HEWLETT hp PACKARD

BASIC CONTROL SYSTEM

BASIC CONTROL SYSTEM
reference manual

HEWLETT · PACKARD COMPUTER

February 1968

PREFACE

This publication is a reference manual for the programmer using the HP 2116A Basic Control System. It includes the necessary programming and operating information for making use of the System.

The reader is assumed to be familiar with the HP 2116A Computer and the Hewlett-Packard standard software systems. Completion of the Hewlett-Packard programming training courses or equivalent experience in programming is a prerequisite to the use of this manual.

Other computer publications provided by Hewlett-Packard include:

ALGOL Programmer's Reference Manual

Assembler/BCS Programmer's Training Manual

Assembler Programmer's Reference Manual

FORTRAN Programmer's Reference Manual

Program Library Routines Manual

Specifications and Basic Operation Manual

Standard Software Systems General Information Manual

Standard Software Systems Operating Manual

Symbolic Editor Programmer's Reference Manual

CONTENTS

INTRODUCTION

The Basic Control System (BCS) provides an efficient loading and input/output control capability for reloctable programs produced by the HP Assembler, HP FORTRAN, or HP ALGOL. The system is modular in design and may be constructed to fit each user's hardware configuration.

The Basic Control System performs the following functions:

Loads and links reloctable programs

Creates indirect and base page addressing when necessary

Selects and loads referenced library routines

Processes I/O requests and services I/O interrupts

Routines associated with the Basic Control System, though not physically part of it, include

Prepare Control System – establishes BCS configuration

Debugging System – loaded with user program to aid in program testing.

The minimum equipment configuration required for the Basic Control System (and Prepare Control System) is as follows:

2116A or 2115A Computer with 4K memory

2752 Teleprinter

GENERAL DESCRIPTION 1

The Basic Control System is comprised of two distinct parts; associated with the Basic Control System are two other systems. They are as follows:

Input/output subroutines

Relocating Loader

Prepare Control System

Debugging System

The Relocating Loader loads and links relocatable object programs generated by the Assembler, FORTRAN, and ALGOL. It also links the object programs with the input/output subroutines and any library subroutines referred to in the programs. The Prepare Control System is used to adapt the Basic Control System program to a particular hardware configuration. The Debugging System is a relocatable program that BCS loads after the object program(s); with the debugging program the programmer can find errors in his program.

1.1 INPUT/OUTPUT SUBROUTINES

The input/output package consists of an Input/Output Control subroutine and driver subroutines for the peripheral devices. Input/output operations are specified as symbolic calling sequences in Assembler language. These requests are translated into object code calls to the I/O Control subroutine. The subroutine interprets the call and directs the request to the proper driver. The driver initiates the operation and returns control to the calling program. Whenever interrupt occurs, the driver temporarily resumes control to transfer the next element of data. When the operation is completed, the I/O Control subroutine makes the status of the operation available for checking by the program.

The input/output package allows device independent programming; a device is specified in terms of a unit-reference number rather than a channel number or select code. Furthermore, the user need not be concerned about how data is transmitted (by bit, by character, etc.), he need only specify the number of words or characters and the location in memory where the data is stored.

1.2 RELOCATING LOADER

The Relocating Loader loads object code programs produced by the Assembler, FORTRAN and ALGOL. The linking capability of the Loader allows the user to divide a program into several subprograms, to assemble and test each separately, and finally to execute all as one program. Object subprograms produced by the Assembler may be combined with object subprograms produced by FORTRAN and ALGOL. The subprograms are linked through symbolic entry points and external references.

The loader also provides indirect addressing whenever an operand of an instruction does not fall in the same page as that into which the instruction is being loaded. This allows a program to be designed without concern for page boundarys.

An optional feature of the Loader allows the user to obtain an absolute version of a relocatable program which may include library subroutines and segments of the Basic Control System that were referenced by the program. The process of generating the absolute program is such that instructions (not just common storage) may be allocated to the area normally occupied by the Loader. This feature may also be utilized for a program which has reached "production" status; absolute format requires less loading time because an absolute program is loaded by the Basic Binary Loader.

1.3 PREPARE CONTROL SYSTEM

Prepare Control System is a special purpose program which produces an absolute version of the Basic Control System from relocatable BCS subprograms. During the construction of the absolute BCS, the user also establishes the relationships among I/O channel numbers, drivers, interrupt entry points in the drivers, and unit-reference numbers. Prepare Control System is used when the configuration of the hardware is defined initially or whenever there is a modification or expansion to the configuration.

1.4 DEBUGGING SYSTEM

The debugging routine provides aids in program testing. Options provided by the routine will print selected areas of memory, trace portions of the program during execution, modify the contents of selected areas in memory, modify simulated computer registers, halt execution of the program at specified breakpoints, and initiate execution at any point in the program.

INPUT/OUTPUT REQUESTS 2

The Basic Control System provides the facility to request input/output operations in the form of five-word calling sequences in assembly language. The Basic Control System interprets the call, initiates the operation, and returns control to the calling program. When the data transfer is complete, the System provides status information which may be checked by the program. Interrupts which occur during or on termination of the transfer are processed entirely by the System; interrupt handling subroutines are not required in the user's program.

2.1 GENERAL CALLING SEQUENCE

The general form of the input/output request is:

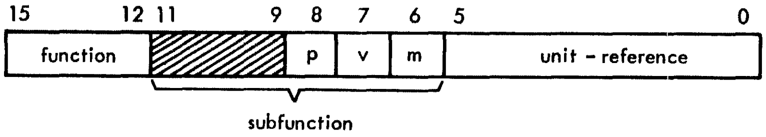
```
     JSB       .IOC.

     OCT       <function> <subfunction> <unit-reference>

    {JSB}
    {JMP}      reject address

     DEF       buffer address

    {DEC}
    {OCT}      buffer length

     <normal return>
          .
          .
          .
     EXT       .IOC.
```

2.1.1 .IOC.

.IOC. is the symbolic entry point name of the input/output control subroutine within the Basic Control System. All input/output operations are requested by performing a JSB to this entry point. The input/output control subroutine returns control to the calling program at the first location following the last word of the calling sequence. Programs referring to .IOC. must declare it as an external symbol.

2.1.2 FUNCTION, SUBFUNCTION, AND UNIT-REFERENCE

The second word of the request determines the function to be performed and the unit of equipment for which the action is to be taken. In assembly language, this information may be supplied in the form of an octal constant. The bit combinations that comprise the constant are as follows:



Function

The function (bits 15-12) is the basic input/output operation; it may be either of the following:

Function Name	Code (octal)
Read	01
Write	02

Subfunction

The subfunction (bits 11-6) defines the options for certain input/output operations:

p = 1 Print input: The ASCII data read from the 2752A Teleprinter is to be printed as it is received.

v = 1 Variable length binary input: The value in bits 15-8 of the first word on an input paper tape indicates the length of the record (including the first word). If the value exceeds the length of the buffer, only the number of words specified as the buffer length are read. If v = 0, the buffer length field always determines the length of record to be transmitted. If the device does not read paper tape, the parameter is ignored.

m = 1 Mode: The data is transmitted in binary form exactly as it appears in memory or on the external device. If m = 0, the data is transmitted in ASCII or BCD format.

Unit-Reference

The value specified for the unit-reference field indicates the unit of equipment on which the operation is to be performed. The number may represent a standard unit assignment or an installation unit assignment. Standard unit numbers are as follows:

Number	Name	Usual Equipment Type
1	Keyboard Input	Teleprinter
2	Teleprinter Output	Teleprinter
3	Program Library	Punched Tape Reader
4	Punch Output	Tape Punch
5	Input	Punched Tape Reader
6	List Output	Teleprinter

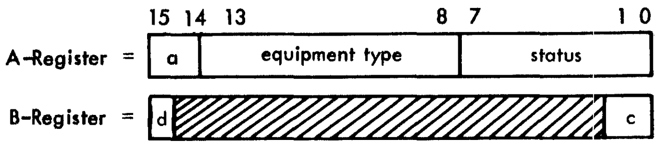
Installation unit numbers may be in the range 7_8-74_8 with the largest value being determined by the number of units of equipment available at the installation. The particular physical unit that is referenced depends on the manner in which equipment is defined within the Basic Control System by the installation. When the Basic Control System configuration is established, an equipment table (EQT) is created. This table defines the type of equipment (Teleprinter, magnetic tape, etc.), the channel on which each unit is connected, and other related details. The ordinal of the unit's entry in this table is the value specified as a unit-reference number for an installation unit. Since numbers 1-6 are reserved as standard unit numbers, the first unit described in the table is referred to by the number 7_8; the second, 10_8; the third, 11_8; and so forth. The entries for one possible equipment table might establish the following relationships:

Installation unit number (ordinal)	Device	I/O Channel
7	Teleprinter	12 or 12 and 13
10	Punched Tape Reader	10
11	Tape Punch	11

The standard unit numbers are associated with physical equipment via a standard equipment table (SQT) and EQT. The SQT is a list of references to the EQT. SQT is also created by the installation when the BCS configuration is established. Each standard unit may be a separate device, or a single device may be accessed by several standard unit numbers as well as an installation unit number. (For complete details on the SQT and EQT, see Appendix B.)

2.1.3 REJECT POINT

The Basic Control System transfers control to the third word of the calling sequence if the input/output operation can not be performed. On transfer, the System provides status information which may be checked by the user's program.



The contents of the A-Register indicate the physical status of the equipment (see Status Request). The contents of the B-Register indicate the cause of the reject (bits 14-1 are zeros):

d = 1 The device or driver subroutine is busy and therefore unavailable, or, for Kennedy 1406 Tape unit, a broken tape condition encountered.

c = 1 A Direct Memory Access channel is not available to operate the device.

d = c = 0 The function or subfunction selected is not legal for the device.

The content of the third word of the calling sequence is normally a JSB or a JMP to a reject address which is the start of a user subroutine designed to determine the cause of a reject and take appropriate action.

2.1.4 BUFFER STORAGE AREA

The buffer address is the location of the first word of data to be written on an output device or the first word of a block reserved for storage of data read from an input device. The length of the buffer area may be specified in terms of words or characters. If the length is given as words, the value in the buffer length field must be a positive integer; if given as characters, a negative integer.

In addition to describing the buffer area in the calling sequence, the area must also be specifically defined in the assembly language program, usually with a BSS or COM pseudo instruction.

2.2 CALLING SEQUENCE: PAPER TAPE SYSTEM

```
 JSB        .IOC.
 OCT        <function> <subfunction><unit-reference>
{JSB}
{JMP}       reject address
 DEF        buffer address
{DEC}
{OCT}       buffer length
 <normal return>
 EXT        .IOC.
```

Allowable combinations of function and subfunction codes are as follows:

Operation	Octal value of Bits 15-6
Read ASCII record	0100
Read ASCII record and print	0104
Read binary record	0101
Read variable length binary record	0103
Write ASCII or BCD record	0200
Write binary record	0201

An illegal combination of codes is rejected.

Record Formats

ASCII Records (Paper Tape)

An ASCII record is a group of characters terminated by an end-of-record mark which consists of a carriage return, CR, and a line feed, LF.

For an input operation, the length of the record transmitted to the buffer is the number of characters or words designated in the request, or less if an end-of-record mark is encountered before the character or word count is exhausted. The codes for CR and LF are not transmitted to the buffer. An end-of-record mark preceding the first data character is ignored.

For an output operation, the length of the record is determined by the number of characters or words designated in the request. An end-of-record mark is supplied at the end of each output record by the input/output system.

If the last character of an output record is ←, however, the end-of-statement mark is omitted. This allows control of Teleprinter line spacing. The user may write a message (the ← is not printed) and expect the reply to be typed on the same line. The reply must be terminated with the CR LF.

If a RUB OUT code followed by a CR LF is encountered on input from the Teleprinter or Punched Tape Reader, the current record is ignored (deleted) and the next record transmitted.†

If less than ten feed frames (all zeros) are encountered before the first data character from a paper tape input device, they are ignored. Ten feed frames are interpreted as an end-of-tape condition (see STATUS REQUEST).

Binary Records

A binary record is transmitted exactly as it appears in memory or an 8-level paper tape. The record length is specified by the number of characters or words designated in the request. The first character of a binary record must be non-zero. On input operations, less than ten feed frames preceding the first data character are ignored. Ten feed frames are interpreted as an end-of-tape condition (see STATUS REQUEST). On output, the system writes four feed frames to serve as a physical record separator.

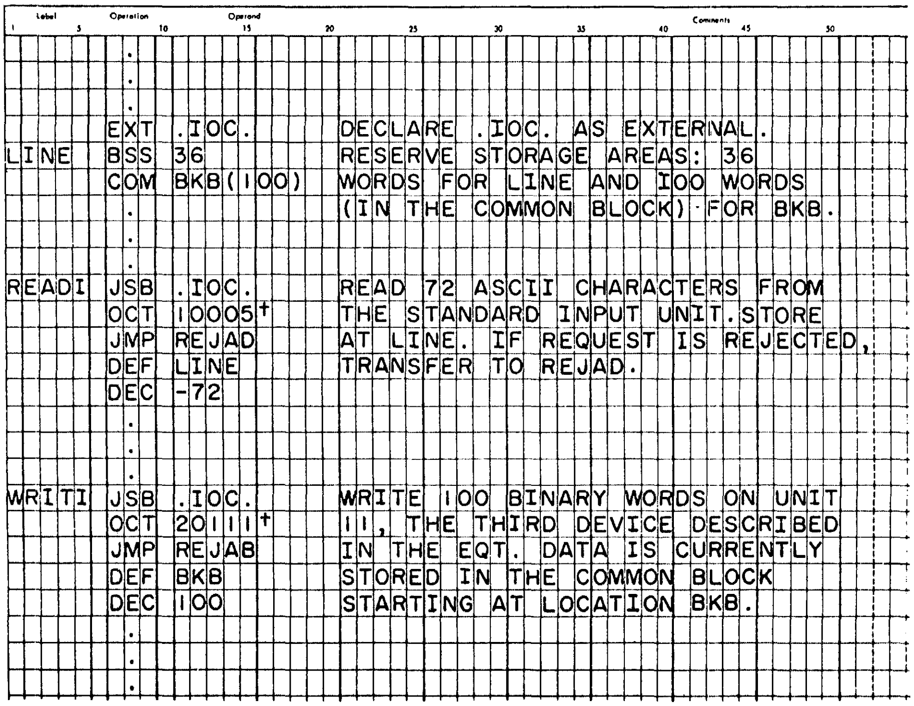
Binary input records may be variable in length. The first word of the record contains a number in bits 15-8 specifying the length of the record (including the first word). The entire record including the word count is transmitted to the buffer. If the actual length exceeds the size of the buffer, only the number of words equivalent to the buffer length is transmitted.

† RUB OUT which appears on the Teleprinter keyboard is synonymous with the ASCII symbol DEL.

2.2.2 BUFFER LENGTH

Character or word transmission may be specified for any paper tape device. The buffer length for data that may be printed on the teleprinter should be no more than 72 characters (36 words).

Examples:


Label
Operation
Operand
Comments
1
5
10
15
20
25
30
35
40
45
50

```
      .
      .
      .
      EXT .IOC.         DECLARE .IOC. AS EXTERNAL.
LINE  BSS 36            RESERVE STORAGE AREAS: 36
      COM BKB(100)      WORDS FOR LINE AND I00 WORDS
      .                 (IN THE COMMON BLOCK)·FOR BKB.
      .
      .
READI JSB .IOC.         READ 72 ASCII CHARACTERS FROM
      OCT 10005†        THE STANDARD INPUT UNIT.STORE
      JMP REJAD         AT LINE. IF REQUEST IS REJECTED,
      DEF LINE          TRANSFER TO REJAD.
      DEC -72
      .
      .
      .
WRITI JSB .IOC.         WRITE 100 BINARY WORDS ON UNIT
      OCT 20111†        11, THE THIRD DEVICE DESCRIBED
      JMP REJAB         IN THE EQT. DATA IS CURRENTLY
      DEF BKB           STORED IN THE COMMON BLOCK
      DEC 100           STARTING AT LOCATION BKB.
      .
      .
      .
```

† The leading 0 of the second word of the calling sequence need not be written in the source language since it is supplied in the object code as a result of using the OCT pseudo instruction.

2.3 CALLING SEQUENCE: KENNEDY INCREMENTAL TRANSPORT

```
     JSB      .IOC.
     OCT      <function><subfunction><unit-reference>
   { JSB }
   { JMP }    reject address
     DEF      buffer address
   { DEC }
   { OCT }    buffer length
     EXT      .IOC.
```

Function and Subfunction

Allowable function codes for the 1406 Kennedy Incremental Tape Transport are as follows:

WRITE (ASCII Mode only)	0200
WRITE End-of-file	0301
CLEAR	0000

Record Formats

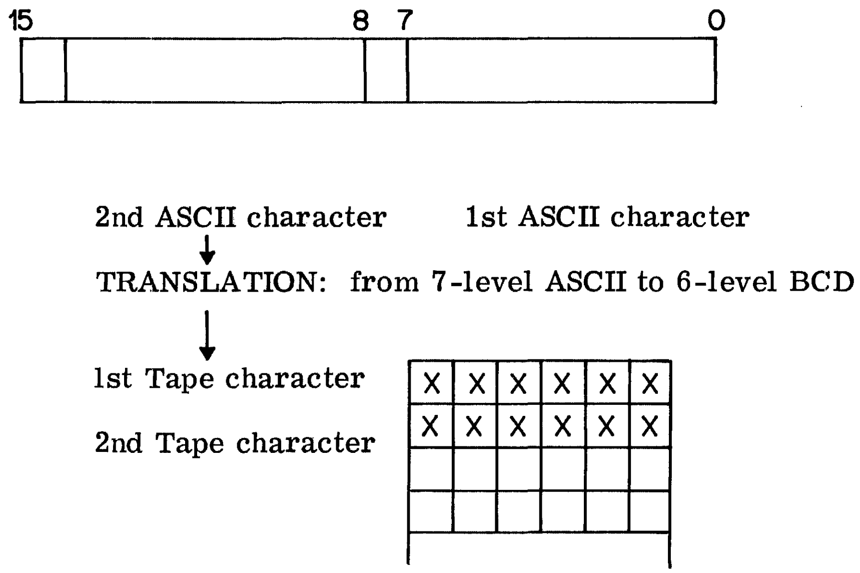
Binary Coded Decimal Records (Magnetic Tape)

A BCD record is a group of BCD characters terminated (on magnetic tape) by a record gap. A request to write a BCD record results in the translation of each 7-level ASCII character in the buffer area into a 6-level BCD character on magnetic tape. The translation process does not alter the original contents of the buffer.

The length of the record is determined by the number of characters or words designated in the request. A record gap is supplied at the end of each record by the input/output system.

If the last character in the buffer area is ←, however, the record gap is omitted. The ← is not written on tape.

A WRITE request specifying a buffer length of zero causes a record gap only to be written.



