SOFTWARE MANUAL

SERIES 200

1/2-INCH TAPE I/O A

GENERAL SYSTEM: SERIES 200/BASIC PROGRAMMING SYSTEM

SUBJECT: 1/2-Inch Tape I/O A, the Input/Output Program for Processing Fixed-Length Records on 1/2-Inch Magnetic Tapes (Formerly TIPTOP1).

SPECIAL INSTRUCTIONS: This software manual completely supersedes the information bulletin entitled TIPTOP1, DSI-267C, dated May 7, 1965, and incorporates the updates and corrections contained in the software bulletin entitled Addendum #1 to TIPTOP1 dated October 5, 1965.

DATE: March 21, 1966

FILE NO. 123.4105.020A.0-2*93

8809
5366
Printed in U. S. A.

*When ordering this publication please specify Title and Underscored portion of File Number.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I
BASIC CONCEPTS

Coding those portions of a program which involve the reading and writing of magnetic tape records can be a lengthy and exacting task. Such operations as checking tape labels, checking file identification, blocking and unblocking records, checking for read/write errors, attempting to correct these errors, checking for end-of-reel conditions, etc., involve coding which requires programming time and skill out of proportion to actual processing time. To relieve the programmer of preparing this complex coding, Honeywell has designed several programs for the handling of tape input and output on the Series 200 computers.

These tested routines make it possible for the programmer to handle the input/output operations mentioned above with simple macro statements such as OPEN, CLOSE, GET, PUT, and FEOR. A macro statement will produce an entire series of machine instructions which causes a desired input/output operation to be performed. For example, to locate the next item in a tape file being read, the programmer includes a GET statement in the source program; to obtain the address of the next available output location for a processed item, he writes a PUT statement. The Tape I/O program produces all the machine-language coding necessary to schedule such operations, to identify possible errors, and, in general, to handle all the functions necessary to complete the operation. 1/2-Inch Tape I/O A, the version for the 4K Easycoder Assembler A using 1/2-inch tapes, is described in this manual. Other versions are described in separate publications. 1/2-Inch Tape I/O A comprises two main elements: the file-handling macro routines, labeled TI01, and the Translator program which specializes these routines for input to Easycoder assembly.

Reduced programming time is only one advantage of 1/2-Inch Tape I/O A. Since the package utilizes tested, efficient routines, programming errors are reduced. Record handling is standardized, providing the most efficient use of the user's system. Perhaps the most distinct advantage of the program is that it allows the programmer to concentrate on the handling of items (logical groups of sequential information within a tape record), rather than on the handling of the tape records themselves.

RECORD FORMS ACCEPTABLE TO 1/2-Inch Tape I/O A

A blocked record consists of more than one item; an unblocked record is one which contains only one item. Blocked records are read and written faster than unblocked records because the number of tape stop and start operations is significantly reduced. Also, the storage capacity of a tape is increased due to the reduction of interrecord gaps.

For input operations, 1/2-Inch Tape I/O A provides routines to unblock records which exist in storage in blocks, making them available for processing. The package also blocks records

during output operations. Thus, the programmer can maintain blocked records in a file yet process the constituent items individually, without having to provide the coding necessary to block and unblock such records.

1/2-Inch Tape I/O A accepts two forms of data records: fixed-length blocked records and fixed-length unblocked records. The basic formats of these records are illustrated in Figure 1-1.

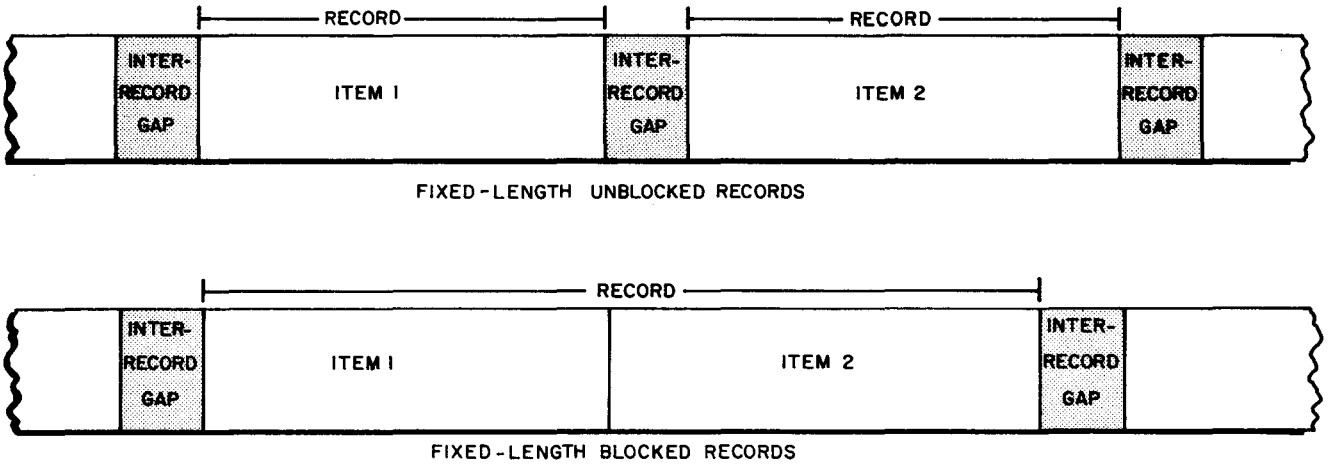


Figure 1-1. Blocked and Unblocked Records

AVAILABLE TAPE INPUT/OUTPUT ROUTINES

1/2-Inch Tape I/O A provides the programmer with routines which:

* "Open" a tape file. For input (tape read) operations, "opening" a tape file involves the performance of all the control operations (e.g., reading and checking the header label, checking the reel sequence number, etc.) prior to reading a file. Similarly, the control operations necessary to write a file (e.g., writing a new header label, checking for errors, etc.) are performed by for output operations.
* "Get" an item from a file being read. This operation provides the programmer with the next item in a file in such a way that he is not concerned with the actual manipulation of the tape.
* "Put" an item into a file being written. This allows the programmer to manipulate the next item to be written on a specified output tape without regard for tape handling.
* "Force" an end-of-reel condition. This macro routine terminates a tape reel before the physical end of reel is sensed.
* "Close" a tape file. All the control operations necessary to close a file are performed by this routine.

Figure 1-2 illustrates the basic use of the routines offered by 1/2-Inch Tape I/O A.

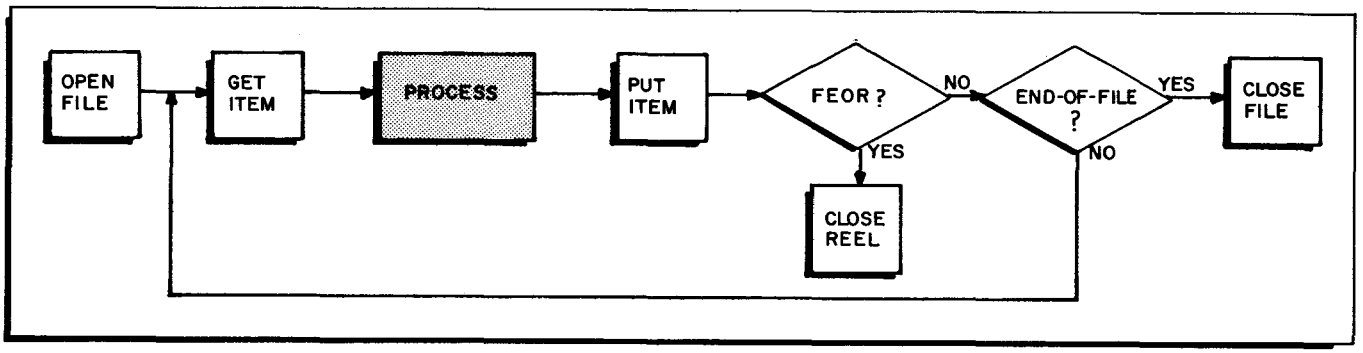


Figure 1-2. Basic 1/2-Inch Tape I/O A Routines

HOW TO USE 1/2-Inch Tape I/O A

The programmer must perform the following operations in each program:

1. Include one DCA (Define Communication Area) entry for each file used by the program. This entry describes the characteristics of the file and is inserted in the program as shown in Figure 1-3.

2. Include macro statements in the program. Five macro statements are available: OPEN, GET, PUT, CLOSE, and FEOR.

3. Reserve and punctuate the input/output area(s) to be used by 1/2-Inch Tape I/O A in handling the tape files. One or two input/output areas may be used for a file, as specified in the corresponding DCA entry. Each input/output area must be followed by a record mark in the next memory location.

   NOTE: The tag of an input/output area must be assigned to the leftmost location of the area by punching this tag starting in column 9.

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			BUFF1	RESV	80	
2		R		DCW	@$@	

PROGRAM ASSEMBLY

The tape input/output routines are included in the user's program by inserting the symbolically coded TI01 routines (a Honeywell-supplied punched card deck) with the source program. Assembly of a source program using these routines is accomplished in three phases: preassembly, phase I, and phase II.

Preassembly Phase

The input to this phase consists of a punched card deck (or a tape containing card images) which includes the program Translator, the TI01 routines, and the source program containing the tape input/output macro statements. DCA entries are grouped together in the source program, following the PROG, ORG, and ADMODE statements. This card sequence is illustrated in Figure 1-3. The source program may be in either two- or three-character addressing mode. The Translator program performs the following functions during this phase:

1. Translates the DCA entries into symbolic coding acceptable to Easycoder Assembler A.
2. Generates an 80-character label area in memory following the first DCA area. This is the area in which labels will be read and/or written.
3. Translates the tape input/output macro statements within the source program into Easycoder symbolic coding, providing for linkages to the TI01 routines.

Two or more source programs may be batch-processed during a Translator run if desired.

Phase I

Easycoder Assembler A (phase I) and the output of the preassembly phase are fed into the machine, producing a partially assembled intermediate deck.

Phase II

Easycoder Assembler A (phase II) and the intermediate deck are fed into the machine, producing a completely assembled machine-language object program. Phase I and phase II correspond to the assembly of a program which does not utilize the tape input/output routines and are described in the Easycoder 4K Operating Procedures manual (DSI-243).

