HONEYWELL EDP

SOFTWARE BULLETIN

MOD1 (TR)

FLOATING TAPE LOADER-MONITOR C AND INTERRUPT CONTROL D ADDENDUM NO. 2

GENERAL SYSTEM: SERIES 200/OPERATING SYSTEM - MOD 1 (TAPE RESIDENT)

SUBJECT: Changes and Additions to the Floating Tape Loader-Monitor C and Interrupt Control D Manual Resulting Primarily from Modifications to Enable Interrupt Control D to Process Multisegmented Foreground Programs.

SPECIAL INSTRUCTIONS: The attached pages should be used to replace the corresponding pages in the manual. All pages will be incorporated into the next edition of the manual.

NOTE: The user can signal the updating of the manual with Addendum No. 2 by inserting this cover immediately behind the manual cover.

DATE: April 27, 1967

FILE NO.: 122.5005.001C.1-005*

UPDATES VERSION DATED 660420 (PAGE 8-8 REPLACED 660608)

SUPERSEDES VERSION DATED 651015

9630
4467
Printed in U.S.A.

*Underscoring denotes Order Number.

FOREWORD

This manual describes programming and operating characteristics of Floating Tape Loader-Monitor C and Interrupt Control D, programs expanding the capabilities of the SERIES 200/OPERATING SYSTEM - MOD 1.

The basic functions of Floating Tape Loader-Monitor C are similar to those of previously released Honeywell Loader-Monitor programs in that their objective is to govern the search for and loading of programs stored in binary run format. Floating Tape Loader-Monitor C has been devised to incorporate the features of Tape Loader-Monitor C(3), Tape Loader-Monitor C(4), and Card Loader-Monitor B. Incorporation of these features has been accomplished through the unique macro structure of Floating Tape Loader-Monitor C, which comprises generalized control routines that can be specialized to provide a more inclusive capability when performing the basic functions. Thus, through selective specialization, programs assembled in three- or four-character address mode can be loaded from magnetic tape only (BRT) or from magnetic tape and punched cards (BRD).

Differing from previous Loader-Monitors, Floating Tape Loader-Monitor C normally operates in the highest available memory bank, thereby enabling subsequent programs to be loaded into bank 0 (if so specified) starting anywhere above decimal location 189. Also, Floating Tape Loader-Monitor C provides the optional capability of using a console typewriter for program and error messages, a feature not presently available with the other Loader-Monitors.

Interrupt Control D has been designed to couple with Floating Tape Loader-Monitor C in order to regulate simultaneous processing of two object programs. Floating Tape Loader-Monitor C loads the programs and Interrupt Control D governs their processing sequence by intervening, on interrupt signal, to reassign program control for the most efficient use of memory cycles.

Both Floating Tape Loader-Monitor C and Interrupt Control D are macro programs which require specialization by Library Processor C or D before being assembled by Easycoder Assembler C or D. Sections V and VIII of this manual contain operating information for Honeywell prespecialized versions of Floating Tape Loader-Monitor C and Interrupt Control D, respectively.

The information presented in this manual has been arranged in sections for direct reference according to the reader's familiarity with Honeywell Loader-Monitor programs. An explanation of Floating Tape Loader-Monitor C is contained in Sections I through VII: (I) basic program characteristics and machine requirements, (II) operational and functional description of the program, (III) parameter information describing options and capabilities, (IV) methods of entering parameter values, (V) provisions for specializing the Loader-Monitor, (VI) summary of programming considerations, (VII) procedures for loading and operating Floating Tape Loader-Monitor C. An explanation of Interrupt Control D is contained in Section VIII; the necessary programming, specializing, and operating procedures are included. Appendices A through D offer additional information concerning visibility, own-coding, address assignments, and character code designations, respectively.

The experienced programmer or operator need only refer to Sections III, V, and VII for the operation of Floating Tape Loader-Monitor C or to Section VIII to use Interrupt Control D.

Reference material for the specific programs mentioned above can be found in the appropriately titled Honeywell Software Manuals: Tape Loader-Monitor C, Order No. 221; Card Loader-Monitor B, Order No. 154; Library Processors C and D, Order No. 051; Easycoder Assemblers C and D, Order No. 041; and Data Conversion A and C, Order No. 231. The reader should be familiar with the information presented in the Equipment Operators' Manual (Model 200), Order No. 040, and the Programmers' Reference Manual (Models 200/1200/2200/4200) Order No. 139.

Electronic Data Processing Division
Wellesley Hills, Massachusetts 02181

TABLE OF CONTENTS

TABLE OF CONTENTS (cont)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

BASIC CONCEPTS OF FLOATING TAPE LOADER-MONITOR C

INTRODUCTION

The Mod 1 Operating System has been given additional control flexibility with the incorporation of Floating Tape Loader-Monitor C. Floating Tape Loader-Monitor C has been designed as a macro program which, through selective specialization, can provide a more extensive capability when performing its basic functions of searching for, loading, and starting programs stored in binary run format. The object programs can be generated in binary run format and stored on magnetic tape (BRT) or punched cards (BRD) by Easycoder Assemblers C or D, COBOL Compiler D, or Fortran Compiler D.

The searching, loading, and starting information required by Floating Tape Loader-Monitor C is reflected in its communication area, which may be modified by entering the desired values before the particular program is called. When the Loader-Monitor resides in memory, a call to search for and start loading a specified program may be issued by the operator or by programmed instruction from a previously loaded program. In addition to the use of program and segment names, a program or segment may be identified by its particular visibility coding as described in Appendix A.

Through specialization, Floating Tape Loader-Monitor C can incorporate the features of Tape Loader-Monitor C(3), Tape Loader-Monitor C(4), and Card Loader-Monitor B (e. g., programs assembled in three- or four-character address mode can be loaded from either magnetic tape or punched cards). The Loader-Monitor is normally resident in the highest memory bank; therefore, object programs (if so specified) can be loaded anywhere above decimal location 189. However, Floating Tape Loader-Monitor C can be specialized to be relocatable at execution time for loading into any available memory bank above bank 0, or it could be specialized to always load into a specific memory bank above bank 0. Other options are available to allow direct communication between a program in four-character address mode and the Loader-Monitor and to provide for console typewriter or control panel diagnostics.

Floating Tape Loader-Monitor C has the capability to communicate with the Interrupt Control D program, enabling both programs to reside in memory and interact to allow simultaneous processing of a main (background) program (e. g., assembly, sort, collate, etc.) and a peripheral (foreground) program (e. g., tape-to-printer, card-to-tape, etc.).

As a macro program, Floating Tape Loader-Monitor C requires specialization by Library Processor C or D. Section V includes two prespecialized versions of Floating Tape Loader-Monitor C which are available for the most probable applications.

EQUIPMENT REQUIREMENTS

The minimum machine requirements for Floating Tape Loader-Monitor C are given below.

1. Memory locations: The number of storage locations needed is dependent upon the macro instruction parameters entered during the specialization process. For example, a minimum number of 1,400 locations is required for the prespecialized version with program loading from tape only, three-character addressing, and no typewriter options; 1,950 locations are required for the prespecialized version with program loading from tape only, three-character addressing, and typewriter options; and a maximum number of 2,750 locations is required for a user-specialized version with program loading from both tape and cards, four-character addressing, and typewriter options.

   For any version of the Loader-Monitor, 126 additional locations (64 through 189) are required for its communication area.

2. Programming: The Loader-Monitor uses the Advanced Programming Instructions, as well as index registers X5 and X6.

3. Tape drives: One Type 204B.

Additional Equipment Usable

A card reader can be used with Floating Tape Loader-Monitor C to enable both the calling of programs via a Console Call card and the loading of programs punched on cards in binary run format. A Type 220-1, -2, or -3 console may be used if the typewriter option is specified. If the Type 220-3 console is used, the programmer must not allow a program to branch to the Loader-Monitor unless the "interrupt-allow" function for data termination interrupts is turned off; e.g., it may be turned off by a PCB instruction with octal 70 for the third control character.

SECTION II

OPERATIONAL DESCRIPTION OF FLOATING TAPE LOADER-MONITOR C

INTRODUCTION

The Floating Tape Loader-Monitor C program is loaded from a BRT and is designed to load other object programs in binary run format. Once activated, the Loader-Monitor can receive a request or "call" for a particular program unit and automatically process the call information to perform its three functions of searching for, loading, and starting the unit called.

Throughout this manual the following terms should be interpreted as defined here. A "program unit" is the unit of data which is found and loaded as the result of a single call to the Loader-Monitor. An assembled program unit consisting of strings of program data interspersed with control (loading) characters may constitute one object program, or several such program units may represent one object program. The term "record" will be used to denote a record on tape or a card image. A "halt" refers to the occurrence of a program condition which requires an operator decision. The condition is relayed to the operator through either the control panel (via a machine halt) or, if used, the console typewriter (via a typed message). All halts associated with Floating Tape Loader-Monitor C are listed in Table 7-1 (on page 7-5) and numbered to correspond to the appropriate operator action.

In this section, the methods of activating the Loader-Monitor are discussed as well as the basic procedures involved for calling a program unit and for utilizing the three functions of the Loader-Monitor.

ACTIVATION OF FLOATING TAPE LOADER-MONITOR C

Floating Tape Loader-Monitor C can be placed on a BRT as output of the Easycoder Assembler C or D programs or the Update and Select C or D programs. The BRT consists of a header label, a tape bootstrap routine, at least one Loader-Monitor program, one or more program units, a trailer label, and two ERI records.

Several Loader-Monitors may reside on the BRT, and since each normally has the program name AAAMON, they should be differentiated by unique visibilities. The tape bootstrap routine is designed to search for and load the first Loader-Monitor with the program name AAAMON and a unique predesignated visibility. Therefore, the two prespecialized versions of Floating Tape Loader-Monitor C with the program names AAAT3H and AAAT3T cannot be bootstrapped as they exist on the transaction BRT. However, as any program unit, the prespecialized versions could be loaded by a resident Loader-Monitor. If the user wishes to bootstrap a prespecialized version, he must execute an Update and Select run to assign the program name AAMON and a unique visibility code for the desired version while positioning it on the new master BRT. [1]

Once the BRT has been generated, it can be bootstrapped into memory, and the tape bootstrap routine will halt for a visibility assignment. The operator then specifies the visibility for the desired version of the Loader-Monitor to be used and activates processing, enabling the tape bootstrap routine to search for and load the specified Loader-Monitor.

Floating Tape Loader-Monitor C consists of two program segments: segment 01 is used only to search for, load, and start segment 02. Segment 02 is the actual Loader-Monitor and will occupy a variable number of memory locations starting at an address based upon the memory bank indicator specified either at specialization time or at execution time. If the Loader-Monitor has been specialized to be relocatable at execution time, segment 01 will halt after starting (halt number 2A on page 7-5) to allow control panel or console entry of the relocation bank indicator. Segment 01 will then relocate segment 02 into the indicated bank, make any prescribed changes in the Loader-Monitor communication area, and start segment 02. The Loader-Monitor then halts (halt number 3 on page 7-5) when it is ready to receive a call, which is the process of directing the Loader-Monitor to perform the three functions of searching for a specified unit, loading the particular unit, and, if directed, starting the unit.

Once a call has been made, Floating Tape Loader-Monitor C performs its functions according to parameters in its communication area, which occupies memory locations 64 through 189 (decimal). The Loader-Monitor establishes initial values in the parameter fields of the communication area insofar as the most commonly used value is predictable; thus, the programmer or operator is normally concerned with entering only a limited number of parameter values to specify the loading of a particular unit. Reset values are also established automatically in many of these same parameter fields. The user can enter parameters into the communication area by any of the three methods discussed in the subsequent paragraphs; for further explanation of any parameters and their initial and reset values mentioned therein, refer to Section III.

CALLING A PROGRAM UNIT

Calling a program unit begins with establishing the required parameter values in the communication area of the Loader-Monitor program. After these parameter values have been

[1] For further information concerning the use of the Update and Select program, refer to the Honeywell Software Bulletin entitled Update and Select C and D, Order No. 025.

established, the Loader-Monitor proceeds to locate and load the designated program unit; then, if specified, the Loader-Monitor starts the unit just loaded.

Types of Calls

The call for a program unit may be accomplished by any of three methods:

1. Manual call via the control panel or console;
2. Card call via the card reader using a Console Call card; and
3. Program call via program instructions in the unit currently being executed.

Of these methods, the second (card call) provides for entry of parameter values into a limited number of parameter locations; however, the first (manual call) and the third (program call) methods can be used to enter parameter values into any of the parameter locations. More specifically, the card call method provides a convenient means of entering a restricted number of parameter values; therefore, this method of parameter entry requires that the remaining parameter values be properly set as a function of the initial or reset values established by the Loader-Monitor. The three methods of entering parameter values into the communication area may be further categorized as follows: the first and second methods are referred to as console calls since they require operator action, whereas the third method is referred to as a program call since it is accomplished by instructions in the user's object program.

MANUAL CALL

Manual entry of parameter values is accomplished via the control panel or console. A manual call can be used to enter or change the value(s) in any of the parameter fields of the communication area. Normally, this method is used at an installation without a card reader; however, it can be used at any time as desired or needed.

CARD CALL

Search parameter values for program name, segment name, and program media (tape or card) can be entered into the communication area via the card reader, using a Console Call card. In addition, the Console Call card may contain a halt name when it is desired to halt after loading a particular program unit.

PROGRAM CALL

Once the first program unit has been located and loaded into memory using a manual or card call, subsequent program units may be located and loaded automatically (without requiring action by the operator). A program call is accomplished by including the appropriate instructions (to set up the parameter values required to call the next program unit) in the unit currently being executed. A program call may be coded in several ways at the programmer's discretion. Six situations are described in Section IV.

A special program call procedure provides for compatibility with the Basic Programming System. Using the special call procedure, the current program unit may call in the next unit without using program and segment names. Once the current unit terminates, the special call procedure activates a forward search for the next unit having the specified visibility, loading and starting that unit at its normal start location.

SEARCHING FOR A PROGRAM UNIT

Once the Loader-Monitor has received parameter specification for calling a program unit, it begins to search for the unit according to seven possible search parameters designated in the communication area:

1. Search Mode,
2. Search Direction,
3. Program Name,
4. Segment Name,
5. Visibility Mask,
6. Relative Position, and
7. Program Media.

The search mode parameter determines which of the remaining six parameters are to be considered during the search; for instance, a search mode of 01 designates the Loader-Monitor to search in a specified direction and load the nth unit of the specified visibility, where "n" is the binary value of the relative position parameter.

The search direction for a program unit may be either forward or backward. Note that card units can only be searched for in a forward direction.

The program name, segment name, and visibility parameters can be used in any combination as search keys to enable the Loader-Monitor to identify the program units.

The relative position parameter enables the user to reload the unit last loaded. This parameter is used in conjunction with the search mode when specified as 01 and the search direction is backward.

Program media designation is required to identify the media from which the program unit is to be loaded. When loading units from tape only, this parameter must designate the appropriate drive number. When loading units from both tape and cards, this parameter must be modified each time the media changes, designating either a tape drive number for a tape unit or a "C" for a card unit in the card reader.

Searching for Program Units on a BRT

As mentioned previously, program units on a BRT can be arranged in any sequence following the Loader-Monitor, as the BRT can be searched in a forward or backward direction. In searching forward, the Loader-Monitor checks banner characters for the identifying code of a segment header record (octal 50 or 54). A forward search may be initiated with the BRT positioned before any record between the header label and trailer label of the applicable file. A backward search may be initiated only with the BRT positioned immediately following the unit last loaded.

When an end-of-reserved information record (1ERIΔ) is encountered during a forward search prior to locating the requested program unit, the Loader-Monitor initiates a backward search for the unit. When the segment header record of the first program unit in the file (identified by the record sequence number Ø) is encountered during a backward search, a halt occurs, and a forward search may be initiated by pressing the RUN button or typing "G."

The search direction and the specific identification of the unit to be loaded are entered into the communication area using the applicable search parameters, as mentioned on page 2-4.

Searching for Loading Units in a BRD

In searching for a program unit on cards, Floating Tape Loader-Monitor C recognizes a segment header card by its banner character. However, a segment header card need not be the first card read after a search is initiated. Any non-header card is ignored.

In a search by program name, recognition is accomplished by comparing the program and segment name in the input buffer with the program and segment name in the Loader-Monitor communication area; the search is successful when a segment header card is found that contains the desired search parameter values (i.e., the corresponding program and segment name). Blank cards, whether placed between programs or within a program, cause unspecified results since they are interpreted by the Loader-Monitor as segment header cards.

NOTE: Card Loader-Monitor B has a search mode parameter entry of 77, which causes automatic loading of the next card unit found in the card reader; but Floating Tape Loader-Monitor C does not have this capability and requires a Console Call card in front of the first card unit to be loaded and any subsequent card unit not called by the preceding program unit.

The Loader-Monitor halts if a 1EOFΔ card is encountered in a search before the requested program unit is found.

