CONTROL DATA

160G

MASS REFERENCE MANUAL

CONTROL DATA

160G

MASS REFERENCE MANUAL

G02075a
15 April 1964

RECORD OF CHANGE NOTICES

C. N. NO.	DATE ORIGINATED	DATE ENTERED	INITIALS	REMARKS

CONTENTS

ILLUSTRATIONS

TABLES

CHAPTER 1
DESCRIPTION OF MASS

The 160G monitor and assembly system (MASS) is an integrated series of programs and operating procedures which provide for easy use and integration of previously written library and systems programs into a programmer's own operations on the CONTROL DATA® 160G computer system.

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

It is assumed that the reader has an understanding of the 160G and GASS instructions. The 160G Programming Manual, Control Data publication G02000, describes the 160G instruction repertoire. The GASS Manual, Control Data publication G01676, contains the GASS pseudo instructions.

DESCRIPTION

The standard I/O routines provide the programmer with a set of input-output routines which take advantage of the buffering capabilities of the 160G computer. A logical unit number indicates the choice of I/O devices. The monitor master I/O routines decode the logical unit number and transfer control to the required specialized I/O routine for a particular device. A decoding table controls the assignment of unit numbers to I/O devices, 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 by substitution of a proper monitor control.

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

The operator (EXECUTIVE) control portion of MASS accepts commands from a standard source (normally a typewriter). Under this control it loads and executes programs and subroutines as desired.

The MASS system is designed to operate on a minimum configuration consisting of a 160G Compute Module, four magnetic tape drives, and an on-line typewriter. The inclusion of additional peripheral equipment increases the system efficiency.

A magnetic tape unit, called the system library tape, contains the MASS system. The first record on the library tape is the auto-load routine which is used to bring the MASS system into the machine.

OPERATION

The sequence of events in the MASS operation is as follows:

1. Depressing the LOAD and START pushbuttons on the Console loads the auto-load operation into memory to start the processing. The action reads the first binary record on the library tape into memory starting at location 00000 in bank 0. It then turns program control over to the auto-load program contained on this record.

2. The auto-load program reads the resident routine into lower core memory and turns program control over to the EXECUTIVE control program.

3. EXECUTIVE then requests commands from the typewriter.

4. The operator types commands to EXECUTIVE. These commands cause EXECUTIVE to call in some program (if it is not in memory) and execute the program.

5. On completion of a program, control returns to EXECUTIVE, which requests additional control information from the operator.

PROGRAMMING CONVENTIONS

All programs written by Control Data for incorporation in 160G software, follow these programming conventions and restrictions.

MEMORY RESTRICTIONS

Index locations 00056 to 00063 (octal locations 00070 to 00077) may be used by all routines as local working storage. To this extent, the programmer should not expect these locations to be unchanged after performing a return jump to some other routine.

The upper portion of bank 0, locations 17400 to 17776 should not be used in any program since this area is reserved for service routines, loaders, dumps, and so forth.

RELOCATION RESTRICTIONS

The relocating binary loader relocates a program within an 8192 bank of memory with no restrictions. Programs which may be relocated in other banks of memory should be written as being in bank 0. It is the programmer's responsibility to determine which bank the program is in at running time, and to set all bank setting instructions correspondingly. Subroutines written for inclusion in a program under LIBA should not change bank control settings.

References to routines and data in other banks under monitor control should be made in the entire memory mode to guarantee correct linking under the loader.

SUBROUTINE CONVENTIONS

Normally a subroutine should return to the main program with all bank settings the same as on the entry. A JPRG instruction is the standard entrance to a subroutine. A SRJP instruction, which is at the entrance location, is the standard exit from a subroutine, after restoring all bank controls.

The following rules should be used for arranging parameters:

1. The A register normally carries a single parameter to the subroutine.

2. The Q register carries the bank number in the lower five bits, with the address of the location in the bank in the A register, if the single parameter is a memory address.

3. More than one parameter to a subroutine is stored in the locations following the JPRG instruction, and the A register is not used.

4. If a parameter is an address including the bank number and is one of a list of parameters following the JPRG instruction, the bank number is in the word preceding the address word.

INSTRUCTION FORMAT

MASS provides sequencing and control over previously written or stored programs. The instructions used to exercise these operations are represented as alphanumeric symbols. These symbols can be used as manual (from the typewriter), magnetic tape, or punched card inputs to MASS. The MASS instruction format for input to magnetic tape or punched cards is the same as for GASS. The GASS instruction format is given in Chapter 2 of the GASS manual. The individual MASS instructions and their descriptions are contained in the following pages.

LIBRARY TAPE FORMAT

The library tape, which contains MASS and GASS, is a general-purpose tape. As such it may contain:

1. Symbolic programs for inclusion under LIBA.
2. Binary card image of programs
3. Macro-instruction definitions
4. Binary core dumps
5. Other types of information as necessary

For efficiency, it is desirable to keep the library tape in the highest density form. The maximum density available varies from installation

to installation. The tape includes provisions that indicate the density of the recording.

The library tape format is as follows:

1. Auto-load routine (one record)
2. End-of-file mark
3. MASS loader routine
4. End-of-file mark
5. MASS routines
6. End-of-file mark
7. Body of library, an end-of-file mark follows each segment
8. End-of-library tape sentinel
9. Two end-of-file marks

BODY OF LIBRARY

The body of the library tape consists of several files separated by end-of-file marks. The files can be any one of four types.

The first record of a file may be thought of as a GASS program card. Columns 10 through 14 classify the information and format of the data to follow. The information in the remarks field is used for recording purposes and should have the date of writing of the routine and the name of the programmer, plus any other information desired.

The format of the four different types of files is as follows.

SYMBOLIC PROGRAM

The first record consists of the symbol IDENT in columns 10 through 14 and the name of the record in columns 16 through 23. Separate records of 80 characters each carry the body of the file in BCD code.

MACRO DEFINITION

The name of the file appears in the first record in columns 1 through 8. Columns 10 through 14 contain the symbol MACRO which describes this file. Separate records of 80 characters each carry the body of the file in BCD code.

BINARY CARD PROGRAM

The symbol CARD, in columns 10 through 14 of the first record, describes the binary card program. The body of the segment is binary card images. Each record is 160 characters or 80 words. An end-of-file mark terminates the record.

BINARY CORE DUMP

The name of the file appears in columns 1 through 8 of the first record. The symbol CORE, in columns 10 through 14, describes this record. The body of the segment follows the binary core dump format. An end-of-file mark terminates the record.

The first record of each segment is recorded in BCD at 200 bits per inch density. The record is 80 or 120 characters long. Column 9 of this record indicates the density of the following recordings in the segment. A blank indicates 200 bits per inch, 1 indicates 556 bits per inch, and 2 indicates 800 bits per inch.

END-OF-LIBRARY SENTINEL

The end-of-library sentinel is a one card record symbolic program. This record indicates the last file on the library tape. This IDENT record should not be used except to indicate the last file on a library tape. The format for an end-of-library sentinel is as follows:

```
COLUMN    1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

                                  I D E N T    E O L B
```

LIBRARY LISTER

A library listing program which includes a summary listing prints the first record (or control record) of each segment and also a detailed lister which provides information on the binary card programs, including entry points, external symbols, starting addresses, and length of programs.

CARD FORMATS

The three types of cards acceptable to MASS are as follows:

1. MASS control cards
2. Hollerith loader cards
3. Binary loader cards

MASS CONTROL CARDS

These control cards also apply to magnetic tape control records and typewriter control records.

MASTER CONTROL CARD

A master control card calls a routine or capability from MASS which is always available in the computer as long as the system is operating under MASS control. The format is the same as for a subsidiary control card.

