CONTROL DATA

160G

PROGRAMMING – REFERENCE MANUAL

CONTROL DATA®

160G

PROGRAMMING – REFERENCE MANUAL

G02000c
11 May 1965

RECORD OF REVISIONS	
REVISION	NOTES
a	Removal of preliminary status
b	ECO 14519. Pages iv, vii, 2-3, 2-34, 2-40A, 2-41,
	2-42, 2-42A, 2-43, 2-44, 2-61, 2-62, 2-65, 2-73,
	D-1, D-5, and E-3 revised.
c	ECO 15736. Pages v, 1-6, 2-36 through 2-39, 2-49,
	2-64, 2-71, 2-72, 3-3, 3-6, 3-9, 3-11, 4-12, E-1,
	E-2, E-3, and Comment Sheet revised.

CONTENTS

APPENDIXES

TABLES

ILLUSTRATIONS

CHAPTER 1

DESCRIPTION

The CONTROL DATA® 160G is a high-speed, modular computing system suited to real-time control and data processing applications. The flexible characteristics of the 160G system permit it to be expanded into a system with large-scale storage, input/output, and computational capabilities. With several options available, the 160G can be tailored to meet particular data needs and later grow with increased demands for data processing.

BASIC 160G CHARACTERISTICS

- Normal or buffered input-output
- Internal and external interrupts
- An expandable magnetic core memory (8192 to 131,072 words)
- 14-bit storage word (13 information bits, 1 parity bit)
- Transistor-diode logic
- Single address logic
- Flexible repertoire of instructions
- 62 index registers (magnetic core)
- Program compatibility with CONTROL DATA 160 and 160-A

The 160G system can be expanded to include the following peripheral equipment:

- Electric typewriters
- Card readers, card punches, and associated control units
- High-speed line printers
- Incremental plotters
- Magnetic tape transports and tape synchronizers
- Provision for reciprocal memory sharing with other computer systems.

The 160G system can be tied, magnetic core to magnetic core, to certain other Control Data systems. The magnetic core memory of the 160G becomes an internal part of the other system or vice versa.

SYSTEM DESCRIPTION

The 160G computer system is a data processing device which is expandable on a modular basis. A minimum system consists of a single 160G Compute Module and associated Console. The Compute Module is a parallel, single address, electronic data processor. It performs calculations and processes data internally in a parallel binary mode. The internally stored program located in sequential storage locations controls the step-by-step execution of individual instructions. Individual instructions in a Compute Module are executed in one to six storage cycle times where one storage cycle is 1.35 microseconds.

The 160G Compute Module, which functions as a complete small-scale computer in the minimum system, provides the arithmetic and control characteristics of the expanded 160G system. Each Compute Module consists of a memory section, which contains 8192 words of magnetic core storage; arithmetic and control logic; and two bidirectional I/O channels, one normal and one buffered.

The Compute Module also contains circuitry for two internal and 21 external interrupts. One of the internal interrupts is manually activated at the Console. The other internal interrupt is activated at the end of a buffering operation on the internal buffer channel.

The 171G I/O Module provides additional data channels for the 160G system. Each module contains two buffered, bidirectional data channels; interrupt circuitry; external function circuitry; and associated control logic. Each of the buffered data channels has three external interrupts associated with it: buffer completion, external 1, and external 2. Any external equipment serviced by the data channel is capable of activating either of the external 1 or 2 interrupts. The 171G I/O Module requires information from the Compute Module to initiate I/O activity.

The 169G Memory Module provides additional core storage for the 160G system. Each module contains storage for 8192 14-bit words. Each Memory Module consists of one 8192-word bank of core storage, priority circuitry, parity circuitry, and memory control logic. The parity circuitry in a Memory Module checks the odd parity of all data read from the core storage within that module. This circuitry also generates a parity bit for all data to be written into core storage within that module. If an error occurs, a parity error signal is sent to the module requesting access.

The expanded 160G system may consist of a number of Compute Modules, additional banks of memory, and additional input/output channels. The combination of any number of the system components may be combined in an expanded system subject to the following limitations:

1. A maximum of three I/O modules can be controlled by one Compute Module.
2. A maximum of nine modules (Compute and/or I/O) can be connected to one Memory Module.
3. A maximum of 15 Memory Modules can be controlled by one Compute Module.

REGISTERS

Temporary storage units for operands, instructions, and control words are known as registers. Those registers available to the programmer by means of computer instructions are called addressable registers; the other registers are called non-addressable registers, Figure 1-1. The contents of all but one of the registers can be displayed on the Console.

ADDRESSABLE REGISTERS

A REGISTER (A)

Nearly all arithmetic and logical operations involve the 13-bit A register. The contents of this register can be shifted to the right or left. The results of logical or arithmetic operations appear in the A register. When the 160G is in the A mode, the highest-order stage is not used.

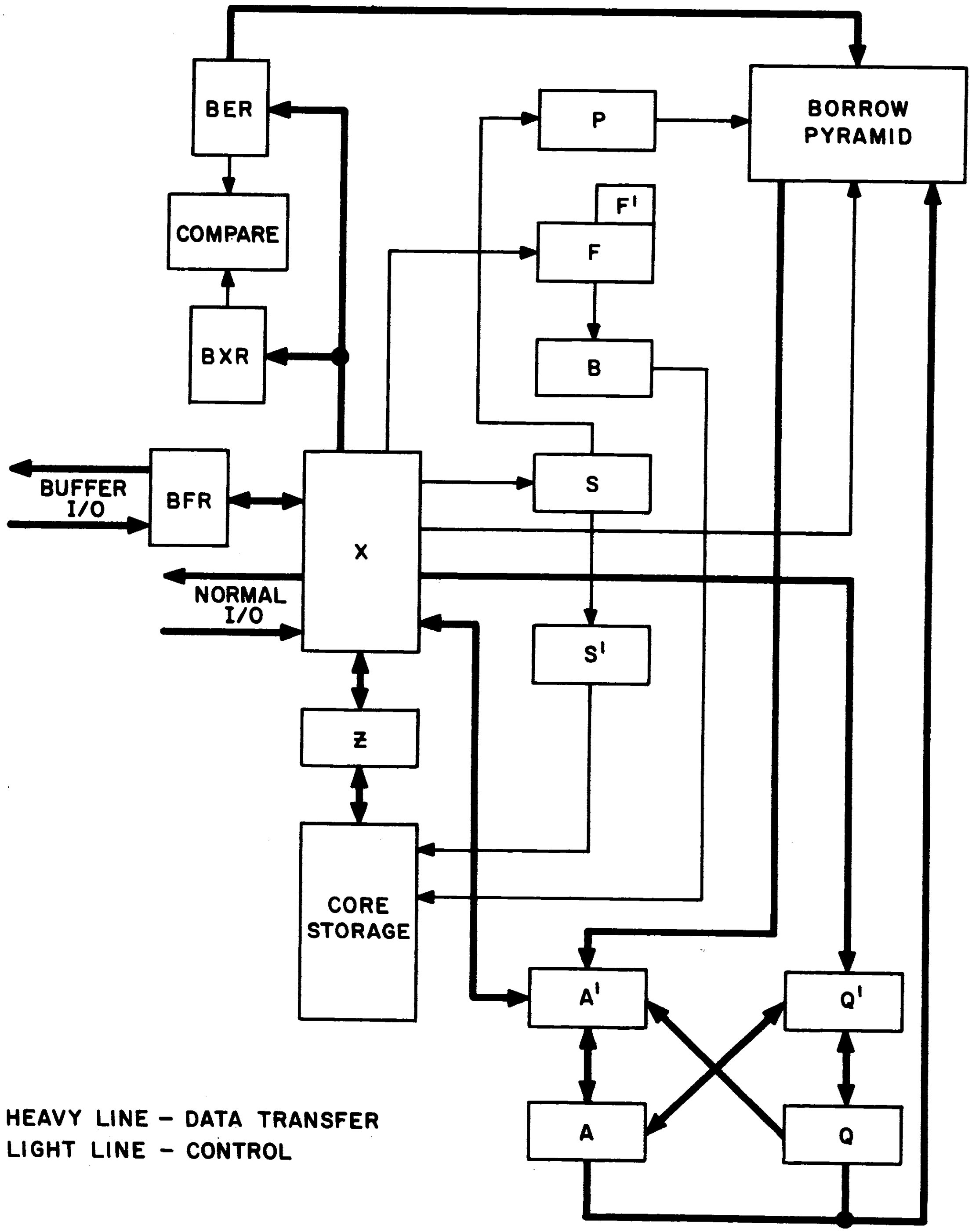


Figure 1-1. Block Diagram of Registers

AUXILIARY ARITHMETIC REGISTER (Q)

The 13-bit Q register assists the A register in performing arithmetic and logical operations. For multiply and divide instructions Q serves as an extension of the A register to form a 26-bit AQ register.

PROGRAM ADDRESS REGISTER (P)

The 13-bit P register contains the memory address of the current instruction. In the A mode, the highest-order bit is not used. After execution of an instruction which does not transfer program control to another routine, the contents of the P register is increased by 1, 2, or 4 as determined by the type of exit. If program control is transferred to a new routine, a new control address is entered into the P register, and the contents of the P register is not changed.

The advancing of the P register is done by a one's complement, subtractive arithmetic. If the 160G is operating in the G mode and the P register contains octal 17776, advancing the contents by one results in 00000; if P contains octal 17777, advancing the contents by one results in 00001. If the 160G is operating in the A mode and the P register contains octal 07776, advancing the contents by one results in 00000. If P contains octal 07777, advancing the contents by one results in 00001.

BUFFER ENTRANCE REGISTER (BER)

During buffered I/O operations, the 13-bit BER holds the address to, or from which the information is being transferred. The contents of this register may be transferred to memory or the A register. In the A mode, only the lower-order 12 bits are used.

BUFFER EXIT REGISTER (BXR)

During buffered I/O operations, the 13-bit BXR holds the terminal address plus one for the buffering operation. When the contents of the BER equals the contents of the BXR, the buffering operation is terminated. In the A mode only the lower-order 12 bits are used.

ERROR REGISTER (ER)

The error register is a 13-bit register which contains bits that represent various error conditions. Only ten error conditions are displayed in the error register. These are available for display by issuing an ERTA instruction. This register indicates add or subtract overflow, parity errors on the buffer I/O channels, and a divide overflow condition.

NON-ADDRESSABLE REGISTERS

MEMORY RESTORATION REGISTER (Z)

The 14-bit Z register restores any information read from memory after this information has been stored in the appropriate Compute Module register or transmitted to an output device. All information entering Compute Module memory passes through the Z register. Each Memory Module also contains a similar register which services core storage within that module.

MEMORY ADDRESS REGISTER (S)

The 13-bit S register holds the memory address currently being referenced for an instruction or operand. This address may apply to Compute Module memory or locations within any of the Memory Modules. Only 12 bits of the S register are used in the A mode.

INTERNAL MEMORY ADDRESS REGISTER (S')

The 13-bit S' register is similar to the S register. However, S' services only Compute Module memory. When the current address references Compute Module memory, S' holds the address until the reference is completed. Each Memory Module contains a similar register which services core storage within that module.

FUNCTION REGISTER (F)

The 13-bit F register contains the instruction being executed.

SHIFT COUNT REGISTER (F')

The 5-bit F' register controls the shift count and holds the bank designations.

BUFFER DATA REGISTER (BFR)

During a buffer operation, the 13-bit BFR holds the data word being transferred to or from memory.

ADDER OUTPUT REGISTER (A')

The 13-bit A' register serves as a second rank of the A register for shifting and transferring purposes. This register also functions as an output register for the Compute Module adder (the borrow pyramid).

SECONDARY AUXILIARY ARITHMETIC REGISTER (Q')

The 13-bit Q' register functions as the second rank of the Q register for shifts and transfers.

STORAGE BANK CONTROL REGISTER (B)

The 20-bit B register consists functionally of four 5-bit bank controls. These controls operate independently of one another to provide the addressing modes available to the 160G system.

EXCHANGE REGISTER (X)

The 13-bit X register functions as an exchange register for the transfer of data within the Compute Module.

CHAPTER 2

PROGRAMMING

ARITHMETIC

The 160G is capable of performing arithmetic in a 12-bit (A mode) or 13-bit (G mode) operation. This 12-bit arithmetic feature makes the 160G compatible with the 160-A. Normally the computer enters the G mode after a master clear. However, the selection of either mode can be made manually or by a programmed entry. Any instruction in the repertoire can be performed in either mode, with the following exceptions:

1. LS3, LS6, MUT, and MUH always operate on 12-bit operands.

2. LS3 and LS6 are circular 12-bit, end-around shifts which leave the highest-order bit unchanged.

3. Multiply, divide, and variable shift instructions are always 13-bit operations.

In the A mode, addition, subtraction, addressing, and logical operations are 12-bit; in the G mode these operations are 13-bit.

The bits within a word are numbered from 00 (least significant) to 12, starting at the right. Bit 12, the most significant bit, is not used in the A mode.

Bit 11 for A mode of operation and bit 12 for G mode of operation are sign bits. All positive numbers have a 0 in the sign bit position; all negative numbers have a 1 in the sign bit position. The sign bit is extended to the most significant bit of a number.

<u>Example:</u>

<table>
<tr><th></th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td>Binary</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>Octal</td><td>1</td><td colspan="3">3</td><td colspan="3">6</td><td colspan="3">1</td><td colspan="3">5</td></tr>
</table>

All arithmetic is binary, one's complement notation. Any number may be represented as a combination of the two binary digits, 0 and 1. Although the computer is operated in the binary system, the octal representation of a binary number is more convenient for notation and display. The 160G word can be considered as five octal digits, with the uppermost octal digit being 0 or 1.

The value zero can be represented in one's complement notation in two separate expressions:

0	000	000	000	000	(plus zero)
1	111	111	111	111	(minus zero)

Both plus and minus zeroes are acceptable as arithmetic operands. There are only two cases in which a zero arithmetic resultant is minus zero; in all other cases, the resultant is plus zero. These two cases are as follows:

$$-0 + (-0) \text{ and } -0 - (+0)$$

In the A mode, with 12-bit arithmetic, the value of the highest-order bit is considered as undefined. When operating in the G mode (13-bit) on a quantity that was generated in the A mode (12-bit), the correct sign on the quantity can be guaranteed by doing an LS1-RS1 sequence of instructions to generate the correct sign bit.

When switching from one mode to another, the action of the P register is also changed. The following actions occur in the 160G to guarantee that program control continues correctly after a switch in modes. When switching from A to G mode, bit 00 of the relative bank control replaces bit 12 of the P register. When switching from G to A mode, bit 12 of the P register replaces bit 00 of the relative bank control. The mode should not be switched when the program is at location 07776, 07777, 17776, or 17777 of any bank, because P + 1 is performed in the mode prior to the switch; therefore, an incorrect next address would be formed. For example, when switching from A to G mode with program control at location 07777, the next address would become 00001 instead of 10000.

In one's complement notation, positive numbers are represented by their binary equivalent; negative numbers are represented by the one's complement of the equivalent positive number. To form the one's complement, reverse each bit of the word.

Example:

+5 is represented as	0	000	000	000	101
-5 is represented as	1	111	111	111	010

The internal arithmetic is based on subtraction. Addition is performed by subtracting the complement of the addend from the augend. In subtraction no complementing is necessary.

Example:

If the computer is programmed to add +6 to +5, the operation is performed as follows:

Augend +5 =	0	000	000	000	101	
Addend +6 =	0	000	000	000	110	
Complementing +6 =	1	111	111	111	001	
+5 =	0	000	000	000	101	
- (-6)=	1	111	111	111	001	
	0	000	000	001	100	(with an
Subtracting the borrow	-				1	end-around borrow)
Procedures	0	000	000	001	011	$= 13_8$ or 11_{10}

During the subtraction process, the borrow from the higher-order end was carried around and subtracted from the lower-order end of the word to provide the correct result. This end-around borrow is the feature which makes the arithmetic of the 160G computer modulus $2^{12}-1$.

INSTRUCTION WORD FORMAT

An instruction word in the 160G may be 13 or 26 bits. The 26-bit instruction word is the most general. The 13-bit instruction word is provided for efficient programming.

The 26-bit instruction is divided into a 7-bit function or operation code (F), a 6-bit code extension and/or execution address (E), and a 13-bit (G) portion. The 26-bit instruction word occupies two sequential words of storage. The first word contains the F and E portion of the instruction. The second word (designated G) contains an address or operand, depending on the instruction (see following example).

The 13-bit instruction is divided into a 7-bit function or operation code (F) and a 6-bit code extension and/or execution address (E).

The control unit of the Compute Module reads an instruction word; it then examines the instruction word to see if the function code specifies a one- or two-word instruction. If a second word is called for, the control unit causes the G portion of the 26-bit instruction word to be read, and the instruction to be executed.

Example:

	Function Code (F)	Execution Address (E)
Word 1	7 Bits	6 Bits

	Operand (G)
Word 2	13 Bits

STORAGE CONTROLS

One Compute Module may have direct access to a maximum of 131,072 words. The Compute Module must be able to provide the storage system with a 17-bit address to uniquely specify one of the 131,072 words of available storage. The instructions provided in the Compute Module, depending on the type of instruction, have space in the instruction word format to provide 17, 13, 12, or 6 bits of address information. The storage bank control registers provide the additional bits required to make a unique 17-bit address.

The 171G data channels and the internal buffer channel of the 160G also have a basic capability of specifying 13 bits of address information. The storage bank control registers, associated with each I/O channel, provide the additional bits needed to arrive at a 17-bit address.

A reference to magnetic core storage, for a word of information, may be classified as one of four types. These references are used as follows:

1. To input or output information on a buffer I/O channel.
2. To read an instruction, or to locate a constant or counter which is associated with a program.
3. To read or store an operand or data.
4. To update an index register, an index register reading, or an address reference for indirect addressing.

The 160G automatically classifies each memory reference (in the Compute Module or in the I/O Module) and uses a storage bank control register, associated with each type of reference, to form the correct 17-bit address.

The setting of these extra bits, for the short address instructions, is implemented by setting storage bank control commands prior to the execution of an instruction.

If a bank control is set to reference a non-existent address in a given computer system, the computer stops when an instruction is executed which uses that particular storage bank control.

The following descriptions of storage bank controls is expanded in the section on address modes.

BUFFER STORAGE BANK CONTROL (b)

Each buffer channel has its own storage bank control which provides the upper four bits of the address for referencing data being read or written. The BER registers provide the remaining 13 bits of the address.

RELATIVE STORAGE BANK CONTROL (r)

The relative storage bank control determines the upper address bits of all references which read instructions into the instruction decoder of the Compute Module. It is also used to specify the upper bits of all relative memory reference instructions.

INDIRECT STORAGE BANK CONTROL (i)

The indirect storage bank control specifies the upper bits of all address references used in the memory, memory index, and indirect instructions to obtain the operand. This storage bank control normally specifies the bank in which data is located.

DIRECT STORAGE BANK CONTROL (d)

The direct storage bank control provides the upper bits needed to specify the location of index registers, addresses used in indirect addressing, and also for the direct instructions which are used to update index registers and counters.

STORAGE CONSTRUCTION

The magnetic core storage of the 160G is divided into modules of 8192 words. These modules may be called banks of memory. Each module has its own read and write controls and operates independently of any other Memory Module in the system. The design of the 160G Compute Module is such that it takes advantage of any memory overlap when referencing various words in memory.

The basic storage capacity of 160G system is 8192 words of storage provided in a Compute Module. The storage capacity may be expanded by adding Memory Modules to the system. A total of 15 Memory Modules may be added to the system for control by one Compute Module to provide a total storage capacity of 131,072 words.

In the G mode of operation, a fully expanded system would contain 16 banks or modules, each with 8192 words. The storage bank control registers are used to specify which bank is to be referenced. The banks are numbered as even numbers 0, 2, 4, 6, 10 to 36_8. Each bank has its own set of 20000_8 addresses, numbered from 00000 to 17777_8.

In the A mode of operation, the banks of storage function logically as two 4096-word banks. This distinction is necessary to maintain compatibility with the 160-A. A fully expanded system would contain 32 logical banks, each containing 4096 words. The banks in this case are numbered 0, 1, 2, 3 to 37_8.

ADDRESSING MODES

The 7-bit function code (F) portion of an instruction word, consists of an operation code and an address mode. The operation code is the basic instruction. The addressing mode permits the programmer to select the most efficient variation of the basic instruction with respect to speed and number of storage locations required.

In the GASS assembly system, the convention has been followed that the operation code and the address mode are symbolized by two, three or four mnemonic letters. The first two mnemonic code letters designate the basic instruction and the last one or two represent the addressing mode. When only two letters are used, the address mode is assumed to be entire memory, the basic 160G address mode.

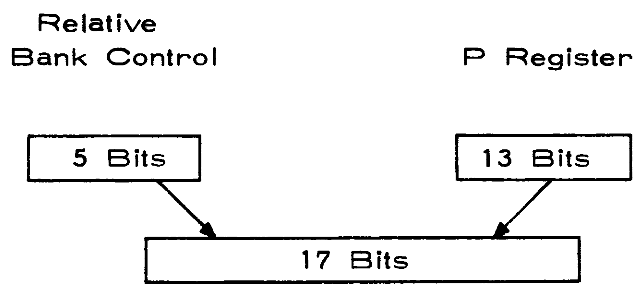
The 160G has 14 basic data handling instructions. Each of these basic instructions may have 12 variations, or addressing modes. The 14 basic instructions, their mnemonic codes, and descriptions are as follows:

Instruction	Mnemonic	Operation
Logical Product	LP	Form the logical product of two operands
Selective Complement	SC	Form the selective complement of two operands
Load	LD	Transfer an operand from memory to the A register

Instruction	Mnemonic	Operation
Load Complement	LC	Transfer complement of operand from memory to the A register
Add	AD	Form the sum of two operands
Subtract	SB	Form the difference of two operands
Store	ST	Place a word in memory
Shift Replace	SR	Shift word and replace in memory.
Replace Add	RA	Form the sum of two operands and replace in memory
Replace Add One	AO	Add one to operand and replace in memory
Load Q	LQ	Transfer an operand from memory to the Q register
Store Q	SQ	Place Q in memory.
Multiply	MU	Form the product of two operands
Divide	DV	Form the quotient of two operands

The program address counter, or the P register, contains the memory address of the current instruction. This register is a 13-bit register and cannot completely address 131,072 memory locations. The additional bits needed to form the 17-bit address are obtained from the relative storage bank control as shown in the following example. In all cases, it is assumed that the instruction has been read and that the F register contains the instruction. The steps for obtaining the address of the operand is then shown, including any additional memory references in the process.

Example:



Instruction Storage Address

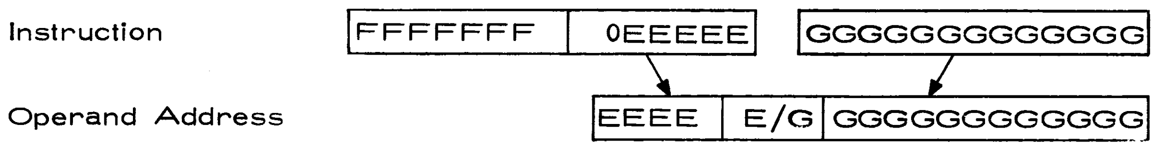
There are 13 bits in the P register and five bits in the relative bank control. This makes 18 bits which are combined to form the 17-bit storage address. To allow for the A (12-bit) and G (13-bit) mode of operation, which applies to all of the registers in the computer, a sharing of the highest-order bit of the P register and the lowest-order bit of the bank control is performed. In the A mode, the lowest-order bit from the bank control is used and the highest-order bit from the P register is ignored. In the G mode, the highest-order bit of the P register is used and the lowest-order bit from the bank control is ignored. This convention applies in all cases where a 13-bit P register is combined with a 5-bit bank control register.

NOTES:

1. The following programming examples shown assume the G mode of operation (13-bit addresses and operands).
2. The letters enclosed within the parentheses designate the storage bank controls. All subsequent letters, enclosed in parentheses, replace the two octal digits which indicate the actual storage bank to be referenced. This format is followed throughout the remainder of this manual.
3. All subsequent numbers are in octal notation unless stated otherwise.
4. The addressing modes are symbolized by one or two mnemonic code letters in the same manner as are the basic instructions. The exception to this is the entire memory address mode. This address mode does not have a mnemonic code.

ENTIRE MEMORY ADDRESS MODE

The entire memory address mode is the most general form of addressing in the 160G. This 26-bit instruction specifies a bank and a location within a bank to completely address the entire memory of the 160G. The G portion of the instruction specifies the location of the operand in the bank specified by the E portion of the first word. The address is formed as follows:



There are only five bits needed to completely specify the storage banks. The highest-order bit in the E portion of the instruction is 0 when an instruction designates the storage bank to be referenced.

A combination of bits is needed to specify the 17-bit storage address. In the G mode, the lowest-order bit of E is ignored and the highest-order bit of G is used. In the A mode, the lowest-order bit of E is used and the highest-order bit of G is ignored.

Example:	Location	F	E	G
	(r)01455	LD	10	
	(r)01456			02222
	(r)01457	next	instruction	
	(10)02222	001	44	

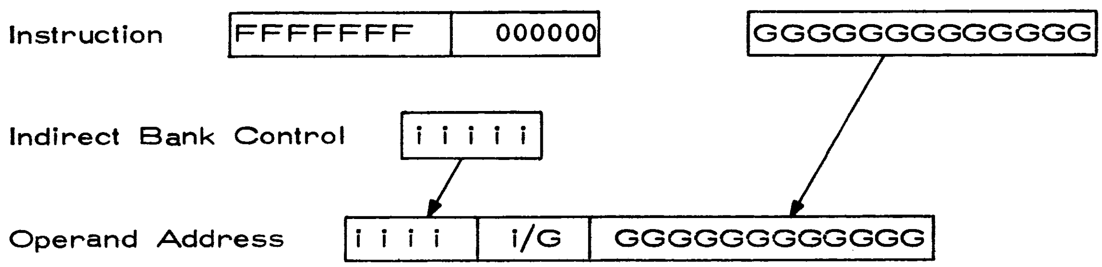
At location (r)01455 is a load entire memory (LD) instruction (no modifying mnemonic code). This instruction transmits the operand from memory to the A register. Location (10)02222 contains the operand (00144) which is transferred to the A register. At the completion of an entire memory instruction, control always continues at the location in the relative storage bank specified by the contents of P + 2. In this case control continues at (r)01457.

MEMORY ADDRESS MODE (M)

The memory address mode provides the address of the operand. This mode specifies the location within a bank of data to be referenced.

The bank number is "super indexed" by the contents of the indirect (memory) bank control that specifies which bank of memory contains the data. The G portion of the 26-bit instruction word contains the address of the operand. The E portion is always equal to zero.

The memory reference address is formed as follows:



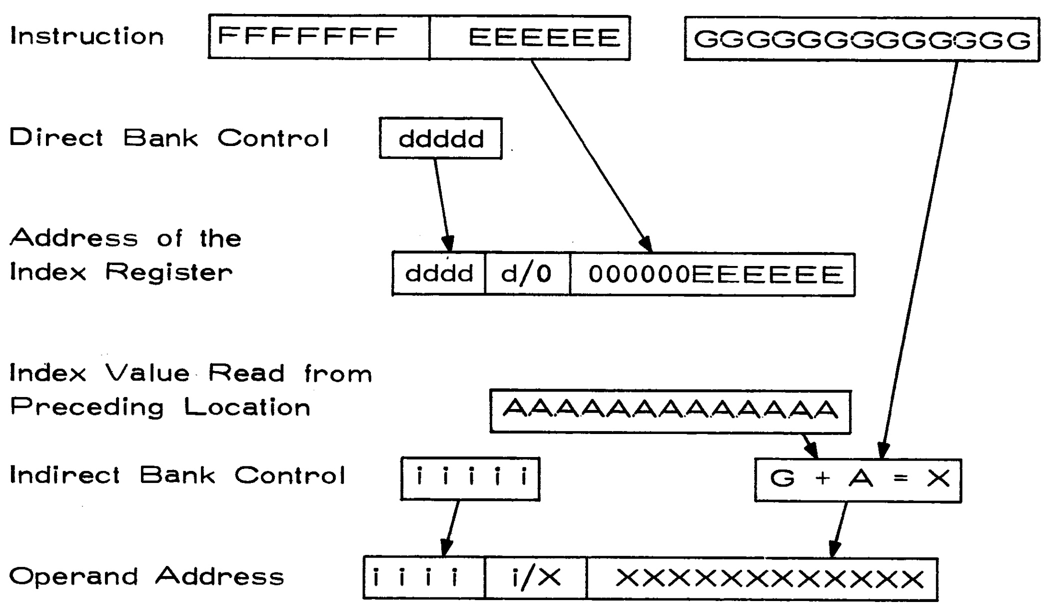
The remaining bits needed to form the 17-bit address are obtained from the indirect bank control. By changing the indirect bank control, a program can be changed to operate on information in another bank.

Example:	Location	F	E	G
	(r)03477	LDM	00	
	(r)03500			01111
	(r)03501	STM	00	
	(r)03502			00024
	(r)03503	next instruction		
	(i) 01111	067	66	
	(i) 00024	002	34	

At location (r)03477 is a load memory (LDM) instruction. The location (i)01111 becomes the operand address, and the quantity 06766 is transferred to the A register. At the completion of an (M) instruction, control continues in the relative storage bank at the location specified by the contents of P + 2. In this case, control continues at location (r)03501 which contains a store memory (STM) instruction. A store instruction causes the contents of the A register to be transferred to the operand address. The operand address of the instruction becomes (i)00024. The quantity 06766 which was in the A register will be stored in location (i)00024, replacing 00234.

MEMORY INDEX ADDRESS MODE (MX)

The memory index address mode applies indexing to the memory mode. This mode allows the programmer to make use of 62 index registers in the area of memory specified by the direct bank control. The formation of the operand address requires accessing of the index register to modify the G portion of the instruction prior to forming the operand address. The E portion of the 26-bit instruction specifies a location in the direct bank. The contents of this location is added to the G portion of the instruction. This sum becomes the effective operand address in the indirect storage bank. The G portion of the instruction, as stored in memory, is unchanged. The graphic formation of the operand address is as follows:

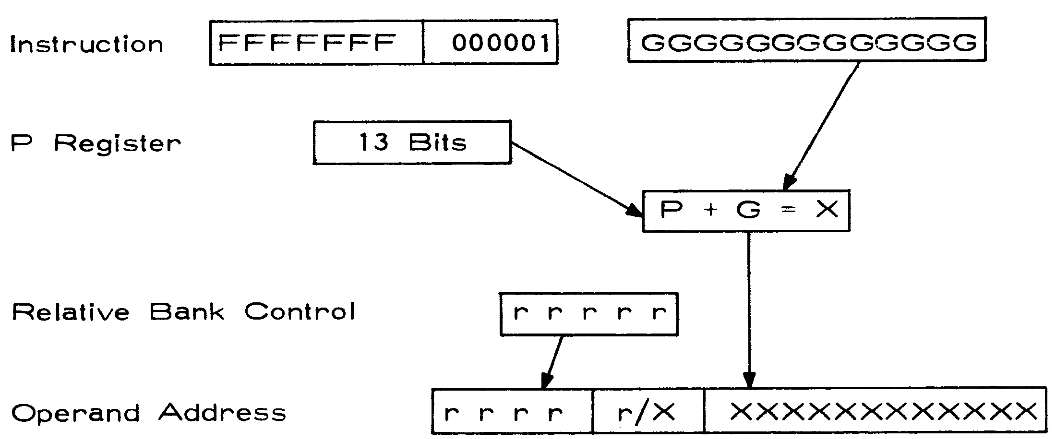


Example:	Location	F	E	G
	(r)00100	LDMX	05	
	(r)00101			10000
	(r)00102	next instruction		
	(d)00005	007	50	
	(i)10750	135	70	

At location (r)00100 is a load memory index (LDMX) instruction. The E portion of the instruction makes an initial reference to location (d)00005, which contains 00750. This number is added to the G portion of the instruction to yield the operand address, (i)10750. The contents of this location is transferred to the A register. At the completion of this instruction, the A register contains the quantity 13570. At the completion of an (MX) instruction, control continues in the relative storage bank at the location specified by the contents of P + 2. In this case, control continues at location (r)00102.

RELATIVE ENTIRE BANK ADDRESS MODE (RB)

The relative entire bank address mode provides a 26-bit form of the relative forward and relative backward address modes. This mode extends relative addressing to an entire bank of memory. The G portion of the instruction is algebraically added to the value in the P register to specify a location forward or backward from the current instruction. The backward ability comes from the modular arithmetic used in forming the address values. The operand address is formed as follows:



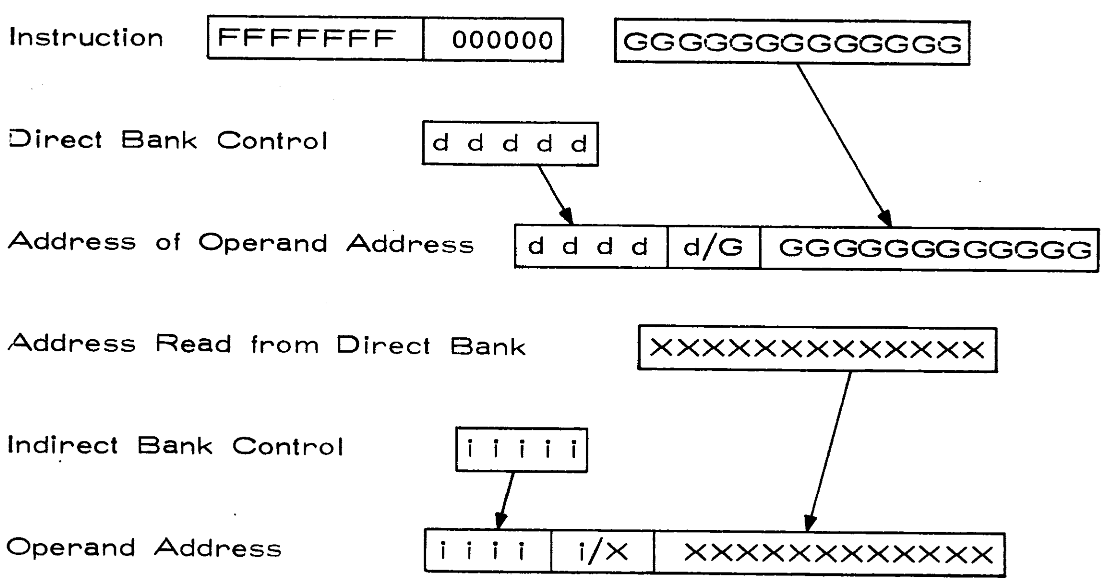
This addressing mode allows the programmer to make references to his program without the necessity of finding out where the program resides. This feature allows the program to be relocated easily. The operand address is obtained by adding the current contents of P to the G portion of the 26-bit instruction. This sum becomes the effective operand address in the relative storage bank.

Example:	Location	F	E	G
	(r)01000	LDRB	01	
	(r)01001			00150
	(r)01002	next instruction		
	(r)01150	123	45	

At location (r)01000 is a load relative entire bank (LDRB) instruction. G is added to the contents of the P register ((r)01000), to yield the address (r)01150. The contents of (r)01150 is transferred to the A register. At the completion of this instruction the A register contains the quantity 12345. At the completion of an (RB) instruction, which does not cause control to be transferred, control continues in the relative storage bank at the location specified by the contents of P + 2. In the preceding example, control continues at location (r)01002. An operand address that precedes the current contents of P can be referenced by inserting a negative number in the G portion of the instruction.

INDIRECT ENTIRE BANK ADDRESS MODE (IB)

The indirect entire bank address mode allows the programmer to use the entire 8192 locations in the direct bank of memory as the source of addresses for the operands. The operands are obtained from the indirect bank of memory. The operand address is formed as follows:



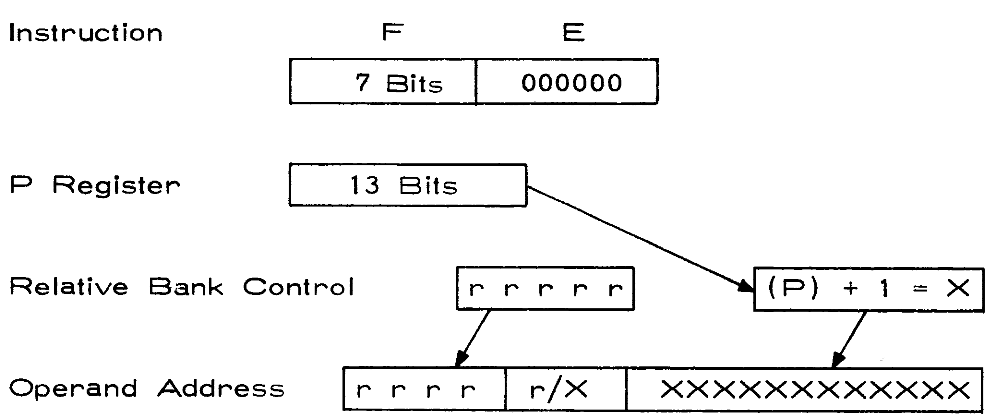
The indirect entire bank address mode allows the programmer to share the same data address with any number of instructions. All (IB) instructions occupy two sequential storage locations which involve entire bank operations. The G portion specifies a location in the direct bank. This location contains the address of the operand in the indirect bank.

