with other peripheral units and uses less than 3 percent of available memory access time.

The Magnetic Tape is a fast method for storage or transmission of information. Transfer rates of the GE 225 system tapes are 15,000, 22,500, 41,662, and 62,500 characters per second. Some advantages of magnetic tapes are:

Millions of bits of data may be recorded on a single tape; thus magnetic tapes provide a compact storage medium.

Magnetic tapes may be used as on-line storage during a computer run, or as off-line storage to retain the mass of information usually required for subsequent related runs.

Magnetic tapes are erasable and may be used repeatedly.

The GE 225 system can employ one or more Tape Controllers, each of which directs from one to eight Tape Handlers, for the sequential storage, or retrieval, of massive files of information. With one Tape Controller, the tape units can read or write concurrently with the operation of other peripheral devices and the Central Processor. Multiple Tape Controllers permit Tape Handlers to read and write concurrently with the other operations. Up to eight Tape Controllers and 64 Tape Handlers may be connected to the GE 225 systems.

Another General Electric development is magnetic ink character recognition (MICR). This capability is extremely useful for banking institutions and for billing applications where re-entry into the system is required. To perform this type of data processing the General Electric Document Handlers are available. The 2-pocket and the 12-pocket Document Handlers are on-line or off-line devices that read and sort documents, printed with magnetic ink in E13B font, at a speed of 1200 items per minute. The documents may vary in quality, size, and, if damaged, degree of mutilation. The Document Handler may be used on-line as a document sorter, and it is possible to operate at least two sorters concurrently, A control unit permits concurrent operation with other peripheral units and the Central Processor.

The Document Handler recognizes 14 characters: the ten decimal digits and four special symbols called Cue characters. The four Cue characters are used to separate fields of decimal digits; for example, dollar amount fields and identification fields.

Mass Random Access File Memory units are available to process transactions as they occur. This eliminates the need to search through unrelated information as is the case in the sequential processing required by punched card and magnetic tape files. Each word is stored as an image of memory on the surface of discs. Record lengths are 64 words, which are equal to 192 alphanumeric characters or 352 binary represented digits. Data are recorded serially in 256 circular tracks on each side of the disc. There are sixteen 64-word records in each of 128 outer tracts, and eight 64-word records in each of 128 inner tracts. Data transfer rates are 500,000 bits per second and 250,000 bits per second for the outer and inner tracts respectively. Mass Random Access File Memory is available in two sizes: a 64-disc file having a capacity of 393,216 records; and a 16-disc file having a capacity of 98,304 records.

Perforated Paper Tape is another data storage and communication medium which can be used with the GE 225 system for on-line input or output. Transactions are commonly recorded on paper tape as a by-product of machines such as typewriters, billing machines, data transmission lines, and calculators. Paper tape is also widely used as an input medium in the scientific field. The Paper Tape Reader can read either 250 or 1000 characters per second. The Paper Tape Punch punches at the rate of 110 characters per second.

The optional Auxiliary Arithmetic Unit is an advantage when many double word or floating point arithmetic calculations are required. This device, with built-in logic circuits, facilitates floating point arithmetic through its increased capacity and the speed with which it computes. The Auxiliary Arithmetic Unit contains two 40-bit registers that perform addition, subtraction, multiplication, and division. At programmer option the unit functions in three modes; namely, normalized and unnormalized floating point calculations, and fixed point computations. All other computer operations, including normal arithmetic performed in the regular arithmetic unit, may occur concurrently with this device.

THE CONTROLLER SELECTOR

An outstanding aspect of the GE 225 system is the Controller Selector. It permits the extreme flexibility of computer hardware organization. The basic equipment that tie in directly to the Central Processor are the typewriter, the card punch and the card reader, and the paper tape punch and paper tape reader. The Controller Selector is a common control and data transfer point between the Central Processor and all other peripheral equipment.

The Controller Selector contains eight channels, ranging in value from 0 through 7. These channels are assigned priority for access to or from memory, 0 is the highest priority and 7 is the lowest priority.

A peripheral controller is given an address, and thus a priority, by easily inserted plug-in switches that are preset to one of the values of 0 through 7.

Any configuration of eight peripheral controllers may be connected to the Controller Selector - through convenient plug-in connectors - and an automatic system of peripheral priority is established without programming. The eight channels facilitate the exchange of various peripherals within a system, or from one system to another.

After a peripheral controller is selected and receives all the necessary instructions from memory, it operates independently of the computer except when it requires memory access to enter or receive information. The Controller Selector permits time sharing, on an automatic basis, of associated peripheral devices with internal computations.

Information is transferred through the Controller Selector at the rate of 55,000 words per second.

SUMMARY OF PERIPHERAL EQUIPMENT SPEEDS

ON - LINE INPUT

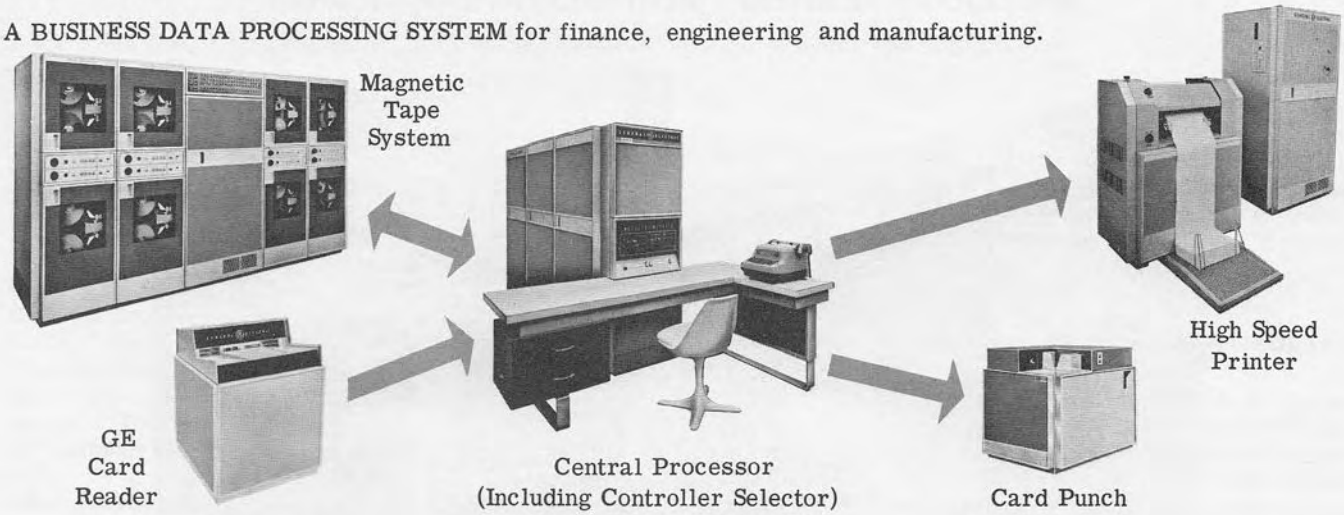
CARD READER	1000 or 400 cards per minute
MAGNETIC TAPE	15,000, 22,500, 41,700, or 62,550 characters per second
PAPER TAPE	1000 or 250 characters per second
MASS RANDOM ACCESS MEMORY	Outer Track: 500,000 bits per second Inner Track: 250,000 bits per second 133 millisecond average seek time and 25 millisecond average latency time.
DOCUMENT HANDLER	1200 items per minute

ON - LINE OUTPUT

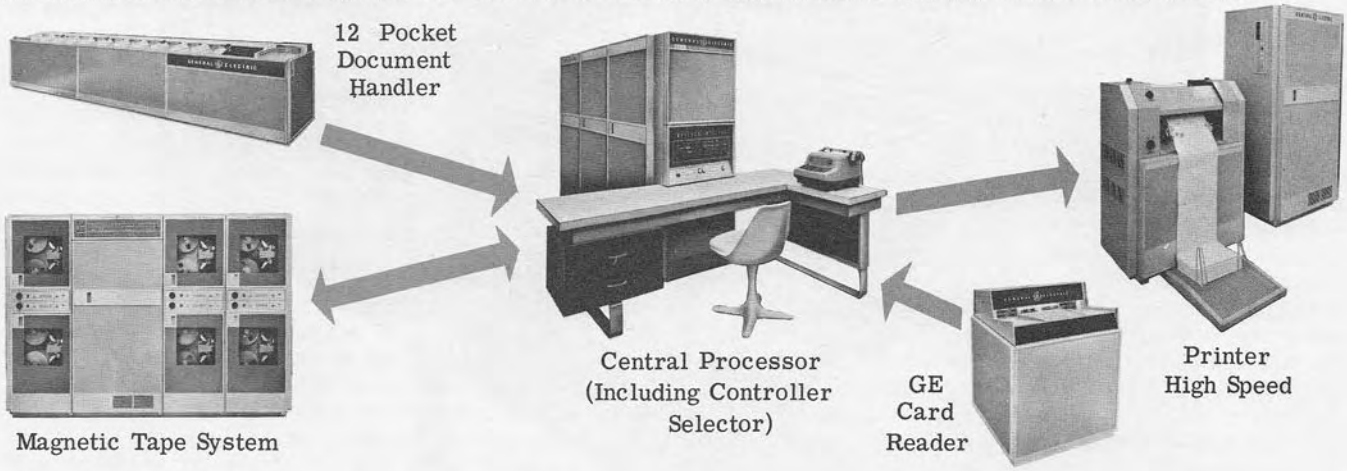
CARD PUNCH	250 or 100 cards per minute
CONSOLE TYPEWRITER	10 characters per second
HIGH SPEED PRINTER	900 lines per minute (120 characters per line)
MAGNETIC TAPE	15,000, 22,500, 41,662, or 62,550 characters per second
PAPER TAPE	110 characters per second
MASS RANDOM ACCESS MEMORY	Outer Track: 500,000 bits per second Inner Track: 250,000 bits per second Average seek time: 133 milliseconds

TYPICAL GE 225 SYSTEM CONFIGURATIONS

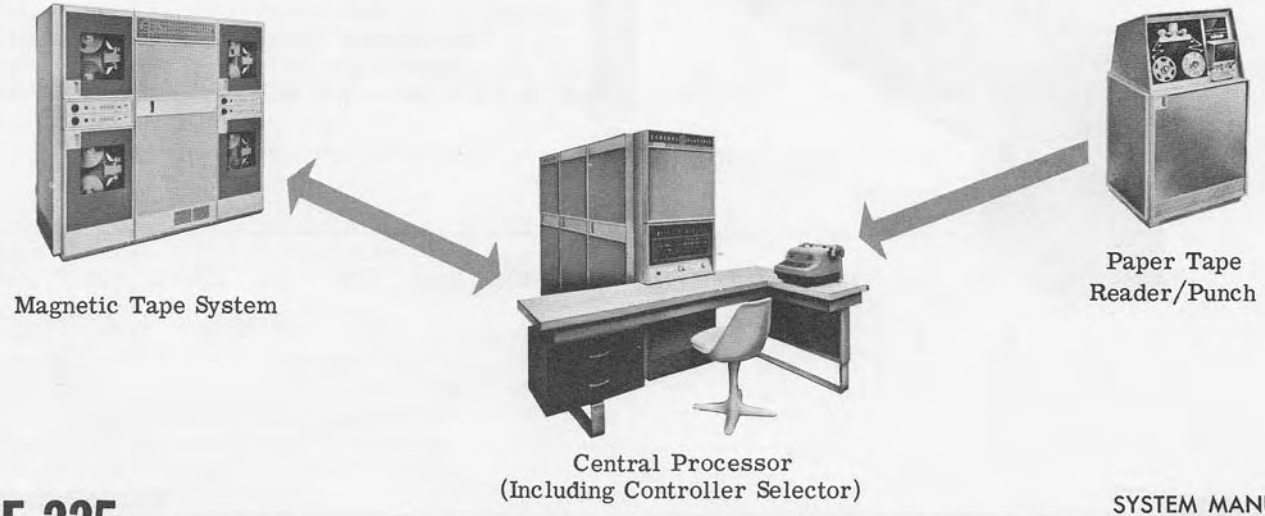
A BUSINESS DATA PROCESSING SYSTEM for finance, engineering and manufacturing.

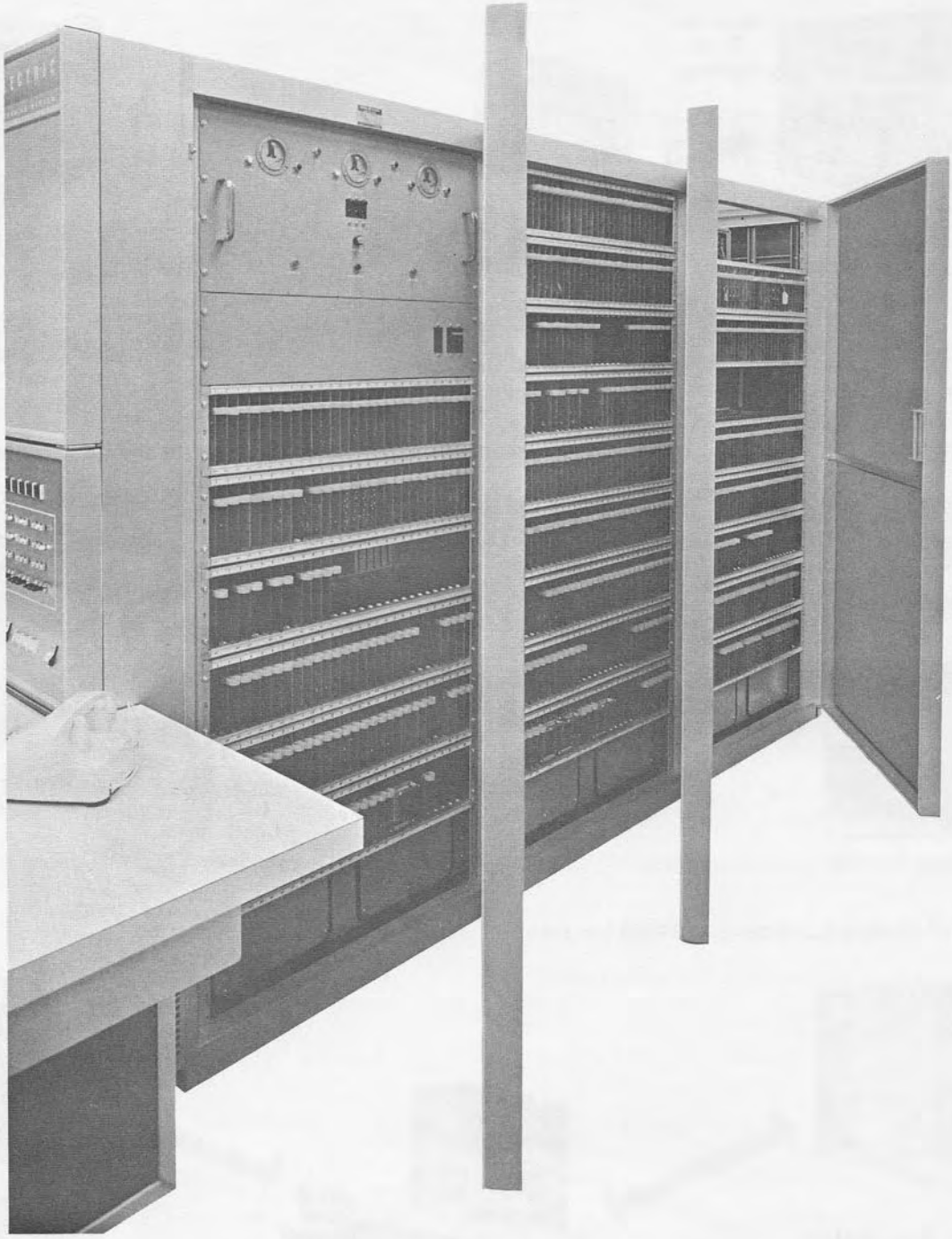


A BUSINESS PAPER PROCESSING SYSTEM for all business operations using paper as source documents.



A SCIENTIFIC COMPUTATIONAL SYSTEM for research development and data reduction.





FUNCTIONAL DESCRIPTION – CENTRAL PROCESSOR

MAGNETIC CORE MEMORY

Both computer instructions and alphanumeric or binary data to be processed may be stored in the memory of the GE 225 Central Processor. This storage capacity enables the complete program to be read into memory when a computer run is to be made. The program then directs the operations of the computer - it calls for data, stores it, calculates and stores results, and arranges data for output. Thus, the entire process is contained within computer memory.

The memory of the GE 225 is composed of magnetic cores; each core contains one piece of information referred to as a bit. A 4,096 word memory is composed of 86,016 cores; a 16,384 word memory contains 344,064 cores. Memory sizes of 4,096, 8,192, and 16,384 words are available.

All information exists in the GE 225 in the form of words. Each word consists of 20 data bits. A word stored in memory. however, has 21 bits which include the parity bit for the purpose of accuracy. Every memory word, or location, may be individually addressed.

Access time, including restoration of the word, is 18 microseconds (millionths of a second). Single word transfers (20 bits) to or from memory, including instruction access time, are accomplished in 36 microseconds; double word transfers (two adjacent 20 bit words) are made in 54 microseconds. The transfer of words to and from memory are made in one-word parallel form; that is, the word bits are transferred simultaneously.

Internal checking is accomplished by the generation and storage of a parity bit when a word is transferred to memory, and the recomputation and verification of that parity bit when the word is read from memory. The effect of a parity error depends upon the setting of a switch on the console; a computer halt or a programmed branch for remedial action may occur at operator option if a parity error should occur.

REPRESENTATION OF INFORMATION IN MEMORY

The GE 225 is a binary computer which provides high speed and versatile command structure; it is capable of processing data in either binary or alphanumeric form. This feature permits both modes of operation to take advantage of the particular characteristics of a given application. Subroutines are provided to transform the small percentage of data required, or desired, to be converted from alphanumeric to binary, or from binary to alphanumeric representation.