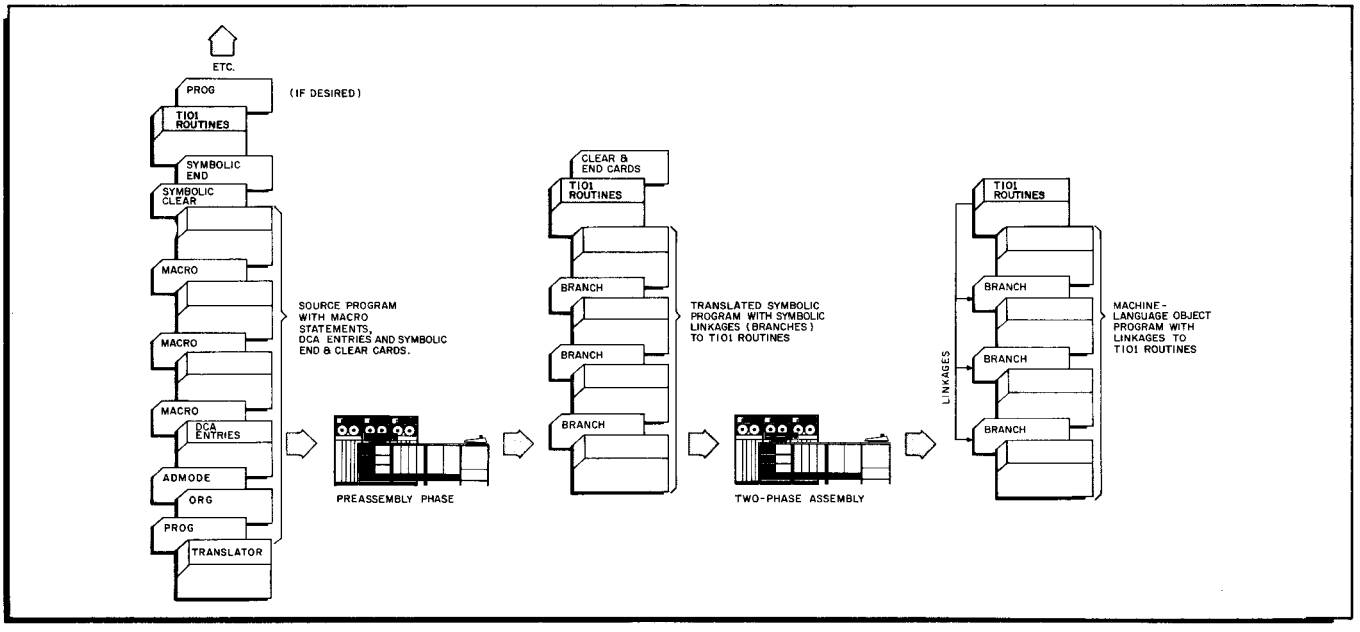


Figure 1-3. Assembly of a Program Using 1/2-Inch Tape I/O A

TAPE LABELS

Tape labels are the initial and terminal records of a tape file. The first record of each file, the header label, identifies the contents of the tape. The last record, the trailer label, identifies the end of a file or the end of a reel. 1/2-Inch Tape I/O A provides routines for tape reels which contain standard Series 200 labels, non-standard labels, or no labels at all.

STANDARD HONEYWELL LABELS

The primary function of the OPEN, CLOSE, and FEOR macro instructions is to check and create standard tape header and trailer labels. The standard Series 200 tape labeling system for half-inch tapes is described below.

Header Labels

The header label is the identification record at the beginning of each file. Before any processing is done, this label is examined to determine whether:

1. The information on the tape is the same information that the program is designed to process.
2. The reel is in the proper sequence.

For input operations, the OPEN macro statement checks the header label against information supplied by the programmer in the DCA entry. For output operations, the OPEN macro statement writes a new header label. The format of header labels for new or unused tapes differs from that for operational tapes.

NEW TAPES: All new or unused Honeywell half-inch tapes contain an 80-character dummy header label (1 BLNK in character positions 1-5 and tape serial number in character positions 6-10).

OPERATIONAL TAPES: The standard header labels for operational tapes have the following format (see Figure 1-4):

Character Position	Contents	Description
1 - 5	1HDRΔ	Identifies the header label.
6 - 10	SSSSS	Tape serial number.
11 - 15	FFFFF	File number.
16 - 20	-NNNΔ	Reel sequence number (if part of a file).
21 - 30	AAAAAAAAAA	File name.
31 - 35	YYDDD	Creation date (see Appendix A).
36 - 40	-NNNΔ	Retention cycle (the number of days a file is to be retained).
41 - 80		Available for additional control data, if desired. Otherwise, these positions are blank.

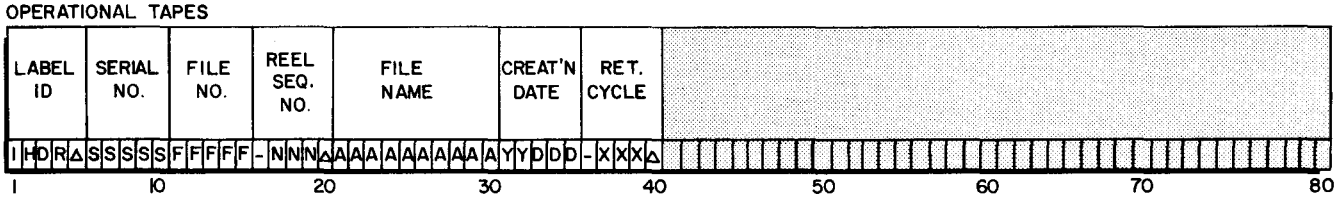


Figure 1-4. Header Label

Trailer Labels

Trailer labels are the identification records at the end of each file or reel. They are included on the tape to ensure that the entire reel (or reels) of a file has been accurately processed and to distinguish between the end of a reel and the end of a file.

During input operations, an end-of-reel or an end-of-file condition is detected by reading a trailer label. For output operations, an end-of-reel condition is produced by sensing the physical end of tape or by an FEOR macro statement; an end-of-file condition is produced by a CLOSE macro statement. A trailer label is written by either statement to indicate the particular condition of the tape:

1. An end-of-reel label indicates that an end-of-reel condition exists.
2. An end-of-file label indicates an end-of-file condition.

The FEOR or CLOSE statement writes two end-of-reserved-information records following the trailer label.

Trailer labels have the following format (see Figure 1-5):

Character Position	Contents	Description
1 - 5		Identifies the trailer label as follows:
	1EORΔ	End of reel;
	1EOFΔ	End of file;
	1ERIΔ	End of reserved information.
6 - 10	RRRRR	Record count.
11 - 20	IIIIIIIII	Item count. [1]
21 - 30	HHHHHHHHHH	Hash total, [1] i.e., number of items in a multi-reel file.
31 - 80		Available for additional information, if desired.

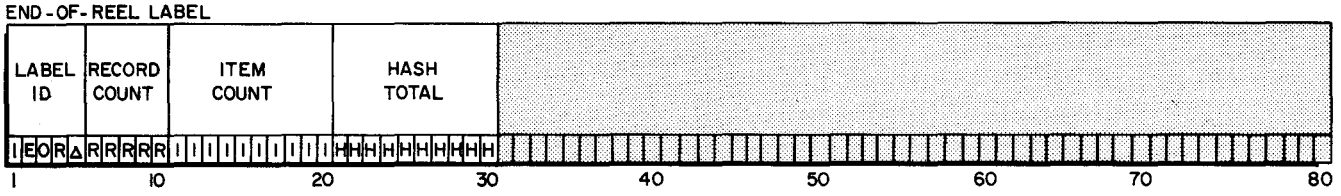


Figure 1-5. Trailer Labels

[1] 1/2-Inch Tape I/O A does not compute these values. If they are desired, they must be computed and stored in the trailer label by the programmer.

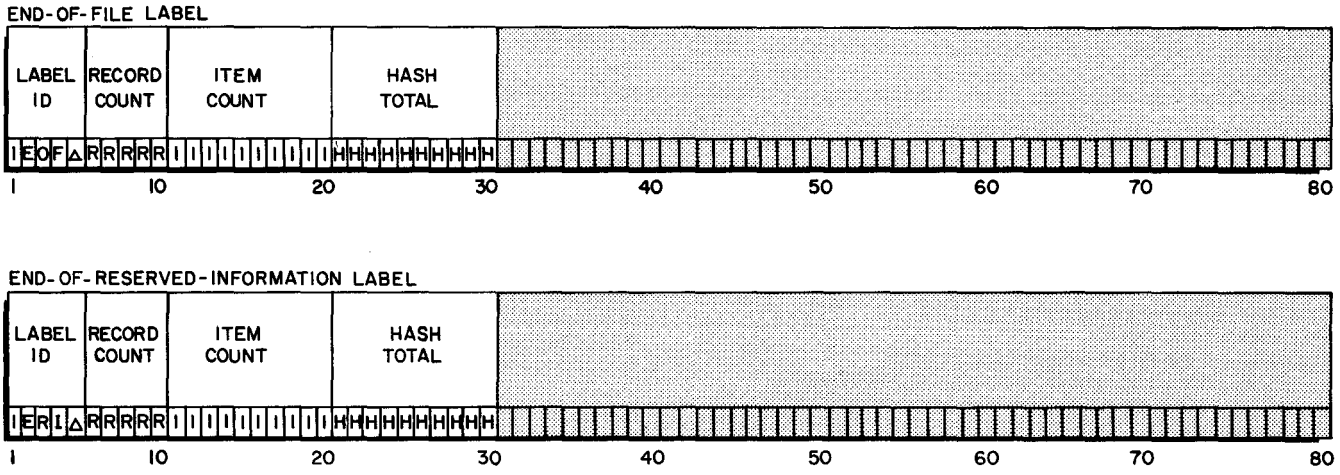


Figure 1-5 (cont). Trailer Labels

NON-STANDARD LABELS

The use of standard labels, non-standard labels, or no labels is defined by the programmer in the LBLTP line of the DCA entry for each file. He also specifies, in the DCA LBLCK line, whether the label being used is to be checked or not. If a file contains standard 80-character Honeywell labels, 1/2-Inch Tape I/O A reads and/or writes these labels and performs any desired checking. If non-standard labels are specified, the routine provides exits which allow the programmer to perform his own reading, writing, and checking of these labels. If a file contains no labels at all, the routine bypasses all label processing.

If non-standard labels do not exceed 80 characters in length, they may be described as standard in the DCA LBLTP line. This allows non-standard labels to be read and written by the tape input/output routine, but they must not be checked. In this case, therefore, the DCA LBLCK line must specify no label checking.