Example:	Location	F	E	G
	(r)01011	LCIB	00	
	(r)01012			00166
	(r)01013	next instruction		
	(d)00166	071	22	
	(i) 07122	063	57	

At location (r)01011 is a load complement indirect entire bank (LCIB) instruction. Location (d)00166 contains the address 07122. At location (i)07122 is the operand 06357. The complement of this quantity, 11420, is transferred to the A register. Control continues in the relative storage bank at the location specified by the contents of P + 2. In the preceding example control continues at location (r)01013.

CONSTANT ADDRESS MODE (C)

The constant address mode may be thought of as a 13-bit instruction that obtains its operand from the next location in the relative bank. There is a provision to skip the constant in obtaining the next instruction. Otherwise, it may be taken as a 26-bit instruction that uses the G portion as the operand. Note that a memory storing instruction will change the G portion of the instruction.



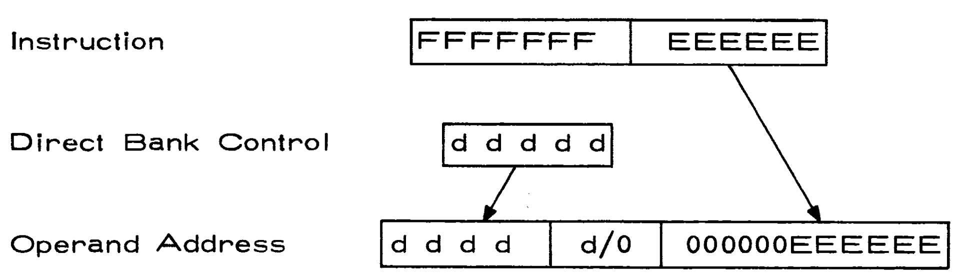
The G portion of the 26-bit instruction word contains the operand. The E portion of the instruction is always equal to zero.

Example:	Location	F	E	G
	(r)00101	LDC	00	
	(r)00102			07337
	(r)00103	STC	00	
	(r)00104			02345
	(r)00105	next instruction		

At location (r)00101 is a load constant (LDC) instruction. The operand address is (r)00102. The quantity 07337 is transferred to the A register. At the completion of a (C) instruction, control continues in the relative storage bank at the location specified by the contents of P + 2. In this case, control continues at (r)00103. This address contains a store constant (STC) instruction. In the preceding example, the operand address of the (STC) instruction is (r)00104. The quantity 07337, which was in the A register as a result of the LDC instruction in (r)00101, is transferred to location (4)00104 and therefore, replaces the constant 02345 now in (r)00104. The final contents of (r)00104 is 07337, and control continues at (r)00105.

DIRECT ADDRESS MODE (D)

The direct address mode is used to operate on index registers. The index registers are contained in the bank of memory specified by the direct bank control. The 17-bit address to reference these locations is formed as follows:



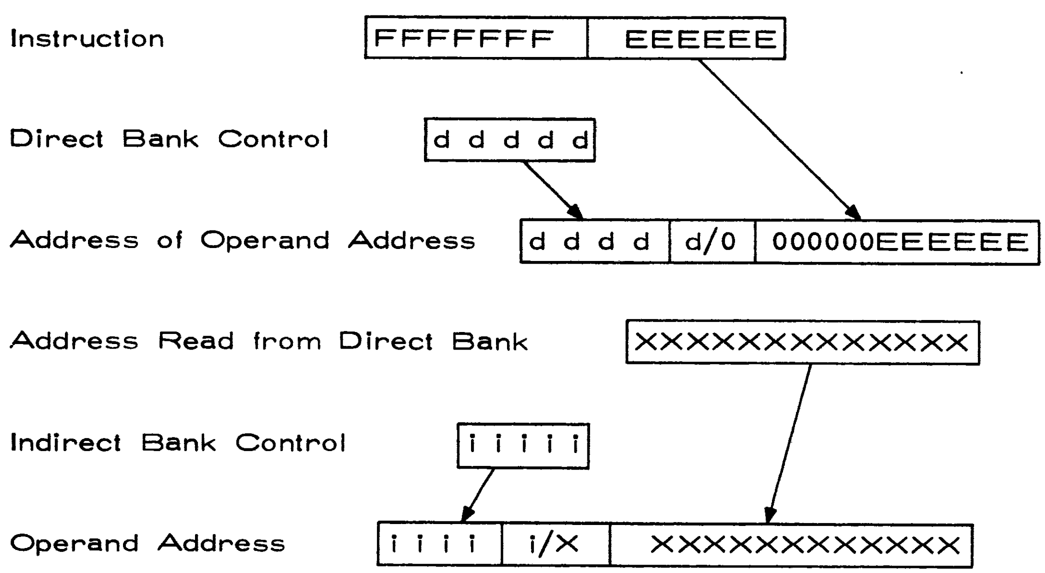
In the direct address mode, E selects one of the first 64 (100 octal) locations in the direct storage bank as the operand address.

Example:	Location	F	E
	(d)00076	012	34
	(r)01075	LDD	76
	(r)01076	next instruction	

At location (r)01075 is a load direct (LDD) instruction. E specifies that the operand address is (d)00076. This address contains the quantity 01234 which is transferred to the A register. At the completion of a direct address instruction, control always continues at the location in the relative storage bank specified by the contents of P + 1. In this case, control continues at location (r)01076.

INDIRECT ADDRESS MODE (I)

The indirect address mode enables a 13-bit instruction to share its address, which is located in the direct bank, with other instructions which may need reference to the same address. The E portion of the instruction specifies a location in the direct bank. This location contains the address of the operand in the indirect bank. The 17-bit operand address is formed as follows:

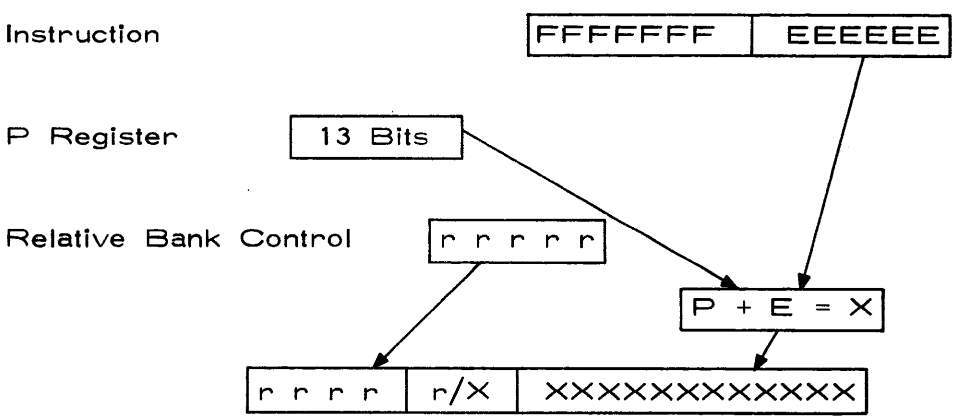


Example:	Location	F	E
	(d)00045	033	65
	(i) 03365	046	57
	(r)04121	LDI	45
	(r)04122	next instruction	

At location (r)04121 is a load indirect (LDI) instruction. E refers to location (d)00045, which contains the address 03365. A final reference is now made to location (i)03365, which contains the number 04657. The quantity 04657 is transferred to the A register. Both the direct (d) and indirect (i) storage bank controls are involved in the indirect (I) address mode. At the completion of an (I) instruction, control always continues at the location in the relative storage bank specified by the contents of P + 1. In the preceding example control continues at (r)04122.

RELATIVE FORWARD ADDRESS MODE (F)

The relative forward address mode is used to reference locations closely related to the current instruction in a program. With this mode it is possible to use a 13-bit instruction to reference a location 63 locations following the current instruction. The 17-bit operand address is formed as follows:



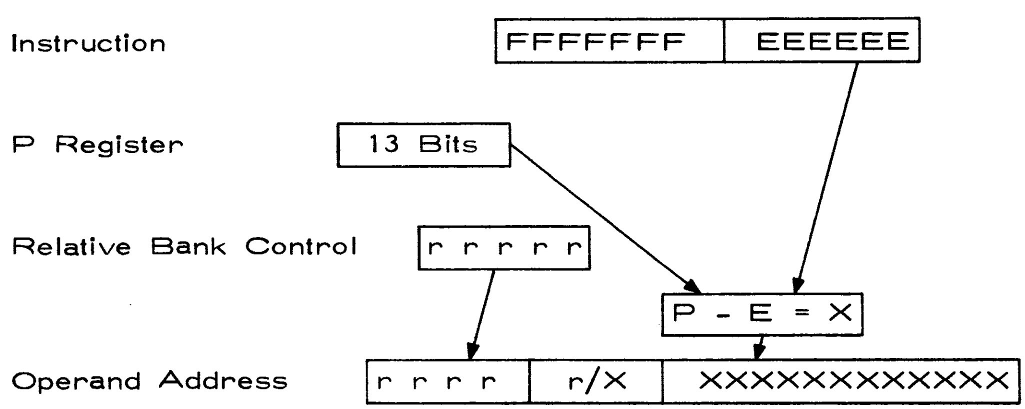
The operand or jump address is obtained by adding E to the current content of P to specify one of the 63 (77 octal) addresses immediately following the address of the current instruction. This sum then becomes the effective operand address in the relative storage bank.

Example:	Location	F	E
	(r)00233	LDF	22
	(r)00234	next instruction	
	(r)00255	077	03

At location (r)00233 is a load forward (LDF) instruction. E is added to the contents of the P register to yield the address (r)00255. The contents of (r)00255 is transferred to the A register. At the completion of this instruction, A contains the quantity 07703. At the completion of an (F) instruction, which does not cause control to be transferred, control continues in the relative storage bank at the location specified by the contents of P + 1. In the preceding example, control continues at location (r)00234. Certain (F) instructions cause control to be transferred E locations forward in the relative bank.

RELATIVE BACKWARD ADDRESS MODE (B)

The relative backward address mode is used to reference locations closely related to the current instruction in a program. With this mode it is possible to use a 13-bit instruction to reference a location 77 octal locations preceding the current instruction. The 17-bit operand address is formed as follows:



The operand address is obtained by subtracting E from the current contents of P to specify one of the 63 (77 octal) locations immediately preceding the location of the current instruction. The difference then becomes the effective operand address in the relative storage bank.

SPECIFIC ADDRESS MODE (S)

The specific address mode is used in 13-bit instructions which work on the specific register. The specific register is core storage location (0)07777 in the A mode of operation and (0)17777 in the G mode of operation. Reference is made to the specific register regardless of the setting of any bank controls. This address mode is carried over to provide compatibility with the 160-A.

The operand address is always octal location 7777 (A mode) or 17777 (G mode) in storage bank zero. The E portion of the instruction word is always equal to zero in the specific address mode.

Example:	Location	F	E
	(r)00177	LDS	00
	(r)00200	next instruction	
	(0)17777	043	21

Location (r)00177 contains a load specific (LDS) instruction. The fact that E equals zero is used in address decoding to specify that the address of the operand of this instruction is (0)17777. Thus, the quantity 04321 is transferred to the A register. At the completion of an (S) instruction, control continues in the relative storage bank at the location specified by the contents of P + 1. In the preceding example, control continues at location (r)00200.

NO ADDRESS MODE (N)

The no address mode specifies an operand, the six bits contained in the E portion of the instruction, that has the higher-order seven bits set to zero. This mode allows arithmetic and logical operations by using a 6-bit operand that is contained in the no address mode instruction. This mode eliminates the need for entering many small constants into memory.

Example:	Location	F	E
	(r)00100	LDN	43
	(r)00101	next instruction	

At location (r)00100 is a load no address (LDN) instruction. The 13-bit operand for LDN is 00043. Therefore, the number 00043 is transferred to the A register. At the completion of a no address instruction, control always continues at the location in the relative storage bank specified by the contents of P + 1. In this case control continues at location (r)00101.

NOTATION

The following abbreviations are used in the description of the instructions and the remainder of this manual.

A	The A register
Q	The Q register
P	The P register
Z	The Z register
BER	The buffer entrance register
BXR	The buffer exit register
BFR	The buffer data register
F	The seven higher-order bits of the first word of an instruction (function code)
E	The six lower-order bits of the first word of an instruction (execution address)
G	The 13 bits of the second word of all two-word instructions (operand or operand address)
X	Any octal digit 0 through 7, generally part of an address
Y	Any octal digit 0 through 7, generally part of an operand
Z	Any octal digit, 0 through 7, generally part of a bank
( )	The contents of whatever location or register is specified within the parentheses. The only exception is a reference to a specific storage bank control in its numeric value. In this case, reference is to the storage bank number. Thus, (0) refers to storage bank zero. This is the only time a single digit is enclosed in parentheses.

(d)	The contents of the direct storage bank control
(r)	The contents of the relative storage bank control
(i)	The contents of the indirect storage bank control
(b)	The contents of the buffer storage bank control
→	The function or quantity on the left of the arrow replaces the function or quantity on the right
FWA	First word address - This term is used when reference is made to any block of data. The core storage address of the first word of such a block is known as its FWA.
LWA	Last word address - This term is used when reference is made to any block of data. The core storage address of the last word of such a block is known as its LWA.

DESCRIPTION OF INSTRUCTIONS

The following notes apply to the discussion of the 160G instruction repertoire.

1. All instruction times are given in storage cycles where one cycle equals 1.35 microseconds. Two instruction times are given for instructions which take advantage of overlap in memory references. The first time given is a maximum time, assuming all memory references are made to the same bank of memory. The second time given is a minimum time, assuming instructions are in one bank of memory, data in another bank, and index registers are in a third bank of memory.

2. All numbers are in octal notation unless otherwise stated.

3. In the description of an instruction, the operations involved are listed in sequence if more than one operation is performed.

4. Instructions which may assume more than one address mode are described under the general command function. The operand formation for each address mode is described in the section on address modes.

5. The G portion of an instruction is shown only with those instructions which occupy two words of storage.

6. Instructions concerned with the buffer data channels have a mnemonic code ending with Y. The value of Y is always numeric and within the limits specified by the instruction.

7. The E portion of several instructions serves as an address or address extension and/or as a function code extension.

   Therefore, the E portion of instructions specifying indirect (I), forward (F), backward (B), relative entire bank (RB) and memory index (MX) addressing modes should not be zero. If it is 0, a different address mode is specified. Similarly, the E portion of memory index address mode (MX) should not be 0 or 1, because the address mode would be interpreted as an indirect entire bank (IB) or a relative entire bank address mode (RB).

ARITHMETIC/LOGICAL INSTRUCTIONS

LOGICAL PRODUCT

FFFEE	G	Mnemonic	Name	Timing
002XX		LPN	Logical Product No Address	1
010XX		LPD	Logical Product Direct	2/1.5
01100	XXXXX	LPM	Logical Product Memory	3/2
001XX		LPI	Logical Product Indirect	3/2
01200	YYYYY	LPC	Logical Product Constant	2
012XX		LPF	Logical Product Forward	2
01300		LPS	Logical Product Specific	2/1.5
013XX		LPB	Logical Product Backward	2
110ZZ	XXXXX	LP	Logical Product Entire Memory	3/2
11100	XXXXX	LPIB	Logical Product Indirect Entire Bank	4/2.5

FFFEE	G	Mnemonic	Name	Timing
11101	XXXXX	LPRB	Logical Product Relative Entire Bank	3
111YY	XXXXX	LPMX	Logical Product Memory Index	4/2.5

Form in A the logical product of the operand and the original contents of A. The operand is not altered by a logical product instruction. The proper operand is formed for each address mode, and control continues as described in the section on addressing modes.

The logical product of two operands is defined as follows:

Operand 1 (bit value)	0	0	1	1	
Operand 2 (bit value)	0	1	0	1	
Logical product of 1 and 2	0	0	0	1	(bit value)

From the preceding definition, it can be seen that selected portions of A may be cleared or retained in A by using the proper operand as a mask.

SELECTIVE COMPLEMENT

FFFEE	G	Mnemonic	Name	Timing
003XX		SCN	Selective Complement No Address	1
014XX		SCD	Selective Complement Direct	2/1.5
01500	XXXXX	SCM	Selective Complement Memory	3/2
015XX		SCI	Selective Complement Indirect	3/2

FFFEE	G	Mnemonic	Name	Timing
01600	YYYYY	SCC	Selective Complement Constant	2
016XX		SCF	Selective Complement Forward	2
01700		SCS	Selective Complement Specific	2/1.5
017XX		SCB	Selective Complement Backward	2
114ZZ	XXXXX	SC	Selective Complement Entire Memory	3/2
11500	XXXXX	SCIB	Selective Complement Indirect Entire Bank	4/2.5
11501	XXXXX	SCRB	Selective Complement Relative Entire Bank	3
115YY	XXXXX	SCMX	Selective Complement Memory Index	4/2.5

Form in A the bit-by-bit complement of A for each bit in the operand equal to 1. The operand is not altered by a selective complement instruction. The proper operand for each address mode is formed, and control continues as described in the section on address modes. The selective complement operation is defined as follows:

(A) register (bit value)	0	0	1	1	
Operand (bit value)	0	1	0	1	
Final content of the A register	0	1	1	0	(bit value)

LOAD

FFFEE	G	Mnemonic	Name	Timing
004XX		LDN	Load No Address	1
020XX		LDD	Load Direct	2/1.5
02100	XXXXX	LDM	Load Memory	3/2
021XX		LDI	Load Indirect	3/2
02200	YYYYY	LDC	Load Constant	2
022XX		LDF	Load Forward	2
02300		LDS	Load Specific	2/1.5
023XX		LDB	Load Backward	2
120ZZ	XXXXX	LD	Load Entire Memory	3/2
12100	XXXXX	LDIB	Load Indirect Entire Bank	4/2.5
12101	XXXXX	LDRB	Load Relative Entire Bank	3
121YY	XXXXX	LDMX	Load Memory Index	4/2.5

Operand ⟶ A

Transfer the operand to A. The operand in storage is not altered by the execution of a load instruction. The proper operand is formed for each address mode and control continues as described in the section on address modes.

LOAD COMPLEMENT

FFFEE	G	Mnemonic	Name	Timing
005XX		LCN	Load Complement No Address	1
024XX		LDC	Load Complement Direct	2/1.5
02500	XXXXX	LCM	Load Complement Memory	3/2
025XX		LCI	Load Complement Indirect	3/2
02600	YYYYY	LCC	Load Complement Constant	2
026XX		LCF	Load Complement Forward	2
02700		LCS	Load Complement Specific	2/1.5
027XX		LCB	Load Complement Backward	2
124ZZ	XXXXX	LC	Load Complement Entire Memory	3/2
12500	XXXXX	LCIB	Load Complement Indirect Entire Bank	4/2.5
12501	XXXXX	LCRB	Load Complement Relative Entire Bank	3
125YY	XXXXX	LCMX	Load Complement Memory Index	4/2.5

Operand' ⟶ A

Transfer the one's complement of the operand to A. The operand in storage is not altered by the execution of a load complement instruction. The proper operand is formed for each addressing mode and control continues as described in the section on addressing modes.

ADD

FFFEE	G	Mnemonic	Name	Timing
006XX		ADN	Add No Address	1
030XX		ADD	Add Direct	2/1.5
03100	XXXXX	ADM	Add Memory	3/2
031XX		ADI	Add Indirect	3/2
03200	YYYYY	ADC	Add Constant	2
032XX		ADF	Add Forward	2
03300		ADS	Add Specific	2/1.5
033XX		ADB	Add Backward	2
130ZZ	XXXXX	AD	Add Entire Memory	3/2
13100	XXXXX	ADIB	Add Indirect Entire Bank	4/2.5
13101	XXXXX	ADRB	Add Relative Entire Bank	3
131YY	XXXXX	ADMX	Add Memory Index	4/2.5

$$(A) + \text{operand} \longrightarrow A$$

Place in A the sum of the original contents of A and the operand. The operand is unchanged by an add instruction. The correct operand is formed for each addressing mode, and control continues as described in the section on addressing modes.

SUBTRACT

FFFEE	G	Mnemonic	Name	Timing
007XX		SBN	Subtract No Address	1
034XX		SBD	Subtract Direct	2/1.5
03500	XXXXX	SBM	Subtract Memory	3/2
035XX		SBI	Subtract Indirect	3/2
03600	YYYYY	SBC	Subtract Constant	2
036XX		SBF	Subtract Forward	2
03700		SBS	Subtract Specific	2/1.5
037XX		SBB	Subtract Backward	2
134ZZ	XXXXX	SB	Subtract Entire Memory	3/2
13500	XXXXX	SBIB	Subtract Indirect Entire Bank	4/2.5
13501	XXXXX	SBRB	Subtract Relative Entire Bank	3
135YY	XXXXX	SBMX	Subtract Memory Index	4/2.5

$$(A) - \text{operand} \longrightarrow A$$

Form in A the difference between the original contents of A and the operand. The operand is unchanged by a subtract instruction. The proper operand is formed for each addressing mode, and control continues as described in the section on addressing modes.

STORE

FFFEE	G	Mnemonic	Name	Timing
040XX		STD	Store Direct	2/1.5
04100	XXXXX	STM	Store Memory	3/2
041XX		STI	Store Indirect	3/2
04200	YYYYY	STC	Store Constant	2
042XX		STF	Store Forward	2
04300		STS	Store Specific	2/1.5
043XX		STB	Store Backward	2
140ZZ	XXXXX	ST	Store Entire Memory	3/2
14100	XXXXX	STIB	Store Indirect Entire Bank	4/2.5
14101	XXXXX	STRB	Store Relative Entire Bank	3
141YY	XXXXX	STMX	Store Memory Index	4/2.5

(A) → Operand address

Transfer the contents of A to the operand address. The contents of A is not altered by a store instruction. The proper operand address is formed for each addressing mode, and control continues as described in the section on addressing modes.

SHIFT REPLACE

FFFEE	G	Mnemonic	Name	Timing
044XX		SRD	Shift Replace Direct	3/2.5
04500	XXXXX	SRM	Shift Replace Memory	4/3
045XX		SRI	Shift Replace Indirect	4/3
04600	YYYYY	SRC	Shift Replace Constant	3
046XX		SRF	Shift Replace Forward	3
04700		SRS	Shift Replace Specific	3/2.5
047XX		SRB	Shift Replace Backward	3
144ZZ	XXXXX	SR	Shift Replace Entire Memory	4/3
14500	XXXXX	SRIB	Shift Replace Indirect Entire Bank	5/3.5
14501	XXXXX	SRRB	Shift Replace Relative Entire Bank	3
145YY	XXXXX	SRMX	Shift Replace Memory Index	5/3.5

Operand $\longrightarrow$ A

Shift A left one-bit position

(A) $\longrightarrow$ Operand address

The operand is placed in the A register, then the operand is shifted left one bit position in A, and the contents of A is transferred back to the operand address. The operand for each addressing mode is formed, and control continues as described in the section on addressing modes. At the completion of a shift replace instruction, both A and the operand address contain the shift resultant.

REPLACE ADD

FFFEE	G	Mnemonic	Name	Timing
050XX		RAD	Replace Add Direct	3/2.5
05100	XXXXX	RAM	Replace Add Memory	4/3
051XX		RAI	Replace Add Indirect	4/3
05200	YYYYY	RAC	Replace Add Constant	3
052XX		RAF	Replace Add Forward	3
05300		RAS	Replace Add Specific	3/2.5
053XX		RAB	Replace Add Backward	3
150ZZ	XXXXX	RA	Replace Add Entire Memory	4/3
15100	XXXXX	RAIB	Replace Add Indirect Entire Bank	5/3.5
15101	XXXXX	RARB	Replace Add Relative Entire Bank	4

FFFEE	G	Mnemonic	Name	Timing
151YY	XXXXX	RAMX	Replace Add Memory Index	5/3.5

(A) + operand ⟶ A

(A) ⟶ Operand address

Form in A the sum of the original contents of A and the operand; transfer this sum to the operand address. At the completion of the replace add instruction, both the operand address and A contain the new sum. The proper operand is formed for each addressing mode, and control continues as described in the section on addressing modes.

REPLACE ADD ONE

FFFEE	G	Mnemonic	Name	Timing
054XX		AOD	Replace Add One Direct	3/2.5
05500	XXXXX	AOM	Replace Add One Memory	4/3
055XX		AOI	Replace Add One Indirect	4/3
05600	YYYYY	AOC	Replace Add One Constant	3
056XX		AOF	Replace Add One Forward	3
05700		AOS	Replace Add One Specific	3/2.5
057XX		AOB	Replace Add One Backward	3

FFFEE	G	Mnemonic	Name	Timing
154ZZ	XXXXX	AO	Replace Add One Entire Memory	4/3
15500	XXXXX	AOIB	Replace Add One Indirect Entire Bank	5/3.5
15501	XXXXX	AORB	Replace Add One Relative Entire Bank	4
155YY	XXXXX	AOMX	Replace Add One Memory Index	5/3.5

Operand ⟶ A

(A) + 1 ⟶ A

(A) ⟶ Operand address

Form in A the sum of the operand and one; transfer this sum to the operand address. At the completion of the replace add one instruction, both the A register and the operand address contain the original operand plus one. The proper operand is formed for each addressing mode, and control continues as described in the section on addressing modes.

LOAD Q

FFFEE	G	Mnemonic	Name	Timing
160ZZ	XXXXX	LQ	Load Q Entire Memory	3/2
16100	XXXXX	LQIB	Load Q Indirect Entire Bank	4/2.5

FFFEE	G	Mnemonic	Name	Timing
16101	XXXXX	LQRB	Load Q Relative Entire Bank	3
161YY	XXXXX	LQMX	Load Q Memory Index	4/2.5
16200	YYYYY	LQC	Load Q Constant	2
162XX		LQF	Load Q Forward	2

Operand ⟶ Q

Transfer the operand to Q. The operand in storage is not altered by the execution of a load instruction. The proper operand is formed for each addressing mode.

STORE Q

FFFEE	G	Mnemonic	Name	Timing
162ZZ	XXXXX	SQ	Store Q Entire Memory	3/2
16500	XXXXX	SQIB	Store Q Indirect Entire Bank	4/2.5
16501	XXXXX	SQRB	Store Q Relative Entire Bank	3
165YY	XXXXX	SQMX	Store Q Memory Index	4/2.5
16600	YYYYY	SQC	Store Q Constant	2
166XX		SQF	Store Q Forward	2

(Q) ⟶ Operand Address

Transfer the contents of Q to the operand address. The contents of Q is not altered by a store Q instruction. The proper operand address is formed for each addressing mode, and control continues as described in the section on addressing modes.

MULTIPLY

FFFEE	G	Mnemonic	Name	Timing
170ZZ	XXXXX	MU	Multiply Entire Memory	5/4
17100	XXXXX	MUIB	Multiply Indirect Entire Bank	6/4.5
17101	XXXXX	MURB	Multiply Relative Entire Bank	5
171YY	XXXXX	MUMX	Multiply Memory Index	6/4.5
17200	YYYYY	MUC	Multiply Constant	4
172XX		MUF	Multiply Forward	4

The contents of A is multipled by the proper multiplier for the address mode specified. The multiplier is not altered by the multiply instruction. The product appears in the 26-bit AQ register, where A contains the most significant 13 bits and Q contains the least significant 13 bits. Control continues at location (r) (P) + 2.

NOTES:

1. The multiply operation takes place as an algebraic multiplication of integers. If the product is less than 2^{12}, Q contains the integer answer. If the product is more than 2^{12}, the high-order bit of Q is a significant bit of the answer and the entire product must be taken as the 26-bit AQ register.

2. If a fractional multiply is required, a long left shift of 1 (LLS, 1) must be performed after the multiply instruction to adjust the binary point of the product. The most significant portion of the fraction resides in the A register.

3. The multiply operation is always 13 bits regardless of the operating mode.

DIVIDE

FFFEE	G	Mnemonic	Name	Timing
174ZZ	XXXXX	DV	Divide Entire Memory	6/5
17500	XXXXX	DVIB	Divide Indirect Entire Bank	7/5.5
17501	XXXXX	DVRB	Divide Relative Entire Bank	6
175YY	XXXXX	DVMX	Divide Memory Index	7/5.5
17600	YYYYY	DVC	Divide Constant	5
176XX		DVF	Divide Forward	5

The contents of the 26-bit AQ register is divided by the proper divisor for the address mode specified. The divisor is not altered by the divide instruction. The algebraic quotient appears in the A register. The remainder appears in the Q register and has the same sign as the original contents of AQ. Control continues at location (r) (P) + 2.

NOTES:

1. The operation may be taken as integer or fractional division depending on the replacement of the dividend in the AQ register. To prevent overflow, the number in the A register must be less than one-half the absolute value of the divisor.

2. To do an integer divide, the dividend must be placed in the Q register with the A register set so all of its bits correspond to the sign of Q. The simplest method is to place the dividend in the A register and then do an LRS 13 (long right shift to place the operand in Q and extend the sign in A).

3. To do a fractional divide, the dividend must be placed in the A register with the sign of the dividend in all bits of the Q register. The dividend must be less in absolute value than the divisor. The simplest method is to place the dividend in the A register and then do an LRS 13, followed by an XAQ instruction.

4. Bit 0 of the error register is set to 1 if there is divide overflow (A portion of the dividend is larger than one-half of the divisor in absolute value), or if there is a divide by zero attempted. In either case, the machine performs the divide operation and the original dividend is destroyed while a meaningless answer is developed.

5. The divide operation is always 13 bits, regardless of the mode of operation.

SHIFT INSTRUCTIONS

FIXED SHIFTS

FFFEE	G	Mnemonic	Name	Timing
00102		LS1	Left Shift One	1
00103		LS2	Left Shift Two	1
00110		LS3	Left Shift Three	1
00111		LS6	Left Shift Six	1
00114		RS1	Right Shift One	1
00115		RS2	Right Shift Two	1

Shift A right or left the number of bit positions specified. Control continues at (r) (P) + 1. All left shifts are circular; a bit shifted out of the highest-order bit position is shifted into bit position 00. All right shifts are end-off shifts: for each bit position shifted, the sign is extended, and the bit in position 00 is discarded.

The next instruction is obtained from location (r) (P) + 1.

NOTES:

1. LS3 and LS6 instructions always operate on the lower 12 bits of the A register. The highest-order bit of A is unchanged. In this case, the shift is always a circular, 12-bit, end-around shift.

2. The remaining shift instructions operate on either 12 or 13 bits depending on the mode of operation.

MULTIPLY BY 10

FFFEE	G	Mnemonic	Name	Timing
00112		MUT	Multiply A by 10	1

$$10_{10}(A) \longrightarrow A$$

$$(P) + 1 \longrightarrow P$$

Multiply the contents of A by 10_{10}. The resultant is reduced modulo $2^{12}-1$ and placed in A at the completion of the instruction. For the range of numbers -314_8 to $+314_8$, the arithmetic resultant is algebraically correct since no reduction modulo $2^{12}-1$ takes place. Control continues at location (r) (P) + 1.

MULTIPLY BY 100

FFFEE	G	Mnemonic	Name	Timing
00113		MUH	Multiply A by 100_{10}	1

$$100_{10}\ (A) \longrightarrow A$$

$$(P) + 1 \longrightarrow P$$

Multiply the contents of A by 100_{10}. The resultant is reduced to modulo $2^{12}-1$ and placed in A at the completion of the instruction. For the range of numbers -24_8 to $+24_8$, the resultant is algebraically correct. Control continues at location (r) (P) + 1.

> NOTE: When MUT and MUH instructions are performed in the G mode, bit 12 is always set to 0 as a result of the operation.

VARIABLE SHIFTS

FFFEE	G	Mnemonic	Name	Timing
113YY		ARS	A Right Shift	See Note 1
117YY		ALS	A Left Shift	
123YY		QRS	Q Right Shift	
127YY		QLS	Q Left Shift	
133YY		LRS	AQ Right Shift	
137YY		LLS	AQ Left Shift	

Shift A, Q, or AQ left or right the number of bit positions specified by YY in the E portion of the instructions. Control continues at (r) (P) + 1. All left shifts are circular; a bit shifted out of the highest-order bit position is shifted into bit position 00. All right shifts are end-off shifts; for each bit position shifted, the sign bit is extended, and the bit in position 00 is discarded.

NOTES:

1. Timing is dependent on the number of shifts involved. If $n \leq 7$, shift time is equal to one memory cycle, where n = number of shifts. If $n > 7$, shift time $= 1 + \frac{n-7}{11}$.
2. The shifts always operate on 13-bit registers regardless of the mode of operation.

STORAGE BANK CONTROL INSTRUCTIONS

This group of instructions is used to set bank assignments in the storage bank control registers. After execution of these instructions, the memory references for data, program, and index registers are in the assigned memory bank. Note that when the relative bank control register is changed, a jump, or transfer of program control is performed.

FFFEE	G	Mnemonic	Name	Timing
0001X		SRJ	Set Relative Bank Control and Jump	1
0002X		SIC	Set Indirect Bank Control	1
0003X		IRJ	Set Indirect and Relative Bank Controls and Jump	1

FFFEE	G	Mnemonic	Name	Timing
0004X		SDC	Set Direct Bank Control	1
0005X		DRJ	Set Direct and Relative Bank Controls and Jump	1
0006X		SID	Set Indirect and Direct Bank Controls	1
0007X		ACJ	Set Direct, Indirect, and Relative Bank Controls and Jump	1
0014X		SBU	Set Buffer Bank Control	1

Set the specified storage bank control or controls to reference bank X. X is any number, 0 through 7. For instructions SIC, SDC, SID, and SBU, control continues at (r) (P) + 1. The remaining instructions of this group (SRJ, IRJ, DRJ, and ACJ) can be used to transfer program control between storage banks. It is the act of setting the relative bank control (r) which alters the bank from which the next program instruction is taken. Whenever an instruction is given which sets (r), the next program instruction is taken from the new (r) at the address specified by the contents of the A register. Not only may (r) be set by itself, but combinations of memory bank controls may be set at the same time. The contents of A is not changed when storage bank control instructions are executed.

FFFEE	G	Mnemonic	Name	Timing
10103	XXXXX	RCJP	AQ to Bank Controls and Jump	2

G02000b

$$A_{10-6} \rightarrow (b)$$

$$A_{4-0} \rightarrow (d)$$

$$Q_{10-6} \rightarrow (i)$$

$$Q_{4-0} \rightarrow (r)$$

$$XXXXX \rightarrow P$$

Transfer the contents of AQ to the storage bank controls in the bit positions shown. Control continues at (r)XXXXX.

NOTES:

1. The order of bits in the AQ register is the same as obtained from the CTA instruction.

2. The jump, after completing this instruction, is to the location XXXXX in the relative bank as set by the RCJP instruction.

FFFEE	G	Mnemonic	Name	Timing
1001Y	000ZZ	BBCY	Set Buffer Bank Control, Channel Y	2
143ZZ	XXXXX	SRJP	Set Relative Bank Control-Entire Memory and Jump	2
147ZZ	XXXXX	DRJP	Set Direct and Relative Bank Control Entire Memory and Jump	2
153ZZ		SDCG	Set Direct Bank Control Entire Memory	1
157ZZ		SICG	Set Indirect Bank Control Entire Memory	1

Set the specified storage bank control or controls to reference bank ZZ. For instructions BBCY, SDCG, and SICG, control continues at the next instruction. For instructions SRJP and DRJP, control is changed to bank ZZ, and program control continues at address XXXXX in that bank. For a BBCY instruction, the bank control of Channel Y is set to bank ZZ. Y is any number, 1 through 7.

NOTES:

1. Setting buffer bank controls, by a BBCY or a SBU instruction, may take place while the buffer is in operation with a resulting change of the source or destination of buffered information. Normally, the programmer should insure that the buffer is not busy before giving this instruction.

2. The following examples illustrate GASS coding of these instructions:

BBC5, 4	Set buffer data channel 5 to input or output from storage bank 4.
SRJP,6 FOO	Transfer control to location FOO in storage bank 6.
SRJP LOOP	Transfer control to location LOOP. The bank setting value is provided by GASS.
SIC,2	Set indirect bank control to reference storage bank 2.