2.4 CALLING SEQUENCE: MAGNETIC TAPE SYSTEM

```
         JSB      .IOC.
         OCT      <function><subfunction><unit-reference>
       { JSB }
       { JMP }    reject address
         DEF      buffer address
       { DEC }
       { OCT }    buffer length
          .
          .
          .
         EXT      .IOC.
```

2.4.1 2020A Magnetic Tape Unit

All allowable combinations of function and subfunction codes are as follows:

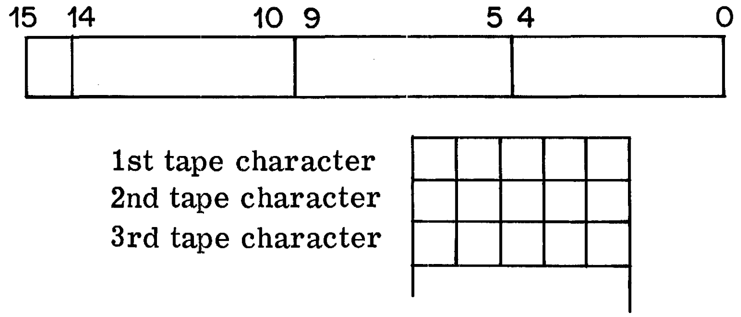
Operation	Octal value of Bit 15-6
Read BCD record and convert to ASCII	0100
Read binary record	0101
Write BCD record after converting from ASCII	0200
Write binary record	0201

RECORD FORMATS

BINARY RECORDS (MAGNETIC TAPE)

A binary record on magnetic tape is a group of 6-level tape "characters" recorded in odd parity and terminated by a record gap. † The record length is determined by the number of characters or words in the buffer as designated in the request.

Each computer word is translated into three tape "characters" (and vice versa) as follows:



For output operations, the minimum buffer length is 3 computer words.

BINARY CODED DECIMAL RECORDS

A BCD record on magnetic tape is a group of BCD characters recorded in even parity and terminated by a record gap. (See Appendix A for BCD character set.) A request to write a BCD record results in the translation of each 7-level ASCII character in the buffer area into a 6-level BCD character on magnetic tape. A request to read a BCD record results in the translation of each BCD character into an ASCII character after the block has been read.

The length of the record may not be more than 120 characters. A record gap is supplied at the end of each record.

† Odd parity: a seventh bit is recorded on tape if the total of the bits in the six levels is an even number.

Even parity: a seventh bit is recorded on tape if the total of the bits in the six levels is an odd number.

A WRITE request for the HP 2020A Magnetic Tape Unit must have a minimum buffer length of seven ASCII characters (four words). If less than seven characters are specified, spaces will be added to fill the seven characters.

2.4.2 3030 MAGNETIC TAPE UNIT

FUNCTION AND SUBFUNCTION

All allowable combinations of function and subfunction codes are as follows:

READ	0101	Binary only
WRITE	0201	Binary only
WRITE FILE MARK	0301	
SPACE	0302	Backspace one record
	0303	Forward space one record
REWIND	0304	Rewind to load point - Ready
	0305	Rewind to load point - Standby

Buffer Length

Character transmission is not applicable since the transmission is via a DMA channel. The minimum data block is twelve tape characters. Output blocks with a block length less than twelve characters are padded with zeroes.

2.5 MAGNETIC TAPE CONTROL REQUESTS

A request directed to .IOC. may also control the positioning of a reel on a magnetic tape device. The calling sequence is similar to the input/output request, but consists of only three words:

```
 EXT      .IOC.
  .
  .
  .
 JSB      .IOC.
 OCT      <function><subfunction> <unit-reference>
{JSB}
{JMP}     reject address
 <normal return>
```

The second word of the request has the following composition:

15 12	11 9	8 6	5 0
function	▨	subfunction	unit - reference

The function defines the calling sequence as a tape positioning request:

Function Name	Code (octal)
Position Tape	03

The subfunction defines the type of positioning:

Subfunction (octal)	Operation
0	Dynamic tape status
1	Write End-of-File
2	Backspace one record
3	Forward space one record
4	Rewind
5	Rewind and Standby

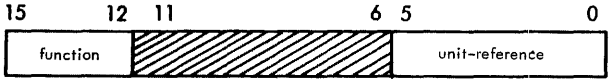
As soon as tape movement operations (Rewind, and Standby) are initiated, the device is considered to be available; the "a" field of a status reply (0400 code) is set to 00. The input/output driver is thus free to process requests for other devices. To obtain the actual status of the device when one of these commands has been issued, the Dynamic tape status request is used. If the tape movement operation is still in progress the "a" field is set to 10.

2.6 CLEAR REQUEST

The clear request terminates a previously issued input or output operation before all data is transmitted. It has the following form:

```
EXT         .IOC.
  .
  .
  .
JSB         .IOC.
OCT         <function>  <unit-reference>
<normal return>
```

The second word consists of the following:

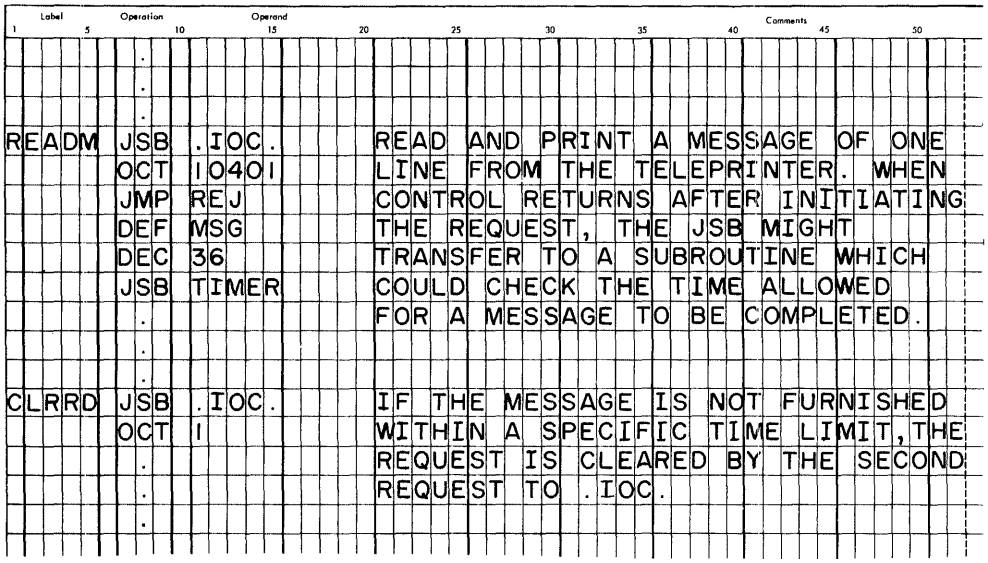


The function has the following value:

Function Name	Code (octal)
Clear	00

The only other parameter required is the unit-reference number. If the unit-reference number is specified as 00 (i.e., the second word of the calling sequence is OCT 0), all previous input and output operations are terminated. This request, the system clear request, makes all devices available for the initiation of a new operation. On return from a system clear request, the contents of the A- and B-Registers are meaningless.

Example:



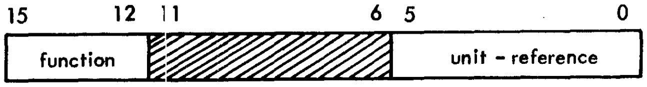
```
Label  Operation  Operand            Comments
1   5     10   15   20   25   30   35   40   45   50
       .
       .
       .
READM JSB .IOC.     READ AND PRINT A MESSAGE OF ONE
      OCT 10401     LINE FROM THE TELEPRINTER. WHEN
      JMP REJ       CONTROL RETURNS AFTER INITIATING
      DEF MSG       THE REQUEST, THE JSB MIGHT
      DEC 36        TRANSFER TO A SUBROUTINE WHICH
      JSB TIMER     COULD CHECK THE TIME ALLOWED
       .            FOR A MESSAGE TO BE COMPLETED.
       .
       .
CLRRD JSB .IOC.     IF THE MESSAGE IS NOT FURNISHED
      OCT 1         WITHIN A SPECIFIC TIME LIMIT,THE
       .            REQUEST IS CLEARED BY THE SECOND
       .            REQUEST TO .IOC.
       .
```

2.7 STATUS REQUEST

A request may be directed to .IOC. to determine the status of a previous input/output request or to determine the physical status of one or all units of equipment. The request has the following form:

```
JSB      .IOC.
OCT      <function><unit-reference>
<normal return>
```

The second word of the request has the following form:

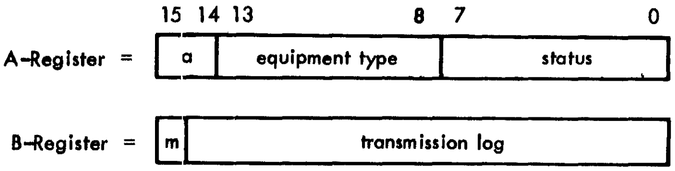


The function has the following value:

Function Name	Code (octal)
Status	04

The calling sequence requires no other parameters. A reject location is not necessary since the status information is always available. If the unit-reference number is specified as 00 (i.e., the second word on the calling sequence is OCT 40000), the request is interpreted as a system request.

If information is requested for a single unit, the Basic Control System returns to the location immediately following the request with the status information in the A and B registers:



a — Availability of device:

0 — The device is available; the previous operation is complete.

1 — The device is available; the previous operation is complete but a transmission error has been detected.

2 — The device is not available for another request; the operation is in progress.

equipment type — This field contains a 6-bit code that identifies the device referenced:

00-07 – Paper Tape devices

- 00 2752A Teleprinter
- 01 2737A Punched Tape Reader
- 02 2753A Tape Punch

10-17 – Unit Record devices

20-37 – Magnetic Tape and Mass Storage devices

- 20 Kennedy 1406 Incremental Tape Transport
- 21 HP 2020A Magnetic Tape Unit
- 22 HP 3030A Magnetic Tape Unit

40-77 – Instrumentation devices

- 40 Data Source Interface
- 41 DVM Programmer
- 42 Scanner Programmer
- 43 Time Base Generator

status — The status field indicates the actual status of the device when the data transmission is complete. The contents depend on the type of device referenced:

Teleprinter reader or Punched Tape Reader:

Bits 7-0	Condition
xx1xxxxx	End-of-Tape (10 Feed Frames)

Tape Punch:

Bits 7-0	Condition
xx1xxxxx	Tape supply low

Kennedy 1406 Incremental Tape Transport:

Bits 7-0	Condition
xx1xxxxx	End-of-Tape mark sensed
xxxx1xxx	Broken tape; no tape on write head
xxxxxxx1	Device busy

HP 2020A and 3030A Magnetic Tape Units:

Bits 7-0	Condition
1xxxxxxx	end-of-file record (17_8) encountered while reading, forward spacing, or backward spacing.
x1xxxxxx	start-of-tape marker sensed
xx1xxxxx	end-of-tape marker sensed
xxx1xxxx	timing error on read/write
xxxx1xxx	I/O request rejected: a. tape motion required but controller busy b. backward tape motion required but tape at load point c. write request given but reel does not have write enable ring.
xxxxx1xx	Reel does not have write enable ring or tape unit is rewinding.
xxxxxx1x	Parity error on read/write
xxxxxxx1	Tape in motion

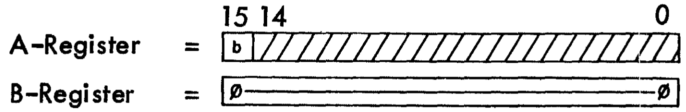
m — This bit defines the mode of the data transmission:

0 ASCII or BCD

1 Binary

transmission log — This field is a log of the number of characters or words transmitted. The value is given as a positive integer and indicates characters or words as specified in calling sequence. The value is stored in this field only when the request is completed, therefore, when all data is transmitted or when a transmission error is detected.

If a system status request is made, the information in the A and B registers is as follows:



b System Status

0 No device is busy

1 At least one device is busy

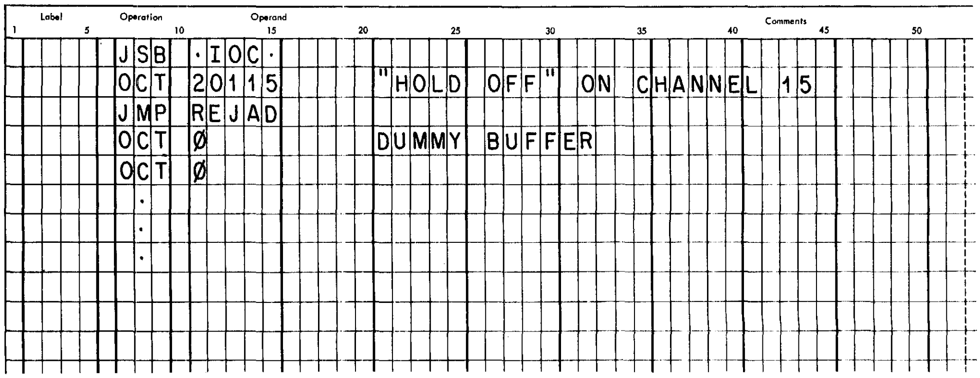
2.8 CALLING SEQUENCE: INSTRUMENTS

2.8.1 DATA SOURCE INTERFACE

A binary output operation causes the removal of "hold-off." The calling influence is as below:

```
JSB      .IOC.
OCT      <function><subfunction><unit-reference>
{ JSB }
{ JMP }  reject address
OCT      0    dummy buffer
OCT      0    buffer length
```

Example:

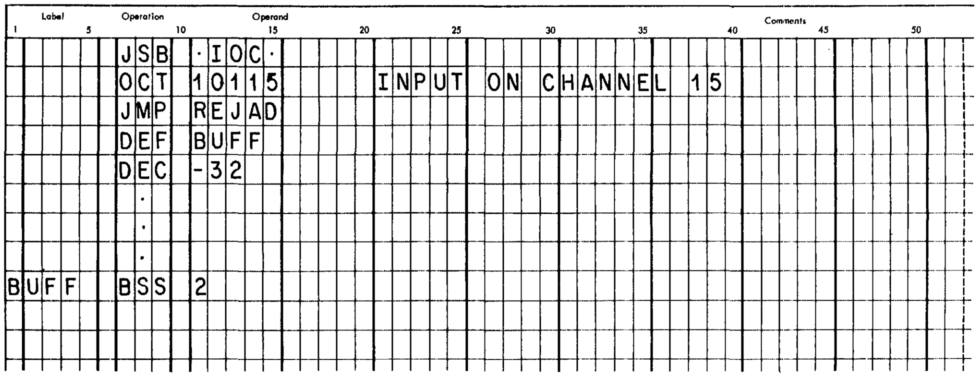

Label
Operation
Operand
Comments
1
5
10
15
20
25
30
35
40
45
50

```
      JSB .IOC.
      OCT 20115     "HOLD OFF" ON CHANNEL 15
      JMP REJAD
      OCT Ø         DUMMY BUFFER
      OCT Ø
       .
       .
       .
```

A binary input operation must have a 2-word buffer. Thirty-two bits (8 BCD characters) are read directly into the 2-word buffer.

```
                   JSB      .IOC.
                   OCT      <function><subfunction><unit-reference>
                 { JSB }
                 { JMB }    reject address
                   DEF      buffer address
                   DEC      -32
                    .
                    .
                    .
buffer address     BSS      2
```

Example:



```
      JSB .IOC.
      OCT 10115     INPUT ON CHANNEL 15
      JMP REJAD
      DEF BUFF
      DEC -32
       .
       .
       .
BUFF  BSS 2
```

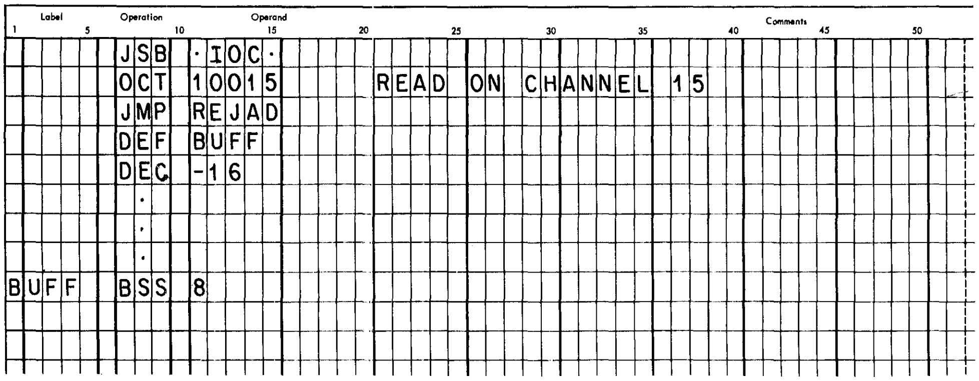
Label
Operation
Operand
Comments
1
5
10
15
20
25
30
35
40
45
50

An ASCII input operation must have an 8 word buffer. 8 BCD characters are converted into 16 ASCII characters in the following format:

$$rf\cdot d_5d_4d_3d_2d_1d_{\emptyset}E - ss_{\wedge\wedge}gg$$

	r	range - a negative power of 10
	f	function
	d_5-$d_{\emptyset}$	six digit data value
	E-ss	range expressed as an exponent of two digits
	∧∧	two blanks
	gg	function expressed as a two-digit number
	JSB	.IOC.
	OCT	<function><subfunction><unit-reference>
	JMP JSB	reject address
	DEF	buffer address
	DEC	-16
	⋮	
buffer address	BSS	8

Example:


Label
1
5
Operation
10
Operand
15
20
25
30
35
40
Comments
45
50

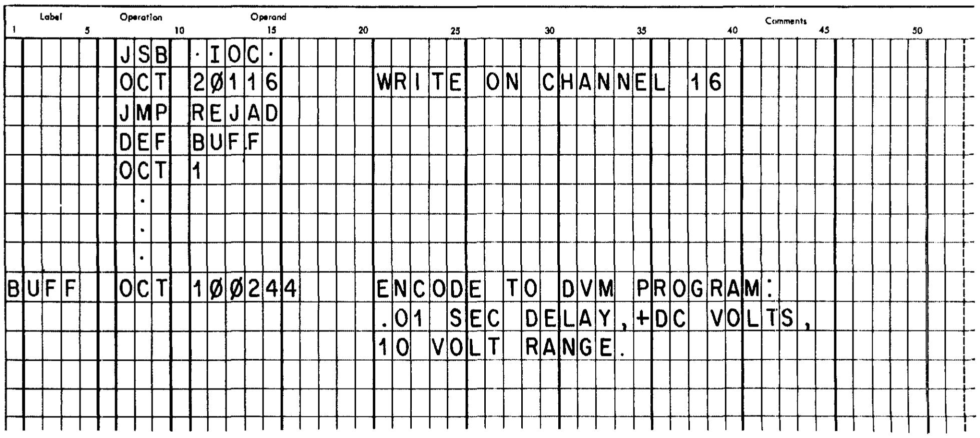
```
      JSB .IOC.
      OCT 10015     READ ON CHANNEL 15
      JMP REJAD
      DEF BUFF
      DEC -16
       .
       .
       .
BUFF  BSS 8
```

2.8.2 DIGITAL VOLTMETER PROGRAMMER

A write request for the Digital Voltmeter Programmer requires that a one-word buffer be specified. This word contains the voltmeter program: sample period (bit 7-6), function (bits 5-3), and range (bits 2-0). If bit 15 contains a 1, encode command is sent to the Voltmeter (bit 15 always be 0 if the configuration includes a Scanner).

```
                 JSB    .IOC.
                 OCT    <function><subfunction><unit-reference>
                 JSB
                 JMP    reject address
                 DEF    buffer address
                 OCT    1
                  .
                  .
                  .
buffer address   OCT    voltmeter program
```

Example:


Label
1
5
Operation
10
Operand
15
20
25
30
35
40
Comments
45
50

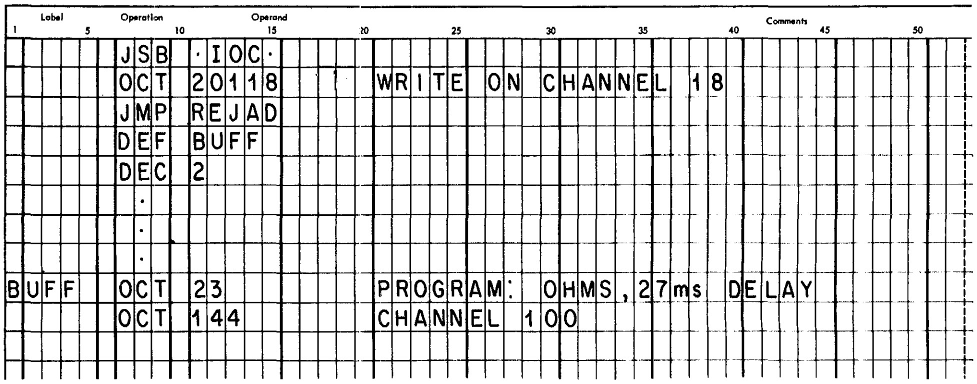
```
      JSB .IOC.
      OCT 2Ø116          WRITE ON CHANNEL 16
      JMP REJAD
      DEF BUFF
      OCT 1
       .
       .
       .
BUFF  OCT 1ØØ244         ENCODE TO DVM PROGRAM:
                         .01 SEC DELAY,+DC VOLTS,
                         10 VOLT RANGE.
```

2.8.3 SCANNER PROGRAMMER

A write request for the Scanner Programmer requires a 2-word buffer. The first word contains the scanner program: the function (bits 4-3) and the delay (bits 2-0). The second word contains the channel number for the start of the scan. The driver subroutine converts the binary channel number value produced by the Assembler to the BCD format required by the device.

```
                 JSB    .IOC.
                 OCT    <function><subfunction><unit-reference>
                 JSB
                 JMP    <reject address>
                 DEF    buffer address
                 DEC    2
                  .
                  .
                  .
                 OCT    xx   scanner program
                 OCT
buffer address   DEC    xxx  starting channel number
```

Example:



```
Label     Operation   Operand             Comments
      JSB .IOC.
      OCT 20118     WRITE ON CHANNEL 18
      JMP REJAD
      DEF BUFF
      DEC 2
       .
       .
       .
BUFF  OCT 23        PROGRAM: OHMS,27ms DELAY
      OCT 144       CHANNEL 100
```

2.8.4 INSTRUMENT CLEAR REQUEST:

A clear request on one of the instrument drivers follows the standard form:

```
JSB      .IOC.
OCT      <function><unit-reference>
<normal return>
```

where the function code = 00.