TAPES ACCEPTABLE TO 1/2-Inch Tape I/O A

1/2-Inch Tape I/O A has the ability to read and write tapes which use tape mark sensing (T/M tapes) as well as standard Honeywell tapes (non-T/M tapes). To read and write T/M tapes, the Type 203B tape control being used must be equipped with feature 050 (IBM format feature).

Non-T/M Tapes

Non-T/M tapes are standard Series 200 half-inch magnetic tapes. They may or may not contain banner characters as the first character of each data record and label.

Banner characters are not processed as input data; instead they are interpreted by input/output routine to distinguish between data records and file labels. For example, if bannered input files are being processed, the routine detects the end of data in a file or reel by noting a "1" as the first character of the trailer label. If the user specifies that banner characters are to be written in an output file, the routine writes a standard banner configuration of 101110 (octal 56) in all data records).

Data records which do not contain banner characters are also acceptable to 1/2-Inch Tape I/O A. In this case, all characters of a record are treated as data. The routine detects the end of data in a file or reel with no banners by checking for 1EOFΔ or 1EORΔ. The presence (or absence) of a banner character is indicated by the second item of the RFORM line in the DCA entry.

T/M Tapes

As non-T/M tapes are equipped with a banner character to provide a distinction between file labels and data records, T/M tapes are equipped with a tape mark as a delimiting character. A tape mark precedes the trailer label of a T/M tape. When the tape mark is sensed, the routine accepts the record which follows as a trailer label. 1/2-Inch Tape I/O A includes routines to read and/or write T/M tapes.

MACHINE REQUIREMENTS

1/2-Inch Tape I/O A is designed to read and write half-inch magnetic tape with Type 204B tape units. It is used with the following minimum Series 200 hardware configuration:

1. 4,096 characters of main memory.

   The TI01 routines require approximately 1700 locations when assembled in two-character addressing mode and 2400 locations in three-character addressing mode. Each DCA area occupies 62 character positions for two-character addressing, 68 character positions for three-character addressing; 80 character positions are required for the label area. The remaining portion of memory may be used by the main program.

2. A card reader and card punch to assemble the program.
3. One or more 204B tape units to be used by the main program during an object run.

Translator is run before the assembly of a 1/2-Inch Tape I/O A source program; therefore, any of the units of the minimum hardware configuration listed below that is needed for 1/2-Inch Tape I/O A may also be used by that program. The Translator run is described in Appendix B.

1. 4,096 characters of main memory.
2. A card reader and card punch to assemble the program.
3. Printer (optional).
4. Two 204B tape units (optional).

SECTION II

DEFINE COMMUNICATION AREA (DCA) ENTRIES

The checking, preparing, reading, and writing of standard header and trailer labels performed by 1/2 Inch Tape I/O A is based on information contained in the DCA (Define Communication Area) entries. One DCA entry must be coded for each tape file to be used by the program. The 22 lines of the entry are coded on the pre-printed form shown in Figure 2-1. Each entry describes the characteristics of one file.

EASYCODER
CODING FORM

PROBLEM DCA - SYMBOLIC STATEMENT _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___

	CARD NUMBER	TYPER	MARK	LOCATION	OPERATION CODE	OPERANDS	
	1 2 3 4 5	6	7	8 ... 14	15 ... 20	21 ... 62	63 ... 80
1					DCA		
2				FTYPE			
3				FUNIT			
4				FMODE			
5				RFORM			
6				RSIZE			
7				ISIZE			
8				AREAS			
9				PADNG			
10				BUFF1			
11				BUFF2			
12				LINKG			
13				RECNT			
14				RWDOP			
15				LBLTP			
16				LBLCK			
17				RLSEQ			
18				LBLID			
19				REDAT			
20				EXIT1			
21				EXIT2			
22				EXIT3			

Figure 2-1. Easycoder Coding Form for DCA Entry

The code DCA is printed in the op code field of the first line (the DCA header line). All subsequent lines of the entry have a code printed in the location field, and the op code fields are blank. All source-program coding is written in the operands fields of the 22 lines. Depending upon the program and the features of the system being used, the programmer need not enter coding in the operands field of every line. However, each line must be punched onto an 80-column card in the sequence shown above.

1. DCA Header Line

This line must be coded in each DCA entry. The operand of this line is the name of the tape file that is described by subsequent lines of the entry. The length of this operand must not exceed six characters. This file name will be the symbolic tag assigned to the first character of the DCA area in memory and must also be the operand of the macro instructions which refer to this file.

2. FTYPE - File Type

Two or three operands, separated by a comma, are coded in the FTYPE line. These operands must appear in the sequence described below.

a. The first operand indicates whether an input or an output operation is to be performed on the specified tape file. The word INPUT specifies an input operation; the word OUTPUT specifies an output operation.

b. The word ODD is coded as the second operand if the tape file is to be read or written in odd parity; the word EVEN specifies even parity.

c. The third operand, which designates the desired interrecord gap length,[1] is only coded for output operations. The word SHORT is coded if the shorter gap length is desired; the word LONG specifies the longer gap length.

In the example below, the program is using a Honeywell 204B-1 tape unit which can read and write records using interrecord gaps of 0.45" or 0.75". This example indicates that an output (tape write) operation is to be performed, that odd parity is desired, and that interrecord gaps of 0.45" are to be written.

```
1
2     FTYPE     OUTPUT,ODD,SHORT
3
```

3. FUNIT - Tape Unit

The FUNIT line must be coded in every DCA entry. The operand of this line is $x_1x_2x_3x_4$ where:

x_1 is the number of the read/write channel to be used by the file;

x_2 is the number of the tape control;

x_3 is the number of the initial tape unit;

[1] An exclusive feature of most Honeywell 204B tape units is that they are able to read and write magnetic tape using either of two interrecord gap lengths. The shorter gap length is normally used, leaving more tape surface available for data. To provide compatibility with non-Honeywell half-inch tapes, the option is provided to read and write using the longer of the two lengths.

x_4 is the number of the alternate tape unit, if desired, when processing a multi-reel file. The second reel of a multi-reel file is mounted on this unit, and input/output operations are automatically transferred to the second reel when an end-of-reel condition occurs on the first reel. Succeeding reels of the multi-reel file are then alternated between the initial and alternate tape drives, if successive reels are alternately mounted by the operator. (x_4 may be left blank.)

In the example below, the programmer specifies read/write channel 1 and tape control 0; tape 1 is the initial tape unit, and tape 2 is the alternate tape unit.

```
FUNIT     1Ø12
```

NOTE: No commas are used in the operands field of the FUNIT line.

4. FMODE - File Mode

The word LOCATE is coded in the operands field of this line for each DCA entry. 1/2-Inch Tape I/O A operates in the LOCATE mode (i.e., items are moved into and out of the processing area by the main program).

5. RFORM - Record Form

Two operands, separated by a comma, must be written in the RFORM line. The first operand is the word FIXED, indicating that fixed-length records are being used. The second entry is BAN if a banner character is the first character of each record in the file; if the records do not contain banner characters, NOBAN is written. The line of coding below indicates that fixed-length records are being used and that the first character of each record in the specified file is a banner character. If banner characters are used, the programmer must reserve an extra position in each input/output area for the banner character. All other banner character functions are handled by the input/output routine.

```
RFORM     FIXED,BAN
```

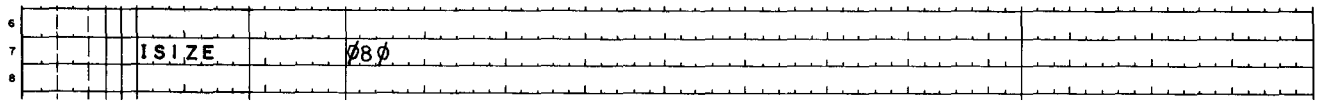
6. RSIZE - Record Size

The RSIZE line, which specifies the number of items in a record, must be coded in each DCA entry. For blocked records, the number of items per record is written in the operands field as a two-digit number. For unblocked records, the two-digit number 01 must be written. The line below indicates that there are four items in each record of the specified file.

```
RSIZE     Ø4
```

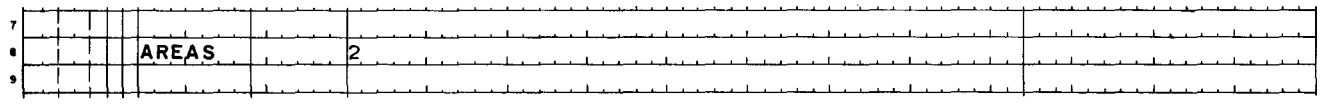
7. ISIZE - Item Size

A three- or four-digit number must be written in the operands field of the ISIZE line. These digits specify the number of characters in each item of the file, up to a maximum of 4095. The example below indicates that 80 characters are contained in each item of the file.



8. AREAS - I/O Areas

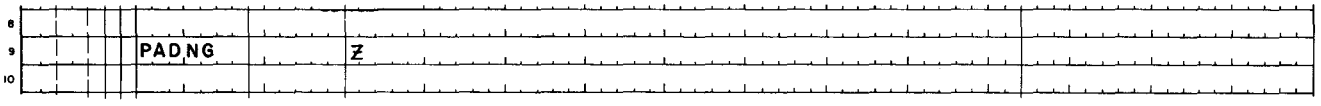
The operand of the AREAS line specifies the number of input or output areas being used for this file. The number 1 indicates one input/output area; 2 indicates that two areas are to be used. The example below indicates that two input/output areas are to be used by the tape file.



9. PADNG - Padding Character

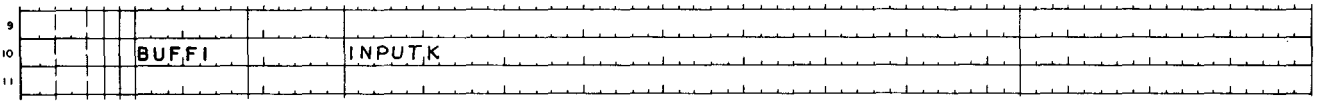
For output files, a single character must be written in the operands field of this line. This character will be used to fill an incomplete record (if necessary) when an output file is closed by a CLOSE macro statement or an end-of-reel condition is produced by an FEOR statement. This line is left blank for input files. The main program must check for padded records of input files. The example below indicates that if an incomplete record is produced when the file is closed, the unfilled character positions of this record are to be "padded" with Z's.

