

LION Input/Output Library

Programmer's Reference

First Edition

May, 1962

TABLE OF CONTENTS

	PAGE
I. INTRODUCTION	1
II. GENERAL FEATURES	2
A. Tape and Drum Formats	2
B. Input/Output Subroutines	3
C. Secondary Input/Output Subroutines	3
D. Special Functions	4
III. INTERNAL FORMAT SUBROUTINES	5
A. Output Subroutine	5
B. Input Subroutine	13
IV. EXTERNAL FORMAT SUBROUTINES	21
A. Output Subroutine	21
B. Input Subroutine	32
V. SECONDARY INPUT/OUTPUT SUBROUTINES	43
A. Auxiliary Routines	43
B. Service Routines	44
VI. SPECIAL FUNCTIONS	47
A. Continue Write	47
B. Locate File	48
C. Tape Check Routine	49
VII. SENTINEL BLOCK FORMATS	50
A. General Description	50
B. Usable Tape Field	52
C. Header Block	52
D. End-of-File Sentinel Block	53
E. End-of-Reel Sentinel Block	53

	PAGE
VIII. INTERNAL TAPE FORMAT	54
A. General Description	54
B. Label Blocks	54
C. Data Blocks	56
D. Bypass Sentinel Blocks	58
IX. EXTERNAL TAPE FORMAT	60
A. General Description	60
B. Label Blocks	60
C. Data Blocks	61
X. DRUM FORMATS	63
A. End-Of-File Sentinel	63
B. Internal Drum Format	63
C. External Drum Format	66
XI. CARD FORMATS	67
A. Start Card	67
B. Stop Card	68
XII. PROGRAMMING EXAMPLES	69
A. Sample Program No. 1	69
B. Sample Program No. 2	75
APPENDIX	82
INDEX	89

I. INTRODUCTION

LION, the Library of Input/Output Numerical Subroutines for the UNIVAC® 1107 Thin Film Computer, has been designed to provide efficient use of the various input and output devices. It will perform the operations of initiating or closing input or output files, identifying or checking them, writing or testing for End-of-File or End-of-Reel Sentinels, specifying Bypass of unneeded data, etc.. Included as part of LION are Service Routines, for transferring data from one type of peripheral equipment to another, e.g., card-to-tape. LION also provides for conversion of numerical data from one form to another, e.g., from fixed point binary to fixed point decimal.

LION is designed to operate through EXEC, the UNIVAC 1107 Executive System, and all actual reading and writing is performed by the Executive System I/O Functional Routines.

The routines are based on the use of UNISERVO * IIA tape units as the primary input and output equipment, with the magnetic drum serving as an auxiliary to the tape equipment. Secondary equipment includes the Card-Reader, Card-Punch, and High-Speed Printer.

All recording on tape is at a density of 250 bits per inch.

The routines which comprise the LION package are called in at assembly time through the use of macro-instructions in the SLEUTH source program.¹ These are system macros, i.e., the macro skeleton exists within LION in unparameterized form. The macro-calls, which are described herein, contain all of the parameters necessary for the operation of the subroutine.

*Trademark of the Sperry Rand Corporation.

¹See the manual on the SLEUTH Assembly System, UT 2574.

II. GENERAL FEATURES

The Input/Output Library is composed of a number of subroutines which control the flow of data between the object program and the peripheral equipment. These routines fall into three categories:

1. Primary Input/Output Subroutines, which control reading and writing of magnetic tape and/or drum.
2. Secondary Input/Output Subroutines, which control reading and writing of the secondary peripheral equipment. Two types of subroutines are provided: (a) Auxiliary Routines for input/output linked directly to the object program, and performed at execution time, and (b) Service Routines, for input/output performed indirectly, via an intermediate magnetic tape.
3. Special Input/Output Functions, which provide miscellaneous operations connected with magnetic tape and/or drum input output. These include Continue Write, Locate File, and Tape Check operations.

A. Tape and Drum Formats

LION will read and write tape or drum files in either of two formats, Internal or External. A full description of these two formats with illustrations, appears in Sections VII to X.

1. The Internal Format is designed for recording data in the same form as the data appears in core memory. This format is used on those files which are to be used for internal transfer of data within the computer system. It need not be in absolute binary form.
2. The External Format is designed for recording data in FIELDATA images for use with the punched card system or High-Speed Printer. This format is used on those tapes which are either input or output for the service routine which controls the operation of the card and printer equipment.
3. The magnetic drum will accept either format.

B. Input/Output Subroutines

In selecting an Input/Output subroutine, the user chooses one of the optional modes of operation which are available, and specifies the format for the tape or drum files by means of parameters and/or macro-instruction calls. The Input/Output subroutines deal with the individual object program input/output, and are incorporated in the object program as open subroutines, in the form of macro-instructions. The operation of these subroutines is such that the logic of the object program is concerned only with the individual data item. The tape format as specified by the user, generally provides for grouping several data items into a block for reading and writing, in order to minimize the time and tape space involved in traversing the inter-block spaces on the tape.