INPUT/OUTPUT INSTRUCTIONS

CLEAR BUFFER CONTROLS

FFFEE	G	Mnemonic	Name	Timing
00104		CBC	Clear Buffer Controls	1

FFFEE	G	Mnemonic	Name	Timing
1017Y		CBCY	Clear Buffer Controls, Channel Y	1

Stop all buffer operations in progress and clear the buffer controls. This instruction does not clear the BER, BXR, or BFR, but it stops any buffer operation in progress. If an I/O operation is stopped when some piece of unit record equipment (such as a magnetic tape unit or punched card reader) is in operation, the buffer disconnects from the equipment; but the peripheral equipment continues to the end of the unit record. The remaining data is not processed by the computer. A buffer complete signal (interrupt 20) (internal interrupts 100, 120, 140, etc.) is not generated when the CBC or CBCY instruction is executed.

NOTES:

1. Y can be any number, 2 through 7, for the CBCY instruction.

2. The instruction immediately following CBCY should not depend on the buffer busy signal from the 171G.

3. The next four instructions following CBCY should be programmed to expect the busy signal when they are executed for the first time.

CLEAR INTERRUPT LOCKOUT

FFFEE	G	Mnemonic	Name	Timing
00120		CIL	Clear Interrupt Lockout	1
1012Y		CILY	Clear Interrupt Lockout, Channel Y	1

Clear the interrupt lockout and allow any waiting interrupts to function. When a CIL or CILY instruction is executed, interrupt lockout is not cleared until the instruction following the CIL or CILY has been executed.

SET INTERRUPT LOCKOUT

FFFEE	G	Mnemonic	Name	Timing
1014Y		SILY	Set Interrupt Lock-out, Channel Y	1

Set interrupt lockout on the buffer data channel specified by Y in the E portion of the instruction. When a SILY instruction is executed, interrupt lockout is not set until the instruction following the SILY has been executed. Interrupt lockout may also be set by one of the following conditions:

1. Execution of an interrupt
2. Execution of an EXF instruction
3. Execution of an EXC instruction
4. Execution of an EXCY instruction

At the completion of a SILY instruction, control continues at (r) (P)+1.

NOTES:

1. Interrupt lockouts exist on two levels. A master lockout which is called lockout 0 and individual buffer channel lockouts which are called lockouts 1 through 7. Lockout 0 prevents the processing of any further interrupts by the computer. Setting one of the other lockouts (1 through 7) prevents any further interrupt processing by the computer from that data channel, but an interrupt may occur from a different data channel.

2. Setting lockout 0 by SILY, or clearing lockout 0 by CIL or CILY does not affect the status of the lower-level lockouts.

3. When in the A mode, CIL also releases lockout 1, which applies to the internal buffer data channel.

4. The value of Y may be any number from 0 through 7. CIL0 or SIL0 clears and sets the master interrupt lockout. The rest of the instructions clears and sets the lower-level buffer data channel interrupt lockouts.

INITIATE BUFFER INPUT

FFFEE	G	Mnemonic	Name	Timing
07200	XXXXX	IBI	Initiate Buffer Input	1 (no jump) 2 (jump)
1006Y	XXXXX	IBIY	Initiate Buffer Input Channel Y	1 (no jump) 2 (jump)

If buffer is busy XXXXX ⟶ P

If buffer is not busy, (P) + 2 ⟶ P

Start input buffering operation. When complete, generate a buffer complete interrupt for the designated buffer.

The IBI instruction initiates an input buffer on the buffered I/O channel. Prior to an IBI instruction, however, the external device must be selected and BER, BXR, and the buffer storage bank control (b) must be set. If the buffer is in operation, no buffering action is taken, and control continues at (r) XXXXX. If the buffer is not in operation, the buffering operation is started, and control continues immediately at location (r) (P) + 2. When the buffer operation terminates, a buffer complete interrupt signal appears on interrupt line 20, 100, 120, 140, 160, 200, or 220 depending on the designated buffer.

NOTE: If the input device is a unit record device (such as card reader or magnetic tape) it is possible to specify buffer input which requires more information than is available on the input record. In this case, the device sends an input disconnect after the last word of information is received from the unit record. This signal causes the input buffer instruction to be terminated. The number contained in the buffer entry register, on completion of input, is the LWA + 1 of the information actually received from the unit record device.

INITIATE BUFFER OUTPUT

FFFEE	G	Mnemonic	Name	Timing
07300	XXXXX	IBO	Initiate Buffer Output	1 (no jump) 2 (jump)
1007Y	XXXXX	IBOY	Initiate Buffer	1 (no jump) 2 (jump)

If buffer is busy, XXXXX ⟶ P

If buffer is not busy, (P) + 2 ⟶ P

Start output buffering operation and when complete, generate a buffer complete interrupt for the designated buffer.

The IBO instruction initiates an output buffer operation on the buffered I/O channel. Prior to an IBO instruction, the external device must be selected; and BER, BXR, and the buffer storage bank control (b) must be set. If the buffer is in operation, no buffer action is taken, and control continues at location (r) XXXXX. If the buffer is not in operation, the buffering operation is started, and control immediately continues at location (r) (P) + 2. When the buffer operation terminates, an interrupt signal appears on interrupt line 20, 100, 120, 140, 160, 200, or 220 depending on the buffer (1 through 7) which terminates

NOTE: If instruction IBI, IBO, IBIY, or IBOY occurs and no external device has been selected, the computer continues in operation; but the buffer is placed in an indefinite busy status, and no input or output operation takes place. This busy status may be removed by the instruction CBC or by a master clear from the Console.

NORMAL INPUT AND OUTPUT

FFFEE	G	Mnemonic	Name	Timing
072XX	YYYYY	INP	Normal Input	3+N*
073XX	YYYYY	OUT	Normal Output	3+N*

((r) (P) + 000XX)= FWA of the input or output area.

YYYYY = the LWA + 1 of the input or output area.

Perform a read or write operation with the previously selected external device. The I/O area is defined as follows:

1. The FWA of the I/O area is found XX locations forward in the relative memory bank (r). This location (r) (P) + 000XX, specifies a FWA in the indirect memory bank (i).

2. The LWA + 1 of the I/O area is location YYYYY in the indirect memory bank (i) YYYYY.

If the external device has been properly selected, the input or output operation takes place; and, at its completion, control continues at (r) (P) + 2. If no external device has been properly selected, the computer is indefinitely delayed, and the operation mode (INP, OUT) is displayed on the COMPUTER STATUS indicator.

Instructions, INP and OUT, which are used to transfer data on the normal I/O channel, are not buffered. The computer waits while the input or output operation is in progress, and the next instruction is not executed until the I/O operation is completed. The contents of A at the completion of an INP or OUT instruction indicates the LWA + 1 actually referenced during the input or output operation. Although the FWA and LWA of the I/O area are found in the relative memory bank (r), they actually specify locations in the indirect memory bank (i).

*The speed of these instructions varies with the speed of the external equipment. N equals the number of words transferred.

The E portion of INP and OUT instructions cannot be equal to zero, because the operation codes would be interpreted as IBI and IBO instructions, respectively.

NOTES:

1. When the internal buffer data channel is not in use, the equipment that is physically attached to the internal buffer data channel is available to the programmer, as if it were on the normal channel. The two channels are logically combined into one normal channel when the buffer is not busy. The programmer may use normal channel instructions to operate equipment on the internal buffer.

2. Issuing normal channel instructions to equipment on the buffer data channel, while the internal buffer is busy, results in a delay of the program until the buffer channel is not busy.

3. The following is an example of input instructions to read 1000_8 words into location 01750 to 02747 of the indirect bank.

Location	FFFEE	Comments
(r)01000	INP03	Input instruction (FWA is in (r) 01003)
(r)01001	02750	LWA + 1 of area
(r)01002	NZF2	Unconditional jump (A) = 02050
(r)01003	01750	FWA of input area
(r)01004	Next instruction	

4. If the input device is a unit record device (such as cards or magnetic tape) it is possible to specify an input instruction which requires more information than

is available on the input record. In this case, the device sends an input disconnect after the last word of information is received from the unit record. This signal causes a word of zeros to be stored following the last word from the unit record and the input instruction to be terminated. The number contained in A, on completion of input, is then LWA + 2 of the information actually received from the unit record device.

5. On completion of a normal input or output, the A register contains the LWA + 1 of the information transmitted except as noted in note 4.

OUTPUT NO ADDRESS

FFFEE	G	Mnemonic	Name	Timing
074YY		OTN	Output No Address	1*

Write on the previously selected output device one word with zero in the seven higher-order bits and YY in the six lower-order bits. The output operation takes place on the normal I/O channel. At the completion of the output operation, control continues at location (r) (P) + 1. If an OTN instruction is given and no external device has been properly selected, the computer is indefinitely delayed.

A REGISTER I/O

FFFEE	G	Mnemonic	Name	Timing
07600		INA	Input to A	1*
07677		OTA	Output from A	1*
1005Y		INAY	Input to A, Channel Y	1*

*Execution time varies with the speed of the external equipment being used.

Read or write by utilizing the previously selected external device, one word to or from A. The operation takes place on the normal I/O channel. On devices which transmit less than one full computer word at a time, the information is transmitted to and from the lower-order portion of A. At the completion of the operation, control continues at location (r) (P) + 1. If an INA, OTA, or INAY instruction is executed and no external device has been properly selected, the computer is indefinitely delayed.

NOTES:

1. Y for the INAY instruction must be a number from 2 through 7.

2. INA or INAY is normally used for input of a status response from an external control unit. It may be used for normal input of information, one word at a time, from a device which allows a slower input rate than is provided by the buffer or normal I/O instruction.

COMPUTER-TO-COMPUTER INTERRUPT

FFFEE	G	Mnemonic	Name	Timing
10102		CTCI	Computer-to-Computer Interrupt	1

The CTCI instruction causes an interrupt signal to be placed on the common computer-to-computer interrupt line (interrupt 240).

At the completion of CTCI instruction, which does not cause control to be transferred, control continues in the relative bank at the location specified by (P) + 1.

NOTES:

1. The computer-to-computer interrupt line is a common line which interrupts all computers tied to it, including the computer issuing the interrupt. Thus the program must make provision in the process of issuing the CTCI to allow for this interrupt.

2. Normal programs, using this instruction, make provisions for storing data in common memory concerning the action required on the occurrence of the interrupt, as well as who issued the interrupt, and who is to respond.

SPECIAL INSTRUCTIONS

NO OPERATION

FFFEE	G	Mnemonic	Name	Timing
0000X		NOP	No Operation	1
1000X		NOPG	No Operation	1

When a NOP or NOPG instruction is executed, the computer does not perform any function but passes on to the next instruction at location (r) (P) + 1. X is a number 1 through 7.

MODE SETTINGS

FFFEE	G	Mnemonic	Name	Timing
10100		AMOD	Select A Mode	1
10101		GMOD	Select G Mode	1

These instructions are used to condition the computer for their respective modes of operation.

In the A mode of operation, all arithmetic is performed modulo $2^{12}-1$, including the advancement of the P register. Addresses are formed as 12-bit numbers. The additional bits, to form the 17-bit storage address, are provided by the 5-bit storage bank controls.

In the G mode of operation, all arithmetic is performed modulo $2^{13}-1$, including the advancement of the P register. Addresses are formed as 13-bit numbers. The upper four bits of the storage bank controls provide the additional bits to form the 17-bit storage address.

NOTES:

1. All data read from storage is 13-bit. It is possible to read and interpret all instructions in the 160G repertoire even though the computer is operating in the A mode (12-bit arithmetic).

2. The 160G computer is provided with the A mode of operation to provide program compatibility with the 160-A.

3. All arithmetic is 12- or 13-bit, depending on the mode of operation. However, in the A mode, addition, subtraction, addressing, and logical operations are 12-bit; in the G mode these operations are 13-bit.

4. There are some instructions in the 160G repertoire which are always 12-bit, regardless of mode of operation. These instructions are the LS3, LS6, MUT, and MUH instructions. The LS3 and LS6 instructions leave bit 12 (the highest-order bit) unchanged in the G mode of operation. The 12-bit shift is end-around.

5. There are some instructions that are 13-bit regardless of mode of operation. These instructions are the multiply, divide, and variable shifts.

6. In the A mode (12-bit arithmetic) the value of the highest-order bit (bit 12) is considered as undefined. When operating in the G mode (13-bit) on a quantity which was generated in the A mode (12-bit), the correct sign of the quantity can be guaranteed by doing a LS1 - RS1 sequence of instructions.

7. When switching from one mode to another, the action on the P register is also changed. The following actions occur in the 160G to guarantee that program control continues correctly after a switch in modes. In switching from G to A mode, bit 12 of the P register replaces bit 00 of the relative bank control register. In switching from A to G mode, bit 00 of the relative storage bank control register replaces bit 12 of the P register.

8. The mode should not be switched when the program is at location 07776, 07777, 17776, or 17777 of any bank. The incrementing of the P register (P + 1) is done in the mode prior to the mode switch. An incorrect next address would be formed as a result. For example, when switching from A to G mode with the instruction at 07776, the next instruction would be 00000 instead of 07777.

9. In the A mode, it is possible to obtain 13-bit information from the memory and to store 13-bit information in memory, as long as no arithmetic or logical operation is performed between the load and store.

10. As a result of any arithmetic or logical operation, bit 12 (the highest-order bit) is always set to 0.

BANK CONTROLS TO A

FFFEE	G	Mnemonic	Name	Timing
00130		CTA	Bank Controls to A	1

$(b) \longrightarrow A_{11-9}$*

$(d) \longrightarrow A_{8-6}$

$(i) \longrightarrow A_{5-3}$

$(r) \longrightarrow A_{2-0}$

$(P) + 1 \longrightarrow P$

*Subscripts indicate bit positions in the specified register.

Transfer the contents of the four storage bank controls to the A register as octal digits which occupy the preceding A register bit positions. Control continues at (r) (P) + 1. The storage bank controls are not changed by the execution of a CTA instruction.

BANK CONTROLS TO AQ

FFFEE	G	Mnemonic	Name	Timing
10130		CTAQ	Bank Controls to AQ	1

$(b) \longrightarrow A_{10-6}$

$(d) \longrightarrow A_{4-0}$

$(i) \longrightarrow Q_{10-6}$

$(r) \longrightarrow Q_{4-0}$

$(P) + 1 \longrightarrow P$

Transfer the contents of the four storage bank controls to AQ in the bit positions shown. Control continues at (r) (P) + 1. The storage bank controls are not changed by execution of a CTAQ instruction.

NOTES:

1. CTA provides only the lower three bits of each bank setting. The instruction is provided for compatibility with the 160-A.

2. No instructions are provided to read the setting of the external buffer bank controls. If the programmer requires the information, there should be provision in the program for the storage of the current setting of the external buffer bank controls.

EXCHANGE A AND Q

FFFEE	G	Mnemonic	Name	Timing
10104		XAQ	Interchange A and Q	1

$$(A) \longrightarrow Q, \quad (Q) \longrightarrow A$$

The contents of the A and Q registers are interchanged.

SET BUFFER ENTRANCE REGISTER

FFFEE	G	Mnemonic	Name	Timing
00105	XXXXX	ATE	A to BER	1 (no jump) 2 (jump)
1002Y	XXXXX	ATEY	A to BER, Channel Y	1 (no jump) 2 (jump)

If the buffer is busy, XXXXX ⟶ P

If the buffer is not busy, (A) ⟶ BER
(P) + 2 ⟶ P

Transfer the contents of A to the designated BER. Control continues at (r) (P) + 2. If the buffer is in operation when an ATE or ATEY instruction is executed, A is not transferred, and control continues at (r)XXXXX.

NOTES:

1. Y has any value, 2 through 7.
2. ATE is used to set the internal buffer entrance register; ATEY is used to set the external buffer entrance register.
3. The following examples illustrate GASS coding of these instructions:

ATE *	Set internal buffer; exit to this instruction if BFR is busy.
ATE3 STRT	Set external buffer data channel 3; exit to STRT if BFR is busy.

SET BUFFER EXIT REGISTER

FFFEE	G	Mnemonic	Name	Timing
00106	XXXXX	ATX	A to BXR	1 (no jump) 2 (jump)
1003Y	XXXXX	ATXY	A to BXR Channel Y	1 (no jump) 2 (jump)

If the buffer is busy, XXXXX $\longrightarrow$ P

If the buffer is not busy, (A) $\longrightarrow$ BXR
(P) + 2 $\longrightarrow$ P

Transfer the contents of A to the designated BXR. Control continues at (r) (P) + 2. If the buffer is in operation when an ATX or ATXY instruction is executed, A is not transferred, and control continues at (r)XXXXX.

NOTES:

1. Y has any value, 2 through 7.
2. ATX is used to set the internal buffer exit register; ATXY is used to set the external buffer exit register.
3. The following examples illustrate GASS coding of these instructions.

ATX BEG	Set internal buffer exit register. If buffer is busy, go to BEG.
ATX5 *	Set external buffer exit register, channel 5. If buffer is busy, go to this instruction and try again.

BER TO A REGISTER

FFFEE	G	Mnemonic	Name	Timing
00107		ETA	BER to A	1
1004Y		ETAY	BER, Channel Y, to A	1/*

(BER) ⟶ A

(P) + 1 ⟶ P

Transfer the contents of the designated BER to A. This instruction may be executed at any time, even while the buffer is in operation. Control continues at (r) (P) + 1.

NOTES:

1. Y has any value, 2 through 7.
2. This instruction allows the programmer to examine the progress of the buffering operation, since the BER is increased by one for each word of data buffered.
3. ETA is used to examine the progress of the internal buffer; ETAY is used to examine the progress of the external buffers.

*Maximum execution time varies depending on length of cables between Compute Modules and I/O Module.

STORE P REGISTER

FFFEE	G	Mnemonic	Name	Timing
0015X		STP	Store P at location 5X	2

(P)⟶(d) 0005X

(P) + 1⟶P

Transfer the contents of P to storage location (d)0005X, where X is any octal digit. Control continues at location (r) (P) + 1. This instruction allows the contents of P, which contains the address of the STP instruction, to be stored in the direct storage bank at any location 00050 through 00057.

NOTE: The following examples illustrate the GASS coding for this instruction.

STP 3 Store the contents of P at 53_8.

STP 5 Store the contents of P at 55_8.

STORE AND RESET BER

FFFEE	G	Mnemonic	Name	Timing
0016X		STE	Store BER at location 6X and transfer A to BER	3

(BER)⟶(d) 0006X

(A)⟶BER

(P) + 1⟶P

Transfer the contents of the internal buffer BER to storage location (d) 0006X, where X is any octal digit, and transfer the contents of A to the BER. A is unchanged by this instruction. Control continues at location (r) (P) + 1. This instruction allows the contents of the BER to be exchanged whenever desired, even while the buffer is in operation.

NOTES:

1. This instruction is provided to allow the programmer complete control of the internal buffer and to process a continuous stream of input data. By examining the BER, the program can establish that the buffering operation is in the change-over area. The STE instruction then allows the program to initiate the buffering operation to a new input area and also determines where the old input area was terminated.

2. The following example illustrates the GASS coding for this instruction:

STE,5	Store BER internal buffer at 65_8 and set new starting value from A register.

BLOCK STORE

FFFEE	G	Mnemonic	Name	Timing
00100	XXXXX	BLS	Block Store	1 + n (no jump) 2 (jump)

If the buffer is busy, XXXXX ⟶ P

If the buffer is not busy, (A) ⟶ BFR

(P) + 2 ⟶ P

Start the cycle: (BFR) ⟶ (b) (BER)

(BER) + 1 ⟶ BER

If (BER) $\neq$ (BXR), repeat cycle

If (BER) = (BXR), terminate buffer operation.

BLS stores the number contained in the A register at each location in a block of memory as defined by the internal buffer. Prior to executing a BLS instruction, the programmer must see that the FWA of the area to be set is placed in BER, the LWA + 1 of the area to be set is placed in BXR. (b) is then set to reference the storage bank that is to be set. The value to which the area is to be set is placed in A; then the BLS instruction is given. If at the time a BLS instruction is given the buffer is in operation, program control is transferred to (r)XXXXX. If the buffer is not in operation when the BLS instruction is given, the store operation takes place, and control continues at (r) (P) + 2.

The buffer storage bank control (b) may be set at any time prior to starting the BLS instruction. Because of the need to refer constantly to memory for the store operation, the next instruction is not executed until the BLS instruction is completed. The total execution time for the BLS operation is 1 + n cycles, where n is the number of locations to be set plus one.

MEMORY-TO-MEMORY TRANSFER

FFFEE	G	Mnemonic	Name	Timing
1016Y	XXXXX	MTMY	Memory to Memory Channel Y	(no jump) 1 (jump) 2

If buffer, Channel Y, is busy, XXXXX$\rightarrow$P

If buffer is not busy, initiate buffer channels Y and Y' for data transfer.

The MTMY instruction is used to move blocks of information from one portion of memory to another. The transmitting and receiving areas may be in the same or different memory banks. The transmission of data occurs within an I/O Module, over a pair of buffer data channels. An I/O Module has a pair of buffer data channels, designated Y and Y'. The channels have been arbitrarily assigned odd and even numbers, 2 through 7. The first I/O Module has 2 assigned to buffer data channel Y; 3 is assigned to channel Y'. Channel Y, in the second I/O Module, is assigned to 4, and so forth.

The MTMY instruction sets up a cross-connection between a pair of external buffer data channels (channels Y and Y'). The transmission of data is initiated by an IBOY instruction. This instruction initiates an output on the even-numbered data channel (channel Y). The data is transmitted from memory, via channel Y, to the I/O Module where the cross-connection is made and the data is transmitted back to memory via channel Y'. A buffer termination interrupt occurs on channel Y at the completion of the memory-to-memory transfer operation.

Prior to issuing the MTMY and IBOY instructions, the program must set the channel Y (even-numbered channel) buffer bank control, BER, and BXR to define the data to be transmitted. The program must also set the channel Y' (odd-numbered channel) buffer bank control and BER to define the receiving area for the data. The two instructions, MTMY (to set up the cross-connection), and IBOY (to initiate output) are then given and the transmission of data proceeds one word at a time until (BER) = (BXR) on channel Y. If channel Y or Y' is in operation when the MTMY instruction is given, the program control jumps to (r) XXXXX. The buffering operating takes place independently of the Computer Module. It is possible to have three memory-to-memory transfers occurring at the same time.

HALF WRITE

FFFEE	G	Mnemonic	Name	Timing
076XX		HWI	Half Write Indirect	3/2
167ZZ	XXXXX	HW	Half Write Entire Memory	3/2

$$(A)_E \longrightarrow \text{Operand}_E$$

Transfer the E portion of the A register (lower six bits) to the E portion of the operand. The contents of the A register and the F portion of the operand are not changed. The E portion of the HWI instruction makes an initial reference to the direct bank ((d) 000XX). This address contains the address of the operand in the indirect bank. For the HW instruction, ZZ specifies the bank and XXXXX the address of the operand within that bank. At the completion of these instructions, control continues as described in the section on addressing modes.

> NOTE: For an HWI instruction the E portion must not equal 00 or 77 because the instruction would then be translated as INA or OTA, respectively.

ERROR REGISTER TO A

FFFEE	G	Mnemonic	Name	Timing
10105		ERTA	Error Register to A	1

The execution of the ERTA instruction causes the contents of the error register to be displayed in the A register. The error register displays parity errors over the buffer data channels and overflow conditions which arise during arithmetic processes. At the completion of the ERTA instruction, control continues at (r) (P) + 1.

The error register contains error conditions indicated by a 1 in the specified bit position. The bit positions in the register and the conditions indicated by them are as follows:

Bit Positions in A	Condition
12 and 11	Add or subtract overflow
10	Computer parity error
7	Storage parity error, buffer data channel 7

Bit Positions in A	Condition
6	Storage parity error, buffer data channel 6
5	Storage parity error, buffer data channel 5
4	Storage parity error, buffer data channel 4
3	Storage parity error, buffer data channel 3
2	Storage parity error, buffer data channel 2
1	Storage parity error, internal buffer data channel
0	Divide overflow, or divide by zero

Add or subtract overflows are sensed by an illegal sign change in an add or subtract operation. An illegal sign change is to add two positive quantities and get a negative result or add two negative quantities and get a positive result. The arithmetic error bits (12, 11, and 0) and the computer parity error bit (10) are set to zero at the completion of an ERTA instruction. When the error bits are set, they remain set until an ERTA instruction is given. The buffer data channel storage parity error bits are set to zero by initiation of the next I/O operation for that channel, by a clear buffer controls (CBC or CBCY) instruction or by a master clear.

TRANSFER P TO A

FFFEE	G	Mnemonic	Name	Timing
00101		PTA	Transfer P to A	1

$(P) \longrightarrow A$

$(P) + 1 \longrightarrow P$

Transfer the contents of P to A. Control continues at (r) (P) + 1. At the time PTA is executed, P contains the address of the PTA instruction.

UNCONDITIONAL JUMPS

INDIRECT JUMPS

FFFEE	G	Mnemonic	Name	Timing
070XX		JPI	Jump Indirect	2/1.5
10106	XXXXX	JPIB	Jump Indirect Entire Bank	3/2.5

((d) 000XX) ⟶ P or ((d) XXXXX) ⟶ P

The E or G portions of the preceding instructions refer to a location in the direct bank. This location in the direct bank contains the jump address. Control is transferred to the relative bank at the location specified.

Example:	Location	F	E	G	Comments
	(d)00005	012	34		
	(d)13052	056	71		
	(r)01000	070	05		JPI transfer control to (r)01234
	(r)02000	101	06	13052	JPIB transfer control to (r)05671

JUMP FORWARD INDIRECT

FFFEE	G	Mnemonic	Name	Timing
071XX		JFI	Jump Forward Indirect	2

$$((r)\ (P) + 000XX) \longrightarrow P$$

Transfer program control to the address in (r) specified by the contents of the storage location XX positions following the JFI instruction. The E portion of the JFI instruction cannot be equal to zero because the operation code would be interpreted as a JPR instruction.

Example:	Location	FFFEE	Comments
	(r)01000	07103	Transfer control to 03517.
	(r)01003	03517	

JUMP RELATIVE INDIRECT

FFFEE	G	Mnemonic	Name	Timing
10107	XXXXX	JRIB	Jump Relative Indirect Entire Bank	3

JRIB is an unconditional jump. The jump address is obtained from the relative bank at (P) + XXXXX. Control is then transferred to the location specified by the jump address.

Example:	Location	FFFEE	G	Comments
	(r)00750	12345		Address of next instruction
	(r)01000	10107		
	(r)01001		17747	G portion = -30

Execution of the JRIB instruction at (r)01000 specifies the jump address at (r)01000 - (r)00030 or (r)00750. The jump address in (r)00750 specifies that the next instruction is to come from (r)12345.

UNCONDITIONAL JUMP

FFFEE	G	Mnemonic	Name	Timing
10114	XXXXX	UJRB	Uncondition Jump Relative Entire Bank	2

The UJRB instruction is an unconditional jump. Control is transferred, in the relative bank, to (P) + XXXXX.

RETURN JUMPS

FFFEE	G	Mnemonic	Name	Timing
07100	XXXXX	JPR	Return Jump	3

$$(P) + 2 \rightarrow (r)\ XXXXX$$

$$XXXXX + 1 \rightarrow P$$

The contents of P is increased by 2 to give the address needed to return to the main program following the JPR instruction. This address is transferred to location (r)XXXXX. Program control is then transferred to location (r)XXXXX + 1. This instruction does not change relative banks.

FFFEE	G	Mnemonic	Name	Timing
103ZZ	XXXXX	JPRG	Return Jump Entire Memory	4

JPRG is a return jump instruction which allows a transfer across memory banks and provides sufficient information for the program to return. The JPRG instruction occurs at (r) (P). The instruction stores (r) as the lower six bits of location XXXXX in bank ZZ. It then stores (P) + 2 at location XXXXX + 1 in bank ZZ. The relative bank control is then set to ZZ, and the program continues at XXXXX + 2 in bank ZZ. Normally, a program contains an SRJP (set relative bank control and jump) instruction at location XXXXX which is used as the return to the main program.

Example:

Main Program

Location	FFFEE	G	Comments
(0)00100	JPRG04		04 specifies the bank where the subroutine is located (bank 4)
(0)00101		01000	01000 is the address of the first instruction in the subroutine
(0)00102	Next instruction		

Subroutine

Location	FFFEE	G	Comments
(4)01000	SRJP00		Reset relative bank controls and jump back to main program
(4)01001		00102	Return address in main program
(4)01002			First instruction of subroutine
(4)01100	JFI01		Jump back to (4)01000 at the end of the subroutine
(4)01101		01000	

SENSE JUMPS

CONDITIONAL JUMPS

FFFEE	G	Mnemonic	Name	Timing
060XX		ZJF	Zero Jump Forward	1
061XX		NZF	Non-Zero Jump Forward	1
062XX		PJF	Positive Jump Forward	1
063XX		NJF	Negative Jump Forward	1
064XX		ZJB	Zero Jump Backward	1
065XX		NZB	Non-Zero Jump Backward	1
066XX		PJB	Positive Jump Backward	1
067XX		NJB	Negative Jump Backward	1
10110	XXXXX	ZJRB	Zero Jump Relative Entire Bank	1 (no jump) 2 (jump)
10111	XXXXX	NZRB	Non-Zero Jump Relative Entire Bank	1 (no jump) 2 (jump)
10112	XXXXX	PJRB	Positive Jump Relative Entire Bank	1 (no jump) 2 (jump)

FFFEE	G	Mnemonic	Name	Timing
10113	XXXXX	NJRB	Negative Jump Relative Entire Bank	1 (no jump) 2 (jump)
104ZZ	XXXXX	ZJ	Zero Jump Entire Memory	1 (no jump) 2 (jump)
105ZZ	XXXXX	NZ	Non-Zero Jump Entire Memory	1 (no jump) 2 (jump)
106ZZ	XXXXX	PJ	Positive Jump Entire Memory	1 (no jump) 2 (jump)
107ZZ	XXXXX	NJ	Negative Jump Entire Memory	1 (no jump) 2 (jump)

The conditions for testing are as follows:

Zero	The contents of A is equal to 00000 (plus zero). Minus zero is not considered equivalent to plus zero to meet the zero jump condition.
Non-Zero	A contains any quantity other than 00000.
Positive	Sign bit of A is 0.
Negative	Sign bit of A is 1.

If the condition tested is not met, control continues at (r) (P) + 1, for forward and backward jumps, and at (r) (P) + 2 for RB or EM jumps.

If the condition is met, a transfer of control is made as follows:

1. For jumps in the forward address mode (F), control is transferred, in the relative bank, to (P) + XX.
2. For jumps in the backward address mode (B), control is transferred, in the relative bank, to (P) - XX.

3. For relative entire bank jumps (RB), a transfer of control is made, in the relative bank, to (P) + XXXXX.

4. For entire memory jumps a transfer of control is made to location XXXXX in bank ZZ. The relative bank control is set to ZZ.

BIT-BY-BIT JUMP

FFFEE	G	Mnemonic	Name	Timing
10115	XXXXX	BITJ	Bit-by-Bit Jump	2

If the contents of A is zero, control is transferred to location XXXXX in the relative bank. If more than one bit of the accumulator is a one, control is continued at ((r) (P) + 2). If only one bit of the accumulator is a one, control is transferred to the address in (r) specified by the content of one of the 13_{10} storage locations from (r) (P) + 3 to (r) (P) + 17_8. The transfer to the address specified by (P) + 3 is made if bit 0 of the A register is a 1; a transfer to address specified by (P) + 4 is made of bit 1 if the A register is a 1, etc.

Example:

The BITJ instruction is a 15-way jump depending on the number contained in the A register. The following program, starting at location 1000, shows its usage.

Location	FFFEE	G	Comments
(r)01000	10115		BITJ instruction
(r)01001		02000	Transfer program control to 2000 if no bits are in the A register.
(r)01002	02010		Transfer program control to 2010 if more than one bit is in the A register.

Location	FFFEE	G	Comments
(r)01003	02020		Transfer program control to 2020 if bit 0 of the A register is a 1.
(r)01004	02040		Transfer program control to 2040 if bit 1 in the A register is a 1.
(r)01005	02060		Transfer program control to 2060 if bit 2 in the A register is a 1.
(r)01006	02100		Transfer program control to 2100 if bit 3 in the A register is a 1.

Only as many jump exits need to be written as there are bits to be tested since this is a 15-way branch instruction with no continuation.

LIMIT COMPARISON

FFFEE	G	Mnemonic	Name	Timing
173ZZ	XXXXX	HILO	High-Low Comparison	4

This instruction contains two jump addresses, one located 2 positions after the HILO instruction, and the other located 3 positions after the HILO instruction. (The G portion of the instruction is located 1 position after the HILO instruction.) There are two constants located in bank ZZ at addresses XXXXX and XXXXX + 1. The instruction subtracts the contents of the accumulator from the first constant. If the result is negative, control is transferred to the first jump address. If the result of the subtraction is positive, the second constant is subtracted from the original contents of the accumulator. If the result of this subtraction is negative, control is transferred to the second jump address. If the result of this subtraction is positive, control continues at (P)+ 4. In any case, the original contents of the accumulator is preserved. The contents of the Q register is changed during the execution of the instruction.

NOTE: The constants at XXXXX and XXXXX + 1, in bank ZZ, normally are taken as an upper and lower limit, respectively. The instruction takes the first jump exit if the number in A is greater than the upper limit. The second jump exit is taken if the number in A is less than the lower limit. If the number in A is between or equal to the two limits, the next instruction is taken from (r) (P) + 4.

Example:	Location	FFFEE	G	Comments
	(r)01000	17302		Compare limits at bank (2)01234 with bank (2)01235
	(r)01001		01234	
	(r)01002	XXXXX		Jump address if greater than high limit.
	(r)01003	YYYYY		Jump address if less than low limit.
	(r)01004	Next instruction		

SELECTIVE STOP AND JUMP

FFFEE	G	Mnemonic	Name	Timing
0770Y		SLS	Selective Stop	1
077Y0	XXXXX	SLJ	Selective Jump	1 (no jump) 2 (jump)
077YY	XXXXX	SJS	Selective Stop and Jump	1 (no jump) 2 (jump)

The selective stop and jump instructions are controlled by six switches on the Console: SELECTIVE STOP switches 1, 2, and 4, and SELECTIVE JUMP switches 1, 2, and 4.

SLS	If the appropriate SELECTIVE STOP switch is set, stop. If the computer stops, computation may be resumed by depressing the START switch on the Console. Whether the computer stops or not, control continues at location (r) (P) + 1.
SLJ	If the appropriate SELECTIVE JUMP switch is set, jump to location (r)XXXXX; otherwise, control continues at location (r) (P) + 2.
SJS	If the appropriate SELECTIVE JUMP switch is set, prepare to jump to location (r)XXXXX; otherwise, prepare to continue control at (r) (P) + 2. Then test the appropriate SELECTIVE STOP switch. If the appropriate SELECTIVE STOP switch is set, stop before executing the next instruction; control continues as selected by the SELECTIVE JUMP switches. If a stop occurs, computation may be resumed by depressing the START switch on the Console.

The values of Y and the switches they control are as follows:

Y	Test Switch
1	1
2	2
3	1 and 2. If either switch is set, a stop or jump is made as defined previously.
4	4
5	1 and 4. If either switch is set, a stop or jump is made as defined previously.
6.	4 and 2. If either switch is set, a stop or jump is made as defined previously.

Y	Test Switch
7	1, 2, and 4. If any of them is set, a stop or jump is made as defined previously.

The three lower-order bits of the E portion of the instructions control the testing of the SELECTIVE STOP switches. The three higher-order bits of E control the testing of the SELECTIVE JUMP switches.

Any value of YY is legal, except 00 and 77, which are treated as halt instructions.

HALTS

FFFEE	G	Mnemonic	Name	Timing
00000		ERR	Error Stop	1
07700		HLT	Halt	1
07777		HLT	Halt	1
10000		ERRG	Error Stop	1
177XX		HLTG	Halt	1

Stop computation. When the START switch on the Console is depressed, control continues at (r) (P) + 1. When ERR or ERRG instruction is executed and the computer stops, the letters ERR are displayed in the COMPUTER STATUS indicator on the Console. There is no difference in the action of the three HLT instructions. The E portion of the HLTG instruction allows 64_8 distinguishable halts.

EXTERNAL FUNCTIONS

FFFEE	G	Mnemonic	Name	Timing
07500	0ZZZZ	EXC	External Function Constant	2*

*Execution time varies with the speed of the external equipment being used.