ALPHANUMERIC DATA WORDS

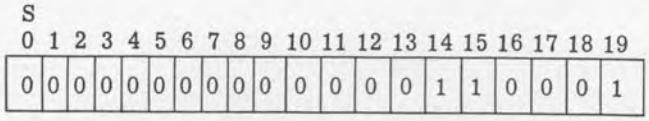
When cards punched in Hollerith code are used as computer input, each alphanumeric symbol is automatically converted into a six-bit binary coded decimal character. Thus, three alphanumeric characters occupy 18 of the 20 bit positions of an alphanumeric data word. (The Appendix contains an equivalent table of Hollerith and binary coded decimal codes). Double word separations permit the automatic handling of six alphanumeric characters with one command. These convenient word sizes eliminate the need for elaborate partial word facility. The 36 alphanumeric and 14 special characters are expressed by the six-bit binary coded decimal configurations.

The table below shows how characters are represented in a six-bit format:

	NUMERIC BITS													
Zone Bit	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1011	1100	1101	1110
00	0	1	2	3	4	5	6	7	8	9	#	@	_	=
01	+	A	B	C	D	E	F	G	H	I	.			
10	-	J	K	L	M	N	O	P	Q	R	$	*		
11	Space	/	S	T	U	V	W	X	Y	Z	,	%	[	]

BINARY DATA WORDS

A binary data word consists of 19 binary bits plus the sign bit. For example, the decimal number 49 is represented in binary form as:



Bit 0 is the sign bit: 0 or 1 indicate plus or minus respectively. Special instructions are provided to process double data words; that is, 38 bits plus the sign.

The table in the Appendix shows that a 20-bit binary word will accommodate a range of positive or negative decimal values from 0 to 524,287. Double word operations permit the efficient processing of positive or negative values from 0 to 274,877,906,943.

ARITHMETIC CAPABILITY

The GE 225 processes data in both binary and alphanumeric forms. The computer normally operates arithmetically in the binary mode. Note, however, that at programmer option, addition and subtraction may be performed in the decimal* mode.

Floating point arithmetic is easily processed in the GE 225 by a utility subroutine provided to perform this type of arithmetic. Voluminous floating point calculations are effectively accomplished by the optional Auxiliary Arithmetic Unit (see page 39) because of its speed, capacity, and logic circuitry.

DATA CONVERSION

Subroutine packages are provided to make the necessary conversion if it is desired to do arithmetic in the binary mode and to work in the binary coded decimal mode (hereafter referred to as BCD) before and after binary arithmetic operations. Specifically, a conversion is necessary to print binary numbers in decimal form on the console typewriter or the high speed printer. Output to punched cards or magnetic tape may be in either binary or BCD.

When information is put out in binary form, the intention is to re-enter the information as binary input to a subsequent program. Thus, the necessity for input and output conversion is eliminated. For instance, portions of a master record (keys and other data) may be in BCD, and other portions (quantities and prices) may remain in binary format. Therefore, transactions may be processed without conversion of the master record. Further, should the transaction input be, for instance, Hollerith punched cards, only the updating quantity need be converted to binary.

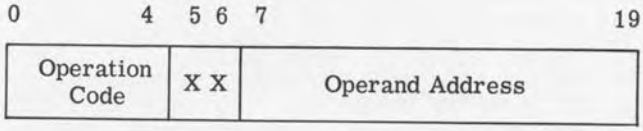
The updated master record may then be written in the binary mode, with specific data in either BCD or binary format, and all data are immediately available for a subsequent run.

INSTRUCTION WORDS

An instruction word in the GE 225 is a single address word consisting of 20 bits. Except for branching operations, instructions are executed sequentially; the reading of the next instruction from memory occurs after the execution of the current instructions. A sequence control counter, the P register, contains the address of the next instruction to be executed. The contents of the P register are displayed on the control console.

The basic format of the instruction word is:

OR



Bits 0 through 4 designate the operation to be performed, bits 5 and 6 indicate whether the instruction is to be automatically modified, and bits 7 through 19 indicate the operand address.

Machine running programs are not prepared in binary notation. They are easily generated by means of the General Compiler, which requires only English language or decimal notation, or the General Assembly Program which recognizes flexible mnemonic codes. The latter program, as well as the compiler, translates the codes into the absolute language of the computer. For example, the addition instruction is expressed as ADD - this mnemonic is translated into the machine language executed by the computer. (The General Compiler and the General Assembly Program are discussed in more detail in the section entitled "Programming Aids").

Since the five-bit positions allowed for the operation code define a maximum of 32 operations, or commands, it is clear that more bit positions are

* This feature is part of an optional unit that provides for decimal addition and subtraction, additional modification groups, and a three-way compare instruction.

required to define all of the operations in the instruction repertoire of the GE 225. This is achieved through the use of bit positions in the operand address field for instructions that require only a limited portion of the field. Examples are shift commands that require only bit positions 15 through 19 to indicate the length of shift; and instructions which do not have an operand address such as word transfers between registers.

AUTOMATIC ADDRESS MODIFICATION

The GE 225 is capable of automatic address modification under program control. This is achieved through the use of special modification words located in memory at 0001, 0002, and 0003.

Bit positions 5 and 6 of an instruction word designate the modification word to be used in revising the operand address portion of the instructions. Binary configurations select these words as follows: 01 = location 0001, 10 = location 0002, and 11 = 0003. Bits 00 in positions 5 and 6 indicate that no modification is to be performed. The programmer, when using the General Assembly Program, merely writes 1, 2, or 3 to select the desired location. The Instruction Repertoire section of this manual indicates instructions that may be automatically modified and those which may not.

If an instruction in the I (instruction) register calls for automatic address modification, an extra word time - 18 microseconds - is required to accomplish the operation. When an instruction word comes from memory to the I register the modification bits are tested to determine whether revision is indicated. If bit positions 5 and 6 are other than 00 the address is modified. The address portion (in this case, bits 5-19) of the selected modification word is sent through the B register to the arithmetic unit; bits 7 through 19 of the I register (operand address bits) are also sent to the arithmetic unit where the address portion of the selected modification word is added to them, and a modified address (bits 5-19) is developed. The revised address is then put in the instruction register (bits 5-19) and the instruction is executed.

To summarize, the results of an automatic address modification are:

the command position (bits 0 through 4) of the word in the I register is unchanged;

bits 5 - 19 of the modification word and bits 7 - 19 of the command word are added;

the result replaces bits 5 - 19 of the I register and is executed as the instruction address.

However, the original instruction in storage is not changed.

Further, memory location 0000 (as well as 0001, 0002, 0003) may be conveniently used as a counter. These locations are incremented by a constant and tested by one of two special branch instructions; desired routes are followed at programmer option. (See instructions INX, BXH, and BXL in the Instruction Repertoire).

Modification words facilitate addressing memory above location 8191 (upper memory). However, if the program is executing commands in upper memory, the sequence register remains set for upper memory and is incremented in the normal manner without the need for modification words.

Thirty-one additional modification word groups* are available; each group contains four words. The three highest words in each group may be used to modify other words. All words in each group may be used as counters.

The total of 32 modification word groups, if included in the system, occupy memory locations 0000 through 0127. Any of these locations may be used for normal storage if not required for address modification in a program. A command is provided to select any groups at any point in the program.

ARITHMETIC AND CONTROL REGISTERS

A Register

The A register is a 20-bit register that serves as the accumulator for the central processor. It performs this function by holding:

The augend during addition.

The sum after addition.

The minuend during subtraction.

The result after subtraction.

The most significant half of the product after multiplication.

The most significant half of the dividend before division.

The quotient after division.

The most significant half of a word after the execution of all double length instructions.

A word transferred from memory or to be transferred to memory.

* This feature is part of an optional unit that provides for decimal addition and subtraction, additional modification groups, and a three-way compare instruction.

The word on which extraction is performed during the execution of the extract instruction.

The word to be shifted during various shift instructions.

A word to be transferred to another register or to be modified in some way during the execution of various data transfer commands.

The word that determines future action during the execution of branch instructions.

Q Register

The Q register is a 20-bit register which acts with the A register to form a 38-bit (plus a sign bit) accumulator during the execution of double word operations. Information is transferred from memory into the Q register through the A register. When double word instructions are used, the least significant half of the double word is automatically transferred into register Q through register A.

The Q register performs the following functions:

After execution of the double word loading operation, it holds the least significant half of the double word read from memory.

During a double word store operation it contains the least significant half of the double word to be transferred to memory.

After multiplication Q holds the least significant half of the result.

During division it holds the least significant half of the dividend.

After division Q holds the remainder (it has the sign of the dividend).

Q contains the least significant half of the augend during double word addition, and the least significant half of the minuend during double word subtraction.

During double shift instructions, it holds the least significant half of the information to be shifted.

In addition, the Q register works in conjunction with the N and A registers in special shift operations.

N Register

The N register is a six-bit register used as a single character buffer between the computer and the typewriter, the paper tape reader, or the paper tape punch. Information is transferred directly between the N register and the A register.

I Register

The I register is the instruction register. It contains the 20 bits of an instruction word during execution of a computer command. While instructions are being processed, bits 0 through 4 indicate the operation to be performed, and bits 5 and 6 control the automatic modification of instructions when specified by the program.

During the execution of instructions involving reading an operand from memory, bits 7 through 19 contain the memory location of the operand. These thirteen bits may have various meanings during execution of other types of instructions.

P Register

The P register is the location counter that contains the memory address of the next instruction to be executed. The P register is incremented by one before the execution of an instruction so that it indicates the next instruction in sequence.

The contents of the P register can be set from the I register when "unconditional branching" is specified by the program.

The 15 bits of the next instruction address are indicated by 15 display lights on the control console.

B Register

The B register is a 20-bit register which performs several functions:

It acts as a buffer register between the M register and the computer during data transfers.

It is a buffer for arithmetic operation and contains:

The addend for addition
The subtrahend for subtraction
The miltiplicand for multiplication, and,
the divisor during division.

It is used in the execution of certain data transfer commands.

M Register

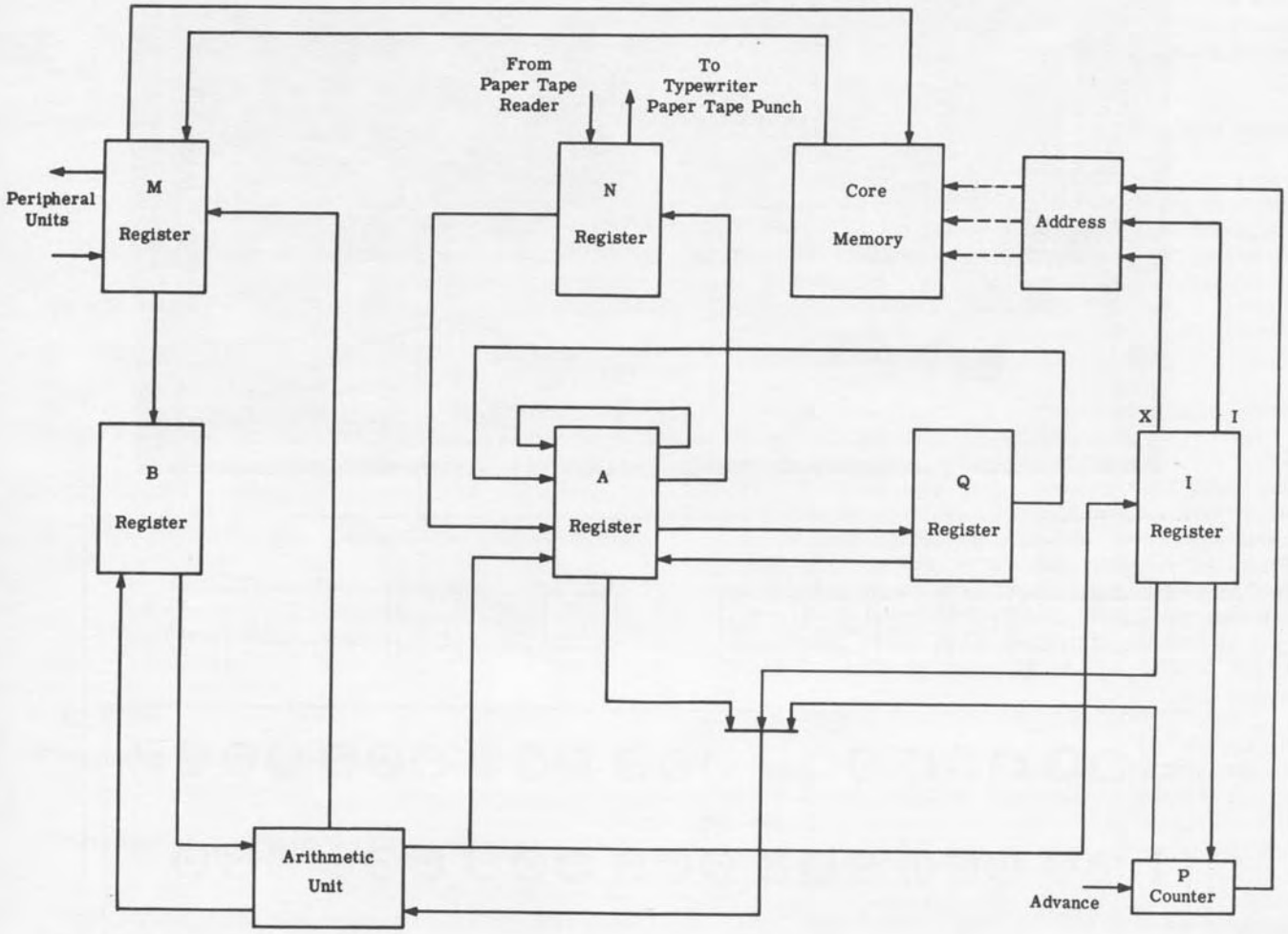
The M register is a 21-bit register and acts as a buffer between the magnetic core memory and the B register, and also, as a buffer to peripheral components. The 21 bits in the M register include 20 data bits plus a parity bit.

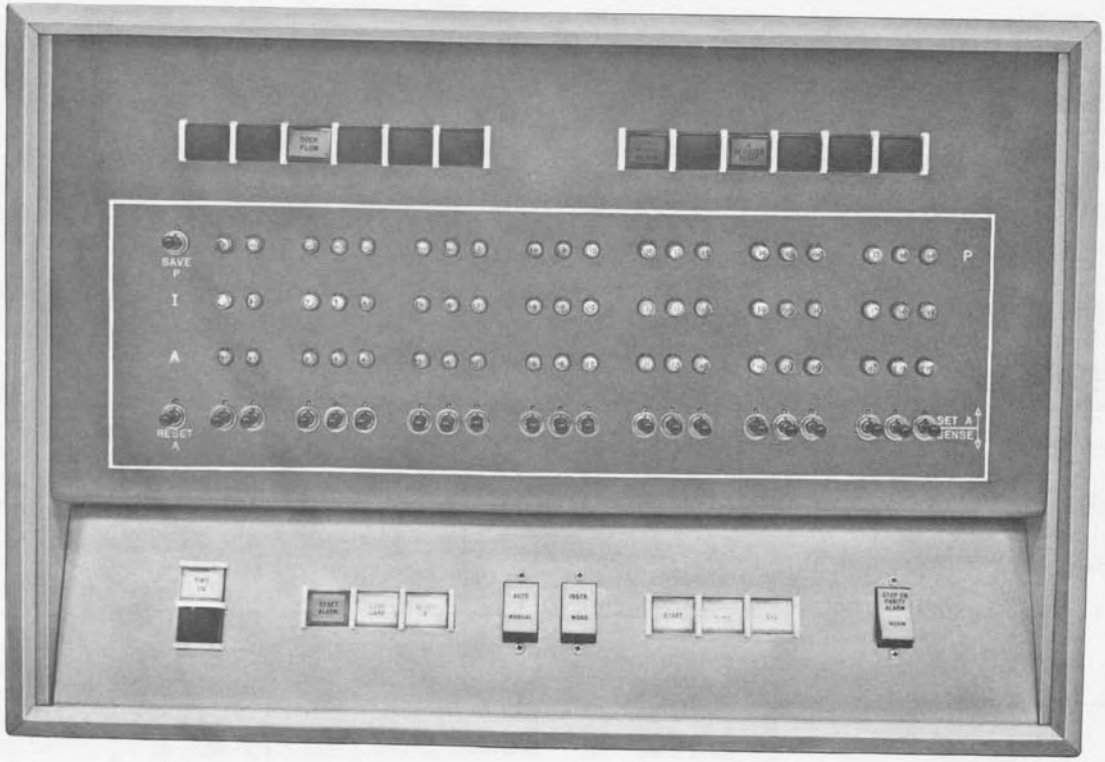
When a word enters the M register in preparation for recording in memory, a parity bit is computed and 21 bits are stored in memory. When a word is read from memory into the M register, parity is again computed and the new parity bit is compared with the one already existing to insure accuracy of data transfers.