SUBSIDIARY CONTROL CARD

A subsidiary control card consists of an entry point name starting in column 1. Commas separate the parameters required by the sub-routine from the entry point name. A period indicates the end of the subsidiary control card.

The reading of a subsidiary control card transfers control to the routine or portion of a routine named in the entry point name. The routine then accepts the parameters provided, and executes. If the entry point name specified does not exist in the 160G, the typewriter prints a message. To use the routine desired, it must first exist in the memory of the 160G. This is performed by a CALL master control card.

HOLLERITH LOADER CARDS

The punch in column 1 classifies the Hollerith loader card. The class and format are given as follows:

CLASS 0 OCTAL

A 0 punch in column 1 indicates an octal data card. All digits must be between 0 and 7. Columns 2 and 3 contain a two-digit bank designation. Columns 4 through 7 contain a four-digit location

designation. Words to be located are punched in five-column groups. The columns for words are 8 through 12, 13 through 17, 18 through 22, 23 through 27, 28 through 32, and 33 through 37.

If any column of a group is blank, that word is not changed in memory. Words are stored starting at the address indicated in columns 2 through 7 and continuing to the end of the card.

CLASS 1 DECIMAL

A 1 punch in column 1 indicates a decimal data card. All digits must be between 0 and 9. Columns 2 and 3 contain a two-octal digit bank designation. Columns 4 through 7 contain a four-octal digit location designation. Words to be loaded are punched in five-column groups as in the octal data card. The first column of a group is the sign digit. A blank is positive and a minus punch (-1) indicates a negative number. The remaining four columns are decimal numbers 0 through 8191. Within the four-column field, blanks are ignored. If there is at least one digit, the number is converted from decimal to binary and justified to the right. If all five columns of a group are blank, that word is not changed in memory. Words are stored starting at the address indicated in columns 2 through 7 and continuing to the end of the card.

BINARY LOADER CARDS

A 7 punch and a 9 punch in column 1 indicate binary loader cards. A 12 punch in column 1 indicates that the card was prepared by GASS and that the card contains a 12-bit checksum in column 4. Columns 2 and 3 contain a bank and location address (if used by the card format). A punch in row 3 of column 2 indicates that the address is fixed and cannot be affected by a relocation constant.

Bits in rows 0 through 6 of column 1 are a word count. This is a card type classification and also a count of the number of computer words represented on a binary program card.

The checksum process used in binary cards is a selective complement of the 80 columns of the card. The checksum word is chosen such that the selective complement of the entire card is zero. Thus, there is an even number of punches in each row of the card.

BINARY TRANSFER CARD (WC = 0)

This card, Figure 1-1, specifies a transfer of control to the bank and address given in columns 2 and 3. The relocatable binary loader

routine causes a transfer to take place to this address if the address is other than 0. The relocation constant increments the actual address if relocation was specified and the transfer is to a relocated location.

The binary transfer card is used only as an end of program identification in the linking relocatable loader. A symbolic transfer card initiates program control from the linking loader.

RELOCATABLE BINARY PROGRAM CARD (WC = 1 to 100_8)

This card, Figure 1-2, contains the binary information on a program. Columns 17 through 80 contain the lower 12 bits of each word of data. The higher-order bit of each word is packed in the area of columns 11 through 16. Columns 5 through 10 contain a relocation bit, indicating that it is possible to add the relocation constant to a word.

SYMBOLIC TRANSFER CARD (WC = 101_8)

Columns 5 through 8 contain an eight-character symbolic transfer address. The binary loader ignores this card, Figure 1-3, but the linking binary loader uses it to establish a transfer to the given symbolic address. An entry point of one of the loaded programs defines this symbolic address.

PROGRAM IDENTIFICATION CARD (WC = 102_8)

The program identification card, Figure 1-4, contains information which is required to place a program on the library tape in binary form and also to locate the card on the binary file.

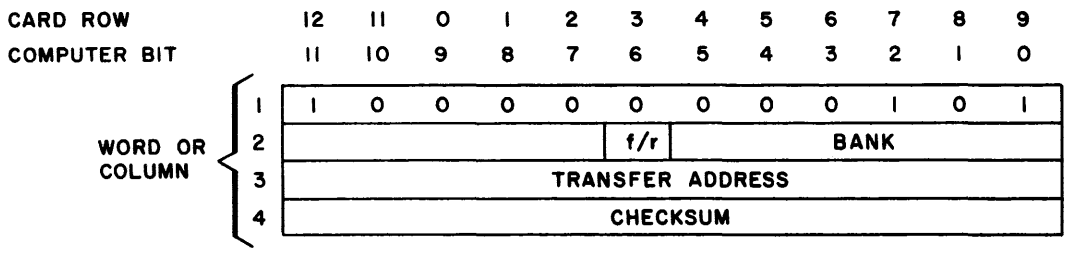


Figure 1-1. Binary Transfer Card (WC = 0)

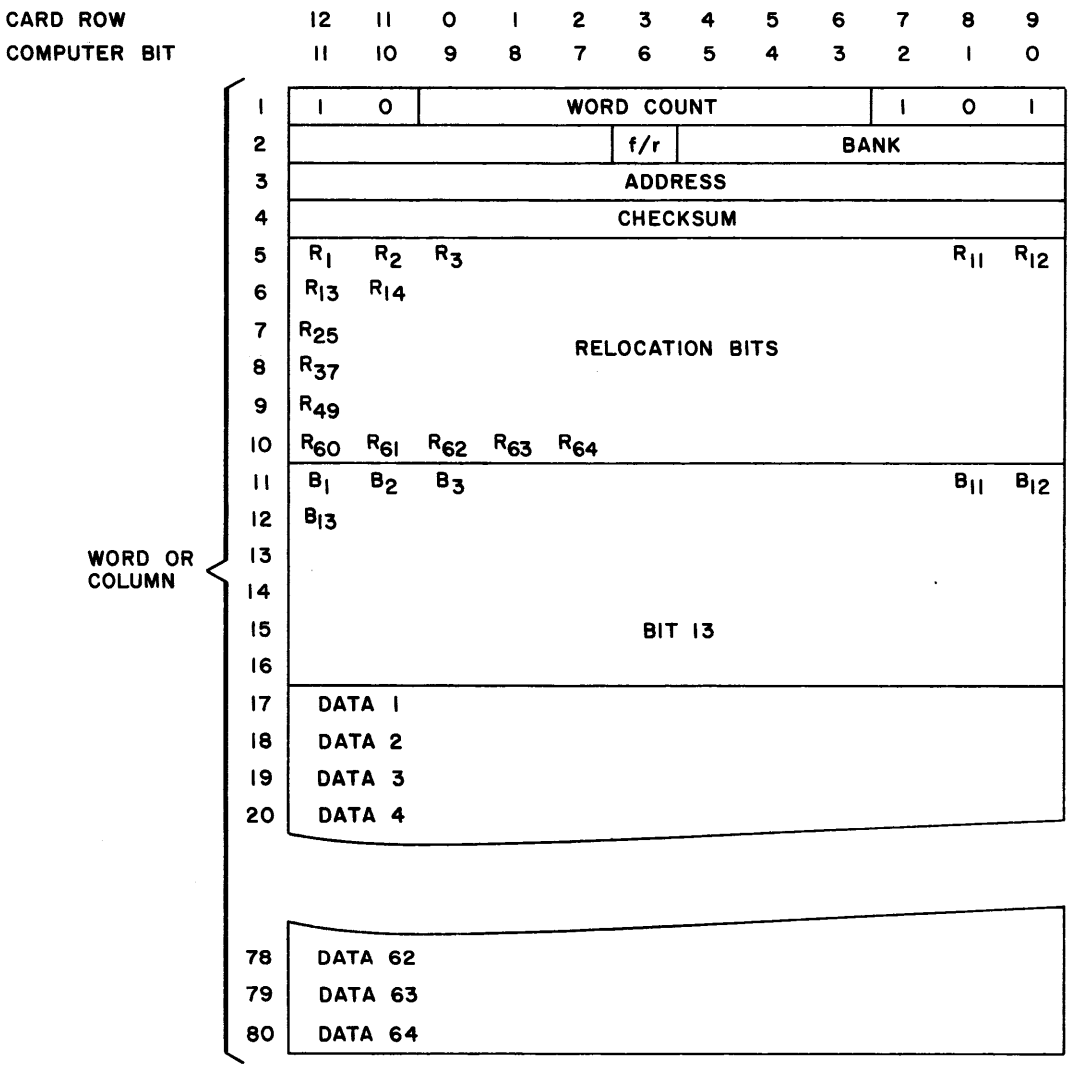
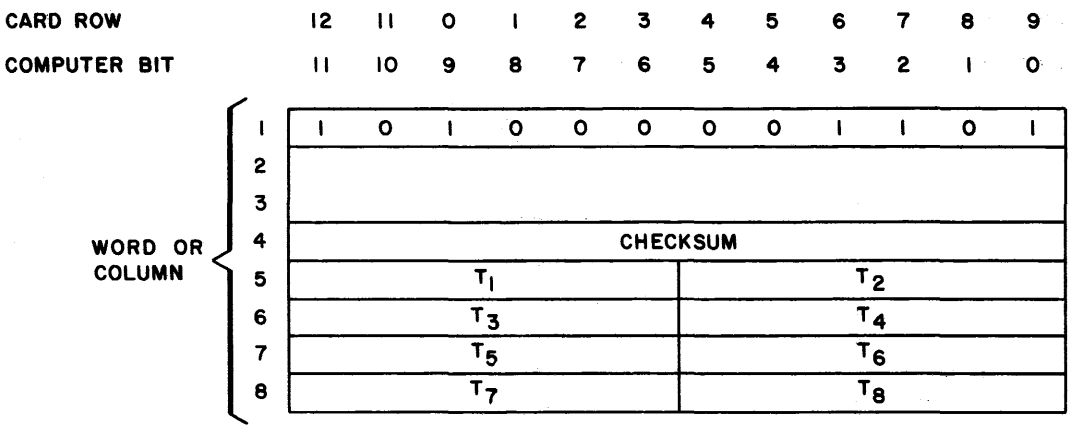