NOTE: If the file is sorted later, Z will be the highest alphanumeric character in the collating sequence. However, to obtain the highest representation in the collating sequence, a 77_8 (0, 8, 7) should be punched.



10. BUFF1 - Buffer #1

An address (either absolute or symbolic) must be written in the operands field of the BUFF1 line. This operand specifies the high-order (leftmost) address of the first input/output area (buffer area) to be used by the specified file. This area must be reserved and punctuated by the programmer and must be immediately followed by a record mark. The BUFF1 line below specifies INPUTK as the address of the first input/output area to be used by TIPTOP1 in handling the tape file.



11. BUFF2 - Buffer #2

If two buffer areas are to be used by this file, the high-order address (either absolute or symbolic) of the second area must be specified in the operands field of this line. This area must also be reserved and punctuated by the programmer and must be immediately followed by a record mark. If only one buffer area is to be used, the BUFF2 line is left blank.

Program time is reduced by the use of two buffer areas ("double buffering"). When processing is completed in one buffer area (BUFF1), the program begins processing the record in the second area (BUFF2). While the record in BUFF2 is being processed, the tape input/output routine reads into (or writes out of) the buffer area in which processing was completed (BUFF1). By this method of fully overlapped reading or writing, the input/output routine provides the opportunity for simultaneous processing.

12. LINKG - Linkage Field

This line must be coded in every DCA entry. The operand of this line is the communication link between the input/output routines and the programmer for this particular file and can be either of the following:

a. The address (either absolute or symbolic) of a field in memory. The programmer must punctuate this field with a word mark in the high-order (leftmost) position. If a symbolic tag is used, it must be defined or equated previously in the program to the low-order position of the linkage field.

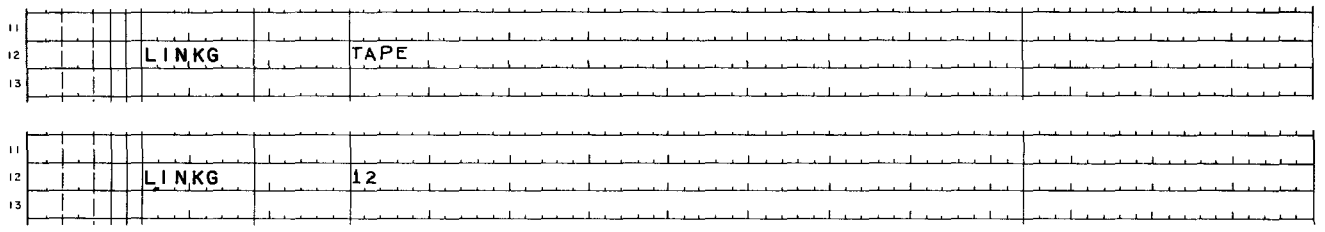
b. The address (either absolute or symbolic) of an index register. The programmer must punctuate this register with a word mark in the high-order position. If a symbolic tag is used, it must be defined or equated previously in the program to the low-order position of the desired index register.

For input operations, a GET macro statement causes an address to be placed in the field specified in the LINKG line. This address specifies the high-order (leftmost) location within an input buffer of the next item available for processing by the main program. For output operations, a PUT macro statement moves the address of the leftmost location of the next available buffer area to be filled into the field specified in the LINKG line (see pages 3-4 and 3-5).[1] The length of the address specified in LINKG (either two or three characters) must be the same as the address mode in which the tape input/output program is assembled.

The first example below specifies TAPE as the linkage tag for an input tape file. Each GET macro statement issued for this file will cause the high-order address of the next available item to be placed in a memory field tagged TAPE. Source-program instructions which address the low-order location of the item designated by TAPE (TAPE + item length-1) will move the next item to a processing area.

1Caution: 1/2-INCH Tape I/O A automatically stores and updates the the address in the linkage field. The results are unspecified if the programmer tries to manipulate the contents of this field.

The second example specifies index register 3 as the linkage field. Each GET macro statement that is issued for this file will load index register 3 with the high-order address of the next available item. This item will be moved to a processing area by a source-program MCW instruction addressing the item length - 1 added to the contents of index register 3.



13. RECNT - Record Count

This line must be left blank. It corresponds to a DCA field used by the input/output routine to maintain a count of records which have been read or written.

14. RWDOP - Rewind Option

This line provides the programmer with a rewind option when processing multi-file reels. The operand of the RWDOP line must be either REWIND; NOREWIND; or REWIND, LOCK. If the word REWIND is written, the tape will be rewound when it is opened or closed by the tape I/O routine. If NOREWIND is coded, the file will not be rewound at these points. Thus, when one file follows another file on the same reel, the tape remains in position for processing of the second file after the first file has been processed. If the words REWIND, LOCK are specified, the tape is rewound when it is opened and rewound and released when the file is closed.

15. LBLTP - Label Type

The LBLTP line describes the form of the tape labels used by the file. If standard Series 200 labels are used, the word STANDARD is written in the operands field. If non-standard labels are used, NON-STANDARD is written in the operands field. If the reel does not contain labels, the operands field is left blank.

16. LBLCK - Label Check

This line specifies the checking procedure to be used with standard Series 200 labels. If header and trailer labels are to be checked, the word CHECK must be written in the operands field. If non-standard labels are used, or if the standard labels being used are not to be checked, or if no labels are used, the operands field must be left blank. If standard labels are not checked and the user does not use own-coding to check trailer labels, the input/output routine assumes each trailer label to be an end of reel and performs a tape swap if necessary.

NOTE: For the second and subsequent files on an output tape (multi-file reel), label checking must not be specified.

17. RLSEQ - Reel Sequence Number

This line is used only with files which use standard tape labels. It is left blank if the first reel of the file described by the DCA entry is number 001. If the programmer assigns an alternate reel sequence number to the first reel, the number must be written in the operands field of this line as a three-digit number.

18. LBLID - Label Identification

For input files with standard tape labels, the operands field of this line contains the data against which character positions 21-30 (file name) of input header labels are to be checked. For output files, this operand is the identification to be written in positions 21-30 of the newly created output header label. If the specified file contains either non-standard labels or no labels at all, the operands field is left blank.

19. REDAT - Retention Date

This line must be left blank for all input files. If the file is being used for output operations, a three-digit number is written in the operands field. This operand specifies the number of days the file is to be retained after the creation date (see Table A-2, page A-3).

The date on which the program is being run must be supplied to the program (see Appendix A).

THE DCA USER EXITS

The last three lines of a DCA entry are user exits. These exits allow the programmer to branch to his own routines if he wishes to modify the label-handling routines of 1/2-Inch Tape I/O A or to check and/or write non-standard labels. For example, if the programmer wishes to write additional information in an output trailer label, he specifies EXIT2 in the DCA entry. If a non-standard label is to be read and checked in an output file, EXIT1 is coded.

The programmer specifies a particular exit by entering an address (either absolute or symbolic) in the operands field of the appropriate EXIT line. This address specifies the beginning location of a specialized labeling routine which he has written elsewhere in the program. If an exit is not desired, the operands field is left blank. However, either EXIT2 or EXIT3 must be coded for all input files.

Each user exit occurs at a specific time during the OPEN, CLOSE, and FEOR labeling routines. The points of departure from these routines are illustrated in the flow charts of these macro statements in Section III. When the exit occurs, the address of the next Tape I/O A instruction to be executed is automatically stored in a standardized area tagged $RE. Re-entry to Tape I/O A is provided by branching to $RE at the conclusion of the programmer's "own-coding" routine. The programmer addresses the 80-character label area in which standard labels are read and written via the tag $LBL, which refers to the high-order (leftmost) address of the label area.

20. EXIT1 - User Exit 1

For input tape files, this exit provides access to an "own-coding" routine which will perform either of the following operations:

a. Check additional information in standard input header labels.

b. Read and check non-standard input header labels.

For output tape files, EXIT1 provides access to a programmer routine which will perform any of the following operations:

a. Enter additional information into standard output header labels.

b. Modify a standard output header label that is to be written by the OPEN routine, or enter the file serial number into the five-character field$SER.

c. Write additional output header labels.

d. Create and write non-standard header labels.

21. EXIT2 - User Exit 2

For input tape files, this exit provides access to a programmer routine which will perform any of the following operations:

a. Check additional information in standard input trailer labels.

b. Check additional input trailer labels.

c. Read and check non-standard trailer labels.

This routine should branch to $RE only if the trailer label indicates end of reel (not end of file).

For output tape files, EXIT2 provides access to a programmer routine which will perform any of the following operations:

a. Enter additional information into standard output trailer labels.

b. Write additional output trailer labels.

c. Create and write non-standard output trailer labels.

d. Write a tape mark preceding the output trailer labels of T/M tapes.

22. EXIT3 - User Exit 3

For input tape files, this exit provides access to a routine which will notify the program that the end of file has been reached. At this point, the programmer's end-of-file procedure is executed. This procedure may be a CLOSE macro statement to close the tape file.

For output tape files, EXIT3 provides access to a routine which will write a tape mark following the standard output trailer label, if desired.

SECTION III

THE FIVE MACRO STATEMENTS

The macro statements used with 1/2-Inch Tape I/O A are OPEN, GET, PUT, CLOSE and FEOR (Force End of Reel). These statements are coded wherever needed in the source program and perform the functions described below.

OPEN - Open Tape File

The OPEN macro statement must be used with 1/2-Inch Tape I/O A to initialize each file in preparation for processing. This macro statement also takes care of the reading and checking of standard input header labels or the creation and writing of standard output header labels if the DCA entry for the file contains STANDARD in the LBLTP field and CHECK in the LBLCK field.

THE PROGRAMMER:

1. May write an address (either absolute or symbolic) in the location field.
2. Writes OPEN in the op code field.
3. Writes, in the operands field, the name of the file to be opened. This name may be up to six characters long and must be the same name that is specified in the operands field of the DCA header line for this particular file.

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
				OPEN	FILE6	

THE 1/2-INCH TAPE I/O A OPEN ROUTINE (input - Figure 3-1):

1. Rewinds the tape reel if the DCA RWDOP line specifies that the tape is to be rewound.
2. If the header label is a standard Series 200 label, the routine reads the label into the 80-character label area. If the file contains non-standard labels or no labels, a branch is made to step 7, below.
3. If a read error occurs, a branch is taken to the label error halt 0001 (see Appendix A).
4. Determines whether the label is to be checked or not. This determination is made from information supplied by the programmer in the DCA LBLCK line.
5. If the label is to be checked, the routine checks character positions 21-30 of the label against the data in the DCA LBLID line. The label is also checked for a reel sequence number of 001, unless otherwise specified in the DCA RLSEQ line.