FFFEE	G	Mnemonic	Name	Timing
075XX		EXF	External Function Forward	2*
1015Y	0ZZZZ	EXCY	External Function, Channel Y	2*

Transmit the 12-bit operand (0ZZZZ) to the external equipment. For the EXC or EXCY instruction, the operand (0ZZZZ) is obtained from the G portion of the instruction. For EXF, the operand is obtained from the location XX positions following the EXF instruction or from the location (r) (P) + XX.

At the completion of an external function instruction, A contains the external function code.

The operand (normally 12-bits) to be transmitted is known as an external function code. The external function instruction is used to select an external device to perform some specific function. With the exception of a status request code, an illegal selection causes the computer to be indefinitely delayed and to display SEL on the COMPUTER STATUS indicator. An example of an illegal selection is the attempt to select a magnetic tape unit when the magnetic tape unit is turned-off. The selection of a non-existent control unit also causes a delay.

Most external devices have a status request code. When such a code is given, a 12-bit status response code is sent to A by executing an INA or INAY instruction. By examining this response code, the computer determines whether further selection of the equipment is possible.

Only one device may be selected at any one time on each channel. Selection of any device automatically disconnects any other selected device. If a select signal is given and the channel is busy, the computer is delayed until the channel is free and a selection can be made. If a device is selected for some function and another select signal is made on the same device for some other function, the previous select signal is nullified.

*Execution time varies with the speed of the external equipment being used.

NOTE: The value of Y, for the EXCY instruction, is any number 2 through 7. External functions, for equipment on the internal buffer channel, are made by an EXC or EXF instruction.

ADDITIONAL OPERATION CODE TRANSLATIONS

To prevent an error stop in real-time applications, the 160G executes all possible operation codes, including those which have not been assigned to instructions in the repertoire. Each unused operation code is translated as one of the instructions in the 160G repertoire. The translations are as follows:

> NOTE: GASS contains no provision to produce these operation codes.

Unused Operation Code	Mnemonic Translation
00116	RS1
00117	RS2
0017X	SBU
102XX	LPN
10116	JPIB
10117	JRIB
11200	LPC
112XX	LPF
11600	SCC
116XX	SCF
12200	LDC
122XX	LDF
12600	LCC
126XX	LCF
13200	ADC
132XX	ADF
13600	SBC
136XX	SBF
14200	STC
142XX	STF
14600	SRC
146XX	SRF
15200	RAC
152XX	RAF
15600	AOC
156XX	AOF
163XX	HILO

USE OF THE INSTRUCTIONS - PROGRAMMING

Included in the 160G systems is a software assembly program (GASS) that allows the programmer to assign meaningful names for the instructions. The assembly program (assembler) translates the mnemonic coding to the actual machine (numeric) codes. The following programs include both the octal numeric codes and the GASS mnemonic codes. The first two columns contain the numeric coding; the next three columns contain the GASS coding. The last three columns and the comments are the only portion that is actually compiled by the programmer. The programmer must have a thorough understanding of the 160G and GASS coding instructions before writing any GASS programs. A complete description of the GASS coding instructions may be found in Control Data publication G01676.

The following are several internal programming problems with solutions which illustrate various uses of the 160G instructions. Some of the problems can be programmed in more than one way; but the method chosen, although in some cases not the best, serves as an illustration.

One programming convention, occurring throughout the examples, is used in most utility and general purpose programs developed by Control Data: The locations 00070 through 00077 of bank zero are used for temporary or transient storage of data, counters, and so forth. These locations should be avoided for program table or constant storage.

Figures 2-1, 2-2, 2-3, and 2-5 are the printouts of the four sample programs. Figures 2-4 and 2-6 are flow diagrams for the programs in Figures 2-3 and 2-5, respectively.

```
                                            PROBLEM = SET ALL STORAGE LOCATIONS FROM
                                            00100 TO 17700 INCLUSIVE IN BANK2 EQUAL TO ZERO
        040100              ORG             40100B    START PROGRAM BANK 4 LOCATION 100
040100  02200               LDC             100B                  FWA OF AREA TO A REGISTER
040101  00100
040102  00105               ATE             *                     A TO BER, IF BUFFER IS BUSY,WAIT
040103  00102
040104  02200               LDC             17701B                LWA+1 TO A
040105  17701
040106  00106               ATX             *                     A TO BXR, IF BUFFER IS BUSY,WAIT
040107  00106
040110  00142               SBU,2                                 SET BUFFER BANK CONTROL TO 2
040111  00400               LDN             0                     SET A=0
040112  00100               BLS             *                     DO CLEAR OPERATION USING BUFFER
040113  00112
                                            NEXT INSTRUCTION
                            END
```

Figure 2-1. Program Clearing Memory - Printout

```
                                    PROBLEM = TRANSFER 100 (DECIMAL) WORDS FROM LOCATION
                                    00700 IN INDIRECT BANK TO LOCATION 12300 IN
                                    INDIRECT BANK. PERFORM AS A JPR SUBROUTINE STARTING
                                    AT 500 (OCTAL)

        000477              ORG     477B
000477  07101               JFI     1                   EXIT FROM ROUTINE
000500  17777     MOVE              **                  RETURN ADDRESS STORED HERE
000501  02600               LCC     100                 OBTAIN = 100 COUNT
000502  00144
000503  04077               STD     77B                 SET (D) 00077 AS COUNTER
000504  02200               LDC     700B                FIRST WORD ADDRESS OF FROM AREA
000505  00700
000506  04205               STF     GET+1               SET PICKUP COMMAND
000507  02200               LDC     12300B              FIRST WORD ADDRESS OF TO AREA
000510  12300
000511  04204               STF     PUT+1               SET STORE COMMAND
000512  02100     GET       LDM     **                  MOVE ONE
000513  17777
000514  04100     PUT       STM     **                       WORD
000515  17777
000516  05703               AOB     GET+1               MODIFY FROM
000517  05702               AOB     PUT+1               MODIFY TO
000520  05477               AOD     77B                 COUNT +1
000521  06507               NZB     GET                 COUNT NOT ZERO DO AGAIN
000522  06423               ZJB     MOVE-1              ZERO,EXIT
                            END
```

Figure 2-2. Program Transferring Data - Printout

```
                                                GIVEN THE EQUATION  Y=V0T+1/2 GT2
                                                WHERE V0 = 160 FEET PER SECOND
                                                G = 32 FEET PER SECOND PER SECOND.
                                                SOLVE FOR VALUES OF T FROM 1 TO 10 SECONDS.
                                                STORE RESULTS IN MEMORY STARTING AT LOCATION 000100
                                                ASSUME THAT THE STORAGE BANK CONTROLS HAVE BEEN SET
                                                TO BANK ZERO. SEE FIGURE 2-4 FOR FLOW CHART
        001000                    ORG           1000B
001000  02200                     LDC           160
001001  00240
001002  04050                     STD           V0                    SET INITIAL VELOCITY
001003  00401                     LDN           1
001004  04051                     STD           T                     SET TIME = 1
001005  02200                     LDC           100B
001006  00100
001007  04052                     STD           RES                   SET RESULT AREA COMMAND
001010  02051         TLOOP       LDD           T
001011  17000                     MU            T                     OBTAIN T X T  IN AQ
001012  00051
001013  10104                     XAQ                                 GET T X T IN A
001014  17000                     MU            HALFG                 GET 1/2 G T X T
001015  01033
001016  16600         TEMPST      SQC           **                    SAVE AT TEMPST+1
001017  17777
001020  02050                     LDD           V0                    GET V0 X T
001021  17000                     MU            T
001022  00051
001023  10104                     XAQ
001024  03305                     ADB           TEMPST+1
001025  04152                     STI           RES                   HAVE Y
001026  05452                     AOD           RES                   PREPARE FOR STORE NEXT RESULT
001027  05451                     AOD           T                     T+1 TO T
001030  00713                     SBN           11                    CHECK DONE ALL 10
001031  06521                     NZB           TLOOP                 NO REPEAT
001032  07700                     HLT                                 STOP
001033  00020         HALFG                     16                    1/2 G= 1/2 X 32 = 16
        000050                    CON           50B
000050  00000         V0                                              INITIAL VELOCITY STORAGE
000051  00000         T                                               TIME
000052  17777         RES                       **                    RESULT ADDRESS
                                  END
```

Figure 2-3. Equation Solving Program - Printout

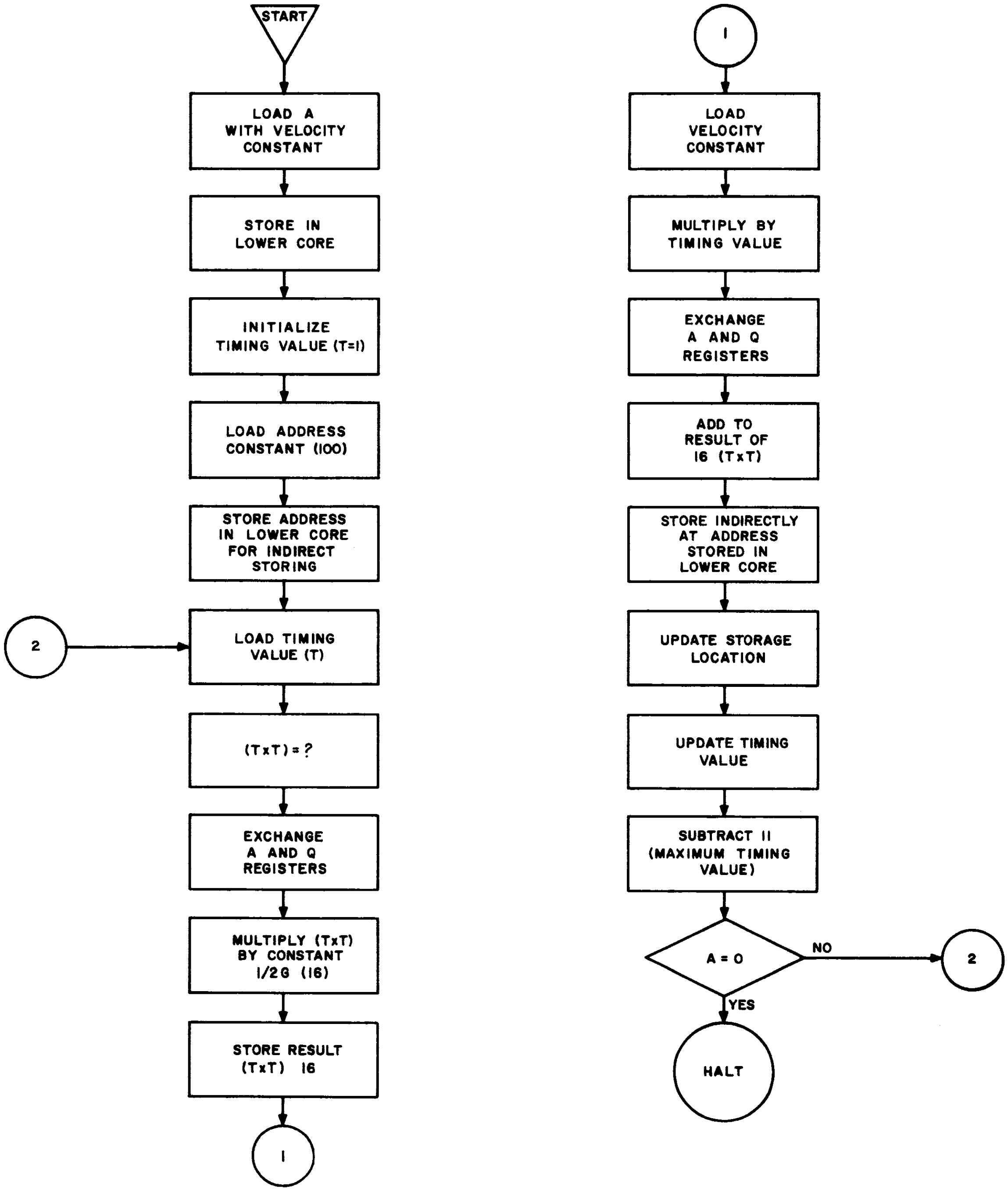


Figure 2-4. Equation Solving Program - Flow Diagram

```
                                PROBLEM - COUNT THE NUMBER OF POSITIVE,NEGATIVE,AND
                                ZERO NUMBERS IN A TABLE AT 011100 TO 011200
                                PUT COUNTS IN 000070,000071,000072. PROGRAM START
                                AT 030500.
                                SEE FIGURE 2-6 FOR FLOW CHART

        000070  POS     EQU     70B             DEFINE LOCATION OF COUNTERS
        000071  NEG     EQU     71B
        000072  ZERO    EQU     72B
        011100  TABL    EQU     011100B         DEFINE START OF TABLE
        030500          ORG     30500B
030500  02200           LDC     TABL
030501  11100
030502  04210           STR     FETCH+1         SET UP START OF PROGRAM
030503  00040           SDC,$POS                SET DIRECT BANK CONTROL TO CTR
030504  00400           LDN     0
030505  04070           STD     POS             CLEAR COUNTERS
030506  04071           STD     NEG             TO ZERO
030507  04072           STD     ZERO
030510  00021           SIC,$TABL               SET INDIRECT BANK TO DATA
030511  02100   FETCH   LDM     **              GET NUMBER
030512  17777
030513  06010           ZJF     ISZERO          IF ZERO GO TO ADD ONE TO ZERO CTR
030514  06211           PJF     ISPOS           IF POSITIVE ADD ONE TO PLUS CTR
030515  05471           AOD     NEG             MUST BE NEGATIVE ADD TO NEG CTR
030516  05704   LOOP    AOR     FETCH+1         ADVANCE ADDRESS FOR NEXT NUMBER
030517  03600           SBC     TABL+101B       CHECK DONE
030520  11201
030521  06510           NZB     FETCH           NOT DONE, REPEAT
030522  06005           ZJF     NXT             DONE EXIT
030523  05472   ISZERO  AOD     ZERO            ADD ONE TO ZERO COUNT
030524  06506           NZB     LOOP
030525  05470   ISPOS   AOD     POS             ADD ONE TO PLUS COUNT
030526  06510           NZB     LOOP
030527  07700   NXT     HLT                     NEXT INSTRUCTION
                        END
```

Figure 2-5. Program Using Counters - Printout

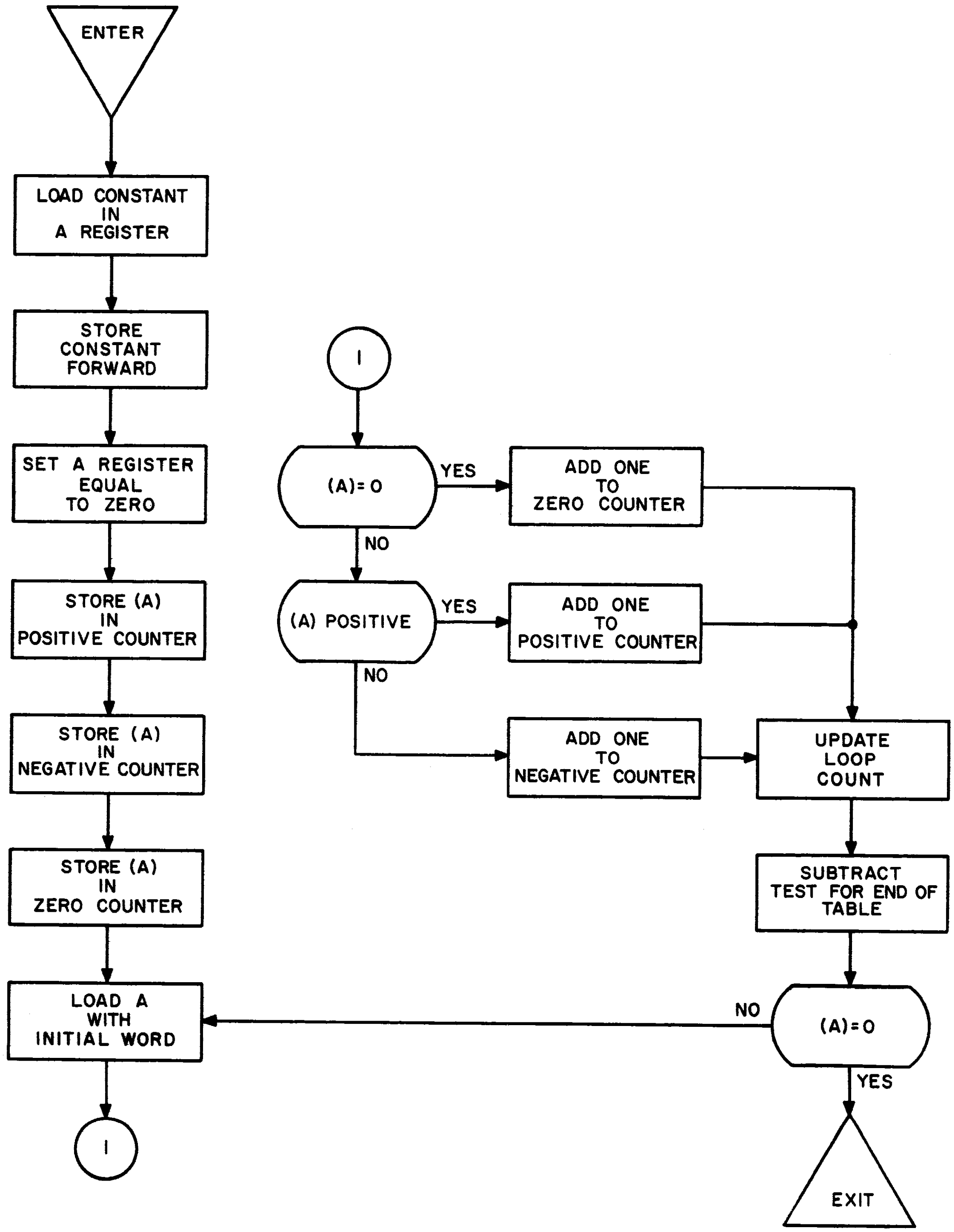


Figure 2-6. Program Using Counter - Flow Diagram

CHAPTER 3
INPUT/OUTPUT

COMPUTE MODULE

Each 160G Compute Module has two I/O channels, a normal, and a buffer channel. Both channels may be used simultaneously for any combination of I/O operations.

A device on the normal channel may be serviced by only a normal operation, but a device on the buffer channel may be serviced by either a normal or buffer operation. However, a device connected to the buffer channel may not be serviced by a normal operation if the buffer channel is busy.

When a normal input or output operation is performed by a Compute Module, the Compute Module is not free to continue computation. It must wait until the I/O operation is completed before computation can continue. Once an input or an output operation is started on a buffer channel, the Compute Module is free to continue computation, perform some other I/O operation on the normal channel, or set up another I/O operation on some other buffer channel.

The buffer channel of the 160G Compute Module has two external interrupts associated with it. These interrupt lines are numbered 30 and 40 (see the section on interrupts for a complete description).

I/O MODULE

A maximum of three I/O Modules can be controlled by a Compute Module. Each 171G I/O Module has a pair of buffer channels, designated Y and Y'. The channels are numbered 2 through 7. Channel Y is designated by the even numbers (2, 4, and 6); Channel Y' is designated by the odd numbers (3, 5, and 7).

There are certain instructions available that utilize the channel designations (IBIY, IBOY, and so forth). Either channel can be selected with these instructions (not limited to Channel Y), with the exception of the MTMY instruction. Channels on the I/O Modules can operate simultaneously with the channels on the Compute Module or other I/O Modules for any combination of I/O operations.

Each channel of an I/O Module has three interrupt lines associated with it: buffer termination interrupt, equipment interrupt 1, and equipment interrupt 2. These interrupt lines are self-explanatory and are discussed in the section on interrupts.

PERIPHERAL EQUIPMENT

All peripheral equipment may be divided into two categories: on-line equipment and off-line equipment. Equipment that is used to prepare data for the computer, which does not feed this data directly in, is called off-line equipment. The equipment used to pass data into or out of the computer is called on-line equipment. External devices may be connected so that they function off-line as well as on-line (magnetic tape units, and so forth).

External function codes are used to initiate communication between the computer system and the external equipment. External function codes are transmitted to equipment attached to the normal input line and the internal buffer by the EXC or EXF instruction. External function codes are transmitted to equipment attached to the external buffer data channels, 2 through 7, by the instructions EXC2, EXC3, EXC4, EXC5, EXC6, and EXC7, respectively. An external function code specifies a particular control unit and a function to be performed by the associated equipment. Normally, when a particular control unit is selected on a channel by an external function instruction, the other equipment on the channel is disconnected from the channel and does not respond to input or output operations until reselected by an external function instruction. A slight exception to this rule is the combined nature of the internal buffer channel and the normal channel. If the buffer is not busy, an external function selection to equipment, on the normal or buffer channel causes the other equipment on both channels to disconnect from the system. If the buffer is busy, a selection on the normal channel does not effect equipment on the buffer channel.

The following paragraphs briefly describe the various types of peripheral equipment currently available for the 160G computer system. A detailed description may be found in the instruction manuals for the various pieces of equipment.

176G TYPEWRITER

The 176G Typewriter unit is an optional input/output device for the 160G computer system. The typewriter unit provides the computer with a flexible monitoring input/output device. Through this medium data may be entered manually into the computer. In the output mode, monitoring information in a printed form may be received from the computer. Some examples of programming the 176G may be found in the following section.

405 CARD READER

The 405 Card Reader reads data from standard punched cards and transfers it to a computer, magnetic tape, or line printer. Data is read column-by-column and sent to associated equipment in a 12-bit per word, parallel mode. Figure 3-1 is a flow chart of the programming operations for the 405.

170G CARD PUNCH CONTROLLER

The 170G Card Punch Controller is a signal adapter used between the 160G Computer and the IBM 523 or 544 Card Punch. During a punch operation, the 170G assembles data from the computer into 80-column words and sets the punch magnets in the 523 Card Punch prior to a punch stroke.

Output operations with the 170G are initiated by coded EF instructions from the computer. Figure 3-2 is the complete programming operation for the 170G. To assure full speed operation, a select code should be issued as soon as possible after the punch resume signal is received.

1612G LINE PRINTER

The 1612G Line Printer functions as a data output device in direct communication with the 160G computer. The line printer operates in the character mode. During a 160G output instruction, the computer words needed to specify a line of print are sent to the 1612G. Only the lower six bits of each word are pertinent; the remaining higher-order bits are ignored.

Each 6-bit character code received by the line printer specifies one of the 64 possible characters. The order in which the characters are received specifies the character position within the proposed line of print. The 1612G stores this information in its own magnetic core

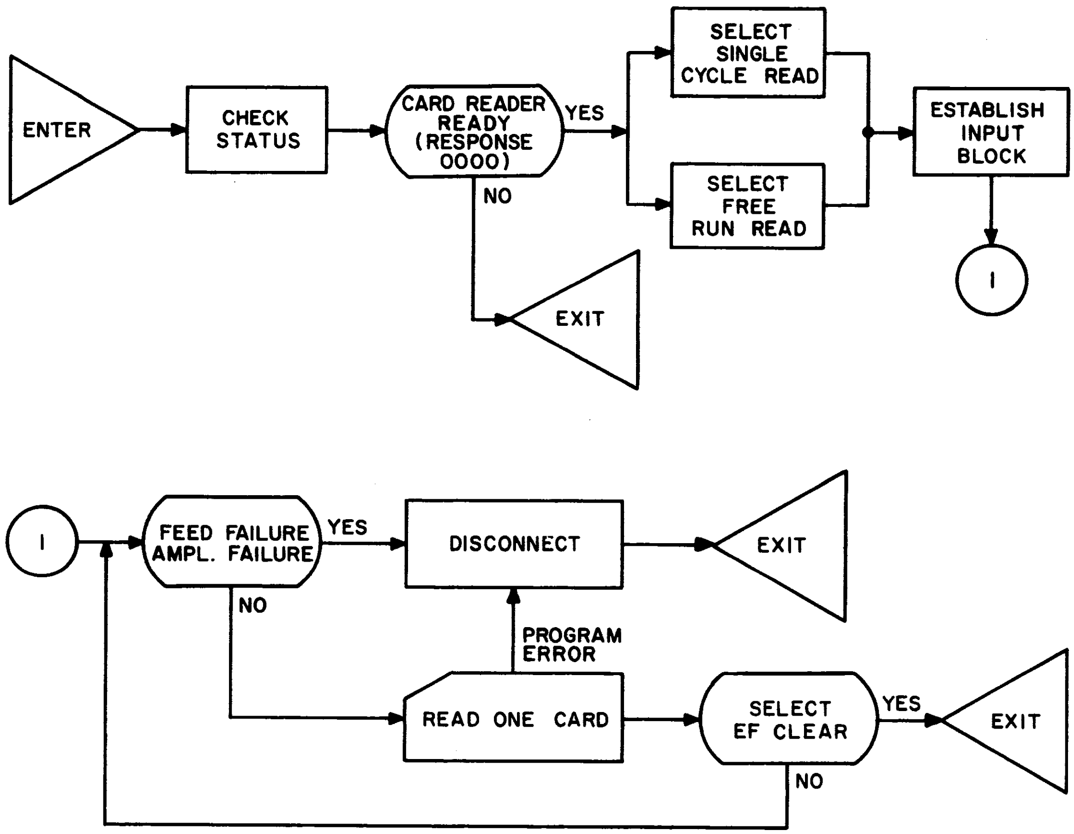


Figure 3-1. 405 Card Reader Operation - Flow Diagram

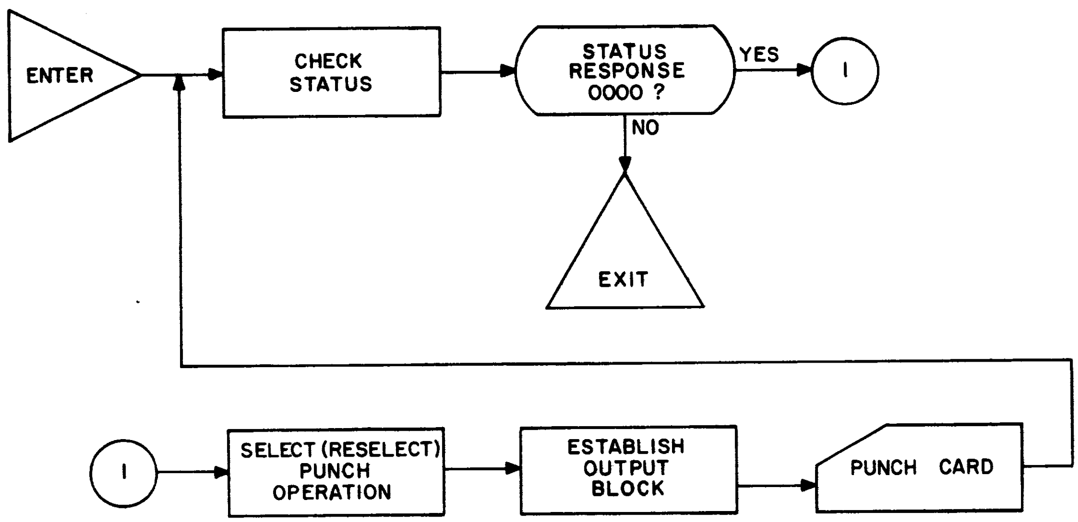


Figure 3-2. 170G Card Punch Controller Operation - Flow Diagram

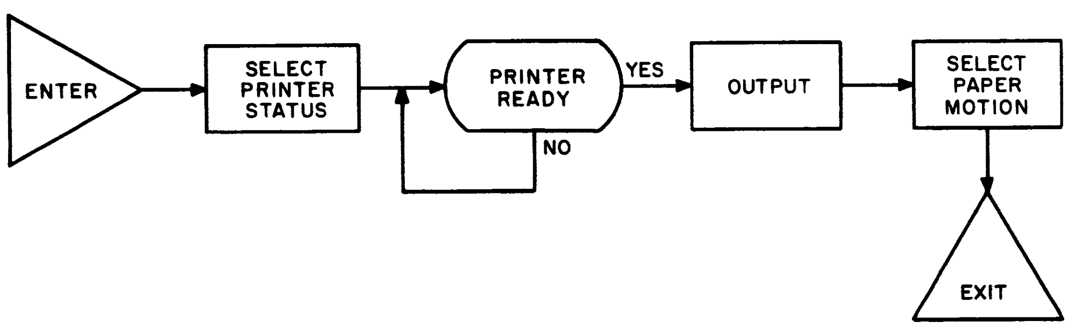


Figure 3-3. 1612G Line Printer Operation - Flow Diagram

storage unit. See Appendix F for a complete list of 1612G Printer Codes.

Figure 3-3 is the flow diagram for printer operation. After the manual controls are set, channel inactive sensed, printer selected and printer ready sensed, the program needs only to provide for successive buffer operation separated by sense channel inactive commands. As soon as one buffer terminates, the next may be initiated. The actual transmitting of information for the next line does not begin until the last character of the current line is printed.

165G PLOTTER

The 165G Plotter is a digital recorder that contains high-speed, two-axis recorders for plotting one variable against another. The plotters consist of a ballpoint pen mounted on a carriage and a bidirectional recording drum. Output words from the computer direct pen carriage movement and drum rotation as well as movement of the pen against or away from the recording surface.

All plotter operations are controlled by the computer and are initiated by coded external function (EF) instructions (Appendix D, page D-2). The upper six bits of the 12-bit code select the plotter control unit; the lower six bits specify mode of operation (read or write). In output mode, output instructions which follow the EF code direct plotter operation in graphing the relationship between any two variables.

606 MAGNETIC TAPE TRANSPORT

The 606 Magnetic Tape Transport is designed to provide the high performance storage capabilities required by the 160G.

The magnetic tape provides a high-speed, non-volatile storage medium for recording and permanently retaining information. Data is recorded in one of two formats: binary or binary coded decimal (BCD). The tape is binary if data is recorded just as it is represented in core storage. In BCD format, digits, characters, and special symbols are represented in core storage by 6-bit binary numbers. The BCD codes are listed in Appendix G of this manual. See Figures 3-8 through 3-10 for some examples of programming magnetic tape operations.

The 606 may be used with computers in an on-line capacity, or with external equipment in an off-line processing system. When used on-line, the operation of the 606 is externally controlled by the 160G Compute Module.

Transfer of data and exchange of control information from computers or off-line equipment to magnetic tape is via a separate external control unit, the 162G Magnetic Tape Synchronizer. The control unit provides the timing information necessary to buffer and control the flow of information into and out of the tape transport.

162G MAGNETIC TAPE SYNCHRONIZER

As stated previously, the 606 is controlled by the 162G; the 162G Magnetic Tape Synchronizer is, in turn, controlled by EF codes which are generated by the Compute Module. In computer magnetic tape operations, the computer EF code selects the following equipment and functions:

1. Synchronizer (162 system)
2. Tape handler (part of the system)
3. Recording mode
   a. Density
   b. Word format
   c. Parity mode
4. Operation

The 162G Magnetic Tape Synchronizer performs the following operations:

1. Status check
2. Preliminary steps
   a. Parity mode
   b. Density
3. Motion directives
   a. Search forward or backward to file mark
   b. Back space one record
   c. Rewind load or rewind unload

4. Information transfer

   a. Write
   b. Write file mark
   c. Read

See the program in the following section as an example of a method that can be used in programming magnetic tape operations.

8528G DIGITAL COMMUNICATION TERMINAL

The 8528G Digital Communication Terminal is an intercomputer communication device which provides long distance serial data transfer. It is entirely program controlled and can be used without modifications on the normal channel of the 160G with or without interrupt.

The terminal equipment does not differentiate between straight data and procedural information. This distinction must be made in programming.

Data flow between computer and terminal equipment is maintained by a sequence of control signals. All local terminal equipment operations are initiated by coded EF instructions. The upper six bits select the terminal equipment; the lower six bits specify the serial communication channel and operation requested. An accompanying function ready signal allows recognition of the 12 bits as an EF code. The EF codes vary with the type of computer using the terminal equipment.

The sense code response or status response indicates to the computer conditions within the terminal equipment. Either the presence or absence of the condition can be sensed.

PROGRAMMING I/O EQUIPMENT

The most general procedure for performing an I/O operation is as follows:

1. Request status of the selected equipment.
2. Test the status of the equipment for capability of performing the required function.
3. Select the equipment to perform the I/O function.
4. Initiate the I/O operation over the correct channel.
5. At the completion of an I/O operation, request the status

of the selected equipment and test to verify that the I/O operation was successfully completed.

Three of the five preceding operations are concerned with status requests. In order to allow the 160G Compute Module to have proper I/O control, most peripheral equipment is designed to transmit codes to the 160G Compute Module which informs the computer whether or not the equipment can be selected and whether or not an I/O operation was properly completed.

Checking status is not mandatory for I/O operations: steps 3 and 4 may be used alone. If status is not checked and the selected equipment is not capable of performing an I/O operation (power to the unit is off, for instance) when an EF instruction is given to select the unit for I/O operation, the computer is delayed indefinitely. During the delay the computer displays SEL on the Console COMPUTER STATUS indicator. Even if the equipment is turned off, but power is on, the control unit status may be requested and determined. Status is also requested at the completion of an I/O operation to test for conditions which might have occurred during the operations. Not all peripheral devices have codes for such conditions (for example, the 176G Typewriter). A peripheral device such as magnetic tape, however, by means of status responses, does check for such conditions as the following:

Tape Ready
End of file
End of tape
Parity error

After an initial status check is made, the external device is selected for input or output, and the correct input or output instructions are given to perform the operation.

Each individual peripheral device is programmed in a manner similar to all other peripheral equipment. The unit is selected and placed in the input or output mode. An EF instruction sends a 12-bit code on the output cable to all equipment connected to the computer. The upper six bits of the code specify the equipment; the lower six bits specify the operations requested. An accompanying function ready signal enables the external equipment to interpret the information as an EF code.

The control circuit interprets and stores the EF code and initiates the operation requested. It also sends an output resume signal to the computer to acknowledge acceptance of the EF code. For all operations, after the output resume is returned to the computer, the control circuits establish a lockout so that further computer EF requests are ignored until the current operation is complete.

A complete list of external function codes and status responses is contained in Appendix D of this manual.

The following descriptions, sample programs, and flow charts are used to illustrate some of the procedures used in programming individual pieces of peripheral equipment. They should not be used implicitly as they are intended to be used only as a guide. The manuals provided for each individual piece of equipment contain complete descriptions of the operation and programming procedures to be used.

INPUT ON THE NORMAL CHANNEL

After the external device has been selected for input, any combination of the following two instructions is used to store the incoming information in memory.

INP - Stores from 1 to 7777 words in memory (A mode)

Stores from 1 to 17777 words in memory (G mode)

INA - Stores one word in A.

When the INP instruction is used, the information is sent to the storage bank specified by the indirect storage bank control (i).

A status response code may be read into the A register by using either an INA or an INP instruction. An INP instruction takes more time, however. In addition, a load instruction is necessary to put the response in A for checking the status conditions when INP is used. Figure 3-4 illustrates how the INP and INA instructions are used in programming.

```
                                      PROBLEM - READ 21 CHARACTERS TYPED ON 176G
                                      TYPEWRITER INTO AN AREA STARTING AT (I)02222.
                                      THEN READ ONE MORE CHARACTER INTO A.
                                      PROGRAM TO START AT 1000 OCTAL

         001000             ORG       1000B
         002222    AREA     EQU       2222B
001000   07500     BEG      EXC       4240B              REQUEST TYPEWRITER STATUS
001001   04240
001002   07600              INA                          READ RESPONSE TO A REGISTER
001003   06003              ZJF       CONT               ZERO SAYS TYPEWRITER IS READY
001004   07700              HLT                          STOP IF STATUS NOT OK
001005   06505              NZB       BEG                TRY AGAIN
001006   07500     CONT     EXC       4220B              SELECT TYPEWRITER INPUT
001007   04220
001010   07204              INP       SA
001011   02247                        AREA+21            LWA+1 OF INPUT AREA
001012   07600              INA                          READ ONE MORE CHARACTER TO A
001013   06202              PJF       2
001014   02222     SA                 AREA               FWA OF INPUT AREA.
001015   07700              HLT                          NEXT INSTRUCTION
                            END
```

Figure 3-4. Normal Input Program – Printout

G02000c

OUTPUT ON THE NORMAL CHANNEL

After the external device has been selected for output, any combination of the following instructions is used to transfer the outgoing information from memory to the selected area.

OUT - Transfers from 1 to 7777 words from storage (A mode)

Transfers from 1 to 17777 words from storage (G mode)

OTA - Transfers one word from the A register

OTN - Transfers one word composed of the six higher-order zero bits and the six lower-order bits from the E portion of the instruction.

Figure 3-5 is a program that illustrates the use of the OUT instruction.

THE BUFFER CHANNEL

The following operations must be performed before an external device is selected for an input or output operation on the buffer channel:

1. Load the BER with the first word address of the I/O area.
2. Load the BXR with the LWA + 1 of the I/O area.
3. Select the proper storage bank for (b).