LOADING A PROGRAM UNIT

After a program unit (the called unit) has been successfully located during the search phase, the Loader-Monitor proceeds to set up the called unit in memory. It does so by reading successive records of the called unit into the input buffer and subsequently loading their data contents into the specified memory locations with appropriate punctuation, as indicated by the data field control characters.[1] Loading is terminated by control character 61 (octal), and the BRT or BRD remains positioned immediately after the record in which control character 61 is encountered.

Loading Phase

Floating Tape Loader-Monitor C will load program units into the locations assigned to them during assembly; therefore, it is the user's responsibility to ensure that the area allotted to the Floating Tape Loader-Monitor C program is not destroyed. During the loading phase, the Loader-Monitor enters the program name and segment name of the called unit into locations 68 through 75 of the communication area. The revision number indicated in the segment header record of the called unit is also read into memory and stored in locations 65 through 67 of the communication area.

The operation of the Loader-Monitor may be modified, using one or both of the two load parameters: the relocation augment and the halt name. The relocation augment parameter provides the ability to load a unit into a higher memory location than that for which it was assembled. However, since the Loader-Monitor performs no address adjustment, the unit is loaded into the new area in the same form as it was assembled. Thus, the relocation augment is normally used in conjunction with object programs that contain indexed addressing. The halt name parameter can be used to cause a halt after loading a particular unit; the halt occurs if the name of that program unit is present in the halt name parameter locations in the communication area.

The Loader-Monitor also provides an own-coding exit after each record is read into the input buffer, thus enabling the execution of own-coding routines during loading (refer to Appendix B). In addition, an own-coding routine may return to the Loader-Monitor at return point 1 (172_8) and use the distribution process (loading data into memory with appropriate punctuation) of the Loader-Monitor; or the return may be to return point 2 (176_8), and the distribution process is assumed to have been executed by own-coding.

[1] For more information on data field control characters, refer to the Tape Loader-Monitor C Manual.

Using One Loader-Monitor to Load Another

The Floating Tape Loader-Monitor C program is capable of loading any of the Honeywell distributed Loader-Monitor programs if they have the program name AAAMON while they reside on the BRT. An additional feature of Floating Tape Loader-Monitor C is its ability to load other specialized versions of itself, regardless of the program name of the called version; similarly, any Loader-Monitor program on the BRT can load Floating Tape Loader-Monitor C regardless of the program name assigned to the called version.

The facility of interchanging Loader-Monitor programs provides for a continuous run within a system using two or more Loader-Monitor programs. Once one Loader-Monitor is in memory, the user need only supply that resident Loader-Monitor with the program and segment name and the unique visibility of the Loader-Monitor to be entered.

The user can display octal location 100 to determine whether Floating Tape Loader-Monitor C is in memory: a record mark indicates Floating Tape Loader-Monitor C, and a word mark indicates the presence of some other Loader-Monitor. The necessary information to search for load, and start the particular Loader-Monitor can be entered through one of the three types of calls. The program and segment names for any Mod 1 Tape Loader-Monitor (with the exception of the prespecialized versions: AAAT3H and AAAT3T) is AAAMONss, where ss is S1 for Tape Loader-Monitor C (both versions) or 01 for Floating Tape Loader-Monitor C.

Normally, all Tape Loader-Monitor programs in use at an installation are grouped at the beginning of the BRT. As a result, the loaders can be automatically included on the assembly output BRT. However, the various Loader-Monitor programs may be placed on the BRT in any desired order, according to the select directors used.[1] Depending upon the position of the BRT with relation to the Loader-Monitor to be called, the search direction parameter may be "forward" or "backward." Normally, this parameter should be "forward" if the BRT is in the rewound state and "backward" otherwise.

The following restrictions are imposed when using the above option:

1. The communication area will be redefined to be consistent with the Loader-Monitor being called;
2. Special or Return starts are not permitted;
3. The "halt name" option is not allowed; and
4. Own-coding is not permitted.

[1] See footnote on page 2-6.

STARTING A LOADING UNIT

The action performed by the Loader-Monitor after loading a called unit is referred to as "starting." Under the direction of the start mode parameter, the Loader-Monitor branches to one of three locations. The first of the three is a branch to the normal (N) starting location of the unit just loaded (called unit). The second is a branch to a special (S) location as designated by the calling unit, and the third is a return (R) branch to the calling unit.

If the Loader-Monitor program is assembled in three-character address mode, it starts the called unit by branching to the starting address in three-character mode. If the Loader-Monitor is assembled in four-character address mode, the starting mode will depend upon the address to which the branch is made. If this address is above 32, 767, the branch is executed in four-character mode; otherwise, it is executed in three-character mode.

Using Start Mode R

Start mode R can be entered by one program unit as part of a program call to load another unit. After the parameters have been entered by the calling unit, it can branch to the Loader-Monitor and then, once the called unit has been loaded, resume processing at the instruction following that branch. For start mode R to be used when a calling unit in four-character address mode executes a branch to the Loader-Monitor from a location above 32, 767, the branch must be to location 168 (decimal) and is coded as follows: B/ (168).

SECTION III

PARAMETERS OF THE COMMUNICATION AREA

INTRODUCTION

The communication area for Floating Tape Loader-Monitor C occupies memory locations 64 through 189 (decimal), and object programs must be assembled above this area. Locations 64 through 155 of the communication area are identical for any MOD 1 Loader-Monitor program; but locations 156 through 189 are used only with Floating Tape Loader-Monitor C, and certain of these locations are used only if the Loader-Monitor is assembled in four-character address mode.

In addition to the various parameters employed during the search, load, and start phases of the Loader-Monitor, the communication area has several other locations available for use and reference by object programs. A summary of each parameter field in the communication area is given in this section, and a sequential layout of the communication area is given in Table 3-4.

Floating Tape Loader-Monitor C establishes initial and reset values for many of the parameter fields involved in the searching, loading, and starting functions. "Initial value", as used in the following paragraphs, refers to the contents of a parameter field within the communication area which are established when the Loader-Monitor is initially loaded into memory. "Reset" refers to a value entered by the Loader-Monitor into a parameter field within the communication area during execution. Refer to Table 3-3 for a summary of these values.

SEARCH PARAMETERS

Searching for a specific program unit is accomplished by the specialization of seven search parameters in the communication area, as mentioned on page 2-4.

Search Mode (Location 111)

The search mode is established by entering one of the five search mode designations into location 111 (octal 157). The octal values of the five search mode designations and the meaning of each are provided in Table 3-1, together with the identifying parameters used when searching in the designated mode.

NOTE: The search mode is always 20 when searching for card units.

The initial value of the search mode parameter is 20, and it is reset to 20 by a console call. However, upon receipt of a special program call (Section II), the search mode parameter is reset to 01.

Table 3-1. Search Mode Designations

Search Mode (Octal)	Parameters Used	Meaning
01	Visibility and relative position	Search in the specified direction, and load the nth unit of specified visibility (n is the binary value of the relative position parameter).
20	Program and segment names	Search for and load the unit with the specified program and segment names, irrespective of visibility.
00	Segment name	Search for and load the unit with the specified segment name within the current program. (Similar to search mode 20 except that a halt occurs upon encountering a segment header record with a program name different from that of the unit last loaded. A search in the opposite direction can then be initiated by pressing RUN or typing "G".)
60	Program name, segment name, visibility	Search for and load the unit with the specified program name, segment name, and visibility.
40	Segment name, visibility	Search for and load the unit with the specified segment name and visibility within the current program. (This mode is similar to 00 with the addition of the visibility mask.)

Search Direction (Location 106)

The search direction is established by entering octal 22 (search forward) or octal 23 (search backward) into location 106 (octal 152). The initial value of the search direction parameter is octal 22. It is reset to 22 by a console call, and it is reset to 22 after loading each unit. Note that the search direction for card units must always be forward.

Program Name (Locations 68-73)

The program name is one of the search parameters (keys) used in search modes 20, 00, 60, and 40. Thus, in all search modes except 01, the calling unit (or the calling or loading operation of a previously loaded unit) enters the program name of the specified loading unit into locations 68 to 73 (octal 104 to 111). However, regardless of the search mode used, the program name of the applicable loading unit is always entered into locations 68 to 73 by the Loader-Monitor during the loading operation, thus enabling subsequent use of search modes 00 and 40 without the necessity of entering the program name.

Segment Name (Locations 74-75)

Similar to the program name, the segment name is one of the search parameters used in search modes 20, 00, 60, and 40; thus, for these modes, the segment name is entered by the calling unit into locations 74 and 75 (octal 112 and 113). Similarly, the segment name is always entered into locations 74 and 75 by the Loader-Monitor during the loading operation.

Visibility Mask (Locations 113-118)

Visibility is one of the search parameters (keys) used in search modes 01, 60, and 40. The initial value of the visibility mask in the communication area (as established by the Loader-Monitor) is octal 40 00 00 00 00 00 (visibility A). When searching by visibility, a visibility match must be obtained to identify the desired loading unit prior to loading. A visibility match occurs when at least one bit position in the visibility key of the program unit and the corresponding bit position of the visibility mask both contain a 1. When it is desired to search for a program unit by visibility other than visibility A, the desired visibility code must be entered into locations 113 to 118 (octal 161 to 166) by either manual or program entry as described in Section IV.

Relative Position (Location 110)

The relative position parameter is a binary number (n) that is used in conjunction with search mode 01. The initial value of the relative position parameter is 1. With a relative position value of 1, the unit last loaded may be reloaded if the search direction parameter is set for backward search. The relative position parameter is reset to 1 by a console call, and it is reset to 1 after loading each unit. The value 0 for the relative position parameter is undefined.

Program Media (Location 76)

Since Floating Tape Loader-Monitor C can load program units from tape or cards, the appropriate loading media parameter must be specified in location 76 (octal 114). When a tape unit is to be called, location 76 must contain the number of the tape drive containing the called unit; and when a card unit is to be called, location 76 must contain a "C" to designate that the unit called is in the card reader.

If a program unit is to be called from tape, the low-order three bits of the character which is entered in location 76 specify the number of the tape drive containing the desired program unit. (The peripheral instructions are specialized from this character before executing each call.) In contrast to the tape drive, the tape control unit address is not designated by a search parameter since the address should have been entered during the bootstrap procedure (see page 7-2). The tape bootstrap routine and the Loader-Monitor both direct the use of the tape control address in forming PDT instructions, and the user need only designate the tape drive number to complete the instruction.

Intermixing tape and card program units may easily be accomplished by the Console Call card procedure, where column 9 of each Console Call card designates the program media of the unit to be called. The entry in column 9 is read into location 76.

LOAD PARAMETERS

Relocation Augment (Locations 107-109)

The relocation augment parameter is a three-character (18-bit) binary number that is entered into locations 107 to 109 (octal 153 to 155) of the communication area. The relocation augment is added to the starting location of each load string by the Loader-Monitor, and it is also added to the program unit's starting location address when the start parameter is set to normal (N).[1] In addition, the relocation augment is added to the high-and low-order addresses of an area to be cleared. The initial value (established by the Loader-Monitor) of the relocation augment parameter is 0; it is reset to 0 after loading each unit, and it is reset to 0 by console call.

Halt Name (Locations 77-84)

The halt name parameter is an eight-character combined program and segment name that is entered into locations 77 to 84 (octal 115 to 124) of the communication area. Although the program name is entered into locations 77 to 82 (octal 115 to 122) and the segment name is entered into locations 83 and 84 (octal 123 and 124), the two names constitute the "halt name," and locations 77 to 84 are treated as one field with only one word mark set in the field at location 77. Before loading begins, the Loader-Monitor compares the name in the halt name parameter field with the name on the segment header record of each program unit that has been successfully located during the search phase. If the comparison indicates that the specified program unit has been identified, a halt occurs after loading the specified unit. Depressing the RUN button or typing "G" causes the Loader-Monitor to continue as directed by the start parameters.

START PARAMETERS

Start Mode (Location 112)

One of the three start mode parameter values is entered into location 112 (octal 160) of the communication area to specify the method of initiating the start phase of the Loader-Monitor program. The value and meaning of each of the three start mode parameters are listed in Table 3-2.

The initial value of the start mode parameter (as established by the Loader-Monitor program) is N; it is reset to N by a console call or a special call (Section II).

Special Start Location (Locations 119-121)

The special start location parameter is used in conjunction with the start mode parameter

[1] A load string consists of data and control characters formatted during assembly of the specific program unit.

value S. Specifically, the special start location parameter value is a three-character address, entered into locations 119 to 121 (octal 167 to 171) of the communication area. The Loader-Monitor program branches to the special start location to start the called program when the start mode parameter value entered in location 112 is S. The initial value of the special start location parameter is 0.

Table 3-2. Start Mode Designations

Parameter Value		Meaning
Alphabetic	Octal	
N	45	Branch to the location specified as the normal starting location in the called program unit. (The relocation augment is added to this address by the Loader-Monitor before executing the branch.)
S	62	Branch to the location specified by the "special start location" parameter value entered in locations 119 to 121. (The relocation augment is not added to this address.)
R	51	Branch to the location in the calling program (unit) immediately following the location from which the call to the Loader-Monitor was made (see page 2-8). For start mode R to be used when the calling program executes a branch to the Loader-Monitor from a location above 32,767, the branch must be to location 168 (decimal) and is coded as follows: B/ (168).

Trapping Mode (Location 147)

The trapping mode parameter may be used in conjunction with any of the three start mode parameter values N, S, or R. If the trapping mode is specified, any instruction whose op code contains an item or record mark is both extracted and executed as if it were a Change Sequencing Mode instruction, regardless of the actual op code present. The trapping mode is established by entering the octal value 04 into location 147 (octal 223). Once the trapping mode has been entered, the value 00 must be entered into location 147 when it is desired to return to the non-trapping mode. Immediately before starting a called program, the Loader-Monitor references the trapping mode parameter value and establishes either the non-trapping mode (00) or the trapping mode (04). Thus, a calling program, or the operator, can designate whether the trapping mode will be in effect when the called unit is started. The initial value of the trapping mode parameter is 00.

Fixed Starts

The communication area contains four Branch instructions that may be used for fixed starts. The first Branch instruction is designated as fixed-start 0, and it is stored in locations 66 to 89 (octal 126 to 131). The execution of fixed start 0 prepares the Loader-Monitor to perform the functions designated as "reset at console call" (see Table 3-3) and then causes a halt to occur, awaiting the entry of the desired parameters (see page 7-3). The other three Branch instructions (locations 90 through 101) are available for use by object programs. The Branch op code is located in the leftmost location of each four-character field, followed by the three-character A-address portion.

RETURN PARAMETERS

Three-Character Mode Returns

The communication area includes a general return feature that enables all program units of a particular system to terminate with a Branch instruction to the same return address or location. When Floating Tape Loader-Monitor C is assembled in three-character address mode, returns are available for program units to prepare the Loader-Monitor to accept or to execute a console call.

GENERAL RETURN ADDRESS

Locations 139 to 141 (octal 213 to 215) contain the general return address that is initially set by the Loader-Monitor to the same address referenced by fixed-start 0; therefore, branching to this address prepares the Loader-Monitor for a console call. To use this feature, each program unit of a system terminates with the standard instruction B/(139).

If a resident control program is to govern the loading of program units in a system, the control routine must set locations 139 to 141 to an address within itself. Once the prescribed units of the system have been processed, the control routine must reset the general return address to its initial value.

ALTERNATE CONSOLE CALL RETURN ADDRESS

The communication area also contains an alternate return feature to enable a program unit to direct the Loader-Monitor to search for a Console Call card without halting or resetting any parameter values. This feature is activated when the program unit executes the instruction B/(148).

PROGRAMMED CALL RETURN LOCATION

Locations 130 to 138 (octal 202 to 212) contain a branch instruction to a start location in the Loader-Monitor which assumes that the parameters to search for, load, and start a unit have been entered by a programmed call. The calling unit can use this feature by terminating with the instruction B/130.

Four-Character Mode Returns

When Floating Tape Loader-Monitor C is assembled in four-character mode, additional return features are made available to the user.

GENERAL RETURN ADDRESS

Locations 164 to 167 (octal 244 to 247) contain a general return address which allows a program unit assembled in four-character mode to return to the Loader-Monitor in four-character mode and to prepare the Loader-Monitor to accept a console call. To use this feature, the program unit terminates with the instruction B/(164).

As with the general return address for three-character mode, a resident control routine may change the four-character general return address to some address within itself, and thereby govern the loading of prescribed units in a system (see three-character general return address above).

PROGRAMMED CALL RETURN ADDRESS

Locations 168 to 171 (octal 250 to 253) contain a return address causing control to revert to a start location in the Loader-Monitor which assumes that the parameters to search for, load, and start a unit have been entered by a programmed call. The calling unit can use this feature by terminating with the instruction B/(168).

SUPPLEMENTAL FEATURES OF THE COMMUNICATION AREA

The remaining locations of the communication area are to be used and referenced by the Loader-Monitor, Interrupt Control D, and object programs, as specifically stated below.

Current Date

The communication area contains a five-character current date field in locations 142 to 146 (octal 216 to 222), into which the operator may load the current date. The Loader-Monitor makes no use of the current date field; it is included only for the convenience of object programs.