Figure 1-2. Relocatable Binary Program Card (WC = 1 to 100_8)



NOTES:

1. T_1 TO T_8 ARE BCD CODE

   0 = 12_4

   1 = 01_8

   ETC.

   A = 61_8

   BLANK = 20_8

2. CAUSE TRANSFER TO SYMBOLIC LOCATION

   T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8

Figure 1-3. Symbolic Transfer Card (WC = 101_8)

CARD ROW		12	11	0	1	2	3	4	5	6	7	8	9
COMPUTER BIT		11	10	9	8	7	6	5	4	3	2	1	0
WORD OR COLUMN	1	1	0	1	0	0	0	0	1	0	1	0	1
	2							BANK					
	3	FIRST WORD ADDRESS											
	4	CHECKSUM											
	5	I_1						I_2					
	6	I_3						I_4					
	7	I_5						I_6					
	8	I_7						I_8					
	9							BANK					
	10	LAST WORD ADDRESS + 1											
	11	I/O 1						I/O 2					
	12	I/O 3						I/O 4					
	13	I/O 5						I/O 6					
	14	I/O 7						I/O 8					
	15	ETC.											

NOTES:

1. PROGRAM IDENTIFICATION I_1 TO I_8 IS IN BCD CODE.
2. STANDARD I/O UNIT NUMBERS USED IN THE PROGRAM APPEAR AS BINARY INFORMATION STARTING IN COLUMN 11. UNIT NUMBER 0 TERMINATES THE I/O LIST.

Figure 1-4. Program Identification Card (WC = 102_8)

ENTRY POINT TABLE CARD (WC = 103_8)

The entry point table card, Figure 1-5, has the information generated by GASS from the ENTRY pseudo instruction. The linking relocatable loader uses this information to complete the program linkage before turning execution over to the object program.

EXTERNAL SYMBOL TABLE CARD (WC = 104_8)

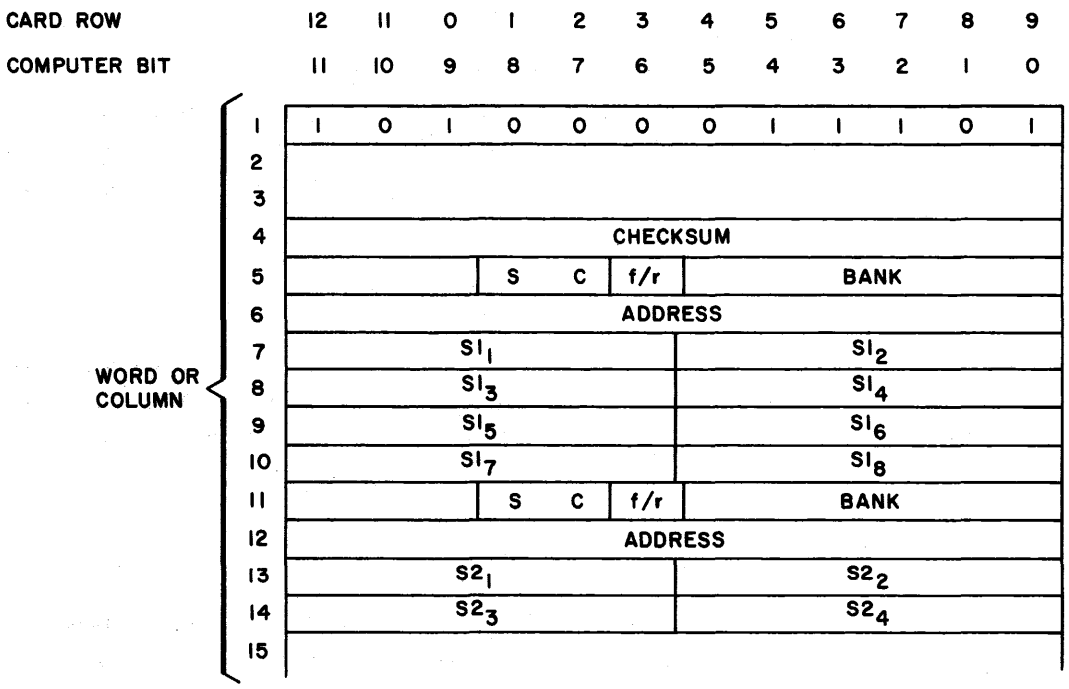
The external symbol table card, Figure 1-6, has the information generated by GASS from the EXTNL pseudo instruction. The linking relocatable loader uses this information to complete the program linkage before turning execution over to the object program.

LIBRARY CALL CARD (WC = 105_8)

GASS produces the library call card, Figure 1-7, from the LIBS pseudo instruction and provides information to MASS as to what routines to call from the library tape for inclusion in the object program.

SYMBOLIC COMMAND CARD (WC = 120_8)

The S card covers a number of symbolic instructions which may generate the operation. The S card contains a punch in column 1, rows 0, 2, 7, and 9. This may be made by punching an S and a 7 and 9. Columns from 2 on contain symbolic instructions to the loader in normal card code, as opposed to the packed BCD code used on other binary cards. The S cards use no checksum.