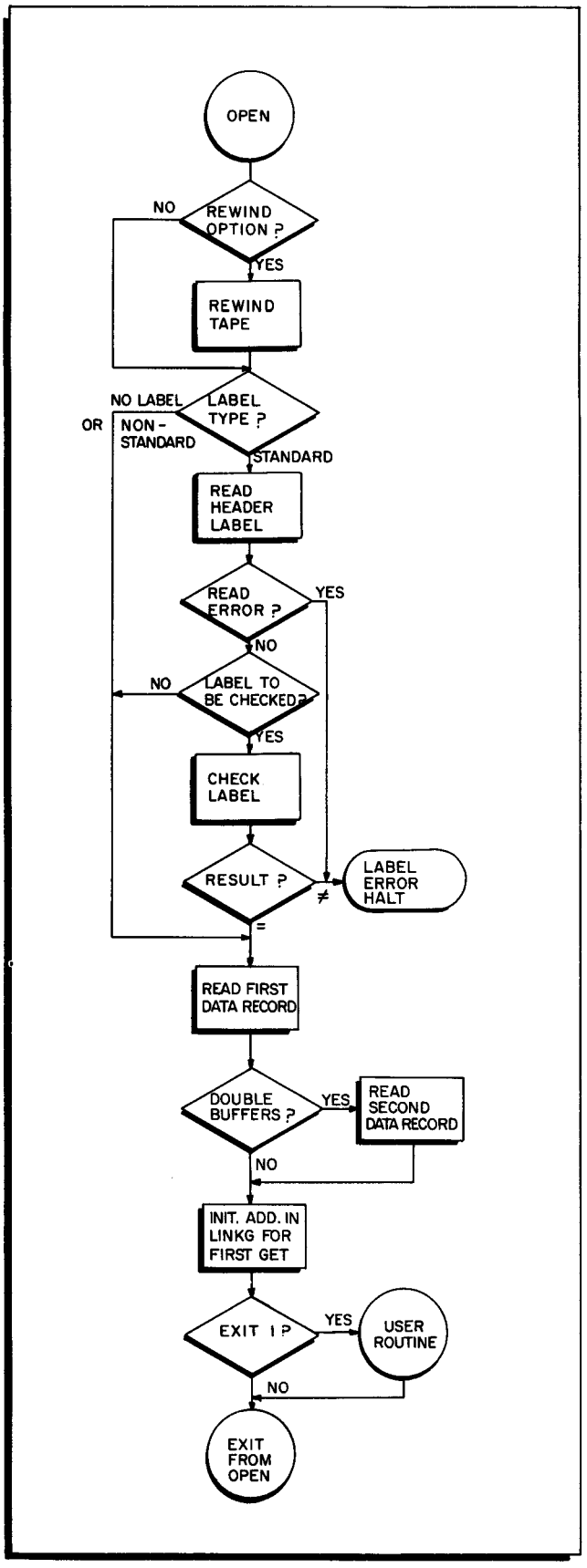


Figure 3-1. Input OPEN Operations

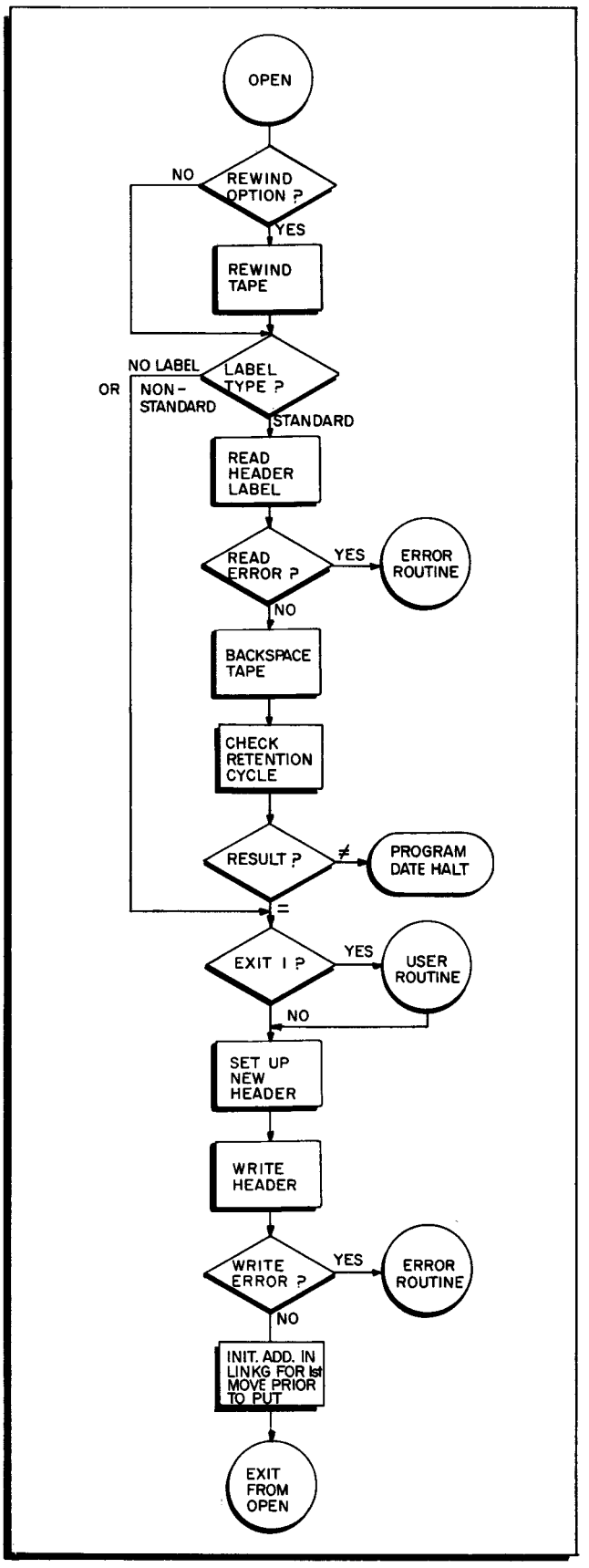


Figure 3-2. Output OPEN Operations

6. If a label-checking discrepancy is detected, a label error halt 0001 occurs. The action to be taken by the operator at this point is described in Appendix A.

7. Reads the first data record into the input area with single buffering or the first two data records into the two input areas with double buffering.

8. Initializes the field specified by LINKG.

9. Exits to the programmer's routine if EXIT1 is specified in the DCA entry.

10. Upon return from the programmer's routine, or if EXIT1 is not specified, control is transferred to the main-program instruction which follows the OPEN macro statement.

THE 1/2-INCH TAPE I/O A OPEN ROUTINE (Output - Figure 3-2):

1. Rewinds the tape if specified in the DCA RWDOP line.

2. Reads the header label into the label area if it is a standard label. If a non-standard label or no label at all, a branch is taken to step 6, below.

3. If a read error occurs, the program branches to the read/write error routine (see Appendix A).

4. Backspaces the tape.

5. Adds the retention cycle to the creation date of the output tape and compares this total with today's date. If this tape which is about to become the output file is still within its retention cycle, a program date halt 0006 occurs.

6. Exits to the programmer's routine if EXIT 1 is specified.

7. Sets up the new header label to be written by storing the label identification and reel sequence number (from the DCA area) in the label area. Positions 41-80 of the label area are not changed.

8. Writes the new header label.

9. If a write error occurs, a branch is taken to the read/write error routine (see Appendix A).

10. Sets the address of the first buffer area into the field specified by LINKG.

11. Exits to the main program.

GET - Get Item

The OPEN macro statement, when addressed to an input file, prepares the specified tape file for processing. The GET statement is always addressed to an input file. This statement reads a record (when necessary) from the specified file and makes an item of the record available to the program for processing.

The programmer specifies a memory field or the symbolic or absolute address of an index register in the LINKG line of the DCA entry. A GET statement places in this specified field or index register the address of the leftmost location of the next item available for processing in the input buffer ("locates" the item). The programmer may then move the item to a processing area by addressing the rightmost location of the item (by adding the item length minus one to the leftmost address) or by addressing the leftmost address and using an Extended Move instruction. The programmer must set his own punctuation in the input area for these moves.

THE PROGRAMMER:

1. May write a tag in the location field.
2. Writes the code GET in the op code field.
3. Writes the name of the file in the operands field. This file name must be the same name that is specified in the operands field of the DCA header line for this file.

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
				GET	FILEX

The basic functions of the routine entered by a GET statement are illustrated in Figure 3-3. A previous GET statement caused a blocked record consisting of four items to be read from a tape file (FILEX) into an input buffer. Each item is 50 characters long. The programmer moved the first item of this record to a processing area (PAREA). A subsequent GET statement located the second item, and the programmer moved the item to PAREA. This left two unprocessed items (items 3 and 4) in the input buffer. A third GET statement now places the memory address (1600) of the leftmost character of the next item (item 3) into a field tagged LINK, since the programmer has written LINK in the LINKG line of the DCA entry. The item may then be moved from the buffer to PAREA (via LINK) by programmed instructions.