1. The Internal Format Input Subroutine will read tape (or drum) files which have been recorded in the Internal Format. The functions provided for use with tape operations are: Open File, Read, Close File, and Close Reel. The functions provided for use with drum operations are Open File, Read, and Close File.
2. The Internal Format Output Subroutine will write tape (or drum) files in the Internal Format. The functions provided for use with tape operations are: Open File, Write, Close File, Close Reel, and Bypass. The functions provided for use with drum operations are: Open File, Write, and Close File.
3. The External Format Input Subroutine will read tape (or drum) files which have been recorded in the External Format. The functions provided for use with both tape and drum operations are: Open File, Image Read, and Close File.
4. The External Format Output Subroutine will write tape (or drum) files in the External Format. The functions provided for use with both tape and drum operations are: Open File, Image Write, and Close File.

C. Secondary Input/Output Subroutines

1. Auxiliary Routines

The Auxiliary Routines provide for input/output on secondary peripheral equipment during the execution of the associated object program. They consists of Card Read, Card Punch, and Print subroutines.

2. Service Routines

The Service Routines provide the linkage between the tape or drum and other peripheral equipment, and are therefore limited to External Format files.

- a. The Card-to-Tape (or Card-to-Drum) Conversion Routine transfers card images from the Card Reader to magnetic tape or drum. The card image may be in FIELDATA, row-binary or column-binary form.
- b. The Tape-to-Card (or Drum-to-Card) Conversion Routine transfers card images from magnetic tape or drum to the card punch. The card images may be in FIELDATA, row-binary or column-binary form.
- c. The Tape-to-Printer (or Drum-to-Printer) Conversion Routine transfers printer images from magnetic tape or drum to the High-Speed Printer, and includes printer spacing information. The printer image will be in FIELDATA form.

D. Special Functions

LION includes two subroutines for positioning the input or output tapes. These can be used with either Internal or External Format tapes. A tape check routine is also provided. They are not applicable to magnetic drum operations.

1. The Continue Write Subroutine makes it possible to continue writing of output data on a previously closed tape file.
2. The Locate File Subroutine provides for the location of a file on a multi-file reel, and positions
3. Tape Check Routine

The Tape Check Routine is used to determine the amount of usable tape and to record a header block.

III. INTERNAL FORMAT SUBROUTINES

These subroutines perform control functions involved in writing and reading tape files in the Internal Tape Format. The control functions include those performed once for each block, once for each reel, and once for each file. The subroutines supply as well as audit non-data conventional control information on the tape.

A. Output Subroutine

The Output subroutine provides for the writing of tape files from data contained in the core buffer areas. The subroutine maintains control of the output items between the time that they are completed by the object program and the time that they are actually written on the output tape. The object program is concerned only with the fabrication of an output item. Its work is complete when it instructs the Output subroutine to place the item in an output file.

The functions provided by use of this Output subroutine include Open File, Write, Close Reel, Close File, and Bypass Setup. These functions are provided via macros which are incorporated into the coding each time they are used.

1. Open Output File, Internal Format: Two macros are required for each output file which is opened in the object program. One macro, IOCON, generates the file control table which is used with the output file. A second macro, OPNOUT, is used in the instruction section of the object program.

a. The Internal Output Control (IOCON) macro is normally used in the DBANK area of the program. It generates the file control routine table for the output file. The program contains only one IOCON macro for each output file. A unique name must be assigned each file. This name is used in associating the various macros which reference the file.

The output buffers are specified in the IOCON macro. Two output buffers are specified if standby operation is used. For demand operation, only one output buffer is specified. The term "buffer" in this context means a block of consecutive core storage locations, and it is not to be interpreted as a computer hardware component.

The maximum block size for the output file is equal to the buffer length. The output file may contain either fixed-length items or variable-length items (not both). The fixed-length item size, when used, is specified in the item length parameter of the IOCON macro. Variable length item size is specified by a parameter in the ITMOUT macro.

The functions of the file control routine include writing of the label block for each output file as well as for each output reel. The block count and amount of usable tape are stored for future reference. These items are updated each time a data block is written.

It is the responsibility of the user to insure that the output tape is properly positioned. Any error in positioning of the output tape results in the use of the File Control Error Exit.

The image for the label block is specified by two parameters. One parameter indicates the location of the File Label image; the second parameter specifies the location of the Free Words image.

The File Label image must include space for the entire image. The user specifies the reel number. If this field is set to zero, the file control routine changes it to one (1). The reel number is incremented after the File Label image is transferred to the output buffer. The date of Recording is placed into the image by the file control routine.

The OPNOUT, Write, Close Reel, Close File, and Bypass macros generate instructions which reference the file control routine. The file control routine contains checks which insure that each output file is opened only once. An attempt to write a closed reel (or closed file) or to close a closed file results in the use of the File Control Error Exit.