The request will result in the following conditions:

Data Source Interface – A clear request causes no action. It is included for compatibility only.

Digital Voltmeter Programmer – A clear request to this driver will remove the present program from the DVM but the program will not be destroyed.

Crossbar Scanner	– A clear request will inhibit the STEP or RESET command on the Scanner programmer driver.

INSTRUMENT STATUS REQUEST: No status information is available from the instrument drivers.

2.9 ERROR CONDITIONS DURING EXECUTION

Illegal conditions encountered during .IOC. request processing are termed irrecoverable and cause a halt.† Diagnostic information is displayed in the A- and B-Registers at the time of the halt.

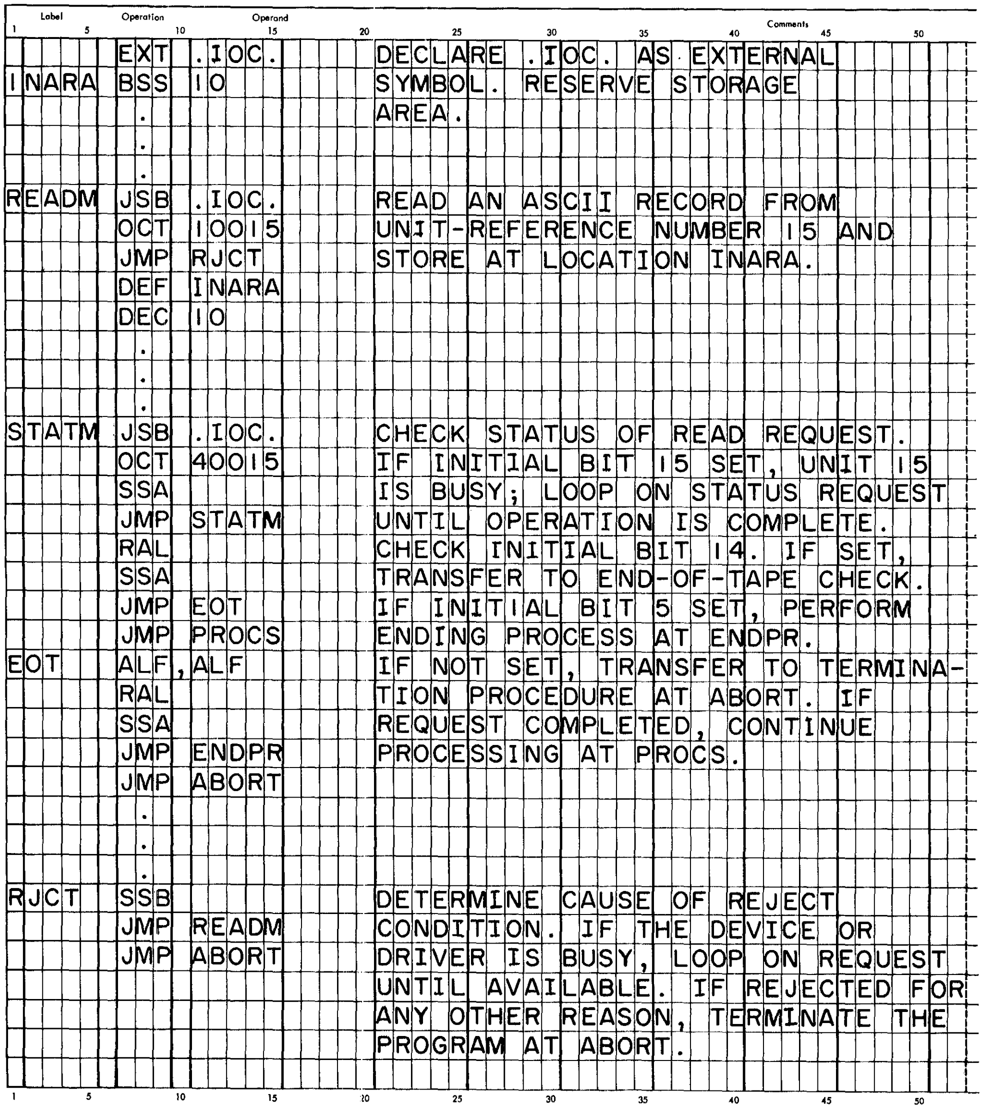
The B-Register contains the absolute location of the JSB instruction of the request call containing the illegal condition.

The A-Register contains a code defining the illegal condition:

A-Register	Explanation
000000	Illegal request code.
000001	Illegal unit-reference number in request.
000002	The Standard unit requested is not defined as a particular device in the Equipment Table.

†The halt is at the absolute location assigned to the symbol IOERR during Prepare Control System processing.

Examples:



```
      EXT .IOC.     DECLARE .IOC. AS EXTERNAL
INARA BSS 10        SYMBOL. RESERVE STORAGE
       .            AREA.
       .
       .
READM JSB .IOC.     READ AN ASCII RECORD FROM
      OCT 10015     UNIT-REFERENCE NUMBER 15 AND
      JMP RJCT      STORE AT LOCATION INARA.
      DEF INARA
      DEC 10
       .
       .
       .
STATM JSB .IOC.     CHECK STATUS OF READ REQUEST.
      OCT 40015     IF INITIAL BIT 15 SET, UNIT 15
      SSA           IS BUSY; LOOP ON STATUS REQUEST
      JMP STATM     UNTIL OPERATION IS COMPLETE.
      RAL           CHECK INITIAL BIT 14. IF SET,
      SSA           TRANSFER TO END-OF-TAPE CHECK.
      JMP EOT       IF INITIAL BIT 5 SET, PERFORM
      JMP PROCS     ENDING PROCESS AT ENDPR.
EOT   ALF,ALF       IF NOT SET, TRANSFER TO TERMINA-
      RAL           TION PROCEDURE AT ABORT. IF
      SSA           REQUEST COMPLETED, CONTINUE
      JMP ENDPR     PROCESSING AT PROCS.
      JMP ABORT
       .
       .
       .
RJCT  SSB           DETERMINE CAUSE OF REJECT
      JMP READM     CONDITION. IF THE DEVICE OR
      JMP ABORT     DRIVER IS BUSY, LOOP ON REQUEST
                    UNTIL AVAILABLE. IF REJECTED FOR
                    ANY OTHER REASON, TERMINATE THE
                    PROGRAM AT ABORT.
```

RELOCATING LOADER 3

The Loader is the module of the Basic Control System that provides the capability of loading, linking, and initiating the execution of relocatable object programs produced by the Assembler, FORTRAN, and ALGOL. It is available in 4K and 8K versions. ALGOL programs and the Program Library stored on magnetic tape require the 8K loader.

3.1 EXTERNAL FORM OF LOADER

The Loader is stored in an absolute record format on an external medium with the Input/Output Control subroutine, .IOC., and the equipment driver subroutines. It is loaded by the Basic Binary Loader. The external medium is determined by the type of device assigned as the Standard Input unit. For the Punched Tape Reader or the Teleprinter, the medium is 8-level paper tape.

3.2 INTERNAL FORM OF LOADER

The Loader is located in high-numbered memory along with the Input/Output Control subroutine and the equipment driver subroutines. The Loader uses .IOC. for input/output operations; it refers to the Standard input and output units. The binary object program is read from the Standard Input unit; comments to the user (e.g., Loader diagnostics) are written on the Teleprinter Output unit; and library routines referenced by the object program are assumed to be on the Program Library unit.

3.3 RELOCATABLE PROGRAMS

The process of assembling or compiling a set of symbolic source program statements may be specified to result in the generation of a relocatable object program. A relocatable program assumes a starting location of 00000. Location 00000 is termed the relative, or relocatable origin. The absolute origin (also termed the relocation base) of a relocatable program is determined by the Loader. The value of the absolute origin is added to the zero-relative value of each operand address to obtain the absolute operand address. The absolute origin, and thus the values of every operand address, may vary each time the program is loaded.

A relocatable program may be made up of several independently assembled or compiled subprograms. Each of the subprograms would have a relative origin of 00000. Each subprogram is then assigned a unique absolute origin upon being loaded. Subprograms executed as a single program may be loaded in any order. The absolute origins will differ whenever the order of loading differs.

The operand values produced by the Assembler, FORTRAN, or ALGOL may be program relocatable, base page relocatable, or common relocatable. Each of these segments of the program has a separate relocation base or origin. Operands that are references to locations in the main portion of the program are incremented by the program relocation base; those referring to the base page, by the base page relocation base; and those referring to common storage, by the common relocation base.

If the Loader encounters an operand that is a reference to a location in a page other than the "current" page, a link is established through the base page. A word in the base page is allocated to contain the full 15-bit address of the referenced location. The address of the word in the base page is then substituted as an indirect address in the instruction in the "current" page. If other similar references are made to the same location, they are linked through the same word in the base page.

3.4 RECORD TYPES

The Loader processes three to five record types for a program. These record types are produced by the Assembler, FORTRAN, or ALGOL in the following sequence:

NAM	Name record
ENT	Entry point record
EXT	External name record
DBL	Data block record
END	End record

The NAM, DBL, and END records exist for every object program; ENT and EXT appear only if the corresponding pseudo instructions are used in the source program.

NAM

The NAM record contains the name of the program and the length of the main, base page, and common segments. The NAM record signifies the beginning of the object program.

ENT

The ENT record defines the names of 1 to 14 entry points within this program. Each of the four-word entries in the record contains the name, the relocatable address of the name; and an indicator which specifies whether the address is program or base page relocatable.

EXT

The EXT record contains from 1 to 19 three-word entries which specify the external references defined in the program. The three words allow a maximum of five ASCII characters for the symbol and a number used by the Loader to identify the symbol.

DBL

A DBL record contains 1 to 45 words of the object program. It indicates the relative starting address for the string of words and whether this portion of the object code is part of the main program or base page segment. For each of the words there is also a relocation indicator which defines the relocation base to be applied to each operand value. Possible relocation factors are:

Absolute	Operand is an absolute expression or constant. There is no relocation base.
15-bit Program Relocatable	Operand is a 15-bit value to which is added the program relocation base.
15-bit Base Page Relocatable	Operand is a 15-bit value to which is added the base page relocation base.

15-bit Common Relocatable

Operand is a 15-bit value to which is added the common relocation base.

External Symbol Reference

Operand is a reference to an external symbol. Value is supplied when the Loader determines the absolute location of the linkage word in the Base Page which contains the 15-bit address of the related entry point.

Memory Reference Instruction

A memory reference instruction in the form of a two-word group which consists of the instruction code, a full 15-bit operand address, and a relocation indicator for the operand address. The relocation indicator can define the operand address to be program, base page, or common relocatable.

END

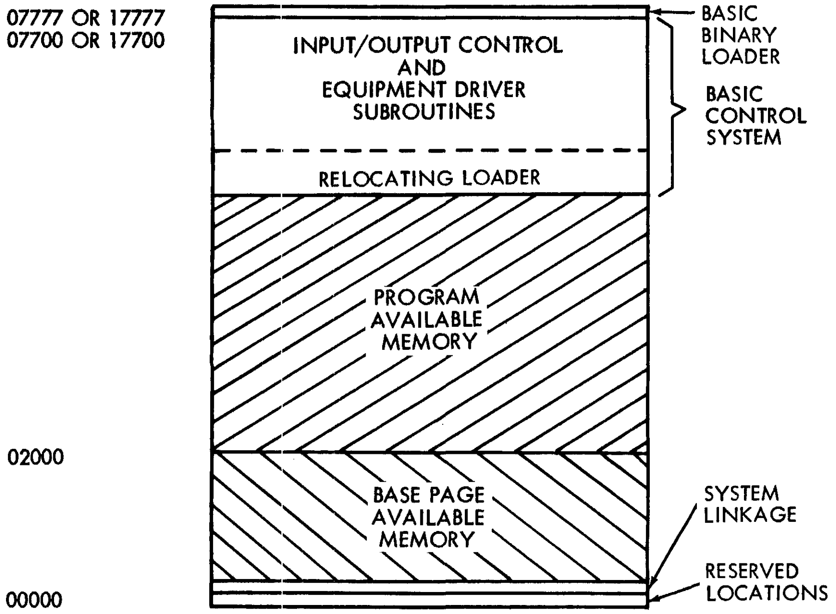
The END record terminates the block of records in an object program. The END record may contain a 15-bit address which is the location to which control is transferred by the Loader to begin program execution.

3.5 MEMORY ALLOCATION

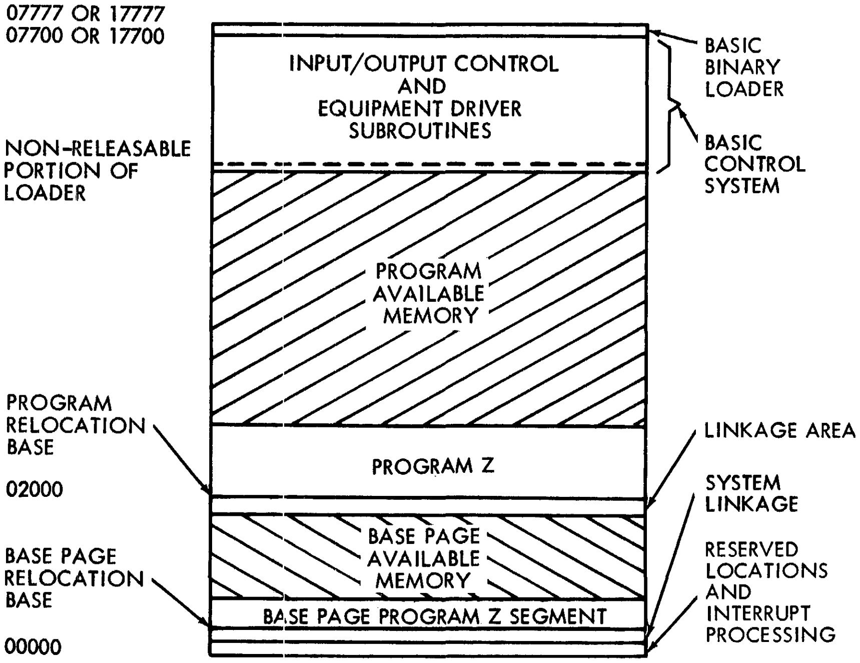
The Loader loads the object program into available memory. Available memory is defined as that area of memory not allocated for hardware and system usage. Available memory is divided into two segments:

> Available Memory in Base Page – used for the operand linkage area, program blocks origined into the Base Page by the Assembler pseudo instruction ORB, and for program blocks assigned to the Base Page by the Loader when the amount of program available memory is insufficient.
>
> Program Available Memory – used for the main body of the program and may be used by the common block should the area used by the Loader be insufficient.

Prior to loading the object program, memory is allocated as follows:



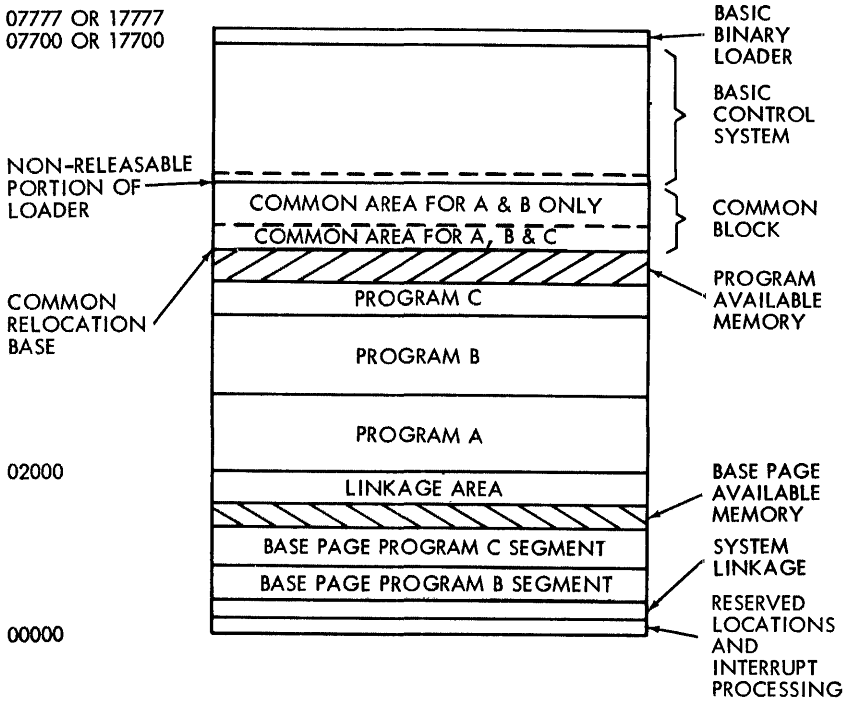
Assuming Program Z is to be loaded and executed – after loading, the memory might be allocated as follows:


07777 OR 17777
07700 OR 17700
NON-RELEASABLE PORTION OF LOADER
INPUT/OUTPUT CONTROL AND EQUIPMENT DRIVER SUBROUTINES
PROGRAM AVAILABLE MEMORY
PROGRAM RELOCATION BASE
PROGRAM Z
02000
BASE PAGE AVAILABLE MEMORY
BASE PAGE RELOCATION BASE
BASE PAGE PROGRAM Z SEGMENT
00000
BASIC BINARY LOADER
BASIC CONTROL SYSTEM
LINKAGE AREA
SYSTEM LINKAGE
RESERVED LOCATIONS AND INTERRUPT PROCESSING

Options selected during PCS processing can define the equipment driver subroutines and other system routines as relocatable programs. If selected, these routines would be allocated to the available memory areas, and the length of the absolute segment of BCS reduced accordingly.