NOTES:

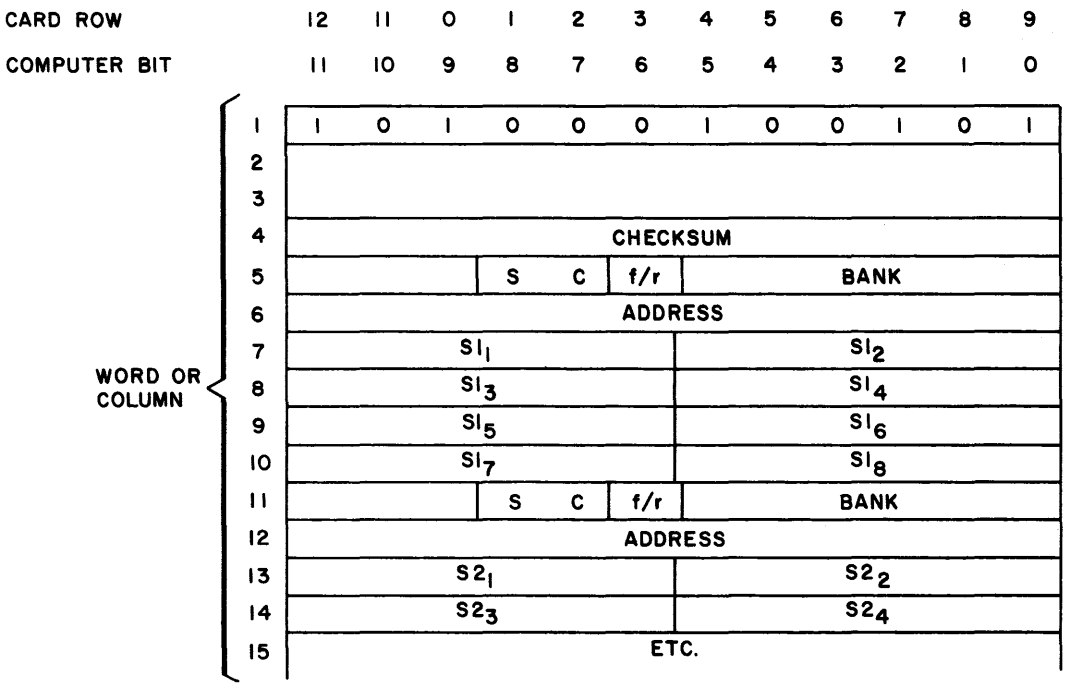
1. SC is a count of the number of words required to hold the symbol.

   Example: JAMES requires three words to hold, thus, SC = 3.

   f/r indicates by a 1 that the symbol is fixed and no increment is added to the address. 0 indicates a relocatable symbol, and the relocation increment must be added to the given address.

2. Symbols are carried in BCD code. A SC=0 indicates the end of symbols in a card. Space required for a symbol depends on the number of characters in the symbol.

Figure 1-5. Entry Point Table Card (WC = 103_8)



NOTES:

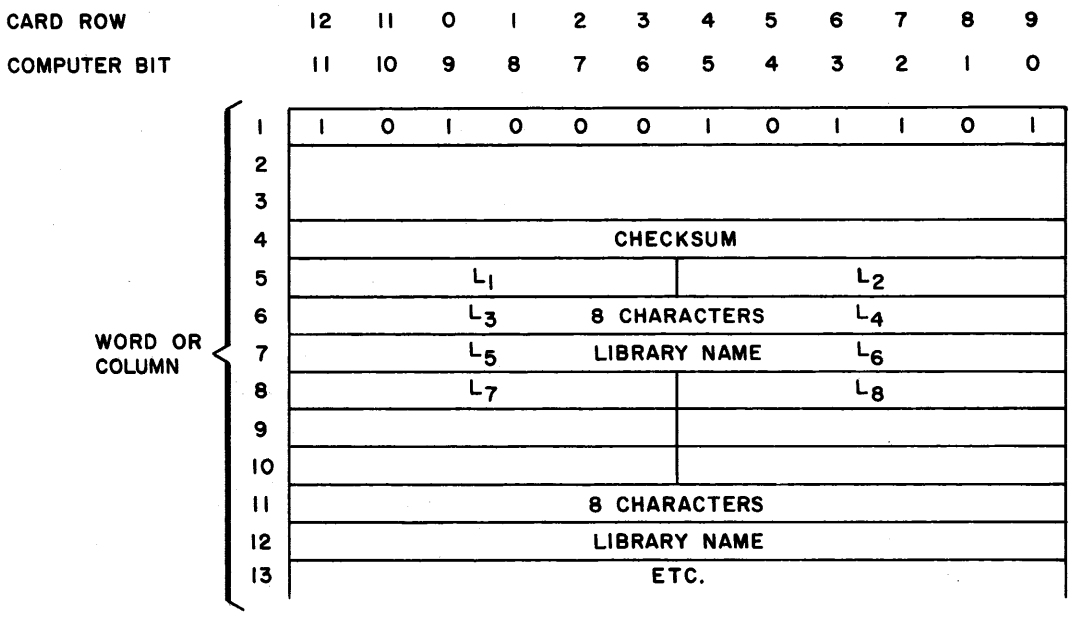
1. SC is a count of the number of words required to hold the symbol.

   Example: JILL requires two words to hold; thus, SC=2.

2. f/r indicates by a 1 that the location of the last reference to the symbol is fixed and no increment is added to the address. 0 indicates a relocatable location, and the relocation constant must be added to the given address.

3. Symbols are carried in BCD code. A SC=0 on the next symbol indicates end of symbols in a card. Space required for a symbol depends on the number of characters in the symbol.

Figure 1-6. External Symbol Table Card (WC = 104_8)



NOTE: Library name is carried in BCD code, two characters per column. The end of card or blank column indicates the end of the list.

Figure 1-7. MASS Library Call Card (WC = 105_8)

CHAPTER 2
MASS LOADERS

RELOCATING BINARY LOADER

The relocating binary loader is a service routine for use outside of MASS, it relocates a program within an 8192 bank of memory with no restrictions. It is normally the first program brought into the 160G through the auto-load routine. As such, it is written using the 160-A subset of instructions until the time the system can establish the 13-bit information required to use the 160G instructions. The standard version is fixed and operates from locations 17400 to 17776.

OPERATION

The general operation, after getting the program relocation constant from the console, is to read each relocatable binary program card. If the f/r bit indicates the card is fixed, it is the output of a portion of the program assembled under the CON counter. In this case, the bank and address are the actual starting location for storage of the information contained in the binary card. If the f/r bit indicates the card is relocatable, the relocation increment is added to the address portion (13 bits) to specify the starting location of the information contained in the card. Information is then stored in successive locations starting at the beginning address. If the relocation bit corresponding to a word is a 1, the relocation constant is added to the word before it is stored in memory.

On reading a transfer card, the relocation increment is added to the transfer address if the f/r bit indicates the address is relocatable and the transfer is made to the program. If the transfer address is 0 and also not relocatable, the loader stops and allows operator intervention.

ERRORS

The conditions that create errors and also cause an abandoning of the loader process are as follows:

1. The occurrence of an external symbol table card.
2. The occurrence of a MASS library call card (WC = 105).

LINKING RELOCATABLE LOADER

The linking relocatable loader is a routine which is always present in MASS. This routine has the capability of accepting binary program decks (or magnetic tape images) from the GASS output and loading the programs in the available storage of the 160G computer. As a part of the loading process, the linking loader is able to move a program from the area in which it was originally assembled and relocate it in other areas of memory. As a completion of the loading process, the linking loader combines cross references between programs and provides a running program.

INPUT DATA

GASS generates the input information to the linking relocatable loader. The main binary card of interest is the binary program card which contains relocation information in the form of a flag bit associated with each storage location of a program. If the flag bit is a 1, the contents of the storage location must be modified by the relocation constant if the program is relocated from its assembled location.

The programmer specifies to GASS those symbolic locations which are referenced from outside of his program. These symbols are called entry points into the program. The specification of entry points in the ENTRY pseudo instruction causes an entry point table card to be included in the output of GASS. The linking relocatable loader accepts this entry point table and stores the information immediately following the program in memory. A linkage device is included to lead to the entry point table associated with each segment of the included program.