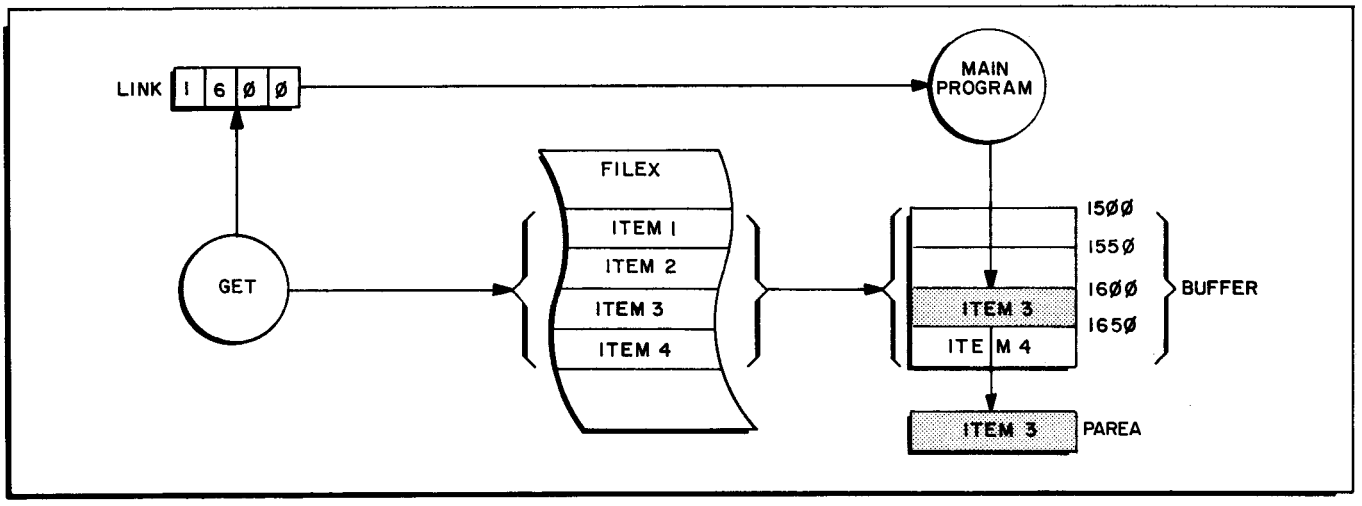


Figure 3-3. GET Macro Routine

PUT - Put Item

The OPEN macro statement, when addressed to an output file, prepares a specified tape to become the output file and initializes LINKG so that the programmer can use it for moving the first item into the output buffer. The programmer then moves an item from his processing area to

the output buffer by referencing the contents of the field specified by LINKG. The programmer can move the item from right to left into the output buffer by referring to the contents of LINKG plus one item length less one, or he can move the item from left to right into the output buffer by referring to the contents of LINKG in an Extended Move instruction.

The PUT statement is always addressed to an output file and always follows the moving of an item. This statement places into LINKG the address of the leftmost location in the output buffer that is available for the next item. When the buffer is filled, its contents are automatically written on the output tape by means of the PUT statement.

NOTE: No initializing PUT statement is required.

THE PROGRAMMER:

1. May write a tag in the location field.
2. Writes the code PUT in the op code field.
3. Writes the name of the file in the operands field. This file name must be the same name that is written in the operands field of the DCA header line.

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
				PUT	FILEY

Figure 3-4 illustrates the basic functions of the routine entered by a PUT macro statement. Two processed items (item 1 and item 2) have been moved from the processing area AREAP to the output buffer OBUFF by programmed instructions and PUT statements. Each item is again 50 characters long. A third processed item (item 3) is ready to be moved to OBUFF. The address of the leftmost character position (1800) of the available buffer area is made available

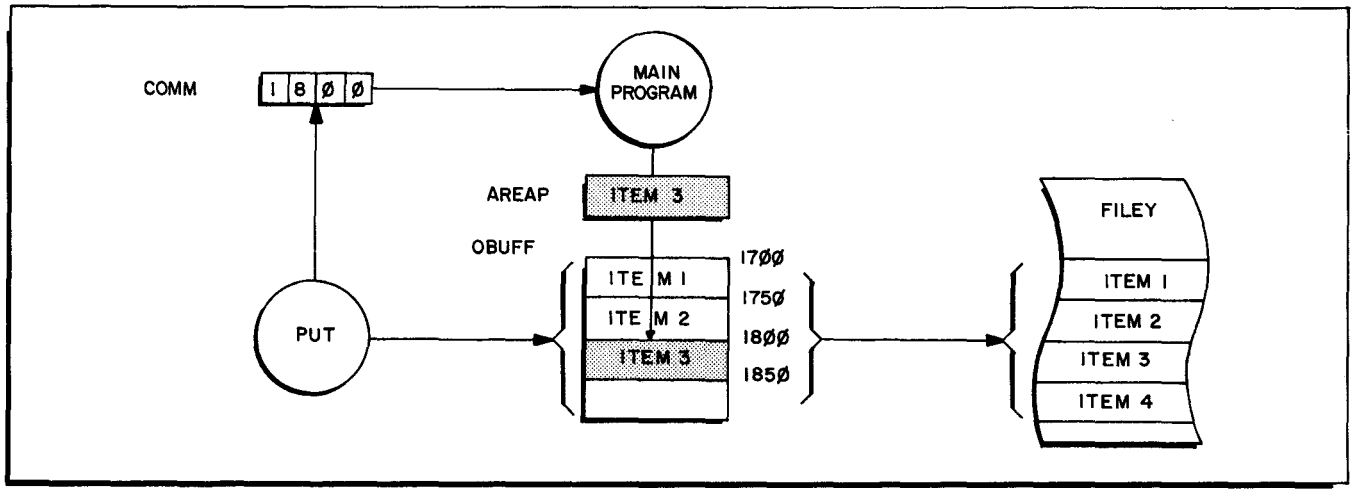


Figure 3-4. PUT Macro Routine

to the program via COMM, the tag entered in the DCA LINKG line. Item 3 is then moved from AREAP to OBUFF+100 (via COMM) by programmed instructions.

FEOR - Force End-of-Reel Condition

The FEOR macro statement is used to produce an end-of-reel condition before the physical end of reel (or tape mark) is sensed. This condition is produced by branching to the standard 1/2-Inch Tape I/O A end-of-reel routine at the points illustrated in Figure 3-5 (input) and Figure 3-6 (output). The FEOR statement causes the input or output tape to be rewound, increments the reel sequence number, writes trailer labels for output files, and performs a "tape swap" if specified in the DCA FUNIT line.

THE PROGRAMMER:

1. **May write a tag in the location field.**
2. **Writes the code FEOR in the op code field.**
3. **Writes the file name in the operands field. This file name must be the same name that is specified in the operands field of the DCA header line.**

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
				FEOR	FILE3	

The standard end-of-reel routine is executed when one of the following conditions occurs during input operations:

1. A "1" is detected as the first character of a record on a non-T/M tape using banner characters; or an end-of-reel record is detected on a non-T/M tape without banner characters.
2. A tape mark followed by a trailer record is detected on a T/M tape.

THE 1/2-INCH TAPE I/O A END-OF-REEL ROUTINE (Input - Figure 3-5):

1. **If standard, the trailer record is read into the label area. If the file contains non-standard labels or no labels, a branch to EXIT2 is taken.**
2. **If a read error occurs, a branch is taken to the trailer label halt 0002 (see Appendix A).**
3. **If label checking is specified, the record count in the label is checked. A program halt 0002 (see Appendix A) occurs if a discrepancy is detected. If no checking is specified, a branch to EXIT2 is taken.**
4. **Exits to the programmer's routine if EXIT2 is specified in the DCA entry.**
5. **If label checking is specified, a test is made for a 1EOR or 1EOF label. If no label checking is specified, steps 5 and 6 are omitted. The FEOR macro statement enters the routine at this point.**