If several programs are to be loaded and executed together, the following might occur:

> Assume three programs, A, B, and C, comprise a running program. Programs A and B share a common block, a portion of which is also shared by C. Programs B and C contain segments which are designated to be allocated to the Base Page. Allocation is as follows:



Common Block Allocation

The first common length declaration (i.e., the first program containing a common segment) processed by the Loader establishes the total common storage allocation in high memory overlaying the major portion of the area occupied by the Loader. Subsequent programs must contain common length declarations which are less than or equal to the length of the first declaration.

To allocate the common area, the Loader subtracts the total length of the block from the address of the last releasable word in the Loader. The resulting memory address +1 is the origin of the common block. This value is used throughout the entire loading process as the common relocation base.

Program Storage

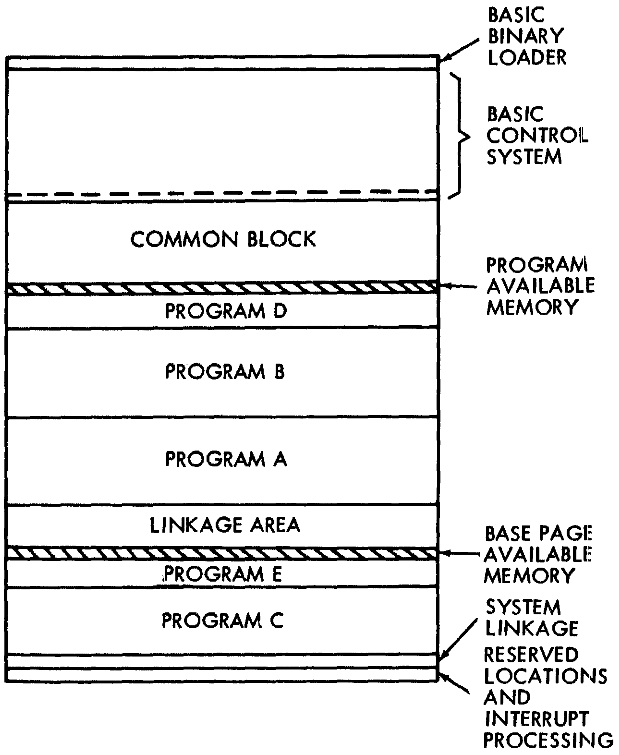
The program length is compared with the amount of available memory. If sufficient space is available, the program is loaded and the upper and lower bounds recorded. If the program has a Base Page segment, or if the program consists entirely of coding to be stored in the Base Page, the length of the segment is compared to the amount of available base page memory. If there is enough space in this area, the segment is loaded and the bounds recorded. The initial main program segment is usually origined at absolute location 02000 (page 1, Module 0). The initial Base Page segment is usually origined immediately following the area set aside for Reserved Locations, Interrupt Processing, and System Linkage. Subsequent main program and Base Page segments are loaded into the next available higher numbered areas contiguous to the previously loaded segments.

Providing the Memory Allocation List option is selected (see Operating Procedures) the name of each program, its upper and lower bounds, and its Base Page upper and lower bounds are printed after the program is loaded. The format is as follows:

```
<program name>
lllll uuuuu (main program bounds)
lllll uuuuu (Base Page bounds)
```

If the Loader finds that the main program segment about to be loaded can not fit in the memory area available for the main segment, it compares the segment's length to the length of available memory in the Base Page. If there is sufficient space, the main segment will be loaded in the Base Page. The next segment will be loaded in the main program area if it will fit, or in the Base Page if not (providing there is sufficient space in the Base Page).

For example, assume that several programs to be loaded in sequence A, B, C, D, E, and have sizes such that they can not all fit in the main program available memory.



3.6 OBJECT PROGRAM RECORD PROCESSING

ENT/EXT Record Processing

The Loader constructs and maintains a Loader Symbol Table which contains entry points and external symbols which are declared in the programs and entry point names of any BCS system subroutines that have been defined as relocatable. As each entry point is encountered its relocated (absolute) address is recorded in the table. As each external reference is processed, a link word is established in the Base Page. The general processing of the entries in an ENT and EXT record involves searching the Table to locate a match between the symbols. When a match is found, the absolute entry point address is stored in the Base Page link word.

The Loader assumes that there is a user program, BCS system routine, or Program Library routine entry point for every external reference. If none exists, the external reference is undefined and considered to be in error. A list of undefined external symbols is printed at the end of the loading operation. If duplicate entry points are detected, a diagnostic is issued. A user entry point duplicated by a Program Library routine results in the library routine being ignored.

Each entry in the Loader Symbol Table occupies five words. The Table is positioned before the beginning of the Loader and extends backwards toward low-numbered memory. If sufficient space is not available in the main program portion of memory to store a five-word entry, a diagnostic message is issued and the loading operation is terminated.

DBL Record Processing

A load address for the data or instruction words in a DBL record is relocated by adding either the program relocation base or the base page relocation base. The resulting value is the absolute address for storing the first word. The second word is stored at address +1, the third at address +2 and so forth. A relocation base is added to each operand address as specified by the relocation indicator.

The processing for an external reference word involves a search of the Loader Symbol Table for the related entry. When found, the address of the link location in the Base Page is extracted and stored as an indirect address in the instruction.

When a memory reference instruction is processed, the Loader first applies the proper relocation base, (program, base page, or common) to the 15-bit operand address. If the resulting absolute operand address references the Base Page, the address (bits 09-00) is set into the operand field and the instruction is stored in memory at the current load address. When the absolute operand address and the current load address are in the same page, the operand address is truncated to bits 09-00 and set as the instruction operand address. If the operand address is in a page other than the current load address page, the operand address is stored in the Linkage area of the Base Page and a reference to this location set as an indirect address in the operand field of the instruction.

A memory overflow condition can occur when insufficient space is available in the base page to allocate a linkage word. A diagnostic message is issued and the loading operation is terminated.

END Record Processing

When an END record is encountered, the Loader determines if it contains a transfer of control address. If it does, the address is saved.

If loading is from the Program Library and there exist no undefined external references, the End-Of-Loading operation is performed.

If loading is from the Standard Input unit or Program Library unit and if undefined external references exist, the Loader requests the next record. If the next record is a NAM record, processing of the next program begins. If the result of the request is an End-of-Information indication, an End Condition exists.

Program Library Loading

Loading from the Program Library differs from loading of user programs in that only those programs in the Library which contain entry points matching undefined external symbols in the Loader Symbol Table will be loaded. After each program is loaded from the Library, the Loader Symbol Table is checked for undefined symbols. If none exist, the loading operation is complete and the.program is ready to be executed.

End Condition

When the Loader requests input and no data is available on the input device an End Condition exists. The Loader acknowledges this condition by writing the message "LOAD" on the Teleprinter Output device. The user responds to this message by setting switches 2-0 of the Switch Register (see Operating Procedures). Four replies are available:

a. Load next program from Standard Input unit. Relocatable BCS system subroutines are considered to be part of the program and must be loaded from the Standard Input unit (unless they are made part of Program Library tape).

b. All programs are loaded; proceed to the end-of-loading operation.

c. Terminate loading operation. This forces program execution even though there may be undefined external references.

d. Load from Program Library unit; all user programs are loaded.

End-of-Loading Operation

The end of loading is signaled by the second or fourth response to an End Condition. The Loader then searches the Loader Symbol Table for any undefined external references. Any such undefined external symbols are written on the Teleprinter Output unit and the "LOAD" message is repeated.

When the loading operation is completed, or when the user has requested termination of the Loading process, the Loader produces a memory allocation list. (This list may be omitted; see Operating Procedures.) The format of the list is as follows:

```
<symbol 1>    aaaaa

<symbol 2>    aaaaa
     .
     .
     .
<symbol n>    aaaaa
```

The symbols are the entry points in the user's program, the Basic Control System, or the Program library and the a's are their absolute addresses.

If a common block was allocated, the lower and upper bounds of the block are listed as follows:

```
*COM      lllll      uuuuu
```

The bounds of the Linkage Area are listed as follows:

```
*LINKS     lllll     uuuuu
```

The l's are the absolute lower bounds and the u's are the absolute upper bounds.

3.7 PROGRAMMING CONSIDERATIONS

When a program has been completely loaded, its execution is initiated by performing a Jump Subroutine to the transfer address (from the last END record containing an address). The initial contents of the transfer address should be a NOP, OCT 0, etc., not the first executable instruction of the program.

3.8 LOADER OPERATING PROCEDURES

The exact operating procedures for the loader depend on the available hardware configuration and the construction of the Basic Control System through use of the Prepare Control System routine. The user should know the assignment of input/output equipment and memory size before using the Loader.†

†As established when configuring BCS.

Loading Options

The Basic Control System Loader is designed to load one or more tapes containing relocatable programs. The message "LOAD" is typed when an end-of-tape condition is encountered. The user then loads the next tape; indicates loading from the program library, specifies that loading is complete, etc. When all programs are loaded and no undefined external references remain, the Loader types the message "LST" allowing the user to bypass part of the Memory Allocation List. Following the response, the Loader types the message "RUN". The user then initiates program execution.

Memory Allocation List

A memory Allocation List may be obtained for the programs being loaded. The list may include the name, main program bounds, and Base Page bounds for each of the programs. This portion of the List may be followed (at the completion of the loading operation) by a list of all entry points and their absolute addresses, the bounds of the common block, and the bounds of the linkage area. The setting of Switch 15 determines the contents of the List.

To obtain the bounds for each program on a tape, Switch 15 must be set to 0 before the tape is loaded (in response to the "LOAD" message). To bypass the program bounds listing, set Switch 15 to 1 before loading the tape. The switch setting may be altered whenever the "LOAD" message is typed.

To obtain the entry point list, the common bounds, and the linkage area bounds, set Switch 15 to 0 in response to the message "LIST", which is printed after all programs are loaded. To bypass this portion of the list, set Switch 15 to 1.

Absolute Binary Output

The user may specify that an absolute binary tape be punched. This option may be selected when it is necessary to utilize the area occupied by the Loader or when an absolute version is desired for "production stage" programs. The process involves a simulated loading operation, however, the absolute program is punched on tape rather than being loaded.

The absolute records produced consist of the relocated programs (including all programs loaded from the Program Library), the Linkage area, all referenced segments of the Basic Control System. These might include:

Input/Output control subroutine (.IOC.)
All input/output equipment drivers
Other BCS system subroutines
Memory Table (.MEM.)
System Linkage Area
Interrupt Processing area
Absolute location 2 and 3

In addition, the Loader Symbol Table, the common and linkage area bounds are punched in ASCII format on the end of the binary tape. Ten inches of feed frames separate the binary instructions and the ASCII data. This feature provides a record of the memory allocation for the absolute program.

At the completion of the "loading" process the Loader types the message "END".

To execute the program, it must be loaded using the Basic Binary Loader. To initiate execution, set 000002 into the P-Register and press RUN. The Loader has stored the transfer address of the program in locations 2 and 3 as follows:

2 contains JMP 3, I
3 contains < transfer address >

Separation of List and Binary Output

If the absolute binary output option is selected and the Teleprinter is used as both a list and punch device, the Loader halts before and after each line is printed to avoid punching the line and altering the binary output.

The halts and related procedures are as follows:

T-Register Contents	Explanation	Action
102055	A line is about to be printed.	Turn punch unit OFF. Press RUN.
102056	A line has been printed.	Turn punch unit ON. Press RUN.

Operating Instructions

The following procedures indicate the sequence of steps for loading and execution of the Basic Control System Loader:

A. Set Teleprinter to LINE and check that all equipment to be used is operable.

B. Load the Basic Control System tape using the Basic Binary Loader:

1. Place the Basic Control System tape in the device serving as the Standard Input unit (e.g., Punched Tape Reader).

2. Set Switch Register to starting address of Basic Binary Loader (e.g., 007700 for 4K memory, 017700 for 8K memory).

3. Press LOAD ADDRESS.

4. Set LOADER switch to ENABLED.

5. Press PRESET.

6. Press RUN.

7. When the computer halts and indicates that the BCS tape is loaded (T-Register contains 102077), set the LOADER switch to PROTECTED.

C. Set Switch Register to 000002, press LOAD ADDRESS, and set Switch Register to 000000.

D. Establish Loader parameters:

1. Set Switch 15 to 1 if <u>no</u> Memory Allocation Listing is desired during first load operation.

2. Set Switch 14 to 1 if an absolute binary tape of the programs is to be punched. (Turn on punch device if this option is selected.)

E. Place relocatable object tape in device serving as Standard Input unit.

F. Press RUN.

During the operation of the Basic Control System Loader, the following halts may occur:

Teleprinter Message	Explanation	Action
LOAD (T-Register contains 102001.)	End-of-tape condition on Standard Input device.	1. To load next tape, set Switches 2-0 to 0_8. If no Memory Allocation Listing of next tape is desired, set Switch 15 to 1. Press RUN to continue loading. 2. To indicate that all programs are loaded and to proceed to the end-of-loading phase, set Switches 2-0 to 1_8. Press RUN.[†] 3. To terminate loading operation, set Switches 2-0 to 2_8. Press RUN. (This forces execution even though undefined external references have not been matched.)

[†] A list of any undefined external symbols is typed following a Switch Register reply of 1_8 or 4_8 to the "LOAD" message. The message "LOAD" is then repeated. The programs containing the matching entry points should be loaded. Loading of user programs from Standard Input must be completed before loading of routines from Program Library.

Teleprinter Message	Explanation	Action
		4. To load from Program Library, set Switches 2-0 to 4_8. If no Memory Allocation Listing of library routines is desired, set Switch 15 to 1. Press RUN to continue loading. When all library routines are loaded, the Loader proceeds directly to the end-of-loading phase.
*LST	The Loader is ready to print the LST, common bounds, and linkage area bounds.	If a list of these items is not desired, set Switch 15 to 1. Press RUN.
*RUN	All programs are loaded and ready for execution.	Check that all I/O devices are ready for operation. Press RUN.
*END	The absolute binary output has been selected and the punched tape is complete.	To execute the program: 1) Load binary tape using Basic Binary Loader as in B. 2) Set Switch Register to 000002. Press LOAD ADDRESS. 3) Press RUN.
*L01	Checksum error: The checksum read on the last record does not agree with the checksum calculated by the Loader.	To re-read record, reposition tape to beginning of record and press RUN.

Teleprinter Message	Explanation	Action
*L02	Illegal record: The last record read was not recognized as one of the five types accepted by the Loader.	To re-read record, reposition tape to beginning of record and press RUN.
*L03	Memory overflow: The length of the main or Base Page portion of the program or the common block exceeds the bounds of available memory.	Irrecoverable error, program must be revised.
*L04	Linkage area overflow: Linkage words supplied by the Loader for references between pages exceed the size of available base page memory.	If program consists of several subprograms, altering the sequence in which the subprograms are loaded may reduce the number of linkage words. Otherwise, irrecoverable error, program must be revised.
*L05	Loader symbol table overflow: The number of EXT/ENT symbols exceed available memory.	Irrecoverable error, program must be revised.
*L06	Common block error: The length of the common block in the current program is greater than the length of the first common block allocated.	Revise sequence in which subprograms are loaded and reload program. Otherwise, revise program.
*L07	Duplicate entry points: An entry point in the current program matches a previously declared entry point.	Irrecoverable error, program must be revised.

Teleprinter Message	Explanation	Action
*L08	No transfer address: The initial starting location (e.g., END statement operand) was not present in any of the programs which were loaded.	To enter the starting address, set the absolute value in the Switch Register, press LOAD A, and press RUN.
*L09	Record out of sequence: ANAM record was encountered before the previous program was terminated with an END record.	Irrecoverable error, program must be revised.

If the absolute binary output option is selected, the following halts may occur:

T-Register Contents	Explanation	Action
102066	Tape supply low on 2753A Tape Punch which is producing absolute binary output. Trailer follows last valid output.	Place new reel of tape in unit. Press RUN. Leader is produced.
102055	A line is about to be printed on Teleprinter output device. (See Separation of List and Binary Output.)	Turn punch unit off. Press RUN.
102056	A line has been printed while Teleprinter punch unit off. (See Separation of List and Binary Output.)	Turn punch unit on. Press RUN.

INPUT/OUTPUT DRIVERS

4.1 GENERAL DESCRIPTION

An I/O driver, operating in the BCS environment, is responsible for controlling all data transfer between an I/O device and the cpu. It operates under control from the program IOC. Its operating parameters are the user I/O request and the information contained in the device associated Equipment Table entry.

4.2 STRUCTURE

An I/O driver is a relocatable program segmented into two closed subroutines, termed the "initiator" and "continuator" sections. The entry point names for these two sections must be "D. nn" and "I. nn", respectively. The numeric value "nn" in the names is the Equipment Type Code assigned to the device. For example, D. ØØ and I. ØØ are the entry points for the Teletype driver; "ØØ" is the Equipment Type Code assigned to a Teletype.

```
                              NAM DRIVER D. nn

                    D. nn

                    ______________________
                    ______________________
Initiator Section   ______________________
                    ______________________
                    ______________________
                    ______________________

                    I. nn

                    ______________________
                    ______________________
Continuator Section ______________________
                    ______________________
                    ______________________
                    ______________________
```

4.2.1 Initiator Section

This section is called directly from IOC with calling parameters including the address of the second word of the user I/O request and the address of the EQT entry for the referenced device. IOC sets these parameters in A and B and performs a JSB to the entry point "D. nn". Return to IOC from this section must be indirectly through D. nn.

On entry to D. nn,

(A) = Address of word 1 of 4-word EQT entry
(B) = Address of word 2 of I/O request

The initiator section of any driver must perform the functions described below.

1) Reject the IOC request and return to IOC (see step 6) if any of the following conditions exist:

   a. the driver is busy operating another device

   b. the referenced device is busy or inoperable

   c. the user request code or other parameters illegal for the device

   d. a DMA channel is not available and DMA is required for data transfer.

2) Extract the parameters from the user I/O request and save them within the driver storage.

3) Configure all I/O instructions in the driver to include the channel number for the reference device.

4) Indicate equipment in operation:

   a. set the "a" field in the EQT entry to 2 (busy) for the device called

   b. set an internal driver "busy" flag for the driver

   c. set a "busy" flag in IOC if a DMA channel is used

(To set a DMA flag in IOC:

Within the IOC program the two entry points DMAC 1, DMAC 2 contain the DMA channel locations (6 and 7 or 7 and 6). The sign bit of the channel used must be set to 1 to indicate that the channel is busy.)

5) Initialize operating conditions and activate the device.

6) Return to IOC with the A and B registers set to indicate initiation or rejection and the cause of the reject:

(A) = Ø, operation initiated
= 1, operation rejected - reason in B-register

(B) = 1ØØØØØ, the device is busy or inoperable, or the driver is busy
= ØØØØØ1, a DMA channel is required but no channel is available
= ØØØØØØ, the request code or sub-function is not legal for the device

4.2.2 Continuator Section

This section is entered by device interrupt to continue or complete an operation. It may also be called from the Initiator Section to begin an operation. The entry point to this section is I. nn. There are no parameters on entry.

The continuator section of any driver must perform the functions described below.

1) Save all registers which will be used by the continuator section.

2) Perform the input or output of the next data item. If the transfer is not completed, restore the "saved" register and return control to the program (see steps 5 and 6).