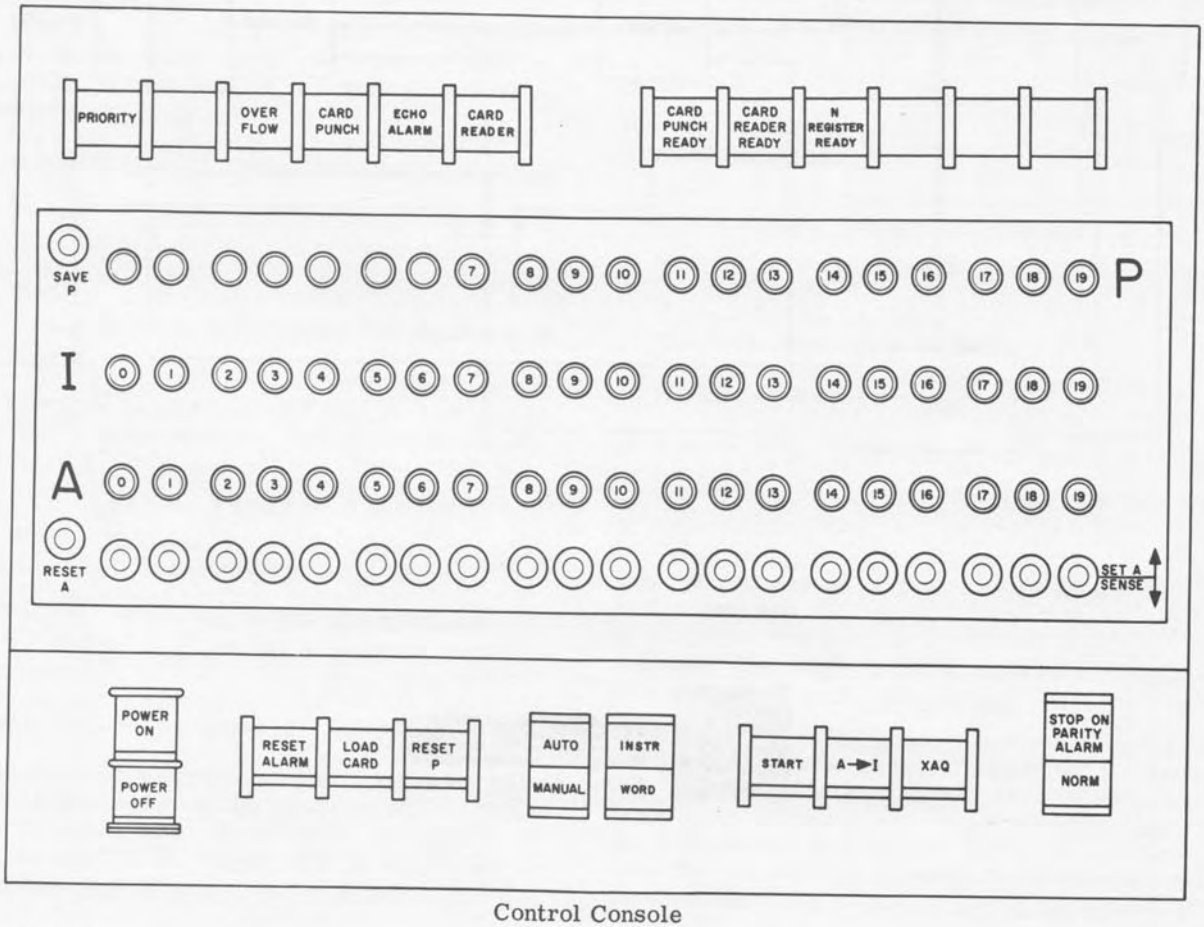
Arithmetic Unit

The Arithmetic Unit serves two functions. During arithmetic operations it executes the calculations specified by the command code in the I register. It also acts as a transfer bus for data words going to the A register and for instruction words going to the I register.

SCHEMATIC DIAGRAM OF REGISTER RELATIONSHIPS







Control Console

FUNCTIONAL DESCRIPTION - CONTROL CONSOLE

THE CONTROL CONSOLE

The purpose of the console is to provide a control center for the computer operator from which he has visual representation and manual control of operation of the system.

Manual control involves the initial reading of the program into memory, starting the execution of the program, and if desired, occasionally stopping the program for checking or other purposes. Switches on the console permit the manual entry of information into the Central Processor.

The indicator panel consists of display lights of various registers and alarms. It occupies the upper two-thirds of the control console. The control panel occupies the lower third.

Indicator Panel

1. Three registers are displayed by lights on the indicator panel:

   The fifteen-bit P counter.
   The twenty-bit I register.
   The twenty-bit A register.

2. Switches

   The Save P switch causes the contents of the P counter to be retained.

   The Reset A switch clears the A register.

   Twenty switches are provided to place any bit configuration into the A register. They may also be used as "test" switches to alter the course of a program.

3. Ready Lights

   The Card Reader Ready light indicates the card reading equipment is ready to operate.

   The Card Punch Ready light indicates the card punch equipment is ready to operate.

   The N Register Ready light indicates that the N register is available for input-output.

4. Alarm Lights

   The Alarm Lights notify the operator of abnormal conditions occurring during operation.

   The Priority alarm light indicates that the Central Processor has lost priority. (See section on Priority Interrupt - Internal Control of access to memory.)

   The Parity alarm light indicates a parity error.

   The Overflow alarm light indicates overflow in the Arithmetic Unit (i.e., its capacity has been exceeded) or in the A register as the result of a shift left instruction shifting a "1" bit out of position one.

   The Card Punch alarm light indicates an error involving the card punch equipment.

   The Card Reader alarm light indicates an error involving the card reading equipment.

   The Echo alarm light indicates that peripheral controllers operating through the Controller Selector are unable to respond when addressed. This may be due to the fact that the peripheral controller is not connected to the proper address or it is in an off-line status. This light may also appear if there is a peripheral controller malfunction, or if its power is off.

Control Panel

The push buttons and switches, constituting the Control Panel, furnish the manual control for the system.

1. Push Buttons

   The two left-most push buttons are the computer Power On and Power Off controls.

   The Reset Alarm push button resets all alarms.

   The Load Card push button allows one load card to be sent through the card mechanism, and its contents are read in 10-row binary mode into memory starting at location 0000. The load card button offers a method of initiating the first program card through the card reading mechanism. The machine can then branch to 0000 for the first instruction.

   The Reset P push button clears the P Counter.

   The Step push button allows step-by-step operation of the computer when the Automatic-Manual

switch to the left of it is in the manual position. Otherwise it starts the computer into automatic operation.

The A→I push button transfers the contents of the A register into the I register. An instruction can be set up in the A register by means of the twenty switches available and then transferred to the I register.

The XAQ push button causes an exchange of information between the A and Q registers; the contents of A go into Q and the contents of Q go into A.

2. Switches

When the Automatic-Manual switch is in the Automatic position, the computer executes instructions in the normal sequential manner. Placing this switch in the manual position while the computer is operating will bring it to a halt after completion of the instruction being executed. When the switch is in the manual position, the computer executes instructions in a step-by-step procedure, going from one step to the next each time the Step push button is pushed. The exact step-by-step procedure followed in the manual mode of operation is determined by position of the Word-Instruction switch discussed below.

When the Word-Instruction switch is in the Instruction position, one instruction is executed each time the Start button is depressed. When the switch is in the Word position, the computer does not step to the next instruction but to the next operation in sequence.

The right-most switch is concerned with parity alarm. It has two positions: (1) Stop On Parity Alarm which stops the computer on a parity error developing from memory transfer or input and (2) Normal which does not stop the computer when there is a parity error. The switch is placed in the Normal position when the program being run has been prepared to take remedial action when any parity error occurs. This permits processing to continue without interruption, or it may be halted under program control. (The instruction for testing the Controller Selector permits interrogation of peripheral conditions which include parity errors.)

CONSOLE TYPEWRITER

The console electric typewriter receives output from the Central Processor through the N register and prints in upper case type style. It types alphanumeric characters, 5 special symbols, and spaces at the rate of ten characters per second.

Characters that may be printed are:

The letters A through Z
The digits 0 through 9
- Minus sign or hyphen
/ Slash
. Period
$ Dollar sign
, Comma
Blank or space

Functions that may be programmed are:

Print black
Print red
Tab
Carriage return

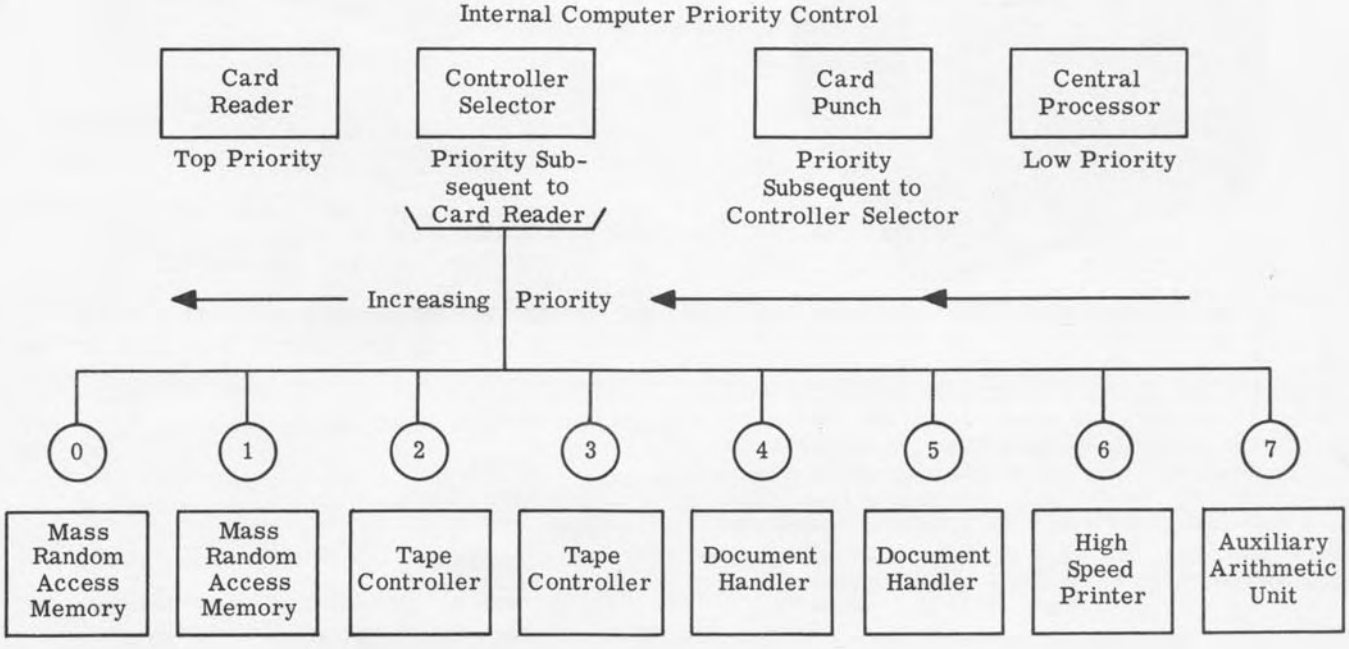
PRIORITY CONTROL OF ACCESS TO MEMORY

Due to the unique design concept of the 225 System, the core memory serves the dual function of main memory and input-output buffer. Thus, two or more operations may be performed concurrently; for example, cards may be read, and tapes may be read and written, while computing continues.

If the Central Processor requests memory access simultaneously with input or output devices currently in operation, it will obtain access at the next free cycle.

Since several requests for memory access might be made at the same time, a provision is made to grant only one request for access during a word time. The analysis of these various requests for memory access and the determination of whether an input-output device or the Central Processor should have access to memory is performed by the Priority Control logic incorporated in the Central Processor.

The system of priority is illustrated in the block diagram below.



Two levels of priority are involved: Internal Priority Control and Controller Selector Priority. Internal Priority Control determines whether priority is given to the card reader, the Controller Selector, the card punch, or the computer. The Controller Selector is treated as an input-output device having a higher priority than the card punch but a lower priority than the card reader. The computer has the lowest priority because no loss of information can result if it ever waits for memory access.

The Controller Selector Priority determines the specific peripheral unit to be granted memory access when the Controller Selector itself has access. Which of the possible eight controllers connected to the Controller Selector function is granted access depends upon the priority address assigned to it; address 0 has the highest priority and address 7 the lowest.

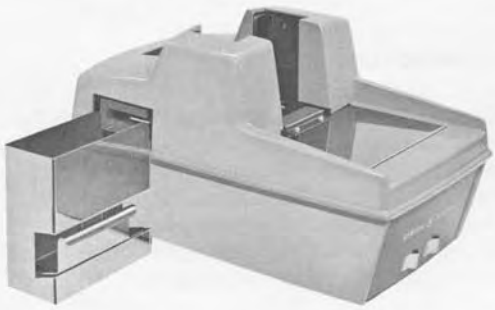
The priority assigned to peripheral units connected through the Controller Selector should be consistent with the repetition rate of pulses (information) going to or coming from the input-output device. Thus, if a request for access is received from two input-output devices simultaneously, the one with the higher repetition rate has top priority and is the first to be granted access. The other device will wait for at least one word time. The reasoning behind this basis for priority assignment is that the slower unit can wait without danger of loss of information. The faster unit cannot wait as long because additional information will soon follow. A magnetic tape controller, for example, should have higher priority (lower priority address) than a printer controller. Once a magnetic tape controller has initiated tape movement, the tape controller must have all the memory accesses it requires to function properly.

FUNCTIONAL DESCRIPTION – PERIPHERAL EQUIPMENT

CARD READING EQUIPMENT

With the GE 225 system it is possible to use either of two card readers. One unit reads at a rate of 400 cards per minute; the input and output hoppers of this reader have capacities of 600 cards. The high speed card input device reads at a rate of 1000 cards per minute and has dual reading equipment; the input and output hoppers of this reader have capacities of 2000 cards. These components read standard 7-3/8" X 3-1/4" 12-row cards.

The Card Reader is an asynchronous device which is ready to feed at any time up to the maximum rate of speed. Thus, reading rates are controlled by



the program rather than by mechanical synchronization.

Cards are read column by column and may be fed card by card in all modes or continuously in 10-row binary or BCD at the rates stated. Provision is made for both manual and computer programmed stop-start. Checking features include timing and optical equipment checks, and card feed error detection.

Two card formats may be used with the GE 225 readers:

1. Alphanumeric (Decimal) Card Format

2. Binary Card Format (10 and 12 row)

The alphanumeric card format is one character per card column, 80 characters per card, punched in Hollerith code. Hollerith codes and the corresponding GE 225 bit configurations are given in the Appendix. Acceptable characters include the 26 letters of the alphabet, numbers 0 through 9, and fourteen special characters and punctuation marks plus a space or blank. These 50 characters are:

0 - 9	Numbers	=	Equal sign
A - Z	Alphabet	.	Period
+	Plus sign	$	Dollar sign
-	Minus sign or hyphen	*	Asterisk
	Space or blank	,	Comma
/	Slash	%	Percent sign
#	Number sign	[	Left bracket
@	"At" sign	]	Right bracket
_	Underscore		

Binary reading is accomplished in one of two formats - 10 row or 12 row; the selection of the desired format is under program control. The 10-row binary mode consists of 10 bits per card column which equals one-half of a 20 bit GE 225 word; therefore, a total of 40 binary words may be read from one card. The first card column of any two columns constituting a full word contains the most significant ten bits of the word.

In the 12 - row binary format, rows 11 and 12 are used so that 12 binary bits can be punched in each column. When placed in memory, the 12 bits of

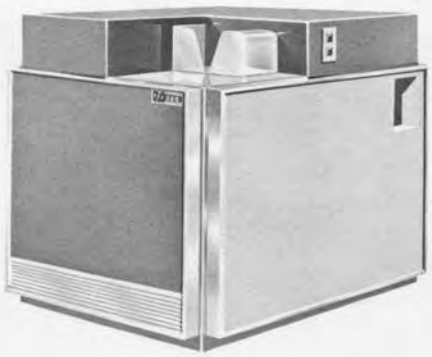
each column are consecutively placed in the 12 least significant bit positions of a word. The remaining eight bits of the word are set to zero.

Card reading occurs concurrently with other input/output and Central Processor operations.

CARD PUNCHING EQUIPMENT

Two card punching devices are available for use in the GE 225 system, both having a set of input logic circuits which link them to the computer. One punches at the rate of 100 cards per minute, and the second punches at the rate of 250 cards per minute. Both devices punch standard 7-3/8" X 3-1/4" 80-column 12-row cards. Information is punched row-by-row at a punching station having 80 punching dies. The input and output hoppers of both devices have a capacity of 800 cards.

Cards are punched under the control of the program; each card punch instruction results in punching one card. However, continuous punching can be achieved by program control.



The control panel on the card punch mechanism provides manual start/stop control and error indicator lights. Provision is made for checking up to 40 columns for double punch or blank column errors; checking is controlled by plugboard wiring in the punch unit. Card feed checks are also made.

The Card Punch equipment can punch cards in two formats:

1. Alphanumeric (Decimal) Card Format
2. Binary Card Format (both 10 and 12 bits per row)

These formats are described in the previous description of the card reading equipment. Card field arrangement is accomplished by program control or by plugboard wiring.

Card punching occurs concurrently with other input-output and Central Processor operations.

MAGNETIC TAPE SYSTEM

A major input and output medium for the GE 225, in many applications, is magnetic tape. It is used for master record file storage, input and output purposes, and is readily available to provide data for successive computer runs. Additional advantages of magnetic tape are compactness, repetitive use and speed.

The GE 225 Magnetic Tape System consists of one or more magnetic tape controllers, each capable of directing from one to eight magnetic tape handlers. Data transfer rates are: 15,000, 22,500, 41,662, and 62,500 characters per second.

Data are recorded on tape in groups of words called records - these records may be of fixed or variable length. Inter-record gaps physically separate magnetic tape records. This gap is a 3/4 inch section of erased tape which is used for starting and stopping tapes between records. An end of file record can be programmed to separate groups of information, or to mark the end of information on a specific tape. The end of file record is an erased section of magnetic tape 3-3/4 inches long, followed by the code combination 0001111.

TAPE CONTROLLER

The Magnetic Tape Controller is the link between the Central Processor and the tape handlers. All tape commands are transmitted to the Magnetic Tape Controller by the Central Processor through the Controller Selector. Upon transference of a command, the Central Processor is released for other operations while the tape instruction is executed.