ECD Field

The communication area contains a four-character ECD field in locations 151 to 154 (octal 227 to 232) which is not used by Loader-Monitor. It is used by systems programs in conjunction with the standard Equipment Configuration Descriptor (ECD). The initial value of the ECD field is JJ0#.

Console Typewriter Availability

Location 155 (octal) of the communication area contains the control unit number of the console typewriter when designated during specialization of the Loader-Monitor. If a console typewriter is present and designated during specialization, location 155 is automatically punctuated to contain a word mark only. If a console typewriter is not present or specified, location 155 is punctuated to contain an item mark only. Care should be taken to avoid changing the punctuation in this location.

Load Indicator for Foreground Program

When Interrupt Control D is interacting with Floating Tape Loader-Monitor C and controlling the execution of a foreground and background program, location 267 (octal) of the Loader-Monitor communication area is interrogated by Interrupt Control D to determine whether another foreground program is to follow the current foreground program being processed. An "*" in octal location 267 signifies that another foreground program is to be loaded after the current foreground program has been executed; a "0" in the location designates that no other foreground programs are to follow the current foreground program being processed.

The load indicator feature enables sequencing of foreground programs for the user having a central processor with the 012 Program Interrupt Feature but having a control panel or console without the external interrupt button.[1] Foreground programs can be sequenced if an asterisk (54_8) is placed in octal location 267 of the Loader-Monitor communication area prior to the exiting of the current foreground program being processed. Interrupt Control D interrogates location 267 when the foreground program exits; and, if the location contains an asterisk, Interrupt Control D gives control to the Loader-Monitor to load the next foreground program. If no asterisk has been specified in octal location 267, Interrupt Control D allows the background program to continue in sequence; and, if there is no external interrupt button, another foreground program cannot be loaded until the background program has completed processing.

For more extensive information on Interrupt Control D, refer to Section VIII.

Systems Communication Area

A portion of the communication area of Floating Tape Loader-Monitor C is included with that portion of the Loader-Monitor which resides in upper memory. Information in this area is needed only by systems programs, in particular, Interrupt Control D. Decimal locations 184 through 186 contain the beginning address of the systems communication area.

Object Program Memory Limit

Decimal locations 187 through 189 contain the address of the highest location which can be used by an object program. Floating Tape Loader-Monitor C and other systems programs, such as Interrupt Control D, reside above this address.

[1] The load indicator feature would be used at an installation with or without a Type 220-1 console and having a Type 201-0 or 201-1 central processor without an external interrupt button on the control panel.

Table 3-3. Summary of Loader-Monitor Parameters

Phase	Parameter	Location		Values	Initial Value	Reset at Console Call	Reset After Loading	Reset at Special Call
		Decimal	Octal					
Presearch	Method of parameter entry	64	100	00: console call card. 01: manual entry via operator's control panel (or console).	00	-	-	-
Searching	Search mode	111	157	01: vis. & rel. pos. 20: prog. & seg. 00: seg. (within prog.) 60: prog., seg., vis. 40: seg. & vis. (within prog.)	20	20	-	01
	Search direction	106	152	22: forward 23: backward	22	22	22	-
	Program name	68-73	104-111	-	-	-	-	-
	Segment name	74-75	112-113	-	-	-	-	-
	Visibility	113-118	161-166	-	40 00 00 00 00 00	-	-	-
	Relative position	110	156	1 (back.): prog. last loaded. 1 (fwd): next prog.	1	1	1	-
	Program media	76	114	d: tape unit (d=drive no.) c: card unit	0	-	-	-
Loading	Relocation augment	107-109	153-155	-	0	0	0	-
	Halt name	77-84	115-124	-	-	-	-	-
Starting	Start mode	112	160	N: normal S: special R: return	N	N	-	N
	Special start location	119-121	167-171	-	0	-	-	-
	Trapping mode	147	223	00: off 04: on	00	-	-	-

Table 3-4. Sequential Layout of the Loader-Monitor Communication Area

Location		Function
Decimal	Octal	
64	100	Console call designation (card or manual entry).
65-67	101-103	Revision number of unit last loaded.
68-73	104-111	Program name.
74-75	112-113	Segment name.
76	114	Program media (tape or card).
77-84	115-124	Halt name.
85	125	Asterisk (ID character from column 18 of Console Call card).
86-89	126-131	Fixed start 0 (Loader-Monitor return for console call).
90-93	132-135	Fixed start 1 (not used by Loader-Monitor).
94-97	136-141	Fixed start 2 (not used by Loader-Monitor).

Table 3-4 (cont). Sequential Layout of the Loader-Monitor Communication Area

Location		Function
Decimal	Octal	
98-101	142-145	Fixed start 3 (not used by Loader-Monitor).
102-105	146-151	Own-code entrance (Branch to own-code start address).
106	152	Search direction.
107-109	153-155	Relocation augment.
110	156	Relative position.
111	157	Search mode.
112	160	Start mode.
113-118	161-166	Visibility mask.
119-121	167-171	Special starting location.
122-125	172-175	Own-code return 1 (before distribution).
126-129	176-201	Own-code return 2 (after distribution).
130-138	202-212	Programmed call return location (3-character mode).
139-141	213-215	General return address (3-character mode).
142-146	216-222	Current date.
147	223	Trapping mode.
148-150	224-226	Alternate console call return address (3-character mode).
151-154	227-232	ECD field.
155	233	Console typewriter availability.
156-163	234-243	Reserved for Loader-Monitor only.
164-167	244-247	General return address (4-character mode). *
168-171	250-253	Programmed call return address (4-character mode). *
183	267	Foreground program load indicator.
184-186	270-272	Base address of systems communication area.
187-189	273-275	Object program memory limit.

*Available if the Loader-Monitor is assembled in four-character address mode.

SECTION IV

ENTERING PARAMETER VALUES INTO THE COMMUNICATION AREA

INTRODUCTION

As mentioned in Section II, there are three methods of entering parameters: manually (at the control panel or console), via a Console Call card, and by program instructions in a previously loaded program unit. The following paragraphs are provided to further discuss the application of the initial and reset parameter values established by the Loader-Monitor and to offer additional information on the three methods of parameter entry, especially the program method.

INITIAL AND RESET VALUES

The parameter fields of the Loader-Monitor communication area contain the initial values which have been predicted as the values most commonly used when loading program units from a BRT. The Loader-Monitor also establishes reset values in many of these same parameter fields for one or more of the three reset conditions: at console call, after loading a unit, and at special call (see Table 3-3). Although the initial and reset values are described individually for each parameter in Section III, they are summarized below to correlate their use with the three methods of entering parameter values into the communication area.

1. The initial value for search mode is octal 20 (search by program and segment names); it is reset to 20 by a console call and reset to 01 (search by visibility and relative position) by a special call.
2. The initial value for search direction is octal 22 (search forward); it is reset to 22 after loading each unit, and it is reset to 22 by a console call.
3. The initial value of the six-character program name is blank; thus, the program name must be entered by one of the three methods described in this section if it is to be used as a search key. Once the program name is entered into the communication area, search modes 00 and 40 may be used.
4. The initial value of the two-character segment name is also blank, and its characteristics are the same as those of the program name described above.
5. The initial value of the visibility mask is octal 40 00 00 00 00 00 (visibility A).
6. The initial value for relative position is 1; it is reset to 1 after loading each unit, and it is reset to 1 by a console call.
7. The initial value for program media is 0 (designating tape drive 0).
8. The initial value for relocation augment is 0; it is reset to 0 by a console call, and it is reset to 0 after loading each unit.
9. The initial value of the eight-character halt name is blank; thus, if it is desired to halt after loading a specified program unit, the program and segment

names of the specified unit must be entered into the halt name parameter location by one of the three methods described in this section.

10. The initial value for start mode is N (normal start location); it is reset to N by a console call and by a special call.

11. The initial value for special start location is 0; thus, if it is desired to use a special start location, the desired address must be entered into the special start location field. This field is not reset.

It may be observed from the preceding summary of the initial and reset values that loading most programs (units) requires the entry of only a very few parameter values. For example, a special program call (Easycoder Assembler A) does not require the entry of any parameter values when using the initial values and searching by visibility A. Loading and starting some Easycoder C programs may require only the entry of octal 01 in the search mode location when searching by visibility A and using a relative position value of 1; and many additional programs will require only two entries when searching by program and segment names. The following paragraphs provide a detailed description of the three methods of entering parameter values to supplement the initial and reset values established by the Loader-Monitor.

METHOD OF PARAMETER ENTRY (LOCATION 64)

To designate whether manual or card entry is to be used, each method requires that its appropriate designator be entered in location 64 (octal 100) of the communication area. The value 0 is used in conjunction with a card entry, and the value 1 is used in conjunction with manual entry. The "method of parameter entry" designator has an initial value of 0 established by the Loader-Monitor. Manual entry via the control panel or console requires that the octal value 01 first be entered (manually) into octal location 100. Once this value has been entered, it must be manually reset to octal 00 before card entry (Console Call card) can again be used.

MANUAL ENTRY

Since manual entry of parameter values requires that each character of a value be entered at the control panel or console, the method becomes most useful when making changes to specific values in the communication area. For instance, manual entry could be used to place addresses in the fixed-start fields provided in the communication area for the use of object programs, thereby providing a method of restarting during the execution of a program unit. Control panel and console procedures for manual entry are given on page 7-3.

CARD ENTRY

A Console Call card can be used to enter the parameter values for program name, segment name, and program media; a halt name may also be included if it is desired to halt after loading a particular program unit. The Console Call card must be punched in the standard Honeywell code since the card reader is conditioned by a PCB instruction from the Loader-Monitor to read in "special" (Honeywell) mode before the PDT instruction is issued to read the card. The Console Call card is read directly into the communication area, overlaying the previous values of the four corresponding parameters; therefore, column 9 must always be punched even though it is the same as the current or initial value. As illustrated in Figure 4-1, the Console Call card is only read through column 18; the remaining columns are not used by the Loader-Monitor.

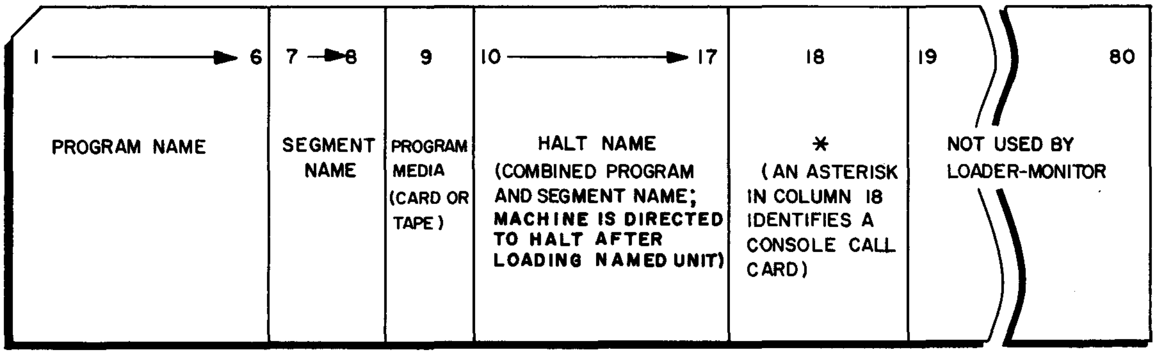

1 → 6
PROGRAM NAME
7 → 8
SEGMENT NAME
9
PROGRAM MEDIA (CARD OR TAPE)
10 → 17
HALT NAME (COMBINED PROGRAM AND SEGMENT NAME; MACHINE IS DIRECTED TO HALT AFTER LOADING NAMED UNIT)
18
*
(AN ASTERISK IN COLUMN 18 IDENTIFIES A CONSOLE CALL CARD)
19
80
NOT USED BY LOADER-MONITOR

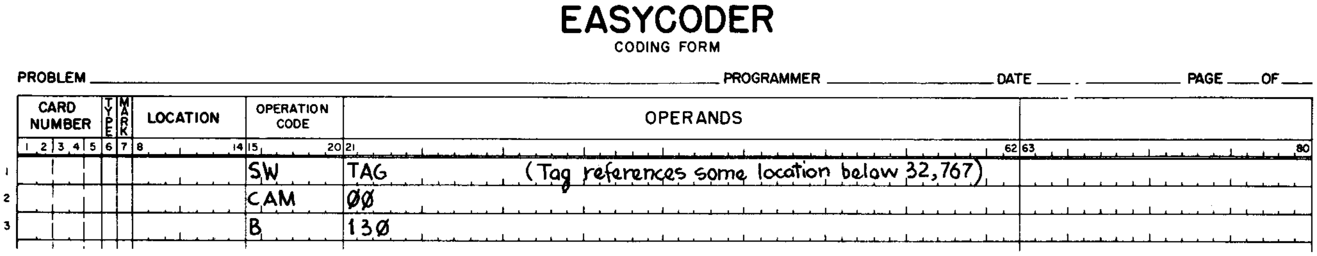
Figure 4-1. Console Call Card

PROGRAM ENTRY

A call to search for and load a particular program unit may originate from instructions in the current program. A normal call to the Loader-Monitor from the current program is executed by (1) ensuring that the parameters in the Loader-Monitor communication area are properly set, and (2) branching either to location 130 (directly) for entry in three-character addressing mode or to location 168 (indirectly) for entry in four-character mode (see Table 3-4). The calling unit must execute the branch in an admode consistent with the entry. Depending upon the value of the start mode parameter in decimal location 112, the Loader-Monitor either starts the called unit or returns to the calling unit.

If Floating Tape Loader-Monitor C is in the three-character addressing mode, it starts the called unit or returns to the calling unit in three-character mode. When the Loader-Monitor is in four-character mode, the addressing mode in which it starts a calling unit will depend upon the address of the start location that is branched to. If this address is above 32,767, the branch is executed in four-character mode; otherwise, it is executed in three-character mode.

When a program operating above 32K branches to a three-character entrance of the Loader-Monitor, it must ensure that the A-address register contains an address less than 32,768. The following sequence of instructions illustrates one way in which this can be accomplished (the SW is executed in the four-character addressing mode).



EASYCODER
CODING FORM

PROBLEM ______ PROGRAMMER ______ DATE ______ PAGE ___ OF ___

CARD NUMBER	TYPE	MARK	LOCATION	OPERATION CODE	OPERANDS	
1				SW	TAG (Tag references some location below 32,767)	
2				CAM	ØØ	
3				B	13Ø	

In setting up parameters, the calling unit must not alter any punctuation marks. All fields in the communication area are initially loaded by the Loader-Monitor with word marks in their leftmost character locations, except for the console typewriter availability field (octal 233), which may have an item mark if the machine configuration does not include a typewriter. To identify Floating Tape Loader-Monitor C, the field for console call entry (octal 100) has a record mark which must not be altered.

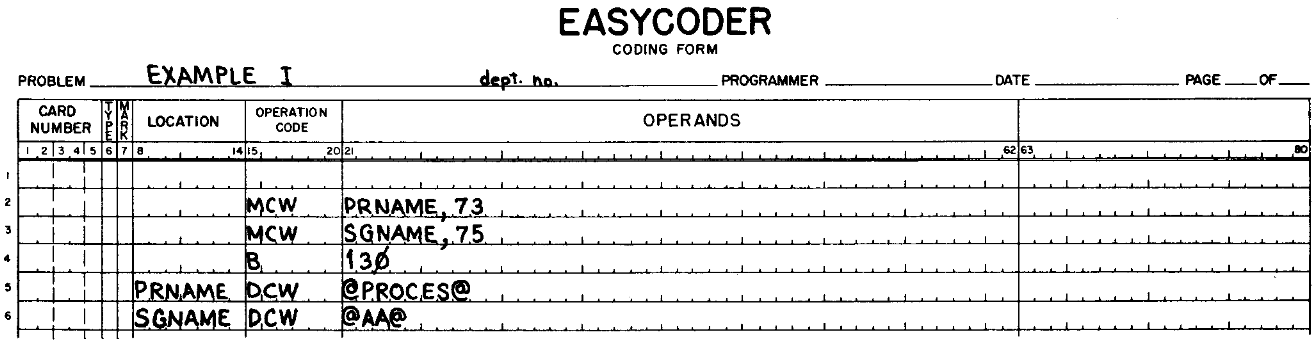
The Loader-Monitor uses and does not restore index registers X5 and X6. These registers have word marks in their leftmost character locations at the completion of loading; the specific locations are dependent upon the address mode of the Loader-Monitor (i.e., locations 22 and 26 for three-character mode and locations 21 and 25 for four-character mode). Index register X6 is the distribution counter which contains an address one higher than that into which the last character of the called unit was loaded. The Loader-Monitor does not use or disturb any locations below location 64 (octal 100) other than index registers X5 and X6.

Examples of Programmed Calls

To aid the programmer in coding the parameter entries, six situations with their corresponding solutions are given below.