After the preceding steps are completed, the external device is selected, and an IBI, IBIY, IBO, or IBOY instruction starts the buffer operation, Figure 3-6. It is possible to select the external device prior to setting the buffer registers and (b). However, on certain external equipment such as magnetic tape units, the external instruction actually starts tape motion. Therefore, the operation is not properly completed if the correct data is not available to the tape unit at the time a data transfer request is issed by the tape unit.

If a buffer termination interrupt is desired at the completion of the buffer operation, a CIL instruction must follow the external function instruction. The execution of any external function instruction automatically sets master interrupt lockout. If a CIL instruction is not executed, all interrupts remain on the lines and do not take effect

```
                                          PROBLEM - TYPE 100 CHARACTERS ON THE I/O
                                          TYPEWRITER FROM THE AREA STARTING AT
                                          LOCATION (I)03200 AND THEN TYPE 120
                                          CHARACTERS FROM THE AREA STARTING AT (I)02701

        000500              ORG        500B
        003200    AREA1     EQU        3200B
        002701    AREA2     EQU        2701B
000500  07500     ST        EXC        4240B                    REQUEST TYPEWRITER STATUS
000501  04240
000502  07600               INA                                 READ STATUS TO A
000503  06003               ZJF        BEG                      IF ZERO BEGIN
000504  07700               HLT                                 HALT, NOT OK
000505  06505               NZB        ST                       GO BACK AND TRY AGAIN
000506  07500     BEG       EXC        4210B                    SELECT TYPEWRITER OUT
000507  04210
000510  07305               OUT        SA1
000511  03344                          AREA1+100
000512  07304               OUT        SA2
000513  03071                          AREA2+120
000514  06203               PJF        3
000515  03200     SA1                  AREA1
000516  02701     SA2                  AREA2
000517  07700               HLT
                            END
```

Figure 3-5. Normal Output Program - Printout

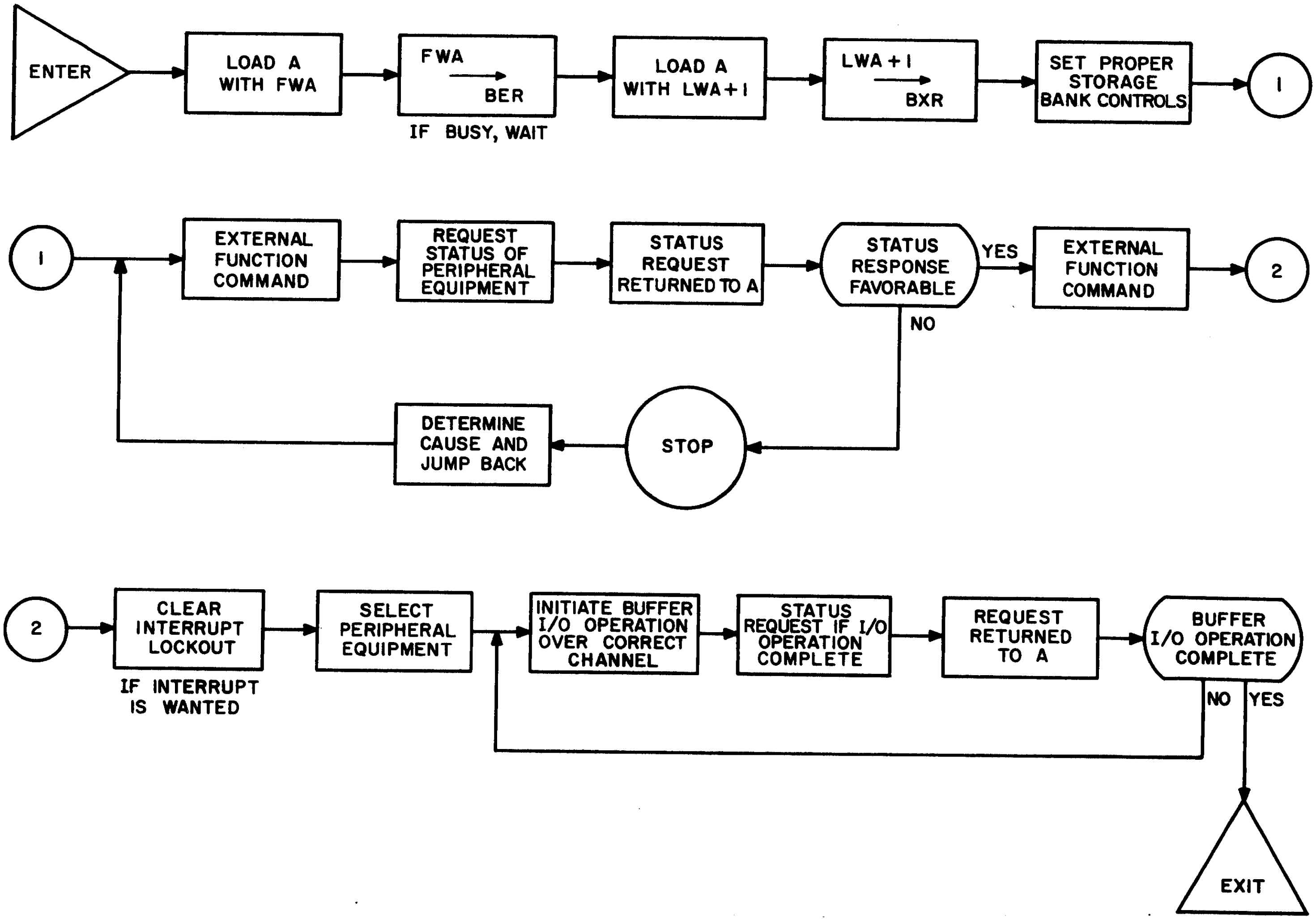


Figure 3-6. Buffer I/O Operation - Flow Diagram

until a CIL instruction is executed. Figure 3-7 is a program that illustrates the buffer operation. Some magnetic tape programs are illustrated in Figures 3-8 through 3-10. Note that none of these programs use the interrupt system or make provisions for interrupts. This feature of the machine is covered in the next section.

```
                                            PROBLEM - READ A MESSAGE OF NOT MORE THAN 50
                                            CHARACTER FROM THE I/O TYPEWRITER OVER THE
                                            INTERNAL BUFFER TO LOCATION 043700 OCTAL.
                                            PROGRAM TO START AT 1750 OCTAL.

        001750              ORG       1750B
001750  02200               LDC       AREA                FWA OF INPUT AREA
001751  03700
001752  00105               ATE       *                   PLACE IN BER, IF BUSY, WAIT
001753  01752
001754  02200               LDC       AREA+50             LWA+1 OF INPUT AREA
001755  03762
001756  00106               ATX       *                   PLACE IN BXR
001757  01756
001760  00144               SBU,$AREA                     SET BFR. BANK CONTROL TO 4
001761  07500               EXC       4240B               REQUEST TYPEWRITER STATUS
001762  04240
001763  07600               INA                           READ STATUS
001764  06004               ZJF       BIN                 START IF ZERO
001765  00720               SBN       20B                 CHECK FOR INPUT WAITING RESPONSE
001766  06002               ZJF       BIN                 IF ZERO NOW, READ, IF NOT
001767  07700               HLT                           TYPEWRITER NOT READY, SO STOP
001770  07500       BIN     EXC       4220B               SELECT TYPEWRITER INPUT
001771  04220
001772  00120               CIL                           CLEAR INTERRUPT LOCKOUT
001773  07200               IBI       *                   START BUFFER OPERATION
001774  01773
        043700      AREA    EQU       43700B
                            END
```

Figure 3-7. Buffer Input Program - Printout

```
                                         PROBLEM - WRITE ON TAPE 3, STARTING AT LOAD POINT
                                         ONE RECORD FROM 123456. THE RECORD IS 100 WORDS
                                         BINARY INFORMATION, HIGH DENSITY. AFTER WRITING,
                                         RETURN TAPE TO LOAD POINT.
                                         USE NORMAL I/O CHANNEL
        001000                 ORG       1000B
001000  07500                  EXC       1163B              REWIND TAPE 3 TO LOAD POINT
001001  01163
001002  07500                  EXC       2103B              SET HIGH DENSITY.
001003  02103
001004  07500                  EXC       1171B              SET ODD PARITY
001005  01171
001006  15712                  SICG      $AREA              SET INDIRECT BANK CONTROL
001007  07500                  EXC       2113B              WRITE (START TAPE MOTION)
001010  02113
001011  07303                  OUT       3                  OUTPUT 100(8) WORDS
001012  03622                            AREA+100           LWA+1
001013  06102                  NZF       2                  A=AREA+100 AND IS NOT ZERO
001014  03456                            AREA               FWA
001015  07500                  EXC       1163B              REWIND LOAD COMMAND
001016  01163
001017  07700                  HLT
        123456        AREA     EQU       123456B            DEFINE OUTPUT AREA
                               END
```

Figure 3-8. Magnetic Tape Program (Normal I/O) - Printout

```
                                         PROBLEM - SAME AS FIGURE 3-8, USE INTERNAL BUFFER
       001000           ORG     1000B              ESTABLISH START OF PROGRAM
001000  07500           EXC     1163B              REWIND TAPE 3 TO LOAD POINT
001001  01163
001002  07500           EXC     2103B              SELECT HIGH DENSITY FOR TAPE 3
001003  02103
001004  07500           EXC     1171B              SET ODD PARITY IN TAPE CONTROL UNIT
001005  01171
001006  02200           LDC     AREA               GET STARTING ADDRESS
001007  03456
001010  00105           ATE     *                  PLACE IN BUFFER ENTRY REGISTER
001011  01010
001012  03200           ADC     100                FORM LWA+1 FOR 100 WORDS
001013  00144
001014  00106           ATX     *                  PLACE IN BUFFER EXIT REGISTER
001015  01014
001016  10011           BBC1    $AREA              SET BUFFER BANK CONTROL
001017  00012
                                                   TO BANK OF OUTPUT AREA
001020  07500           EXC     2113B              SELECT WRITE (START TAPE MOTION)
001021  02113
001022  07300           IBO     *                  INITIATE BUFFER OUTPUT
001023  01022
001024  07500           EXC     1163B              REWIND TAPE 3 TO LOAD POINT
001025  01163
001026  07700           HLT
       123456   AREA    EQU     123456B            DEFINE OUTPUT AREA
                        END
```

Figure 3-9. Magnetic Tape Program (Internal Buffer) - Printout

```
                                    PROBLEM - SAME AS FIGURE 3-8, USE EXTERNAL BUFFER 2
         001000        ORG     1000B       ESTABLISH START OF PROGRAM
001000   10152         EXC2    1163B       REWIND TAPE 3 TO LOAD POINT
001001   01163
001002   10152         EXC2    2103B       SELECT HIGH DENSITY TAPE 3
001003   02103
001004   10152         EXC2    1171B       SET ODD PARITY IN TAPE CONTROL UNIT
001005   01171
001006   02200         LDC     AREA        GET STARTING ADDRESS
001007   03456
001010   10022         ATE2    *           PLACE IN BUFFER 2 ENTRY REGISTER
001011   01010
001012   03200         ADC     100         FORM LWA+1 FOR 100 WORD RECORD
001013   00144
001014   10032         ATX2    *           PLACE IN BUFFER 2 EXIT REGISTER
001015   01014
001016   10012         BBC2    $AREA       SET BUFFER 2 BANK CONTROL
001017   00012
001020   10152         EXC2    2113B       SELECT WRITE (START TAPE MOTION)
001021   02113
001022   10072         IBO2    *           INITIATE BUFFER 2 OUTPUT
001023   01022
001024   10152         EXC2    1163B       REWIND TAPE 3 TO LOAD POINT
001025   01163
001026   07700         HLT
         123456  AREA  EQU     123456B     DEFINE OUTPUT AREA
                       END
```

Figure 3-10. Magnetic Tape Program (External Buffer) - Printout

INTERRUPT

Certain internal and external conditions arise which make it necessary for the main program to be informed of their presence. The interrupt feature of the 160G provides this notification. An interrupt provides for the transfer of program control to some fixed location in memory without losing the information needed to return to the main program. Obviously, the main program must provide for the occurrence and processing of these interrupts.

The following conditions may cause an interrupt if allowed for:

1. Completion of buffer transmission of information.
2. Manual interrupt by the operator from the Console.
3. Occurrence of a parity error in reading an instruction or operand.
4. Interrupt from some other computer in the 160G system.
5. Occurrence of some condition in the peripheral equipment controlled by the 160G Compute Module.

INTERRUPT LOCATIONS AND ACTION

The interrupts are numbered: 10, 20, 30, 40, 100, 110, 114, 120, 130, 134, 140, 150, 154, 160, 170, 174, 200, 210, 214, 220, 230, 234, and 240. Each number is used to refer to the interrupt line and also to the interrupt location. The action of an interrupt depends on the mode in which the computer is operating. In the G mode (13-bit arithmetic), the occurrence of an interrupt on a given line (for example, 100) causes the following actions.

1. The interrupt is recognized by the interrupt circuit, if the interrupt lockout is not in effect.
2. The interrupt waits until the current instruction is completed and the computer goes into the read next instruction sequence.
3. At this point, the interrupt circuitry stores the relative bank control setting as the lower six bits of the interrupt location (100 for this example) and leaves the upper seven bits of the interrupt location unchanged. The interrupt circuitry then stores the address of the next instruction in the main program at the interrupt location +1 (at 101 in this example). Control is then transferred to the interrupt location +2 (102 in this example).

4. As a result of the interrupt, the relative bank control is set to bank 0, but the bank of the main program is stored as the lower six bits of the interrupt location.

It is the responsibility of the interrupt processing routine to save the contents of the A and Q registers and the bank controls if they are going to be changed by the interrupt processing. Normally the interrupt location has a set relative back control and jump instruction (SRJP) stored in it. On completion of the interrupt processing, the routine restores all controls except the relative bank control, and the A and Q registers. It then performs a jump to the interrupt location. This action returns control to the main routine at the point at which it was interrupted with no noticeable effect on the main routine except a loss of time in execution.

The interrupt action is different when the 160G is operating in the A mode (160-A compatibility). In this case, the address of the next instruction in the main program is stored at the interrupt location in bank zero and control is transferred to the interrupt processing routine in the relative bank (wherever the main program is operating) at the interrupt location +1. This mode is provided only for 160-A compatibility and normally is not used except when running 160-A programs on the 160G.

INTERRUPT LOCKOUTS

Interrupt lockouts are provided to allow the programmer to protect time dependent portions of his program. When an interrupt lockout is set, the interrupt circuits do not honor or process any interrupt until the lockout is removed. The time dependent areas of a program which normally occur are:

1. The time between issuance of an external function instruction and the actual initiation of the input or output command.
2. The time required for processing an interrupt.

To allow for these times, and other time sensitive areas of a program, the interrupt lockout is automatically placed in force on the issuance of an external function instruction (EXC, EXF, EXC2, EXC3, EXC4, EXC5, EXC6 or EXC7) or on the occurrence of an interrupt. The programmer may also place an interrupt lockout in effect by issuing a SILY command where Y is 0 to 7.

There are two levels of interrupt lockout provided in the 160G. The main level is the master interrupt lockout which prevents any other interrupts from being processed. This lockout may be imposed by the programmer giving a SIL0 instruction, and it is automatically imposed on the execution of an external function instruction (EXC, EXF, EXC2, EXC3, EXC4, EXC5, EXC6, EXC7). The master lockout is cleared, allowing interrupts to be processed, by a CIL or CIL0 instruction.

Lower level interrupt lockouts apply individually to each buffer data channel in the 160G system. Setting of a lower level lockout prevents further interrupts from being processed on that data channel, but interrupts may be processed on other data channels. For instance, if a lockout is placed on buffer channel 2, interrupts 100, 110, and 114 are not processed, but all other interrupts may be processed. The lower level interrupt lockouts which are set and cleared individually are not affected by the setting of the master interrupt lockout. Thus, it is possible to have an interrupt lockout applied to channel 2, then a master lockout may be set and cleared, but the lower level (channel 2 in this example) lockout is not changed. When it is desired to release the lockout, the programmer must give a CIL2 instruction.

A lower level interrupt lockout is imposed by the programmer by a SIL1, SIL2, SIL3, SIL4, SIL5, SIL6, or SIL7 instruction to impose the lockout on the buffer data channel. The lower level interrupt lockout is also imposed by the occurrence of an interrupt on a data channel. The lower level lockout on a data channel is cleared, allowing interrupts to be processed on that channel (provided master lockout is not in effect) by giving a CIL1 instruction for internal buffer channel, and CIL2, CIL3, CIL4, CIL5, CIL6, or CIL7 for the corresponding external buffer channel.

COMPUTE MODULE INTERRUPTS

The 160G compute module has five interrupt lines: 10, 20, 30, 40, and 240. The cause and action of the interrupts are given in the following paragraphs.

INTERRUPT 10

Interrupt 10 occurs as a manual interrupt which is activated from the 160G Console by momentarily depressing the MANUAL INTERRUPT pushbutton switch, or it may be caused by a parity error in reading an instruction or an operand. The program may determine which is the cause by examining the contents of the error register.

INTERRUPT 20

Interrupt 20 occurs each time the Compute Module completes an internal buffer operation. This interrupt is set and stored in a register flip flop. It can be cleared only by servicing the interrupt or by a master clear.

INTERRUPTS 30 AND 40

The buffer channel in the Compute Module has two external interrupt lines associated with it. These lines are called external 1 and external 2 corresponding to interrupt 30 and 40. Whenever a signal is placed on one of these lines, an interrupt 30 or 40 (depending on the line) occurs. This interrupt is removed by the computer program informing the external equipment to remove the interrupt.

INTERRUPT 240

Circuitry is provided for any Compute Module to interrupt all other Compute Modules in a common system. Interrupt 240 is designed for this purpose.

The computer-to-computer interrupt (CTCI) instruction causes an interrupt signal to be placed on the common computer-to-computer interrupt line (interrupt 240). This signal causes all 160G computers in the system to be interrupted to the location specified for the computer-to-computer interrupt. The computer issuing the interrupt is also interrupted. Prior to issuing the CTCI instruction, the computer issuing the interrupt places a word in common storage specifying which computer is doing the interrupt and which computer is to react to the interruption.

CHAPTER 4

CONSOLE

Each 160G Compute Module has an associated Console. The Console contains the controls and indicators necessary for maintenance and operation of the 160G system. The Console consists of a display panel and a switch panel, Figure 4-1. The display panel contains three status indicator displays and four 13-bit register displays. The switch panel is a row of pushbuttons used to exercise manual control over the operation of the 160G.

DISPLAY PANEL

REGISTER DISPLAY

The four 13-bit register displays can display the contents of 12 of the 13 registers in the 160G. Three registers are assigned to each display. Only one of these three registers can be displayed at one time. The registers are displayed in octal by using arabic numerals. Each register display contains five octal digits. The displays are illuminated only when the computer is stopped; the display is blank when the computer is running. Associated with each octal digit, except the highest-order digit, are three momentary pushbuttons which represent the powers of two, from right to left, starting with zero. Each group of three pushbuttons represents an octal digit. To aid in distinguishing between octal digits, the pushbuttons for adjacent octal digits are alternately light blue and white. These 13 associated pushbuttons are used to enter information into the register. A dark blue button at the right of the 13 pushbuttons clears that particular register. The only registers which can be so entered or cleared are the P, A, Q, and F registers. Each of these registers is assigned to a different display.

Centered beneath each register display are three pushbutton switches which select the register to be displayed. Each pushbutton is for a different register. A pushbutton which is illuminated when depressed, stays activated until another pushbutton in the group of three is activated.

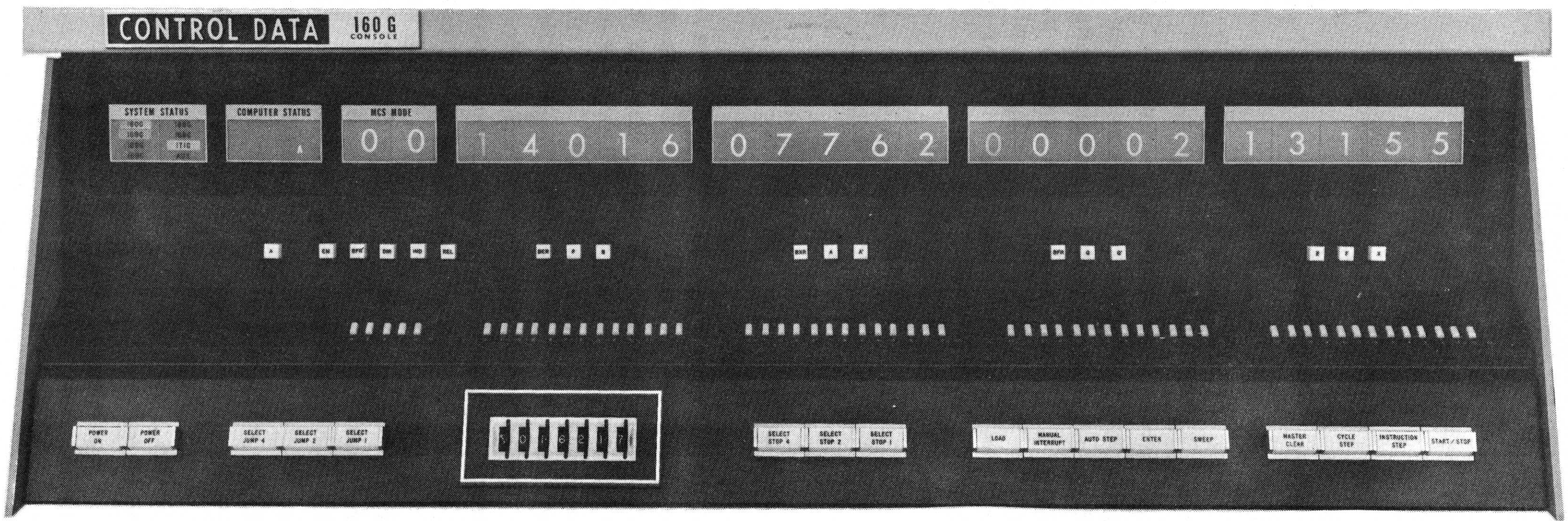


Figure 4-1. Console Displays and Indicators

I/O MODULE

Each data channel of an I/O Module has three interrupt lines associated with it. The interrupts that are associated with them would be numbered as given in Table 3-1.

The area between 00102 and 00110, 00122 and 00130, and so forth, is left open to provide space for storage of the A and Q registers and the bank controls.

Each external device may contain interrupt circuitry. The channel interrupt may be activated by any peripheral device designed to provide an interrupt signal. The actual meaning of these interrupts is a

function of the device causing the signal. If more than one device is connected to a line, each device must be interrogated following an interrupt to determine which generated the signal.

Once an interrupt signal is placed on a line it remains on the line until it is recognized or until a master clear is performed. Whenever any interrupt is recognized or whenever an external function is executed, all further interrupts are locked out until a clear interrupt lockout (CIL or CILY) instruction is performed. The CIL instruction clears master interrupt lockout. The CILY (Y = 1, 2, 3, 4, 5, 6,or 7) instruction clears interrupt lockout on only one channel.

TABLE 3-1. INTERRUPT NUMBER ASSIGNMENTS

Buffer Data Channels	Buffer Termination Interrupt	External Interrupt 1	External Interrupt 2
2(Channel Y)	100	110	114
3(Channel Y')	120	130	134
4(Channel Y)	140	150	154
5(Channel Y')	160	170	174
6(Channel Y)	200	210	214
7(Channel Y')	220	230	234

The pushbuttons are marked with alphabetic symbols for the various registers. These are as follows:

BER	The buffer entrance register indicates the address of the last word transferred on the buffer channel.
P	The program address register indicates the address of the current instruction.
S	The address selection register indicates the address of the word about to be transferred to or from storage.
BXR	The buffer exit register indicates the LWA + 1 set up for the last buffer operation.
A	This register indicates the current contents of the A register.
A'	This register indicates the result of the last operation in the adder.
BFR	This register indicates the last word processed during the last buffer operation.
Q	This register indicates the current contents of the Q register.
Q'	This register indicates the current contents of the Q' register.
F	This register indicates the current effective function code.
Z	This register indicates the current contents of the Z register.
X	This register indicates the current contents of the X register.

ERROR REGISTER

The 160G has a 10-bit error register which contains various error conditions. The contents of this register may be read into the A register at any time by issuing an ERTA instruction. The conditions, indicated by a 1 in a given bit position, are as given in Table 4-1.

TABLE 4-1. ERROR CONDITIONS

Bit position in A	Condition
12 and 11	Add or subtract overflow
10	Computer parity error
7	Parity error on buffer data channel 7
6	Parity error on buffer data channel 6
5	Parity error on buffer data channel 5
4	Parity error on buffer data channel 4
3	Parity error on buffer data channel 3
2	Parity error on buffer data channel 2
1	Parity error on internal buffer data channel
0	Divide overflow or divide by zero

Add and subtract overflows are sensed by an illegal sign change as a result of the add or subtract operation. The add, subtract, divide overflow, and computer parity error indicators are reset at the completion of an ERTA instruction. The buffer data channel parity error indicators by a clear buffer controls (CBC) instruction, or by a master clear. Bit positions 8, 9, and 10 are always set to 1 because they do not indicate any condition.

INDICATORS

MCS MODE INDICATOR

This display indicates which storage bank control has been set and the number to which it was set. The display is divided into upper and lower sections. The upper (numeric) section displays the numbers that have been assigned to the storage banks. It contains one of the numbers, 0 through 31, that are set by the lower section. The lower section contains five pushbuttons marked EM, BFR, DIR, IND, and REL. These pushbuttons, except for EM, are used to select the desired storage bank control, and are indicated as follows:

BFR - Buffer storage bank control
DIR - Direct storage bank control
IND - Indirect storage bank control
REL - Relative storage bank control

When one of the pushbuttons is activated, a light illuminates that pushbutton. This light is also illuminated and the bank assignment displayed when the corresponding bank control is selected by the Compute Module logic. A second lower row of six buttons is used to set the selected storage bank controls. Once a bank control is manually selected it remains selected until another bank control is manually selected or the START switch is depressed.

> NOTE: The rightmost pushbutton is a clear button which sets the bank control to zero; the other five buttons control the binary value of the bank number. All storage banks may be cleared by holding the clear pushbutton down and depressing the MASTER CLEAR pushbutton.

The relative bank control is selected when a master clear is performed. Normally, the storage bank for the previous memory reference is displayed in the MCS MODE indicator. However, any of the other bank controls may be manually displayed.

COMPUTER STATUS INDICATOR

The COMPUTER STATUS indicator displays various conditions existing within the Compute Module during the execution of a program or when the computer is stopped. The status is indicated by alphabetic symbols displayed in the COMPUTER STATUS indicator window or by the illuminated START/STOP pushbutton.

The START/STOP pushbutton switch is illuminated when the computer is running. If the computer is stopped, the light goes out and indicates one of the following conditions:

1. A HLT or ERR instruction is executed.

2. A SLS or SJS instruction is executed and computer stops.

3. The computer is manually taken out of run status by depressing the START/STOP switch.

The alphabetic symbols on the COMPUTER STATUS indicator indicate the following computer conditions.

ERR	Computer has executed an ERR or ERRG instruction and is stopped.
SEL	Displayed each time an EXF, EXCY, or EXC instruction is executed; remains until the selection of equipment is completed. A constant display of SEL with no apparent I/O action usually indicates the computer has attempted an illegal instruction.
OUT	Displayed during all normal output operations. A constant display of OUT with no apparent output action usually indicates that output was attempted without proper unit selection.
IN	Displayed during all normal input operations. A constant display of IN with no apparent input action usually indicates that input was attempted without proper unit selection. IN is also displayed when the computer is waiting for an external device to supply data.
IBA	Displayed during all buffer input operations.
OBA	Displayed during all buffer output operations.
BPE	Displayed when a parity error is recognized during a buffer operation.
CPE	Displayed when a parity error is recognized in the transfer of data from memory in the execution of a program.
INT	Displayed during interrupt sequences.
OFL	Displayed when an overflow condition is recognized.
A, B, C, D, or E	These letters indicate the next memory reference cycle to be executed in a cycle step operation.

SYSTEM STATUS INDICATOR

Each module in a system reports back to the Console concerning the status of power and temperature within an individual module. The SYSTEM STATUS indicator displays the condition of the modules by the use of colored lights behind the unit designators. Each unit designator is lighted by one of three different colors: red, amber, or green. An interlock bypass (BYP) toggle switch is located in each module to allow the module to be operated in an over-temperature condition. The colors and the conditions represented by them are as follows:

Normal (Green light on):

1. The -20 and +20 logic voltages are present.
2. Cabinet temperature is below 100° F.
3. Interlock bypass switch is not in bypass position.
4. Memory power supply switches are on (applies only to the 169G and 160G cabinets).

Warning (Green and Yellow lights on):

1. -20 and +20 logic voltages are present.
2. Cabinet temperature is above 100° F but below 110° F.
3. Interlock bypass switch is not in bypass position.
4. Memory power supply switches are on (applies only to the 169G and 160G cabinets).

Warning (Yellow light on):

1. -20 and +20 logic voltages are present.
2. Cabinet temperature is not determined.
3. Interlock bypass switch is in bypass position.
4. Memory power supply switches are on (applies only to the 169G and 160G cabinets).

Malfunction (Red light on):

One or more of the following troubles may exist:

1. Cabinet temperature is over 110° F.
2. The -20 volt or +20 volt logic voltage has failed.
3. A memory power supply switch is off (applies only to the 169G and 160G cabinets).

SWITCH PANEL

CONTROLS

The various switches and pushbuttons provide manual control over the operation of the computer. When activated, each pushbutton is illuminated thus showing it is activated. The names and purposes of the switches and pushbuttons are as follows.

START/STOP	This alternate-action pushbutton switch is used in starting and stopping the operation of the computer. It is lighted when the computer is running and extinguished when the computer is stopped.
INSTRUCTION STEP	This pushbutton selects instruction-by-instruction (one step at a time) mode of program execution.
CYCLE STEP	This pushbutton selects a cycle mode (one memory cycle at a time) of program execution.
MASTER CLEAR	The MASTER CLEAR pushbutton switch clears the registers and portions of control logic.
SWEEP	This pushbutton permits the displaying of the contents of memory locations.
ENTER	This pushbutton permits manual entry from Console to core memory.
EM (Entire Memory)	This pushbutton enables the same quantity to be entered throughout a memory bank.
MANUAL INTERRUPT	This pushbutton activates interrupt 10.

AUTO STEP	This pushbutton switch causes the 160G to automatically step through a program to observe the contents of the registers. It is sometimes called an oscillator-speed run. The START pushbutton is not used to activate the operation once the AUTO STEP pushbutton is depressed. Depressing the AUTO STEP pushbutton initiates the stepping operation. Prior to depressing the AUTO STEP pushbutton, the CYCLE STEP or INSTRUCTION STEP pushbutton must be activated.
LOAD	This pushbutton activates the initial loading routine.
SELECTIVE STOP	These three pushbutton switches condition the computer for selective stop instructions.
SELECTIVE JUMP	These three pushbutton switches condition the computer for selective jump instructions.
POWER ON	This pushbutton switch applies power to the 160G system.
POWER OFF	This pushbutton switch removes power from the 160G system.
A (Mode)	This pushbutton is used when programs are to be run in the A mode (160-A programs). It forces the 160-A instructions in the 160G computer to operate as in a 160-A computer.
Breakpoint	These seven 8-position thumb switches are used to select an instruction reference address at which to stop computation.. The highest-order switch activates or disables the breakpoint feature. The computer stops operation, on reference to an address specified in breakpoint, while the breakpoint is active.

OPERATION OF THE 160G SYSTEM

STARTING THE 160G

The recommended procedure for starting the 160G is as follows:

1. Be sure that the room temperature is within the prescribed limits.

2. Depress the POWER ON pushbutton on the Console. The power on master clear feature is released by depressing the MASTER CLEAR pushbutton before operating the 160G.

3. The contents of the selected storage bank control is displayed on the display panel.

4. The master clear should remove part of the information and depressing the START/STOP pushbutton once in the INSTRUCTION STEP mode should remove the rest of any information that might be displayed in the COMPUTER STATUS indicator. If repeated, master clears do not remove these symbols. Turn machine off by depressing the POWER OFF pushbutton, and call maintenance.

5. A green light for each unit the system displayed in the SYSTEM STATUS indicator indicates that the system is normal and ready to operate. If an amber or a red light is displayed, that unit should be checked for a malfunction. Turn machine off and call maintenance.

CLEARING OR ENTERING AN ENTIRE STORAGE BANK

A bank may be cleared by the following steps:

1. Master clear. Depress the ENTER pushbutton.

2. Set the relative storage bank control to select the bank to be cleared.

3. Hold down the EM pushbutton while performing step 4.

4. Depress and hold down the START pushbutton for about a second. This step effectively enters all zeros into the memory locations.

In much the same manner, the same quantity can be entered throughout a bank. The procedure would be as follows:

1. Master clear. Depress ENTER pushbutton.

2. Manually enter the quantity into the A register.

3. Hold down the EM pushbutton and depress START pushbutton. The EM pushbutton should be held down about a second.

STORING A PROGRAM

AUTO LOAD OPERATION

The 160G has an automatic loading feature which allows the loading of initial programs from any peripheral device located on the normal or buffer channels.

By depressing the MASTER CLEAR, LOAD, and START/STOP pushbuttons, the following sequence of instructions is stored in the relative memory bank, starting at location zero.

Location	FFFEE	G	Comments
(r)00000	07500		External function constant instruction
(r)00001		0YYYY	External function code set on switches
(r)00002	07500		External function constant instruction
(r)00003		0YYYY	Second external function code set on switches
(r)00004	07201		Input instruction

Location	FFFEE	G	Comments
(r)00005	00000		Starting and terminating address (zero), unless a disconnect is received
(r)00006			Next instruction on completion of input as indicated by input disconnect from input equipment

The two function codes, 0YYYY, are set by switches at the back of the Compute Module. Normally the first code is used to specify rewind load of a tape unit, and the second code is used to specify a binary input, two characters per word. The loading of the program then takes place when the START/STOP pushbutton is depressed. When the first record of the input device is read, control of the program continues at location 00006 in the relative bank. It is the function of the first record to then read in the rest of the program from the input device. To operate correctly, the relative and indirect bank controls must be set to the same value.

MANUALLY STORING PROGRAM

A program can be manually stored in the computer by the following steps:

1. Depress MASTER CLEAR and ENTER pushbuttons.
2. Set relative storage bank control to select the bank into which data is to be entered.
3. Depress START pushbutton once.
4. Set P to the first location into which data is to be entered.
5. Enter one word of data into the A register.

6. Depress the START pushbutton once. The computer stops with the A register clear, the data word in storage and X, and P advanced by 1. The A register is cleared each time P is advanced.

7. If data is to be entered into consecutive locations, repeat steps 5 and 6. If data is to be entered into non-consecutive locations, clear P and repeat steps 4 through 6.

EXAMINING CONTENTS OF STORAGE

The contents of any storage location or the entire bank can be examined at the Console. The steps are as follows:

1. Master clear. Depress the SWEEP and STEP pushbuttons.

2. Set relative bank control to select the bank to be examined.

3. Set P to the location to be examined.

4. Depress START pushbutton. The contents of the locations specified by P appears in X. P automatically advances by 1.

5. To examine consecutive locations, repeat step 4. To examine non-consecutive locations, master clear and repeat steps 3 and 4.

CHAPTER 5

MASS and GASS

MASS, the 160G system for monitor and assembly, is an integrated series of programs and operating procedures which provide for easy use and integration of previously written library and system programs into a programmer's own operations.

The monitor portion of the system permits the sequencing of jobs on the 160G system and also provides I/O control operations as well as the linkage function to tie programs together. The assembly portion of the system (GASS) provides the means for preparing the programs with a minimum of difficulty for the programmer.

The standard I/O routines provide the programmer with a set of I/O routines which take advantage of the buffering capabilities of the 160G. The choice of I/O devices is indicated by a logical unit number assigned to each device. The monitor master I/O routine decodes the standard unit number and transfers control to the required specialized I/O routine for a particular device. The assignment of unit numbers to I/O devices is controlled by a decoding table which may be changed under operator or program control to allow for substitution of equipment as required. The design of the standard I/O routines is such that multiprogramming or parallel processing may take place easily.

The linking relocatable loader provides the facility for locating programs anywhere within the memory of the 160G and also to provide facility for referencing data and transferring control to other programs which are assembled separately.

The operator (executive) control portion of MASS accepts "commands" from a standard source (normally a typewriter) and under this control loads and execute programs and subroutines as required.