When the output file is on the magnetic drum, an entry in the IOCON macro specifies the drum table length. This value is used to check that drum recording is done only in the assigned drum table. The File Control Error Exit is used when the number of words to be recorded from a buffer exceeds the number of words in the drum table.

The format for the IOCON macro is:

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37
t					I O C O N			(pa) (pb) (pc) (pd) (pe) (pf) (pg) (ph) (pj) (pk) (pl) (pm) (pn):	

where:

t = Macro Location Tag

pa = File Name (unique to file)

pb = File Label image location, in form:

a,n

a = core address of first word

n = number of words

pc = Free Words image location, in form:

r,s

r = core address of first word

s = number of words

pd = Output Medium No. 1, in form:

g,h

g = coded digit to indicate equipment type:

= 2 for UNISERVO IIA tape unit

= 6 for magnetic drum

h = output tape unit or drum table (EXEC ROC I/O tag)

pe = Output Medium No. 2, in same form as item pd.

pf = Output Buffer No. 1, address of first word

pg = Output Buffer No. 2, address of first word

ph = Buffer length, number of words in each Buffer

pj = End-of-Reel Exit - when physical end-of-tape is encountered

pk = File Control Error Exit - when closed file is written or closed or when opened file is opened

pl = Write Error Exit - when an unrecoverable write error occurs

pm = Item length - number of words in fixed length items

pn = Drum table length (drum table length tag - used only for drum files)

Parameter pg is omitted, i.e., coded as (Ø), for demand buffer operation. Parameter pe is omitted when only one tape is assigned to the file.

Parameter pj is omitted when the standard ending reel procedure is used.

Parameters pb and pc are omitted when Label blocks are to be omitted on the output tape. Parameter pc (Free Words) may be omitted, however, pb (File Label) must be present if pc is present and to be used.

Parameter pm is used to specify the size of fixed-length items. It is omitted for files composed of variable-length items. For fixed-length item files, the block size (ph) is:

$n(pm) + 3$. where (n) is the number of items in a block.

b. The Open Output File (OPNOUT) macro generates a set of instructions which references the file control routine. It is normally used in the instruction section (IBANK) of the object program. There are no restrictions on the number of OPNOUT macros which may be coded to reference an output file, but only one of them can be selected by the program to perform the actual function. Each OPNOUT macro must contain the unique file name assigned to the output file.

The instructions generated by the OPNOUT macro provide for opening the output file. The format for this macro is:

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37
	t				OP N O U T		(pa)	:	

where:

t = Macro Location Tag

pa = File Name (unique to this file - same as used in IOCON)

2. Write, Internal Format:

The Write macro (ITMOUT) generates a set of instructions which references the file control routine. There are no restrictions on the number of ITMOUT macros which reference a single output file. Each ITMOUT macro must contain the unique file name assigned to the output file.

The instructions generated by the ITMOUT macro provide for transferring the individual output item to the output tape. The location of the output item is specified as a parameter.

As output items are specified, they are moved to the core output buffer(s). The file control routine initiates a "tape write" each time an output buffer is filled. When a block is written, the block count and amount of usable tape are updated.

The file control routine provides an End-of-Reel exit. This exit is specified in the IOCON macro. It is used when the amount of usable tape remaining on a reel is less than that necessary to write an output block.

If the End-of-Reel exit is omitted, the following standard ending-reel procedure is used:

- 1) Write End-of-Reel sentinel on terminated tape
- 2) Rewind terminated tape with interlock
- 3) Write label block on alternate tape
- 4) Transfer output item to buffer

The effect of this procedure is that of an ordinary ITMOUT.

When a tape write error (character count error) is discovered, the tape is reversed one block if the block size is 240 words or more. The block in error is rewritten. Rewriting is attempted three times in the event of repeated error. If this procedure is unsuccessful, the Write Error Exit is used.

Tape writing is always performed in the forward tape direction.

The format for the ITMOUT macro is:

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37			
	t				I	T	M	O	U	T	(pa) (pq)	:

where:

t = Macro Location Tag

pa = File Name (unique to file - same as used in IOCON)

pq = Output Item, in form:

s, t
where s = number of words per item
 t = core address of first word

Item s must always be specified for variable-length item files.
Item s is disregarded for fixed-length item files.

3. Close Reel, Internal Format Output File:

The Close Reel (OENREL) macro is used to terminate the writing on a particular tape reel. The OENREL macro must be included in the coding for all files which do not use the standard ending reel procedure. The OENREL macro generates a set of instructions which reference the file control routine. There are no restrictions on the number of OENREL macros which may be coded to reference a single output file, but only one of them can be selected by the program to actually perform the function. Each OENREL macro must contain the unique name assigned to the output file.

The output buffer(s) are emptied as a result of the OENREL only if the end of physical tape has not been encountered. Thus an OENREL execution before the End-of-Reel exit is used causes the output buffer(s) to be written on the output tape. The End-of-Reel sentinel is then recorded.