The programmer also specifies those locations which are external to a program in the EXTNL pseudo instruction. The operation of GASS is to leave sufficient information in the program so that when combined with the external symbol table the linking loader can complete the assembly process at load time. The information from the external symbol table is loaded at the high end of core memory and downward.* At the completion of loading of all subprograms, a complete entry point table exists, giving the entry points to all subprograms loaded into memory. This entry point table is fragmented and exists in part at the end of each program. A last value specifies the start of the entry point table. The external symbol table is loaded into the top of the available memory and is a continuous table.

After the loading and relocation are completed and the full tables exist as given previously, the relocating loader turns control over to the linking portion of the loader which matches external symbol values with the corresponding entry point values and modifies the programs so they correctly reference locations in memory.

*The highest location in a bank is 17777_8 (G mode). The lowest location in a bank is 00000_8.

CHAPTER 3
EXECUTIVE CONTROL SYSTEM

The executive control system of MASS allows the operator to issue a series of high level commands or instructions which call for programs to be loaded into the memory of the 160G from the library tape, and also cause the execution of the programs while providing control information to the program.

DESCRIPTION

EXECUTIVE INSTRUCTIONS

An executive instruction (normally input from the typewriter) begins with an entry point name. Commas separate the following arguments and a period closes the instruction.

In general, the arguments are quantities specifying which of several options are to be used in the execution of the called program. A series of parameters in a return jump calling sequence provide the arguments to the program when the execution calls for the program.

TYPES OF INSTRUCTIONS

There are two types of executive instructions: library calls and execution instructions. A library call transfers a program from a library tape into storage for later execution. The routine with the entry point CALL in the resident portion of MASS performs this function. For example, the statement:

```
CALL,1,LOAD,VERIFY,EFMARK .
```

This library call directs that the program LOAD, VERIFY, and EFMARK (end-of-file mark) are to be transferred from magnetic tape unit 1. The name given in the IDENT card identifies the programs. Ordinarily, a library call clears storage of previously called programs. However, the resident routine with the entry point HOLD retains the

previous programs indefinitely. The resident routine named CLEAR clears storage of the programs being held.

An execution instruction prepares arguments as parameters of a return jump calling sequence and then performs a JPRG instruction, to the called program. An example of an execution instruction is:

```
VERIFY,3,4.
```

Note that an execution instruction may be given for any routine in the resident program without a preceding library call. However, a library call must precede an execution instruction for a library program.

SOURCE OF EXECUTIVE INSTRUCTIONS

An executive program consisting of a number of executive instructions always is input from the keyboard initially. Usually, further sequencing of programs is then provided from some other input device. Thus, if control program is on, the operator turns control over to the card reader by typing in:

```
CONTROL,32.
```

The number 32 is the logical unit number assigned to the card reader. Under typewriter control, when an executive instruction is typed in, a carriage return should precede the first character of the instruction. If a mistake is made, returning the carriage nullifies the previous entry before typing the period. Once the period is typed, a carriage return does not nullify the entry.

After an executive instruction is entered from the control medium, the resident routine types the statement on the typewriter (unless it was entered from the typewriter) and searches the entry point table for the resident routines and the main program storage for the specified program. The program is executed and the next executive control instruction is acted on in a similar manner. If the program named the executive instruction is not in core storage, the typewriter types out: PROGRAM NOT CALLED. The executive control program is then abandoned and control reverts to the typewriter keyboard.

RESIDENT PROGRAMS

The resident program of MASS includes some basic callable programs for input-output and the manipulation of programs. Executive instructions for these programs may be given without a previous library call.

Each of the programs are described as follows: The title lines give the format of an executive instruction, that is, the name and verbal description of the type of arguments required. The arguments must be numeric, with the exception of the CALL program.

CALL

CALL takes up to eight programs from a specified library tape and places them in main core storage. CALL saves time by requiring only a single search of a library tape to extract several programs. The library tape parameter must be numeric. CALL has the following format:

CALL,Library Tape,Program Ident,Program Ident.

Example: CALL,1,LOAD,VERIFY,EFMARK.

During the execution of CALL, magnetic tape unit 1 reads the designated programs LOAD, VERIFY, and EFMARK into main core storage. They are not executed. To be executed, the programs must have an executive instruction to turn control over to them.

If a program is not found on the library tape (say its IDENT name is ALPHA), the typewriter prints:

ALPHA NOT AVAILABLE

CLEAR

CLEAR releases main core storage of programs retained therein because of the execution of HOLD; CLEAR uses no arguments. CLEAR has the following format:

CLEAR.

CONTROL

CONTROL is used to turn executive program control to another input media, which is in card-image forms. The resident program considers the typewriter to be the source of executive instructions until CONTROL is used to turn executive control over to another device. The routine uses one argument to specify the medium holding the executive program. CONTROL has the following format:

CONTROL,Input.

HOLD

HOLD provides for retaining in main storage all programs previously called from the library. The execution of CLEAR releases the programs held in main storage. Unless HOLD is executed before a library call, the CALL clears main storage of previously called programs. HOLD uses no arguments. HOLD has the following format:

HOLD.

INPUT

INPUT changes the organization of the input-output system so that the logic unit number references a different equipment. The equipment designator varies with each installation. INPUT has the following format:

INPUT,Logic Number,Equipment Number.

OUTPUT

OUTPUT assigns the equipment number to correspond to the logical unit number for output only. The equipment number designator varies with each installation. The format for OUTPUT is as follows:

OUTPUT,Logic Number,Equipment Number.

INOUT

INOUT assigns the equipment number to correspond to the logical unit number for input and output. The equipment number designator varies with each installation. The format for INOUT is as follows:

INOUT,Logic Number,Equipment Number.

PREPARING PROGRAMS FOR OPERATION BY EXECUTIVE

The executive control system input routine reads each specified parameter (numeric only) and stores it in successive locations starting at 00070_8 through 00077_8. It then turns control over to the called program by a JPRG instruction to the specified entry point. The control program places the number of parameters in the A register before executing the JPRG instruction.

A maximum of eight parameters may be specified to any program. The program being executed then takes the parameters from location 00070_8 and particularizes the program to do the required job. A programmer may call for any of the routines through the executive linkage system by setting the parameters in locations 00070_8 and executing a JPRG instruction to the entry point of the program. The entry point of the programs to be executed must be specified as EXTERNAL to the driving routine.

CHAPTER 4
INPUT-OUTPUT CONTROL SYSTEM

DESCRIPTION

The standard input/output system is based on the use of four routines to establish input or output, to check the I/O process, or to establish external conditions. The four routines are:

READ, WRITE, CONTROL, and CHECK

READ and WRITE are used to initiate the input or output of information. On the initiation of I/O, the main program is allowed to continue and an interrupt routine is established to complete the I/O operation. The use of CHECK verifies the completion of an I/O operation. Normally, CONTROL initiates nonstandard movement of data in I/O devices and establishes control operations in the I/O device.

The main routine interprets the unit number of the standard calling sequence of any of the four routines and establishes the linkage to the particular I/O routine corresponding to the unit number. At any installation, the MASS routine is particularized to cover only the I/O equipment installed, and also to the channel configuration of the I/O equipment.

The I/O routines occupy the first part of bank 0 following the interrupt areas. The individual READ and WRITE routines consist of two parts: The first part to initiate the input or output; and a part which is linked via a buffer completion interrupt to perform error checking and completion of the operation. At the completion of the operation, the status is saved in case of minor errors for turn over to the check routine. The main READ or WRITE routines use locations 00074 to 00077 to pass parameters to the individual equipment routines.