The Magnetic Tape Controller provides logic for selecting the magnetic tape handler, and for reading, writing, and rewinding the magnetic tape. It also:

Monitors the data flow between tape units and memory

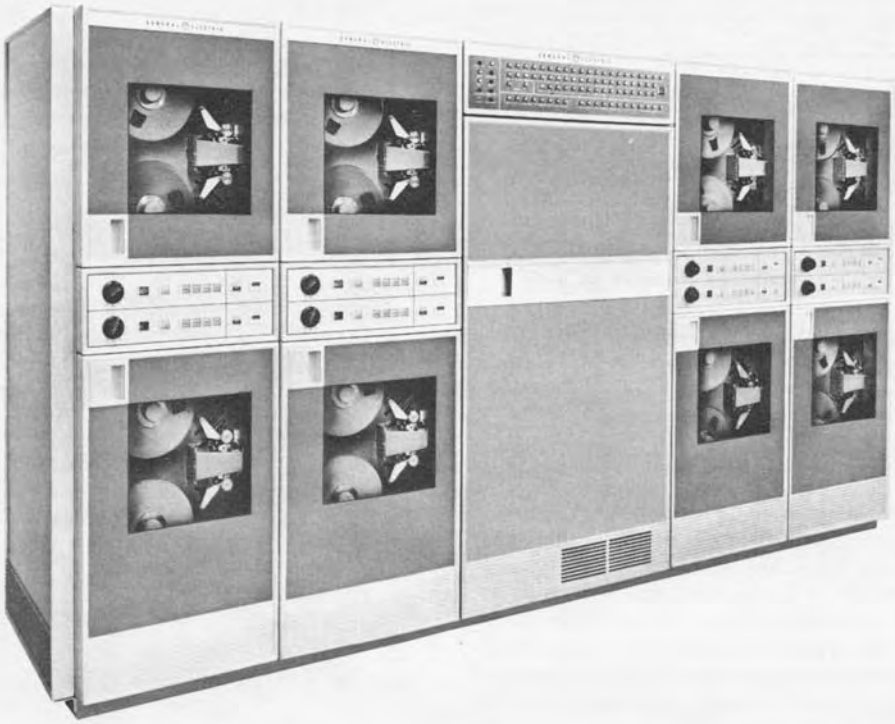
Initiates and times the starting and stopping of the selected tape unit

Forms and/or detects the end of record, end of tape, and end of file conditions.

Furthermore, the Magnetic Tape Controller contains error checking circuitry to insure reliability of the magnetic tape system.

TAPE HANDLERS

Each Tape Handler contains two 10-1/2 inch reels,(or four on a dual handler), one for feeding and the other for take-up, that hold tapes up to 2400 feet long. There are separate read and write heads for the reading and



recording processes. Photoelectric cells sense the beginning-of-tape marker and the end-of-tape marker. The handler mechanism, with its associated controller, processes one-half inch wide mylar magnetic tape, up to 2400 feet in length. There are three types of handlers available, each having the following characteristics:

Transfer Rate - characters per second	15 KC	15 and 41.6 KC	22.5 and 62.5 KC
Density - characters per inch	200	200 and 555.5	200 and 555.5
Forward speed - inches per second	75	75	112.5
Average combined start and stop time in milliseconds	12.0	10.8	7.9

Rewind speed: The rewind speed is never less than twice the forward speed regardless of the number of feet being rewound.

Checking features: These features are described on page 22.

Word format: Both binary coded decimal and binary - identical with other major tape systems in current use.

Each tape handler contains a file protect ring that fits into a guide in the tape reel. If the file protect ring is removed, it is impossible to write on the tape.

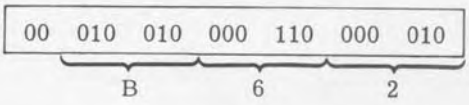
MODES OF OPERATION

The tape unit reads (forward or backward) and writes magnetic tape in three modes - 6 - bit binary coded decimal, binary and special binary. Special binary is a mode that permits the interchange of tapes from other systems with the GE 225 tape system. The mode of operation of the tape unit is determined by the programmed command. Tapes packed at low density may be read backward.

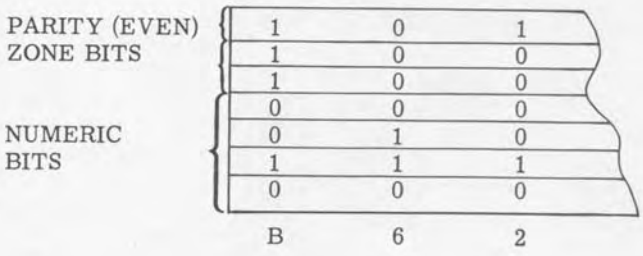
Each reel of magnetic tape contains seven tracks for recording. Six bits of data are recorded across the tape, and the seventh track contains a parity bit, as required, which is generated at the time of recording. In the binary mode the number of lateral "1's" will always be odd; this is known as odd parity. The number of "1's" when using the BCD mode will always be even; this is referred to as even parity. The two gap tape head allows information to be read and checked for correct parity as it is written on tape.

Binary Coded Decimal Mode

In the BCD mode of operation, a word in memory consists of three six-bit characters. Positions 2 through 19 contain data. For instance, the alphanumeric "B62" exists in memory as three BCD characters as follows:



When written on magnetic tape these characters appear as three six-bit characters accompanied by a parity bit, as shown below:



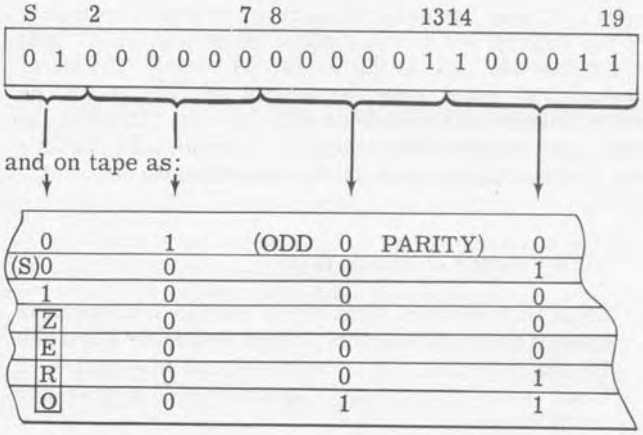
When writing on tape in the BCD mode, bit positions 0 and 1 of the word in memory are ignored; when reading from tape, these positions are automatically set to zero. In this mode of operation, the zone bits of some characters are altered for compatibility with other tape systems. This alteration is accomplished automatically upon reading or writing as follows:

Memory	Tape
00	00
01	11
10	10
11	01

Binary Mode

In this method of operation, the 20 bits comprising a word in memory are written as four magnetic tape positions. Three tape positions contain 6 bits of data and the fourth position contains 2 bits of data. The remaining bits of this fourth position contain zeros which are ignored when reading from tape, and which are automatically inserted when recording on tape.

As an illustration of the appearance of a binary word on tape, the number 262,243 as it appears in memory is:



The GE 225 can also operate in the special binary mode. This is the same as the regular binary mode except that the sign and one bit positions of a word are ignored. The 18 bits are written as three positions on tape. This is an efficient method of combining binary and BCD data and of achieving maximum data storage for large master files.

Control

On-line control of the tape handlers is achieved by the computer program. The operator panel, located on the tape handler, provides off-line control.

Manual controls include:

1. Power On/Off
2. Forward
3. Reverse
4. Rewind
5. Stop
6. Local/Remote
7. Tape Transport address selector switch with addresses 0 through 7.

The panel also contains two indicator lights - Ready/Not Ready and Write Inhibit.

ERROR DETECTION CIRCUITS

The GE 225 tape system incorporates error detection circuits to insure accurate data transfers between memory and tape. These circuits are:

1. Controller Input/Output Register Exhaust or Overflow (not numerical overflow). Checks to prevent reading new data over old data before the latter is processed; and to prevent writing a second time before new data is received.

2. Lateral Parity. This parity bit insures the accuracy of each character when read from tape.

3. Longitudinal Parity. This is a parity bit check on each of the seven record tracks, recorded at the end of each record.

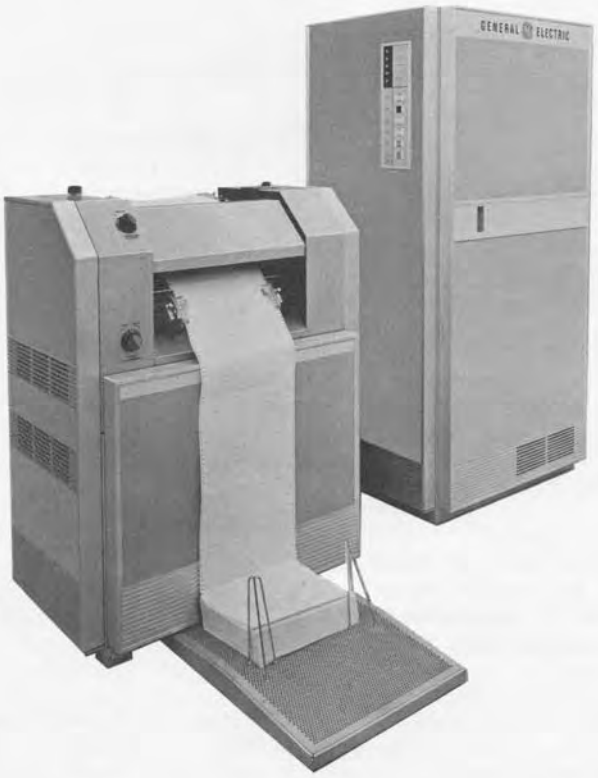
4. Modulo Three or Four. When information is read from tape a check is made to determine that the proper number of characters (three in BCD and special binary modes, and four in binary mode) constitute a word.

5. Write Check. All data written on magnetic tape is checked immediately after it is written by reading back on a physically separate read head and verifying lateral parity, longitudinal parity and modulo check.

6. All information is parity checked when transferred to and from memory.

HIGH SPEED PRINTER

A High Speed Printer is available for applications where large amounts of data are developed for visual output.



The High Speed Printer is an on-line buffered device which prints 120 alphanumeric characters per line at a speed of 900 lines per minute. Skipping (or paper slew) is 22 inches per second.

There are a total of 50 printable characters: the numbers 0 through 9, the 26 letters of the alphabet, and 14 special characters, plus a space or blank. The special characters are:

+	Plus sign	=	Equal
.	Period	,	Comma
-	Minus	[	Left bracket
$	Dollar sign	]	Right bracket
*	Asterisk	#	Number sign
%	Percent sign	@	"At" sign
_	Underscore		Space or blank
/	Slash		

Printing is open Gothic style, spaced 10 characters per inch horizontally and 6 lines per inch vertically. The printer accommodates fan folded paper in any form length up to 22 inches per page; the paper width may be from 6 to 16-1/2 inches. Paper is sprocket fed. Up to 5 carbon copies can be made.

Data to be printed are transferred from memory to the Printer Controller. Three methods, which may be used in any combination, are available to edit the print line format:

1. Initial setup on the record to conform exactly to the print line desired.

2. Rearrangement of the record in memory by programmed insertion or deletion of characters.

3. Editing performed by logic circuits in the Printer Controller, based on a combination of related data and format words transferred to the Printer Controller. Editing includes zero suppression, deletion of data, and insertion of special symbols, constants, and spaces. Many program steps are eliminated because of the editing ability of the GE 225 Printer Controller.

The Printer Controller operates independently of the Central Processor after transfer of the data to be printed.

Slewing (vertical movement of paper) is accomplished by specifying the number of spaces to be skipped in the print command, or by writing separate slew commands to skip before or after the line is printed. Slew commands are also available to cause paper to skip to one of eight channels in a paper tape loop.

Memory dumps are readily available without programming or using library routines. The contents of memory are printed in octal notation, starting at location 0000, when the Memory Dump button is depressed, the High Speed Printer is halted when the Manual Clear button is depressed. Both buttons are located on the Printer Controller.

Checking features include a parity check on the word received for printing, and an "end of paper" detection.

MASS RANDOM ACCESS FILE MEMORY

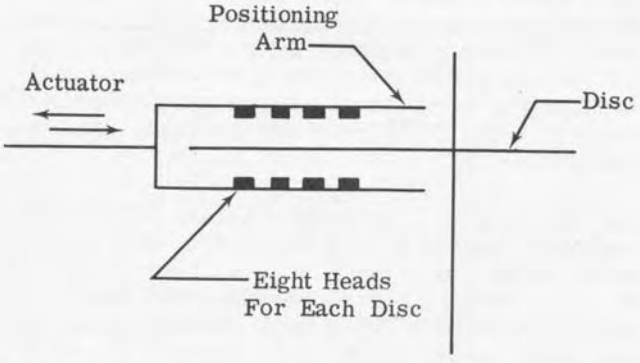
Mass Random Access File Memory units eliminate the costly and time-consuming sequential sorting necessary in the use of magnetic tapes. With these units, transactions may be processed in any sequence (without searching through unrelated information) simultaneously with all GE 225 computing and input-output operations.

The storage area of the Random Access File Memory consists of circular 31-inch discs placed on a vertical motor driven shaft. Each side of each disc is served by four read-write heads. The heads are mounted in opposed pairs on a two-pronged positioning arm for each disc as shown below.

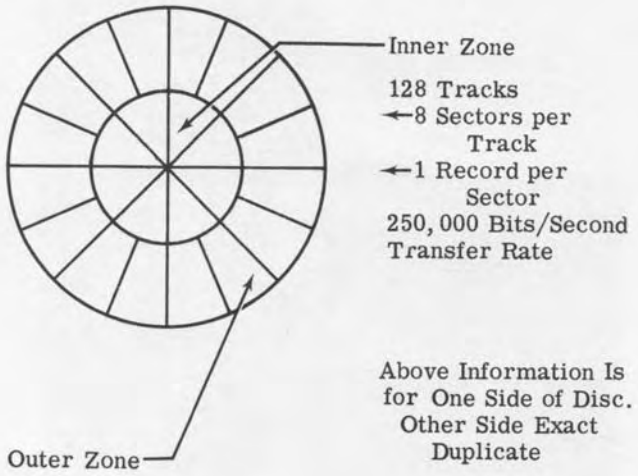
This arrangement permits the two surfaces of each disc to be served by a read-write head through a small movement of the positioning arm. This limited movement, parallel to the disc, provides a positioning time of less than 200 milliseconds. The discs revolve at high speeds; latency time is a maximum of 50 milliseconds. The average access time for any track is 133 milliseconds (average latency time is 25 milliseconds).

Mass Random Access File Memory is available in two sizes: a 16-disc file capable of storing 98,304 64-word records; and a 64-disc file capable of storing 393,216 64-word records. A maximum of four 16-disc units, or one 64-disc unit may be connected to one controller. The controller communicates with the Central Processor through the Controller Selector like other peripheral equipment. A panel permitting local test and control is provided on each controller.

Each surface of a disc is divided into two circular sections, each section consisting of 128 circular tracks. Information is recorded serially in these tracks. Transfers of data to the inner tracks occur at the rate of 250,000 bits per second - the outer tracks have a transfer rate of 500,000 bits per second. Eight 64-word records may be recorded in each of the inner 128 tracks, and sixteen 64-word records may be recorded in each of the outer 128 tracks.



The diagram below shows a disc and the division of the surface into inner and outer tracks; it depicts how the inner zone is divided into eight sectors, and how the outer zone is divided into sixteen sectors.



Each 20-bit word, plus a parity bit, is stored as an image of memory. The parity of a word received from the Central Processor is checked in the controller, which also checks for errors when reading from a disc. Should a parity error exist upon transfer from the file, a check word is transferred to the Central Processor (word 65 of read in). Through programming, this word may be used to correct the erroneous condition.

PAPER TAPE READER

The Paper Tape Reader is an on-line component. It reads alphanumeric information from perforated paper tape at a speed of either 1000 or 250 characters per second. This unit reads standard paper tape containing channels as follows: 8 channel, one inch wide; 5 channel, 11/16 inch wide; or 6 or 7 channel tape 7/8 inch wide. (The number of channels desired is selected by the operator.) Paper tape density is 10 characters per inch. The reader accommodates paper tape .004 inch thick.

At operator option, the Paper Tape Reader can operate at the high speed of 100 inches per second or at a low speed of 16.7 inches per second. The stop distance is less than .15 inch at high speed, and .025 inch at low speed.

The Central Processor converts paper tape characters into the absolute language of the computer. Data rearrangement is accomplished by the computer program.

Checking features include a parity check and a media presence check.

PAPER TAPE PUNCH

Paper Tape Punches are on-line output devices which punch alphanumeric information at the rate of 110 characters per second. Separate units are available to punch 5 channel tape only, and to punch, at operator option, 6, 7, or 8 channel tape. In either case, the punching density is 10 characters per inch.

The punch units hold an 800 foot roll of paper tape .004 inch thick.

Checking features include parity generation and a media presence check. Data rearrangement is accomplished by the computer program.

Paper Tape characters are:

The numbers 0 through 9	, Comma
The letters A through Z	@ At sign
/ Slash	☐ Box
- Minus	% Percent
& Ampersand	* Asterisk
. Period	# Number sign
$ Dollar sign	Delete character for 8 channel tape

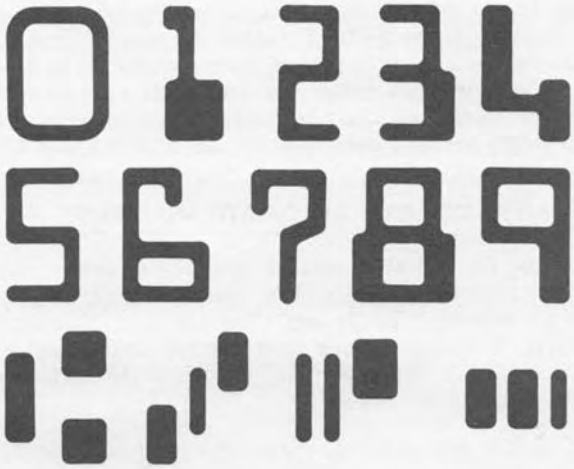
Paper Tape operations that may be programmed are:

Carriage Return

DOCUMENT HANDLER SYSTEM

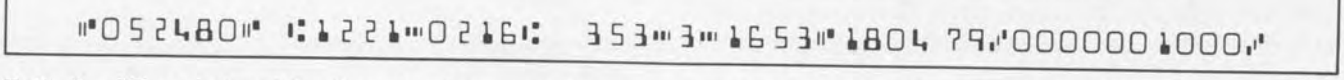
An advance in data processing is the development of magnetic ink character recognition by the General Electric Company. This process simplifies verification, reconciliation, and other accounting procedures because the original source information is recorded on the machine-processed document in a language easily recognized by human beings. Magnetic ink characters recognized by the GE 225 system are neither destroyed nor obliterated by overprinting, dirt, or even adhesive tape.