NOTE: A driver for a device which inputs or outputs data independent of program control such as DMA would not include step 2. The device is turned "on" by the initiator section (step 5) and the data transfer is immediately accomplished. The continuator section for such drivers merely completes the input or output operation.

3) When data transfer is completed (end-of-operation) or if a device malfunction is detected, set the following information in the EQT entry:

> The number of words or characters transferred (corresponding to the request) is set as a positive value in word 3. Bit 15 of word 3 is set to Ø or 1 to indicate the mode of transfer.
>
> The device status, actual or simulated, is set in bits 07-00 of word 2 and the "a" field (bits 15-14) in word 2 set to:
>
> 0 - device available (not busy)
> 1 - device available; the operation is complete but an error has been detected
>
> Bits 13-08 of word 2 must not be altered.

4) Clear all "busy" indicators. Clear the driver busy flag. If a DMA channel was used clear the flag in IOC.

5) Restore all registers saved at the entry.

6) Return indirectly through the entry point I. nn, with the following exception:

If end-of-operation and the operation completed was an output or Function request, return must be made to the entry point ".BUFR" in IOC. This enables the Buffered version of IOC to perform the automatic output buffering function. The standard version of IOC at this entry point just performs a normal return to the point of interruption. The calling sequence to .BUFR is:

```
         EXT   .BUFR
(P)      JSB   .BUFR
(P+1)    NOP          (holds return address from I. nn)
(P+2)    NOP          (holds EQT entry address)
```

PREPARE CONTROL SYSTEM 5

The Prepare Control System (PCS) program processes relocatable modules of the Basic Control System and produces an absolute version designed to work on a specific hardware configuration. It creates operating units of the Input/Output Control subroutine (.IOC.), the equipment driver subroutines, and the Relocating Loader. It also establishes the contents of certain locations used in interrupt handling. Options are available to define the equipment driver modules and other BCS system subroutines as relocatable programs to be loaded with the user's object program.

The Prepare Control System is an absolute program which is loaded by the Basic Binary Loader. It operates on a minimum configuration of 4K memory and a 2752A Teleprinter. However, if a 2737A Punched Tape Reader and a 2753A Tape Punch are available, the Prepare Control System will utilize these devices; PCS requests their assignment during the Initialization phase.

After the Initialization phase is completed, each module of BCS is loaded and processed by PCS. The order in which the modules are processed is not significant except that the BCS Loader must be the last module loaded. Two modules, the Input/Output Control subroutine and the Loader, require that parameters be entered via the Keyboard Input unit after being loaded.

5.1 INITIALIZATION PHASE

During the Initialization phase, the System requests that the operator provide the channel assignments of the Punched Tape Reader and the Tape Punch if available. This is followed by a request for the first and last words of available memory. The first word is the location in the Base Page following the locations required for interrupt processing (the interrupt locations and the locations containing the addresses of the Interrupt Processors). This location defines the start of the BCS System Linkage area. The last word of available memory is usually the location prior to the protected area (e.g., 7677 for 4K memory, 17677 for 8K memory).

Example:

```
HS INP?          message
10               reply
HS PUN?          message
11               reply
FWA MEM?         message
30               reply
LWA MEM?         message
17677            reply
   .
   .
   .
```

5.2 LOADING OF BCS MODULES

After the Initialization phase is completed, the System types the message "LOAD". The modules of BCS are then loaded using the Punched Tape Reader, if available, or the Teleprinter reader. The modules may include .IOC., the equipment drivers, and the Relocating Loader. They may be loaded in any order provided that the Relocating Loader is last. The message is repeated after each module is loaded until the Loader has been processed. Diagnostics are printed if certain error conditions occur during the loading. The absolute lower and upper bounds of each program within BCS are listed after the program is loaded.

The format is as follows:

```
<program name>
lllll  uuuuu
```

Equipment driver subroutines and interrupt processing sections which are to be used in relocatable form are identified during PCS processing but are not loaded. At the completion of the processing, PCS requests the missing subroutines. The proper response identifies each as relocatable.

5.3 INPUT/OUTPUT EQUIPMENT PARAMETETERS

After the Input/Output Control module is loaded, PCS requests the information needed to construct the Equipment Table (EQT) and Standard Equipment Table (SQT).†

Equipment Table Statements

PCS first types the messages "TABLE ENTRY" and "EQT". The operator responds by supplying the Equipment Table entries in the following format:

nn, D.ee [,D] [,Uu]

nn The channel number (select code) for the device. For a device connected to two or more channels, nn is the lower numbered channel.

D.ee The Basic Control System symbolic name for the related equipment driver subroutine. ee is the equipment type code used by BCS. Driver names are as follows:

D. 00 – 2752A Teleprinter
D. 01 – 2737A Punched Tape Reader
D. 02 – 2753A Tape Punch
D. 20 – Kennedy 1406 Incremental Tape Transport
D. 21 – 2020A Magnetic Tape Unit
D. 22 – 3030A Magnetic Tape Unit
D. 40 – Data Source Interface
D. 41 – Integrating Digital Voltmeter
D. 42 – Guarded Crossbar Scanner
D. 43 – Time Base Generator

D A Direct Memory Access channel is required to operate the device.

Uu The physical unit number u (0-7) for addressing the device if it is attached to a multi-unit controller.

The same response is used regardless of whether the related subroutine driver is to be relocatable or absolute (part of BCS). If the driver is not encountered during processing, PCS prints the following:

```
I/O DRIVER?
D•EE
```

† See Appendices B and C for description of EQT and SQT.

A response of "!" indicates that the driver is to be in relocatable form. (Any other response at this time is an error.)

The order in which the EQT statements are submitted defines the position of the entry in the Equipment Table. It also establishes the unit-reference number that the programmer uses in writing input/output requests to .IOC. The first statement entered describes the unit which is to be referenced as number 7_8; the second statement, number 10_8; the third statement, number 11_8; etc. Numbers 1 through 6 are reserved for Standard unit definition in the Standard Equipment Table.

The statement "/E" is entered to terminate the EQT input.

Example:

		Unit-Reference Number
*TABLE ENTRY	Message	
EQT?	Message	
10,D.01	Statement	7
11,D.02	Statement	10
12,D.00.	Statement	11
/E	Terminator	

Standard Equipment Table Statements

In constructing the Standard Equipment Table, PCS types a mnemonic for the Standard unit and waits for the reply. The reply consists of the unit-reference number for a device previously described in the Equipment Table.

Example:

SQT?	message	
-KYBD?	message to assign Keyboard Input	
11 -TTY?	reply:	unit-reference number for Teleprinter message to assign Teleprinter Output
11 -LIB?	reply:	unit-reference number for Teleprinter message to assign Program Library

7	reply: unit-reference number for Punched Tape Reader
-PUNCH?	message to assign Punch Output
10	reply: unit-reference number for Tape Punch
-INPUT?	message to assign Input
7	reply: unit-reference number for Punched Tape Reader
-LIST?	message to assign List Output
11	reply: unit-reference number for Teleprinter

Direct Memory Access Statement

After the equipment tables are completed, PCS requests information about the availability of DMA channels to be controlled by the Input/Output Control and equipment driver subroutines. PCS types the message "DMA?" and the operator responds with the available DMA channel numbers. The format of the reply is:

$$c_1\ [,c_2]$$

c_1 is 6 if one channel is available

c_2 is 7 if the second channel is available

If no DMA channel is available, the reply is 0 (zero).

Example:

DMA?	message
6,7	reply for two channels

If the reply contains any characters other than 6 or 7 it is considered to be in error and a diagnostic is issued.

5.4 INTERRUPT LINKAGE PARAMETERS

After the Relocating Loader is loaded, PCS requests the parameters needed to set the Interrupt Linkage for Input/Output processing. The information required for each device includes:

> The interrupt location within the Reserved Location area in low core.
>
> The entry point name of the interrupt processing section in the equipment driver subroutine for the device.
>
> The address of the word in the Base Page which is to contain the 15-bit absolute address of this entry point name.

The same response is used regardless of whether the subroutine driver is to be relocatable or absolute (part of BCS). If the entry point was not encountered during processing, PCS prints the following:

```
*UN NAME
```

A response of "!" indicates that the driver is to be in relocatable form. (Any other response at this time redefines the linkage.)

Given this information, PCS sets in the interrupt location a Jump Subroutine (Indirect) to the word holding the absolute address for the entry point of the Interrupt Processor.

Location	Content
10	JSB 20B,I
.	.
.	.
.	.
20	DEF I.01

10 is the interrupt location

20 holds the address of the entry point, I.01, of the Interrupt Processor.

PCS types the message "INTERRUPT LINKAGE?" The operator responds with a message in the following format:

$$a_1, a_2, \text{I.ee}$$

a_1 The address in low core of the interrupt location for the device (channel).

a_2 The address in the Base Page of the word to contain the absolute address of the Interrupt Processor entry point.

I.ee The entry point name of the Interrupt Processor section of the equipment driver subroutine. ee is the equipment type code used by BCS. Entry point names are as follows:

I.00 – 2752A Teleprinter
I.01 – 2737A Punched Tape Reader
I.02 – 2753A Tape Punch
I.20 – Kennedy 1406 Incremental Tape Transport
I.21 and C.21† – 2020A Magnetic Tape Unit
I.22 and C.22† – 3030A Magnetic Tape Unit
I.43 – Time Base Generator

The statement "/E" is entered to terminate the Interrupt Linkage parameter input.

Example:

`INTERRUPT LINKAGE?`	message	
`10,20,I.01`	reply:	The Punched Tape Reader uses interrupt location 10. The absolute address for entry point I.01 is location 20 in the Base Page.
`11,21,I.02`	reply:	The Tape Punch uses interrupt location 11. The address of I.02 is at location 21.

† Both the magnetic tape systems are connected to two channels; the lower numbered channel transfers data (D.21, D.22); the higher numbered channel transfers commands (C.21, C.22).

12,22,I.00 13,22,I.00	reply:	The Teleprinter, which requires two channels, uses interrupt locations 12 and 13. The Interrupt Processor entry point address is stored at location 22.
/E		Terminates linkage parameters.

The response to the "INTERRUPT LINKAGE?" message may have the following form if a constant, for example a halt, is to be set in the interrupt location.

a, c

a The address in low core of the interrupt location for the device (channel).

c The constant in octal form that is to be stored at location a.

Example:

INTERRUPT LINKAGE?	message	
27,102027	reply:	A halt executed when interrupt occurs on channel 27.
26,0	reply:	A NOP is executed when interrupt occurs on channel 26; the program resumes normal execution.

5.5 PROCESSING COMPLETION

When the Interrupt Linkage parameters have been supplied, PCS performs the following functions:

1. Prints the message "*UNDEFINED SYMBOL" followed by the entry point names of all system subroutines which have been referenced as externals but not loaded. At this point, PCS may be rerun and the missing subroutines loaded or, the symbols may be added to the Relocating Loader's Loader Symbol Table. Undefined symbols are assigned a value of 77777 for an absolute address.

2. Completes the construction of the Loader Symbol Table.
3. Sets the Memory Table (symbolic location .MEM.) in the Relocating Loader to reflect the final bounds of available memory.

Following this, PCS prints a list of all Basic Control System entry points and the bounds of the System Linkage area in the Base Page.

Example:

```
.SQT.   17472
.EQT.   17500
.IOC.   17515
DMAC1   17676
DMAC2   17677
IOERR   17656
XSQT    17674
XEQT    17675
D.00    16745
I.00    17107
D.01    16406
I.01    16521
D.02    16115
I.02    16226
.LDR.   15413
HALT    16110
.MEM.   16110
LST     14102


*SYSTEM LINK
00030  00071
```

The final step in PCS processing is the punching of an absolute binary tape of the configured Basic Control System. This tape can be loaded by the Basic Binary Loader. When the tape has been punched, BCS types the message "*BCS ABSOLUTE OUTPUT". At the completion of the PCS run, the message "*END" is typed. The tape is punched using the Tape Punch unit if available, or the Teleprinter punch.

5.6 OPERATING PROCEDURES

The following procedures indicate the sequence of steps for loading and execution of the Prepare Control System.

A. Set Teleprinter to LINE and check that all equipment to be used is operable.

B. Load the Prepare Control System tape using the Basic Binary Loader.

   1. Place the Prepare Control System tape in the device serving as the Standard Input unit (e.g., Punched Tape Reader).

   2. Set Switch Register to starting address of Basic Binary Loader (e. g. , 007700 for 4K memory, 017700 for 8K memory).

   3. Press LOAD ADDRESS.

   4. Set LOADER switch to ENABLED.

   5. Press PRESET.

   6. Press RUN.

   7. When the computer halts and indicates that the PCS tape is loaded (T-Register contains 102077) set the Loader Switch to Protected.

C. Set Switch Register to 002000, press LOAD ADDRESS.

D. Set Switches 5-0 to the value of the channel number of the Teleprinter. (On a two channel teleprinter use the lower numbered channel.)

E. Press RUN.

The initialization Phase is executed. During this phase the following messages may occur:

Teleprinter Message	Explanation	Action
HS INP ?	Request for Punched Tape Reader channel assignment.	Type channel number. If Punched Tape Reader not available, type 0. †
HS PUN ?	Request for Tape Punch channel assignment.	Type channel number. If Tape Punch not available, type 0.
FWA MEM ?	Request for first word of available memory.	Type address of word in Base Page following the locations required for interrupt processing.
LWA MEM ?	Request for last word of available memory.	Type address of word preceding protected area.
*ERROR	A non-numeric or illegal character has been entered as a reply.	Type the correct value.

Following the completion of the Initialization Phase the relocatable object tapes of the Basic Control System are to be loaded. Only those modules which are to be included in the absolute tape are loaded; modules which are to be loaded with the user's object program are not submitted. The modules may include .IOC., the equipment drivers, and the Relocating Loader; they may be loaded in any order provided that the Relocating Loader is last. During this phase, the following halts may occur:

†All replies from the keyboard must be terminated by an end-of-statement mark which consists of a carriage return, (CR), and a line feed, (LF). If an error is made in typing a reply, type (RUBOUT) (CR) (LF) and repeat the reply.

Teleprinter Message	Explanation	Action
*LOAD	PCS is requesting the first or the next BCS module.	Place BCS tape in Punched Tape Reader if available, or Teleprinter reader. Press RUN.
*L01	Check sum error	To re-read record, reposition tape to beginning of record and press RUN.
*L02	Illegal record: the last record read was not recognized as a valid relocatable record type.	To re-read record, reposition tape to beginning of record and press RUN.
*L03	Memory overflow: the length of BCS exceeds available memory.	Irrecoverable error.
*L04	System linkage area overflow in Base Page.	Irrecoverable error.
*L05	Symbol table for BCS symbols exceeds available memory.	Irrecoverable error.
*L06	PCS interprets the program length of BCS to be zero.	Irrecoverable error.
*L07	Duplicate entry points within BCS.	Irrecoverable error.
*EOT	End-of-Tape	Place next tape in read unit and press RUN to continue loading.

When the .IOC. module is loaded, PCS requests the EQT and SQT parameters. PCS halts after typing the messages "*TABLE ENTRY? EQT?". If the Teleprinter serves both as the reader and keyboard unit, turn reader off, press **RUN**. Begin typing response to message. (Turn reader on after all replies have been typed.)

Teleprinter		
Message	Explanation	Action
*TABLE	Request for EQT entry information.	Press Run and for each I/O device, type: nn, D. ee, [,D] [,Uu] nn – channel number D. ee – driver name: ee = 00 Teleprinter = 01 Punched Tape Reader = 02 Tape Punch = 20 Kennedy 1406 Incremental Tape Transport = 21 2020A Magnetic Tape Unit = 22 3030 Magnetic Tape Unit = 40 Data Source Interface = 41 Integrating Digital Voltmeter = 42 Guarded Crossbar Scanner = 43 Time Base Generator D – device uses DMA channel Uu – physical unit number (0-7) if attached to multi-unit controller To terminate EQT input, type /E.
*ERROR	A non-numeric value has been typed for nn, ee, or u.	Retype the entire correct entry.
SQT? -KYBD?	Request for EQT unit-reference number of unit serving as Keyboard Input.	Type number.
-TTY?	Request for EQT unit-reference number of unit serving as Teleprinter Output.	Type number.

Teleprinter Message	Explanation	Action
-LIB ?	Request for EQT unit-reference number of unit serving as Program Library.	Type number.
-PUNCH ?	Request for EQT unit-reference number of unit serving as Punch Output.	Type number.
-INPUT ?	Request for EQT unit-reference number of unit serving as Input.	Type number.
-LIST ?	Request for EQT unit-reference number of unit serving as List Output.	Type number.
DMA ?	Request for DMA channel numbers.	If one DMA channel, type 6. If two DMA channels, type 6, 7. If no DMA channels, type 0.
*ERROR	A non-numeric parameter or a parameter not equal to 6 or 7 has been entered.	Re-type correct parameter.

After the Relocating Loader is loaded, PCS requests the information needed to set the interrupt linkage for input/output processing. PCS halts after typing the message "INTERRUPT LINKAGE ?". If the Teleprinter is serving both as the reader and the keyboard unit, turn reader off and press Run. Begin typing response to message.

Teleprinter Message	Explanation	Action
INTERRUPT LINKAGE?	Request for interrupt information.	Press Run and for each I/O device, type: a_1, a_2, I. ee a_1 – interrupt location address a_2 – location containing absolute address of Interrupt Processor entry point I. ee entry point name: ee = 00 Teleprinter = 01 Punched Tape Reader = 02 Tape Punch = 20 Kennedy 1406 Incremental Tape Transport = 21 (and C.21) 2020A Magnetic Tape System = 22 (and C.22) 3030A Magnetic Tape System = 43 Time Base Generator If a constant is to be set into the interrupt location, type: a, c a - interrupt location address c - 1 to 6 digit octal constant to be stored at a. Constants should be entered for the following instrument drivers: Data Source Interface (D. 40): 1067 sc (CLC sc, 4) Integrating Digital Voltmeter (D. 40): Ø (NOP) Guarded Crossbar Scanner (D. 42): Ø (NOP) To terminate linkage input, type /E.

Teleprinter Message	Explanation	Action
*ERROR	A non-numeric value has been typed for a_1, a_2, a or c.	Retype the entire correct entry.
*UN NAME	The name I. ee is not defined as an entry point in any I/O driver previously loaded.	1) If the driver name was typed incorrectly, retype the entire correct entry. 2) If related driver is to be loaded with user's program at object program load time, type an exclamation mark (!). The name is added to the Loader's LST. 3) If the driver should have been loaded, rerun PCS.

When the Interrupt Linkage parameters have been supplied, PCS performs the following functions:

1. Prints the message "*UNDEFINED SYMBOL" followed by the entry point names of all system subroutines which have been referenced as externals but not loaded. At this point, PCS may be rerun and the missing subroutines loaded or, the symbols may be added to the Relocating Loader's Loader Symbol Table. Undefined symbols are assigned a value of 77777 for an absolute address.

2. Completes the construction of the Loader Symbol Table.

3. Sets the Memory Table (symbolic location .MEM.) in the Relocating Loader to reflect the final bounds of available memory.

Following this, PCS prints a list of all Basic Control System entry points and the bounds of the System Linkage area in the Base Page.