EXAMPLE 1

Call program unit named PROCES AA, search in the forward direction by program name and segment name, and start PROCES AA at its normal starting location (established during assembly). Note that the coding in Figure 4-2 does not include entries for search mode, search direction, or start mode, since the desired values for these parameters are the initial values established by the Loader-Monitor. The solution for example 1 given in Figure 4-2 assumes that PROCES AA is on tape 0.



EASYCODER
CODING FORM

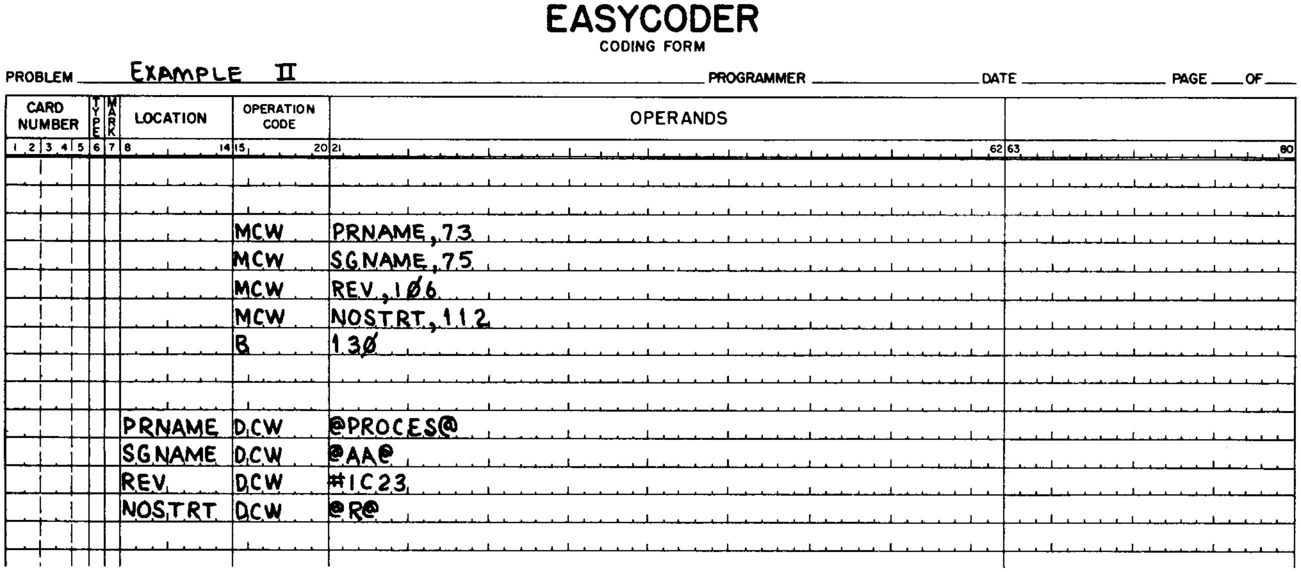
PROBLEM EXAMPLE I dept. no. PROGRAMMER ______ DATE ______ PAGE ___ OF ___

CARD NUMBER	TYPE	MARK	LOCATION	OPERATION CODE	OPERANDS	
1						
2				MCW	PRNAME, 73	
3				MCW	SGNAME, 75	
4				B	13Ø	
5			PRNAME	DCW	@PROCES@	
6			SGNAME	DCW	@AA@	

Figure 4-2. Symbolic Coding for Example 1

EXAMPLE 2

Call the program unit named PROCES AA; search in the backward direction; and do not start execution after loading PROCES AA, but return to the next instruction in the calling unit (the instruction which immediately follows B/130). Note that the search direction and start mode specified in this example are non-standard; therefore, the required values for these parameters must be entered by the calling program, as shown in Figure 4-3.



EASYCODER
CODING FORM

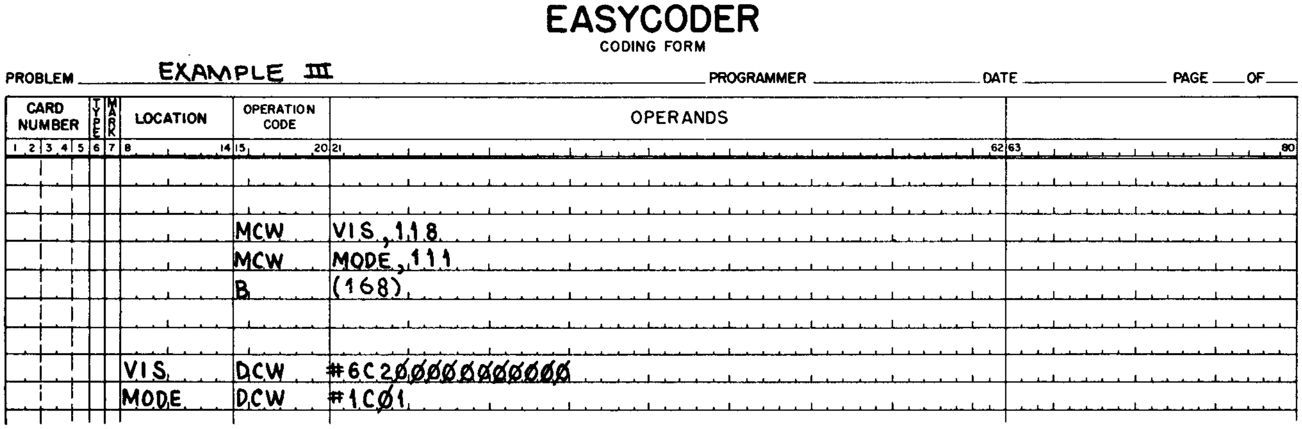
PROBLEM EXAMPLE II PROGRAMMER DATE PAGE OF

```
         MCW     PRNAME,73
         MCW     SGNAME,75
         MCW     REV,1Ø6
         MCW     NOSTRT,112
         B       13Ø


PRNAME   DCW     @PROCES@
SGNAME   DCW     @AA@
REV      DCW     #1C23
NOSTRT   DCW     @R@
```

Figure 4-3. Symbolic Coding for Example 2

EXAMPLE 3

Use the four-character general return address to call the next program unit that is identified by visibility B, search in the forward direction, and start the specified unit at the normal starting location (see Figure 4-4).



EASYCODER
CODING FORM

PROBLEM EXAMPLE III PROGRAMMER DATE PAGE OF

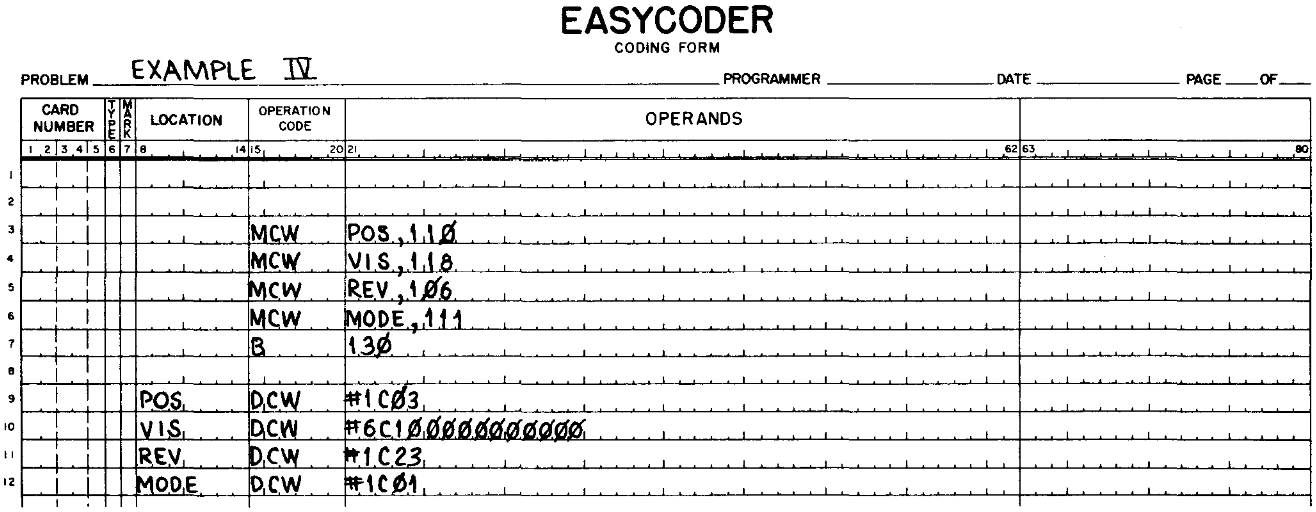
```
         MCW     VIS,118
         MCW     MODE,111
         B       (168)


VIS      DCW     #6C2ØØØØØØØØØØØ
MODE     DCW     #1CØ1
```

Figure 4-4. Symbolic Coding for Example 3

EXAMPLE 4

Call the third program unit that is identified by visibility C, search in the backward direction, and start the specified unit at its normal starting location (see Figure 4-5). The Loader-

Monitor directs a backward search of the program units stored on the BRT and increments a counter by one each time a unit of visibility C is encountered. When the count reaches three, the specified program unit has been located and is subsequently loaded and started at its normal starting location.



EASYCODER
CODING FORM

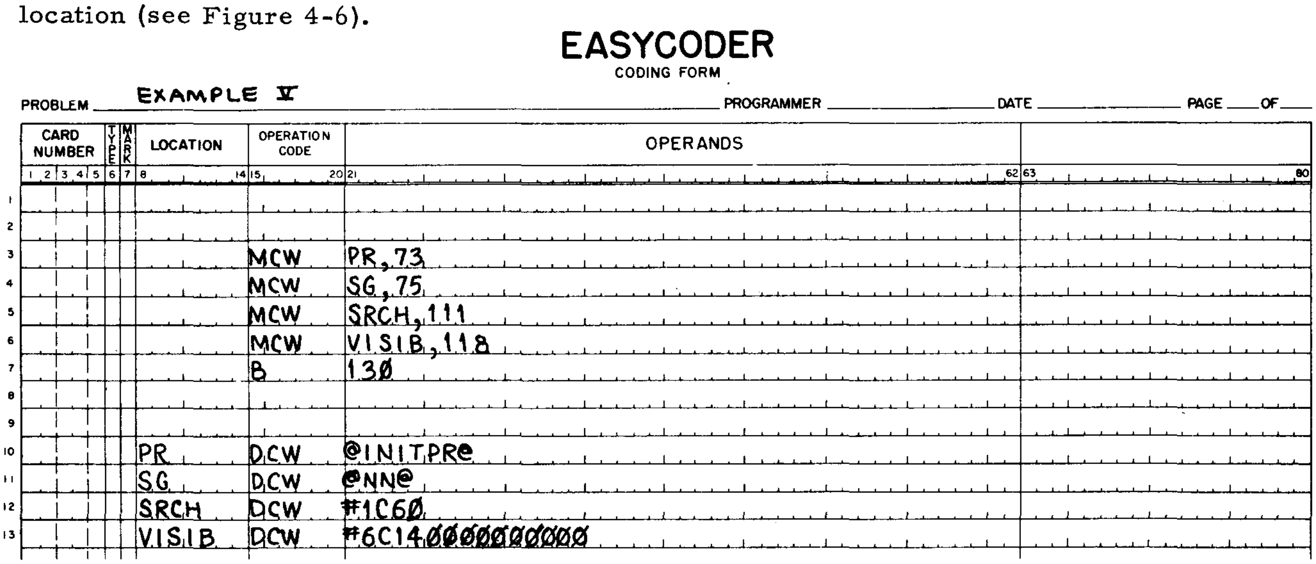
PROBLEM EXAMPLE IV PROGRAMMER ______ DATE ______ PAGE ___ OF ___

	CARD NUMBER 1 2 3 4 5	TYPE 6	MARK 7	LOCATION 8 14	OPERATION CODE 15 20	OPERANDS 21 62	63 80
1							
2							
3					MCW	POS,11Ø	
4					MCW	VIS,118	
5					MCW	REV,1Ø6	
6					MCW	MODE,111	
7					B	13Ø	
8							
9				POS	DCW	#1CØ3	
10				VIS	DCW	#6C1ØØØØØØØØØØØ	
11				REV	DCW	#1C23	
12				MODE	DCW	#1CØ1	

Figure 4-5. Symbolic Coding for Example 4

EXAMPLE 5

Call the program unit named INITPR NN that also is identified by either visibility C or visibility D, search in the forward direction, and start the specified unit at its normal starting location (see Figure 4-6).



EASYCODER
CODING FORM

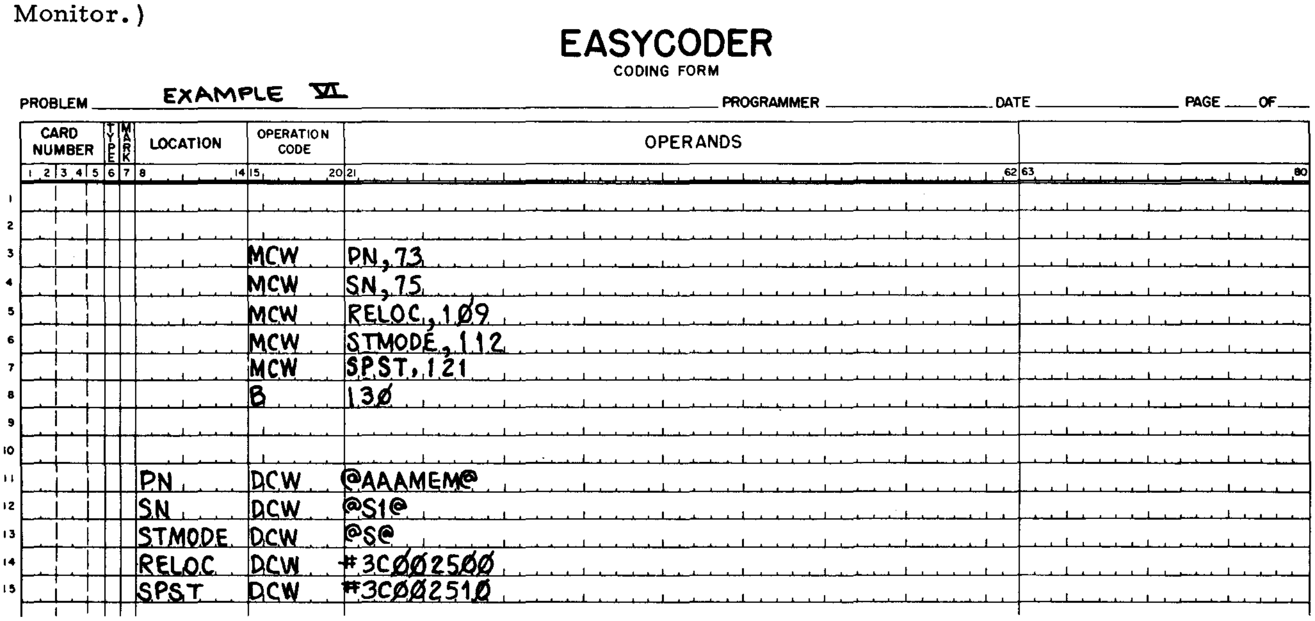
PROBLEM EXAMPLE V PROGRAMMER ______ DATE ______ PAGE ___ OF ___

	CARD NUMBER 1 2 3 4 5	TYPE 6	MARK 7	LOCATION 8 14	OPERATION CODE 15 20	OPERANDS 21 62	63 80
1							
2							
3					MCW	PR,73	
4					MCW	SG,75	
5					MCW	SRCH,111	
6					MCW	VISIB,118	
7					B	13Ø	
8							
9							
10				PR	DCW	@INITPR@	
11				SG	DCW	@NN@	
12				SRCH	DCW	#1C6Ø	
13				VISIB	DCW	#6C14ØØØØØØØØØØ	

Figure 4-6. Symbolic Coding for Example 5

EXAMPLE 6

Call the program unit named AAAMEM S1, search in the forward direction, relocate the loading location by a relocation augment value of octal 2500, and start AAAMEM S1 at octal location 2510 (see Figure 4-7). (The octal number 2500 is added only to the normal loading location established during assembly. Thus, if the normal loading location were established as 0 by assembly, AAAMEM S1 would now be loaded into memory, starting at octal location 2500. Note that the operand addresses of instructions within the program are not altered by the Loader-Monitor.)



EASYCODER
CODING FORM

PROBLEM EXAMPLE VI PROGRAMMER ____ DATE ____ PAGE ___ OF ___

	CARD NUMBER (1–5)	TYPE (6)	MARK (7)	LOCATION (8–14)	OPERATION CODE (15–20)	OPERANDS (21–62)	(63–80)
1							
2							
3					MCW	PN,73	
4					MCW	SN,75	
5					MCW	RELOC,1Ø9	
6					MCW	STMODE,112	
7					MCW	SPST,121	
8					B	13Ø	
9							
10							
11				PN	DCW	@AAAMEM@	
12				SN	DCW	@S1@	
13				STMODE	DCW	@S@	
14				RELOC	DCW	#3CØØ25ØØ	
15				SPST	DCW	#3CØØ251Ø	

Figure 4-7. Symbolic Coding for Example 6

Special Programed Call

A special program calling procedure provides for compatibility with the Basic Programming System. Using the special calling procedure, the current program unit must include an instruction at its starting location to store the contents of the B-address register; the next unit can then be called by terminating the called unit with a branch to the stored address. Note that the branch must be in an admode consistent with the admode in which the current unit was started. This action reverts control to the Loader-Monitor to activate a forward search for the next unit having the specified visibility and causes that unit to be loaded and started at its normal starting location.