The Document Handler, or sorter, recognizes 14 characters of the special E13B font, the ten decimal digits and four special symbols called cue characters as shown below:



Cue characters are normally used to separate fields of decimal digits. For example, in a banking application the cue characters separate fields of decimal digits in the following manner:

The cue characters in this illustration separate such fields as the Federal Reserve routing district, a number assigned to the bank by the American Bankers Association, the account, and the dollar amount.



Both the 12-pocket and the 2-pocket Document Handlers are available with the GE 225 system. The 2-pocket Document Handler is essentially an on-line reading device, the second pocket being for rejects. In on-line operations, the controller accommodates both types of handlers, in any combination not to exceed two per controller, for simultaneous input. Off-line, any number of document handlers may be utilized. Each Document Handler can process 1200 documents per minute.

When a document handler is operated on-line, data is transferred to the central computer through the controller and the Controller Selector. The controller converts the bi-quinary output of the sorter to GE 225 binary coded decimal format. Each BCD character is stored in memory consecutively, one BCD character per word, in any sequential area selected by the programmer.

The six BCD bits occupy the six least significant bit positions of each memory location. If the BCD character is a numeric digit, all remaining bit positions are zeros. If the BCD character in memory is a cue character, a one bit is placed in the sign and the most significant bit positions. If the BCD character in memory is the invalid character, a one bit is placed in the sign position only.

PRIORITY PROGRAM AUTOMATIC INTERRUPT

With the GE 225 system it is possible to program for an automatic interruption of the main program to process a "priority" program. The optional Automatic Priority Interrupt device provides for concurrent operation of any or all peripherals while the main program is being processed.

The Automatic Priority Interrupt monitors the card reader, the card punch, and the devices connected through the Controller Selector. The interrupt feature takes effect only when a peripheral unit has been previously engaged and has returned to an idle status. The initial engagement is controlled by the programmer, depending upon the specific application desired. "Priority" programs are, for example, those where it is desired to transfer data from cards to tape, or from tape to cards, or, perhaps, to transfer data from cards, tape, or memory to the high speed printer - or to mass random access memory.

A special instruction at the beginning of the main program sets up the automatic interruption to permit exit from the program when a peripheral signals the computer that it is idle. The "priority" routine is entered and it performs its function; control is then returned to the main program and at the same time, the peripheral device is operating. Upon transfer from the main program, the instruction address contained in the P counter is stored in a special modification word; upon completion of the "priority" program, a branch BRU 0000 instruction, modified by the modification word, transfers control to the instruction; the main program would have processed if it had not been interrupted.

Entry to a "priority" program automatically nullifies the setting of the Automatic Priority Interrupt. The "priority" program must include a resetting (one instruction) of the Automatic Priority Interrupt circuit just prior to return to the main program if it is desired to keep this feature operative.

AUXILIARY ARITHMETIC UNIT

The addition of the Auxiliary Arithmetic Unit increases the arithmetic capability of the GE 225 System - it is particularly useful in applications where numerous floating point or double word calculations are required. The circuitry of this unit processes floating point arithmetic at much higher speeds than those possible when using automatic coding techniques. The latter method is suggested when floating point arithmetic is done on a limited basis. However, programs written for either method of processing are compatible because the same data format is used.

The Auxiliary Arithmetic Unit provides increases capability because it contains two 40-bit registers - the AX and QX registers. (These units correspond to the A and Q registers in the Central Processor.) Eighty bits permit both floating point and fixed point calculations on extremely large numbers.

This arithmetic device is connected to the Central Processor through the Controller Selector. It operates simultaneously with other peripheral equipment or the Central Processor.

Three modes of calculations may be performed by the Auxiliary Arithmetic Unit: unnormalized floating point, normalized floating point, and fixed point operations. It is usual to do floating point operations in the normalized mode. The word normalized refers to the results of calculations performed in the normalized mode. A normalized number is one in which the most

significant non-zero digit of the mantissa is next to the decimal point. For example, the number 6786 might be represented in unnormalized form as:

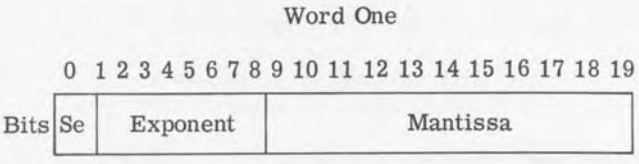
$$.006786 \times 10^{6}$$

In normalized form, this number would be:

$$.6786 \times 10^{4}$$

Addition, subtraction, multiplication, and division may be done under any of the three modes of operation. Special commands, presented in the instruction repertoire section, enable the execution of the desired number and type of calculation.

During floating point operations, data to be operated upon by the internal logic of the Auxiliary Arithmetic Unit comes from the main memory of the central processor. This information is available in memory in a two-word, floating point format:

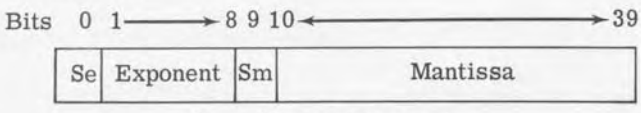


Se = Sign of Exponent (0 = plus; 1 = minus)
Sm = Sign of Mantissa (0 = plus; 1 = minus)

FLOATING POINT WORD FORMAT IN MEMORY

The binary point is assumed to be placed before the first bit (bit 9) of the mantissa. This format produces a binary number with a 30-bit mantissa and a binary characteristic range of -255 to +255. This is approximately equal to a decimal format of a 9-digit mantissa and a decimal range of -77 to +77. The use of two words allows one of the sign positions to be applied to the exponent which, in turn, allows the use of the full range of the exponent.

When data in the two-word floating format enters the Auxiliary Arithmetic Unit, it is converted into one word 40 bits long. A word exists in the 40-bit AX register as:

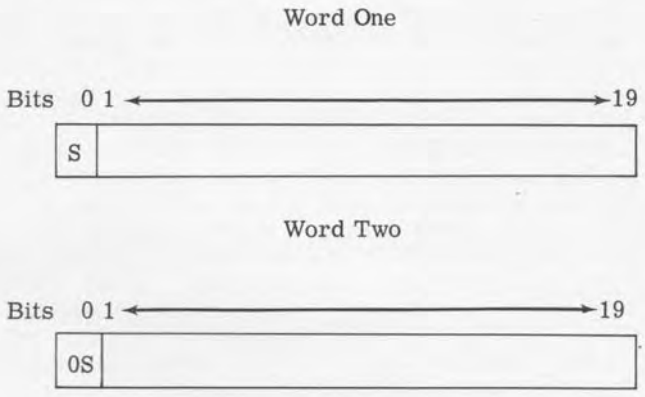


Se = Sign of Exponent
Sm = Sign of Mantissa

FLOATING POINT WORD FORMAT IN THE AX REGISTER

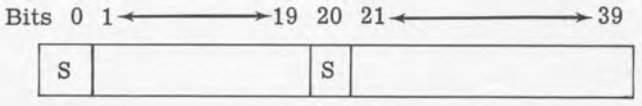
The QX register is an extension of the AX register. It also consists of an eight bit exponent with sign and a thirty bit mantissa with sign. The value of the exponent of the QX register is the value of the AX register minus 30.

The word format in memory for fixed point is as follows:



S = Sign of Word One is Sign of Fixed Point Double Word

When the two fixed point words from memory enter the Auxiliary Arithmetic Unit, they appear in the AX register as one word 40 bits long:



S = Sign

FIXED POINT WORD FORMAT IN THE AX REGISTER

The fixed point double word in the AX register consists of 38 information bits plus two sign bits. Similarly, a fixed point double word in the QX register consists of 38 information bits and two sign bits. The sign bits in AX or QX are identical.

The instruction words for the Auxiliary Arithmetic Unit are contained in one 20-bit word identical to the format of Central Processor instructions: bits 0-4 contain the operation code; bits 5-6 contain the automatic modification bits; and bits 7-9 contain the address of the operand. As in the case of the Central Processor, the floating point instructions are available to the user through the use of mnemonics which are listed in the instruction repertoire. The use of any of the Auxiliary Arithmetic Unit instructions for floating point operations assumes that the operands to be acted upon are already in floating point format. Information is put in floating point format by means of a subroutine furnished for this purpose.

INSTRUCTION REPERTOIRE

The GE 225 operates under the programmed control of over one hundred and fifty instructions. These instructions are classified in the following categories:

1. Arithmetic
2. Data Transfer
3. Shift Operations
4. Internal Test-and-Branch
5. Console Operation
6. Punched Card Input-Output
7. Paper Tape Input-Output
8. Controller Selector
   - High Speed Printer System
   - Magnetic Tape System
   - Mass Random Acess File System
   - Document Handler System
   - Auxiliary Arithmetic Unit
9. Optional Central Processor Instructions

The following list of instructions gives the mnemonic code for the command and an indication of whether a memory location (operand address) or a constant is required.

Machine running programs developed by the use of the General Assembly Program are prepared by specifying in tabular form the mnemonics of the instructions, the memory locations or constants involved, and whether the instructions are to be modified. For example, if it is desired to store the contents of the A register in a modified memory location, the following command structure is used:

Operation Code	Operand Address	Modification
STA	Y	X

The mnemonic STA (store A) stores the contents of the A register in memory location (Y). The operand address (Y) may be a decimal number (for example, memory location 1050) or it may be a symbolic designation (for example, NETPAY) which is intelligible to the GE 225 assembler routines.

With the General Assembly Program the programmer may use symbols for operand addresses whenever he desires, and memory locations will be automatically assigned. The letter "X" indicates whether the instruction is to be automatically modified. A zero indicates that there is to be no modification and a 1, 2, or 3 selects memory modification words 0001, 0002, or 0003. All instructions may be automatically modified unless stated otherwise. In all instructions involving bringing a word from memory, the word in memory remains unchanged; in all instructions involving the transfer of information from registers, the condition of the register after execution is unchanged unless otherwise stated. Some instructions do not require an operand, so for these no entry will be found in the column labelled "Operand Address". A "Y" indicates an operand address; a "K" indicates the operand itself. The execution times (given in word times) include the fetching of the next instruction. For the purposes of this manual the description of each of these instructions is abbreviated. Extensive explanations are included in the GE 225 Reference Manual. Programs can also be written in General Compiler language - this compiler generates the Mnemonics listed below.

Mnemonic Oper Code	Operand Address	Word Time

ARITHMETIC

ADD Y 2

ADD. The contents of the specified memory address (Y) are added to the contents of the A register; result is placed in A.

SUB Y 3

SUBTRACT. The contents of the specified memory address (Y) are subtracted from the contents of the A register; result is placed in A.

DAD Y 3

DOUBLE LENGTH ADD. The contents of Y and Y + 1 are added to the contents of the A and Q registers. Result is placed in A and Q.

DSU Y 5

DOUBLE LENGTH SUBTRACT. The contents of Y and Y + 1 are subtracted from the contents of A and Q. Result is placed in A and Q.

Mnemonic Oper Code	Operand Address	Word Time
MPY	Y	minimum 9 maximum 23

MULTIPLY. The contents of the specified memory location (Y) are multiplied by the contents of the Q register. The result is placed in registers A and Q. If the contents of A are not set to zero before the MPY command is given, the contents of A will be added algebraically to the least significant half of the product.

DVD	Y	minimum 26 maximum 29

DIVIDE. The contents of registers A and Q are divided by the contents of the specified memory location (Y). The quotient is placed in A; the remainder in Q.

INX	K	X	3

INCREMENT X BY K. K, positions 7 through 19 of the I register, is added absolutely to the contents of Register X (5-19), and the result replaces the contents of Register X (5-19). Any carry from position 5 of Register X is lost. This instruction cannot be automatically modified. X may be one of four memory locations: 0000, 0001, 0002, 0003. Locations 0001, 0002, and 0003 are the automatic modification words. Location 0000 (as well as 0001, 0002, and 0003) can be used as a counter.

ADO	3

ADD ONE. One is added to the contents of the A register.

SBO

SUBTRACT ONE. One is subtracted from the contents of the A register.

DATA TRANSFER

LDA	Y	2

LOAD A. The contents of the specified memory location (Y) replace the contents of the A register.

STA	Y	2

STORE A. The contents of register A replace the contents of the specified memory location (Y).

DLD	Y	3

DOUBLE LENGTH LOAD. The contents of Y and Y + 1 replace the contents of A and Q.

DST	Y	3

DOUBLE LENGTH STORE. The contents of A and Q replace the contents of Y and Y + 1.

Mnemonic Oper Code	Operand Address		Word Time
LDX	Y	X	2

LOAD X. The contents of a specified memory location Y are stored in the index register specified by the X portion of the command.

STX	Y	X	2

STORE X. The contents of the index register X are stored in memory location Y.

MOV	Y	4 + 2N

MOVE (N) number of words from one location to another location. Y (bits 5-19) specifies the starting address from where data is moved. The A register must contain the starting address to where data is to be moved. The Q register must contain the number of words to be moved. (This is an optional instruction.)

LQA	3

LOAD Q FROM A. The contents of register A replace the contents of register Q.

LAQ	3

LOAD A FROM Q. The contents of Q replace the contents of A.

XAQ	3

EXCHANGE A AND Q. The contents of A and Q are interchanged.

MAQ	3

MOVE A TO Q. The contents of register A replace the contents of register Q. Zeros replace the contents of A.

STØ	Y	3

STORE OPERAND ADDRESS. The contents of A (bits 7 through 19) replace the contents of Y (bits 7 through 19).

ØRY	Y	3

OR A INTO Y. Each bit in A is examined. If there is a one in A in a bit position, a one is placed in Y in the corresponding bit position. If there is a zero in A, the corresponding bit position in Y is left unchanged.

EXT	Y	3

EXTRACT. Each bit in specified memory location (Y) is examined. If there is a one in Y in a bit position, a zero is placed in A in the corresponding bit position. If there is a zero in Y, the corresponding bit position in A is left unchanged.

Mnemonic Oper Code	Operand Address	Word Time
LDZ		3

LOAD ZERO INTO A. The contents of register A are replaced by zeros.

LDØ		3

LOAD ONE INTO A. The contents of register A are set to zero and a one is placed in A in the right most bit position.

LMØ		3

LOAD MINUS ONE INTO A. The contents of register A are replaced by ones.

CPL		3

COMPLEMENT A. Each bit in register A is inverted; that is, a one is replaced by a zero and a zero by a one. This instruction gives the one's complement of a number.

NEG		3

NEGATE A. The two's complement of the contents of A replaces the contents of A. If the capacity of A is exceeded, the overflow indicator is turned on.

CHS		2

CHANGE SIGN OF A. The sign of A is changed.

NØP		3

NO OPERATION. Zero is added to the contents of A.

SHIFT OPERATIONS

All shift commands vary between two and twelve word times, depending upon the length of the shift. A maximum of 31 places can be shifted. Two word times are required for a shift of two bit positions or less. One additional word time is required for each additional three-bit shift or fraction thereof.

Mnemonic Oper Code	Operand Address	Word Time
SRA	K	2+

SHIFT RIGHT A. The contents of register A are shifted right K places. If A is positive, zeros are inserted in the vacated positions of A. If A is negative, ones are inserted in the vacated positions of A.

SLA	K	2+

SHIFT LEFT A. The contents of A are shifted left K places. Vacated positions of A are filled with zeros. If a non-zero bit is shifted out of position 1, the overflow indicator will be turned on.

Mnemonic Oper Code	Operand Address	Word Time
SCA	K	2+

SHIFT CIRCULAR A. The contents of A are shifted right K places in a circular fashion; that is, the bit shifted out of position 19 is inserted into position 1.

SRD	K	2+

SHIFT RIGHT DOUBLE. The contents of A and the contents of Q together are shifted K places to the right. Bits shifted out of A shift into Q; bits shifted out of Q are lost. The filling of vacated positions is the same as for SHIFT RIGHT A.

SLD	K	2+

SHIFT LEFT DOUBLE. The contents of registers A and Q together are shifted K places to the left. Bits shifted out of Q shift into A; vacated positions of A are filled with zeros. If a non-zero bit is shifted out of position 1, the overflow indicator will be turned on.

SCD	K	2+

SHIFT CIRCULAR DOUBLE. The contents of A and Q together are shifted K places to the right in a circular fashion. Bits shifted out of A shift into Q and those from Q shift into A.

SAN	K	2+

SHIFT A AND N RIGHT. The contents of A and N together are shifted K places to the right. Bits shifted out of A shift into N; bits shifted out of N are lost. The filling of vacated positions is the same as for SHIFT RIGHT A.

SNA	K	2+

SHIFT N AND A RIGHT. The contents of N and A together are shifted K places to the right. Bits shifted out of N shift into A; vacated positions in N are filled with zeros.

NAQ	K	2+

SHIFT N, A AND Q RIGHT. The contents of N, A and Q together are shifted K places to the right. Bits shifted out of N shift into A; bits shifted out of A shift into Q. Vacated positions in N are filled with zeros.

ANQ	K	2+

SHIFT A INTO N AND Q. The contents of A are shifted K places to the right into both N and Q. The filling of vacated positions is the same as for SHIFT RIGHT A.

Mnemonic Oper Code	Operand Address	Word Time
NOR	K	3+

NORMALIZE A REGISTER. If the sign of register A is plus, the contents of A are shifted left and a count is maintained of leading zeros. If the sign of A is minus, the contents of A are shifted left and a count is made on leading ones. Vacated positions of A are filled with zeros.

DNO	K	3+

DOUBLE LENGTH NORMALIZE. If the sign of register A is plus, the contents of A and Q are shifted left and a count is maintained of leading zeros. If the sign of A is minus, the contents of A and Q are shifted left and a count is made on leading ones. Vacated positions of Q are filled with zeros.

INTERNAL TEST AND BRANCH

BRU	Y	1

BRANCH UNCONDITIONALLY. Control is transferred to the instruction located at memory location Y.

SPB	Y	X	2

STORE P AND BRANCH. The location of the SPB command (bits 5-19 of register P) is stored in register X (bits 5-19) and control is transferred to the instruction located at Y in the same memory bank where the SPB instruction is located. This instruction cannot be automatically modified.