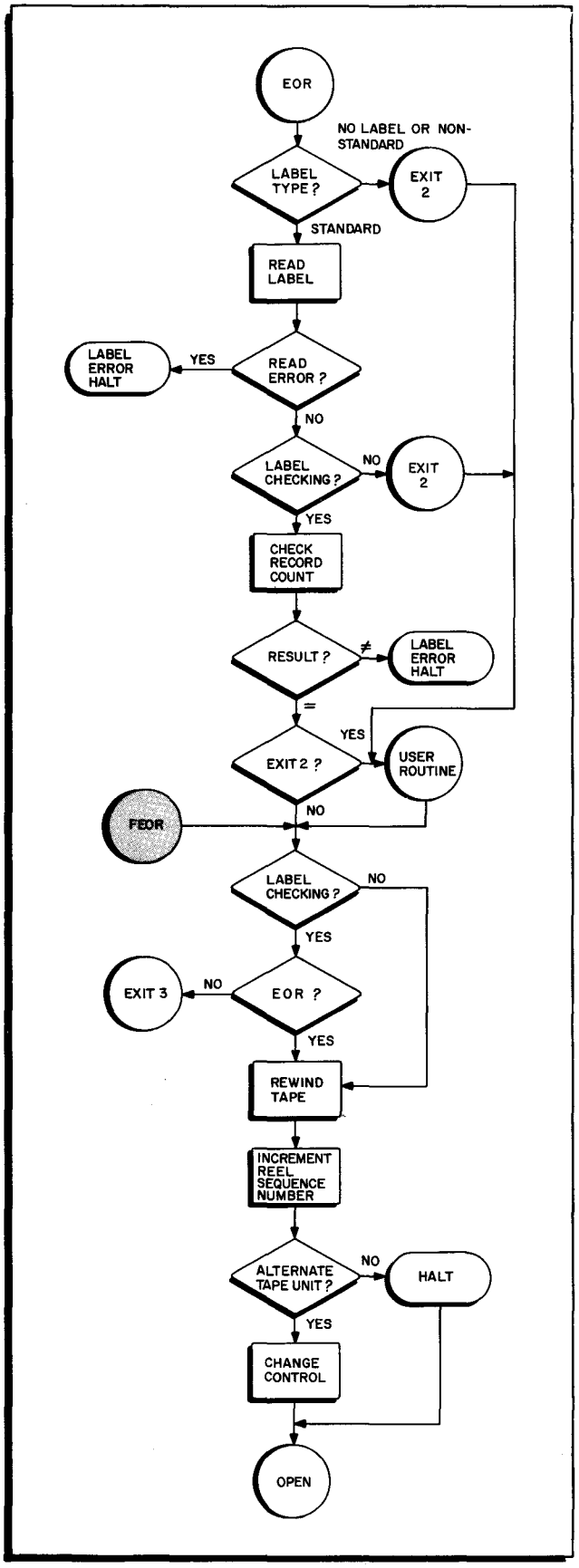


Figure 3-5. Input End-of-Reel Routine

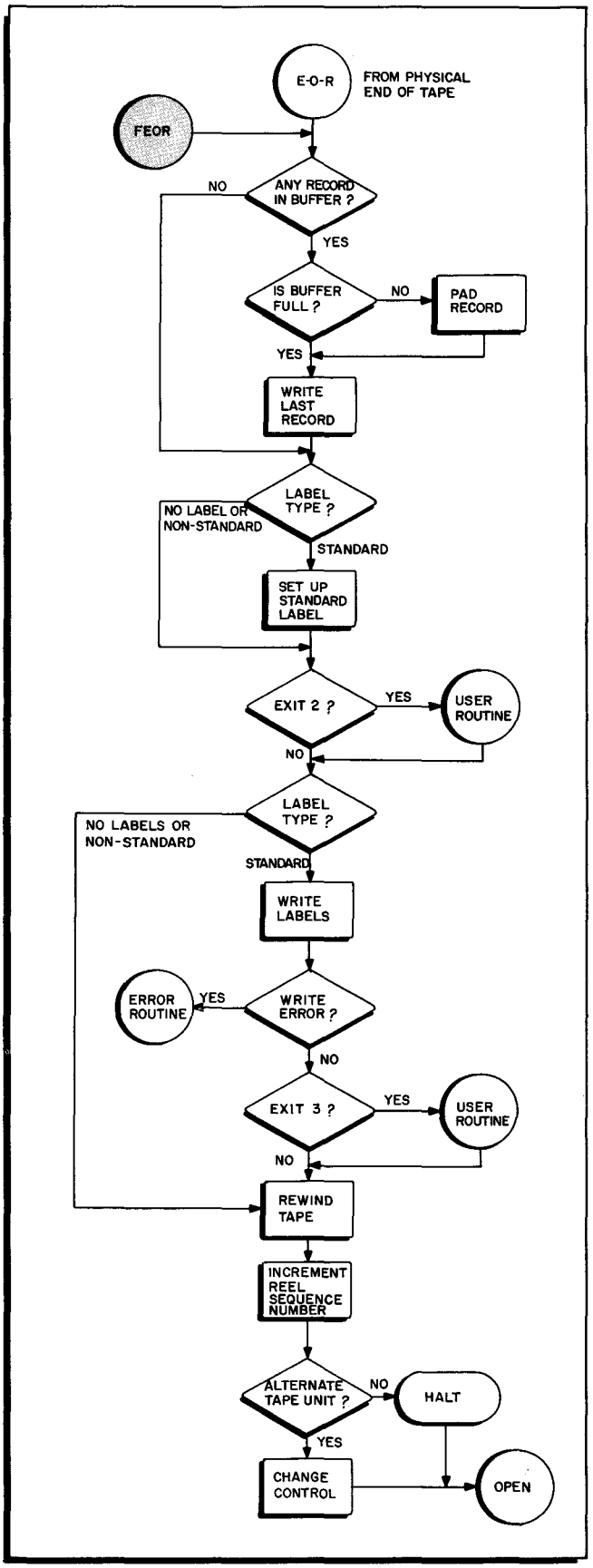


Figure 3-6. Output End-of-Reel Routine

6. If a 1EOR label is sensed, a branch is made to step 7. If a 1EOF label is sensed, a branch to EXIT3 is taken.

7. Rewinds the tape reel.

8. Increments the reel sequence number.

9. Transfers control to the alternate tape unit if an alternate unit is specified in the DCA FUNIT line. Control is then transferred to the OPEN routine.

10. If an alternate tape unit is not specified, halt 0005 occurs. The operator then changes tape reels and presses the RUN button. Control is then transfered to the OPEN routine.

The standard end-of-reel routine is executed for output tape files when the physical end of tape is sensed.

THE 1/2-INCH TAPE I/O A END-OF-REEL ROUTINE (Output - Figure 3-6):

1. The FEOR macro statement enters the routine at this point. If an incomplete record exists in the output buffer, the record is padded with the DCA PADNG character and written on the output tape.

2. If standard labels are specified, a new standard trailer label is constructed in the label area. If non-standard labels or no labels are specified, step #2 is bypassed.

3. Exits to the programmer's routine if EXIT2 is specified.

4. If standard labels are specified, writes the contents of the label area on the output tape, followed by two end-of-reserved-information (1ERIΔ) labels. If non-standard labels or no labels are specified, a branch is made to step 7, below.

5. If a write error occurs, a branch is taken to the read/write error routine (see Appendix A).

6. Exits to the programmer's routine if EXIT3 is specified.

7. Rewinds the tape reel.

8. Increments the reel sequence number.

9. Transfers control to the alternate tape unit (if specified). If an alternate unit is not specified, a halt occurs (same as input).

10. Branches to the beginning-of-reel (OPEN) routine.

CLOSE - Close File

During input operations, an end-of-file condition is detected by reading an end-of-file trailer label (1EOFΔ). When this condition occurs, an exit is automatically taken to the programmer's routine specified by EXIT3. The programmer's end-of-file procedure (which may be a CLOSE macro statement) is executed through this routine. For an output tape file, an end-of-file condition occurs only when a CLOSE macro statement is included in the main program.

THE PROGRAMMER:

1. May write a tag in the location field.
2. Writes CLOSE in the op code field.
3. Writes the name of the file to be closed in the operands field. This file name must be the same name that is written in the operands field of the DCA header line.

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					CLOSE	FILE7	
2							

THE 1/2-INCH TAPE I/O A CLOSE ROUTINE (Input - Figure 3-7):

1. If specified in the DCA RWDOP line, the tape is rewound.
2. When the tape rewind instruction is executed, or if the tape is not to be rewound, an exit to the main program is taken.

THE 1/2-INCH TAPE I/O A CLOSE ROUTINE (Output - Figure 3-8):

1. If an incomplete record exists in the output buffer, the record is padded with the DCA PADNG character and written on the output tape.
2. If the trailer label is a standard label, a new end-of-file label is constructed in the label area. If it is a non-standard label or if no label is used, this step is bypassed.
3. Exits to the programmer's routine if EXIT2 is specified.
4. Writes the contents of the label area on the output tape if it is a standard label; otherwise branches to step 8 below.
5. Writes two end-of-reserved-information labels following the new trailer label.
6. Branches to the read/write routine if a write error occurs.
7. Exits to the programmer's routine if EXIT3 is specified.
8. Rewinds the tape if specified in the DCA RWDOP line.
9. When the tape is rewound, or if the tape is not to be rewound, an exit to the main program is taken.