SECTION V

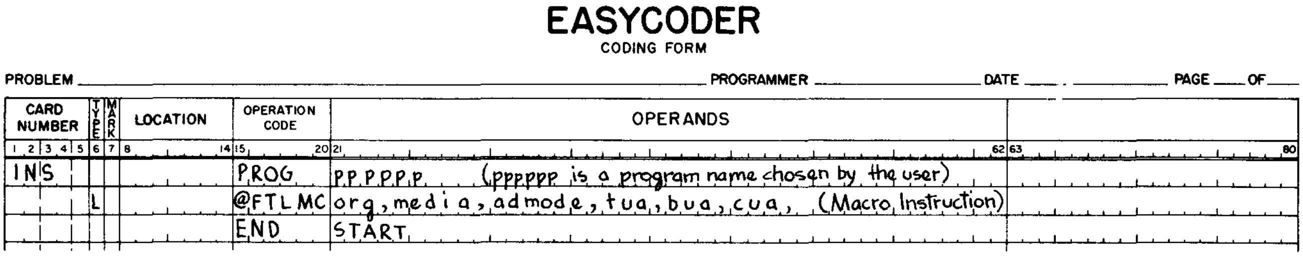
MACRO SPECIALIZATION OF FLOATING TAPE LOADER-MONITOR C

INTRODUCTION

In order to accommodate the various options of the Floating Tape Loader-Monitor C program, it has been written as a generalized macro routine, which requires specialization by the Library Processor C program before assembly by Easycoder Assembler C. Two prespecialized versions of the Loader-Monitor are available for the most probable applications (see page 5-2). The user that wishes to specialize his own routine must write a macro instruction, given below, for input to Library Processor C.

INPUT FOR MACRO SPECIALIZATION

The Easycoder coding form below lists the three instructions needed to specialize the Loader-Monitor. During specialization, the user must substitute any program name for "pppppp" other than "AAAMON." The name specified will be recorded on the assembly output symbolic program tape (SPT) and transaction binary run tape (TBT). The segment names for the program are automatically generated as 01 and 02. If it is desired to have the tape bootstrap routine load Floating Tape Loader-Monitor C, an Update and Select run is needed to position Floating Tape Loader-Monitor C on a new production binary run tape. For the Update and Select run, the program name must be changed to "AAAMON" and a unique visibility must be assigned to the program.



EASYCODER
CODING FORM

PROBLEM ______ PROGRAMMER ______ DATE ___ . ______ PAGE ___ OF ___

	CARD NUMBER	TYPE	MARK	LOCATION	OPERATION CODE	OPERANDS	
	1 2 3 4 5	6	7	8 14	15 20	21 62	63 80
1	INS				PROG	pppppp (pppppp is a program name chosen by the user)	
2		L			@FTLMC	org,media,admode,tua,bua,cua, (Macro Instruction)	
3					END	START	

The following parameters of the macro instruction indicate the various options chosen to be present in the Loader-Monitor:

1. org - specifies the relocation bank indicator designating where the Loader-Monitor will reside. A 0 or blank for this entry implies that the relocation bank indicator is to be entered at execution time. Table 5-1 lists the acceptable relocation bank indicators, the indicated memory bank in which the Loader-Monitor will reside, and the last address used by the Loader-Monitor.

2. media - indicates the program unit media. A "T" or blank designates that program units are only to be loaded from binary run tape; "B" specifies both card (BRD) and tape (BRT) unit loading.

3. admode - signifies the permissible address mode entries, 3 or 4. If the machine has a memory size greater than 32K, this parameter must be 4.

4. tua - designates the control unit address for the console typewriter, if available. If a typewriter is used, the peripheral control unit address is specified as two octal characters, as it would appear in the second control character of a PDT instruction. The Loader-Monitor will always use read/write channel 2 (RWC2) for all "type" instructions.

Table 5-1. Relocation Bank Indicators

Indicator	Bank Indicated	Last Address Used by Loader-Monitor (octal)
02	12K	027777
03	16K	037777
04	20K	047777
05	24K	057777
06	28K	067777
07	32K	077777
11	40K	117777
13	49K	137777
15	57K	157777
17	65K	177777
23	81K	237777
27	98K	277777
33	114K	337777
37	131K	377777
47	163K	477777
57	196K	577777
67	229K	677777
77	262K	777777

5. bua - specifies the control unit address of the binary run program tape. If this parameter is blank or omitted, the BRT is assumed to be mounted on a drive connected to peripheral control unit 0. If the BRT is on any drive connected to a control unit other than 0, enter two octal characters to designate the unit.

6. cua - indicates the control unit address of the card reader. If this parameter is blank or omitted, the card reader is assumed to be connected to the peripheral control unit addressed as 41 (octal). Should the card reader be connected to some other peripheral control unit, enter two octal characters to designate the address.

PRESPECIALIZED VERSIONS OF FLOATING TAPE LOADER-MONITOR C

The two versions given below were specialized by Library Processor C according to the macro parameters designated, and they may be identified by their appropriate program and segment name; the segment names are 01 and 02. For additional information concerning the operation of these two versions, refer to page 2-2.

Program Name	org	media	admode	tua	bua	cua
AAAT3H	relocatable at execution	tape only	3	no	pcu 00	pcu 41
AAAT3T	"	"	3	yes on pcu 07	pcu 00	pcu 41

SECTION VI

SUMMARY OF PROGRAMMING CONSIDERATIONS

General programming considerations and restrictions associated with the Floating Tape Loader-Monitor C are outlined below.

1. The Loader-Monitor is restricted to operating below 262K, since some addresses in the communication area are limited to three characters (18 bits).

2. The assignment of the Loader-Monitor starting address (either at assembly or at execution time) must be consistent with the admode in which the Loader-Monitor is assembled. A Loader-Monitor assigned to an address above 32K must be assembled in address mode four; but an address assigned below 32K may be assembled in either three or four-character address mode.

3. Since Floating Tape Loader-Monitor C resides in upper memory and has a variable starting location, the programmer must ensure that no other program overlaps the Loader-Monitor area. In particular, programs which operate with a variable amount of memory locations must consider the address stored in locations 187 through 189 of the communication area before computing the memory available for their program.

4. When using the special call for compatibility with the Basic Programming System (page 2-4), the return to the Loader-Monitor must be made in the same addressing mode in which the called unit was started.

5. If the Loader-Monitor has been assembled in three-character addressing mode, it can only load programs up to 32K and will always start programs in address mode three.

6. If a 220-3 console typewriter is used for control messages, the interrupt-allow function must always be turned "off" before entering the Loader-Monitor (see page 1-2).

7. Floating Tape Loader-Monitor C uses index registers X5 and X6 while loading a program or segment thereof. Therefore, it is imperative that the user's program does not disturb the contents of X5 and X6 during the loading process.

   NOTE: The most likely cause of difficulties is the use of a CLEAR, DCW, DC or ORG statement in conjunction with the locations occupied by X5 and X6.

8. If assembled in four-character addressing mode, index registers X5 and X6 are used as four-character index registers, and a work mark is set on the high-order locations (21 and 25, respectively).

SECTION VII

OPERATING PROCEDURES FOR LOADING A PROGRAM UNIT

GENERAL

Locating and loading a program unit is accomplished initially in three functional operations:

1. Bootstrapping the BRT header label record and bootstrapping the tape bootstrap routine;
2. Locating and loading the Floating Tape Loader-Monitor C program; and
3. Calling the first program unit (object program or segment thereof).

The first two functional operations listed above are accomplished by performing the steps outlined in the paragraph below entitled "Bootstrap Procedure," and the third is accomplished by one of the methods outlined in "Console Call Procedure." There are two normal halts which are always encountered in conjunction with the above operations; they are designated as halts 1 and 3 in Table 7-1. In addition, there is an optional coded halt (halt number 6) that may be selected when it is desired to halt after loading a specified program unit.

Page 6-1 lists some general programming and operating considerations and restrictions to be accounted for when using Floating Tape Loader-Monitor C.

HALTS

Types of Halts

There are three categories of halts:

1. Programmed halts as described in the following paragraph;
2. Peripheral stalls; and
3. Machine malfunctions.

Programmed Halts

The BRT tape bootstrap routine and the Floating Tape Loader-Monitor C program have two normal halts which are always encountered in the process of loading a unit from a BRT into memory: halt number 1 and halt number 3. Halt number 1 provides the opportunity to enter a non-standard visibility (any visibility other than visibility A) to direct the bootstrap routine to search for a Loader-Monitor according to its unique visibility; and halt number 3 indicates that the Loader-Monitor is ready to accept the entry of the required parameter values into the communication area. One additional normal halt may be selected (halt number 6) by entering the program and segment names of a specific program unit into the halt name parameter locations in the Loader-Monitor communication area: the halt occurs after loading the specified unit.

The remaining programmed (coded) halts listed in Table 7-1 are operating and error halts, for which the applicable coding and appropriate operator action is indicated. Halt points in the following procedures are referred to by the designated halt number in Table 7-1.

All programmed halts are of the form H/A, B and are coded in the B-address register (with supplemental information for halt numbers 1 and 3 coded in the A-address register) to identify the halt condition. Thus, when the central processor halts, the reason may be determined by displaying the A- and B-address registers; if a typewriter is available and specified, the condition is indicated in a typed message.

PRELIMINARY PROCEDURES

The preliminary or initial setup procedures include mounting the BRT on an available tape drive which is assigned the logical address of zero (tape 0), activating the card reader if the Console Call card is to be used, and activating and loading the additional input/output devices to be used in the execution of the program run. Refer to the applicable sections of the Equipment Operators' Manual (Model 200) for the appropriate procedures to activate and load the required peripheral devices and prepare them to receive read and write instructions.

BOOTSTRAP PROCEDURE

Prior to loading a unit of a user's object program, the tape header label record, the tape bootstrap routine, and the Loader-Monitor program are read into memory by performing the following steps.

1. Initialize the central processor and activate the peripheral devices to be used.
2. Check that the BRT is mounted on drive 0, that tape 0 is rewound, and that drive 0 is in PROTECT status.
3. Set the CONTENTS buttons to 1nnnnn, where nnnnn is the peripheral control unit address of the BRT, as specified in parameter 5 of the macro instruction (see page 5-2).
4. Depress BOOTSTRAP. (This causes the tape label record to be bypassed.)
5. Set the CONTENTS buttons to 1nnnnn, as in step 3.
6. Depress BOOTSTRAP. (This causes the first tape bootstrap record to be read into location 0.)
7. Proceed to step 8 if the peripheral control unit address in step 5 was 0. Otherwise, set octal locations 65, 73, and 101 to 1nnnnn, where nnnnn is the correct address. If the user has specialized the Loader-Monitor, the address entered for macro parameter "tua" should be used as 1nnnnn.
8. Depress RUN. (Loader-Monitor halt number 1 will occur for entry of the visibility mask if different from octal 40 00 00 00 00 00, visibility A.)
9. Depress RUN. (Halt number 2A will occur only if the Loader-Monitor was specialized to relocate at execution time; if so, enter the appropriate relocation bank indicator, as given on page 5-2, into location 0 and press RUN.)
10. Proceed to "Console Call Procedure" when halt number 3 occurs.

OBTAINING HALT NUMBER THREE

Halt number three is reached automatically by performing the bootstrap procedures above. Halt number three can also be reached at any time through fixed-start 0 by performing the following steps:

1. Enter octal 126 (the address of fixed-start 0) into the instruction-address register.
2. Set ADDRESS MODE to 3.
3. Ensure that the sector bits (high-order bits 16 through 18) in the A-address register are zero.
4. Depress RUN.

When halt number 3 occurs, the Loader-Monitor parameters have been reset to search forward, by program and segment name, and execute a normal start. The search parameters may be entered into the communication area through the console, control panel, or the card reader.

CONSOLE CALL PROCEDURE

Card Entry of Parameter Values

1. Punch required parameter values in appropriate columns of Console Call card.
2. Place Console Call card in card reader input hopper and depress card reader START button. [1]
3. Check that halt number 3 has been obtained and that card entry is designated by the presence of 0 in octal location 100.
4. Depress RUN.
5. When a halt occurs, perform the required steps to fulfill programmer's requests and/or instructions. Take the proper steps to preserve input and output programs and printed listings as applicable. Refer to Table 7-1, page 7-5 for halt messages.

Manual Entry of Parameter Values

1. Designate manual entry by entering octal 01 into octal location 100.
2. Enter program name of called (requested) program unit into octal locations 104 through 111.
3. Enter segment name of called unit into octal locations 112 and 113.
4. Enter tape-drive number of tape drive containing the called unit into octal location 114.
5. Enter halt name into octal locations 115 through 124, if requested.
6. Depress RUN.

[1] At least two additional cards must be placed behind the Console Call card when using a Type 227 Card Reader.

7. Take same action as outlined in step 5 of "Card Entry of Parameter Values" above.

For more explicit operating procedures for parameter entry via the console typewriter, refer to Table 7-1 on page 7-5.

HALT PROCEDURES

If a console typewriter is not available to the user, control and halt messages can be transferred through a control panel. The term "halt", as used here, shall indicate either a machine halt or a program loop caused by a "busy" test following a typed message. The possible halt configurations for Floating Tape Loader-Monitor C are identical to those given for Tape Loader-Monitor C(3) and Tape Loader-Monitor C(4) with the addition of four halts: 2A, 2B, 2C, and 8 (see Table 7-1). Halt number 2A requests the relocation bank indicator when relocation has been designated during specialization. Whenever the Loader-Monitor halts, the operator should display the B-address register to determine the reason for the halt and display the A-address register for additional information concerning the halt.

When a console typewriter is available (Type 220-1, -2, or -3) and is designated during the specialization process, the halts as they would appear on the control panel would be replaced by typewriter messages. There are two categories of typewriter messages: informative and responsive. Informative messages indicate an erroneous or predetermined condition. The informative message causes the Loader-Monitor to loop until the operator takes the indicated action and enters a "G" into memory. Halts 2B, 2C, and 3 through 9 are informative messages. A responsive message indicates a request for a parameter entry by the Loader-Monitor. When a responsive message occurs, a "busy" test condition occurs until the operator enters the appropriate information. Upon receiving the information, the Loader-Monitor again loops until the operator confirms the entry by entering a "G." A "T" entry at confirmation time will allow the operator to re-enter the necessary parameter by causing the TYPE light to illuminate and the Loader-Monitor to loop. Responsive messages would be associated with halts numbered 1, 2, 2A and 3.

NOTE: In Table 7-1, the configuration "cu" refer to the address of the peripheral control unit, and "d" refers to the device number attached to control unit.

Table 7-1. Coded Halts for Tape Bootstrap Routine and the Floating Tape Loader-Monitor C Program

Halt No.	Cause	B Address	Operator Action at the Control Panel	Console Typeout	Operator Action at the Console Typewriter
1	Bootstrap routine requests loader visibility if non-standard.	017001	If a Loader-Monitor of non-standard visibility (not visibility A) is desired, enter the six-character visibility code, starting at the address specified in the A-address register, and press RUN. If standard visibility is to be used, press RUN.	N/A	N/A
2	Loader of requested visibility missing.	014002	Check mounted BRT and requested visibility. Correct error, rewind BRT, and start again from beginning of bootstrap procedure.	N/A	N/A
2A	Segment 1 of relocatable loader requests relocation bank indicator.	017070	Enter the relocation bank indicator at the address contained in the A-address register.	■RELOCATION INDICATOR□	Type the bank indicator into memory as two octal characters. When the TYPE light illuminates, type "G" for the loader to accept the indicator; a "T" entry will cause the original message to be typed so that the bank indicator can be re-entered.
2B	Uncorrectable read error on BRT in segment 1.	01culd	Examine and clean tape. Repeat bootstrap procedure (see page 7-2). cu = peripheral control unit address. d = device number.	■RD ER cu d□	Examine and clean tape. Repeat bootstrap procedure.
2C	Segment 2 of the Loader-Monitor was not found.	014012	Depress RUN to search again from beginning of tape.	■SEG 2 NOT FOUND□	Type a "G" to cause segment 1 to rewind the BRT and search for segment 2 again.
3	Loader-Monitor requests a console call.	017002	For card call, place the Console Call card in the card reader input hopper and press RUN. For manual call, enter the following information beginning at location 104_8: 1. "pppppps s" for program and segment name; 2. "d" for drive number from which program is to be loaded; 3. "pppppps s" for halt name (if used).	■NEXT PROG□	1. If this call is to come from the same device as the previous call, type "G" and go to step 5 or 6. NOTE: To continue a foreground program without a background program, perform the appropriate action in steps 2 through 4 (see page 8-9). 2. Type a "T" to change device. 3. When TYPE light illuminates, type console call indicator - 0 for card reader or 1 for manual entry; or type asterisk (*) to continue foreground program.
			Complete entry and press RUN. NOTE: If call device is to be changed, enter at the address specified in the A-address register either a 0 for a card call or a 1 for a manual call and proceed to execute the call. For continuing a foreground program without a background program, enter an asterisk (54_8) into location 100_8 and press RUN (see page 8-9).	■NEXT PROG NAME□	4. TYPE light will illuminate again. Type "G" for Loader-Monitor to accept preceding entry. A "T" entry will cause the Loader-Monitor to return to step 3. 5. If a Console Call card is used, place the card in the card reader input hopper and press RUN. 6. If manual entry is used, the second message associated with halt will be typed asking the program and segment name of the unit to be loaded. 7. When the TYPE light illuminates, type the following information: a. "ppppppss" for the program and segment name; b. "d" for the drive number from which the program is to be loaded. 8. When the TYPE light illuminates, type "G" for the loader to accept the preceding entry. A "T" entry will cause the loader to return to step 7.
				■HALT NAME□	9. The third message will be typed asking if there is a halt name. The TYPE light will illuminate. If there is no halt name, type "G" to start the program. If there is a halt name to be entered, type "T"; then perform steps 10 and 11.