The following branch instructions test to determine whether a particular internal condition is true or false. If the condition tested is true, the computer executes the next sequential instruction. If the condition tested is false, the computer skips the next instruction and executes the second sequential instruction.

Mnemonic Oper Code	Operand Address	Word Time
BPL		2

BRANCH ON PLUS. Register A is tested for a plus sign.

BMI		2

BRANCH ON MINUS. Register A is tested for a minus sign.

BZE		2

BRANCH ON ZERO. Register A is tested for total zero content.

Mnemonic Oper Code	Operand Address	Word Time
BNZ		2

BRANCH ON NO ZERO. Register A is tested for non-zero content.

BOD		2

BRANCH ON ODD. Register A is tested for an odd value (least significant bit is a one).

BEV		2

BRANCH ON EVEN. Register A is tested for an even value (the least significant bit is a zero).

BOV		2

BRANCH ON OVERFLOW. The overflow indicator is tested for the ON condition (if overflow occurred, indicator is automatically turned OFF).

BNO		2

BRANCH ON NO OVERFLOW. The overflow indicator is tested for the OFF condition (if overflow occurred, the indicator is automatically turned OFF).

BPE		2

BRANCH ON PARITY ERROR. The parity error indicator is tested for the ON condition (if a parity error occurred, the indicator is automatically turned OFF).

BPC		2

BRANCH ON PARITY CORRECT. The parity error indicator is tested for the OFF condition (if a parity error occurred, indicator is automatically turned OFF).

BXH	K	3

BRANCH IF X IS HIGH. X is tested to determine if it is greater than or equal to K. X may be one of four memory locations: 0000, 0001, 0002, or 0003. X is not changed. This instruction cannot be automatically modified.

BXL	K	3

BRANCH IF X IS LOW. X is tested to determine if it is less than K. X may be one of four memory locations: 0000, 0001, 0002, or 0003. X is not changed. This instruction cannot be automatically modified.

CONSOLE OPERATION

Mnemonic Oper Code	Operand Address	Word Time
RCS		2

READ CONTROL SWITCHES. A word is set up in the A register, corresponding to the 20 manually set switches on the control console. The A register must be cleared before this instruction is given.

TON		2

TYPEWRITER ON. The power for the Typewriter is turned ON and the power for the Paper Tape Reader and Paper Tape Punch is turned OFF.

BNR		2

BRANCH ON N REGISTER READY. If the N register is available for input-output, (if the last TYPE or PAPER TAPE READ/WRITE instruction has been executed) the computer takes the next instruction in sequence; if not, the computer skips the next instruction and proceeds from there.

BNN		2

BRANCH ON N REGISTER NOT READY. If the N register is not available for input-output (if the last TYPE or PAPER TAPE READ/WRITE instruction has not been executed), the computer takes the next instruction in sequence. If it is available, the computer skips the next instruction and proceeds from there.

TYP		2

TYPE. If the power for the Typewriter is turned ON the six-bit coded character in N is typed. The contents of N are not changed.

OFF		2

POWER OFF. Power for the Typewriter, Paper Tape Punch and Paper Tape Reader is turned OFF.

PUNCHED CARD INPUT-OUTPUT COMMANDS

BCR		2

BRANCH ON CARD READER READY. If the card reader is ready to read cards and the card hopper is not empty, the computer takes the next instruction in sequence; if not, the computer skips the next instruction and proceeds from there.

BCN		2

BRANCH ON CARD READER NOT READY. If the card reader is not ready to read cards, or if the card hopper is empty, the computer takes the next instruction in sequence; if not, the computer skips the next instruction and proceeds from there.

Mnemonic Oper Code	Operand Address	Word Time
RCD	Y	2

READ CARDS DECIMAL. This command initiates continuous reading of cards punched in Hollerith alphanumeric code into memory starting at location Y, where Y is a multiple of 128 and less than 2048.

RCB	Y	2

READ CARDS BINARY. This command initiates continuous reading of cards punched in a binary mode into memory starting at location Y, where Y is a multiple of 128 and less than 2048.

RCF	Y	2

READ CARDS FULL. The 12 positions of each of the 80 columns of a card are read. The 12 punched positions of each column are placed in the 12 least significant bit positions of successive core memory locations. Y is the address into which the first word is placed and must be a multiple of 128 and less than 2048. If the card reader is not in ready status when this instruction is given, the computer halts.

HCR		2

HALT CARD READER. This command halts the card feed.

BPR		2

BRANCH ON CARD PUNCH READY. If the card punch is in a ready status, the computer takes the next instruction in sequence; if not, the computer skips the next instruction and proceeds from there.

BPN		2

BRANCH ON CARD PUNCH NOT READY. If the card punch is not in a ready status, the computer takes the next instruction in sequence; if it is, the computer skips the next instruction and proceeds from there.

WCD	Y	2

WRITE CARD DECIMAL. This command causes the information in memory locations Y through Y + 26 (where Y is a multiple of 128) to be punched into a card in Hollerith alphanumeric format.

WCB	Y	2

WRITE CARD BINARY. This command causes the information in memory locations Y through Y + 39)where Y is a multiple of 128) to be punched into a card in binary format.

Mnemonic Oper Code	Operand Address	Word Time
WCF	Y	2

WRITE CARDS FULL. 12 bit positions are punched in each column of a card. The 12 bits are the information contained in the 12 least significant positions of successive memory locations. Bit position 8 is punched in row 12 and bit position 19 is punched in row 9. Y is the address of the first memory location from which information is to be extracted and must be a multiple of 128 but less than 2048. If the card punch is not in ready status when this instruction is given, the computer will halt.

PAPER TAPE INPUT-OUTPUT

RON 2

READER ON. The power for the Paper Tape Reader is turned ON. A delay of 200 milliseconds must be programmed before an RPT (Read Paper Tape) command can be given. This instruction turns OFF the power for the Paper Tape Punch and Typewriter.

RPT 2 2

READ PAPER TAPE. This instruction initiates the reading of 6-bit characters from paper tape. If the Paper Tape Reader power is ON, the N register is cleared and one six-bit character is read into N. When operating at tape speeds of 250 characters per second, four milliseconds are available for processing before the next character is transferred to N; at operating speeds of 1000 characters per second, 1 millisecond is available.

HPT 2 2

HALT PAPER TAPE READER. This instruction halts the Paper Tape Reader. At speeds of 250 characters per second, the tape stops on the character being read; at speeds of 1000 characters per second, the tape stops after the next character enters the N register.

PON 2

PUNCH ON. The power for the Paper Tape Punch is turned ON. A delay of 500 milliseconds must be programmed before giving a command to punch paper tape. This instruction turns OFF the power for the Paper Tape Reader and Typewriter.

WPT 2

WRITE PAPER TAPE. If the power for the Paper Tape Punch is ON, the six-bit coded character in register N is punched. The contents of register N are not changed. Other instructions not using register N may be executed during this time.

CONTROLLER SELECTOR

Mnemonic Oper Code	Operand Address		Word Time
SEL	P	X	2

SELECT. The peripheral connected to controller P (addresses 0 through 7) is selected for the operation indicated by an associated instruction. For example, it may be followed by WPL which causes a line to be printed by the High Speed Printer.

BCS XXX P 2

BRANCH ON CONTROLLER SELECTOR. The peripheral connected to controller P is tested for the condition indicated by a mnemonic placed in the operand address field identified by XXX above. The mnemonics which can be placed in the operand field (XXX) are listed with the instructions for each peripheral.

HIGH SPEED PRINTER

The WPL, WFL, SLW and SLT commands must be preceded by the SEL P instruction which selects the specific controller P. These instructions cannot be automatically modified.

WPL Y N 2

WRITE PRINT LINE. One line of BCD information 1 to 120 characters long is printed. Y is the starting location in memory of the information to be printed. When present N indicates that information is numeric BCD only (if N is blank, information is alphanumeric only).

WFL Y 2

WRITE FORMAT LINE. One line of BCD information is printed under format control. Y is the starting address in memory of the format control words. This instruction should always be followed by a WPL instruction to specify the location of the first word of information to be printed and whether the information is alphanumeric or numeric.

SLW N 2

SLEW PAPER N LINES. The printer paper is spaced N (0-63) number of lines.

SLT K 2

SLEW PAPER TO TAPE PUNCH. The printer paper is spaced until a hole is detected in the vertical format tape. K is the specified tape channel.

The following branch instructions test whether a particular printer controller condition is true or false. If the condition tested is true, the computer executes the next sequential instruction. If the condition tested is false, the computer skips the next instruction and executes the second sequential instruction.

Mnemonic Oper Code	Operand Address		Word Time
BCS	BPN	P	2

BRANCH ON PRINTER NOT READY. The printer on controller P is tested for the not ready status.

BCS	BPR	P	2

BRANCH ON PRINTER READY. The printer on controller P is tested for the ready status.

BCS	BOP	P	2

BRANCH ON OUT OF PAPER. The printer on controller P is tested to determine if it is out of paper.

BCS	BNP	P	2

BRANCH IS NOT OUT OF PAPER. The printer on controller P is tested to determine if it is not out of paper.

BCS	BER	P	2

BRANCH ON ERROR. The printer on controller P is tested for any error.

BCS	BNE	P	2

BRANCH ON NO ERROR. The printer on controller P is tested for a no-error condition.

MAGNETIC TAPE

The WTD, WTB, RTD, RTB, RWD, WEF, BKW, BKR, RTS, WTS, RBB, RBD and RBS instructions must be preceded by a SEL P instruction which selects the specific controller P. These instructions cannot be automatically modified.

WTD	M	T	2
(blank)	N		

WRITE TAPE DECIMAL. N decimal words from memory starting at location M are written by tape handler T. The tape controller converts the GE 225 internal characters into magnetic tape binary coded decimal characters.

WTB	M	T	2
(blank)	N		

WRITE TAPE BINARY. N binary words (20 bits per word) from memory starting at location M are written by tape handler T.

Mnemonic Oper Code	Operand Address		Word Time
WTS	M	T	2
(blank)	N		

WRITE TAPE SPECIAL BINARY MODE. N words (18 bits per word) of information from memory starting at location Y are written by tape handler T. Bits 2-19 of a word are written on tape exactly as in memory (zone bits are not altered).

RTD	M	T	2
(blank)	N		

READ TAPE DECIMAL. A maximum of N decimal words are read by tape handler T and placed in memory starting at location M. The tape controller alters the magnetic tape characters into GE 225 internal binary decimal coded characters.

RTB	M	T	2
(blank)	N		

READ TAPE BINARY. A maximum of N binary words (20 bits per word) are read by tape handler T and placed in memory starting at location T.

RTS	M	T	2
(blank)	N		

READ TAPE SPECIAL BINARY MODE. A maximum of N binary words (18 bits per word) are read by tape handler T and stored in memory starting at location Y. A word of information is stored in bit positions 2-19 of a memory location exactly as it appeared on tape (zone bits are not altered).

RBB	M	T	2
(blank)	N		

READ BACKWARDS BINARY. Binary information is read from tape moving backward. A maximum of N words are read into memory, the first word being placed in location M. The second word is placed in M minus 1 and so on until N words are read.

RBD	M	T	2
(blank)	N		

READ BACKWARDS DECIMAL. Decimal information is read from tape moving backwards. A maximum of N words are read into memory, the first word being placed in location M. The second word is placed in M minus 1 and so on until N words are read. The tape controller alters the zone bits of characters read so that they conform to GE 225 internal binary coded decimal characters.

Mnemonic Oper Code	Operand Address		Word Time
RBS (blank)	M N	T	2

READ BACKWARDS SPECIAL BINARY. Information is read from tape moving backwards. Bit positions 2-19 of each word read are placed in memory exactly as on tape (zone bits are not altered). A maximum of N words are read into memory, the first word being placed in M. The second word read is placed in M minus one and so forth until N words are read.

RWD		T	2

REWIND. Rewind tape handler T.

WEF		T	2

WRITE END OF FILE. The end of file character (0001111) and end of file gap are written on tape by tape handler T.

BKW		T	2

BACKSPACE AND POSITION WRITE HEAD. The tape on tape handler T is backspaced one record and the write head is positioned to write.

The following branch instructions test to determine whether a magnetic tape controller condition is true or false. If the condition tested is true, the computer executes the next sequential instruction. If the condition tested is false, the computer skips the next instruction and executes the second sequential instructions.

Mnemonic Oper Code	Operand Address		Word Time
BCS	BTN	P	2

BRANCH ON TAPE CONTROLLER NOT READY. The tape on controller P is tested for the not ready status.

BCS	BTR	P	2

BRANCH ON CONTROLLER READY. The tape on controller P is tested for the ready status.

BCS	BEF	P	2

BRANCH ON END OF FILE. The controller on P is tested for end-of-file indicator ON.

BCS	BNF	P	2

BRANCH ON NO END OF FILE. The controller on P is tested for end-of-file indicator OFF.

Mnemonic Oper Code	Operand Address		Word Time
BCS	BET	P	2

BRANCH ON END-OF-TAPE. The controller on P is tested for end-of-tape indicator ON.

BCS	BNT	P	2

BRANCH ON NO END-OF-TAPE. The controller on P is tested for end-of-tape indicator OFF.

BCS	BRW	P	2

BRANCH ON TAPE REWINDING. The controller on P is tested for a tape rewinding.

BCS	BNR	P	2

BRANCH ON NO TAPE REWINDING. The controller on P is tested for no tape rewinding.

BCS	BPE	P	2

BRANCH ON TAPE PARITY ERROR. The controller on P is tested for parity error indicator ON.

BCS	BPC	P	2

BRANCH ON TAPE PARITY CORRECT. The controller on P is tested for tape parity error OFF.

BCS	BIO	P	2

BRANCH ON INPUT/OUTPUT BUFFER ERROR. The controller on P is tested for input/output buffer error indicator ON.

BCS	BIC	P	2

BRANCH ON INPUT/OUTPUT BUFFER CORRECT. The controller on P is tested for input/output buffer error indicator OFF.

BCS	BME	P	2

BRANCH ON MOD 3 OR 4 ERROR. The controller on P is tested for mod 3 or 4 error indicator ON.

BCS	BNM	P	2

BRANCH ON NO MOD 3 OR 4 ERROR. The controller on P is tested for mod 3 or 4 error indicator OFF.

BCS	BER	P	2

BRANCH ON ERROR. The controller on P is tested for error indicator ON.

BCS	BNE	P	2

BRANCH ON NO ERROR. The controller on P is tested for error indicator OFF.

MASS RANDOM ACCESS FILE MEMORY

The PRF, RRF, and WRF instructions must be preceded by the SEL P instruction which selects the specific controller P. These instructions cannot be automatically modified.

Mnemonic Oper Code	Operand Address		Word Time
PRF	(MRAF Address)	F	2
OCT			

POSITION MRAF. One of the MRAF units F (0 through 3) is positioned to receive or transmit a specific record. The line OCT contains the actual MRAF address (octal) of the selected MRAF.

RRF	N	F	2
(blank)	M		

READ MRAF. N is the number (1 through 16) of 64-word records to be transmitted from disc storage to core storage. R is the number (0 through 3) of the selected MRAF. M is the core memory address into which the first word of the record is stored.

WRF	N	F	2
(blank)	M		

WRITE MRAF. N is the number (1 through 16) of 64-word records to be transmitted from core storage to disc storage. R is the number (0 through 3) of the selected MRAF. M is the memory location of the first word to be transmitted from core storage to disc storage.

The following branch instructions test to determine whether a particular MRAF condition is true or false. If the condition tested is true, the computer executes the next sequential instruction. If the condition tested is false, the computer skips the next instruction and executes the second sequential instruction.

Mnemonic Oper Code	Operand Address		Word Time
BCS	BRN	P	2

BRANCH ON MRAF NOT READY. The MRAF on controller P is tested for MRAF not ready.

BCS	BRR	P	2

BRANCH ON MRAF READY. The MRAF on controller P is tested for MRAF ready.

BCS	FKR	P	2

BRANCH ON FILE NO. K READY. The specific MRAF on controller P is tested for ready status.

Mnemonic Oper Code	Operand Address		Word Time
BCS	FKN	P	2

BRANCH ON FILE K NOT READY. The specific MRAF on controller P is tested for ready status.

BCS	BIO	P	2

BRANCH ON INPUT-OUTPUT ERROR. The MRAF controller on P is tested for input-output error indicator ON.

BCS	BIC	P	2

BRANCH ON INPUT-OUTPUT CORRECT. The MRAF controller on P is tested for the input-output error indicator OFF.

BCS	RPE	P	2

BRANCH ON PARITY ERROR. The MRAF on controller P is tested for parity error indicator ON.

BCS	RPC	P	2

BRANCH ON PARITY CORRECT. The MRAF on controller P is tested for parity indicator OFF.

BCS	BER	P	2

BRANCH ON ERROR. The MRAF on controller P is tested for error indicator ON.

BCS	BNE	P	2

BRANCH ON NO ERROR. The MRAF on controller P is tested for error indicator OFF.

DOCUMENT HANDLER

The RSD, RDC, PKT, HTL, and ERB instructions must be preceded by a SEL P instruction to select a specific controller P. These instructions cannot automatically be modified.

RSD	M	N	2

READ SINGLE DOCUMENT. A document is read by document handler N. Information is read into memory - the first character goes into location M which must be a multiple of 64.

RDC	M	N	2

READ DOCUMENT AND CONTINUE FEEDING NEXT DOCUMENT. A document is read from document handler N. Information is read into memory - the first character goes into M which must be a multiple of 64. Approximately 50 milliseconds are available for computation before another RDC or halt feed instruction may be given. This instruction must be used to achieve the 1200 documents per minute reading speed.

Mnemonic Oper Code	Operand Address		Word Time
PKT	M	N	2

POCKET SELECT. The document read by handler N is stacked in pocket M. M is specified by S (special), 0 through 9, and R (reject).

HLT	M	N	2

HALT CONTINUOUS FEEDING. The continuous feeding of documents by handler N is halted. M is the memory address into which is read the first character of the document currently approaching the read head.