Teleprinter Message	Explanation	Action
*UNDEFINED SYMBOL <symbol>	An entry point in a BCS module cannot be located.	1) To enter the symbol in the Loader Symbol Table, press RUN. 2) If the subroutine should have been loaded, rerun PCS.
I/O DRIVER? D. ee	A driver has been named in the EQT parameter entry, but has not been loaded.	1) If the driver is to be loaded with user's program at object program load time, type an exclamation mark (!). The name is added to the Loader's LST. 2) If the driver should have been loaded (or if a character other than ! is typed), rerun PCS.
*BCS ABSOLUTE	PCS is ready to punch absolute output tape.	Turn on punch unit and press RUN.

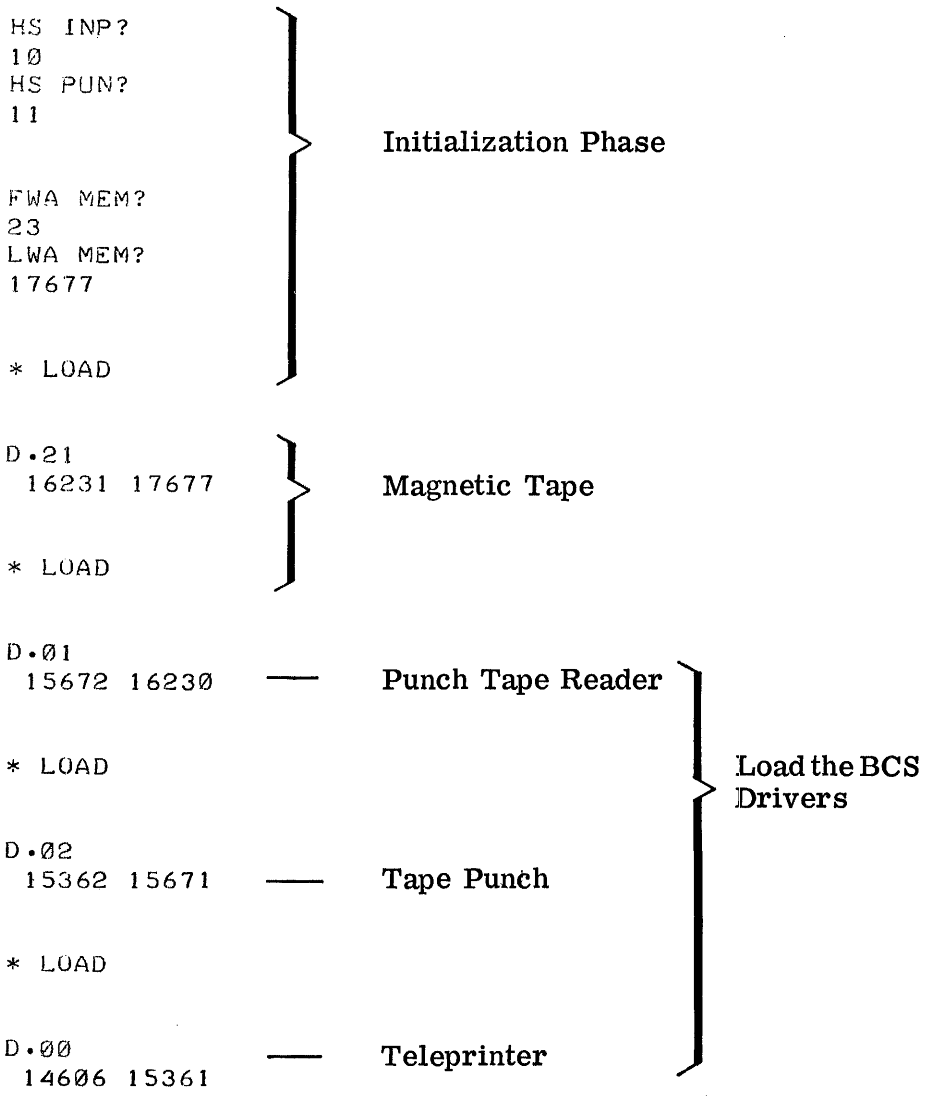
When the binary tape is punched the following halts may occur:

T-Register Contents	Explanation	Action
102077	BCS tape is punched.	To produce additional copies, set Switch 15 to 1 and press RUN.
102066	Tape supply low on 2753A Tape Punch.	Place a new reel of tape in Tape Punch and press RUN to continue.

5.7 EXAMPLE

The teleprinter listing shown below was output from a Prepare Control System operation that configured a BCS tape for an HP 2116 system which included a Teleprinter, a Tape Punch, a Punched Tape Reader, and a 2020A Magnetic Tape Unit.

The PCS tape was loaded; 2000$_8$ was set in the P-Register; the Switch Register was set to the channel number of the teleprinter, in this case the lower channel number (13$_8$). The RUN button was pressed and the Teleprinter printed:



```
HS INP?
1Ø
HS PUN?
11


FWA MEM?
23
LWA MEM?
17677


* LOAD

D.21
 16231  17677


* LOAD

D.Ø1
 15672  1623Ø


* LOAD

D.Ø2
 15362  15671


* LOAD

D.ØØ
 146Ø6  15361
```

Initialization Phase
Magnetic Tape
Punch Tape Reader
Load the BCS Drivers
Tape Punch
Teleprinter

```
* LOAD


IOC
 14367 14605
```

Load IOC

```
* TABLE ENTRY

EQT?
1Ø,D.21
12,D.Ø1
13,D.Ø2
14,D.ØØ
/E
```

Enter the EQT Table

```
SQT?
-KYBD?
12
-TTY?
12
-LIB?
7
-PUNCH?
11
-INPUT?
1Ø
-LIST?
12


DMA?
Ø
```

Enter the SQT Table

```
* LOAD


LOADR
 1217Ø 14337
```

Load the Reloctable Loader

```
 INTERRUPT LINKAGE ?

1Ø,16,I.21
11,17,C.21
12,2Ø,I.Ø1
13,21,I.Ø2
14,22,I.ØØ
15,22,I.ØØ
/E
```

Enter the Interrupt Linkage

```
.SQT.   1434Ø
.EQT.   14346
D.21    16231
I.21    17216
C.21    1713Ø
D.Ø1    15672
I.Ø1    16ØØ7
D.Ø2    15362
I.Ø2    15476
.BUFR   14535
D.ØØ    146Ø6
I.ØØ    14755
.IOC.   14367
DMAC1   146Ø4
DMAC2   146Ø5
IOERR   14563
XSQT    146Ø2
XEQT    146Ø3
.LDR.   13624
HALT    14333
.MEM.   14333
LST     12214
```

BCS entry points

```
*SYSTEM LINK
 ØØØ23 ØØ166
```

Call to turn on the tape punch for absolute binary tape of the configured BCS.

```
*BCS ABSOLUTE OUTPUT
*END
```

DEBUGGING SYSTEM 6

The Debugging routine for the HP 2116A provides the following facilities to aid in program testing:

Print (dump) selected areas of memory in octal or ASCII format
Trace portions of the program during execution
Modify the contents of selected areas in memory
Modify simulated computer registers
Instruction and operand breakpoint halts
Initiate execution at any point in program
Debugging routine restart
Specifying relocatable program base

The Debugging routine supervises the operation of a program in the check-out (debugging) phase through the use of an interpretive mode of execution with simulated A, B, E overflow and P registers.

The Debugging routine is a relocatable program. It is loaded into memory after the user's relocatable programs and before the library subroutines are loaded. The Debugging routine makes use of the input/output control subroutine, IOC.

6.1 OPERATOR COMMUNICATION

All communication between the Debugging routine and the user formed by the Standard Keyboard Input and Standard Teleprinter Output units which are normally assigned to a Teleprinter.

After the program is loaded, the Debugging routine pauses to allow the first type-in. The operator then types one or more control statements to direct the operation of the Debugging routine. Each statement must be terminated by an end-of-statement mark which consists of a carriage return, CR , and a line feed LF . The last statement of the set must be a Run statement.

When an operation requested by a control statement is completed, a pause occurs (except for the Trace operation). The operator may then continue by typing a Run statement, or he may enter new control statements. To regain control at any other time, the operator must use Switch 15. Caution must be used, however, when input/output operations are in progress; setting the switch causes a message to be typed. This action may disrupt any incomplete I/O operation.

6.2 CONTROL STATEMENTS

The basic format of the control statement is a single alphabetic character representing the requested operation followed by a parameter list containing the arguments for the operation separated by commas. The statement is of variable length and is terminated by (CR) (LF). The numeric fields in the parameter list must be in octal; leading zeros may be omitted.

Program Relocation Base

M, a

This statement defines the program relocation base, a, as the absolute origin in memory of the user's relocatable program. This address may be obtained from the listing produced by the Relocating Loader during loading. If not specified, a value of zero is assumed. The value is addedto all address parameters entered by the operator.

Specification of this value allows subsequent reference in the control statements to addresses as shown on the program listing produced by the Assembler or the FORTRAN compiler. If this control statement is not used, program address parameters for other control statements must be absolute.

Example:

```
M, 2ØØØ
```

Set Memory

S, a, v_1, v_2, . . . , v_n

The above statement allows the user to set one or more values into locations defined by the first address, a. The value specified for v_1 is stored in location a; the value for v_2, in location a + 1; and so forth. To specify that an existing value in memory is to remain unchanged, two consecutive commas are used in the control statement. Any number of values may be entered via one control statement provided the length of the statement does not exceed 72 characters.

Example:

```
S,5,062006
S,30,136100,026040
S,40,136101,026050
```

Set Register

W,r,v

This statement sets the value, v, into register, r, where the register is defined as follows:

```
r = A,A-Register
  = B,B-Register
  = E,E-Register
  = O,Overflow
```

Since the Debugging routine simulates the register, the results of a Set Register operation are not reflected on the computer front panel.

Examples:

```
W,B,2
W,A,102000
W,E,1
```

Dump Memory

D, A, a_1, a_2

D, B, a_1, a_2

The second parameter indicates the format of the print-out: A specifies ASCII, B specifies octal (See Output Formats). The address a_1 designates the location of the word or the first of a series of words that is to be dumped. If the second address, a_2, is greater than a_1, a block of memory, a_1 through a_2, is printed. If a_2 is the same as a_1, only one location is printed.

After the data is printed, the Debugging routine waits for the operator to enter another control statement.

Example:

```
D,A,430,477
```

Breakpoint Halt

B,I,a
B,O,a

The first form specifies the address, a, of an instruction breakpoint. Before the instruction at address a is executed, the Debugging routine writes a standard breakpoint message (See Output Formats).

The second form specifies the address, a, of an operand breakpoint. When the Debugging routine detects an effective operand address equal to the value of a, it writes a standard breakpoint message. The operand breakpoint occurs before the memory reference is completed and the register contents in the message are the contents during the instruction execution and not at completion.

After the breakpoint message is transmitted, the Debugging routine waits for the user to enter another control statement.

One or both types of breakpoint halts may be selected. Once selected, a breakpoint address remains in effect until a new address is selected, until a Restart statement is entered, or until the selection is terminated by the statements:

B,I,Ø or B,O,Ø

Trace

T, a_1[, a_2]

When the Trace operation is specified, the execution of the instruction located at address A1, or the execution of every instruction within the area a_1 through a_2, causes the printing of a standard breakpoint message (See Output Formats). The printing occurs before each instruction is executed. Each time the a_1 - a_2 area is reached, the printing resumes; no pause occurs on completion as in the other Debugging routine operations.

The area to be traced must not contain calls to the input/output control routine, IOC. The Trace operation uses IOC to print the breakpoint message. An attempt to trace I/O operations will result in I/O errors.

The trace of the area remains in effect until a new area is selected or until the selection is terminated by the statement:

T,Ø

To enter a new Trace control statement while the program is in operation, Switch 15 must be used.

Run

R [,a]

This statement is used to initiate or continue the execution of the program being debugged. As a result of using Switch 15, or following the printing of a standard breakpoint message (with the exception of those produced by Trace), the Debugging program pauses. The operator may then type a Run statement, or one or more control statements followed by a Run statement to resume program execution. If the letter "R" only is entered, execution starts with the next sequential instruction in the user's program. To start at another location, the operator enters the address, a.

Restart

A

This statement, consisting of the letter "A" is used to abort the current operation and restart. This results in all debugging routine and input/output operations in progress being cleared.

6.3 CONTROL STATEMENT ERROR

If an incorrect control statement is entered, the following message is typed:

"ENTRY ERROR"

This indicates that the character representing the operation is invalid, or that an illegal parameter has been typed. To recover, type in the correct control statement.

6.4 HALT

Any Halt operations coded within the user's program are interpreted by the Debugging routine and result in a typeout consisting of the letter "H" followed by the standard breakpoint message. The operator may then type in one or more control statements or may reinitiate program execution (with the R control statement).

6.5 INDIRECT LOOP

The Debugging routine maintains a count of levels when indirect addressing is detected. When ten consecutive levels of indirect addressing have occurred, an indirect address loop is assumed and the following is typed out:

"INDIRECT LOOP"

L <standard breakpoint message>

6.6 OUTPUT FORMATS

The Debugging routine operations may produce either of two printed outputs: the standard breakpoint message and the memory dump.

Standard Breakpoint Message

Each output line from operations which produce the standard breakpoint message has the following format:

<id>P = v_1 I = v_2 A = v_3 B = v_4 E = v_5 O = v_6 MA = v_7 MC = v_8

The <id> is a letter identifying the operation producing the output:

id = I, Instruction breakpoint
= O, Operand breakpoint
= T, Trace
= S, Switch 15 set up
= L, Indirect Loop
= H, Halt in object program

The v's are octal values of registers and memory locations as follows:

P - P-Register (instruction address)
I - Instruction (contents)
A - A-Register
B - B-Register
E - E-Register
O - Overflow
MA - Effective operand address of a memory reference instruction
MC - Contents of effective address of a memory reference instruction

Dump

The Dump output record format consists of the contents up to 8 consecutive words preceded by the address of the first word:

	addr.	word_1	word_2 ...	word_8
Octal:	aaaaa	oooooo	oooooo ...	oooooo
ASCII:	aaaaa	cc	cc	cc

Octal words consist of 6 octal digits; ASCII words are listed as two ASCII characters. The contents of eight or more consecutive words are not written or they are the same as the last word of the previous record. Instead, a record containing only an asterisk is produced.

6.7 OPERATING PROCEDURES

The following procedures indicate the sequence of steps for use of the Debugging routine.

A. Set the Teleprinter to LINE and check that all equipment to be used is operable.

B. Load Basic Control System using the Basic Binary Loader.

C. Set Switch Register to 000002, press LOAD ADDRESS, and set Switch Register to 000000.

D. Establish Relocating Loader parameters. (If relocation base is to be entered during operation of the Debugging routine, the address must be obtained during loading by setting Switch 15 to 0 (down).)

E. Load reloctable object programs.

F. Load Debugging program (treated as a relocatable program). †

G. Load Program Library routines.

† The Debugging routine need not be loaded as the last relocatable program. If loaded in any other order, however, the absolute address assigned to the symbolic location DEBUG must be entered manually as the starting address for the program.

H. Press RUN.

I. The program pauses to allow the operator to type in the control statements.

J. The program may be restarted at any point by entering the absolute address assigned to the symbolic location DEBRS into the P-Register, and pressing RUN.

6.8 EXAMPLE

The routine employed in this example is a simple loop which totals the contents of a block of data. In order to imbue it with a practical aspect, assume that program "TOTAL" computes personal expenses for a 31-day month. Data (each day's expenses) is read in from the Punched Tape Reader. The sum is printed out on the Teleprinter.

The program is written and assembled as below. To check it out a data tape, consisting of a series of 10's is prepared:

```
10  ( (CR) (LF) )
10  ( (CR) (LF) )
10  ( (CR) (LF) )
 .
 .
 .
```

```
PAGE 0002 #01

0001  00000                    NAM TOTAL
0002  00000 000000    START    NOP
0003  00001 062162R            LDA =D-31
0004  00002 072156R            STA CTR
0005  00003 062163R            LDA =B5
0006  00004 006404             CLB,INB
0007  00005 016004X            JSB .DIO.
0008  00006 000000             ABS 0
0009  00007 000014R            DEF *+5
0010  00010 016006X            JSB .IOR.
0011  00011 016001X            DST INPUT,I
      00012 100055R
0012  00013 016005X            JSB .RAR.
0013  00014 066055R            LDB INPUT               INPUT THE DATA
0014  00015 046164R            ADB =B2
0015  00016 076055R            STB INPUT
0016  00017 036156R            ISZ CTR
0017  00020 026003R            JMP START+3
```

```
0019  00021 062162R       LDA =D-31
0020  00022 072156R       STA CTR            INITIALIZE
0021  00023 016002X       DLD =F0.0
      00024 000165R
0022  00025 016001X       DST ANSW
      00026 000154R

0024  00027 016002X       DLD .MON,I
      00030 100054R
0025  00031 016003X       FAD ANSW
      00032 000154R
0026  00033 016001X       DST ANSW
      00034 000154R
0027  00035 066054R SUM   LDB .MON
0028  00036 046164R       ADB =B2            ADDITION LOOP
0029  00037 076054R       STB .MON
0030  00040 036156R       ISZ CTR
0031  00041 026035R       JMP SUM

0033  00042 062164R       LDA =B2
0034  00043 006400        CLB
0035  00044 016004X       JSB .DIO.
0036  00045 000157R       DEF OUTPT
0037  00046 000053R       DEF *+5
0038  00047 016002X       DLD ANSW           OUTPUT THE RESULT
      00050 000154R
0039  00051 016006X       JSB .IOR.
0040  00052 016007X       JSB .DTA.
0041  00053 102077        HLT 77B

0043  00054 000056R .MON  DEF MONTH
0044  00055 000056R INPUT DEF MONTH
0045  00056 000000  MONTH BSS 62
0046  00154 000000  ANSW  BSS 2
0047  00156 000000  CTR   BSS 1
0048                      EXT .DIO.,.RAR.,.IOR.,.DTA.
0049  00157 024106  OUTPT ASC 3,(F8.2)
      00160 034056
      00161 031051

      00162 177741
      00163 000005
      00164 000002
      00165 000000
      00166 000000
0050                      END START
**  NO ERRORS*
```

The "TOTAL" object tape is loaded by the Basic Control System. The debugging system is loaded next and then the library tape. The program is executed using the Debugging System by the following instructions:

```
M,2ØØØ       Set program relocation base

B,I,53       Breakpoint instruction is 53, the location of the terminating
             halt in the program.

R,1          Initiate execution at statement 1

1Ø.ØØ
H P=ØØØ53   I=1Ø2Ø77   A=177777   B=ØØ6115   E=Ø   O=1
```

The correct answer for the test data would be "31.00", not the 10.00 that was output.