Table 7-1 (cont). Coded Halts for Tape Bootstrap Routine and the Floating Tape Loader-Monitor C program

Halt No.	Cause	B Address	Operator Action at the Control Panel	Console Typeout	Operator Action at the Console Typewriter
3	Loader-Monitor requests a console call. (cont).				10. When the TYPE light illuminates, enter the halt name as "ppppppss" for the program and segment name. 11. When TYPE light illuminates, enter "G" for the loader to accept the halt name entry and start the program. A "T" entry causes the loader to step 10.
4	Uncorrectable read error has occurred on the BRT during segment 2.	00culd	Examine and clean tape, repeat last console call.	■RD ER cu d□	Type "G" to have the Loader-Monitor reread the record 64 times. Otherwise, examine and clean tape, and repeat last console call.
5	Illegal punch on Console Call card.	010110	Correct the card, refeed it, and press RUN.	■RD ER cu 0□	Correct card and refeed it. Type "G" to prepare the Loader-Monitor for parameter entry.
6	"Halt Name" unit has been loaded.	014000	Perform action requested for the specific object program and press RUN.	■pppppss LOADED□	If no entries into memory are requested, type "G" to start the program. Otherwise, type "T" to cause a halt (B-address register contains 014000) so that actions can be performed as requested by the object program. Press RUN when requested action is completed. When the TYPE light illuminates, enter "G" to start the program.
7	Called unit not found on backward search.	014010	Press RUN to search forward.	■PROG pppppss NOT FOUND□	Type "G" to have the Loader-Monitor search forward.
8	The Called unit was not found before the end card was read.	01cu30	To continue the search, place a binary run deck in the card reader input hopper and press RUN.	■ENDΔΔΔ c u Δ 0□	To continue search, place a binary run deck in the card reader input hopper and type "G". A "T" entry causes the Loader-Monitor to loop waiting for a "G" entry.
9	The Called unit was not found by searching in the specified direction.	014012	When loading from tape: Press RUN to search in the opposite direction. When loading from cards: Place the binary run deck in the card reader input hopper and press RUN.	■SEG ss NOT FOUND□	When loading from tape: Type "G" to search in the opposite direction. When loading from cards: Place the binary run deck in the card reader input hopper and type "G". If in either of the above cases a "T" is typed in rather than a "G", the Loader-Monitor will loop waiting for a "G" entry.

SECTION VIII

INTERRUPT CONTROL D

INTRODUCTION

The Interrupt Control D program operates in the Mod 1 Operating System and controls the simultaneous processing of background and foreground programs. A background program performs extensive data manipulation and processing, whereas a foreground program executes many terminal input/output operations (e.g., transferring data from magnetic tape to a printer). Interrupt Control D allows dual program processing by efficiently governing the distribution of central processor memory cycles between the two programs. Both background and foreground activities may consist of several programs executed serially. One type of program is executed independently of the other. Only one foreground program can be executed concurrently with one background program.

During interrupt processing, a background program is run in normal mode while a foreground program is always executed in interrupt mode. When a peripheral data transfer (PDT) instruction is initiated in the foreground program, control will go to the background program in normal mode until the foreground program's data transfer ends. At that time, an interrupt signal will be issued by the peripheral control unit involved and control will revert to the foreground program in interrupt mode. Control will alternate between the foreground and background programs according to the number of peripheral data transfers occurring in the foreground program.

The foreground program uses relatively few memory cycles (two microseconds each) to initiate a PDT instruction but requires several milliseconds to complete the data transfer for most peripheral devices. Therefore, Interrupt Control D has been programmed to shift control to the background program in proper sequence upon initiation of a foreground program PDT, thereby allowing the background program to use the available memory cycle time and enabling both programs to run concurrently. Upon completion of the data transfer, control reverts to the foreground program in proper sequence. The proper sequence is maintained for both the foreground and background program because Interrupt Control D saves the necessary index and sequence registers before transferring control.

The Floating Tape Loader-Monitor C program must be resident in memory together with Interrupt Control D in order to achieve simultaneous processing; and, like the Loader-Monitor, Interrupt Control D is a generalized macro program that requires specialization by Library

Processor C or D before being assembled by Easycoder Assembler C or D. Pages 8-4 and 8-5 explain the parameters necessary for the macro call for specialization, as well as the six prespecialized routines available for the most common applications.

EQUIPMENT REQUIREMENTS

Interrupt Control D will operate with all Series 200 central processors having the equipment configuration listed below:

1. A central processor having a minimum of 16K character locations of memory;
2. At least one terminal control unit equipped with the peripheral interrupt feature;
3. 500 to 625 memory locations starting at location 200 (decimal); this area may be used immediately after the Interrupt Control D program is loaded; and
4. 1180 to 2580 locations immediately below the Floating Tape Loader-Monitor C program.

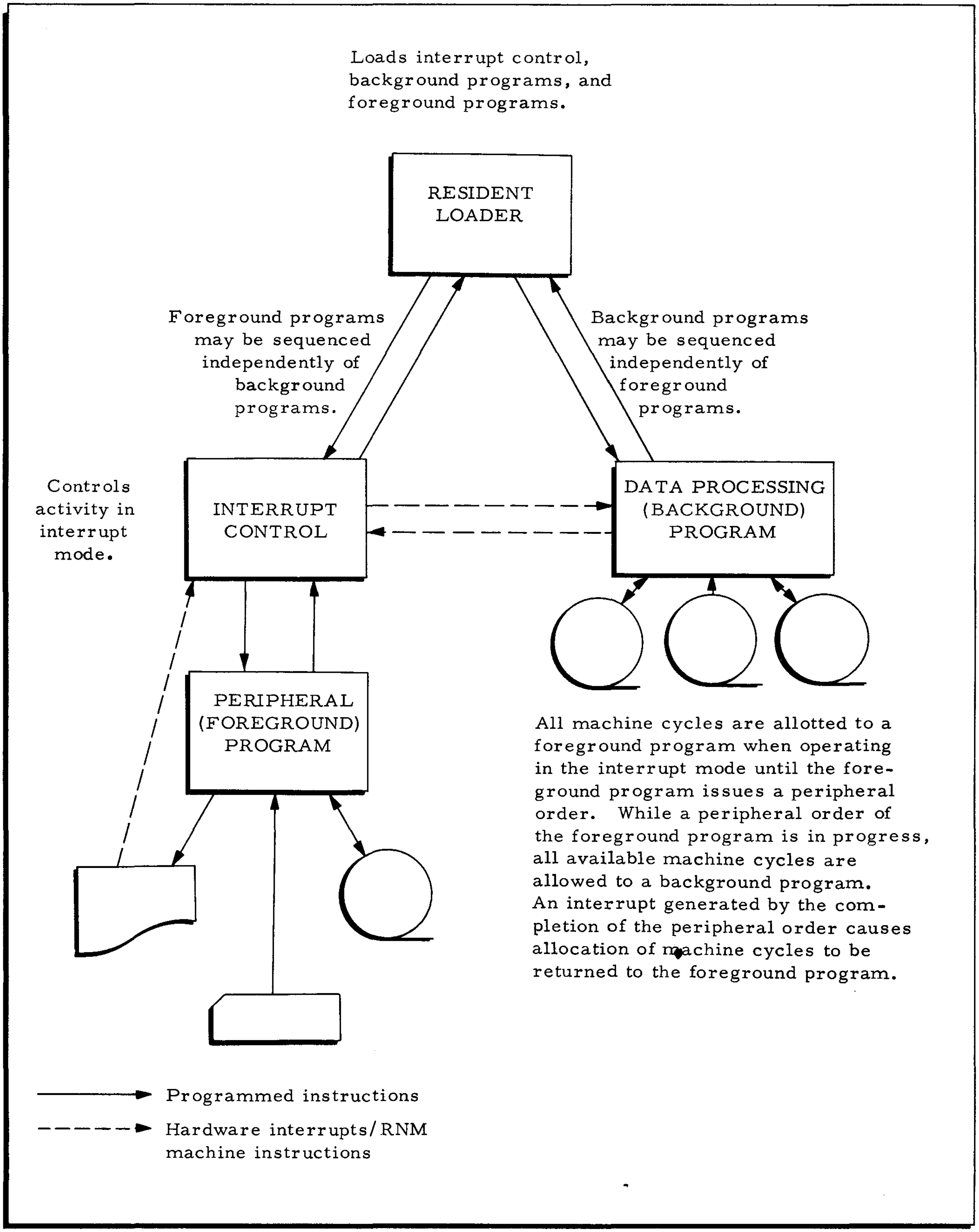
However, to operate with the 201-0 central processor, the 012 Program Interrupt Feature is also required.

NOTE: In items 3 and 4 above, the number of locations varies according to the version of Interrupt Control D that is used.

INTERRUPT PROCESSING WITH INTERRUPT CONTROL D

Basically, interrupt processing with Interrupt Control D consists of saving and restoring the necessary register settings of the foreground and background programs after a data transfer begins and when a data transfer ends. Figure 8-1 illustrates a general example of interrupt processing with the printer as the terminal interrupt device.

As an example of interrupt processing with Interrupt Control D, let us assume that the background and foreground programs have been loaded, interrupts have occurred, a tape-to-printer foreground program (in interrupt mode) is currently sequencing instructions in the sequence register, and the external interrupt register (EIR) contains the current sequence setting of the background program. A PDT instruction is now encountered in the foreground program. The PDT is followed by a macro routine specialized by the user to revert control to the Interrupt Control D program at an entrance point where a series of instructions will first store both the contents of the sequence register and the contents of any index registers used by the foreground program. The contents of any registers stored for the background program during the previous interrupts are now restored, while the EIR still contains the current setting of the background program. Finally, a resume normal mode (RNM) instruction is encountered, causing the contents of the sequence register and the EIR to exchange. Control is transferred to the background program at the proper sequence in normal mode.


Loads interrupt control, background programs, and foreground programs.
RESIDENT LOADER
Foreground programs may be sequenced independently of background programs.
Background programs may be sequenced independently of foreground programs.
Controls activity in interrupt mode.
INTERRUPT CONTROL
DATA PROCESSING (BACKGROUND) PROGRAM
PERIPHERAL (FOREGROUND) PROGRAM
Programmed instructions
Hardware interrupts/RNM machine instructions

All machine cycles are allotted to a foreground program when operating in the interrupt mode until the foreground program issues a peripheral order. While a peripheral order of the foreground program is in progress, all available machine cycles are allowed to a background program. An interrupt generated by the completion of the peripheral order causes allocation of machine cycles to be returned to the foreground program.

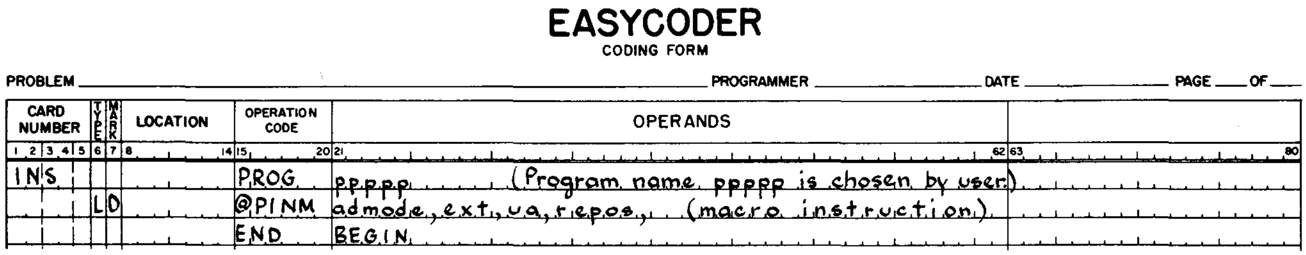
Figure 8-1. Interrupt Processing with Interrupt Control D

When the peripheral data transfer ends, an interrupt signal is sent from the printer to the peripheral control unit involved in order to place the central processor in interrupt mode, which causes the contents of the sequence register and the EIR to interchange. Control is then transferred to Interrupt Control D at a second entrance point where instructions are provided to store the contents of the registers used by the background program and restore the contents of the registers that were stored for the foreground program at the initiation of the PDT. A branch is then made into the foreground program at the address stored from the sequence register. That address references the instruction following the macro routine that followed the PDT, and control is now returned to the foreground program at the proper sequence in interrupt mode.

The alternating of control between the foreground program and background programs is a function of the peripheral data transfers issued in the foreground program.

SPECIALIZING INTERRUPT CONTROL D

If the user does not wish to utilize one of the prespecialized Interrupt Control D routines given in Table 8-1, he must specialize the generalized routine by entering the three instructions given below as input to a Library Processor C or D run.



EASYCODER
CODING FORM

PROBLEM ______ PROGRAMMER ______ DATE ______ PAGE ___ OF ___

	CARD NUMBER	TYPE	MARK	LOCATION	OPERATION CODE	OPERANDS	
	1 2 3 4 5	6	7	8 14	15 20	21 62	63 80
1	INS				PROG	ppppp (Program name ppppp is chosen by user)	
2		L	D		@PINM	admode, ext, ua, repos, (macro instruction)	
3					END	BEGIN	

The parameters of the macro instruction may be specified as follows.

1. admode - designates the address mode in which Interrupt Control D is to be assembled. This field must be coded as in an assembly ADMODE statement, and permissible values are 3 and 4. If Floating Tape Loader-Monitor C is located above 32K, this value must be 4.

2. ext - indicates the presence or absence of an external interrupt button. If the console or control panel has an external interrupt button, place B in this field; otherwise, leave this field blank.

3. ua - specifies the octal address of the peripheral control unit for the console typewriter containing the external interrupt button. This field must be coded identical to the second control character of a PDT instruction.

   NOTE: This parameter must be set only when a Type 201 or 201-1 Central Processor with a console will be used for processing.

4. repos - implies one of two conditions. If this field is blank, repositioning of the BRT (or background BRT if separate foreground and background BRT's are used) is not required. This situation exists if the background programs do not load by relative position or if the foreground and background programs are on separate BRT's. If this field contains the letter R, BRT repositioning is desired. This situation exists if any of the background programs load by relative position and are on the same BRT as the current foreground program.

Prespecialized Interrupt Control D Programs

Table 8-1 lists six prespecialized programs of Interrupt Control D. The six versions have been specialized under the names given; the segment names for each program are 01 and 02. For loading purposes, 01 is the segment call name; segment 02 is loaded automatically by segment 01.

Table 8-1. Prespecialized Versions of Interrupt Control D

Program Name	Address Mode	Presence of External Interrupt Button	Extent of Specialization	Memory Locations Required
AAAI3Δ	3	N/A	Assembled	500 locations starting at location 200 (decimal), and 1480 locations available below the Loader-Monitor.
AAAI3C	3	On console or control panel (201-2 Central processor or larger)	Specialized	500 locations starting at location 200 (decimal), and 1180 locations available below the Loader-Monitor.
AAAI3T	3	On console (201 or 201-1 Central Processor)	Specialized with 07 as the console control unit address	500 locations starting at location 200 (decimal), and 1300 locations available below the Loader-Monitor.
AAAI4Δ	4	N/A	Assembled	625 locations starting at location 200 (decimal) and 1880 locations available below the Loader-Monitor.
AAAI4C	4	On console or control panel (201-2 Central processor or larger)	Specialized	625 locations starting at location 200 (decimal), and 1600 locations available below the Loader-Monitor.
AAAI4T	4	On console (201 or 201-1 Central Processor)	Specialized with 07 as the console control unit address	625 locations starting at location 200 (decimal), and 1450 locations available below the Loader-Monitor.

NOTE: Add 600 locations to the 3-character programs and 700 locations to the 4-character programs if BRT repositioning is desired; i.e., if parameter 4 of the @PINM macro instruction is R.