ERB		N	2

END READ BUSY. When a HLT instruction is given to document handler N, the ERB instruction is used to reset the document handler to a ready condition for further operation.

The following branch instructions test whether a particular document handler condition is true or false. If the condition tested is true, the computer executes the next sequential instruction. If the condition tested is false, the computer skips the next instruction and executes the second sequential instruction.

BCS	SKN	P	2

BRANCH ON SORTER K NOT READY. The sorter on controller P is tested for the not ready status.

BCS	SKR	P	2

BRANCH ON SORTER K READY. The controller on P is tested for the ready status.

BCS	NPK	P	2

BRANCH ON NO POCKET DECISION. The controller on P is tested to determine if a PKT instruction was not given to sorter K in the required time.

BCS	PDK	P	2

BRANCH ON POCKET DECISION, SORTER K. The controller on P is tested to determine if a PKT decision was given to sorter K in the required time.

BCS	FSK	P	2

BRANCH ON FEEDING SORTER K. The sorter on controller P is tested to determine if the sorter is feeding.

BCS	NFK	P	2

BRANCH ON NOT FEEDING SORTER K. The sorter K on controller P is tested to determine if the sorter is not feeding.

Mnemonic Oper Code	Operand Address		Word Time
BCS	ICK	P	2

BRANCH ON INVALID CHARACTER SORTER K. The sorter K on controller P is tested to determine if an invalid character was read.

BCS	VCK	P	2

BRANCH ON VALID CHARACTER SORTER K. The sorter K on controller P is tested to determine if a valid character was read.

BCS	SKE	P	2

BRANCH ON SORTER K ERROR. The sorter K on controller P is tested for error indicator ON.

BCS	SKC	P	2

BRANCH ON SORTER K CORRECT. The sorter K on controller P is tested for error indicator OFF.

OPTIONAL CENTRAL PROCESSOR INSTRUCTIONS

THREE-WAY COMPARE

The instructions in this section relate to an optional unit that provides for decimal addition and subtraction, additional modification word groups, and a three-way compare instruction.

The CAB and DCB instructions described below provide for immediate determination and branching for "greater than", "equal to", and "less than" comparisons.

Mnemonic Oper Code	Operand Address	Word Time
CAB	Y	3 minimum 4 maximum

COMPARE. The contents of the A register are algebraically compared with the contents of memory location Y. If the contents of Y are greater than the contents of A, the next instruction in sequence is executed. If the contents of Y are equal to the contents of A, the computer skips the next instruction and executes the second sequential instruction. If the contents of Y are less than the contents of A, the computer skips the next two instructions and executes the third sequential instruction.

Mnemonic Oper Code	Operand Address	Word Time
DCB	Y	3 minimum 7 maximum

DOUBLE COMPARE. The contents of the A and Q registers are algebraically compared with the contents of memory locations Y and Y + 1. If the contents of Y and Y + 1 are greater than the contents of A and Q, the next instruction in sequence is executed. If the contents of Y and Y + 1 are equal to the contents of A and Q, the computer skips the next instruction and executes the second sequential instruction. If the contents of Y and Y + 1 are less than the contents of A and Q, the computer skips the next two instructions and executes the third sequential instruction. Y must be an even memory location.

DECIMAL MODE ARITHMETIC

SET DEMODE 2

DECIMAL MODE SHIFT. The execution of this instruction permits the following instructions to operate on decimal information: Add (ADD), Double Add (DAD), Subtract (SUB), Double Subtract (DSU), Add One (ADO), Subtract One (SBO). The computer remains in the decimal mode until the execution of a SET BINMODE (Set Binary Mode) instruction returns the arithmetic circuitry to the binary mode. (All other instructions are performed in binary while SET DECMODE is in effect.) Decimal and binary arithmetic word times are identical.

SET BINMODE 2

BINARY MODE SHIFT. The execution of this instruction permits all instructions to operate on binary information. The computer remains in the binary mode until the execution of a SET DECMODE (Set Decimal Mode) instruction puts the arithmetic circuitry in the decimal mode. (The central processor is set to the binary mode when the power is turned on.)

INDEX GROUP SELECT

SXG Y 2

SELECT INDEX GROUP. SXG causes one of the 32 possible modification word groups to become operative. The specific group is designated by Y which is a decimal number from 0 through 31.

AUXILIARY ARITHMETIC UNIT

Three modes of operation are available in the Auxiliary Arithmetic Unit:

1. Unnormalized floating point arithmetic operations.
2. Normalized floating point arithmetic operations.
3. Double word fixed point arithmetic operations.

The command SET selects one of the three modes under program control.

The command BAR permits the interrogation of the Auxiliary Arithmetic Unit for certain conditions. This instruction and the conditions tested are listed under Test and Branch Instructions for the Auxiliary Arithmetic Unit.

Data Transfers within the Auxiliary Arithmetic Unit.

The mnemonics for these commands consist of the normal three alphabetics plus the tag A in the automatic modification field. For example, LAQ A moves the contents of the QX register to the AX register in the Auxiliary Arithmetic Unit while LAQ alone moves the contents of the Q register to the A register in the central computer.

Mnemonic Oper Code	Operand Address	Micro-seconds
LAQ	A	40

LOAD AX FROM QX. The contents of Register QX replace the contents of Register AX. The contents of Register QX are unchanged.

MAQ A 40

MOVE AX TO QX. The contents of Register AX replace the contents of Register QX. AX is cleared to zero.

LQA A 40

LOAD QX FROM AX. The contents of Register AX replace the contents of Register QX. The contents of Register AX are unchanged.

XAQ A 108

EXCHANGE A AND Q. The contents of Register AX and Register QX are interchanged.

Data Transfers Between Memory and the Auxiliary Arithmetic Unit

FLD Y 72

LOAD AUXILIARY STORAGE UNIT. The contents of memory locations Y and Y + 1 replace the contents of Register AX. The contents of Y and Y + 1 are not changed. This command operates the same under all modes.

FST Y 72

STORE AUXILIARY STORAGE UNIT. The contents of Register AX replace the contents of memory locations Y and Y + 1. The contents of Register AX are not changed. This command operates the same under all modes.

Commands Which Set Modes

Before giving an arithmetic instruction to the Auxiliary Arithmetic Unit, it is necessary to set the mode by one of the following SET instructions.

Mnemonic Oper Code	Operand Address	Micro-seconds
SET	UFLPOINT	40

To set the mode for unnormalized floating point operations.

SET	NFLPOINT	40

To set the mode for normalized floating point operations.

SET	FIXPOINT	40

To set the mode for double word fixed point operations.

Arithmetic Operations Within the Auxiliary Arithmetic Unit.

FAD	Y	Min 153 Max 700

NORMALIZED FLOATING POINT MODE. The Floating Point Number in memory location Y and Y + 1 is added algebraically to the Floating Point Number in Register AX. The result is placed in Register AX in normalized form. The contents of Y and Y + 1 are not changed.

Min 130
Max 265

UNNORMALIZED FLOATING POINT MODE. Same as above except the result is placed in Register AX in unnormalized form.

85

DOUBLE WORD FIXED POINT MODE. The contents of Y and Y + 1 are algebraically added to the contents of Register AX. The result is placed in Register AX as a 38 bit fixed point number. The contents of Y and Y + 1 are unchanged. The contents of Register QX are unchanged.

FMP	Y	Min 265 Max 1053

NORMALIZED FLOATING POINT MODE. The Floating Point Number in memory locations Y and Y + 1 is algebraically multiplied by the Floating Point number in Register QX. The 60-bit product of the two mantissas is normalized. The most significant half of the normalized product is stored with its exponent in Register AX. The least significant half of the normalized product is stored in Register QX (with the exponent less by 30 than the Floating Point exponent in Register AX). The previous contents of Register AX are destroyed.

Mnemonic Oper Code	Operand Address	Micro-seconds
		Min 220 Max 580

UNNORMALIZED FLOATING POINT MODE. Same as above except the result is placed in Register AX in unnormalized form.

Min 265
Max 715

DOUBLE WORD FIXED POINT MODE. The fixed point number in memory locations Y and Y + 1 is algebraically multiplied by the fixed point number in Register QX giving a 76-bit product and 4 identical sign bits. The most significant half of the product is stored with 2 sign bits in Register AX, and the least significant half of the product is stored with 2 sign bits in Register QX. The previous contents of Register AX is destroyed.

FDV	Y	Min 805 Max 1205

NORMALIZED FLOATING POINT MODE. The Floating Point Number in Register AX and Register QX is algebraically divided by the normalized floating point number in memory locations Y and Y + 1. The normalized quotient is stored in Register AX and the unnormalized remainder is stored in Register QX. If the contents of Y and Y + 1 are not normalized a Divide Check Error condition will occur when the absolute value of the mantissa in Register AX is equal to or greater than twice the absolute value of the mantissa in the divisor. If the contents of Y and Y + 1 are zero, the division is invalid and a Divide Check Error condition will occur. The sign of the remainder is the sign of the dividend.

Min 783
Max 980

UNNORMALIZED FLOATING POINT. Same as above except the quotient is stored in Register AX in unnormalized form.

Min 1008
Max 1161

DOUBLE WORD FIXED POINT MODE. The contents of Register AX and Register QX are divided algebraically by the contents of memory locations Y and Y + 1. The quotient is stored in Register AX and the remainder is stored in Register QX. The magnitude of the divisor in memory locations Y and Y + 1 must be greater than the absolute value of the dividend in Register AX. If not, a Divide Check Error condition will occur. If the divisor is zero, the division is invalid, and a Divide Check Error condition will be generated. The sign of the remainder is the sign of the dividend.

Mnemonic Oper Code	Operand Address	Micro-seconds
FSU	Y	Min 153 Max 700

NORMALIZED FLOATING POINT MODE. The Floating Point Number in memory location Y and Y + 1 is subtracted algebraically from the Floating Point number in Register AX. The result is placed in Register AX in normalized form. The contents of Y and Y + 1 are not changed.

Min 130
Max 265

UNNORMALIZED FLOATING POINT MODE. Same as above except the result is placed in Register AX in unnormalized form.

85

DOUBLE WORD - FIXED POINT MODE. The contents of Y and Y + 1 are algebraically subtracted from the contents of Register AX. The result is placed in Register AX as a 38-bit fixed point number. The contents of Y and Y + 1 are unchanged.

Auxiliary Arithmetic Unit

Test and Branch Instructions

BAR XXX

BRANCH ON AUXILIARY ARITHMETIC UNIT INTERROGATED CONDITIONS. The BAR instruction interrogates the Auxiliary Arithmetic Unit for specific conditions. The condition tested is indicated by a mnemonic placed in the operand field and indicated by XXX above. If the condition tested is true, the computer executes the next sequential instruction. If the condition tested is false the computer skips the next instruction and executes the second sequential instruction. The mnemonics and the conditions tested are:

BAR BAN 2

BRANCH ON AAU NOT READY. The AAU is tested to determine if it is not ready to accept another instruction.

BAR BAR 2

BRANCH ON AAU READY. The AAU is tested to determine if it is ready to receive another instruction.

BAR BPL 2

BRANCH ON AAU PLUS. Register AX is tested for a plus sign.

BAR BMI 2

BRANCH ON AAU MINUS. Register AX is tested for a minus sign.

BAR BZE 2

BRANCH ON AAU ZERO. Register AX is tested for all zero content.

Mnemonic Oper Code	Operand Address	Micro-seconds
BAR	BNZ	2

BRANCH ON AAU NOT ZERO. Register AX is tested for non-zero content.

BAR BOV 2

BRANCH ON OVERFLOW. The AAU is tested for overflow indicator ON.

BAR BNO 2

BRANCH ON NO OVERFLOW. The AAU is tested for overflow indicator OFF.

BAR BUF 2

BRANCH ON UNDERFLOW. The AAU is tested for underflow indicator ON.

BAR BNU 2

BRANCH ON NO UNDERFLOW. The AAU is tested for underflow indicator OFF.

BAR BER 2

BRANCH ON ERROR. The error indicator is tested for ON.

BAR BNE 2

BRANCH ON NO ERROR. The error indicator is tested for OFF.

PRIORITY PROGRAM AUTOMATIC INTERRUPT

Under program control the computer is set in the automatic interrupt mode and a signal from any of the eight controllers, card punch, or card reader may interrupt the execution of a main program. When this interrupt occurs the index group currently being used by the program is changed to index group 32. The present setting of the P counter is stored in index word 1 of group 32. Program control is then transferred to the instruction at the location immediately following index group 32.

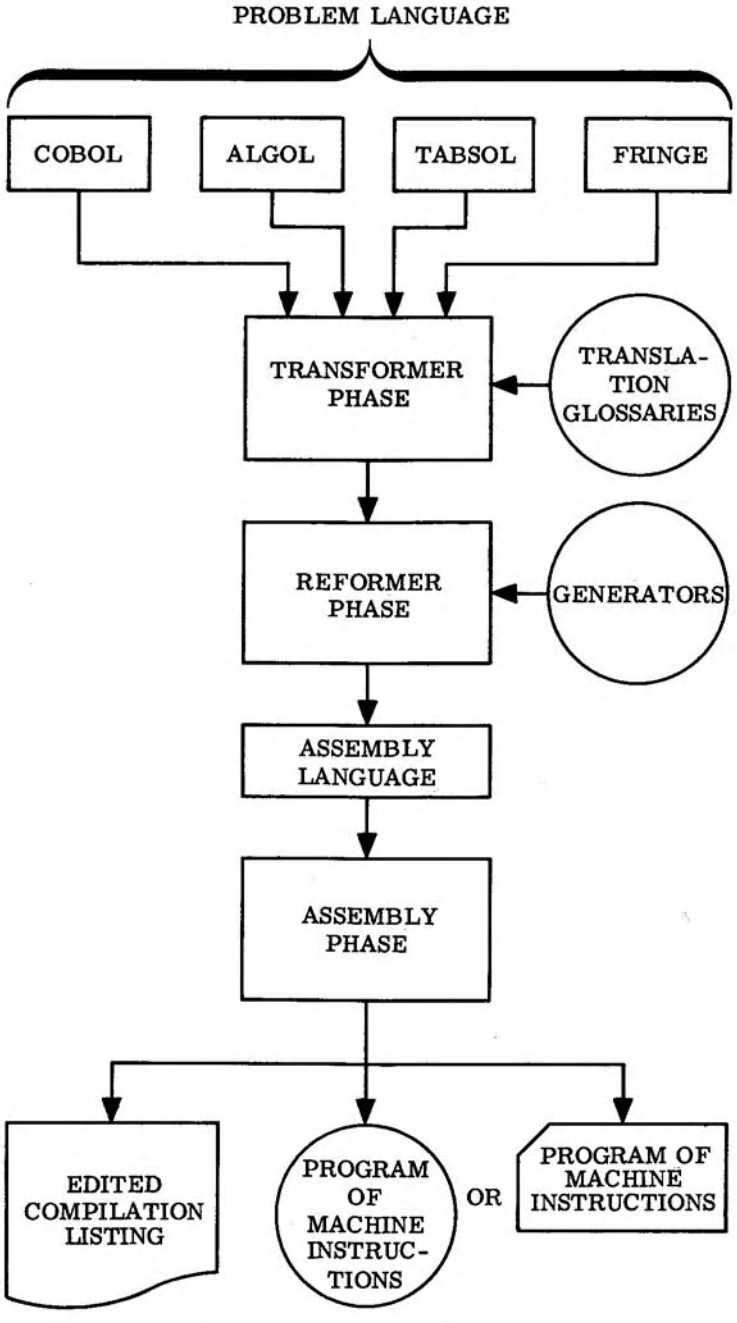
		Word Time
SET	PST	2

SET PRIORITY INTERRUPT ON. This instruction causes the computer to enter the interrupt mode where it will remain until the priority program is completed, and directions are given for return to the main program.

SET PBK 2

SET PRIORITY INTERRUPT OFF. This instruction nullifies the interrupt mode where it will remain until it is reset to the interrupt mode.

STRUCTURE OF GECOM



PROGRAMMING AIDS

GECOM — THE GENERAL COMPILER

GECOM, the General Compiler developed for the GE 225 Information Processing System introduces a fresh, versatile approach to computer communication. It makes available various proved, and some newly developed, programming techniques in one package. It accepts many languages so problem statements may be written in familiar terminology. The four source languages of the General Compiler are general and comprehensive.

GECOM processes English language statements (COBOL), Algebraic expressions (ALGOL), Structured decision tables (TABSOL), and the File and Report Information Processing Generators (FRINGE). The user may select only that portion of the system applicable to his needs disregarding all other capabilities; or he may select one, two, or any combination of the language features for any specific program run. Since the machine coding is derived directly from the logic of the problem statement, it is only at the logic level that debugging may have to be done.

Because GECOM problems are written in familiar languages, they can be more easily read and understood. In addition, program format provides a high degree of standardization.

Personnel training time and expense is sharply reduced since the novice programmer may use the familiar phrases of his profession. Manual coding is eliminated and debugging cut to a minimum. Thus, a machine program may be produced much faster and much more efficiently.

ADVANTAGES OF GECOM

GECOM is exclusively a General Electric product! The selected approach allows the user to accommodate most of the common languages and still to incorporate later changes conveniently. Several distinct advantages over manual programming methods can be realized.

GECOM automatically produces a documented record of the program it produces. A permanent record of the program, in its original source language form, together with a detailed listing of its transformation to machine instruction is available for reference, revision, or augmentation.

Since the language does not dictate the design of GECOM, advances in automatic coding languages can be readily adapted to the system. Special functions peculiar to particular installations can be added with the same comparative ease.

Since plans call for implementing GECOM on the General Electric family of computers, programming conversion costs are reduced as installations outgrow their present computer equipment.

THE TOOLS OF THE GENERAL COMPILER

COBOL — THE LANGUAGE OF THE BUSINESS MAN

COBOL (Common Business Oriented Language) is the language developed as a result of a Department of Defense sponsored conference on Data Systems Languages. Produced by computer manufacturers and users, its purpose is to achieve machine compatibility and standardization of data processing languages. COBOL reduces programming effort and achieves a more effective utilization of computers. The language first available with the General Compiler is based primarily on COBOL since it satisfies the needs of the broadest spectrum of data processing applications.