All four standard routines READ, WRITE, CHECK, and CONTROL include as the first parameter a control word and unit number designation. In all cases the lower six bits of the control word are a logical unit number designation. Each of the standard routines obtains

the parameters from the calling sequence and stores the parameters in locations 00074 to 00077_8. The routines then use the logical unit numbers to perform a double-table lookup to determine the required equipment control routine which performs the required function.

The first table lookup is a unit number table. The contents of the cell corresponding to a logical unit number is a 6-bit I/O specification table reference number. The unit number table provides an easy reassignment of logical unit numbers to other equipments to allow different units to be specified and substituted at will by the operator.

The information contained in each cell of the I/O specification table requires 24 bits. The bit usage is given in Table 4-1.

The standard logical unit number assignment is given in Table 4-2.

The I/O specification table is constructed for each installation to correspond to the type of equipment and channel configuration of the equipment.

TABLE 4-1. I/O SPECIFICATION TABLE

No. of bits	Purpose
2	Specifies allowable usage for a piece of equipment. 00 = not available 01 = output only 10 = input only 11 = input or output
3	Specifies I/O channel assignment for equipment. 000 = normal I/O channel (non-buffered) 001 = internal buffer data channel 010 = external buffer data channel 2 011 = external buffer data channel 3 100 = external buffer data channel 4 101 = external buffer data channel 5 110 = external buffer data channel 6 111 = external buffer data channel 7
2	Specifies control unit number 00 = first control unit 01 = second control unit 10 = third control unit
3	Specifies equipment number within control unit, for example, on magnetic tape control. 000 = tape unit number 0 001 = tape unit number 1 010 = tape unit number 2 011 = tape unit number 3 100 = tape unit number 4 101 = tape unit number 5 110 = tape unit number 6 111 = tape unit number 7
4	
13	Specifies address of group of entrances to particular equipment routine.

TABLE 4-2. LOGICAL UNIT NUMBERS

Unit Number	Allowable Usage	Peripheral Equipment
00	Input	Magnetic tape 0 on control unit 1 (library tape)
01	I/O	Magnetic tape 1 on control unit 1
02	I/O	Magnetic tape 2 on control unit 1
03	I/O	Magnetic tape 3 on control unit 1
04	I/O	Magnetic tape 4 on control unit 1
05	I/O	Magnetic tape 5 on control unit 1
06	I/O	Magnetic tape 6 on control unit 1
07	I/O	Magnetic tape 7 on control unit 1
1X	I/O	Magnetic tape x on control unit 2
2X	I/O	Magnetic tape x on control unit 3
30		Unassigned
31	I/O	161G Typewriter
32	Input	405G Card Reader
33	Output	170G Card Punch
34	Output	1612G Printer
35	Output	165G Plotter
36 to 77		Unassigned

READ

The action of READ is to first check to see if the I/O channel required is available. If the channel is not available, the routine waits until it is available and then turns control over to the individual equipment routine. If there is a condition such that it is impossible to execute the READ function (no such unit defined, or unit out-of-order, etc.) the program is abandoned and control returned to EXECUTIVE with a typewritten message sent to the operator.

The calling sequence and operation for the READ routine are as follows:

LOCATION	OPERATION	ADDRESS FIELD	COMMENTS
	JPRG	READ	
		CWUN	
		$FWA	
		FWA	
		NØWDS	

The word CWUN contains a control code in the upper seven bits and a logical unit number in the lower six bits. $FWA specifies the bank number which contains the area to be read into, and FWA gives the first word address of the area to be read into. NOWDS specifies the number of words of data to be read.

The control word CW is represented in binary as ccccfffuuuuuu, where uuuuuu is the logical unit number.

fff specifies the internal format of the data to be read or written, and in octal has the following meaning:

0 Binary internal formal lower 12 bits of a word. On a WRITE, the high-order bit is ignored. On a READ, the high-order bit is read in as a 0.

1 BCD internal format lower six bits are the character. On a WRITE, the upper seven bits are ignored. On a READ, the upper seven bits are set to 0.

2 BCD internal format stored two characters per word. On a WRITE, the high-order bit is ignored. On a READ, the high-order bit is set to 0.

cccc is used to specify operation of the printers.

WRITE

WRITE operates the same as READ. The calling sequence for WRITE is as follows:

LOCATION	OPERATION	ADDRESS FIELD	COMMENTS
	JPRG	!WRIT	
		!CWUN	
		!$FWA	
		!FWA	
		!NØWDS	

ERROR CONTROL

The individual routines written for READ and WRITE operations provide as complete a recovery as possible from error conditions. This includes re-reading of magnetic tape on input, etc. If it is not possible to recover from the error, control is transferred to the operator with an indication of the type of error. The operator can then call for a re-read, deletion of record, or abandonment of the job.

DATA FORMATS

Internal data is accepted in three formats:

1. Binary data 12 bits per word.
2. BCD data with the lower 6 bits of a word as the BCD character.
3. BCD data stored two characters per word.

The preceding numbers are taken as bits 7 and 6 of the control word. The logical unit number is bits 5 through 0 of the control word and bits 12 through 9 are taken as special functions.

CHECK

CHECK provides a check on the logic unit number specified in CWUN if the unit is available, and the last operation was correct, the routine returns to the next instruction. Depending on the control word (CW) and the conditions checked for, control is transferred to the routine specified as ALTRET, if there is an unusual condition. The calling sequence for CHECK is as follows:

LOCATION	OPERATION	ADDRESS FIELD	COMMENTS
	JPRG	ICHEK	
		ICWUN	
		IALTRET	

The control word CW is represented in binary as ccccccuuuuuu where uuuuuu is the logical unit number. The control value ccccccc has the octal values and meaning as:

00	Wait until the completion of the previous I/O operation on the unit and then check for the successful completion of the I/O operation. If an unusual operation has occurred, transfer control to ALTRET in the same bank as the calling sequence to CHECK.
01	Wait until equipment is ready for I/O operation and take normal return to the next instruction.
02	Take alternate return if the equipment is not ready or available for I/O operation. Take next instruction if equipment is available.

In the check of the last I/O operation, the alternate return is made with the length of the input (if read) in the Q register and an error code in the A register. The error code may be:

00000	Input is less than specified in the READ calling sequence
00001	End-of-file mark read on READ operation
00002	End-of-tape mark sensed on WRITE operation

All other I/O troubles require operator intervention and are handled via the I/O typewriter.

CONTROL

This routine causes the control action, given in CW, to be applied to the logical unit specified in UN. The control actions are normally equipment movement such as write file mark, rewind, eject paper, etc. However, some control functions on the printer, for example, establish some operating option. The calling sequence for CONTROL is as follows:

LOCATION	OPERATION	ADDRESS FIELD	COMMENTS
	JPRG	ICTRL	
		ICWUN	

The control word CW is represented in binary as ccccccuuuuuu where uuuuuu is a logical unit number and ccccccc is a control action which depends on the unit. The value of ccccccc in octal and the action on each unit is given in Table 4-3.

TABLE 4-3. EQUIPMENT CONTROL WORDS

ccccccc	Magnetic Tape	Card Reader	Printer	Card Punch
000	Ignored	Ignored	Ignored	Ignored
001	Search EOF forward		Eject 2 lines	
002	Search EOF backward			
003	Skip 1 record forward	Pass 1 card	Space 1 line	Eject 1 blank card
004	Backspace 1 record			
005	Rewind to load point			
006	Rewind unload			
007	Write EOF		Page eject	Punch EOF card
010	Set low density		Clear control channel on format tape	
011	Set high density		Select monitor Ch. 1	
012	Set hyperdensity		Select monitor Ch. 2	
013			Select monitor Ch. 3	
014			Select monitor Ch. 4	
015			Select monitor Ch. 5	

Table 4-3. EQUIPMENT CONTROL WORDS (Continued)

ccccccc	Magnetic Tape	Card Reader	Printer	Card Punch
016			Select monitor Ch. 6	
017			Skip to monitor Ch. 7	

NOTE: All codes not specified are ignored.