FOREGROUND PROGRAM INSTRUCTIONS

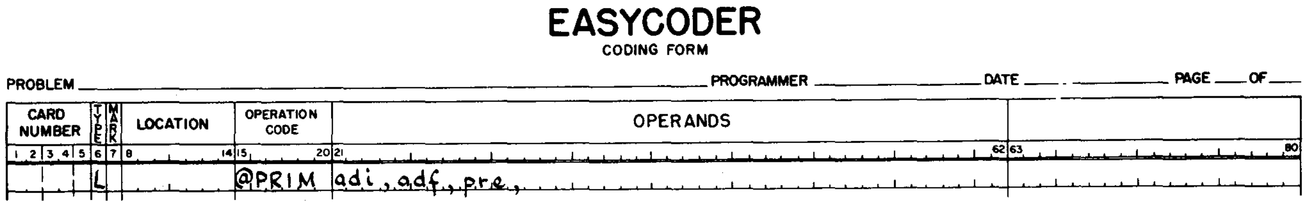
Any foreground program written by a user or supplied by Honeywell is essentially a generalized macro routine, for macro instructions are required after each PDT that is to initiate an interrupt in the foreground program. When specialized, these macro instructions provide the necessary coding to enter the Interrupt Control D program at the proper location.

Currently, Honeywell offers prespecialized foreground programs to perform tape-to-printer, tape-to-card, and card-to-tape data transfers, as explained in the manual Data Conversion A and C. If the user desires to write a foreground program, he must include the following macro instructions.

Foreground Return Macro

Communication with the foreground program is possible only through the Interrupt Control D program; the foreground program must never communicate directly with the Loader-Monitor during processing. Macro instructions are used to perform the communication, so the foreground program must be specialized before processing.

The macro instruction given below is necessary to return control to the Interrupt Control D program after the initiation of a PDT which will cause an interrupt in the foreground program. When the interrupt at the end of the peripheral data transfer occurs, Interrupt Control D returns control to the foreground program at the instruction following the macro routine.



EASYCODER
CODING FORM

PROBLEM ______ PROGRAMMER ______ DATE ______ PAGE ___ OF ___

CARD NUMBER	TYPE	MARK	LOCATION	OPERATION CODE	OPERANDS	
1 2 3 4 5	6	7	8 14	15 20	21 62	63 80
	L			@PRIM	adi, adf, pre,	

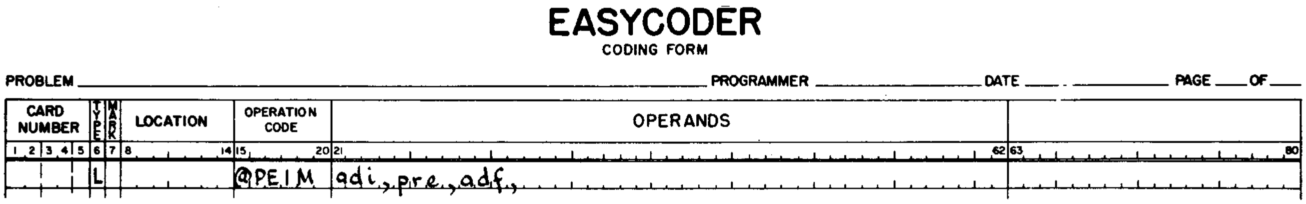
The parameters of the macro instruction may be designated as follows.

1. adi - indicates the address mode in which the call to the Interrupt Control D program is to be assembled. This parameter value must be the same as the "admode" parameter value given for Interrupt Control D (see page 8-4).
2. adf - specifies the address mode in which the foreground program is assembled at the point of the call to Interrupt Control D.
3. pre - designates the one, two, or three characters to be prefixed to tags in the macro routine to avoid duplication.

Foreground Exit Macro

The foreground exit macro is used to terminate the foreground program and to revert control to Interrupt Control D, which in turn assigns control to either the background program (if still operating) or the Loader-Monitor in order to load the next foreground or background program.

The macro instruction necessary for the exit is given below.



EASYCODER
CODING FORM

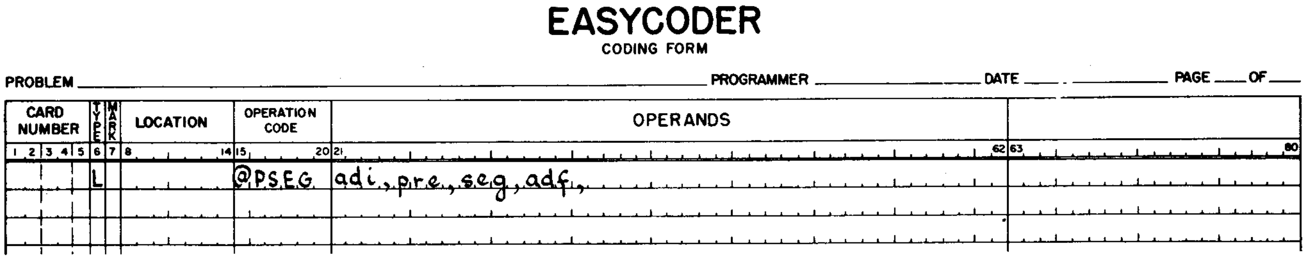
PROBLEM ______ PROGRAMMER ______ DATE ______ PAGE ___ OF ___

	CARD NUMBER	TYPE	MARK	LOCATION	OPERATION CODE	OPERANDS	
	1 2 3 4 5	6	7	8 14	15 20	21 62	63 80
1		L			@PEIM	adi,pre,adf,	

The parameters adi, pre, and adf for the exit macro are the same as for the return macro @PRIM. Their order is different, however, and must be maintained as indicated.

Next Foreground Segment Macro

If the user desires to write a multisegment foreground program, he must issue the foreground segment macro whenever he wants to load a foreground segment. This macro instruction returns control to Interrupt Control D and, eventually, to the Loader-Monitor.



EASYCODER
CODING FORM

PROBLEM ______ PROGRAMMER ______ DATE ______ PAGE ___ OF ___

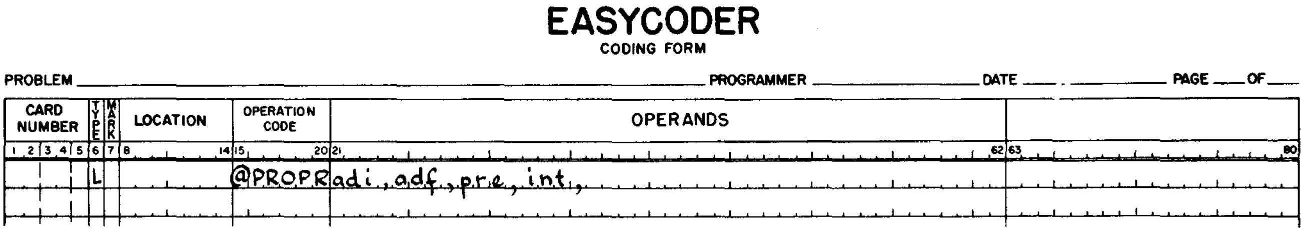
	CARD NUMBER	TYPE	MARK	LOCATION	OPERATION CODE	OPERANDS	
	1 2 3 4 5	6	7	8 14	15 20	21 62	63 80
1		L			@PSEG	adi,pre,seg,adf,	
2							
3							

The parameters of the macro instruction may be designated as follows.

1. adi - indicates the address mode in which the call to Interrupt Control D is to be assembled. This parameter value must be the same as the "admode" parameter value given for Interrupt Control D (see page 8-4).
2. pre - designates one, two, or three characters to be prefixed to tags in the macro routine to avoid duplication.
3. seg - is the 2-character name of the next foreground segment to be loaded.
4. adf - specifies the address mode in which the foreground program is assembled at the point of the call to Interrupt Control D.

Communications I/O C Return Macro Call

This macro instruction is issued only by Communications I/O C. The instruction is issued when an interrupt from a communication device has been processed, and additional communication processing in the normal mode is required.



EASYCODER
CODING FORM

PROBLEM ______ PROGRAMMER ______ DATE ______ PAGE ___ OF ___

CARD NUMBER	TYPE	MARK	LOCATION	OPERATION CODE	OPERANDS	
1 2 3 4 5	6	7	8 14	15 20	21 62	63 80
1	L			@PROPR	adi,adf,pre,int,	
2						

The first three parameters of the@PROPR macro instruction are the same as the first three parameters of the @PRIM macro.

4. int - is the address to which control is to be transferred when the next interrupt occurs. That is, if the user communication program desires to enter normal mode and wants the next interrupt to transfer control to TAGX, then TAGX must be entered as parameter 4. If this parameter is omitted, control will be transferred to the instruction following the last @PRIM executed.

Figure 8-2 illustrates the organization of a foreground program using the previously described macros with a 4-character version of Interrupt Control D.

Interrupt-Allow Instruction

If the user is writing his foreground program, he must include an instruction that will allow peripheral interrupts to occur. A peripheral control and branch (PCB) instruction with 71 (octal) for the third control character will set the interrupt-allow function for the peripheral control. After the last data transfer, both the interrupt and allow functions should be turned off to avoid causing an interrupt before the next foreground program is loaded. A PCB instruction with 70 (octal) as the third control character will turn off the allow function for the peripheral control unit, and a PCB with 74 (octal) for the third control character will turn off the interrupt function for the peripheral control unit.

PROGRAMMING CONSIDERATIONS

The following information summaries should be considered when programs are to operate in a multiprogram environment under Interrupt Control D.

1. The timing or speed of peripheral devices should not be assumed in the logic of a user's program.

2. The memory locations required by each program must be predetermined so that conflicts do not arise at execution time. If a background program uses additional memory at execution time, the user must ensure that the additional memory allocated does not overlap the areas reserved for Interrupt Control D and the foreground program. The base of the Interrupt Control D program is reflected in locations 273 through 275 (octal) of the Floating Tape Loader-Monitor communication area; but the user is responsible for noting the location of the foreground program.

3. When processing interrupts, Interrupt Control D places a record mark on location 0 and does not restore the original punctuation.

4. The foreground and background programs cannot share read/write channels, unless the background program does not rely on the settings of the read/write current and starting location counters that are shared.

5. Foreground programs must not use read/write channel 2 because it is required by Floating Tape Loader-Monitor C.

6. Programs must not load the external interrupt register.

7. Programs must not share the peripheral control unit equipped with the interrupt feature if any foreground program sets the interrupt-allow function for that control unit.

8. All foreground programs must set the interrupt-allow function for their peripheral device(s). Normally, only terminal devices should have the interrupt-allow function set.

9. Foreground programs must turn off both the interrupt and allow functions for their device(s) before the final exit is made to Interrupt Control D.

10. Foreground programs must do their own testing of device status, e.g., error tests.

11. Foreground programs must not communicate directly with Floating Tape Loader-Monitor C; all communication must be made through Interrupt Control D.

12. Foreground programs must not use index register 1 when processing data, because this register is used exclusively for communication between a foreground program and Interrupt Control D. Foreground programs may use index registers 2 through 6, because Interrupt Control D stores and restores the respective foreground and background registers.

13. Index registers must not contain record marks.

14. Programs that have been converted using Easytran Symbolic Translator D cannot be run as foreground programs under Interrupt Control D. However programs of a background nature that have been converted by Easytran Symbolic Translator D can be run as background programs, provided that the foreground program does not use or distrub the change sequence register (CSR).

OPERATING PROCEDURES

Interrupt Control D is loaded by methods applied to any program unit called by Floating Tape Loader-Monitor C, but it will always reside in locations immediately below the Loader-Monitor. The loading process consists of subtracting from the base address of the Loader-Monitor the number of locations required for the specialized Interrupt Control D program and using the result as the base address of Interrupt Control D. After loading Interrupt Control D, control immediately returns to the Loader-Monitor.

After Floating Tape Loader-Monitor C and Interrupt Control D have been loaded, the foreground and background programs are called and loaded according to the presence or absence of an external interrupt button on the console or control panel.

Operating with the External Interrupt Button

When the external interrupt button is present and the Interrupt Control D program is so specialized, all foreground programs will be loaded through use of the button. Whenever the interrupt button is depressed, an interrupt occurs and control goes to Interrupt Control D, which checks for the presence of a foreground program. If a foreground program is loaded and processing, the interrupt will be ignored. If a foreground program is not in memory, control will go to the Loader-Monitor, in interrupt mode, to prepare for a console call (halt number 3). Normally, the foreground program is then called using a Console Call card.

If the external interrupt button is depressed accidentally, the Loader-Monitor halt number 3 occurs (17002 in the B-address register). Recovery may be accomplished by setting the sequence register to 0 and depressing RUN.

The normal method for loading foreground and background programs when the interrupt button is available is to load the background program first and press the interrupt button when it is desired to load a foreground program. Interrupt Control D checks the Loader-Monitor to ascertain whether it is busy loading a background program; if so, Interrupt Control D will "loop" on the busy indicator until the background segment is loaded. When the background segment is loaded, Interrupt Control D gives control to the Loader-Monitor to call in the next segment, which should be a foreground segment.

Operating without the External Interrupt Button

When an external interrupt button is not present, the first foreground program must be loaded prior to the first background program. In this case, the foreground program starts sequencing instructions in normal mode until the first PDT instruction that will cause an interrupt has been completed. At the initiation of this PDT, control will be reverted to Interrupt Control D and then to the Loader-Monitor for loading of the first background segment. It is the user's responsibility to order the foreground and background program segments accordingly.

Background programs are loaded, started, and sequenced normally under control of Floating Tape Loader-Monitor C. To sequence foreground programs, however, an asterisk (octal 54) must be entered in octal location 267 of the Loader-Monitor communication area prior to the exiting of the current foreground program. When a foreground program exits, Interrupt Control D checks location 267 for the presence of an asterisk. If an asterisk is present, Interrupt Control D will transfer control to the Loader-Monitor to load another foreground segment. If an asterisk was not entered in octal location 267 before the foreground program exited, Interrupt Control D resumes normal mode to allow the background program to continue processing, and another foreground program cannot be loaded until the background program has completed processing.

Continuing a Foreground Program without a Background Program

Whether or not the external interrupt button is present, a foreground program cannot continue processing without a resident background program unless certain precautions are taken. If a background program has finished processing (the program's end-of-run halt has occurred) and the next background program is not ready to be loaded, or there are no more background programs to be run, the operator may enter an asterisk into octal location 100 of the Loader-Monitor communication area and press RUN to allow the foreground program to continue processing.

Location 100 (octal) in the Loader-Monitor communication area normally contains either 00 to signify a console call entry or 01 to designate a manual call. The Loader-Monitor will "loop" on a test of the asterisk in location 100, thereby allowing interrupts to occur. When a background program is ready to be loaded, the operator must enter either 00 or 01 into location 100 in order to load and start the background program.

Coded Halts for Interrupt Control D

Interrupt Control D performs some tape manipulation. Read errors may occur. The following halts or console typeouts will result.

Console —	40d	RD	ER	d is tape drive number
Halt —	00	40	1d	

If a version of Interrupt Control D which incorporates a console typewriter is being used, one of the following actions may be initiated.

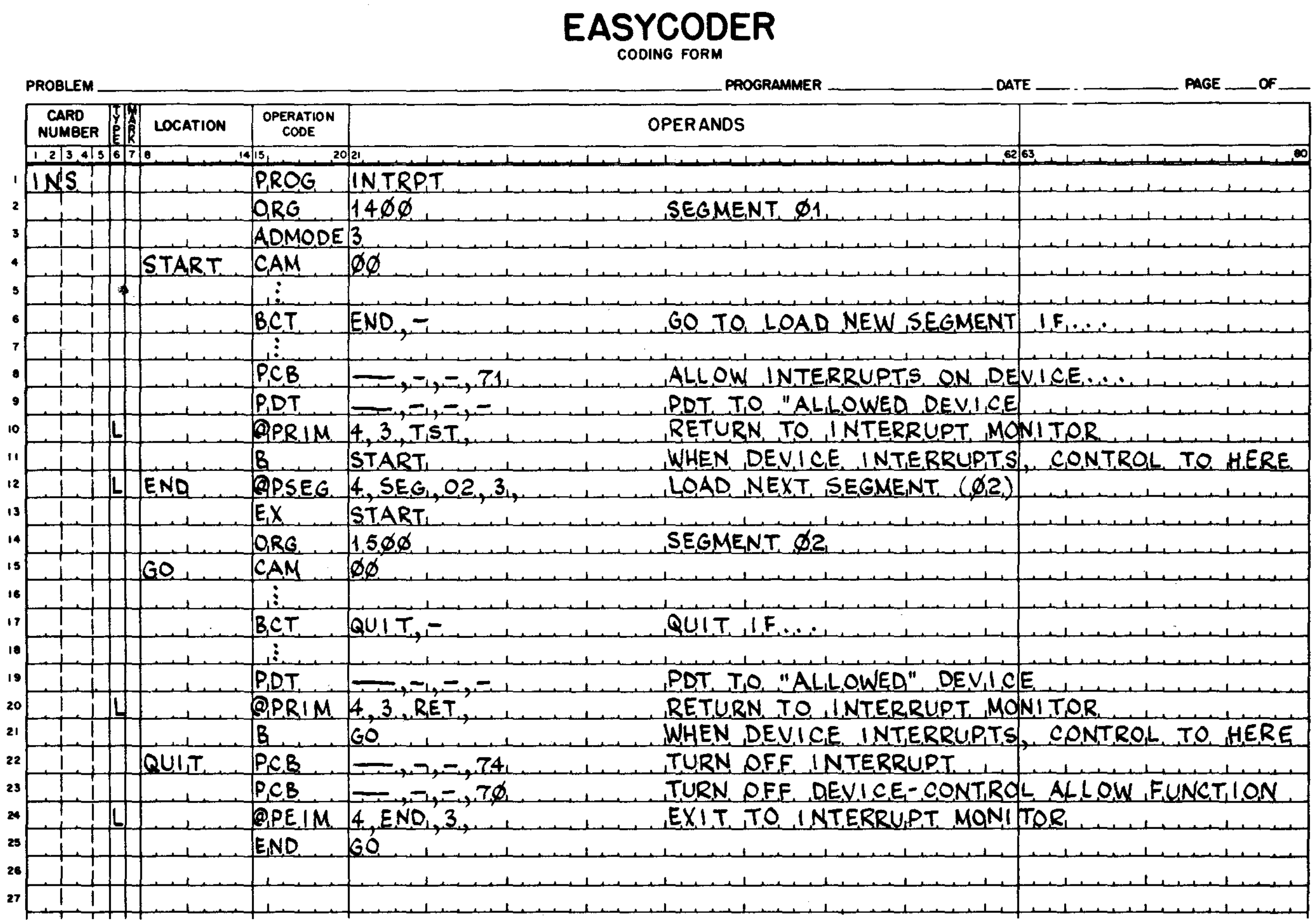
1. Type G. The error record will be read as many as 64 times if necessary, in an attempt to process the record.
2. Mount another tape having the current background program and type L. Control will be returned to the Loader-Monitor to read a Console Call

card, presumably, to reload the current background program. The foreground program can continue.