GASS, generalized assembly system, operates within MASS. An assembly program has two major functions. The first is to allow the programmer to use meaningful names for instructions. The assembly program thus translates the mnemonic operation codes to the actual machine codes required. The second function is to allow the programmer to assign symbols to locations which are referenced in memory. The assembly program takes care of the assignment of symbols to locations and the resultant translation by the construction of a symbol table which gives the absolute location and the symbol to which it has been assigned. Figures 5-1 and 5-2 show how a program prepared in the normal format would appear in GASS format.

The MASS system is designed to operate on a minimum configuration consisting of a 160G Compute Module, with 8192 words of memory, four magnetic tape drives, and an on-line typewriter. The system efficiency is increased by the inclusion of additional peripheral equipment and Memory Modules.

The MASS system is carried on a magnetic tape which is called the system library tape. The first record on the library tape is the loading routine which is used to bring the MASS system into the computer. A complete description of MASS and GASS may be found in their respective manuals.

```
                                SUBROUTINE TO UNPACK BCD DATA


000100   15711                  SET (I) TO BANK OF PACKED DATA
000101   00040                  SET (D) TO INDEX REG
000102   00400
000103   04005                  SET INDEX REG TO
000104   04006                  ZERO
000105   16105                  PICK PACKED WORD TO Q
000106   10200
000107   00400                  CLEAR A
000110   13707                  UPPER CHAR TO A
000111   12706                  LOWER CHAR TO Q
000112   14106                  STORE UPPER
000113   10300
000114   16506                  STORE LOWER
000115   10301
000116   00402
000117   05006                  UPDATE UNPACK INDEX
000120   05405                  UPDATE PACK INDEX
000121   00710                  CHECK DONE
000122   06515                  NO, DO AGAIN
000123   07700                  HALT
```

Figure 5-1. Numeric Coded Program

```
          IDENT     UNPACK
                    SUBROUTINE TO UNPACK BCD DATA
          CON       5
INDEX1
INDEX2
          ORG       100B
PACK      EQU       110200B
UNPACK    EQU       110300B
          SICG      $PACK                 SET (I) TO BANK OF PACKED DATA
          SDC       $INDEX1               SET (D) TO INDEX REG
          LDN       0
          STD       INDEX1                SET INDEX REG TO
          STD       INDEX2                ZERO
CONT      LQMX,INDEX1 PACK                PICK PACKED WORD TO Q

          LDN       0                     CLEAR A
          LLS,7                           UPPER CHAR TO A
          QLS,6                           LOWER CHAR TO Q
          STMX,INDEX2 UNPACK              STORE UPPER

          SQMX,INDEX2 UNPACK+1            STORE LOWER

          LDN       2
          RAD       INDEX2                UPDATE UNPACK INDEX
          AOD       INDEX1                UPDATE PACK INDEX
          SBN       10B                   CHECK DONE
          NZB       CONT                  NO, DO AGAIN
          HLT                             HALT
          END
```

Figure 5-2. GASS Coded Program

APPENDIX A

GLOSSARY

The following glossary gives the meaning of terms that are used in a relatively specialized sense in this manual.

ADDER	In general, a device used to add two quantities; specifically, the borrow pyramid in the 160G.
ADDRESS	The number designating a storage location; also the storage location itself.
ASSEMBLER (GASS)	A routine which automatically produces a specific program for a particular problem. The routine determines the meaning for information expressed in mnemonic code, selects or generates the required subroutine, transforms the subroutine into specific (numeric) coding, assigns storage registers, and enters the information as an element of the problem program.
BANK	A unit of core storage with provisions for storing 4096 or 8192 words. The use of more than one bank permits increasing the storage capacity of a computer without increasing the word length implicitly necessary to extend the range of storage addresses. In the 160G, this address is effectively increased by a separate instruction (set storage bank control) which determines which bank(s) is used during a program.
BIT	Binary digit; may be either one or zero.

BORROW	In a subtractive counter or accumulator, a signal indicating that in stage n, a 1 was subtracted from a 0.
BREAKPOINT	A point in a routine at which the computer may be stopped by a manual switch for a visual check of progress.
BUFFER	Noun: A device in which data is stored temporarily in the course of transmission from one point to another. Verb: To store data temporarily.
BUFFERED INPUT/OUTPUT	A term indicating that the computer may carry on high-speed computation at the same time it is exchanging data with a peripheral device. In the 160G, this term must be distinguished from normal I/O during which the computer cannot engage in computation.
CARRY	In an additive counter or accumulator, a signal indicating that in stage n a 1 was added to a 1.
CHANNEL	A transmission path that connects the computer to a given external equipment.
CHARACTER	Information handled by the computer: 1) A group of six bits representing bioctal information; may denote a binary quantity, a digit, letter or symbol. 2) A group of seven bits representing an item of information. When the capacity of an input device is six or seven bits, those bits are deposited in the lower portion of the selected storage address, the remaining bits are zeros.
CLEAR	A command that removes a quantity from a register by placing every stage in the 0 state.

COMMAND	A signal that performs a unit operation, such as transmitting contents of one register to another, shifting a register, and setting a FF.
COMPLEMENT	Noun: See one's complement or two's complement. Verb: A command which produces the one's complement of a given quantity.
CONTENTS	The quantity of word held in a register or storage location.
CORE	A small ferromagnetic toroid used as the bistable device for storing a bit in a memory plane.
COUNTER	A register with provisions for increasing or decreasing its contents by 1 on receiving the appropriate instruction.
DIRECT ADDRESSING	A mode of addressing that does not modify any portion of the instruction word. The function code and execution address are interpreted without modification. It is also an address mode that limits direct addressing to the first 64 storage locations ($2^6 = 64$) in the direct bank.
END-AROUND BORROW	A borrow that is generated in the highest-order stage of an accumulator or counter and sent directly to the lowest-order stage.
ENTER	To manually place in a register a quantity that is not from storage. In the 160G, quantities may be entered in only the A, Q, P, and F registers.
EXECUTION ADDRESS	The lower six bits of a 13-bit instruction. Most often used to specify the storage address of an instruction or operand. Sometimes used as the operand.

EXTERNAL FUNCTION CODE — Is used to select and specify that a external equipment is to perform some specific function.

FUNCTION CODE — A 7-bit code which designates the instruction. (See operation code.)

INDIRECT ADDRESSING — A mode of addressing which extends the length of the execution address (E) to a full computer word, thereby permitting operand references to be made upon any location in storage. All indirect addresses must be contained in storage locations which are available by means of direct addressing. In the 160G, the memory mode is a special form of indirect addressing.

INPUT DISCONNECT — During an input instruction, a signal sent to the computer by the external device to indicate that the device has completed all available transmissions to the computer.

INPUT REQUEST — A request, by the computer, for information from an external device. Occurs during instruction only. (See resume.)

INSTRUCTION — A 13- or 26-bit quantity consisting of a function (or operation) code and an execution address.

INTERRUPT — A signal (or class thereof) which, when received and recognized by the computer, forces the computer to forestall its current operation and jump to a subroutine, the starting address of which is determined by the class of the interrupt. A subroutine may have any number of options. It may merely stop the computer, determine the nature of the interrupt in order to take corrective measure, or return the computer to another phase of the main program.

LOAD — To place a quantity from storage in the A or Q register.

LOCKOUT — Any function (usually of machine logic) that inhibits an action which would normally occur if the lockout is not imposed.

LOGICAL PRODUCT — In Boolean algebra, the AND function of several terms. The product is 1 only when all the terms are 1; otherwise it is 0. Sometimes referred to as the result of bit-by-bit multiplication.

MASK — In the formation of the logical products of two quantities, one of them may be used as a mask for the other. The mask determines what part of the other quantity is to be considered. Wherever the mask is 0, that part of the other quantity is cleared, but wherever the mask is a 1, the other quantity is left unaltered.

MASTER CLEAR (MC) — A general command which clears all of the crucial registers and control FF's to prepare for a new mode of operation. It does not affect core storage.

MNEMONIC CODE — A two, three, or four letter code which represents the operation code or the function of an instruction.

MODULUS — An integer which describes certain arithmetic characteristics of registers, especially counters and accumulators within a digital computer. The modulus of a device is defined by r^n for an open ended device and r^n-1 for a closed (end-around) device, where r is the base of the number system used and n is the number of digit positions (stages) in the device. Generally, devices with modulus r^n use two's complement arithmetic procedures, and devices with modulus $r^n - 1$ use one's complement procedures.

NORMAL JUMP — An instruction that jumps from one sequence of instructions to a second and makes no preparation for returning to the first sequence.

NUMERIC CODE — A system of abbreviation in which all information is reduced to numerical quantities. The 160G computer operates on binary numeric coded programs.

ONE'S COMPLEMENT — With reference to a binary number, that number which results from subtracting each bit of the given number from the bit 1. A negative number is expressed by the one's complement of the corresponding positive number.

OPERAND — Usually refers to the quantity specified by the execution address. This quantity is operated upon in the execution of the instruction.

OPERATION CODE — The upper seven bits of a 13-bit instruction which identify the instruction. After the code is translated, it conditions the computer for execution of the specified instruction. This code, which is expressed by three octal digits, is designated by the letter F. The operation code is designated by seven bits (numeric code) or by two, three ,or four letters (mnemonic code) which specifies the function code and the address mode.

OVERFLOW — The condition in which the capacity of a register is exceeded.

PARITY CHECK — A summation check in which the binary digits in a character are added and the sum checked against a previously computed parity digit. In the 160G, it is a check which tests whether the number of ones is odd.

PARTIAL ADD	An addition without carries. Accomplished by complementing each bit of the augend where the corresponding bit of the addend is a 1.
PROGRAM	A precise sequence of instructions that accomplisheds a computer routine; a plan for the solution of a problem.
PYRAMID	A network of inverters that senses borrow conditions and produces borrow signals.
READ	To place a quantity from a storage location into a register. The quantity in storage remains unchanged.
READY	The I/O control signal sent by either the computer or an external equipment to altert the device that is to receive a transmission. The ready signal indicates that the word or character has been transmitted.
RELATIVE ADDRESSING	A mode of addressing wherein the address of the operand is determined by adding (or subtracting) the contents of the execution address portion (E) of the instruction word to (or from) the instruction address.
REPLACE	In the title of an instruction, the result of the execution of the instruction is stored in the location from which the initial operand was obtained.

RESUME	The output control signal sent by an external equipment to indicate that it is prepared to receive another word or character. The resume signal is thus a request for data. (See input request.)
RETURN JUMP	A jump instruction which prepares for continuing the first sequence after the second is completed.
ROUTINE	The sequence of operations which the computers perform under the direction of a program.
SELECTIVE COMPLEMENT	In Boolean algebra, the exclusive OR function of several terms. The sum is 1 when any but not all of the terms are 1; it is 0 only when all are 0 or 1.
SHIFT	To move the bits of quantity right or left.
SIGN BIT	The bit in the highest-order stage of the register (in registers where a quantity is treated as signed by use of one's complement notation). If the bit is 1, the quantity is negative; if the bit is 0, the quantity is positive.
SIGN EXTENSION	The duplication of the sign bit in the highest-order stages of a register.
STATUS	1) The condition of an external device, as reflected in the response given a status request interrogation by the computer. 2) The condition of the computer as shown by the COMPUTER STATUS indicator on the Console. May at various times indicate what it is presently doing, why it stopped, or what it will do when it starts next.
TRANSMISSION, FORCED	A transmission where both set and clear inputs, only one of which is 1, are

simultaneously gated into a FF which has not been cleared previously.

TRANSLATION	An indication of the content of a group of bit registers. A complete translation gives the exact content, while a partial translation indicates only that the content is within certain limits.
TWO'S COMPLEMENT	That number which results from subtracting each bit of a number from 0. The two's complement may be formed by complementing each bit of the given number and then adding one to the result and by performing the required carries.
WORD	A unit of information which has been coded for use in the computer as a series of bits.
WRITE	To enter a quantity into a storage location.

APPENDIX B

NUMERIC INSTRUCTION CODES

NOTE: EB-Entire Bank; EM-Entire Memory; MX-Memory Index

FFFEE	G	MNEMONIC	OPERATION
00000		ERR	Error Stop
0000X		NOP	No Operation
0001X		SRJ	Set Rel Bank Control; Jump
0002X		SIC	Set Ind Bank Control
0003X		IRJ	Set Ind and Rel Bank Controls; Jump
0004X		SDC	Set Dir Bank Control
0005X		DRJ	Set Dir and Rel Bank Controls; Jump
0006X		SID	Set Ind and Dir Bank Controls; Jump
0007X		ACJ	Set Dir, Ind, and Rel Bank Controls; Jump
00100	X	BLS	Block Store
00101		PTA	P to A
00102		LS1	Left Shift One
00103		LS2	Left Shift Two
00104		CBC	Clear Buffer Controls
00105	X	ATE	A to BER
00106	X	ATX	A to BXR
00107		ETA	BER to A
00110		LS3	Left Shift Three
00111		LS6	Left Shift Six
00112		MUT	Multiply A by 10_{10}
00113		MUH	Multiply A by 100_{10}
00114		RS1	Right Shift One
00115		RS2	Right Shift Two
00120		CIL	Clear Interrupt Lockout
00130		CTA	Bank Controls to A
0014X		SBU	Set Buffer Bank Control
0015X		STP	Store P at Location 5X
0016X		STE	Store BER at 6X, A to BER
002YY		LPN	Logical Product No Address
003YY		SCN	Selective Complement No Address

FFFEE	G	MNEOMONIC	OPERATION
004YY		LDN	Load No Address
005YY		LCN	Load Complement No Address
006YY		ADN	Add No Address
007YY		SBN	Subtract No Address
010XX		LPD	Logical Product Direct
01100	X	LPM	Logical Product Memory
011XX		LPI	Logical Product Indirect
01200	X	LPC	Logical Product Constant
012XX		LPF	Logical Product Forward
01300		LPS	Logical Product Specific
013XX		LPB	Logical Product Backward
		LPR	Logical Product Relative, 012XX or 013XX
014XX		SCD	Selective Complement Direct
01500	X	SCM	Selective Complement Memory
015XX		SCI	Selective Complement Indirect
01600	Y	SCC	Selective Complement Constant
016XX		SCF	Selective Complement Forward
01700		SCS	Selective Complement Specific
017XX		SCB	Selective Complement Backward
		SCR	Selective Comp. Relative, 016XX or 017XX
020XX		LDD	Load Direct
02100	X	LDM	Load Memory
021XX		LDI	Load Indirect
02200	Y	LDC	Load Constant
022XX		LDF	Load Forward
02300		LDS	Load Specific
023XX		LDB	Load Backward
		LDR	Load Relative, 022XX or 023XX
024XX		LCD	Load Complement Direct
02500	X	LCM	Load Complement Memory
025XX		LCI	Load Complement Indirect
02600	Y	LCC	Load Complement Constant
026XX		LCF	Load Complement Forward
02700		LCS	Load Complement Specific
027XX		LCB	Load Complement Backward
		LCR	Load Comp. Relative, 026XX or 027XX

FFFEE	G	MNEMONIC	OPERATION
030XX		ADD	Add Direct
03100	X	ADM	Add Memory
031XX		ADI	Add Indirect
03200	Y	ADC	Add Constant
032XX		ADF	Add Forward
03300		ADS	Add Specific
033XX		ADB	Add Backward
		ADR	Add Relative, 032XX or 033XX
034XX		SBD	Subtract Direct
03500	X	SBM	Subtract Memory
035XX		SBI	Subtract Indirect
03600	Y	SBC	Subtract Constant
036XX		SBF	Subtract Forward
03700		SBS	Subtract Specific
037XX		SBB	Subtract Backward
		SBR	Subtract Relative, 036XX or 037XX
040XX		STD	Store Direct
04100	X	STM	Store Memory
041XX		STI	Store Indirect
04200	Y	STC	Store Constant
042XX		STF	Store Forward
04300		STS	Store Specific
043XX		STB	Store Backward
		STR	Store Relative, 042XX or 043XX
044XX		SRD	Shift Replace Direct
04500	X	SRM	Shift Replace Memory
045XX		SRI	Shift Replace Indirect
04600	Y	SRC	Shift Replace Constant
046XX		SRF	Shift Replace Forward
04700		SRS	Shift Replace Specific
047XX		SRB	Shift Replace Backward
		SRR	Shift Replace Relative, 046XX or 047XX
050XX		RAD	Replace Add Direct
05100	X	RAM	Replace Add Memory
051XX		RAI	Replace Add Indirect
05200	Y	RAC	Replace Add Constant
052XX		RAF	Replace Add Forward

FFFEE	G	MNEMONIC	OPERATION
05300		RAS	Replace Add Specific
053XX		RAB	Replace Add Forward
		RAR	Replace Add Relative, 052XX or 053XX
054XX		AOD	Replace Add One Direct
05500	X	AOM	Replace Add One Memory
055XX		AOI	Replace Add One Indirect
05600	Y	AOC	Replace Add One Constant
056XX		AOF	Replace Add One Forward
05700		AOS	Replace Add One Specific
057XX		AOB	Replace Add One Backward
		AOR	Replace Add One Relative, 056XX or 057XX
060XX		ZJF	Zero Jump Forward
061XX		NZF	Non-Zero Jump Forward
062XX		PJF	Positive Jump Forward
063XX		NJF	Negative Jump Forward
064XX		ZJB	Zero Jump Backward
065XX		NZB	Non-Zero Jump Backward
066XX		PJB	Positive Jump Backward
067XX		NJB	Negative Jump Backward
		ZJR	Zero Jump Relative, 060XX or 064XX
		NZR	Non-Zero Jump Relative, 061XX or 065XX
		PJR	Positive Jump Relative, 062XX or 066XX
		NJR	Negative Jump Relative, 063XX or 067XX
070XX		JPI	Jump Indirect
07100	X	JPR	Return Jump
071XX		JFI	Jump Forward Indirect
07200	X	IBI	Initiate Buffer Input
072XX		INP	Normal Input
07300	X	IBO	Initiate Buffer Output
073XX		OUT	Normal Output
074YY		OTN	Output No Address
07500	Y	EXC	External Function Constant
075XX		EXF	External Function Forward

FFFEE	G	MNEMONIC	OPERATION
07600		INA	Input to A
076XX		HWI	Half Write Indirect
07677		OTA	Output from A
07700		HLT	Halt
0770Y		SLS	Selective Stop
077Y0	X	SLJ	Selective Jump
077YY	X	SJS	Selective Stop; Jump
07777		HLT	Halt
10000		ERRG	Error Stop
1000X		NOPG	No Operation
10011	Y	BBC1	Set Internal BBC
10012	Y	BBC2	Set External 2 BBC
10013	Y	BBC3	Set External 3 BBC
10014	Y	BBC4	Set External 4 BBC
10015	Y	BBC5	Set External 5 BBC
10016	Y	BBC6	Set External 6 BBC
10017	Y	BBC7	Set External 7 BBC
10022	X	ATE2	A to BER, Channel 2
10023	X	ATE3	A to BER, Channel 3
10024	X	ATE4	A to BER, Channel 4
10025	X	ATE5	A to BER, Channel 5
10026	X	ATE6	A to BER, Channel 6
10027	X	ATE7	A to BER, Channel 7
10032	X	ATX2	A to BXR, Channel 2
10033	X	ATX3	A to BXR, Channel 3
10034	X	ATX4	A to BXR, Channel 4
10035	X	ATX5	A to BXR, Channel 5
10036	X	ATX6	A to BXR, Channel 6
10037	X	ATX7	A to BXR, Channel 7
10042		ETA2	BER, Channel 2, to A
10043		ETA3	BER, Channel 3, to A
10044		ETA4	BER, Channel 4, to A
10045		ETA5	BER, Channel 5, to A
10046		ETA6	BER, Channel 6, to A
10047		ETA7	BER, Channel 7, to A
10052		INA2	Input to A, Channel 2
10053		INA3	Input to A, Channel 3
10054		INA4	Input to A, Channel 4
10055		INA5	Input to A, Channel 5

FFFEE	G	MNEMONIC	OPERATION
10056		INA6	Input to A, Channel 6
10057		INA7	Input to A, Channel 7
10062	X	IBI2	Initiate Buffer Input, Channel 2
10063	X	IBI3	Initiate Buffer Input, Channel 3
10064	X	IBI4	Initiate Buffer Input, Channel 4
10065	X	IBI5	Initiate Buffer Input, Channel 5
10066	X	IBI6	Initiate Buffer Input, Channel 6
10067	X	IBI7	Initiate Buffer Input, Channel 7
10072	X	IBO2	Initiate Buffer Output, Channel 2
10073	X	IBO3	Initiate Buffer Output, Channel 3
10074	X	IBO4	Initiate Buffer Output, Channel 4
10075	X	IBO5	Initiate Buffer Output, Channel 5
10076	X	IBO6	Initiate Buffer Output, Channel 6
10077	X	IBO7	Initiate Buffer Output, Channel 7
10100		AMOD	Select A Mode
10101		GMOD	Select G Mode
10102		CTCI	Computer-to-Computer Interrupt
10103	X	RCJP	AQ to Bank Controls; Jump
10104		XAQ	Interchange A and Q
10105		ERTA	Error Register to A
10106	X	JPIB	Jump Indirect - EB
10107	X	JRIB	Jump Relative - EB
10110	X	ZJRB	Zero Jump Relative - EB
10111	X	NZRB	Non-Zero Jump Relative - EB
10112	X	PJRB	Positive Jump Relative -EB
10113	X	NJRB	Negative Jump Relative - EB
10114	X	UJRB	Unconditional Jump Relative-EB
10115	X	BITJ	Bit-by-Bit Jump
10120		CIL0	Clear Master Lockout
10121		CIL1	Clear Interrupt Lockout, Channel 1
10122		CIL2	Clear Interrupt Lockout, Channel 2
10123		CIL3	Clear Interrupt Lockout, Channel 3
10124		CIL4	Clear Interrupt Lockout, Channel 4
10125		CIL5	Clear Interrupt Lockout, Channel 5

FFFEE	G	MNEMONIC	OPERATION
10126		CIL6	Clear Interrupt Lockout, Channel 6
10127		CIL7	Clear Interrupt Lockout, Channel 7
10130		CTAQ	Bank Controls to AQ
10140		SIL0	Set Master Lockout
10141		SIL1	Set Interrupt Lockout, Channel 1
10142		SIL2	Set Interrupt Lockout, Channel 2
10143		SIL3	Set Interrupt Lockout, Channel 3
10144		SIL4	Set Interrupt Lockout, Channel 4
10145		SIL5	Set Interrupt Lockout, Channel 5
10146		SIL6	Set Interrupt Lockout, Channel 6
10147		SIL7	Set Interrupt Lockout, Channel 7
10152	Y	EXC2	External Function Constant, Channel 2
10153	Y	EXC3	External Function Constant, Channel 3
10154		EXC4	External Function Constant, Channel 4
10155	Y	EXC5	External Function Constant, Channel 5
10156	Y	EXC6	External Function Constant, Channel 6
10157	Y	EXC7	External Function Constant, Channel 7
10162	X	MTM2	Memory-to-Memory Transfer, Channel 2
10164	X	MTM4	Memory-to-Memory Transfer, Channel 4
10166	X	MTM6	Memory-to-Memory Transfer, Channel 6
10172		CBC2	Clear Buffer Controls, Channel 2
10173		CBC3	Clear Buffer Controls, Channel 3
10174		CBC4	Clear Buffer Controls, Channel 4
10175		CBC5	Clear Buffer Controls, Channel 5
10176		CBC6	Clear Buffer Controls, Channel 6
10177		CBC7	Clear Buffer Controls, Channel 7
103ZZ	X	JPRG	Return Jump - EM
104ZZ	X	ZJ	Zero Jump - EM

FFFEE	G	MNEMONIC	OPERATION
105ZZ	X	NZ	Non-Zero Jump - EM
106ZZ	X	PJ	Positive Jump - EM
107ZZ	X	NJ	Negative Jump - EM
110ZZ	X	LP	Logical Product - EM
11100	X	LPIB	Logical Product Indirect - EB
11101	X	LPRB	Logical Product Relative - EB
111YY	X	LPMX	Logical Product - MX
113YY		ARS	A Right Shift
114ZZ	X	SC	Selective Complement - EM
11500	X	SCIB	Selective Complement Indirect - EB
11501	X	SCRB	Selective Complement Relative - EB
115YY	X	SCMX	Selective Complement - MX
117YY		ALS	A Left Shift
120ZZ	X	LD	Load - EM
12100	X	LDIB	Load Indirect - EB
12101	X	LDRB	Load Relative - EB
121YY	X	LDMX	Load - MX
123YY		QRS	Q Right Shift
124ZZ	X	LC	Load Complement - EM
12500	X	LCIB	Load Complement Indirect - EB
12501	X	LCRB	Load Complement Relative - EB
125YY	X	LCMX	Load Complement - MX
127YY		QLS	Q Left Shift
130ZZ	X	AD	Add - EM
13100	X	ADIB	Add Indirect - EB
13101	X	ADRB	Add Relative - EB
131YY	X	ADMX	Add - MX
133YY		LRS	AQ Right Shift
134ZZ	X	SB	Subtract - EM
13500	X	SBIB	Subtract Indirect - EB
13501	X	SBRB	Subtract Relative - EB
135YY	X	SBMX	Subtract - MX
137YY		LLS	AQ Left Shift
140ZZ	X	ST	Store - EM
14100	X	STIB	Store Indirect - EB
14101	X	STRB	Store Relative - EB
141YY	X	STMX	Store - MX

FFFEE	G	MNEMONIC	OPERATION
143ZZ	X	SRJP	Set Rel Bank Control; Jump-EM
144ZZ	X	SR	Shift Replace - EM
14500	X	SRIB	Shift Replace Indirect - EB
14501	X	SRRB	Shift Replace Relative - EB
145YY	X	SRMX	Shift Replace - MX
147ZZ	X	DRJP	Set Dir and Rel Bank Controls; Jump - EM
150ZZ	X	RA	Replace Add - EM
15100	X	RAIB	Replace Add Indirect - EB
15101	X	RARB	Replace Add Relative - EB
151YY	X	RAMX	Replace Add - MX
153YY		SDCG	Set Dir Bank Control - EM
154ZZ	X	AO	Replace Add One - EM
15500	X	AOIB	Replace Add One Indirect - EB
15501	X	AORB	Replace Add One Relative - EB
155YY	X	AOMX	Replace Add One - MX
157YY		SICG	Set Indir Bank Control - EM
160ZZ	X	LQ	Load Q - EM
16100	X	LQIB	Load Q Indirect - EB
16101	X	LQRB	Load Q Relative - EB
161YY	X	LQMX	Load Q - MX
16200	Y	LQC	Load Q Constant
164ZZ	X	SQ	Store Q - EM
16500	X	SQIB	Store Q Indirect - EB
16501	X	SQRB	Store Q Relative - EB
165YY	X	SQMX	Store Q - MX
16600	Y	SQC	Store Q Constant
167ZZ	X	HW	Half Write - EM
170ZZ	X	MU	Multiply - EM
17100	X	MUIB	Multiply Indirect - EB
17101	X	MURB	Multiply Relative - EB
171YY	X	MUMX	Multiply - MX
17200	Y	MUC	Multiply Constant
173ZZ	X	HILO	High-Low Comparison
174ZZ	X	DV	Divide - EM
17500	X	DVIB	Divide Indirect - EB
17501	X	DVRB	Divide Relative - EB
175YY	X	DVMX	Divide - MX

FFFEE	G	MNEMONIC	OPERATION
17600	Y	DVC	Divide Constant
177YY		HLTG	Halt

APPENDIX C

MNEMONIC INSTRUCTION CODES

NOTE: EB - Entire Bank; EM - Entire Memory; MX - Memory Index

MNEMONIC	FFFEE	G	OPERATION
ACJ	0007X		Subtract No Address
AD	130ZZ	X	Add - EM
ADB	033XX		Add Backward
ADC	03200	Y	Add Constant
ADD	030XX		Add Direct
ADF	032XX		Add Forward
ADI	031XX		Add Indirect
ADIB	13100	X	Add Indirect - EB
ADM	03100	X	Add Memory
ADMX	131YY	X	Add - MX
ADN	006YY		Add No Address
ADR			Add Relative (032XX or 033XX)
ADRB	13101	X	Add Relative - EB
ADS	03300		Add Specific
AMOD	10100		Select A Mode
AO	154ZZ	X	Replace Add One - EM
AOB	057XX		Replace Add One - Backward
AOC	05600	Y	Replace Add One Constant
AOD	054XX		Replace Add One Direct
AOF	056XX		Replace Add One Forward
AOI	055XX		Replace Add One Indirect
AOIB	15500	X	Replace Add One Indirect - EB
AOM	05500	X	Replace Add One Memory
AOMX	155YY	X	Replace Add One - MX
AOR			Replace Add One Relative (056XX or 057XX)
AORB	15501	X	Replace Add One Relative - EB
AOS	05700		Replace Add One Specific
ALS	117YY		A Left Shift
ARS	113YY		A Right Shift

MNEMONIC	FFFEE	G	OPERATION
ATE	00105	X	A to BER
ATE2	10022	X	A to BER, Channel 2
ATE3	10023	X	A to BER, Channel 3
ATE4	10024	X	A to BER, Channel 4
ATE5	10025	X	A to BER, Channel 5
ATE6	10026	X	A to BER, Channel 6
ATE7	10027	X	A to BER, Channel 7
ATX	00106	X	A to BXR
ATX2	10032	X	A to BXR, Channel 2
ATX3	10033	X	A to BXR, Channel 3
ATX4	10034	X	A to BXR, Channel 4
ATX5	10035	X	A to BXR, Channel 5
ATX6	10036	X	A to BXR, Channel 6
ATX7	10037	X	A to BXR, Channel 7
BBC1	10011	Y	Set Internal BBC
BBC2	10012	Y	Set External 2 BBC
BBC3	10013	Y	Set External 3 BBC
BBC4	10014	Y	Set External 4 BBC
BBC5	10015	Y	Set External 5 BBC
BBC6	10016	Y	Set External 6 BBC
BBC7	10017	Y	Set External 7 BBC
BITJ	10115	X	Bit-by-Bit Jump
BLS	00100	X	Block Store
CBC	00104		Clear Buffer Controls
CBC2	10172		Clear Buffer Controls, Channel 2
CBC3	10173		Clear Buffer Controls, Channel 3
CBC4	10174		Clear Buffer Controls, Channel 4
CBC5	10175		Clear Buffer Controls, Channel 5
CBC6	10176		Clear Buffer Controls, Channel 6
CBC7	10177		Clear Buffer Controls, Channel 7
CIL	00120		Clear Interrupt Lockout
CIL0	10120		Clear Master Lockout
CIL1	10121		Clear Internal Lockout
CIL2	10122		Clear Interrupt Lockout, Channel 2
CIL3	10123		Clear Interrupt Lockout, Channel 3
CIL4	10124		Clear Interrupt Lockout, Channel 4
CIL5	10125		Clear Interrupt Lockout, Channel 5
CIL6	10126		Clear Interrupt Lockout, Channel 6
CIL7	10127		Clear Interrupt Lockout, Channel 7

MNEMONIC	FFFEE	G	OPERATION
CTA	00130		Bank Controls to A
CTAQ	10130		Bank Controls to AQ
CTCI	10102		Computer-to-Computer Interrupt
DRJ	0005X		Set Direct and Relative Bank Controls; Jump
DRJP	147ZZ	X	Set Direct and Relative Bank Controls; Jump - EM
DV	174ZZ	X	Divide - EM
DVC	17600	Y	Divide Constant
DVIB	17500	X	Divide Indirect - EB
DVMX	175YY	X	Divide - MX
DVRB	17501	X	Divide Relative - EB
ERR	00000		Error Stop
ERRG	10000		Error Stop
ERTA	10105		Error Register to A
ETA	00107		BER to A
ETA2	10042		BER, Channel 2, to A
ETA3	10043		BER, Channel 3, to A
ETA4	10044		BER, Channel 4, to A
ETA5	10045		BER, Channel 5, to A
ETA6	10046		BER, Channel 6, to A
ETA7	10047		BER, Channel 7, to A
EXC	07500	Y	Ext Function Constant
EXC2	10152	Y	Ext Function Constant, Ch 2
EXC3	10153	Y	Ext Function Constant, Ch 3
EXC4	10154	Y	Ext Function Constant, Ch 4
EXC5	10155	Y	Ext Function Constant, Ch 5
EXC6	10156	Y	Ext Function Constant, Ch 6
EXC7	10157	Y	Ext Function Constant, Ch 7
EXF	075XX		Ext Function Forward
GMOD	10101		Select G Mode
HILO	173ZZ	X	High-Low Comparison
HLT	07700		Halt
HLT	07777		Halt
HLTG	177YY		Halt
HW	167ZZ	X	Half Right - EM
HWI	076XX		Half Right Indirect
IBI	07200	X	Initiate Buffer Input
IBI2	10062	X	Initiate Buffer Input, Channel 2

MNEMONIC	FFFEE	G	OPERATION
IBI3	10063	X	Initiate Buffer Input, Channel 3
IBI4	10064	X	Initiate Buffer Input, Channel 4
IBI5	10065	X	Initiate Buffer Input, Channel 5
IBI6	10066	X	Initiate Buffer Input, Channel 6
IBI7	10067	X	Initiate Buffer Input, Channel 7
IBO	07300	X	Initiate Buffer Output
IBO2	10072	X	Initiate Buffer Output, Channel 2
IBO3	10073	X	Initiate Buffer Output, Channel 3
IBO4	10074	X	Initiate Buffer Output, Channel 4
IBO5	10075	X	Initiate Buffer Output, Channel 5
IBO6	10076	X	Initiate Buffer Output, Channel 6
IBO7	10077	X	Initiate Buffer Output, Channel 7
INA	07600		Input to A
INA2	10052		Input to A, Channel 2
INA3	10053		Input to A, Channel 3
INA4	10054		Input to A, Channel 4
INA5	10055		Input to A, Channel 5
INA6	10056		Input to A, Channel 6
INA7	10057		Input to A, Channel 7
INP	072XX		Normal Input
IRJ	0003X		Set Indirect and Relative Bank Controls; Jump
JFI	071XX		Jump Forward Indirect
JPI	070XX		Jump Indirect
JPIB	10106	X	Jump Indirect - EB
JPR	07100	X	Return Jump
JPRG	103ZZ	X	Return Jump - EM
JRIB	10107	X	Jump Relative Indirect - EB
LC	124ZZ	X	Load Complement - EM
LCB	027XX		Load Complement Backward
LCC	02600	Y	Load Complement Constant
LCD	024XX		Load Complement Direct
LCF	026XX		Load Complement Forward
LCI	025XX		Load Complement Indirect
LCIB	12500	X	Load Complement Indirect-EB
LCM	02500	X	Load Complement Memory
LCMX	125YY	X	Load Complement - MX
LCN	005YY		Load Complement, No Address
LCR			Load Complement, Relative (026XX or 027XX)

MNEMONIC	FFFEE	G	OPERATION
LCRB	12501	X	Load Complement Relative-EB
LCS	02700		Load Complement Specific
LD	120ZZ	X	Load - EM
LDB	023XX		Load Backward
LDC	02200	Y	Load Constant
LDD	020XX		Load Direct
LDF	022XX		Load Forward
LDI	021XX		Load Indirect
LDIB	12100	X	Load Indirect - EB
LDM	02100	X	Load Memory
LDMX	121YY	X	Load - MX
LDN	004YY		Load No Address
LDR			Load Relative (022XX or 023XX)
LDRB	12101	X	Load Relative - EB
LDS	02300		Load Specific
LLS	137YY		AQ Left Shift
LP	110ZZ	X	Logical Product - EM
LPB	013XX		Logical Product Backward
LPC	01200	X	Logical Product Constant
LPD	010XX		Logical Product Direct
LPF	012XX		Logical Product Forward
LPI	011XX		Logical Product Indirect
LPIB	11100	X	Logical Product Indirect - EB
LPM	01100	X	Logical Product Memory
LPMX	111YY	X	Logical Product - MX
LPN	002YY		Logical Product No Address
LPR			Logical Product Relative (012XX or 013XX)
LPRB	11101	X	Logical Product Relative - EB
LPS	01300		Logical Product Specific
LQ	160ZZ	X	Load Q
LQC	16200	Y	Load Q Constant
LQIB	16100	X	Load Q Indirect - EB
LQMX	161YY	X	Load Q - MX
LQRB	16101	X	Load Q Relative - EB
LRS	133YY		AQ Right Shift
LS1	00102		Left Shift One
LS2	00103		Left Shift Two
LS3	00110		Left Shift Three