The procedure below illustrates one method for detecting errors in the program.

```
M,2ØØØ       Set program relocation base

             Dump a portion of the storage area MONTH
D,B,56,7Ø
DUMP--BASE = Ø2ØØØ

ØØØ56                                                  Ø5Ø5Ø3 ØØØ333
ØØØ6Ø ØØ1253 ØØØØØØ ØØØØØØ ØØ4267 Ø177ØØ ØØØØØØ Ø53Ø7Ø Ø1177Ø
ØØØ7Ø ØØ2256
```

Read in the data:

```
B,I,21
R,1
I P= ØØØ21   I=Ø62162   A=ØØØØØØ   B=ØØ2154   E=Ø O=Ø   MA= ØØ162 MC=177741
```

Check to see that the data has been stored in memory:

```
D,B,56,7Ø
DUMP--BASE = Ø2ØØØ

ØØØ56                                                          Ø5ØØØØ   ØØØØ1Ø

ØØØ6Ø   Ø5ØØØØ   ØØØØ1Ø   Ø5ØØØØ   ØØØØ1Ø   Ø5ØØØØ   ØØØØ1Ø   Ø5ØØØØ   ØØØØ1Ø

ØØØ7Ø   Ø5ØØØØ
```

Knowing that the data has been stored in MONTH, perform the first addition:

```
B,I,35
R,21
I P= 00035   I=066054   A=050000   B=000010   E=0  O=0   MA= 00054  MC=002056
```

Check to see that the first day's expenses have been stored at ANSW:

```
D,B,154,155
DUMP--BASE = 02000

 00154                                         050000   000010
```

The first addition was executed. Perform the remaining additions by looping:

```
B,I,42
R,35
I P= 00042   I=062164   A=050000   B=002154   E=0  O=0   MA= 00164  MC=000002
```

Check final total in ANSW.

```
D,B,154,155
DUMP--BASE = 02000

 00154                                         050000   000010
```

Here, if not previously, the error should be detected; the program does not perform more than the first addition. The label sum has been placed in the wrong instruction. It should be in location 27 preceding the "DLD . MON, I" instruction.

```
ØØØ1   ØØØØØ                      NAM TOTAL
ØØØ2   ØØØØØ ØØØØØØ  START NOP
ØØØ3   ØØØØ1 Ø62162R              LDA =D-31
ØØØ4   ØØØØ2 Ø72156R              STA CTR
ØØØ5   ØØØØ3 Ø62163R              LDA =B5
ØØØ6   ØØØØ4 ØØ64Ø4               CLB,INB
ØØØ7   ØØØØ5 Ø16ØØ4X              JSB .DIO.
ØØØ8   ØØØØ6 ØØØØØØ               ABS Ø
ØØØ9   ØØØØ7 ØØØØ14R              DEF *+5
ØØ1Ø   ØØØ1Ø Ø16ØØ6X              JSB .IOR.
ØØ11   ØØØ11 Ø16ØØ1X              DST INPUT,I
       ØØØ12 1ØØØ55R
ØØ12   ØØØ13 Ø16ØØ5X              JSB .RAR.
ØØ13   ØØØ14 Ø66Ø55R              LDB INPUT              INPUT THE DATA
ØØ14   ØØØ15 Ø46164R              ADB =B2
ØØ15   ØØØ16 Ø76Ø55R              STB INPUT
ØØ16   ØØØ17 Ø36156R              ISZ CTR
ØØ17   ØØØ2Ø Ø26ØØ3R              JMP START+3

ØØ19   ØØØ21 Ø62162R              LDA =D-31
ØØ2Ø   ØØØ22 Ø72156R              STA CTR                INITIALIZE
ØØ21   ØØØ23 Ø16ØØ2X              DLD =FØ.Ø
       ØØØ24 ØØØ165R
ØØ22   ØØØ25 Ø16ØØ1X              DST ANSW
       ØØØ26 ØØØ154R

ØØ24   ØØØ27 Ø16ØØ2X              DLD .MON,I
       ØØØ3Ø 1ØØØ54R
ØØ25   ØØØ31 Ø16ØØ3X              FAD ANSW
       ØØØ32 ØØØ154R
ØØ26   ØØØ33 Ø16ØØ1X              DST ANSW
       ØØØ34 ØØØ154R
ØØ27   ØØØ35 Ø66Ø54R  SUM         LDB .MON
ØØ28   ØØØ36 Ø46164R              ADB =B2                ADDITION LOOP
ØØ29   ØØØ37 Ø76Ø54R              STB .MON
ØØ3Ø   ØØØ4Ø Ø36156R              ISZ CTR
ØØ31   ØØØ41 Ø26Ø35R              JMP SUM

ØØ33   ØØØ42 Ø62164R              LDA =B2
ØØ34   ØØØ43 ØØ64ØØ               CLB
ØØ35   ØØØ44 Ø16ØØ4X              JSB .DIO.
ØØ36   ØØØ45 ØØØ157R              DEF OUTPT
ØØ37   ØØØ46 ØØØØ53R              DEF *+5
ØØ38   ØØØ47 Ø16ØØ2X              DLD ANSW               OUTPUT THE RESULT
       ØØØ5Ø ØØØ154R
ØØ39   ØØØ51 Ø16ØØ6X              JSB .IOR.
ØØ4Ø   ØØØ52 Ø16ØØ7X              JSB .DTA.
ØØ41   ØØØ53 1Ø2Ø77               HLT 77B
```

HP CHARACTER SET

ASC II CHARACTER FORMAT

b_7				0	0	0	0	1	1	1	1
b_6				0	0	1	1	0	0	1	1
b_5				0	1	0	1	0	1	0	1
b_4	b_3	b_2	b_1								
0	0	0	0	NULL	DC_0	ƀ	0	@	P	UNASSIGNED	UNASSIGNED
0	0	0	1	SOM	DC_1	!	1	A	Q		
0	0	1	0	EOA	DC_2	"	2	B	R		
0	0	1	1	EOM	DC_3	#	3	C	S		
0	1	0	0	EOT	DC_4 (STOP)	$	4	D	T		
0	1	0	1	WRU	ERR	%	5	E	U		
0	1	1	0	RU	SYNC	&	6	F	V		
0	1	1	1	BELL	LEM	' (APOS)	7	G	W		
1	0	0	0	FE_0	S_0	(	8	H	X		
1	0	0	1	HT / SK	S_1	)	9	I	Y		
1	0	1	0	LF	S_2	*	:	J	Z		
1	0	1	1	V_{TAB}	S_3	+	;	K	[		
1	1	0	0	FF	S_4	, (COMMA)	<	L	\		ACK
1	1	0	1	CR	S_5	−	=	M	]		①
1	1	1	0	SO	S_6	.	>	N	↑		ESC
1	1	1	1	SI	S_7	/	?	O	←		DEL

Standard 7-bit set code positional order and notation are shown below with b_7 the high-order and b_1 the low-order, bit position.

	b_7	b_6	b_5	b_4	b_3	b_2	b_1
EXAMPLE: The code for "R" is:	1	0	1	0	0	1	0

LEGEND

NULL	Null/Idle
SOM	Start of message
EOA	End of address
EOM	End of message
EOT	End of transmission
WRU	"Who are you?"
RU	"Are you . . . ?"
BELL	Audible signal
FE_0	Format effector
HT	Horizontal tabulation
SK	Skip (punched card)
LF	Line feed
V_{TAB}	Vertical tabulation
FF	Form feed
CR	Carriage return
SO	Shift out
SI	Shift in
DC_0	Device control reserved for data link escape
DC_1-DC_3	Device Control
DC_4(Stop)	Device control (stop)
ERR	Error
SYNC	Synchronous idle
LEM	Logical end of media
S_0-S_7	Separator (information)
ƀ	Word separator (space, normally non-printing)
<	Less than
>	Greater than
↑	Up arrow (Exponentiation)
←	Left arrow (Implies/Replaced by)
\	Reverse slant
ACK	Acknowledge
①	Unassigned control
ESC	Escape
DEL	Delete/Idle

BINARY CODED DECIMAL FORMAT

Kennedy 1406/1506 ASCII-BCD Conversion

Symbol	BCD (octal code)	ASCII Equivalent (octal code)	Symbol	BCD (octal code)	ASCII Equivalent (octal code)
(Space)	2Ø	Ø4Ø	A	61	1Ø1
!	52	Ø41	B	62	1Ø2
#	13	Ø43	C	63	1Ø3
$	53	Ø44	D	64	1Ø4
%	34	Ø45	E	65	1Ø5
&	6Ø	Ø46	F	66	1Ø6
'	14	Ø47	G	67	1Ø7
(	34	Ø50	H	7Ø	11Ø
)	74	Ø51	I	71	111
*	54	Ø52	J	41	112
+	6Ø	Ø53	K	42	113
,	33	Ø54	L	43	114
-	4Ø	Ø55	M	44	115
.	73	Ø56	N	45	116
/	21	Ø57	O	46	117
			P	47	12Ø
Ø	12	Ø6Ø	Q	50	121
1	Ø1	Ø61	R	51	122
2	Ø2	Ø62	S	22	123
3	Ø3	Ø63	T	23	124
4	Ø4	Ø64	U	24	125
5	Ø5	Ø65	V	25	126
6	Ø6	Ø66	W	26	127
7	Ø7	Ø67	X	27	13Ø
8	1Ø	Ø7Ø	Y	30	131
9	11	Ø71	Z	31	132
:	15	Ø72	[	75	133
;	56	Ø73	\	36	134
<	76	Ø74	]	55	135
=	13	Ø75			
>	16	Ø76			
?	72	Ø77			
@	14	1ØØ			

Other symbols which may be represented in ASCII are converted to spaces in BCD (20)

HP 2020A/B ASCII - BCD Conversion

Symbol	ASCII (Octal code)	BCD (Octal code)
(Space)	4Ø	2Ø
!	41	52
"	42	37
#	43	13
$	44	53
%	45	34
&	46	60 †
'	47	36
(	5Ø	75
)	51	55
*	52	54
+	53	6Ø
,	54	33
-	55	4Ø
.	56	73
/	57	21
Ø	6Ø	12
1	61	Ø1
2	62	Ø2
3	63	Ø3
4	64	Ø4
5	65	Ø5
6	66	Ø6
7	67	Ø7
8	7Ø	1Ø
9	71	11
:	72	15
;	73	56
<	74	76
=	75	35
>	76	16
?	77	72
@	1ØØ	14
A	1Ø1	61
B	1Ø2	62
C	1Ø3	63
D	1Ø4	64
E	1Ø5	65
F	1Ø6	66
G	1Ø7	67
H	11Ø	70
I	111	71
J	112	41
K	113	42
L	114	43
M	115	44
N	116	45
O	117	46
P	12Ø	47
Q	121	50
R	122	51
S	123	22
T	124	23
U	125	24
V	126	25
W	127	26
X	13Ø	27
Y	131	30
Z	132	31
[	133	75 ‡
]	135	55 ‡
↑	136	77
←	137	32

† BCD code of 60 always converted to ASCII code 53 (+).

‡ BCD code of 75 always converted to ASCII code 50 (() and BCD code of 55 always converted to ASCII code 51 ()).

EQUIPMENT TABLE

The Equipment Table (EQT) provides information for the input/output control routine, .IOC., and the equipment driver subroutines. The table contains an entry for each peripheral device attached to an HP 2116A Computer configuration.

The table is constructed as a block of entries assembled by the Prepare Control System routine. The first word of the table, at the symbolic entry point .EQT., contains the number of entries in the table. An entry in the table is referenced according to its position. The numbers 1 through 6 are reserved for Standard units (see Standard Equipment Table). The number 7_8 appearing in a program refers to the 1st table entry; the number 10_8, the second, and so forth. The numbers may be in the range 7_8-74_8 with the largest value being determined by the number of units of equipment available at the installation.

The 4-word entry for each device contains the following information:

The channel number of the device (10_8-76_8)

A Direct Memory Access channel indicator if pertinent

Absolute address of equipment driver subroutine

Equipment type identification code.

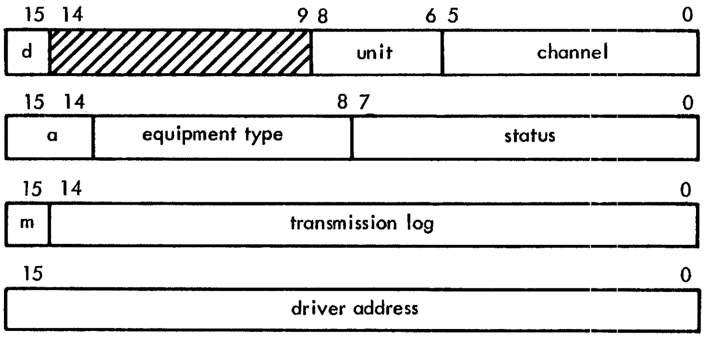
The above information is static for each installation; it is not altered by .IOC. The entry also contains dynamic information which is supplied by the equipment driver subroutine. This information includes:

Status of operation (i.e., in progress or complete)

Status of equipment

Number of characters or words transmitted when the operation is completed.

The format of the entry is as follows:



d

Direct Memory Access channel indicator

1 DMA channel is to be used for each data transmission operation

0 DMA channel not required

unit

Physical unit number (0-7) used to address the device if it is attached to a multi-unit controller.

channel

The channel number (select code) for the physical device (also the low core location containing a JSB to the related interrupt subroutine.)

a

Availability of device:

0 The device is available; the previous operation is complete.

1 The device is available; the previous operation is complete but a transmission error has been detected.

2 The device is not available for another request; the operation is in progress.

equipment type

This field contains a 6-bit code that identifies the device:

00-07 – Paper Tape devices
00 2752A Teleprinter
01 2737A Punched Tape Reader
02 2753A Tape Punch

10-17 – Unit Record devices
20-37 – Magnetic Tape and Mass Storage devices
- 20 Kennedy 1406 Incremental Tape Transport
- 21 2020A Magnetic Tape Unit
- 22 3030 Magnetic Tape Unit

40-77 – Instrumentation devices
- 40 Data Source Interface
- 41 Integrating Digital Voltmeter
- 42 Guarded Crossbar Scanner
- 43 Time Base Generator

status — The status field indicates the actual status of the device when the data transmission is complete. The contents depend on the type of device (see Status Table).

m — This bit defines the mode of the data transmission:

0 ASCII or BCD

1· Binary

transmission log — This field is a log of the number of characters or words transmitted. The value is given as a positive integer and indicates characters or words as specified in the calling sequence. The value is stored in this field only when the input/output request has been completed, therefore, when all data is transmitted or when a transmission error is detected.

driver address — Absolute address of the entry point for the associated driver subroutine for the device.

STATUS TABLE

Status Bit / Device	7	6	5	4	3	2	1	0
2752A Teleprinter			End of Input Tape					
2737A Punched Tape Reader			End of Tape					
2753A Tape Punch			Tape Supply Low					
Kennedy 1406 Incremental Tape Transport			End of Tape		BT			DB
2020A Magnetic Tape Unit	EOF	ST	End of Tape	TE	I/O R	NW	PA	DB
3030 Magnetic Tape Unit	EOF	ST	End of Tape	TE	I/O R	NW	PA	DB

BT	=	Broken Tape
DB	=	Device Busy
EOF	=	End of File
ST	=	Start of Tape
TE	=	Timing Error
I/OR	=	I/O Reject
NW	=	No Write (write enable ring missing or tape unit is rewinding)
PA	=	Parity Error

STANDARD UNIT EQUIPMENT TABLE

The Standard Unit Equipment Table (SQT) allows reference to input/output devices designated as Standard units. The Table contains six 1-word entries. Each entry corresponds to a particular Standard unit and contains a pointer to the Equipment Tàble. The Standard units are as follows:

Number	Name
1	Keyboard Input
2	Teleprinter Output
3	Program Library
4	Punch Output
5	Input
6	List Output

The number defines the position in the SQT at which the device is listed. Each Standard unit may be a different device, or a single physical device may represent several Standard units. The value of the pointer in the SQT is the position of the physical unit's entry in the EQT, with the lowest value being 7_8 (see EQT).

IOC WITH OUTPUT BUFFERING

IOC with Output Buffering is an extension of the standard version and provides for automatic stacking and buffering of all output and function requests. This involves moving an output request and associated buffer into available memory and adding the request location into a thread of stacked requests for the referenced unit. At the completion of an output operation, the next entry in the stack for the unit is initiated by IOC. The processing of output/function requests for a particular unit is according to the order of the requests (first in/first out). This version of IOC requires the use of the program MEMRY to perform the allocation and release of blocks of available memory. If available memory is exhausted when an allocation is attempted, IOC repeats the call until space is made available, i. e. , previous blocks are released.

PRIORITY OUTPUT

A "Priority" write or function request has been added for use with the Buffered version of IOC. A "Priority" request is processed immediately without the request and buffer being moved to available memory. The current operation in the stack for the referenced unit is suspended, the "Priority" request processed, and the suspended operation re-initiated. The "Priority" feature is useful for writing messages or diagnostics for immediate action or for performing output without reserving a segment of available memory for request/buffer storage. (All output performed by the BCS Relocating Loader is done as Priority requests because of the latter reason.) (NOTE: If two (2) or more Priority requests are called in immediate succession (without intervening status checks), the last requested operation is performed with the previous ones being "lost".)

A "Priority" request (i.e., Write function) is indicated by setting bit Ø9 of Word 2 of the request call = 1. Bit Ø9 = 0 means normal operation with the Standard IOC and means the request will be stacked and buffered with the extended version.

Example: "Priority" Write to Teleprinter

```
JSB   .IOC.
OCT   21ØØ2
JMP   REJ
DEF   BUFFR
DEC   -37
```

OPERATING ENVIRONMENT

IOC with Output Buffering provides for writing a data block on more than one output device in parallel and does not restrict output rates to the lowest speed device. Because all requests and buffers are moved into available memory for subsequent processing, peak load output processing is not delayed due to device speed or saturated buffer storage within the bounds of user programs. System I/O saturation occurs when available memory is exhausted.

RESTRICTIONS

The routines used to allocate/release blocks in available memory and to initiate stacked output requests operate with the Interrupt System disabled. Therefore, the use of medium/high speed synchronous I/O devices under program control is not recommended because of possible data loss (e.g., HP 2020 A/B Magnetic Tape).

An I/O driver routine operating under the extended version of IOC may not be used to control more than one like device. This is because the buffering control routine in IOC only checks for stacked requests referencing the unit on which an operation was just completed.

HALT CONDITIONS

Irrecoverable error conditions are identical to the Standard version of IOC. The location of the halt is at the entry point "IOERR". These conditions are:

A-Register	B-Register	Meaning
∅	Location at Request	Request Code Illegal
1	Location at Request	Unit Reference Illegal
∅	∅	Write request for an input only device.

I/O ERROR CONDITIONS

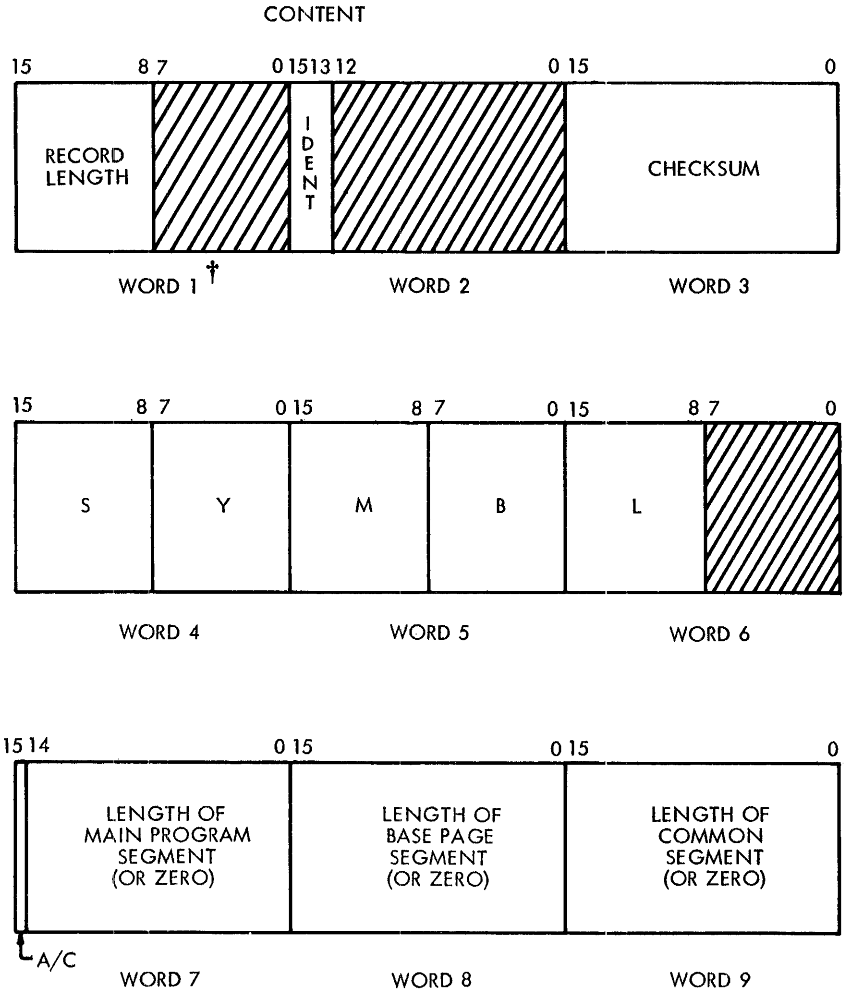
The routine .BUFR in the version of IOC with Output Buffering checks for error conditions of the end of each output operation. If any error conditions and End-of-Tape or Tape Supply Low, etc. conditions are present, IOC halts to allow the condition to be corrected. Processing is continued by pressing RUN,

Halt: (T) = 1∅2∅7∅
(A) = Word 2 of EQT entry (Status word)
(B) = Hardware I/O address of unit

RELOCATABLE TAPE FORMAT

E

NAM RECORD



EXPLANATION

RECORD LENGTH = 9 WORDS

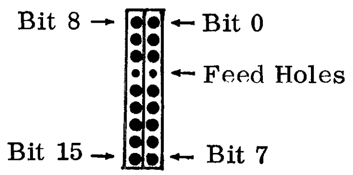
IDENT = 001

CHECKSUM: ARITHMETIC TOTAL OF ALL WORDS IN RECORD EXCLUDING WORDS 1 AND 3.