3. Type C. An asterisk will be entered into location 100 (octal). The foreground program can continue to completion..

If a version of Interrupt Control D is being processed on a machine with a control panel, the RUN button should be pressed. This will cause another halt immediately. The following alternate actions are possible.

1. The A-address register contains the address of the routine which is executed by typing L on the console version of Interrupt Control D (see 2., above). Set the sequence counter to the address contained in the A-address register and press RUN.
2. The B-address register contains the address of the routine which is executed by typing C on the console version of Interrupt Control D (see 3., above). Set the sequence counter to the address contained in the B-address register and press RUN.
3. Press RUN to read the error record as many as 64 more times, if necessary, in an attempt to process the record.



EASYCODER
CODING FORM

PROBLEM ____ PROGRAMMER ____ DATE ____ PAGE ___ OF ___

```
CARD  T M
NUMBER  LOCATION OPERATION OPERANDS
              CODE
INS           PROG  INTRPT
              ORG   14ØØ                    SEGMENT Ø1
              ADMODE3
       START  CAM   ØØ
                :
              BCT   END,-                   GO TO LOAD NEW SEGMENT IF...
                :
              PCB   —,-,-,71                ALLOW INTERRUPTS ON DEVICE...
              PDT   —,-,-,-                 PDT TO "ALLOWED DEVICE
     L        @PRIM 4,3,TST,                RETURN TO INTERRUPT MONITOR
              B     START                   WHEN DEVICE INTERRUPTS, CONTROL TO HERE
     L END    @PSEG 4,SEG,02,3,             LOAD NEXT SEGMENT (Ø2)
              EX    START
              ORG   15ØØ                    SEGMENT Ø2
       GO     CAM   ØØ
                :
              BCT   QUIT,-                  QUIT IF...
                :
              PDT   —,-,-,-                 PDT TO "ALLOWED" DEVICE
     L        @PRIM 4,3,RET,                RETURN TO INTERRUPT MONITOR
              B     GO                      WHEN DEVICE INTERRUPTS, CONTROL TO HERE
       QUIT   PCB   —,-,-,74                TURN OFF INTERRUPT
              PCB   —,-,-,7Ø                TURN OFF DEVICE-CONTROL ALLOW FUNCTION
     L        @PEIM 4,END,3,                EXIT TO INTERRUPT MONITOR
              END   GO
```

Figure 8-2. Use of Foreground Program Instructions to Communicate with Interrupt Control D

APPENDIX A

USE OF VISIBILITY

Since the only realistic test environment for new programs is operation with other systems programs, the possibility of a new program causing problems in other parts of the system increases as the number of individual programs grows. Malfunctions in unchecked programs can cause errors which affect other programs and cause a programmer to waste time searching for errors in one program, only to find that they originated in another.

Ideally, each programmer should have his own system tape for test purposes. Such a tape would consist of all the checked-out programs plus the new programs of his own. In this way, new programs can be tested without creating problems in any other part of the system also under test. Any improper output could be assumed to originate in the new program or programs of a particular system tape. In effect, the visibility code system provides for such separate tapes.

As an example, consider that in a system under development there are three checked-out programs. For convenience, these will be called P-ONE, P-TWO, and P-THREE. A fourth program, P-FOUR, is still under test; however, it gives proper output within known limits which can be used as input to a new program, P-FIVE. For a given computer run, it is desired to test a new version of P-FOUR which incorporates new coding (this version will be called P-FOUR-M) and also to test P-FIVE. Obviously, there is a possibility that P-FOUR-M can introduce new errors, making the output completely useless for any realistic test of P-FIVE. On the other hand, the coding patches may be correct, allowing P-FOUR-M to be the final version of the program that is to be included as part of the system. In this case, P-FIVE should be tested with P-FOUR-M. By use of the visibility codes, a binary run tape could be produced using the following visibility coding:

PROGRAM	VISIBILITY		
P-ONE	A	B	C
P-TWO	A	B	C
P-THREE	A	B	C
P-FOUR		B	
P-FOUR-M	A		C
P-FIVE		B	C

In this way, the three versions of the same system exist on the same tape. The first version called for could be version A. If version-A output were not satisfactory (i. e., P-FOUR-M did not produce the desired results), P-FIVE could still be tested by calling in version B which includes P- FOUR without the corrective coding included in version A. If version-A output did include the desired output, version C could be called for and executed. In either case, the testing of P-FIVE could proceed independently of the work being done on P-FOUR. Improper output from any version could be isolated to a specific program with a high degree of certainty.

Note that the visibility assignment given above was only one of the many ways in which the desired results could be obtained.

Table A-1. Visibility Codes

Visibility Code	Octal Visibility Key	Visibility Code	Octal Visibility Key
A	40 00 00 00 00 00	T	00 00 00 20 00 00
B	20 00 00 00 00 00	U	00 00 00 10 00 00
C	10 00 00 00 00 00	V	00 00 00 04 00 00
D	04 00 00 00 00 00	W	00 00 00 02 00 00
E	02 00 00 00 00 00	X	00 00 00 01 00 00
F	01 00 00 00 00 00	Y	00 00 00 00 40 00
G	00 40 00 00 00 00	Z	00 00 00 00 20 00
H	00 20 00 00 00 00	0	00 00 00 00 10 00
I	00 10 00 00 00 00	1	00 00 00 00 04 00
J	00 04 00 00 00 00	2	00 00 00 00 02 00
K	00 02 00 00 00 00	3	00 00 00 00 01 00
L	00 01 00 00 00 00	4	00 00 00 00 00 40
M	00 00 40 00 00 00	5	00 00 00 00 00 20
N	00 00 20 00 00 00	6	00 00 00 00 00 10
O	00 00 10 00 00 00	7	00 00 00 00 00 04
P	00 00 04 00 00 00	8	00 00 00 00 00 02
Q	00 00 02 00 00 00	9	00 00 00 00 00 01
R	00 00 01 00 00 00	*	00 00 00 00 00 00
S	00 00 00 40 00 00		

APPENDIX B
OWN-CODING

INTRODUCTION

The Tape Loader-Monitor program's own-coding provisions are intended primarily for the use of Honeywell systems programs. However, although the use of own-coding is not recommended for the average user, the own-coding provisions may be required by the user concerned with the development of programming systems.

OWN-CODE EXECUTION DURING LOADING

Functions Provided

The Loader-Monitor provides an own-coding exit after each record of a program unit is read into the input buffer, thereby enabling the execution of onw-code routines during loading. The own-code routine may return to the same point and use the distribution process of the Loader-Monitor to distribute the data portion of the program unit records, or it may bypass the distribution process of the Loader-Monitor and distribute the data portions of the unit records itself.

Own-Code Exit

A calling unit may execute own-coding during the loading of a called unit by setting up an appropriate Branch instruction in the Loader-Monitor communication area. Specifically, the starting address of the own-code routine must be entered into locations 103 to 105 (octal 147-151) of the communication area. The starting address of the own-code routine is then the A-address of the Branch instruction whose op code is stored in location 102. Locations 103 to 105 contain no punctuation, and the calling program must not place punctuation in these locations. The branch to the starting address of the own-code routine is made immediately after reading each record. However, before the branch is executed, the Loader-Monitor sets index register X5 to the address of the first data character in the record. The own-code routine must expect to be started in the three-character addressing mode, and the return to the Loader-Monitor must also be in the same mode. The own-code exit branch is reset after loading each unit.

Own-Code Return Points

The own-code routine must conclude with a branch to location 122 or 126 in the Loader-Monitor communication area.

When the return branch is to location 122, the Loader-Monitor performs distribution in the normal manner. Use of this option requires that the setting of index register X6 (the index register used by the Loader-Monitor for distribution) must not be altered by the own-code routine.

When the return branch is made to location 126, the Loader-Monitor bypasses the distribution process and reads the next record. Use of this option requires that the own-code routine must be able to identify the last record of the called unit. When this record is identified, the own-code routine must not return to location 126; instead, it must set index register X5 to the address of a location in which the control character 61 is stored, followed by the 3-character starting address of the unit just loaded. The own-code routine must then provide a branch to location 122 in the communication area.

The return branch to either location 122 or location 126 must be executed in 3-character address mode. If the user program is operating in 3-character mode above the first 32K of memory, the following sequence of instructions must be executed before returning to the Loader-Monitor communication area.

EASYCODER
CODING FORM

PROBLEM ____ PROGRAMMER ____ DATE ____ PAGE ___ OF ___

	CARD NUMBER (1-5)	TYPE (6)	MARK (7)	LOCATION (8-14)	OPERATION CODE (15-20)	OPERANDS (21-62)	(63-80)
1					ADMODE	4	
2					CAM	6Ø	
3					SW	126 (or 122)	
4					ADMODE	3	
5					CAM	ØØ	
6					B	126 (or 122)	
7					NOP		
8							

If the user program is operating in 4-character mode above the first 32K of memory, the following instructions are required to return to the communication area.

EASYCODER
CODING FORM

PROBLEM ____ PROGRAMMER ____ DATE ____ PAGE ___ OF ___

	CARD NUMBER (1-5)	TYPE (6)	MARK (7)	LOCATION (8-14)	OPERATION CODE (15-20)	OPERANDS (21-62)	(63-80)
1					SW	126 (or 122)	
2					ADMODE	3	
3					CAM	ØØ	
4					B	126 (or 122)	
5					NOP		
6							

APPENDIX C

ADDRESS ASSIGNMENTS FOR PERIPHERAL CONTROLS

A peripheral device may be addressed automatically by program instruction or manually from the operator's control panel. The address of a peripheral device is contained in the second control character (C2) of the Peripheral Data Transfer (PDT) and the Peripheral Control and Branch (PCB) instructions. A peripheral device may also be addressed manually by entering its octal address assignment into the CONTENTS buttons on the operators' control panel. Four bits of the second control character (the high-order bit and the low-order three bits) are used to provide the address assignment of a specific peripheral device. The recommended address assignments for the peripheral controls are listed in the following table.

Peripheral Control	Second Control Character (C2)						Octal Address
	B_6	B_5	B_4	B_3	B_2	B_1	
Magnetic Tape (output)	0	0	0	0	0	0	00
(input)	1	0	0	0	0	0	40
Card Reader or Paper Tape Reader	1	0	0	0	0	1	41
Card Punch or Paper Tape Punch	0	0	0	0	0	1	01
High-Speed Printer	0	0	0	0	1	0	02
On-Line Adapter	1	0	0	0	1	0	42
Console (output)	0	0	0	1	1	1	07
(input)	1	0	0	1	1	1	47

In the interest of uniformity among Series 200 installations, it is desirable to use the recommended address assignments for peripheral controls; however, the ability to assign any address to a peripheral control is maintained to permit changing of the recommended assignments on an installation basis when necessary.

If the Series 200 system includes tape controls for both half-inch and three-quarter inch magnetic tape units, the octal addresses 00 and 40 should be assigned to the half-inch tape control.

APPENDIX D

SERIES 200 CHARACTER CODES

Key Punch	Card Code	Central Processor Code	Octal	High Speed Printer	Key Punch	Card Code	Central Processor Code	Octal	High Speed Printer
0	0	000000	00	0		X, 0 or X[†]	100000	40	-
1	1	000001	01	1	J	X, 1	100001	41	J
2	2	000010	02	2	K	X, 2	100010	42	K
3	3	000011	03	3	L	X, 3	100011	43	L
4	4	000100	04	4	M	X, 4	100100	44	M
5	5	000101	05	5	N	X, 5	100101	45	N
6	6	000110	06	6	O	X, 6	100110	46	O
7	7	000111	07	7	P	X, 7	100111	47	P
8	8	001000	10	8	Q	X, 8	101000	50	Q
9	9	001001	11	9	R	X, 9	101001	51	R
	8, 2	001010	12	'		X, 8, 2	101010	52	#
#	8, 3	001011	13	=	$	X, 8, 3	101011	53	$
@	8, 4	001100	14	:	*	X, 8, 4	101100	54	*
Space	Blank	001101	15	Blank		X, 8, 5	101101	55	"
	8, 6	001110	16	>		X, 8, 6	101110	56	≠
	8, 7	001111	17	&	-	X or X, 0[†]	101111	57	!
&	R, 0 or R[†]	010000	20	+		8, 5	110000	60	<
A	R, 1	010001	21	A	/	0, 1	110001	61	/
B	R, 2	010010	22	B	S	0, 2	110010	62	S
C	R, 3	010011	23	C	T	0, 3	110011	63	T
D	R, 4	010100	24	D	U	0, 4	110100	64	U
E	R, 5	010101	25	E	V	0, 5	110101	65	V
F	R, 6	010110	26	F	W	0, 6	110110	66	W
G	R, 7	010111	27	G	X	0, 7	110111	67	X
H	R, 8	011000	30	H	Y	0, 8	111000	70	Y
I	R, 9	011001	31	I	Z	0, 9	111001	71	Z
	R, 8, 2	011010	32	;		0, 8, 2	111010	72	@
.	R, 8, 3	011011	33	.	,	0, 8, 3	111011	73	,
□	R, 8, 4	011100	34	)	%	0, 8, 4	111100	74	(
	R, 8, 5	011101	35	%		0, 8, 5	111101	75	CR
	R, 8, 6	011110	36	■		0, 8, 6	111110	76	□
	R or R, 0[†]	011111	37	?		0, 8, 7	111111	77	¢

[†] Special Code. This card code-central processor code equivalency is effective when control character 26 is coded in a card read or punch PCB instruction.

COMPUTER-GENERATED INDEX

HONEYWELL EDP TECHNICAL PUBLICATIONS
USERS' REMARKS FORM

TITLE: SERIES 200
FLOATING TAPE LOADER-MONITOR C
AND INTERRUPT CONTROL D
SOFTWARE MANUAL

DATED: APRIL, 1966
FILE NO: 123.5005.001C.1-005

ERRORS NOTED:

Fold

SUGGESTIONS FOR IMPROVEMENT:

Fold

FROM: NAME ______________________ DATE __________

COMPANY ______________________

TITLE ______________________

ADDRESS ______________________

____

Cut Along Line

FIRST CLASS
PERMIT NO. 39531
WELLESLEY HILLS
MASS.

Cut Line

BUSINESS REPLY MAIL

No postage stamp necessary if mailed in the United States

POSTAGE WILL BE PAID BY

HONEYWELL
ELECTRONIC DATA PROCESSING DIVISION
60 WALNUT STREET
WELLESLEY HILLS, MASS. 02181

ATT'N: TECHNICAL COMMUNICATIONS DEPARTMENT

Honeywell
ELECTRONIC DATA PROCESSING

HONEYWELL EDP
TECHNICAL PUBLICATIONS REMARKS FORM

TITLE: MOD 1 (TR) FLOATING TAPE
LOADER-MONITOR C AND INTERRUPT
CONTROL D ADDENDUM NO. 2
SOFTWARE BULLETIN

DATED: APRIL, 1967

FILE NO: 122.5005.001C.1-005

ERRORS NOTED IN PUBLICATION:

Fold

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION:

Fold

(Please Print)

FROM: NAME ____________________ DATE __________

COMPANY ____________________

TITLE ____________________

ADDRESS ____________________

____

ut Along Line

Cut Along Line

Honeywell
ELECTRONIC DATA PROCESSING