PERIPHERAL EQUIPMENT

This section details the steps used in programming the standard I/O peripheral equipment using the individual MASS I/O routines. It also includes instructions that the operator can use to exercise manual control over the equipment in case of a malfunction.

TYPEWRITER (161G)

READ

LOCATION	OPERATION	ADDRESS FIELD	COMMENTS
	JPRG	READ	
		xx31B	
		$FWA	
		FWA	
		NØWDS	

xx may have the values 01 and 02 specifying one character per word and two characters per word format internally. The value 00 is an error and causes the program to be abandoned and control turned over to the operator. One READ statement may enter a maximum of 80 characters from the typewriter. The characters are converted to BCD code and stored in the required internal format. If a two-character per word format is specified and an odd number of words is read in, the last character is entered as a space or blank code.

WRITE

LOCATION	OPERATION	ADDRESS FIELD	COMMENTS
	JPRG	WRIT	
		xx31B	
		$FWA	
		FWA	
		NØWDS	

Limitations for WRITE are the same as for READ.

CONTROL

LOCATION	OPERATION	ADDRESS FIELD	COMMENTS
	JPRG	CTRL	
		xx31B	

The value of xx and their functions are:

00	Ignored
01	Perform two carriage returns
03	Perform one carriage return
07	Perform eight carriage returns

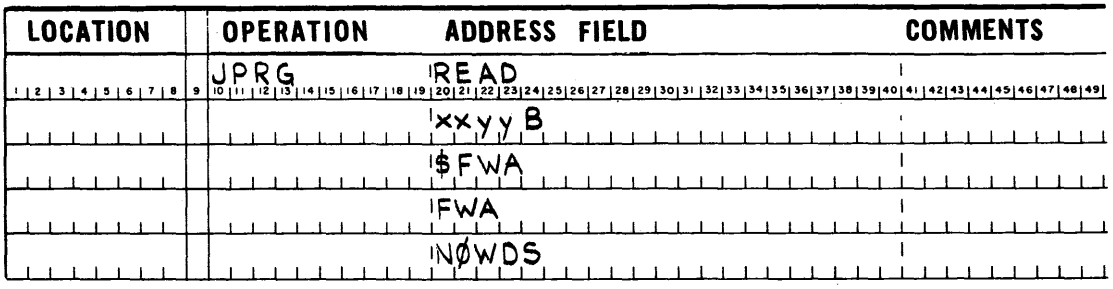
CHECK

LOCATION	OPERATION	ADDRESS FIELD	COMMENTS
	JPRG	CHEK	
		xx31B	
		ALTRET	

Standard response to typewriter error checking. A carriage return or the issuance of a manual interrupt terminates the input data.

MAGNETIC TAPE (606)

READ



The value of yy in octal specifies the tape control unit and the tape unit number as shown in the standard unit number table, Table 4-2. Tape control unit 1 in an installation is expressed as 0y, unit 2 is 1y, and unit 3 is expressed as 2y. The tape unit number within a control unit is expressed as 0 through 7. Thus, tape No. 2 of unit 2 is written as 12.

The value of xx is as follows:

00	Indicating that the internal format is binary, 12 bits per word. The tape is read or written in odd parity.
01	Indicating that the internal format is BCD, one character per word. The tape is read or written in even parity.
02	Indicating that the internal format is BCD, two characters per word. The tape is read or written in even parity.

If a parity error occurs during the READ operation, the operator is informed at the typewriter. This error is indicated after the READ routine tries three times to read the record. The operator is also informed of the length read in and the actual length read from the tape. The program waits for an operator decision from the keyboard. The operator types a one-character command. These commands are as follows:

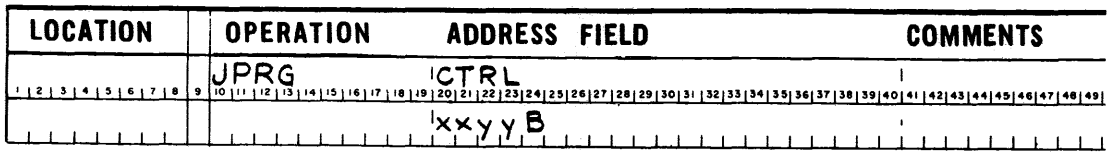
a Accept the record and continue with the program.
r Reject the record and read another record.
q Quit the program and return to master control.
t Try to read the bad record again.

WRITE

WRITE operates on the same basis as READ. If during the WRITE, a parity error is discovered, the program attempts to pass over the bad spot on the tape, and it also checks the bad spot to see if there is any residual noise. If there is noise, the operator is informed of the fact, and the program waits on an operator decision from the keyboard. The operator commands are as follows:

a Accept the noisy inter-record gap.
q Quit the program and return to master control.

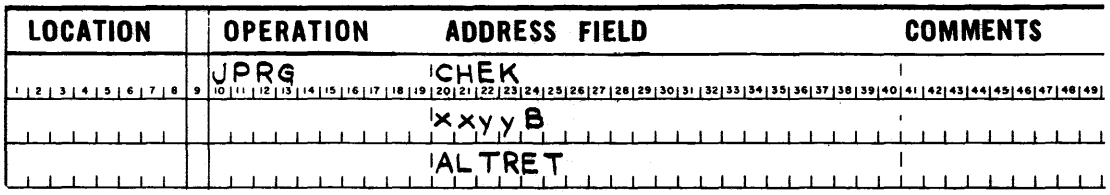
CONTROL



The value of xx and their functions are:

00 Ignored
01 Search end-of-file forward on tape yy
02 Search end-of-file backward on tape yy
03 Skip one record forward on tape yy
04 Backspace one record on tape yy
05 Rewind to load point tape yy
06 Rewind unload tape yy
07 Write end-of-file on tape yy
10 Set low density on tape yy
11 Set high density on tape yy
12 Set hyperdensity on tape yy

CHECK



If all operations are normal, or if the code xx requires the previously stated conditions, return is to the next instruction. Control is transferred to the routine ALTRET in the same bank as the calling sequence in case of a trouble condition on the last READ or WRITE operation. The A and Q registers contain the status information. The Q register contains the actual length of the information read on the last READ operation. The A register may have the following numbers for each condition:

00000	Last input is shorter than specified on the input instruction.
00001	End-of-file mark read on the last input operation.
00002	End-of-tape mark sensed on the last output operation.

CARD READER (405/167G)

READ

LOCATION	OPERATION	ADDRESS FIELD	COMMENTS
	JPRG	READ	
		xx32B	
		$FWA	
		FWA	
		NØWDS	

The values of xx are as follows:

00	Indicating that the internal format is column binary, one column per word. The card is copied in a binary hole for bit form.
01	Indicates that the internal format is BCD, one character per word. The card information is converted to BCD code.
02	Indicates that the internal format is BCD, two characters per word.

If there is any malfunctioning during the card read, before or after, the typewriter informs the operator and he has the option of continuing or not. The possible conditions and operator options are given in Table 4-4.

TABLE 4-4. OPERATOR COMMANDS

Condition	Operator Option (type on typewriter)
Any	q Quit the program and return to master control
Hopper empty before read	t Try to read record again (operator had added cards)
	f Return as file mark read
Stacker full	t Try to read record
Feed failure, program error, or amp. failure	t Try to read record (operator put in new card)
Motor power off	t Try to read (operator turned power on)

CONTROL

LOCATION	OPERATION	ADDRESS FIELD	COMMENTS
	JPRG	CTRL	
		03328	

This instruction causes one card to be read without information input to the computer.