When the End-of-Reel exit has been used, the End-of-Reel sentinel is recorded immediately following the data blocks being written. In this case, the unrecorded item is recorded on the next reel.

The OENREL macro affords the user three options for the terminated reel: rewind, rewind with interlock, and no rewind. The functions of the OENREL macro include the recording of the label block on the alternate tape.

When the standard ending reel procedure is not used, the file control routine uses the End-of-Reel exit. When this exit is used, the OENREL macro must be executed before another ITMOUT macro is given. In the event that this exit is used and an ITMOUT or an OENREL is executed prior to an OENREL, the File Control Error Exit is used.

The format for the OENREL macro is:

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37
		t			O E N R E L			(pa) (pr)	:

where:

t = Macro Location Tag

pa = File Name (unique to file - same as used in IOCON

pr = Positioning procedure for terminated reel,
with parameters:

REWIND = rewind terminated tape

REWINT = rewind terminated tape with
interlock

NOREWI = do not reposition terminated
tape

4. Close File, Internal Format Output File:

The Close File (OENFIL) is used to terminate the writing of an output file. The OENFIL macro must be included in the coding for all output files. The OENFIL macro generates a set of instructions which reference the file control routine. There are no restrictions on the number of OENFIL macros which may be coded to reference a single output file, but only one of them may be selected by the program to actually perform the function. Each OENFIL macro which references an output file must contain the unique name assigned to the output file.

When the OENFIL macro is executed, any unrecorded items in the output buffer are written on the output tape. The End-of-File sentinel is then recorded on the output tape.

The OENFIL allows the user three options with respect to the output tape(s) assigned to the output file: rewind, rewind with interlock, or no rewind. If the rewind or rewind with interlock options are chosen, the procedure will apply to all output servos assigned to the file. When the no rewind option is used, all output tapes, with the exception of the last one, are left as positioned by the OENREL executed on the specific tape. The last output tape is positioned at the End-of-File sentinel.

The OENFIL macro must be included in the instructions which are executed when the output file is terminated. If an ITMOUT, an OENREL, or a BYPASS is executed subsequent to the execution of an OENFIL, the File Control Error Exit is used.

The format for the OENFIL macro is:

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37
					O E N F I L			(pa) (ps)	:

where:

t = Macro Location Tag

pa = File Name (unique to file-same as used in
IOCON)

ps = Positioning procedure for output tape
units, with parameters:

REWIND = rewind tape(s)

REWINT = rewind tape(s) with interlock

NOREWI = do not reposition tape(s)

5. Bypass, Internal Format:

The Bypass macro (BYPASS) provides a means of signalling the recording of a Bypass sentinel. There are no restrictions on the number of Bypass macros which reference an output file. Each Bypass macro must contain the unique name assigned to the output file.

The effect of the Bypass macro is to record any items in the buffer(s) on the output tape. The necessary Bypass sentinel information is placed in the output buffer. The address and number of words in the output buffer is presented to the object program in Register Q0.

The format of Q0 is:

35	17	0
BUFFER LENGTH		ADDRESS OF BUFFER

The BYPASS macro-instruction does not actually write the Bypass sentinel block on tape; it merely sets up the sentinel block in the output buffer, and the actual writing must be done by direct programming. The Bypass information must be followed by another Bypass sentinel, End-of-Reel sentinel, or End-of-File sentinel. The amount of usable tape and block count in the file control routine must be updated by the user to reflect the Bypass information recording before executing an ITMOUT, OENREL, or OENFIL on the output file.

The Bypass macro may be executed at any point in the object program after the output file has been opened via an IOCON macro. Use of a Bypass sentinel on a closed file results in the use of the File Control Error Exit. The Bypass macro may be used as part of the End-of-Reel procedure referenced by the End-of-Reel exit of the file control routine.

The format for the Bypass macro is:

1	TAG	7	8	9	FUNCTION 14	15	SUB FIELDS	37
	t			B	Y	P	A	S

where:

t = Macro Location Tag

pa = File Name (unique to file - same as used in IOCON)

B. Input Subroutine

The Input subroutine provides for advance reading of tape files into the core input buffers if standby buffer operation is used. The subroutine maintains the input items in the core input buffer area(s) until the individual items are requested by the object program.

The functions provided by use of this Input subroutine include Open File, Read, Close Reel, and Close File. These functions are provided via macros which are incorporated into the coding each time they are used.

1. Open Input File, Internal Format:

Two macros are required for each input file which is opened in the object program. One macro, IICON, generates the file control table which is used with the input file. A second macro, OPNIN, is used in the instruction section of the object program.

a. The Internal Input Control (IICON) macro is normally used in the DBANK of the program. It generates the file control routine table for the input file. The program contains only one IICON macro for each input file. A unique name must be assigned each file. This name is used for associating the various macros which reference the file.

The input buffers are specified in the IICON macro. Two input buffers are specified if standby operation is used in reading the input file. For demand operation, only one input buffer is specified.