SYMBL: FIVE CHARACTER NAME OF PROGRAM

A/C: BINARY TAPE PROCESSOR
= 0 IF ASSEMBLER PRODUCED
= 1 IF COMPILER PRODUCED

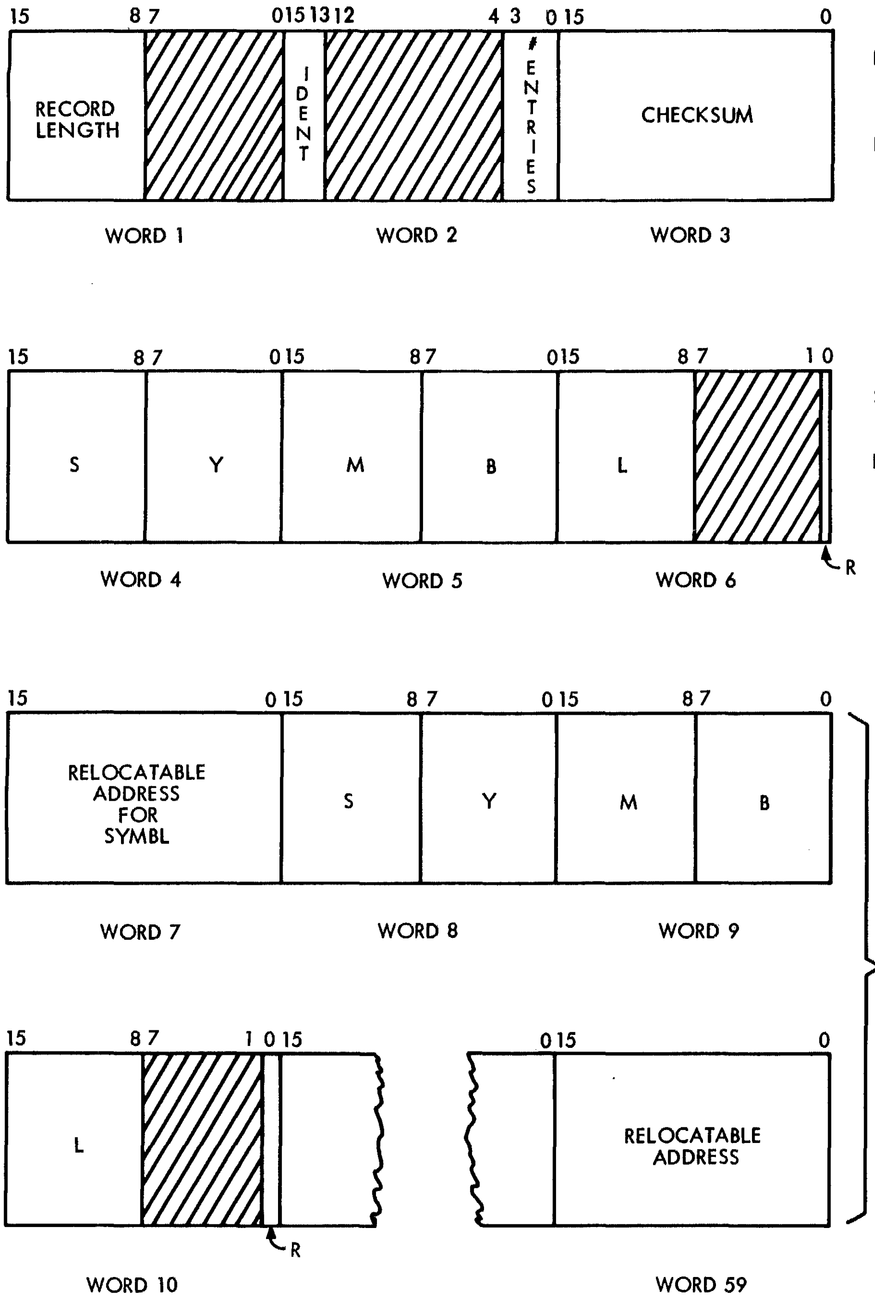
†Each word represents two frames arranged as follows:



ENT RECORD

CONTENT

EXPLANATION



RECORD LENGTH = 7-59 WORDS

IDENT = 010

ENTRIES: 1 to 14 ENTRIES PER PROGRAM; EACH ENTRY IS FOUR WORDS LONG.

SYMBL: 5 CHARACTER ENTRY POINT SYMBOL

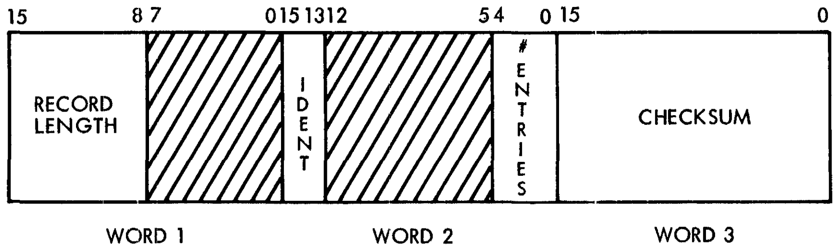
R: RELOCATION INDICATOR
= 0 IF PROGRAM RELOCATABLE
= 1 IF BASE PAGE RELOCATABLE

WORDS 4 THROUGH 7 ARE REPEATED FOR EACH ENTRY POINT SYMBOL.

EXT RECORD

CONTENT

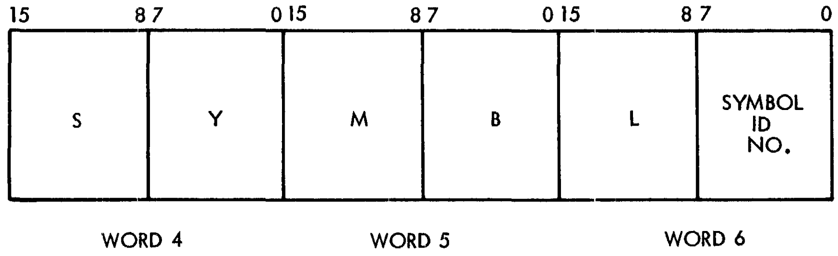
EXPLANATION



RECORD LENGTH = 6-60 WORDS

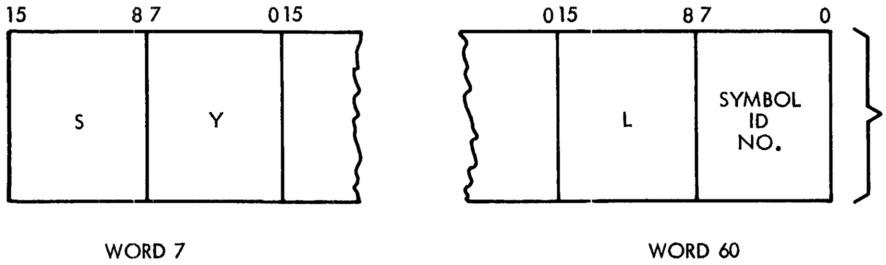
IDENT = 100

ENTRIES: 1 TO 19 PER RECORD; EACH ENTRY IS THREE WORDS LONG



SYMBL: 5 CHARACTER EXTERNAL SYMBOL

SYMBOL ID. NO.: NUMBER ASSIGNED TO SYMBL FOR USE IN LOCATING REFERENCE IN BODY OF PROGRAM.

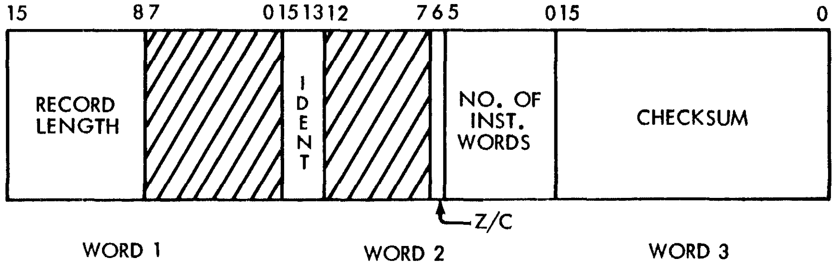


WORDS 4 THROUGH 6 REPEATED FOR EACH EXTERNAL SYMBOL (MAXIMUM OF 19 PER RECORD).

DBL RECORD

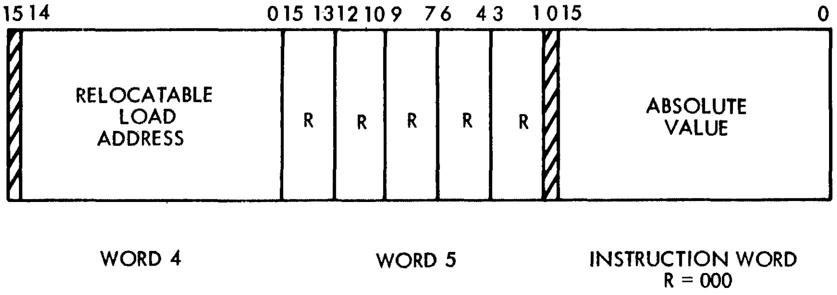
CONTENT

EXPLANATION



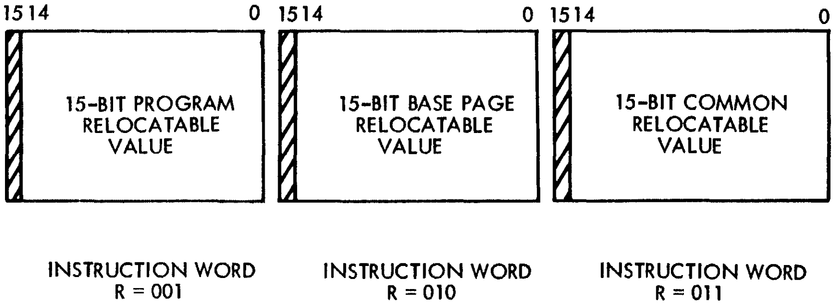
RECORD LENGTH = 5-60 WORDS
IDENT = 011
Z/C: BASE/CURRENT PAGE LOADING
= 0 FOR BASE PAGE
= 1 FOR CURRENT PAGE

NO. OF INST. WORDS: 1 TO 45 LOADABLE INSTRUCTION WORDS PER RECORD



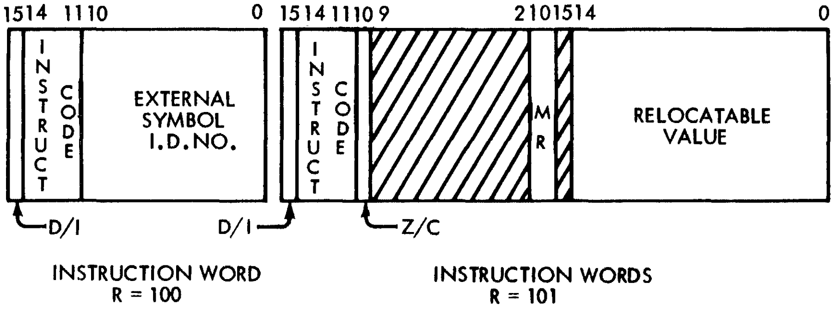
RELOCATABLE LOAD ADDRESS: STARTING ADDRESS FOR LOADING THE INSTRUCTIONS WHICH FOLLOW.

R's: RELOCATION INDICATORS:
000 = ABSOLUTE
001 = 15-BIT PROGRAM RELOCATABLE
010 = 15-BIT BASE PAGE RELOCATABLE
011 = 15-BIT COMMON RELOCATABLE
100 = EXTERNAL REFERENCE
101 = MEMORY REFERENCE


15 14
0
15 14
0
15 14
0
15-BIT PROGRAM RELOCATABLE VALUE
15-BIT BASE PAGE RELOCATABLE VALUE
15-BIT COMMON RELOCATABLE VALUE
INSTRUCTION WORD
R = 001
INSTRUCTION WORD
R = 010
INSTRUCTION WORD
R = 011

R_1 IS RELOCATION INDICATOR FOR INSTRUCTION $WORD_1$; R_2, FOR INSTRUCTION $WORD_2$; ETC-MEMORY REFERENCE INSTRUCTIONS USE TWO WORDS, WITHIN THE TWO-WORD GROUP, "MR" INDICATES RELOCATABILITY OF OPERAND SPECIFIED IN SECOND WORD:

00 = PROGRAM RELOCATABLE
01 = BASE PAGE RELOCATABLE
10 = COMMON RELOCATABLE



D/I: INDIRECT ADDRESSING

0 = DIRECT
1 = INDIRECT

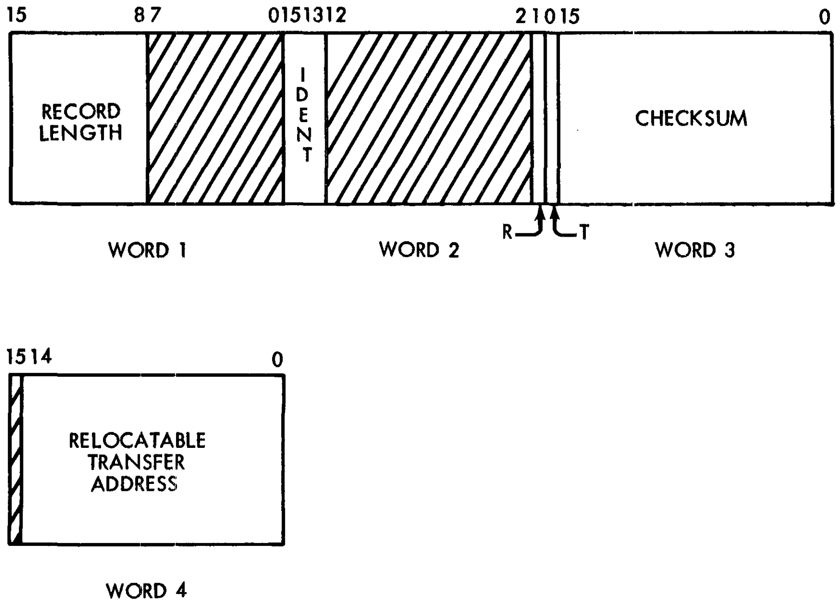
Z/C: BASE/CURRENT PAGE LOCATION OF OPERAND ADDRESS AS DETERMINED BY LOADER.

0 = BASE PAGE
1 - CURRENT PAGE

END RECORD

CONTENT

EXPLANATION



RECORD LENGTH = 4 WORDS

IDENT = 101

R: RELOCATION INDICATOR FOR TRANSFER ADDRESS

= 0 IF PROGRAM RELOCATABLE
= 1 IF BASE PAGE RELOCATABLE

T: TRANSFER ADDRESS INDICATOR

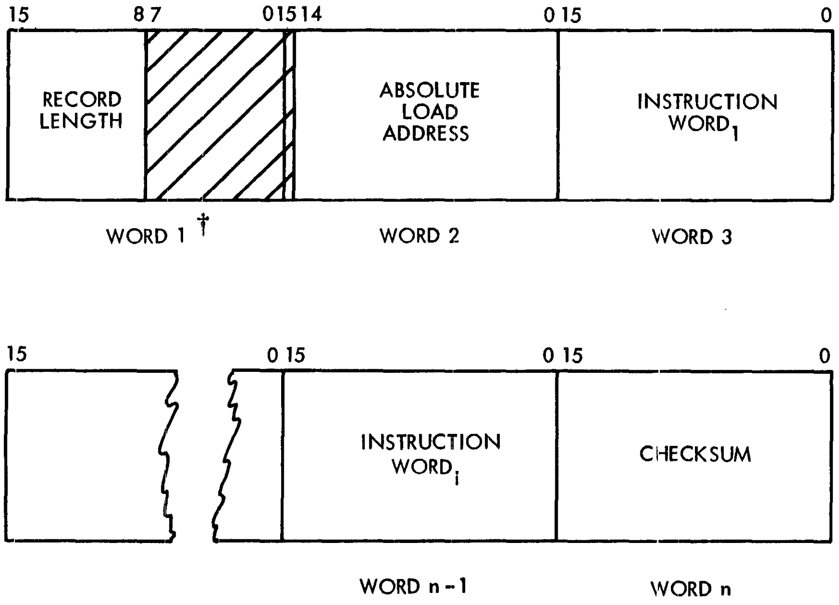
= 0 IF NO TRANSFER ADDRESS IN RECORD

= 1 IF TRANSFER ADDRESS PRESENT

ABSOLUTE TAPE FORMAT

F

CONTENT



EXPLANATION

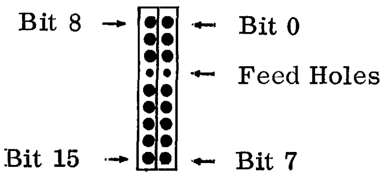
RECORD LENGTH = NUMBER OF WORDS IN RECORD EXCLUDING WORDS 1 AND 2 AND THE LAST WORD.

ABSOLUTE LOAD ADDRESS: STARTING ADDRESS FOR LOADING THE INSTRUCTIONS WHICH FOLLOW

INSTRUCTION WORDS: ABSOLUTE INSTRUCTIONS OR DATA

CHECKSUM: ARITHMETIC TOTAL OF ALL WORDS EXCEPT FIRST AND LAST

† Each word represents two frames arranged as follows:



INDEX

HEWLETT hp PACKARD

READER COMMENT SHEET

BASIC CONTROL SYSTEM

HP 02116-9017 February 1968

Hewlett-Packard welcomes your evaluation of this text. Any errors, suggested additions, deletions, or general comments may be made below. Use extra pages if you like.

CUT ALONG LINE

FROM

NAME: ____________________

ADDRESS: ____________________

____

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

FOLD ON DOTTED LINES AS SHOWN ON OTHER SIDE AND TAPE

FOLD FOLD

MANAGER, SOFTWARE PUBLICATIONS
HEWLETT - PACKARD
CUPERTINO DIVISION
11000 Wolfe Road
Cupertino, California
95014

FOLD FOLD

02116-9017