MNEMONIC	FFFEE	G	OPERATION
LS6	00111		Left Shift Six
MTM2	10162	X	Memory-to-Memory Transfer, Channel 2
MTM4	10164	X	Memory-to-Memory Transfer, Channel 4
MTM6	10166	X	Memory-to-Memory Transfer, Channel 6
MUH	00113		Multiply A by 100_{10}
MUT	00112		Multiply A by 10_{10}
MU	170ZZ	X	Multiply - EM
MUC	17200	Y	Multiply Constant
MUIB	17100	X	Multiply Indirect - EB
MUMX	171YY	X	Multiply - MX
MURB	17101	X	Multiply Relative - EB
NJ	107ZZ	X	Negative Jump - EM
NJB	067XX		Negative Jump Backward
NJF	063XX		Negative Jump Forward
NJR			Negative Jump Relative
NJRB	10113	X	Negative Jump Relative - EB
NOP	0000X		No Operation
NOPG	1000X		No Operation
NZ	105ZZ	X	Non-Zero Jump - EM
NZB	065XX		Non-Zero Jump Backward
NZF	061XX		Non-Zero Jump Forward
NZR			Non-Zero Jump Relative
NZRB	10111	X	Non-Zero Jump Relative - EB
OTA	07677		Output From A
OTN	074YY		Output No Address
OUT	073XX		Normal Output
PJ	106ZZ	X	Positive Jump - EM
PJB	066XX		Positive Jump Backward
PJF	062XX		Positive Jump Forward
PJR			Positive Jump Relative
PJRB	10112	X	Positive Jump Relative - EB
PTA	00101		P to A
QLS	127YY		Q Left Shift
QRS	123YY		Q Right Shift
RA	150ZZ	X	Replace Add - EM
RAB	053XX		Replace Add Backward

MNEMONIC	FFFEE	G	OPERATION
RAC	05200	Y	Replace Add Constant
RAD	050XX		Replace Add Direct
RAF	052XX		Replace Add Forward
RAI	051XX		Replace Add Indirect
RAIB	15100	X	Replace Add Indirect - EB
RAM	05100	X	Replace Add Memory
RAMX	151YY	X	Replace Add - MX
RAR			Replace Add Relative (052XX or 053XX)
RARB	15101	X	Replace Add Relative - EB
RAS	05300		Replace Add Specific
RCJP	10103	X	AQ Bank Controls; Jump
RS1	00114		Right Shift One
RS2	00115		Right Shift Two
SB	134ZZ	X	Subtract - EM
SBB	037XX		Subtract Backward
SBC	03600	Y	Subtract Constant
SBD	034XX		Subtract Direct
SBF	036XX		Subtract Forward
SBI	035XX		Subtract Indirect
SBIB	13500	X	Subtract Indirect - EB
SBM	03500	X	Subtract Memory
SBMX	135YY	X	Subtract - MX
SBN	007YY		Subtract No Address
SBR			Subtract Relative (036XX or 037XX)
SBRB	13501	X	Subtract Relative - EB
SBS	03700		Subtract Specific
SBU	0014X		Set Buffer Bank Control
SC	114ZZ	X	Selective Complement - EM
SCB	017XX		Selective Complement Backward
SCC	01600	Y	Selective Complement Constant
SCD	014XX		Selective Complement Direct
SCF	016XX		Selective Complement Forward
SCI	015XX		Selective Complement Indirect
SCIB	11500	X	Selective Complement Indirect-EB
SCM	01500	X	Selective Complement Memory

MNEMONIC	FFFEE	G	OPERATION
SCMX	115YY	X	Selective Complement - MX
SCN	003YY		Selective Complement No Address
SCR			Selective Complement Relative (016XX or 017XX)
SCRB	11501	X	Selective Complement Relative-EB
SCS	01700		Selective Complement Specific
SDC	0004X		Set Direct Bank Control
SDCG	153YY		Set Direct Bank Control-EM
SIC	0002X		Set Indirect Bank Control
SICG	157YY		Set Indirect Bank Control-EM
SID	0006X		Set Indirect and Direct Bank Controls
SIL0	10140		Set Master Lockout
SIL1	10141		Set Indirect Lockout, Channel 1
SIL2	10142		Set Indirect Lockout, Channel 2
SIL3	10143		Set Indirect Lockout, Channel 3
SIL4	10144		Set Indirect Lockout, Channel 4
SIL5	10145		Set Indirect Lockout, Channel 5
SIL6	10146		Set Indirect Lockout, Channel 6
SIL7	10147		Set Indirect Lockout, Channel 7
SJS	077YY	X	Selective Stop; Jump
SLJ	077Y0	X	Selective Jump
SLS	0770Y		Selective Stop
SQ	164ZZ	X	Store Q - EM
SQC	16600	Y	Store Q Constant
SQIB	16500	X	Store Q Indirect - EB
SQMX	165YY	X	Store Q - MX
SQRB	16501	X	Store Q Relative - EB
SRJ	0001X		Set Relative Bank Control; Jump
SRJP	143ZZ	X	Set Relative Bank Control; Jump - EM
SR	144ZZ	X	Shift Replace - EM
SRB	047XX		Shift Replace Backward
SRC	04600	Y	Shift Replace Constant
SRD	044XX		Shift Replace Direct
SRF	046XX		Shift Replace Forward

MNEMONIC	FFFEE	G	OPERATION
SRI	045XX		Shift Replace Indirect
SRIB	14500	X	Shift Replace Indirect - EB
SRM	04500	X	Shift Replace Memory
SRMX	145YY	X	Shift Replace - MX
SRR			Shift Replace Relative (046XX or 047XX)
SRRB	14501	X	Shift Replace Relative - EB
SRS	04700		Shift Replace Specific
STE	0016X		Store BER
STP	0015X		Store P at Location 5X
ST	140ZZ	X	Store - EM
STB	043XX		Store Backward
STC	04200	Y	Store Constant
STD	040XX		Store Direct
STF	042XX		Store Forward
STI	041XX		Store Indirect
STIB	14100	X	Store Indirect - EB
STM	04100	X	Store Memory
STMX	141YY	X	Store - MX
STR			Store Relative (042XX or 043XX)
STRB	14101	X	Store Relative - EB
STS	04300		Store Specific
UJRB	10114	X	Unconditional Jump Relative - EB
XAQ	10104		Interchange A and Q
ZJ	104ZZ	X	Zero Jump - EM
ZJB	064XX		Zero Jump Backward
ZJF	060XX		Zero Jump Forward
ZJR			Zero Jump Relative (060XX or 064XX)
ZJRB	10110	X	Zero Jump Relative - EB

APPENDIX D

INSTRUCTION EXECUTION TIMES

NOTES: 1. Timing is the approximate number of memory cycles, where one memory cycle is equal to 1.35 microseconds. Two instruction times are given for instructions which take advantage of overlap in memory references. The first time given is a maximum time, assuming all references are made to the same bank of memory. The second time given is a minimum time, assuming instructions are in one bank of memory, data in another bank, and index registers are in a third bank of memory.

2. A single asterisk instead of a timing number indicates timing is determined by external equipment. A double asterisk indicates that timing is dependent on cable length. A triple asterisk indicates timing is dependent on the number of shifts involved. If $n \leq 7$, shift time is equal to one memory cycle, where n = number of shifts. If $n > 7$, shift time $= 1 + \frac{n - 7}{11}$.

FFFEE	G	Mnemonic		Timing
00000		ERR		1
0000X		NOP		1
0001X		SRJ		1
0002X		SIC		1
0003X		IRJ		1
0004X		SDC		1
0005X		DRJ		1
0006X		SID		1
0007X		ACJ		1
00100	X	BLS	(no jump)	1
			(jump)	2
00101		PTA		1
00102		LS1		1

FFFEE	G	Mnemonic		Timing
00103		LS2		1
00104		CBC		1
00105	X	ATE	(no jump)	1
			(jump)	2
00106	X	ATX	(no jump)	1
			(jump)	2
00107		ETA		1
00110		LS3		1
00111		LS6		1
00112		MUT		1
00113		MUH		1
00114		RS1		1
00115		RS2		1
00120		CIL		1
00130		CTA		1
0014X		SBU		1
0015X		STP		2
0016X		STE		3
002XX		LPN		1
003XX		SCN		1
004XX		LDN		1
005XX		LCN		1
006XX		ADN		1
007XX		SBN		1
010XX		LPD		2/1.5
01100	X	LPM		3/2
011XX		LPI		3/2
01200	Y	LPC		2
012XX		LPF		2
01300		LPS		2/1.5
013XX		LPB		2
014XX		SCD		2/1.5
01500	X	SCM		3/2
015XX		SCI		3/2
01600	Y	SCC		2
016XX		SCF		2
01700		SCS		2/1.5
017XX		SCB		2
020XX		LDD		2/1.5
02100	X	LDM		3/2
021XX		LDI		3/2

FFFEE	G	Mnemonic	Timing
02200	Y	LDC	2
022XX		LDF	2
02300		LDS	2/1.5
023XX		LDB	2
024XX		LDC	2/1.5
02500	X	LCM	3/2
025XX		LCI	3/2
02600	Y	LCC	2
026XX		LCF	2
02700		LCF	2/1.5
027XX		LCB	2
030XX		ADD	2/1.5
03100	X	ADM	3/2
031XX		ADI	3/2
03200	Y	ADC	2
032XX		ADF	2
03300		ADS	2/1.5
033XX		ADB	2
034XX		SBD	2/1.5
03500	X	SBI	3/2
035XX		SBI	3/2
03600	Y	SBC	2
036XX		SBF	2
03700		SBS	2/1.5
037XX		SBB	2
040XX		STD	2/1.5
04100	X	STM	3/2
041XX		STI	3/2
04200	Y	STC	2
042XX		STF	2
04300		STS	2/1.5
043XX		STB	2
044XX		SRD	3/2.5
04500	X	SRM	4/3
045XX		SRI	4/3
04600	Y	SRC	3
046XX		SRF	3
04700		SRS	3/2.5
047XX		SRB	3
050XX		RAD	3/2.5
05100	X	RAM	4/3

FFFEE	G	Mnemonic		Timing
051XX		RAI		4/3
05200	Y	RAC		3
052XX		RAF		3
05300		RAS		3/2.5
053XX		RAB		3
054XX		AOD		3/2.5
05500	X	AOM		4/3
055XX		AOI		4/3
05600	Y	AOC		3
056XX		AOF		3
05700		AOS		3/2.5
057XX		AOB		3
060XX		ZJF		1
061XX		NZF		1
062XX		PJF		1
063XX		NJF		1
064XX		ZJB		1
065XX		NZB		1
066XX		PJB		1
067XX		NJB		1
070XX		JPI		2/1.5
07100	X	JPR		3
071XX		JFI		2
07200	X	IBI	(no jump)	1
			(jump)	2
072XX	Y	INP		*
07300	X	IBO	(no jump)	1
			(jump)	2
073XX	Y	OUT		*
074YY		OTN		*
07500	Y	EXC		*
075XX		EXF		*
07600		INA		*
076XX		HWI		3/2
07677		OTA		*
07700		HLT		1
0770Y		SLS		1
077Y0	X	SLJ	(no jump)	1
			(jump)	2
077YY	X	SJS	(no jump)	1
			(jump)	2

FFFEE	G	Mnemonic		Timing
07777		HLT		1
10000		ERRG		1
1000X		NOPG		1
1001Y	Z	BBCY		2
1002Y	X	ATEY	(no jump)	1
			(jump)	2
1003Y	X	ATXY	(no jump)	1
			(jump)	2
1004Y		ETAY		1/*
1005Y		INAY		*
1006Y	X	IBIY	(no jump)	1
			(jump)	2
1007Y	X	IBOY	(no jump)	1
			(jump)	2
10100		AMOD		1
10101		GMOD		1
10102		CTCI		1
10103	X	RCJP		2
10104		XAQ		1
10105		ERTA		1
10106	X	JPIB		3/2.5
10107	X	JRIB		3
10110	X	ZJRB	(no jump)	1
			(jump)	2
10111	X	NZRB	(no jump)	1
			(jump)	2
10112	X	PJRB	(no jump)	1
			(jump)	2
10113	X	NJRB	(no jump)	1
			(jump)	2
10114	X	UJRB		2
10115	X	BITJ		2
1012Y		CILY		1
10130		CTAQ		1
1014Y		SILY		1
1015Y	Z	EXCY		*
1016Y	X	MTMY	(no jump)	1
			(jump)	2
1017Y		CBCY		1
103ZZ	X	JPRG		4
104ZZ	X	ZJ	(no jump)	1
			(jump)	2

FFFEE	G	Mnemonic		Timing
105ZZ	X	NZ	(no jump)	1
			(jump)	2
106ZZ	X	PJ	(no jump)	1
			(jump)	2
107ZZ	X	NJ	(no jump)	1
			(jump)	2
110ZZ	X	LP		3/2
11100	X	LPIB		4/2.5
11101	X	LPRB		3
111YY	X	LPMX		4/2.5
113YY		ARS		***
114ZZ	X	SC		3/2
11500	X	SCIB		4/2.5
11501	X	SCRB		3
115YY	X	SCMX		4/2.5
117YY		ALS		***
120ZZ	X	LD		2-1/2
12100	X	LDIB		4/2.5
12101	X	LDRB		3
121YY	X	LDMX		4/2.5
123YY		QRS		***
124ZZ	X	LC		3/2
12500	X	LCIB		4/2.5
12501	X	LCRB		3
125YY	X	LCMX		4/2.5
127YY		QLS		***
130ZZ	X	AD		3/2
13100	X	ADIB		4/2.5
13101	X	ADRB		3
131YY	X	ADMX		4/2.5
133YY		LRS		***
134ZZ	X	SB		3/2
13500	X	SBIB		4/2.5
13501	X	SBRB		3
135YY	X	SBMX		4/2.5
137YY		LLS		***
140ZZ	X	ST		3/2
14100	X	STIB		4/2.5
14101	X	STRB		3
141YY	X	STMX		4/2.5

FFFEE	G	Mnemonic	Timing
143ZZ	X	SRJP	2
144ZZ	X	SR	4/3
14500	X	SRIB	5/3.5
14501	X	SRRB	3
145YY	X	SRMX	5/3.5
147ZZ	X	DRJP	2
150ZZ	X	RA	4/3
15100	X	RAIB	5/3.5
15101	X	RARB	4
151YY	X	RAMX	5/3.5
153ZZ		SDCG	1
154ZZ	X	AO	4/3
15500	X	AOIB	5/3.5
15501	X	AORB	4
155YY	X	AOMX	5/3.5
157ZZ		SICG	1
160ZZ	X	LQ	3/2
16100	X	LQIB	4/2.5
16101	X	LQRB	3
161YY	X	LQMX	4/2.5
16200	Y	LQC	2
164ZZ	X	SQ	3/2
16500	X	SQIB	4/2.5
16501	X	SQRB	3
165YY	X	SQMX	4/2.5
16600	Y	SQC	2
167ZZ	X	HW	3/2
170ZZ	X	MU	5/4
17100	X	MUIB	6/4.5
17101	X	MURB	5
171YY	X	MUMX	6/4.5
17200	Y	MUC	4
173ZZ	X	HILO	4
174ZZ	X	DV	6/5
17500	X	DVIB	7/5.5
17501	X	DVRB	6
175YY	X	DVMX	7/5.5
17600	Y	DVC	5
177XX		HLTG	1

APPENDIX E

EXTERNAL FUNCTION CODES AND STATUS RESPONSES

1. 176G INPUT/OUTPUT TYPEWRITER

A. External Function Codes
4210 Select typewriter output
4220 Select typewriter input
4240 Request typewriter status

B. Status Response Codes
0000 Typewriter ready
0004 Typewriter power off
0010 Typewriter not in computer status
0020 Input character ready
0040 Output in use

NOTE: If a second typewriter is added, the master bits will be 43.

2. 162G MAGNETIC TAPE SYNCHRONIZER

A. External Function Codes
111X Write tape X (6-bit) if OUT is given
111X Write end-of-file mark if no OUT is given
211X Write tape X (12-bit) if OUT is given
112X Backspace tape X one record if INA is given
112X Search backward to end-of-file mark on tape X if no INA is given
113X Read forward tape X (6-bit) if INP is given
213X Read forward tape X (12-bit) if INP is given
113X Search forward tape X for end-of-file mark if no INP is given
114X Request tape X status (6-bit)
115X Rewind unload tape X
116X Rewind load tape X
1171 Set tapes to odd parity

1172 Set tapes to even parity (BCD)
214X Request tape X status (12-bit)
210X High density, tape X
110X Low density, tape X

B. Status Response Codes
0000 Odd parity selected - no errors
0001 Even parity selected - no errors
0002 Tape X not ready
0004 Horizontal and/or vertical parity error
0015 Illegal BCD detected on write
0020 End-of-file mark read
0040 End-of-tape or load point sensed
0100 Tape X high density
0200 Tape X busy

NOTE: The master bits 12, 13, 22, and 23 are used for second and third tape control. The "X" in the above codes can range from 0 to 7.

3. 165G PLOTTER

A. External Function Codes
4401 Select plotter for write operation
4440 Select plotter for read operation

B. Follow 4401 with output instruction and transmit one or more of these:
0001 Move carriage and pen .01" in +X direction
0002 Move carriage and pen .01" in -X direction
0004 Rotate drum .01" in -Y direction
0005 Carriage and pen move .01" in +X direction, drum rotates in -Y direction .01"
0006 Carriage and pen move .01" in -X direction, drum rotates in -Y direction .01"
0010 Rotate drum .01" in +Y direction
0011 Carriage and pen move .01" in +X direction, drum rotates in +Y direction .01"
0012 Carriage and pen move .01" in -X direction, drum rotates in +Y direction .01"
0020 Move pen down to paper
0040 Move pen away from paper

C. Status Response Codes
Status is obtained by selecting the unit for reading. The obtained status is the value of the 12 switches on the unit.

4. 405 CARD READER (Hollerith Facility)

A. External Function Codes
4500 EF clear
4501 Free run read (automatic Hollerith to BCD and pack)
4502 Single cycle read (automatic Hollerith to BCD and pack)
4505 FRR, card image input
4506 SCR, card image input
4510 Gate card to secondary bin
4540 Check status

B. Status Response Codes
0000 Card reader ready
0001 Hopper empty
0002 Primary or secondary stacker jammed
0004 Read failure
0010 Late input request
0020 Amplifier failure
0040 Manual or motor power off
0100 Read comparison error
0200 End-of-file
0400 Hollerith to BCD on last read

5. 170G CARD PUNCH CONTROL UNIT

A. External Function Codes
3000 EF clear
3002 Punch
3040 Check status

B. Status Response Codes
0000 170G ready
0200 MS in 1604 position
2000 Punch not ready

6. 1612G HIGH-SPEED PRINTER

A. External Function Codes

0600 Select printer and do not interrupt on ready
0601 Space paper one line
0602 Space paper two lines
0603 Skip to format channel 7
0604 Skip to format channel 8
0605 Print information and advance paper
0606 Do not advance paper after next print
0607 Select printer and interrupt on ready
0610 Clear monitor channels 1-6
0611 Select monitor channel 1
0612 Select monitor channel 2
0613 Select monitor channel 3
0614 Select monitor channel 4
0615 Select monitor channel 5
0616 Select monitor channel 6

B. Status Response Codes

0000 Printer not ready
4000 Printer ready

NOTE: Status is always available on the 1612G. No request is necessary.

7. 8528G DIGITAL DATA TERMINAL

A. External Function Codes

3400 Select status
3401 Select interrupt
3402 Select receive
3404 Select transmit
3406 Clear interrupt selection

B. Status Response Codes

0001 Error
0002 Fake ready set
4000 Interrupt

APPENDIX F

INPUT/OUTPUT TYPEWRITER CODES

Characters UC	LC	Code	Characters UC	LC	Code
A	a	30	X	x	27
B	b	23	Y	y	25
C	c	16	Z	z	21
D	d	22	)	0	56
E	e	20	*	1	74
F	f	26	@	2	70
G	g	13	#	3	64
H	h	05	$	4	62
I	i	14	%	5	66
J	j	32	¢	6	72
K	k	36	&	7	60
L	l	11	1/2	8	33
M	m	07	(	9	37
N	n	06	_	-	52
O	o	03	?	/	44
P	p	15	"	'	54
Q	q	35	°	+	46
R	r	12	.	.	42
S	s	24	:	;	50
T	t	01	,	,	40
U	u	34	+	=	02
V	v	17	tab	tab	51
W	w	31	space		04
Backspace		61	Carriage Return		45
Lower Case		57	Upper Case		47

APPENDIX G

1612G PRINTER CODES

Char	Code	Char	Code	Char	Code	Char	Code
Blank	20	F	66	V	25	≤	15
0	12	G	67	W	26	|	16
1	01	H	70	X	27	[	17
2	02	I	71	Y	30	]	32
3	03	J	41	Z	31	→	35
4	04	K	42	.	73	≡	36
5	05	L	43	-	40	~ ∧	37
6	06	M	44	+	60	% or ∨	52
7	07	N	45	=	13	$ or ¬	53
8	10	O	46	(	34	↑	55
9	11	P	47	)	74	↓	56
A	61	Q	50	/	21	>	57
B	62	R	51	*	54	<	72
C	63	S	22	,	33	≥	75
D	64	T	23	:	00	?	76
E	65	U	24	≠	14	;	77

In last column, codes ~ % $ appear if business application, ∧ ∨ ¬ for scientific application.

APPENDIX H

MAGNETIC TAPE BCD CODES

Character	Code (Octal)	Character	Code (Octal)
A	61	1	01
B	62	2	02
C	63	3	03
D	64	4	04
E	65	5	05
F	66	6	06
G	67	7	07
H	70	8	10
I	71	9	11
J	41	&	60
K	42	-	40
L	43	(blank)	20
M	44	/	21
N	45	. (period)	73
O	46	$	53
P	47	*	54
Q	50	, (comma)	33
R	51	%	34
S	22	#	13
T	23	@	14
U	24	⌑	74
V	25	0 (numerical zero)	12
W	26	record mark	32
X	27	0 (minus zero)	52
Y	30	0 (plus zero)	72
Z	31	group mark	77
0	12	tape mark	17

APPENDIX I

PUNCHED CARD CODES

Char	Card	BCD	Char	Card	BCD	Char	Card	BCD	Char	Card	BCD
			+	12	60	-	11	40			20
1	1	01	A	12 1	61	J	11 1	41	/	0 1	21
2	2	02	B	12 2	62	K	11 2	42	S	0 2	22
3	3	03	C	12 3	63	L	11 3	43	T	0 3	23
4	4	04	D	12 4	64	M	11 4	44	U	0 4	24
5	5	05	E	12 5	65	N	11 5	45	V	0 5	25
6	6	06	F	12 6	66	O	11 6	46	W	0 6	26
7	7	07	G	12 7	67	P	11 7	47	X	0 7	27
8	8	10	H	12 8	70	Q	11 8	50	Y	0 8	30
9	9	11	I	12 9	71	R	11 9	51	Z	0 9	31
0	0	12		12 0	72	$\bar{O}$	11 0	52			
=	8,3	13	.	12 8,3	73	$	11 8,3	53	.	0 8,3	33
-	8,4	14	)	12 8,4	74	*	11 8,4	54	(	0 8,4	34

APPENDIX J

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

APPENDIX K

OCTAL-DECIMAL INTEGER CONVERSION TABLE

OCTAL-DECIMAL INTEGER CONVERSION TABLE

0000 to 0777 (Octal) — 0000 to 0511 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
0000	0000	0001	0002	0003	0004	0005	0006	0007
0010	0008	0009	0010	0011	0012	0013	0014	0015
0020	0016	0017	0018	0019	0020	0021	0022	0023
0030	0024	0025	0026	0027	0028	0029	0030	0031
0040	0032	0033	0034	0035	0036	0037	0038	0039
0050	0040	0041	0042	0043	0044	0045	0046	0047
0060	0048	0049	0050	0051	0052	0053	0054	0055
0070	0056	0057	0058	0059	0060	0061	0062	0063
0100	0064	0065	0066	0067	0068	0069	0070	0071
0110	0072	0073	0074	0075	0076	0077	0078	0079
0120	0080	0081	0082	0083	0084	0085	0086	0087
0130	0088	0089	0090	0091	0092	0093	0094	0095
0140	0096	0097	0098	0099	0100	0101	0102	0103
0150	0104	0105	0106	0107	0108	0109	0110	0111
0160	0112	0113	0114	0115	0116	0117	0118	0119
0170	0120	0121	0122	0123	0124	0125	0126	0127
0200	0128	0129	0130	0131	0132	0133	0134	0135
0210	0136	0137	0138	0139	0140	0141	0142	0143
0220	0144	0145	0146	0147	0148	0149	0150	0151
0230	0152	0153	0154	0155	0156	0157	0158	0159
0240	0160	0161	0162	0163	0164	0165	0166	0167
0250	0168	0169	0170	0171	0172	0173	0174	0175
0260	0176	0177	0178	0179	0180	0181	0182	0183
0270	0184	0185	0186	0187	0188	0189	0190	0191
0300	0192	0193	0194	0195	0196	0197	0198	0199
0310	0200	0201	0202	0203	0204	0205	0206	0207
0320	0208	0209	0210	0211	0212	0213	0214	0215
0330	0216	0217	0218	0219	0220	0221	0222	0223
0340	0224	0225	0226	0227	0228	0229	0230	0231
0350	0232	0233	0234	0235	0236	0237	0238	0239
0360	0240	0241	0242	0243	0244	0245	0246	0247
0370	0248	0249	0250	0251	0252	0253	0254	0255

	0	1	2	3	4	5	6	7
0400	0256	0257	0258	0259	0260	0261	0262	0263
0410	0264	0265	0266	0267	0268	0269	0270	0271
0420	0272	0273	0274	0275	0276	0277	0278	0279
0430	0280	0281	0282	0283	0284	0285	0286	0287
0440	0288	0289	0290	0291	0292	0293	0294	0295
0450	0296	0297	0298	0299	0300	0301	0302	0303
0460	0304	0305	0306	0307	0308	0309	0310	0311
0470	0312	0313	0314	0315	0316	0317	0318	0319
0500	0320	0321	0322	0323	0324	0325	0326	0327
0510	0328	0329	0330	0331	0332	0333	0334	0335
0520	0336	0337	0338	0339	0340	0341	0342	0343
0530	0344	0345	0346	0347	0348	0349	0350	0351
0540	0352	0353	0354	0355	0356	0357	0358	0359
0550	0360	0361	0362	0363	0364	0365	0366	0367
0560	0368	0369	0370	0371	0372	0373	0374	0375
0570	0376	0377	0378	0379	0380	0381	0382	0383
0600	0384	0385	0386	0387	0388	0389	0390	0391
0610	0392	0393	0394	0395	0396	0397	0398	0399
0620	0400	0401	0402	0403	0404	0405	0406	0407
0630	0408	0409	0410	0411	0412	0413	0414	0415
0640	0416	0417	0418	0419	0420	0421	0422	0423
0650	0424	0425	0426	0427	0428	0429	0430	0431
0660	0432	0433	0434	0435	0436	0437	0438	0439
0670	0440	0441	0442	0443	0444	0445	0446	0447
0700	0448	0449	0450	0451	0452	0453	0454	0455
0710	0456	0457	0458	0459	0460	0461	0462	0463
0720	0464	0465	0466	0467	0468	0469	0470	0471
0730	0472	0473	0474	0475	0476	0477	0478	0479
0740	0480	0481	0482	0483	0484	0485	0486	0487
0750	0488	0489	0490	0491	0492	0493	0494	0495
0760	0496	0497	0498	0499	0500	0501	0502	0503
0770	0504	0505	0506	0507	0508	0509	0510	0511

1000 to 1777 (Octal) — 0512 to 1023 (Decimal)

	0	1	2	3	4	5	6	7
1000	0512	0513	0514	0515	0516	0517	0518	0519
1010	0520	0521	0522	0523	0524	0525	0526	0527
1020	0528	0529	0530	0531	0532	0533	0534	0535
1030	0536	0537	0538	0539	0540	0541	0542	0543
1040	0544	0545	0546	0547	0548	0549	0550	0551
1050	0552	0553	0554	0555	0556	0557	0558	0559
1060	0560	0561	0562	0563	0564	0565	0566	0567
1070	0568	0569	0570	0571	0572	0573	0574	0575
1100	0576	0577	0578	0579	0580	0581	0582	0583
1110	0584	0585	0586	0587	0588	0589	0590	0591
1120	0592	0593	0594	0595	0596	0597	0598	0599
1130	0600	0601	0602	0603	0604	0605	0606	0607
1140	0608	0609	0610	0611	0612	0613	0614	0615
1150	0616	0617	0618	0619	0620	0621	0622	0623
1160	0624	0625	0626	0627	0628	0629	0630	0631
1170	0632	0633	0634	0635	0636	0637	0638	0639
1200	0640	0641	0642	0643	0644	0645	0646	0647
1210	0648	0649	0650	0651	0652	0653	0654	0655
1220	0656	0657	0658	0659	0660	0661	0662	0663
1230	0664	0665	0666	0667	0668	0669	0670	0671
1240	0672	0673	0674	0675	0676	0677	0678	0679
1250	0680	0681	0682	0683	0684	0685	0686	0687
1260	0688	0689	0690	0691	0692	0693	0694	0695
1270	0696	0697	0698	0699	0700	0701	0702	0703
1300	0704	0705	0706	0707	0708	0709	0710	0711
1310	0712	0713	0714	0715	0716	0717	0718	0719
1320	0720	0721	0722	0723	0724	0725	0726	0727
1330	0728	0729	0730	0731	0732	0733	0734	0735
1340	0736	0737	0738	0739	0740	0741	0742	0743
1350	0744	0745	0746	0747	0748	0749	0750	0751
1360	0752	0753	0754	0755	0756	0757	0758	0759
1370	0760	0761	0762	0763	0764	0765	0766	0767

	0	1	2	3	4	5	6	7
1400	0768	0769	0770	0771	0772	0773	0774	0775
1410	0776	0777	0778	0779	0780	0781	0782	0783
1420	0784	0785	0786	0787	0788	0789	0790	0791
1430	0792	0793	0794	0795	0796	0797	0798	0799
1440	0800	0801	0802	0803	0804	0805	0806	0807
1450	0808	0809	0810	0811	0812	0813	0814	0815
1460	0816	0817	0818	0819	0820	0821	0822	0823
1470	0824	0825	0826	0827	0828	0829	0830	0831
1500	0832	0833	0834	0835	0836	0837	0838	0839
1510	0840	0841	0842	0843	0844	0845	0846	0847
1520	0848	0849	0850	0851	0852	0853	0854	0855
1530	0856	0857	0858	0859	0860	0861	0862	0863
1540	0864	0865	0866	0867	0868	0869	0870	0871
1550	0872	0873	0874	0875	0876	0877	0878	0879
1560	0880	0881	0882	0883	0884	0885	0886	0887
1570	0888	0889	0890	0891	0892	0893	0894	0895
1600	0896	0897	0898	0899	0900	0901	0902	0903
1610	0904	0905	0906	0907	0908	0909	0910	0911
1620	0912	0913	0914	0915	0916	0917	0918	0919
1630	0920	0921	0922	0923	0924	0925	0926	0927
1640	0928	0929	0930	0931	0932	0933	0934	0935
1650	0936	0937	0938	0939	0940	0941	0942	0943
1660	0944	0945	0946	0947	0948	0949	0950	0951
1670	0952	0953	0954	0955	0956	0957	0958	0959
1700	0960	0961	0962	0963	0964	0965	0966	0967
1710	0968	0969	0970	0971	0972	0973	0974	0975
1720	0976	0977	0978	0979	0980	0981	0982	0983
1730	0984	0985	0986	0987	0988	0989	0990	0991
1740	0992	0993	0994	0995	0996	0997	0998	0999
1750	1000	1001	1002	1003	1004	1005	1006	1007
1760	1008	1009	1010	1011	1012	1013	1014	1015
1770	1016	1017	1018	1019	1020	1021	1022	1023

OCTAL-DECIMAL INTEGER CONVERSION TABLE

	0	1	2	3	4	5	6	7
2000	1024	1025	1026	1027	1028	1029	1030	1031
2010	1032	1033	1034	1035	1036	1037	1038	1039
2020	1040	1041	1042	1043	1044	1045	1046	1047
2030	1048	1049	1050	1051	1052	1053	1054	1055
2040	1056	1057	1058	1059	1060	1061	1062	1063
2050	1064	1065	1066	1067	1068	1069	1070	1071
2060	1072	1073	1074	1075	1076	1077	1078	1079
2070	1080	1081	1082	1083	1084	1085	1086	1087
2100	1088	1089	1090	1091	1092	1093	1094	1095
2100	1096	1097	1098	1099	1100	1101	1102	1103
2120	1104	1105	1106	1107	1108	1109	1110	1111
2130	1112	1113	1114	1115	1116	1117	1118	1119
2140	1120	1121	1122	1123	1124	1125	1126	1127
2150	1128	1129	1130	1131	1132	1133	1134	1135
2160	1136	1137	1138	1139	1140	1141	1142	1143
2170	1144	1145	1146	1147	1148	1149	1150	1151
2200	1152	1153	1154	1155	1156	1157	1158	1159
2210	1160	1161	1162	1163	1164	1165	1166	1167
2220	1168	1169	1170	1171	1172	1173	1174	1175
2230	1176	1177	1178	1179	1180	1181	1182	1183
2240	1184	1185	1186	1187	1188	1189	1190	1191
2250	1192	1193	1194	1195	1196	1197	1198	1199
2260	1200	1201	1202	1203	1204	1205	1206	1207
2270	1208	1209	1210	1211	1212	1213	1214	1215
2300	1216	1217	1218	1219	1220	1221	1222	1223
2310	1224	1225	1226	1227	1228	1229	1230	1231
2320	1232	1233	1234	1235	1236	1237	1238	1239
2330	1240	1241	1242	1243	1244	1245	1246	1247
2340	1248	1249	1250	1251	1252	1253	1254	1255
2350	1256	1257	1258	1259	1260	1261	1262	1263
2360	1264	1265	1266	1267	1268	1269	1270	1271
2370	1272	1273	1274	1275	1276	1277	1278	1279

	0	1	2	3	4	5	6	7
2400	1280	1281	1282	1283	1284	1285	1286	1287
2410	1288	1289	1290	1291	1292	1293	1294	1295
2420	1296	1297	1298	1299	1300	1301	1302	1303
2430	1304	1305	1306	1307	1308	1309	1310	1311
2440	1312	1313	1314	1315	1316	1317	1318	1319
2450	1320	1321	1322	1323	1324	1325	1326	1327
2460	1328	1329	1330	1331	1332	1333	1334	1335
2470	1336	1337	1338	1339	1340	1341	1342	1343
2500	1344	1345	1346	1347	1348	1349	1350	1351
2510	1352	1353	1354	1355	1356	1357	1358	1359
2520	1360	1361	1362	1363	1364	1365	1366	1367
2530	1368	1369	1370	1371	1372	1373	1374	1375
2540	1376	1377	1378	1379	1380	1381	1382	1383
2550	1384	1385	1386	1387	1388	1389	1390	1391
2560	1392	1393	1394	1395	1396	1397	1398	1399
2570	1400	1401	1402	1403	1404	1405	1406	1407
2600	1408	1409	1410	1411	1412	1413	1414	1415
2610	1416	1417	1418	1419	1420	1421	1422	1423
2620	1424	1425	1426	1427	1428	1429	1430	1431
2630	1432	1433	1434	1435	1436	1437	1438	1439
2640	1440	1441	1442	1443	1444	1445	1446	1447
2650	1448	1449	1450	1451	1452	1453	1454	1455
2660	1456	1457	1458	1459	1460	1461	1462	1463
2670	1464	1465	1466	1467	1468	1469	1470	1471
2700	1472	1473	1474	1475	1476	1477	1478	1479
2710	1480	1481	1482	1483	1484	1485	1486	1487
2720	1488	1489	1490	1491	1492	1493	1494	1495
2730	1496	1497	1498	1499	1500	1501	1502	1503
2740	1504	1505	1506	1507	1508	1519	1510	1511
2750	1512	1513	1514	1515	1516	1517	1518	1519
2760	1520	1521	1522	1523	1524	1525	1526	1527
2770	1528	1529	1530	1531	1532	1533	1534	1535

2000 to 2777 (Octal) — 1024 to 1535 (Decimal)

Octal	Decimal
10000	4096
20000	8192
30000	12288
40000	16384
50000	20480
60000	24576
70000	28672