The input buffers may be specified to contain more than one block of input data. In this case the file control routine reads sufficient blocks to fill the buffer(s). Processing of input items is initiated as soon as a single block is available in the buffer. Filling of a buffer is initiated only when the entire buffer is emptied.

The functions of the file control routine include the reading and checking of the label block for each input file as well as for each input reel. The block count and the amount of usable tape are stored for future reference. These items are updated each time a data block is read.

It is the responsibility of the user to insure that the input tape is properly positioned at a label block. Any error in positioning the tape results in the use of the File Control Error Exit.

The File Label is checked against the image provided by the object program. The location of this image is specified by a parameter. The image must include all words in the File Label. The input tape File Label is checked against this image.

The block and item length fields are not checked against the values in the image. However the input buffer length must equal or exceed the value in the block length field. The date check may also be omitted by clearing the date word in the image to zero. If the reel number is set to zero, it is changed to one. The reel number is incremented after the label check is completed.

Any error in checking the File Label results in the use of the Label Error Exit. A parameter to the IICON macro indicates if the file is to be read in the forward or backward direction for input files on magnetic tape. Magnetic drum files are read in the forward direction only.

For backward tape motion, the label checking procedure is omitted. However, the user must insure that the tape is positioned at an End-of-File sentinel or a label block. The block count and amount of usable tape are stored for future reference. The data blocks are presented to the object program starting with the data block which immediately follows the End-of-File sentinel or label block, in the backward sense.

The OPNIN, Read, Close Reel, and Close File macros generate instructions which reference the file control routine. The file control routine contains checks which insure that each input file is opened only once. An attempt to read a closed reel (or closed file) or an attempt to close a closed file results in the use of the File Control Error Exit.

The format for the IICON macro is:

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37
	t			I	I	C	O	N	(pa) (pb) (pc) (pd) (pe) (pf) (pg) (ph) (pj) (pk) (pl) (pm) :

where:

t = Macro Location Tag

pa = File Name (unique to this file)

pb = File Label image location, in form:

n,p

n = Core address of first Label word

p = Number of words in Label

pc = Direction of tape motion, with parameters:

FORW = Forward tape motion

BACK = Backward tape motion

(this parameter is omitted for drum files)

pd = Input Medium No. 1, in form:

g,h

g = coded digit to indicate equipment type

= 2 for UNISERVO IIA tape unit

= 6 for magnetic drum

h = Input tape unit or drum table (EXEC ROC I/O tag)

pe = Input Medium No. 2, in same form as item pd.

pf = Input Buffer No. 1, core address of first word

pg = Input Buffer No. 2, core address of first word

ph = Buffer length, number of words in each Input Buffer

pj = End-of-Reel Exit - when End-of-Reel sentinel is encountered on ITMIN

pk = File Control Error Exit - when closed file is read or closed or when opened file is opened

pl = Read Error Exit - when an unrecoverable read error occurs in reading the file

pm = Label Error Exit - when the tape label does not match Label image

For demand buffer operation, pg is omitted. When only one tape unit is assigned the file, pe is omitted. If pj is omitted, the standard ending reel procedure is used.

When the label block is not to be checked on the input file, pb is set to zero.

b. The Open Input File (OPNIN) macro generates a set of instructions which reference the file control routine. It is generally used in the IBANK of the program. There are no restrictions on the number of OPNIN macros which may be coded to reference an input file, but only one of them can be selected by the program to perform the actual function. Each macro must specify the unique file name assigned to the input file.

The instructions generated by the OPNIN macro provide for opening the output file. If label checking is specified in the IICON macro, the check is made as a result of the execution of the OPNIN macro. The label block is divided into two sections; the File Label and the Free Words. The File Label is checked as specified in the IICON macro. The Location of the Free words, if any, is presented to the object program in Register Q0. The format of Q0 is:

35	17	0
NUMBER OF FREE WORDS		ADDRESS OF FREE WORDS

The format for the OPNIN macro is:

1	TAG	7	8	9	FUNCTION	14	15	SUB. FIELDS	37
	t			O	P	N	I	N	(pa)

where:

t = Macro Location Tag

pa = File Name (unique to this file - same as used in IICON)

2. Read, Internal Format:

The Read macro (ITMIN) generates a set of instructions which references the file control routine. There are no restrictions on the number of ITMIN macros which reference a single input file. Each ITMIN macro must contain the unique name assigned to the input file.

The instructions generated by the ITMIN macro provide for filling the object program's request for an individual input item. The location of the input item is presented to the object program in Register Q0. The format of Q0 is:

35	17	0
NUMBER OF WORDS IN ITEM		ADDRESS OF ITEM

The item-advance technique is employed in the implementation of the ITMIN macro. The file control routine initiates a "tape read" each time an input buffer is emptied. As the data blocks are read from the input tape, the block count and amount of usable tape are updated.