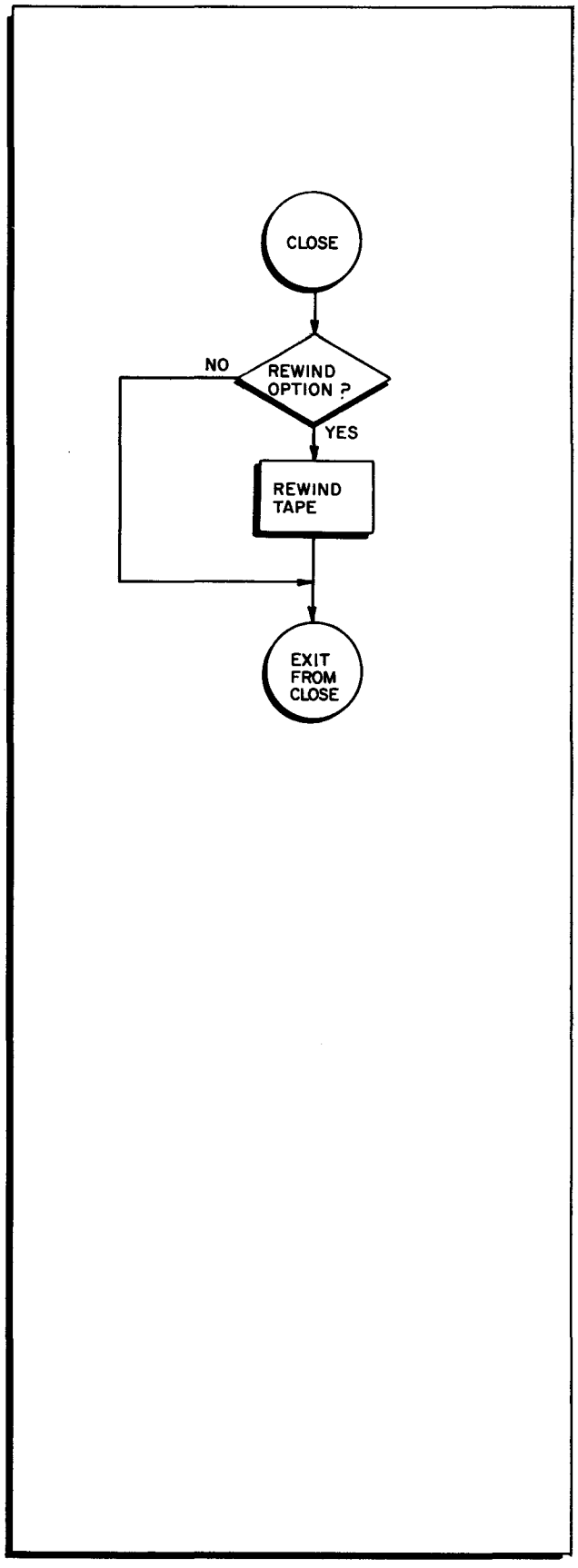


Figure 3-7. Input CLOSE Routine

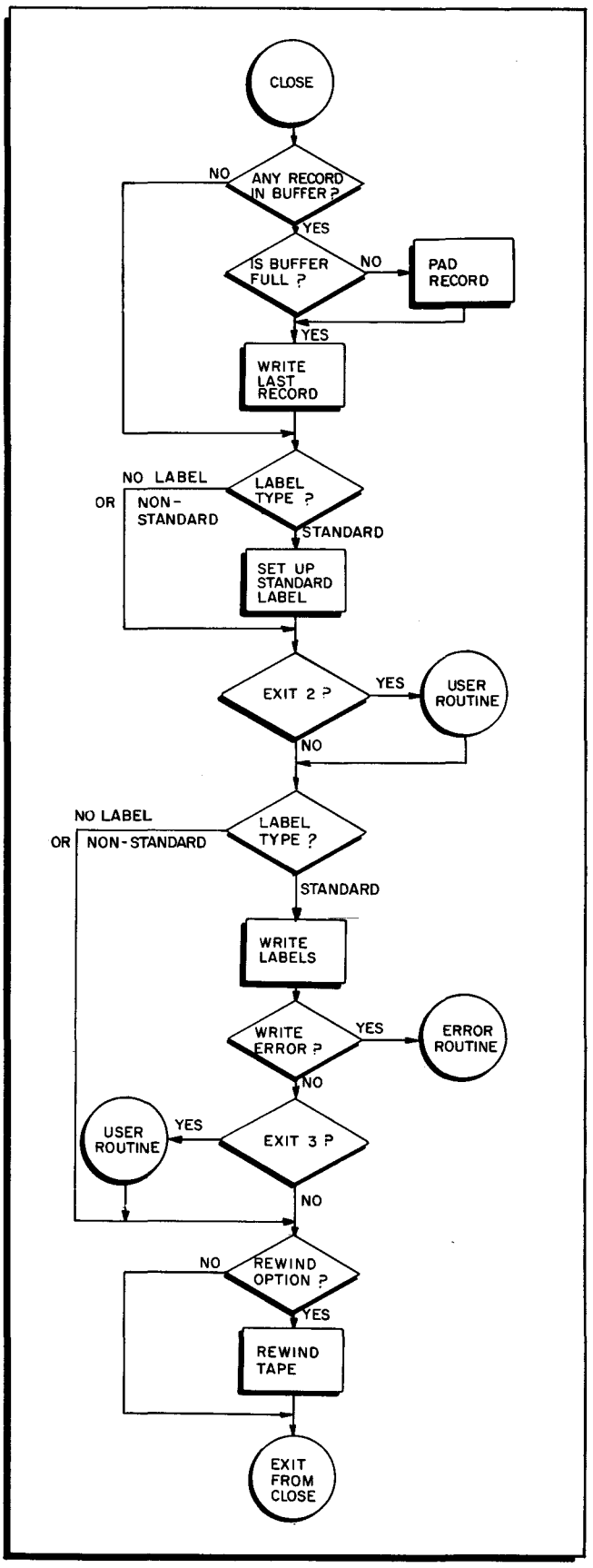


Figure 3-8. Output CLOSE Routine

APPENDIX A

PROGRAM HALTS WITHIN 1/2-INCH TAPE I/O A

Three types of halts may occur during the running of 1/2-Inch Tape I/O when the latter is assembled as part of an object program. A halt numbering system is used to identify the various halts within the system (see Table A-1). The contents of the A-address register can be read out on the control panel to display the halt number for the operator. The B-address register contains 3146_8 (IO) for 1/2-Inch Tape I/O A halts, except for the end-of-reel halt; in this case, the B-address register contains the drive number for mounting the next reel.

Table A-1. Halt Codes and Operator Action for the Program 1/2-Inch Tape I/O A

A-Address Register (octal)	Cause	Action
0005	End of reel	The end of reel of a multi-reel file has been encountered and no alternate tape drive has been specified for this file. Mount next reel for this file on the drive specified in the B-address register.
0001	Header label error; file ID or reel sequence number not equal	Turn SENSE switch 1 ON and depress the RUN button to reopen the file. (Depressing the RUN button while SENSE switch 1 is OFF will give control to exit 1 of the user's routine.)
0002	Trailer label error; record count on trailer not equal or trailer read error	Turn SENSE switch 1 ON and depress the RUN button. (If SENSE switch 1 is OFF, the trailer record will not be accepted, and the program must be rerun.)
0003	Write error	Write error persists after 63 tries; therefore, depress RUN button to attempt 63 more writes.
0004	Read error	Read error persists after 63 tries; therefore, turn SENSE switch 1 ON, and depress RUN button to try 63 more times. If read error still persists, turn SENSE switch 1 OFF, and depress RUN button to bypass the bad record.
0006	Program date halt; retention cycle not expired for this output tape	SENSE switch 1: ON - Check the tape mounted and if necessary, remount another tape. Depress the RUN button to reopen the file. OFF - Depress the RUN button, and accept the tape as is.

READ/WRITE ERRORS

Before each tape record is read by a GET statement or written by a PUT statement, a test is made for a tape transmission error on the previous read/write operation performed on the file. The OPEN, CLOSE, and FEOR macro statements read or write tape labels. If a read/write error exists in any of these cases, a branch is taken to the tape error routine.

For input files, the error routine attempts to correct the read error by repeating the tape read operation up to 63 times. If the tape is correctly read during this process, control is transferred to the appropriate point in the input/output routine. If the error persists after 63 correction attempts, the program halts.

If a write error occurs, the error routine backspaces the record (or label) in error, erases approximately three inches of tape, and attempts to rewrite it. If the error is still present after 63 attempts to erase and rewrite, the program halts.

LABEL-CHECKING HALTS

The OPEN and CLOSE routines check the contents of tape labels against information supplied in the DCA statement for the specified tape file. If a discrepancy exists, the machine halts. Label checking halts must be investigated to insure that the proper tape file is mounted.

PROGRAM DATE HALT

If the programmer desires that the retention cycle for a particular output tape be checked in order to determine if the file is outdated, he must load the five-character area in the program tagged $TDY (rightmost position) with today's date. The programmer can either make provisions in his program for a halt to occur (i.e., set SENSE switch 1 ON before loading) and, when it occurs (i.e., after the program is loaded), have the operator enter the day's date into $TDY via the control panel, or he can load $TDY using instructions within his program.

The decimal configuration of the program date is YYDDD where:

YY are the two terminal digits of the year, and

DDD are the three digits of the day within the year. The number of any day within the year is given in Table A-2.

If the file being checked is not obsolete, a halt as specified in Table A-1 will occur.

1/2-INCH TAPE I/O A RESTRICTIONS

The only restriction imposed on the system by 1/2-Inch Tape I/O A is that SENSE switch 1 must not be used by the main program.

Table A-2. Program Date Table

Day of Mo.	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.	Day of Mo.
1	1	32	60	91	121	152	182	213	244	274	305	335	1
2	2	33	61	92	122	153	183	214	245	275	306	336	2
3	3	34	62	93	123	154	184	215	246	276	307	337	3
4	4	35	63	94	124	155	185	216	247	277	308	338	4
5	5	36	64	95	125	156	186	217	248	278	309	339	5
6	6	37	65	96	126	157	187	218	249	279	310	340	6
7	7	38	66	97	127	158	188	219	250	280	311	341	7
8	8	39	67	98	128	159	189	220	251	281	312	342	8
9	9	40	68	99	129	160	190	221	252	282	313	343	9
10	10	41	69	100	130	161	191	222	253	283	314	344	10
11	11	42	70	101	131	162	192	223	254	284	315	345	11
12	12	43	71	102	132	163	193	224	255	285	316	346	12
13	13	44	72	103	133	164	194	225	256	286	317	347	13
14	14	45	73	104	134	165	195	226	257	287	318	348	14
15	15	46	74	105	135	166	196	227	258	288	319	349	15
16	16	47	75	106	136	167	197	228	259	289	320	350	16
17	17	48	76	107	137	168	198	229	260	290	321	351	17
18	18	49	77	108	138	169	199	230	261	291	322	352	18
19	19	50	78	109	139	170	200	231	262	292	323	353	19
20	20	51	79	110	140	171	201	232	263	293	324	354	20
21	21	52	80	111	141	172	202	233	264	294	325	355	21
22	22	53	81	112	142	173	203	234	265	295	326	356	22
23	23	54	82	113	143	174	204	235	266	296	327	357	23
24	24	55	83	114	144	175	205	236	267	297	328	358	24
25	25	56	84	115	145	176	206	237	268	298	329	359	25
26	26	57	85	116	146	177	207	238	269	299	330	360	26
27	27	58	86	117	147	178	208	239	270	300	331	361	27
28	28	59	87	118	148	179	209	240	271	301	332	362	28
29	29	*	88	119	149	180	210	241	272	302	333	363	29
30	30		89	120	150	181	211	242	273	303	334	364	30
31	31		90		151		212	243		304		365	31

*In leap years, after February 28, add 1 to the tabulated number.

APPENDIX B

OPERATING AND HALT PROCEDURES FOR TRANSLATOR

OPERATING PROCEDURES

Based on the following SENSE switch settings, the Translator program (program name TRANSL) uses peripheral devices and modifies the mode of addressing used in the generated routines:

SENSE Switch 1: ON - TIO1 routines read from tape 3

OFF - TIO1 routines read from cards

SENSE Switch 2: ON - Symbolic input to assembly written on tape 1

OFF - Symbolic input to assembly punched on cards

SENSE Switch 3: ON - Bypass printer

OFF - Printed listing of symbolic output prepared

SENSE Switch 4: ON - Tape input/output portion of the program is prepared in three-character mode

OFF - Tape input/output portion of the program is prepared in two-character mode

The input deck is arranged in the following sequence (as shown in the leftmost deck in Figure 1-3, page 1-4):

1. Translator program deck
2. Program card for first source deck
3. Origin card
4. Addressing mode card
5. DCA statements to be used by Translator
6. Source coding with HSM, CLEAR and/or END cards
7. TIO1 deck, if SENSE switch 1 is OFF
8. Number "2" through "7" above for the next and subsequent source programs (if desired)
9. 1EOFΔ card

Note that although the TI01 routines may be read either from cards or from tape, the Translator program and the remainder of the input deck are always read from cards.

The procedure for loading and starting the program follows:

1. Depress the STOP button;
2. Depress the INITIALIZE button;
3. Enter into the CONTENTS buttons the octal address assignment of the card reader control;

4. Depress the BOOTSTRAP button; and

5. Depress the RUN button.

PROGRAMMED HALTS

All programmed halts are coded in the A and B addresses to identify the halt condition. When the program halts, the reason may be determined by displaying the B-address register, and the buffer address may be displayed from the A-address register. The halt codes and operator actions in Translator are listed in Table B-1.

Table B-1. Halt Codes and Operator Action for the Program Translator

A-Address Register (Octal)	B-Address Register (Octal)	Cause	Action
4777	4777	End of translation.	Prepare to assemble the card-image output.
Card input buffer address	4040	DCA entry code (location field) in error or out of sequence.	Correct the next to last card in the stacker, refeed it and the last card, and press RUN.
Card input buffer address	0011	Hole-count error or illegal punch.	Correct the ejected card, refeed it, and press RUN.
Card-image input buffer address	0013	Tape read error.	Press RUN to try again.
Card-image output buffer address	0021	Tape write error.	Press RUN to try again.

HONEYWELL EDP TECHNICAL PUBLICATIONS
USERS' REMARKS FORM

TITLE: SERIES 200
1/2-INCH TAPE I/O A
SOFTWARE MANUAL

DATED: MARCH, 1966

FILE NO: 123.4105.020A.0-293

ERRORS NOTED:

Fold

SUGGESTIONS FOR IMPROVEMENT:

Fold

FROM: NAME _____ DATE _____

COMPANY _____

TITLE _____

ADDRESS _____

_____

Cut Along Line

FIRST CLASS
PERMIT NO. 39531
WELLESLEY HILLS
MASS.

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

Cut Along Line

Honeywell
ELECTRONIC DATA PROCESSING

CRP