	0	1	2	3	4	5	6	7
3000	1536	1537	1538	1539	1540	1541	1542	1543
3010	1544	1545	1546	1547	1548	1549	1550	1551
3020	1552	1553	1554	1555	1556	1557	1558	1559
3030	1560	1561	1562	1563	1564	1565	1566	1567
3040	1568	1569	1570	1571	1572	1573	1574	1575
3050	1576	1577	1578	1579	1580	1581	1582	1583
3060	1584	1585	1586	1587	1588	1589	1590	1591
3070	1592	1593	1594	1595	1596	1597	1598	1599
3100	1600	1601	1602	1603	1604	1605	1606	1607
3110	1608	1609	1610	1611	1612	1613	1614	1615
3120	1616	1617	1618	1619	1620	1621	1622	1623
3130	1624	1625	1626	1627	1628	1629	1630	1631
3140	1632	1633	1634	1635	1636	1637	1638	1639
3150	1640	1641	1642	1643	1644	1645	1646	1647
3160	1648	1649	1650	1651	1652	1653	1654	1655
3170	1656	1657	1658	1659	1660	1661	1662	1663
3200	1664	1665	1666	1667	1668	1669	1670	1671
3210	1672	1673	1674	1675	1676	1677	1678	1679
3220	1680	1681	1682	1683	1684	1685	1686	1687
3230	1688	1689	1690	1691	1692	1693	1694	1695
3240	1696	1697	1698	1699	1700	1701	1702	1703
3250	1704	1705	1706	1707	1708	1709	1710	1711
3260	1712	1713	1714	1715	1716	1717	1718	1719
3270	1720	1721	1722	1723	1724	1725	1726	1727
3300	1728	1729	1730	1731	1732	1733	1734	1735
3310	1736	1737	1738	1739	1740	1741	1742	1743
3320	1744	1745	1746	1747	1748	1749	1750	1751
3330	1752	1753	1754	1755	1756	1757	1758	1759
3340	1760	1761	1762	1763	1764	1765	1766	1767
3350	1768	1769	1770	1771	1772	1773	1774	1775
3360	1776	1777	1778	1779	1780	1781	1782	1783
3370	1784	1785	1786	1787	1788	1789	1790	1791

	0	1	2	3	4	5	6	7
3400	1792	1793	1794	1795	1796	1797	1798	1799
3410	1800	1801	1802	1803	1804	1805	1806	1807
3420	1808	1809	1810	1811	1812	1813	1814	1815
3430	1816	1817	1818	1819	1820	1821	1822	1823
3440	1824	1825	1826	1827	1828	1829	1830	1831
3450	1832	1833	1834	1835	1836	1837	1838	1839
3460	1840	1841	1842	1843	1844	1845	1846	1847
3470	1848	1849	1850	1851	1852	1853	1854	1855
3500	1856	1857	1858	1859	1860	1861	1862	1863
3510	1864	1865	1866	1867	1868	1869	1870	1871
3520	1872	1873	1874	1875	1876	1877	1878	1879
3530	1880	1881	1882	1883	1884	1885	1886	1887
3540	1888	1889	1890	1891	1892	1893	1894	1895
3550	1896	1897	1898	1899	1900	1901	1902	1903
3560	1904	1905	1906	1907	1908	1909	1910	1911
3570	1912	1913	1914	1915	1916	1917	1918	1919
3600	1920	1921	1922	1923	1924	1925	1926	1927
3610	1928	1929	1930	1931	1932	1933	1934	1935
3620	1936	1937	1938	1939	1940	1941	1942	1943
3630	1944	1945	1946	1947	1948	1949	1950	1951
3640	1952	1953	1954	1955	1956	1957	1958	1959
3650	1960	1961	1962	1963	1964	1965	1966	1967
3660	1968	1969	1970	1971	1972	1973	1974	1975
3670	1976	1977	1978	1979	1980	1981	1982	1983
3700	1984	1985	1986	1987	1988	1989	1990	1991
3710	1992	1993	1994	1995	1996	1997	1998	1999
3720	2000	2001	2002	2003	2004	2005	2006	2007
3730	2008	2009	2010	2011	2012	2013	2014	2015
3740	2016	2017	2018	2019	2020	2021	2022	2023
3750	2024	2025	2026	2027	2028	2029	2030	2031
3760	2032	2033	2034	2035	2036	2037	2038	2039
3770	2040	2041	2042	2043	2044	2045	2046	2047

3000 to 3777 (Octal) — 1536 to 2047 (Decimal)

OCTAL-DECIMAL INTEGER CONVERSION TABLE

4000 to 4777 (Octal) — 2048 to 2559 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
4000	2048	2049	2050	2051	2052	2053	2054	2055
4010	2056	2057	2058	2059	2060	2061	2062	2063
4020	2064	2065	2066	2067	2068	2069	2070	2071
4030	2072	2073	2074	2075	2076	2077	2078	2079
4040	2080	2081	2082	2083	2084	2085	2086	2087
4050	2088	2089	2090	2091	2092	2093	2094	2095
4060	2096	2097	2098	2099	2100	2101	2102	2103
4070	2104	2105	2106	2107	2108	2109	2110	2111
4100	2112	2113	2114	2115	2116	2117	2118	2119
4110	2120	2121	2122	2123	2124	2125	2126	2127
4120	2128	2129	2130	2131	2132	2133	2134	2135
4130	2136	2137	2138	2139	2140	2141	2142	2143
4140	2144	2145	2146	2147	2148	2149	2150	2151
4150	2152	2153	2154	2155	2156	2157	2158	2159
4160	2160	2161	2162	2163	2164	2165	2166	2167
4170	2168	2169	2170	2171	2172	2173	2174	2175
4200	2176	2177	2178	2179	2180	2181	2182	2183
4210	2184	2185	2186	2187	2188	2189	2190	2191
4220	2192	2193	2194	2195	2196	2197	2198	2199
4230	2200	2201	2202	2203	2204	2205	2206	2207
4240	2208	2209	2210	2211	2212	2213	2214	2215
4250	2216	2217	2218	2219	2220	2221	2222	2223
4260	2224	2225	2226	2227	2228	2229	2230	2231
4270	2232	2233	2234	2235	2236	2237	2238	2239
4300	2240	2241	2242	2243	2244	2245	2246	2247
4310	2248	2249	2250	2251	2252	2253	2254	2255
4320	2256	2257	2258	2259	2260	2261	2262	2263
4330	2264	2265	2266	2267	2268	2269	2270	2271
4340	2272	2273	2274	2275	2276	2277	2278	2279
4350	2280	2281	2282	2283	2284	2285	2286	2287
4360	2288	2289	2290	2291	2292	2293	2294	2295
4370	2296	2297	2298	2299	2300	2301	2302	2303

	0	1	2	3	4	5	6	7
4400	2304	2305	2306	2307	2308	2309	2310	2311
4410	2312	2313	2314	2315	2316	2317	2318	2319
4420	2320	2321	2322	2323	2324	2325	2326	2327
4430	2328	2329	2330	2331	2332	2333	2334	2335
4440	2336	2337	2338	2339	2340	2341	2342	2343
4450	2344	2345	2346	2347	2348	2349	2350	2351
4460	2352	2353	2354	2355	2356	2357	2358	2359
4470	2360	2361	2362	2363	2364	2365	2366	2367
4500	2368	2369	2370	2371	2372	2373	2374	2375
4510	2376	2377	2378	2379	2380	2381	2382	2383
4520	2384	2385	2386	2387	2388	2389	2390	2391
4530	2392	2393	2394	2395	2396	2397	2398	2399
4540	2400	2401	2402	2403	2404	2405	2406	2407
4550	2408	2409	2410	2411	2412	2413	2414	2415
4560	2416	2417	2418	2419	2420	2421	2422	2423
4570	2424	2425	2426	2427	2428	2429	2430	2431
4600	2432	2433	2434	2435	2436	2437	2438	2439
4610	2440	2441	2442	2443	2444	2445	2446	2447
4620	2448	2449	2450	2451	2452	2453	2454	2455
4630	2456	2457	2458	2459	2460	2461	2462	2463
4640	2464	2465	2466	2467	2468	2469	2470	2471
4650	2472	2473	2474	2475	2476	2477	2478	2479
4660	2480	2481	2482	2483	2484	2485	2486	2487
4670	2488	2489	2490	2491	2492	2493	2494	2495
4700	2496	2497	2498	2499	2500	2501	2502	2503
4710	2504	2505	2506	2507	2508	2509	2510	2511
4720	2512	2513	2514	2515	2516	2517	2518	2519
4730	2520	2521	2522	2523	2524	2525	2526	2527
4740	2528	2529	2530	2531	2532	2533	2534	2535
4750	2536	2537	2538	2539	2540	2541	2542	2543
4760	2544	2545	2546	2547	2548	2549	2550	2551
4770	2552	2553	2554	2555	2556	2557	2558	2559

5000 to 5777 (Octal) — 2560 to 3071 (Decimal)

	0	1	2	3	4	5	6	7
5000	2560	2561	2562	2563	2564	2565	2566	2567
5010	2568	2569	2570	2571	2572	2573	2574	2575
5020	2576	2577	2578	2579	2580	2581	2582	2583
5030	2584	2585	2586	2587	2588	2589	2590	2591
5040	2592	2593	2594	2595	2596	2597	2598	2599
5050	2600	2601	2602	2603	2604	2605	2606	2607
5060	2608	2609	2610	2611	2612	2613	2614	2615
5070	2616	2617	2618	2619	2620	2621	2622	2623
5100	2624	2625	2626	2627	2628	2629	2630	2631
5110	2632	2633	2634	2635	2636	2637	2638	2639
5120	2640	2641	2642	2643	2644	2645	2646	2647
5130	2648	2649	2650	2651	2652	2653	2654	2655
5140	2656	2657	2658	2659	2660	2661	2662	2663
5150	2664	2665	2666	2667	2668	2669	2670	2671
5160	2672	2673	2674	2675	2676	2677	2678	2679
5170	2680	2681	2682	2683	2684	2685	2686	2687
5200	2688	2689	2690	2691	2692	2693	2694	2695
5210	2696	2697	2698	2699	2700	2701	2702	2703
5220	2704	2705	2706	2707	2708	2709	2710	2711
5230	2712	2713	2714	2715	2716	2717	2718	2719
5240	2720	2721	2722	2723	2724	2725	2726	2727
5250	2728	2729	2730	2731	2732	2733	2734	2735
5260	2736	2737	2738	2739	2740	2741	2742	2743
5270	2744	2745	2746	2747	2748	2749	2750	2751
5300	2752	2753	2754	2755	2756	2757	2758	2759
5310	2760	2761	2762	2763	2764	2765	2766	2767
5320	2768	2769	2770	2771	2772	2773	2774	2775
5330	2776	2777	2778	2779	2780	2781	2782	2783
5340	2784	2785	2786	2787	2788	2789	2790	2791
5350	2792	2793	2794	2795	2796	2797	2798	2799
5360	2800	2801	2802	2803	2804	2805	2806	2807
5370	2808	2809	2810	2811	2812	2813	2814	2815

	0	1	2	3	4	5	6	7
5400	2816	2817	2818	2819	2820	2821	2822	2823
5410	2824	2825	2826	2827	2828	2829	2830	2831
5420	2832	2833	2834	2835	2836	2837	2838	2839
5430	2840	2841	2842	2843	2844	2845	2846	2847
5440	2848	2849	2850	2851	2852	2853	2854	2855
5450	2856	2857	2858	2859	2860	2861	2862	2863
5460	2864	2865	2866	2867	2868	2869	2870	2871
5470	2872	2873	2874	2875	2876	2877	2878	2879
5500	2880	2881	2882	2883	2884	2885	2886	2887
5510	2888	2889	2890	2891	2892	2893	2894	2895
5520	2896	2897	2898	2899	2900	2901	2902	2903
5530	2904	2905	2906	2907	2908	2909	2910	2911
5540	2912	2913	2914	2915	2916	2917	2918	2919
5550	2920	2921	2922	2923	2924	2925	2926	2927
5560	2928	2929	2930	2931	2932	2933	2934	2935
5570	2936	2937	2938	2939	2940	2941	2942	2943
5600	2944	2945	2946	2947	2948	2949	2950	2951
5610	2952	2953	2954	2955	2956	2957	2958	2959
5620	2960	2961	2962	2963	2964	2965	2966	2967
5630	2968	2969	2970	2971	2972	2973	2974	2975
5640	2976	2977	2978	2979	2980	2981	2982	2983
5650	2984	2985	2986	2987	2988	2989	2990	2991
5660	2992	2993	2994	2995	2996	2997	2998	2999
5670	3000	3001	3002	3003	3004	3005	3006	3007
5700	3008	3009	3010	3011	3012	3013	3014	3015
5710	3016	3017	3018	3019	3020	3021	3022	3023
5720	3024	3025	3026	3027	3028	3029	3030	3031
5730	3032	3033	3034	3035	3036	3037	3038	3039
5740	3040	3041	3042	3043	3044	3045	3046	3047
5750	3048	3049	3050	3051	3052	3053	3054	3055
5760	3056	3057	3058	3059	3060	3061	3062	3063
5770	3064	3065	3066	3067	3068	3069	3070	3071

OCTAL-DECIMAL INTEGER CONVERSION TABLE

	0	1	2	3	4	5	6	7
6000	3072	3073	3074	3075	3076	3077	3078	3079
6010	3080	3081	3082	3083	3084	3085	3086	3087
6020	3088	3089	3090	3091	3092	3093	3094	3095
6030	3096	3097	3098	3099	3100	3101	3102	3103
6040	3104	3105	3106	3107	3108	3109	3110	3111
6050	3112	3113	3114	3115	3116	3117	3118	3119
6060	3120	3121	3122	3123	3124	3125	3126	3127
6070	3128	3129	3130	3131	3132	3133	3134	3135
6100	3136	3137	3138	3139	3140	3141	3142	3143
6110	3144	3145	3146	3147	3148	3149	3150	3151
6120	3152	3153	3154	3155	3156	3157	3158	3159
6130	3160	3161	3162	3163	3164	3165	3166	3167
6140	3168	3169	3170	3171	3172	3173	3174	3175
6150	3176	3177	3178	3179	3180	3181	3182	3183
6160	3184	3185	3186	3187	3188	3189	3190	3191
6170	3192	3193	3194	3195	3196	3197	3198	3199
6200	3200	3201	3202	3203	3204	3205	3206	3207
6210	3208	3209	3210	3211	3212	3213	3214	3215
6220	3216	3217	3218	3219	3220	3221	3222	3223
6230	3224	3225	3226	3227	3228	3229	3230	3231
6240	3232	3233	3234	3235	3236	3237	3238	3239
6250	3240	3241	3242	3243	3244	3245	3246	3247
6260	3248	3249	3250	3251	3252	3253	3254	3255
6270	3256	3257	3258	3259	3260	3261	3262	3263
6300	3264	3265	3266	3267	3268	3269	3270	3271
6310	3272	3273	3274	3275	3276	3277	3278	3279
6320	3280	3281	3282	3283	3284	3285	3286	3287
6330	3288	3289	3290	3291	3292	3293	3294	3295
6340	3296	3297	3298	3299	3300	3301	3302	3303
6350	3304	3305	3306	3307	3308	3309	3310	3311
6360	3312	3313	3314	3315	3316	3317	3318	3319
6370	3320	3321	3322	3323	3324	3325	3326	3327

	0	1	2	3	4	5	6	7
6400	3328	3329	3330	3331	3332	3333	3334	3335
6410	3336	3337	3338	3339	3340	3341	3342	3343
6420	3344	3345	3346	3347	3348	3349	3350	3351
6430	3352	3353	3354	3355	3356	3357	3358	3359
6440	3360	3361	3362	3363	3364	3365	3366	3367
6450	3368	3369	3370	3371	3372	3373	3374	3375
6460	3376	3377	3378	3379	3380	3381	3382	3383
6470	3384	3385	3386	3387	3388	3389	3390	3391
6500	3392	3393	3394	3395	3396	3397	3398	3399
6510	3400	3401	3402	3403	3404	3405	3406	3407
6520	3408	3409	3410	3411	3412	3413	3414	3415
6530	3416	3417	3418	3419	3420	3421	3422	3423
6540	3424	3425	3426	3427	3428	3429	3430	3431
6550	3432	3433	3434	3435	3436	3437	3438	3439
6560	3440	3441	3442	3443	3444	3445	3446	3447
6570	3448	3449	3450	3451	3452	3453	3454	3455
6600	3456	3457	3458	3459	3460	3461	3462	3463
6610	3464	3465	3466	3467	3468	3469	3470	3471
6620	3472	3473	3474	3475	3476	3477	3478	3479
6630	3480	3481	3482	3483	3484	3485	3486	3487
6640	3488	3489	3490	3491	3492	3493	3494	3495
6650	3496	3497	3498	3499	3500	3501	3502	3503
6660	3504	3505	3506	3507	3508	3509	3510	3511
6670	3512	3513	3514	3515	3516	3517	3518	3519
6700	3520	3521	3522	3523	3524	3525	3526	3527
6710	3528	3529	3530	3531	3532	3533	3534	3535
6720	3536	3537	3538	3539	3540	3541	3542	3543
6730	3544	3545	3546	3547	3548	3549	3550	3551
6740	3552	3553	3554	3555	3556	3557	3558	3559
6750	3560	3561	3562	3563	3564	3565	3566	3567
6760	3568	3569	3570	3571	3572	3573	3574	3575
6770	3576	3577	3578	3579	3580	3581	3582	3583

6000 to 6777 (Octal)
3072 to 3583 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
7000	3584	3585	3586	3587	3588	3589	3590	3591
7010	3592	3593	3594	3595	3496	3497	3598	3599
7020	3600	3601	3602	3603	3604	3605	3606	3607
7030	3608	3609	3610	3611	3612	3613	3614	3615
7040	3616	3617	3618	3619	3620	3621	3622	3623
7050	3624	3625	3626	3627	3628	3629	3630	3631
7060	3632	3633	3634	3635	3636	3637	3638	3639
7070	3640	3641	3642	3643	3644	3645	3646	3647
7100	3648	3649	3650	3651	3652	3653	3654	3655
7110	3656	3657	3658	3659	3660	3661	3662	3663
7120	3664	3665	3666	3667	3668	3669	3670	3671
7130	3672	3673	3674	3675	3676	3677	3678	3679
7140	3680	3681	3682	3683	3684	3685	3686	3687
7150	3688	3689	3690	3691	3692	3693	3694	3695
7160	3696	3697	3698	3699	3700	3701	3702	3703
7170	3704	3705	3706	3707	3708	3709	3710	3711
7200	3712	3713	3714	3715	3716	3717	3718	3719
7210	3720	3721	3722	3723	3724	3725	3726	3727
7220	3728	3729	3730	3731	3732	3733	3734	3735
7230	3736	3737	3738	3739	3740	3741	3742	3743
7240	3744	3745	3746	3747	3748	3749	3750	3751
7250	3752	3753	3754	3755	3756	3757	3758	3759
7260	3760	3761	3762	3763	3764	3765	3766	3767
7270	3768	3769	3770	3771	3772	3773	3774	3775
7300	3776	3777	3778	3779	3780	3781	3782	3783
7310	3784	3785	3786	3787	3788	3789	3790	3791
7320	3792	3793	3794	3795	3796	3797	3798	3799
7330	3800	3801	3802	3803	3804	3805	3806	3807
7340	3808	3809	3810	3811	3812	3813	3814	3815
7350	3816	3817	3818	3819	3820	3821	3822	3823
7360	3824	3825	3826	3827	3828	3829	3830	3831
7370	3832	3833	3834	3835	3836	3837	3838	3839

	0	1	2	3	4	5	6	7
7400	3840	3841	3842	3843	3844	3845	3846	3847
7410	3848	3849	3850	3851	3852	3853	3854	3855
7420	3856	3857	3858	3859	3860	3861	3862	3863
7430	3864	3865	3866	3867	3868	3869	3870	3871
7440	3872	3873	3874	3875	3876	3877	3878	3879
7450	3880	3881	3882	3883	3884	3885	3886	3887
7460	3888	3889	3890	3891	3892	3893	3894	3895
7470	3896	3897	3898	3899	3900	3901	3902	3903
7500	3904	3905	3906	3907	3908	3909	3910	3911
7510	3912	3913	3914	3915	3916	3917	3918	3919
7520	3920	3921	3922	3923	3924	3925	3926	3927
7530	3928	3929	3930	3931	3932	3933	3934	3935
7540	3936	3937	3938	3939	3940	3941	3942	3943
7550	3944	3945	3946	3947	3948	3949	3950	3951
7560	3952	3953	3954	3955	3956	3957	3958	3959
7570	3960	3961	3962	3963	3964	3965	3966	3967
7600	3968	3969	3970	3971	3972	3973	3974	3975
7610	3976	3977	3978	3979	3980	3981	3982	3983
7620	3984	3985	3986	3987	3988	3989	3990	3991
7630	3992	3993	3994	3995	3996	3997	3998	3999
7640	4000	4001	4002	4003	4004	4005	4006	4007
7650	4008	4009	4010	4011	4012	4013	4014	4015
7660	4016	4017	4018	4019	4020	4021	4022	4023
7670	4024	4025	4026	4027	4028	4029	4030	4031
7700	4032	4033	4034	4035	4036	4037	4038	4039
7710	4040	4041	4042	4043	4044	4045	4046	4047
7720	4048	4049	4050	4051	4052	4053	4054	4055
7730	4056	4057	4058	4059	4060	4061	4062	4063
7740	4064	4065	4066	4067	4068	4069	4070	4071
7750	4072	4073	4074	4075	4076	4077	4078	4079
7760	4080	4081	4082	4083	4084	4085	4086	4087
7770	4088	4089	4090	4091	4092	4093	4094	4095

7000 to 7777 (Octal)
3584 to 4095 (Decimal)

APPENDIX L

OCTAL-DECIMAL FRACTION CONVERSION TABLE

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000	.000000	.100	.125000	.200	.250000	.300	.375000
.001	.001953	.101	.126953	.201	.251953	.301	.376953
.002	.003906	.102	.128906	.202	.253906	.302	.378906
.003	.005859	.103	.130859	.203	.255859	.303	.380859
.004	.007812	.104	.132812	.204	.257812	.304	.382812
.005	.009765	.105	.134765	.205	.259765	.305	.384765
.006	.011718	.106	.136718	.206	.261718	.306	.386718
.007	.013671	.107	.138671	.207	.263671	.307	.388671
.010	.015625	.110	.140625	.210	.265625	.310	.390625
.011	.017578	.111	.142578	.211	.267578	.311	.392578
.012	.019531	.112	.144531	.212	.269531	.312	.394531
.013	.021484	.113	.146484	.213	.271484	.313	.396484
.014	.023437	.114	.148437	.214	.273437	.314	.398437
.015	.025390	.115	.150390	.215	.275390	.315	.400390
.016	.027343	.116	.152343	.216	.277343	.316	.402343
.017	.029296	.117	.154296	.217	.279296	.317	.404296
.020	.031250	.120	.156250	.220	.281250	.320	.406250
.021	.033203	.121	.158203	.221	.283203	.321	.408203
.022	.035156	.122	.160156	.222	.285156	.322	.410156
.023	.037109	.123	.162109	.223	.287109	.323	.412109
.024	.039062	.124	.164062	.224	.289062	.324	.414062
.025	.041015	.125	.166015	.225	.291015	.325	.416015
.026	.042968	.126	.167968	.226	.292968	.326	.417968
.027	.044921	.127	.169921	.227	.294921	.327	.419921
.030	.046875	.130	.171875	.230	.296875	.330	.421875
.031	.048828	.131	.173828	.231	.298828	.331	.423828
.032	.050781	.132	.175781	.232	.300781	.332	.425781
.033	.052734	.133	.177734	.233	.302734	.333	.427734
.034	.054687	.134	.179687	.234	.304687	.334	.429687
.035	.056640	.135	.181640	.235	.306640	.335	.431640
.036	.058593	.136	.183593	.236	.308593	.336	.433593
.037	.060546	.137	.185546	.237	.310546	.337	.435546
.040	.062500	.140	.187500	.240	.312500	.340	.437500
.041	.064453	.141	.189453	.241	.314453	.341	.439453
.042	.066406	.142	.191406	.242	.316406	.342	.441406
.043	.068359	.143	.193359	.243	.318359	.343	.443359
.044	.070312	.144	.195312	.244	.320312	.344	.445312
.045	.072265	.145	.197265	.245	.322265	.345	.447265
.046	.074218	.146	.199218	.246	.324218	.346	.449218
.047	.076171	.147	.201171	.247	.326171	.347	.451171
.050	.078125	.150	.203125	.250	.328125	.350	.453125
.051	.080078	.151	.205078	.251	.330078	.351	.455078
.052	.082031	.152	.207031	.252	.332031	.352	.457031
.053	.083984	.153	.208984	.253	.333984	.353	.458984
.054	.085937	.154	.210937	.254	.335937	.354	.460937
.055	.087890	.155	.212890	.255	.337890	.355	.462890
.056	.089843	.156	.214843	.256	.339843	.356	.464843
.057	.091796	.157	.216796	.257	.341796	.357	.466796
.060	.093750	.160	.218750	.260	.343750	.360	.468750
.061	.095703	.161	.220703	.261	.345703	.361	.470703
.062	.097656	.162	.222656	.262	.347656	.362	.472656
.063	.099609	.163	.224609	.263	.349609	.363	.474609
.064	.101562	.164	.226562	.264	.351562	.364	.476562
.065	.103515	.165	.228515	.265	.353515	.365	.478515
.066	.105468	.166	.230468	.266	.355468	.366	.480468
.067	.107421	.167	.232421	.267	.357421	.367	.482421
.070	.109375	.170	.234375	.270	.359375	.370	.484375
.071	.111328	.171	.236328	.271	.361328	.371	.486328
.072	.113281	.172	.238281	.272	.363281	.372	.488281
.073	.115234	.173	.240234	.273	.365234	.373	.490234
.074	.117187	.174	.242187	.274	.367187	.374	.492187
.075	.119140	.175	.244140	.275	.369140	.375	.494140
.076	.121093	.176	.246093	.276	.371093	.376	.496093
.077	.123046	.177	.248046	.277	.373046	.377	.498046

OCTAL-DECIMAL FRACTION CONVERSION TABLE

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000000	.000000	.000100	.000244	.000200	.000488	.000300	.000732
.000001	.000003	.000101	.000247	.000201	.000492	.000301	.000736
.000002	.000007	.000102	.000251	.000202	.000495	.000302	.000740
.000003	.000011	.000103	.000255	.000203	.000499	.000303	.000743
.000004	.000015	.000104	.000259	.000204	.000503	.000304	.000747
.000005	.000019	.000105	.000263	.000205	.000507	.000305	.000751
.000006	.000022	.000106	.000267	.000206	.000511	.000306	.000755
.000007	.000026	.000107	.000270	.000207	.000514	.000307	.000759
.000010	.000030	.000110	.000274	.000210	.000518	.000310	.000762
.000011	.000034	.000111	.000278	.000211	.000522	.000311	.000766
.000012	.000038	.000112	.000282	.000212	.000526	.000312	.000770
.000013	.000041	.000113	.000286	.000213	.000530	.000313	.000774
.000014	.000045	.000114	.000289	.000214	.000534	.000314	.000778
.000015	.000049	.000115	.000293	.000215	.000537	.000315	.000782
.000016	.000053	.000116	.000297	.000216	.000541	.000316	.000785
.000017	.000057	.000117	.000301	.000217	.000545	.000317	.000789
.000020	.000061	.000120	.000305	.000220	.000549	.000320	.000793
.000021	.000064	.000121	.000308	.000221	.000553	.000321	.000797
.000022	.000068	.000122	.000312	.000222	.000556	.000322	.000801
.000023	.000072	.000123	.000316	.000223	.000560	.000323	.000805
.000024	.000076	.000124	.000320	.000224	.000564	.000324	.000808
.000025	.000080	.000125	.000324	.000225	.000568	.000325	.000812
.000026	.000083	.000126	.000328	.000226	.000572	.000326	.000816
.000027	.000087	.000127	.000331	.000227	.000576	.000327	.000820
.000030	.000091	.000130	.000335	.000230	.000579	.000330	.000823
.000031	.000095	.000131	.000339	.000231	.000583	.000331	.000827
.000032	.000099	.000132	.000343	.000232	.000587	.000332	.000831
.000033	.000102	.000133	.000347	.000233	.000591	.000333	.000835
.000034	.000106	.000134	.000350	.000234	.000595	.000334	.000839
.000035	.000110	.000135	.000354	.000235	.000598	.000335	.000843
.000036	.000114	.000136	.000358	.000236	.000602	.000336	.000846
.000037	.000118	.000137	.000362	.000237	.000606	.000337	.000850
.000040	.000122	.000140	.000366	.000240	.000610	.000340	.000854
.000041	.000125	.000141	.000370	.000241	.000614	.000341	.000858
.000042	.000129	.000142	.000373	.000242	.000617	.000342	.000862
.000043	.000133	.000143	.000377	.000243	.000621	.000343	.000865
.000044	.000137	.000144	.000381	.000244	.000625	.000344	.000869
.000045	.000141	.000145	.000385	.000245	.000629	.000345	.000873
.000046	.000144	.000146	.000389	.000246	.000633	.000346	.000877
.000047	.000148	.000147	.000392	.000247	.000637	.000347	.000881
.000050	.000152	.000150	.000396	.000250	.000640	.000350	.000885
.000051	.000156	.000151	.000400	.000251	.000644	.000351	.000888
.000052	.000160	.000152	.000404	.000252	.000648	.000352	.000892
.000053	.000164	.000153	.000408	.000253	.000652	.000353	.000896
.000054	.000167	.000154	.000411	.000254	.000656	.000354	.000900
.000055	.000171	.000155	.000415	.000255	.000659	.000355	.000904
.000056	.000175	.000156	.000419	.000256	.000663	.000356	.000907
.000057	.000179	.000157	.000423	.000257	.000667	.000357	.000911
.000060	.000183	.000160	.000427	.000260	.000671	.000360	.000915
.000061	.000186	.000161	.000431	.000261	.000675	.000361	.000919
.000062	.000190	.000162	.000434	.000262	.000679	.000362	.000923
.000063	.000194	.000163	.000438	.000263	.000682	.000363	.000926
.000064	.000198	.000164	.000442	.000264	.000686	.000364	.000930
.000065	.000202	.000165	.000446	.000265	.000690	.000365	.000934
.000066	.000205	.000166	.000450	.000266	.000694	.000366	.000938
.000067	.000209	.000167	.000453	.000267	.000698	.000367	.000942
.000070	.000213	.000170	.000457	.000270	.000701	.000370	.000946
.000071	.000217	.000171	.000461	.000271	.000705	.000371	.000949
.000072	.000221	.000172	.000465	.000272	.000709	.000372	.000953
.000073	.000225	.000173	.000469	.000273	.000713	.000373	.000957
.000074	.000228	.000174	.000473	.000274	.000717	.000374	.000961
.000075	.000232	.000175	.000476	.000275	.000720	.000375	.000965
.000076	.000236	.000176	.000480	.000276	.000724	.000376	.000968
.000077	.000240	.000177	.000484	.000277	.000728	.000377	.000972

OCTAL-DECIMAL FRACTION CONVERSION TABLE

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000400	.000976	.000500	.001220	.000600	.001464	.000700	.001708
.000401	.000980	.000501	.001224	.000601	.001468	.000701	.001712
.000402	.000984	.000502	.001228	.000602	.001472	.000702	.001716
.000403	.000988	.000503	.001232	.000603	.001476	.000703	.001720
.000404	.000991	.000504	.001235	.000604	.001480	.000704	.001724
.000405	.000995	.000505	.001239	.000605	.001483	.000705	.001728
.000406	.000999	.000506	.001243	.000606	.001487	.000706	.001731
.000407	.001003	.000507	.001247	.000607	.001491	.000707	.001735
.000410	.001007	.000510	.001251	.000610	.001495	.000710	.001739
.000411	.001010	.000511	.001255	.000611	.001499	.000711	.001743
.000412	.001014	.000512	.001258	.000612	.001502	.000712	.001747
.000413	.001018	.000513	.001262	.000613	.001506	.000713	.001750
.000414	.001022	.000514	.001266	.000614	.001510	.000714	.001754
.000415	.001026	.000515	.001270	.000615	.001514	.000715	.001758
.000416	.001029	.000516	.001274	.000616	.001518	.000716	.001762
.000417	.001033	.000517	.001277	.000617	.001522	.000717	.001766
.000420	.001037	.000520	.001281	.000620	.001525	.000720	.001770
.000421	.001041	.000521	.001285	.000621	.001529	.000721	.001773
.000422	.001045	.000522	.001289	.000622	.001533	.000722	.001777
.000423	.001049	.000523	.001293	.000623	.001537	.000723	.001781
.000424	.001052	.000524	.001296	.000624	.001541	.000724	.001785
.000425	.001056	.000525	.001300	.000625	.001544	.000725	.001789
.000426	.001060	.000526	.001304	.000626	.001548	.000726	.001792
.000427	.001064	.000527	.001308	.000627	.001552	.000727	.001796
.000430	.001068	.000530	.001312	.000630	.001556	.000730	.001800
.000431	.001071	.000531	.001316	.000631	.001560	.000731	.001804
.000432	.001075	.000532	.001319	.000632	.001564	.000732	.001808
.000433	.001079	.000533	.001323	.000633	.001567	.000733	.001811
.000434	.001083	.000534	.001327	.000634	.001571	.000734	.001815
.000435	.001087	.000535	.001331	.000635	.001575	.000735	.001819
.000436	.001091	.000536	.001335	.000636	.001579	.000736	.001823
.000437	.001094	.000537	.001338	.000637	.001583	.000737	.001827
.000440	.001098	.000540	.001342	.000640	.001586	.000740	.001831
.000441	.001102	.000541	.001346	.000641	.001590	.000741	.001834
.000442	.001106	.000542	.001350	.000642	.001594	.000742	.001838
.000443	.001110	.000543	.001354	.000643	.001598	.000743	.001842
.000444	.001113	.000544	.001358	.000644	.001602	.000744	.001846
.000445	.001117	.000545	.001361	.000645	.001605	.000745	.001850
.000446	.001121	.000546	.001365	.000646	.001609	.000746	.001853
.000447	.001125	.000547	.001369	.000647	.001613	.000747	.001857
.000450	.001129	.000550	.001373	.000650	.001617	.000750	.001861
.000451	.001132	.000551	.001377	.000651	.001621	.000751	.001865
.000452	.001136	.000552	.001380	.000652	.001625	.000752	.001869
.000453	.001140	.000553	.001384	.000653	.001628	.000753	.001873
.000454	.001144	.000554	.001388	.000654	.001632	.000754	.001876
.000455	.001148	.000555	.001392	.000655	.001636	.000755	.001880
.000456	.001152	.000556	.001396	.000656	.001640	.000756	.001884
.000457	.001155	.000557	.001399	.000657	.001644	.000757	.001888
.000460	.001159	.000560	.001403	.000660	.001647	.000760	.001892
.000461	.001163	.000561	.001407	.000661	.001651	.000761	.001895
.000462	.001167	.000562	.001411	.000662	.001655	.000762	.001899
.000463	.001171	.000563	.001415	.000663	.001659	.000763	.001903
.000464	.001174	.000564	.001419	.000664	.001663	.000764	.001907
.000465	.001178	.000565	.001422	.000665	.001667	.000765	.001911
.000466	.001182	.000566	.001426	.000666	.001670	.000766	.001914
.000467	.001186	.000567	.001430	.000667	.001674	.000767	.001918
.000470	.001190	.000570	.001434	.000670	.001678	.000770	.001922
.000471	.001194	.000571	.001438	.000671	.001682	.000771	.001926
.000472	.001197	.000572	.001441	.000672	.001686	.000772	.001930
.000473	.001201	.000573	.001445	.000673	.001689	.000773	.001934
.000474	.001205	.000574	.001449	.000674	.001693	.000774	.001937
.000475	.001209	.000575	.001453	.000675	.001697	.000775	.001941
.000476	.001213	.000576	.001457	.000676	.001701	.000776	.001945
.000477	.001216	.000577	.001461	.000677	.001705	.000777	.001949

CUT ALONG LINE

COMMENT SHEET

CONTROL DATA

160G Programming-Reference Manual Publication No. G02000c

FROM NAME : ______________________________

BUSINESS ADDRESS : ______________________________

COMMENTS: (DESCRIBE ERRORS, SUGGESTED ADDITION OR DELETION AND INCLUDE PAGE NUMBER, ETC.)

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE STAPLE

FOLD FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
8100 34TH AVENUE SOUTH
MINNEAPOLIS 20, MINNESOTA

ATTN: TECHNICAL PUBLICATIONS DEPT.
GOVERNMENT SYSTEMS DIVISION
PLANT 1G

CUT ALONG LINE

FOLD FOLD

STAPLE STAPLE