The file control routine provides an exit for use when End-of-Reel sentinels are encountered during the reading of the input tape. The file control routine uses this exit when an ITMIN request references an End-of-Reel sentinel. If the exit is omitted, the standard ending-reel procedure is used for the subject input file.

The standard ending-reel procedure involves four steps:

- 1) Rewind terminated tape (without interlock)
- 2) Check File Label on alternate tape
- 3) Fill input buffer(s) from alternate tape
- 4) Present input item to object program

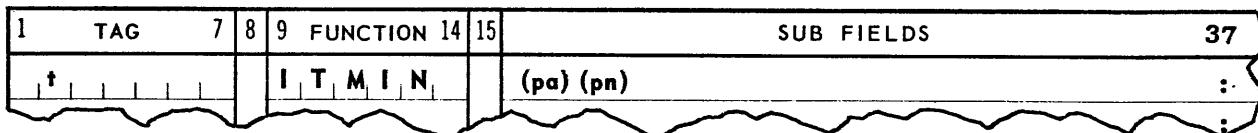
The effect of this procedure is that of an ordinary ITMIN. The File Label on the alternate tape is checked.

When tape reading is in the backward direction, the individual data items are presented in the reverse order from which they were written. However, the words within the data item are always in the same order as written.

When a tape read error (either parity or character count errors) is discovered in reading the data tapes, all possible means of recovery have already been initiated by the Executive System I/O Functional Routines. This error results in the use of the Read Error Exit.

Each ITMIN macro must specify an End-of-File exit. This exit is used when an ITMIN request references an End-of-File sentinel.

The format for the ITMIN macro is:



where:

t = Macro Location Tag

pa = File Name (unique to this file - same as used in IICON)

pn = End-of-File Exit - when End-of-File sentinel is encountered

3. Close Reel, Internal Format Input File:

The Close Reel (IENREL) macro is used to terminate the reading of a particular input reel. The IENREL macro must be included in the coding for all files which do not use the standard ending reel procedure (i.e., when pj in the IICON macro is not omitted).

The IENREL macro generates a set of instructions which reference the file control routine. There are no restrictions on the number of IENREL macros which may be coded to reference a single input file, but only one of them can be selected by the program to actually perform the function. Each IENREL macro must contain the unique name assigned to the file.

The IENREL macro allows the user three options with respect to the terminated reel: rewind, rewind with interlock and no rewind.

The functions of IENREL include checking the File Label on the alternate tape. The input buffer(s) are filled before control is returned to the object program. If an error is discovered in checking the File Label, the Label Error Exit is used.

If the standard ending reel procedure is not used, the file control routine uses the End-of-Reel exit. When this exit is used, the IENREL must be executed before another ITMIN macro is given. In the event that this exit is used and an ITMIN macro is executed prior to an IENREL, the File Control Error Exit is used; however, the use of an IENFIL rather than an IENREL is permitted. When only one input tape unit is assigned to an input file, only two input tape reel terminations are applicable: rewind and rewind with interlock. If no rewind is specified, the file control routine interprets this mode as a rewind.

The format for the IENREL macro is:

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37
		t		I	E	N	R	E	L

where:

t = Macro Location Tag

pa = File Name (unique to this file - same as used in IICON)

pr = Positioning procedure for terminated reel, with parameters:

REWIND = rewind terminated tape

REWINT = rewind terminated tape with interlock

NOREWI = do not reposition terminated tape

4. Close File, Internal Format Input File:

The Close File (IENFIL) macro is used to terminate the reading for a particular input file. The IENFIL macro must be included in the coding for all input files. The IENFIL macro generates a set of instructions which reference the file control routine. There are no restrictions on the number of IENFIL macros which may be coded to reference a single input file, but only one of them may be selected by the program to actually perform the function. Each IENFIL macro must contain the unique name assigned to the file.

The IENFIL macro allows the user three options for the tape(s) assigned to the input file: rewind, rewind with interlock, or no rewind. If either rewind or rewind with interlock is chosen, the procedure will apply to all tapes assigned to the file. When the no rewind option is used, all tapes, with the exception of the last one, are left as positioned by the last IENREL on the specific tape. The last tape is left as positioned by the last ITMIN on the tape.

The IENFIL macro must be included in the instructions which are executed as a result of the End-of-File exit. If an ITMIN or an IENREL is executed as the next operation following the sensing of an End-of-File sentinel, the File Control Error Exit is used.

The format for the IENFIL macro is:

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37
	t				I E N F I L			(pa) (ps)	:

where:

t = Macro Location Tag

pa = File Name (unique to this file - same as used in IICON)

ps = Positioning procedure for input servo with parameters:

REWIND = rewind tape(s)

REWINT = rewind tape(s) with interlock

NOREWT = do not reposition tape(s)

IV. EXTERNAL FORMAT SUBROUTINES

These subroutines perform control functions involved in writing and reading tape files in External Tape Format. The control functions include those performed once for each reel, as well as once for each file. The subroutines supply as well as audit non-data conventional control information on the tape.