CHECK

LOCATION	OPERATION	ADDRESS FIELD	COMMENTS
	JPRG	CHEK	
		xx328	
		ALTRET	

If all operations are normal, or if the code xx requires the previously stated conditions, return is to the next instruction. Control is transferred to the routine ALTRET in the same bank as the calling

sequence in case of a trouble condition on the last read. The A register contains 00001 if there was an end of file signaled by the operator on the last operation.

CARD PUNCH (170G)

WRITE

LOCATION	OPERATION	ADDRESS FIELD	COMMENTS
	JPRG	WRIT	
		xx33B	
		$FWA	
		FWA	
		NØWDS	

The values of xx are as follows:

00	Indicates that the internal format is column binary, one column per word. The card write routine makes the appropriate change of format to work with the card punch in the system.
01	Indicates that the internal format is BCD, one character per word.
02	Indicates that the internal format is BCD, two characters per word.

If the card punch is not ready, the system typewriter informs the operator and he has the chance to correct the condition. The operator may then type one of the following:

q	Quit the program and return to master control.
t	Try to write the record (the punch is turned on).

CONTROL

LOCATION	OPERATION	ADDRESS FIELD	COMMENTS
	JPRG	CTRL	
		0332B	

This instruction causes one blank card to be ejected in the card punch.

CHECK

LOCATION	OPERATION	ADDRESS FIELD	COMMENTS
	JPRG	CHEK	
		xx33B	
		ALTRET	

No abnormal conditions are returned from the card punch operation.

PRINTER (1612G)

WRITE

LOCATION	OPERATION	ADDRESS FIELD	COMMENTS
	JPRG	WRIT	
		xx34B	
		$FWA	
		FWA	
		NØWDS	

The value xx specifies internal format by the lower octal digit and printer options by the upper two digits. The lower digit options allowed are as follows:

1. Internal format is BCD, one character per word.
2. Internal format is BCD, two characters per word.

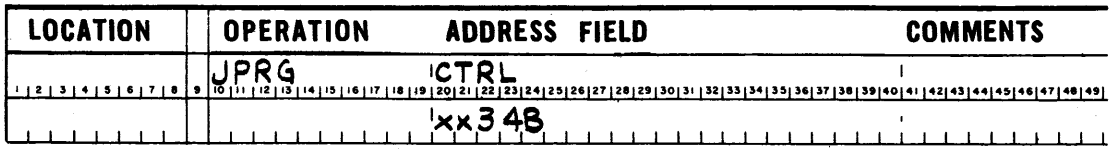
The upper digits are printer options accompanying the WRITE instruction. The values are:

0x	Advance paper one line and print
1x	Advance paper one line and print
2x	Advance paper two lines and print
3x	Advance paper to top of page and print
4x	Do not advance paper and print on same line
5x	Print under column 1 format control
6x	Print under printer format tape control

(Only the first 120 or 121 characters of the area specified enter the print operation.)

The operator is informed of any programming errors or any printer error condition.

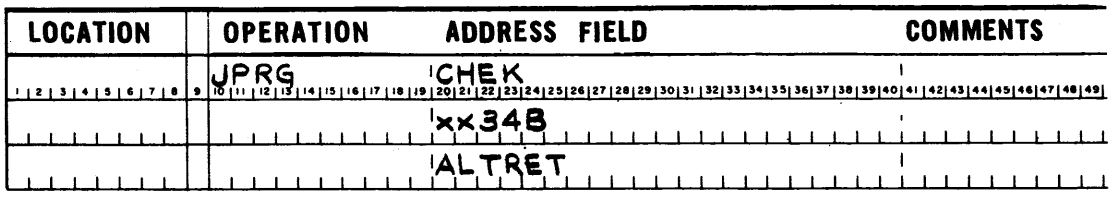
CONTROL



The value xx selects a printer condition or operation mode. The values of xx are as follows:

01	Eject two lines
03	Space one line
07	Page eject
10	Clear format tape control
11	Select monitor channel 1
12	Select monitor channel 2
13	Select monitor channel 3
14	Select monitor channel 4
15	Select monitor channel 5
16	Select monitor channel 6
17	Skip to monitor channel 7

CHECK



CHECK only provides exits as required to check the busy status of the printer and the I/O channel.

CHAPTER 5
INTERRUPT HANDLING SYSTEM

Interrupts are handled by two routines in MASS which allow the activation and deactivation of interrupt linkages. The routines are ACIN and DCIN. The calling sequences and action are as follows:

DESCRIPTION

ACIN

```
LOCATION    OPERATION    ADDRESS FIELD              COMMENTS
            JPRG         ACIN
            NØ
            RTN
```

ACIN sets a linkage from an interrupt to the location RTN. NO is a code number specifying the interrupt desired. The assignment of interrupt numbers in octal is given in Table 5-1.

TABLE 5-1. INTERRUPT NUMBER ASSIGNMENT (OCTAL)

Channel	Buffer Termination	External 1	External 2
1	10	11	12
2	20	21	22
3	30	31	32
4	40	41	42
5	50	51	52
6	60	61	62
7	70	71	72
Console Interrupt 01			
Computer to Computer Interrupt 02			

On the occurrence of a specified interrupt, the interrupt control system saves the contents of the registers and bank controls and sets up the necessary instructions to return to the main program. The routine specified in the ACIN is then entered with a JPRG instruction. On completion of the interrupt routine, the return to the interrupt handling routine is made by a jump to the entrance of the interrupt program. The A register then contains the interrupt number on entrance to the interrupt routine.

DCIN

LOCATION	OPERATION	ADDRESS FIELD	COMMENTS
	JPRG	DCIN	
		NØ	
		NØ1	
		etc.	

The deactivate interrupt removes the linkages established by activate interrupt. A list of linkages may be removed. The value of the interrupt code numbers is less than 77_8 and the occurrence of any instruction terminates the list. Instructions with the operation codes 000xx should not immediately follow a deactivate interrupt code list.

COMMENT SHEET

CONTROL DATA

MASS Reference Manual Publication No. G02075a

FROM NAME :_____

BUSINESS ADDRESS :_____

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

CUT ALONG LINE

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

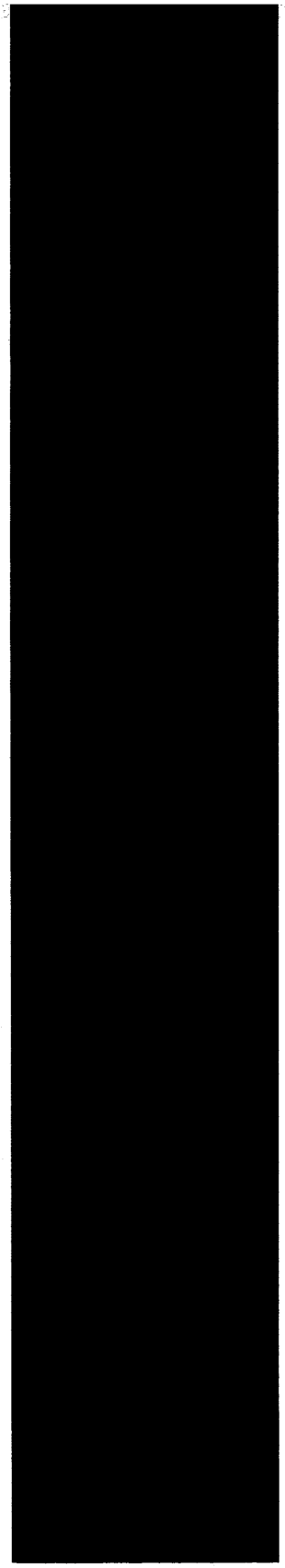
CUT ALONG LINE

FOLD

FOLD

STAPLE

STAPLE



CONTROL DATA

CORPORATION

8100 34th Avenue South, Minneapolis 20, Minnesota