COBOL is so close to English syntax that it can easily be read and understood by management, systems and accounting personnel, as well as by the experienced programmer. As a result, close coordination between management and computer application is both practical and efficient. The language structure falls into three categories: procedure statements, data description, and environment description. Procedure statements tell how to manipulate the data; data descriptions name and explain the elements of information. Together these two categories define the individual problem. The environment description provides a link between the data involved and the hardware media. In this way maximum compatibility among present day computers is achieved.

With GECOM, COBOL is supplemented by other languages for even greater flexibility and scope. For the user who is interested in machine compatibility, GECOM accepts pure COBOL. For applications that extend beyond business data processing, GECOM provides for the acceptance of other source languages.

ALGOL — THE LANGUAGE OF THE SCIENTIST, ENGINEER, AND MATHEMATICIAN

Persons familiar with COBOL recognize that it is well suited for creating and processing information contained in data files. In contrast, ALGOL provides

an excellent means for expressing the mathematics and logic associated with scientific applications. Prompted by a growing interest in a standardized notation for numerical methods for computers, an international group studied the problem. ALGOL (ALGOrithmic Language) evolved from this effort - and has proved to be far superior to any of its predecessors. ALGOL enjoys the first wide-spread acceptance and respect accorded a computer language. The notations are accepted internationally in numerical methods; textbooks and university classes have adopted the notations.

ALGOL is an algebraic formula language whose purpose is to describe, in mathematical terms, computational processes. The basic concept used for description of calculating rules is the well-known arithmetic expression containing, as constituents, numbers, variables, and functions. By applying rules of arithmetic composition, such expressions compound self-contained units of the language -- explicit formulae-called assignment statements. For the communication of algoriths and effective computing techniques, ALGOL is a powerful tool. A program written in this language can be published directly by interested parties without expensive conversion and recording.

GECOM will accept ALGOL. This allows scientists and engineers to combine ALGOL with the powerful input, output, and editing features of COBOL. Thus united, GECOM accommodates an even greater range of problems.

TABSOL — THE LANGUAGE OF DECISION MAKING

Recent investigations by General Electric's Manufacturing Services have uncovered an area of applications which requires the solution to an unwieldy number of sequential decisions but does not involve extensive data file processing or profound mathematics. A major obstacle to the effective use of computers in industry has been a lack of efficient methods for thinking through and recording the logic of complex information systems. To overcome this obstacle, and to cope with the associated decisions, TABSOL, the Tabular Systems Oriented Language, was developed.

TABSOL depicts, by means of tables, the relationships of logical decisions which are written in terms of the conditions to be satisfied and the subsequent action to be taken. The TABSOL structure table provides a readable, understandable table of decisions. A horizontal double line separates the parameters from their corresponding table values. A vertical double line separates the conditions from the actions. The table is evaluated from left to right and can be compared to a sentence which reads "IF condition, THEN action."

TABSOL encompasses both scientific and business applications. GECOM accepts COBOL and ALGOL within the framework of the table, thus providing an even more efficient method for stating the logic of complex information systems.

TABSOL has been used widely by General Electric to analyze and solve problems in many fields, such as manufacturing methods, cost accounting, product engineering, and production control. TABSOL may also be used to specify the transfer of control associated with the values of one or more fields and to control the printing of detailed and summary lines of a report. As new applications are discovered, the versatility of the system becomes more pronounced.

FRINGE — THE LANGUAGE OF REPORT GENERATION

In the business world, the work of incorporating data from various sources into master files of information, and extracting these data in various sequences into printed reports constitutes the major work of most computer programs. File maintenance, report, and sort and merge generators work from a precise problem definition. These can take the form of extraction of file information onto printed reports, updating a master information file from current transactions, or merging or rearranging file data preparatory to the main tasks of file updating and reporting. The final, detailed program follows a fixed logic framework tailored to the specification of your specific problem.

The primary value of the generator languages comes from the statement of implied rather than explicit procedures. One of the more important features is a method of layout and definition of the report as a whole on a form the size of a printed page. The structure of each unique type of line (heading, detail, total, footing) is displayed along with specifications for line and page control and data editing.

As an extension to the General Compiler, this report, file maintenance, and sort generator is provided to further ease the task of programming business data processing problems.

GENERAL ELECTRIC

GENERAL COMPILER
SENTENCE FORM

PROGRAM: SAMPLE CUSTOMER ORDER RUN USING A COMBINATION OF COBOL, TABSOL, AND ALGOL — DATE: 1/20/61
PROGRAMMER — COMPUTER: GE 225 — PAGE: 3 OF 9

SEQUENCE NUMBER	
305	OPEN INPUT MASTER~SPEC, CUST~SPEC, AND PARAMETER FILES.
310	READ PARA~CARD RECORD.
315	GET~SPEC. READ CUST~SPEC RECORD, AT END FILE GO TO END~ROUTINE.
320	READ MASTER~SPEC RECORD UNTIL ORDER~NO OF MASTER~SPEC EQUALS
325	ORDER~NO OF CUST~SPEC, AT END FILE GO TO END~ROUTINE.
330	IR = SQRT (A**2 + B**2). MOVE A1 TO A. MOVE B1 TO B.
335	K~SPEC TABLE. 3 CONDITIONS, 3 ACTIONS, 4 ROWS.

COBOL

	K EQ	IR EQ	LOT~NO EQ	DRWG~NO	HOLE~DIA	PERFORM TABLE
340						
345	0.0763	0.00761	"AB33"	"5007AB33"	0.77	COST~01
350	1.1127	0.3451	"CV33"	"5016CV33"	1.34	COST~07
355	2.9001	0.7942	"FE331"	"5020FE331"	1.99	COST~09
360	3.7667	0.81175	"AL331"	"5024AL331"	2.09	COST~10

TABSOL

365	IF K~SPEC TABLE NOT SOLVED, DISPLAY "K~SPEC N.S." ON TYPEWRITER.
370	PERFORM AREA~P.
375	SPEC~CALC. AREA(J) = P~AREA
380	IF AREA(J) EQ D~AREA(L(I*3),M(Q+N,Z)) OR HOLE~WD NGR 3.012E-5 THEN
385	PDZ = (A*B)**3 - SQRT AREA(J).
390	IF J EQ Q-1 THEN GO TO GET~SPEC
395	J = Q.
400	GO TO COST~ADJUST.

ALGOL

Computer Department, Phoenix, Arizona CD13(2/60)

THE GENERAL ASSEMBLY PROGRAM

GENERAL DESCRIPTION

The General Assembly Program allows the programmer to work with the individual instructions of the GE 225 computer. However, with this assembly program the programmer employs symbolic notation rather than the absolute code of the computer. The symbolic notation selected to designate each instruction is a mnemonic code. These codes are carefully chosen to provide maximum significance to the user. For example, the mnemonic code for the addition instruction is ADD, for the subtraction instruction the code is SUB. The assembly program translates these mnemonic codes into the absolute code of the computer.

Memory addresses may be assigned by using decimal notation (location 1500 for example) or by using symbolic notation chosen for maximum convenience to the particular program or programmer. For example, in a payroll program the symbolics (NETPAY) or (FICA) might be used to designate memory locations. The General Assembly Program automatically assigns the memory locations in the language of the computer. This program also provides for the assignment of addresses relative to some starting point. Assume for example that the symbolic (AMOUNT) is established as being equal to location 500. Using the technique of relative addressing, memory location 510 can be addressed by simply writing (AMOUNT + 10).

In addition to the mnemonic codes for the instructions of the GE 225, and the symbolic addresses for data and the address of other instructions, the General Assembly Program uses mnemonic codes called pseudo-instructions. These instructions are used by the General Assembly Program for the assembly of a program but never appear in the actual instructions assembled. For example, ORG is a pseudo-instruction which is used to indicate the starting address in the assignment of a program to memory. Thus, ORG 400 indicates That a program is to enter memory with the first instruction starting at decimal location 400. DEC is a pseudo-instruction which is used to indicate decimal data which a program might use as constants. The pseudo-instruction has the same general form as a computer instruction in the General Assembly Program and it is listed like a normal instruction in the preparation of the machine program listing, but is never executed by the computer as an actual instruction. See page 47 which illustrates the ease with which programs are coded using the General Assembly Program.

EXTENSIONS OF THE GENERAL ASSEMBLY PROGRAM

To extend the use of the General Assembly Program the programmer can call on various subroutines as required by the program. For example, if a program requires the square root of a number the programmer need only write a SPB instruction and name the routine and the data which the square root routine will use. The General Assembly Program will assemble this routine as part of the program. Detailed descriptions of the standard routines available are covered in a manual (GE 225 UTILITY ROUTINES). Also, routines unique to a given application can be written and included in the General Assembly Program by the user.

Also to help the programmer are input-output library routines. These routines perform all the functions required to get data into or out of the central processor with only one statement written by the programmer. For example, when the tape input-output routine is called on to read tape, it will test to see that the tape unit is ready, read the information from tape into core storage, test for a read error and branch to a corrective routine if an error should occur, and test for end of file. This routine will also open the tape file (read a tape label to insure that the proper tape file is on the tape unit).

These input-output routines greatly reduce the coding effort required for many applications which use a large selection of peripheral devices. The ability to easily include a variety of subroutines makes the General Assembly Program a powerful programming aid.

GENERAL ELECTRIC 225 GENERAL ASSEMBLY PROGRAM CODING SHEET

PROGRAMMER | PROGRAM EXAMPLE | DATE 1/20/61 | PAGE 1 OF 1

Symbol	Opr	Operand	X	REMARKS	Sequence
	ORG	400			
	DEC	3.14159			
START	ADD	NETPAY			
	SUB	FICA + 5			
	BRU	START			

OTHER PROGRAMMING AIDS

In addition to the powerful General Compiler and General Assembly Program, many other programming aids are available with the GE 225.

UTILITY ROUTINES

1. Memory Loaders
2. Memory Dumps
3. Input/Output Routines
4. Trace Routines
5. Internal Sort Routines
6. Decimal to Binary Conversion
7. Binary to Decimal Conversion
8. Random Number Generators
9. Tape Comparator and Corrector Routines
10. Print Routines
11. Card to Tape
12. Tape to Printer

MATH ROUTINES

1. Statistical Routines
2. Matrix Routine
3. Linear Programs
4. Programmed Arithmetic
   Programmed floating point package
   Double Length multiplication
   Double Length division
5. Math Functions:

   Sine A, fixed point, floating point
   Cosine A, fixed point, floating point
   Tangent A, fixed point, floating point
   Arc Tangent A, fixed point, floating point
   Exponential, fixed point, floating point
   Log Base K of X, fixed point, floating point
   Square Root, fixed point, floating point

SORT AND MERGE GENERATOR

Large volume data sorting can be done efficiently on GE 225 systems that have three or more magnetic tape units on line with the computer. All the computer instructions necessary to do a particular magnetic tape sorting operation can be produced automatically from the GE 225 Sort and Merge Generator Routine. The sorting routine produced by the Generator will take advantage of any ascending sequence that may exist in the input data to reduce the sorting time. It will also allow for special processing of the data on both input passes and the final output pass of the sort.

REPORT GENERATOR

The principle value of report generation is the ease and speed of report program preparation. The extraction of printed reports is one of the primary functions of any file processing system. Programs that create reports in the business world follow common logic patterns. With report generation, programmers need not repeat similar detailed program steps for each individual report. Instead, a concise report description is fed to the Generator which adapts the basic report logic to the individual specifications received. This approach minimizes programming and debugging effort and provides readily understandable program documentation.

The description of a report includes a layout or visual image of the report as a whole and an associated definition of the variable data elements to be displayed. Special codes and names have been adopted to convey meaning to the Generator. Logical and arithmetic expressions can be included to control the selection of file data, the issue of report lines and the calculation of report values. A Report Description Form has been designed for all types of entries.

The report generator operates with or without the procedural elements which are now a necessary part of a General Compiler source program. In the first case, the languages of GECOM (ALGOL, COBOL, TABSOL) are used to control the logic flow of the file processing. The report program functions primarily as an output subroutine. Without procedural elements, the generator supplies the total file processing function. The procedure is derived from specifications relevant to input files and output reports, and the basic assumption of sequential record processing.

650 SIMULATOR

This program, in GE 225 machine language, accepts 650 programs and data as inputs, selects and executes the required routines to simulate the 650 computer commands, and produces the same results and outputs as the 650 computer.

The simulator program for the GE 225 achieves the following objectives without loss of accuracy or flexibility:

1. The basic 650 with 2000 words of drum memory, one 533 card reader and punch with alphabetic device is simulated. The simulator program runs on the GE 225 having 8192 words of memory, card reader and card punch, and typewriter.

2. The simulator executes 650 production programs during changeover to the GE 225 computer. Infrequently used programs need not be recoded in GE 225 machine language.

3. The simulator is designed to eliminate programmer intervention. A program written for the 650 configuration requires no changes for simulator execution. Control cards preceding the 650 program deck define plugboard wiring and console switch settings.

4. The simulator is constructed so that it can be modified to include other 650 configurations or features with a minimum of programming effort. Documentation is detailed and complete so that features peculiar to certain applications may be readily incorporated.

Other Programming Media are being made available. Some of these include the FORTRAN to GECOM translator and tape executive service system. Other standard programs will be made available through the GE 225 Users Group.

APPENDIX

REPRESENTATION OF CHARACTERS

CHARACTER	HOLLERITH CODE (PUNCH IN ROWS)	BCD MEMORY (OCTAL)	BCD MAGNETIC TAPE (OCTAL)	HIGH SPEED PRINTER SYMBOLS	CONSOLE TYPEWRITER CHARACTER OR ACTION	PAPER TAPE CHARACTER (8 CHANNEL)
0	0	00	12	0	0	SPACE
1	1	01	01	1	1	1
2	2	02	02	2	2	2
3	3	03	03	3	3	3
4	4	04	04	4	4	4
5	5	05	05	5	5	5
6	6	06	06	6	6	6
7	7	07	07	7	7	7
8	8	10	10	8	8	8
9	9	11	11	9	9	9
A	12-1	21	61	A	A	/
B	12-2	22	62	B	B	S
C	12-3	23	63	C	C	T
D	12-4	24	64	D	D	U
E	12-5	25	65	E	E	V
F	12-6	26	66	F	F	W
G	12-7	27	67	G	G	X
H	12-8	30	70	H	H	Y
I	12-9	31	71	I	I	Z
J	11-1	41	41	J	J	J
K	11-2	42	42	K	K	K
L	11-3	43	43	L	L	L
M	11-4	44	44	M	M	M
N	11-5	45	45	N	N	N
O	11-6	46	46	O	O	O
P	11-7	47	47	P	P	P
Q	11-8	50	50	Q	Q	Q
R	11-9	51	51	R	R	R
S	0-2	62	22	S	S	B
T	0-3	63	23	T	T	C
U	0-4	64	24	U	U	D
V	0-5	65	25	V	V	E
W	0-6	66	26	W	W	F
X	0-7	67	27	X	X	G
Y	0-8	70	30	Y	Y	H
Z	0-9	71	31	Z	Z	I
+	12	20	60	+		0
-	11	40	40	–	–	–
Δ	BLANK	60	20	BLANK	BLANK	&
/	0-1	61	21	/		A
	2-8	12				
#	3-8	13	13	#	/	STOP
@	4-8	14	14	@		
–	5-8	15		–		
=	6-8	16		=		
	7-8	17				
	12-2-8	32	72			
.	12-3-8	33	73	.		
	12-4-8	34	74			
	12-5-8	35				
	12-6-8	36				TAB
	12-7-8	37			CARRIAGE RETURN	
	11-2-8	52	52			
$	11-3-8	53	53	$	$	$
*	11-4-8	54	54	*		
	11-5-8	55				
	11-6-8	56				
	11-7-8	57				
	0-2-8	72	32		PRINT RED	
,	0-3-8	73	33	,	,	
%	0-4-8	74	34	%		
[	0-5-8	75		[	PRINT BLACK	
]	0-6-8	76		]	TAB	
	0-7-8	77				DELETE

TABLE OF POWERS OF 2

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125
1 099 511 627 776	40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5

GE 225 SYSTEM INSTALLATION DATA

COMPONENT SIZES

	Size (Inches) Floor Size	Height
Central Processor with 4K or 8K Memory	117 x 32	76
Console including Typewriter and Card Reader	88 x 48	30
Card Punch	41 x 26	50
Printer	40 x 25-1/2	52
Printer Controller	40 x 32	76
Magnetic Tape Handler (single or dual)	24 x 32	76
Magnetic Tape Controller	40 x 32	76
Document Sorter (12 pocket)	183-1/4 x 33	50-1/4
Document Sorter (2 pocket)	64 x 33	41
Document Sorter Controller	40 x 32	76
Auxiliary Arithmetic Unit	80 x 32	76
Mass Random Access File Memory	84 x 36	60
Mass Storage Controller	40 x 32	76
Mass Storage Electronics	40 x 32	76
Paper Tape Punch Paper Tape Reader	38 x 30	62
8K Core Memory	40 x 32	76

ELECTRICAL REQUIREMENTS

The GE 225 is designed to operate from a standard 208Y/120 volts, 4-wire 3-phase, 60-cycle AC source of electrical power. A 230-volt, single-phase, 3-wire power supply can also be used. If a document sorter or a mass random access file memory is used, a 208-volt, 3-phase power must be supplied.

ENVIRONMENTAL REQUIREMENTS

The ambient room temperature within the machine area must remain within range of 65 to 85 degrees F. under all conditions of operation. Any area in which tapes are stored or handled must also meet these requirements. The recommended temperature for optimum operation is approximately 72 to 78 degrees. The relative humidity must remain within a range of 30 to 50 percent.

INFORMATION PROCESSING
AUTOMATED BY GENERAL ELECTRIC

Progress Is Our Most Important Product

GENERAL ELECTRIC

COMPUTER DEPARTMENT, PHOENIX, ARIZONA

CPB 98A (10M 7-61)

Printed in U.S.A.