A. Output Subroutine

The Output subroutine fabricates FIELDATA output images for punching an 80 column card or for printing 128 characters to the line. The images are generated as specified by operation parameters to the subroutine. The subroutine includes control for assembly of output images in the core buffers and for writing blocks of images on tape. The subroutine maintains control of the item between the time they are fabricated and the time they are written on tape. The object program is concerned only with specifying the parameters to the subroutine. The work of the object program is complete when the parameters are transferred to the subroutine.

The functions provided by this subroutine include Open File, Image Write, and Close File. These functions are provided by macros which are incorporated into the object program each time they are used.

1. Open Output File, External Format: Two macros are required for each output file which is opened in the object program. One more EOCON, generates the file control table for the output file. A second macro, EOPOUT, is used in the instruction section of the object program.
 - a. The External Output Control (EOCON) macro is normally used in the DBANK of the program. It generates the file control table for the output file. The program contains only one EOCON macro for each output file. A unique name must be assigned each file. This name is used in associating the various macros which reference the file.

The functions of the file control routine include the checking of the block count and amount of usable tape. These items are updated each time a data block is recorded on the output tape. The file control routine generates and records a label block for each file as well as for each reel.

The user specifies the type of output image desired (either card or printer) by a parameter. Two buffers are allocated for output data. The standby mode of operation is used by the file control routine.

Depending on the position of the output tape, one of two tape checks are performed. First, if the tape is positioned at the header block, this block is read and its information recorded. The tape is rewound and the label block is recorded.

Second, if the output tape is positioned at an End-of-File sentinel, the block is read and its information recorded. The tape is positioned at the beginning of this sentinel and the label block is written.

It is the responsibility of the user to insure that the output tape is positioned at an End-of-File sentinel when recording on multi-file reels. Any error in positioning of the output tape results in the use of the File Control Error Exit.

The EOPOUT Image Write and Close File macros generate instructions which reference the file control routine. The file control routine contains checks which insure that each output file is opened only once. An attempt to write or close a closed file results in the use of the File Control Error Exit.

When the output file is on the magnetic drum, an entry in the EOCON macro specifies the drum table length. This value is used to check that drum recording is done only in the assigned drum table. The File Control Error Exit is used when the number of words to be recorded from a buffer exceeds the number of words in the drum table.

The format of the EOCON macro is:

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37
	t			E	O	C	O	N	:

where:

t = Macro Location Tag

pa = File Name (unique to file)

pb = Output Medium No. 1, in form:

g,h

g = coded digit to indicate equipment type:

= 2 for UNISERVO IIA tape unit.

= 6 for magnetic drum.

h = output tape unit or drum table (EXEC ROC I/O tag)

pc = Output Medium No. 2, in same form as item pb.

pd = Number of characters per output image (80 or 128)

pe = File Control Error Exit - when closed is written or closed or when opened file is opened.

pf = Write Error Exit - when unrecoverable write error occurs

pg = Drum table length (drum table length tag used only for drum files)

The following parameters are used only for printer images:

ph = Page layout in form:

q,r,s

where:

q = number of printed lines, including line spacing, on output page; p > 0

r = number of lines to be spaced at top of page; if omitted, space two lines

s = number of lines to be spaced at bottom of page; if omitted, space two lines

q+r+s = page length (\leq 126 lines)

Note: If "q" is omitted, all of ph is omitted. The output page is considered to have 62 printed lines plus two lines spaced both at the top and the bottom of the page.

pj = line spacing increment for printing on
the output page; if omitted, single
spacing is implied.

pk = page numbering format, in form:

t, u

where:

t = characters to be skipped preceding
first page number, $1 < t \leq 127$. If
omitted, pk is omitted.

u = beginning output page number. If
omitted, page numbering commences on
second output page. Page numbers be-
gin with 2.

pl = date format, in form:

v, w

where:

v = characters to be skipped preceding
date. $1 \leq v \leq 116$. If omitted,
 pl is omitted.

w = printed line number on which date is to
be printed, $0 \leq w \leq 2$, set to one or
omitted, the date is printed on the same
line as the page number. If set to two,
the date is printed on the line following
the page number.

Note: The date occupies twelve character positions
and is in the form:

$M\Delta D, \Delta Y$

where:

M = month; three letter abbreviation

D = day of month; $1 \leq D \leq 31$

Y = year, four digits

pm = End of Page Exit - If not set to zero,
this exit is used before a line
is printed on a new page.

b. The Open Output File (EOPOUT) generates a set of instructions which reference the file control routine. It is normally used in the IBANK of the object program. There are no restrictions on the number of EOPOUT macros which may be coded to reference an output file, but only one of them can be selected by the program to perform the actual function. Each EOPOUT macro contains the file name assigned the output file.

The instruction generated by the EOPOUT macro provide for opening the output file. The format for this macro is:

1	TAG	7	8	9	FUNCTION 14	15	SUB FIELDS	37
	t			E O P O U T		(pa)	:	

where:

t = Macro Location Tag

pa = File Name (unique to this file - same as EOCON)

2. Image Write, External Format: The Image Write macro (IMGCUT) generates a set of instructions which reference the file control routine. There are no restrictions on the number of IMGCUT macros which reference a single output file. Each IMGCUT macro must contain the unique file name.

The instructions generated by the IMGCUT macro provide for the conversion of a binary field into FIELDATA characters and also for the storage of these characters in an image suitable for output on the card punch or on the printer. These images are subsequently recorded on the output tape. The length of the binary field as well as the type of conversion to be performed is specified by operation parameters. The location of the operation parameters is specified as a parameter.

The operation parameters specified in any IMGCUT macro must not define more than one output image. However, more than one IMGCUT may be used to define a single output image. For example a portion of a printed line might require either of two formats, which would be selected by the object program. The complete print operation then might call for three IMGCUT macros: A, B₁, and C, or A, B₂, and C. If the character count specified by any IMGCUT (or group of IMGCUT macros) exceeds the character count specified for the output image, the Write Error Exit is used.

As output images are specified by the IMGOUT, they are moved to the core output buffers. The file control routine initiates a "tape write" each time an output buffer is filled. The block count and amount of usable tape are updated each time a block is recorded.

If a tape write error (character count error) is discovered in writing the output tape, the tape is reversed one block. The block that caused the error is rewritten. This procedure is repeated three times in the event of repeated error. If it is not successful, the Write Exit is used.

Tape writing is always performed in the forward tape direction.

The format for the IMGOUT macro is:

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37			
	t				I	M	G	O	U	T	(pa) (pn)	:

where:

t = Macro Location Tag

pa = File Name (unique to file - same as used in EOCON)

pn = Operation parameter location, in form:

s, t

where: s = Number of parameter words

t = Core address of first parameter

- a. The format for operation parameters is the same as for instructions. Operation parameters may be written in SLEUTH Assembly System form. The order in which the parameter word is written is:

f, a, u, b, j

An asterisk (*) in the u-field sets the i-field to one, while an asterisk (*) in the b-field sets the h-field to one.

- b. The Table of Operations, Figure 1, gives a complete explanation of the conversions and functions that may be performed with operation parameters. The f-field specifies the number of

		35	29	25	21	17	16	15	0	
		f	j	a	b	h	i	u		Footnote
CONVERSIONS	Floating Point Binary to Floating Point Decimal	$6 \leq y \leq 13$	-	01						1
	Floating Point Binary to Fixed Point Decimal	$3 \leq y \leq 10$	$0 \leq n \leq 8$	02						2
	Fixed Point Binary to Fixed Point Decimal	$3 \leq y \leq 12$	$0 \leq n \leq 10$	03						2,3
	Octal Transfer	$1 \leq y \leq 63$	$1 \leq n \leq 12$	04						4
	Fielddata Transfer	$1 \leq y \leq 63$	$1 \leq n \leq 6$	05						5
	BCD Transfer	$1 \leq y \leq 63$	$1 \leq n \leq 9$	06						6
	Character Spacing	$1 \leq y \leq 63$	-	07	-	-	-	-	-	7
	Repeat Cycle	$1 \leq y \leq 63$	-	08	-	-	-	-	p	8
	Transfer Image	$1 \leq y \leq 63$	-	09	-	-	t	-	-	9

NOTE: The effective U is the location(s) of the binary number.

FIGURE 1: Table of Operations, IMGOUT Macro

Notes to Figure 1.

A space may be used in lieu of the plus sign for positive numbers; however, a minus sign will always be used with negative numbers.

(1) The floating point output format is:

±.XXXXXXXXXMM

where:

XXXXXXXX is the mantissa

MM is the power to the base 10 such that $-37 \leq MM \leq 38$

The character count, (y) includes:

sign of the mantissa
decimal point
sign of the characteristic
two digit characteristic

Inasmuch as this adds up to 5, a count of 6 indicates a 1-digit mantissa. A count of 13 indicates a 8-digit mantissa.

(2) The fixed point output format is:

±XXXXXXXXX.X

Leading zeros are suppressed for printer output.

Note: The number to be converted must be
 $10^{-10} \leq \text{number} < 10^{10}$

The character count, (y) includes the sign and decimal point. A character count of 3 indicates a sign, decimal point, and one digit. A count of 12 indicates a sign, decimal point, and ten digits.

The value in the j-field, (n) indicates the number of digits to the right of the decimal point.

(3) A parameter containing the operation 03 must be followed by another parameter with the following format. It is generally coded in SLEUTH language as an S (Sixth word generative) instruction.

35	29	23	17	11	0
SF	FL	BP	SD	NOT USED	

The unscaled binary point is located to the left of the most significant bit of the magnitude of the number.

SF field - the number of bit positions within the specified field binary point is moved to the right, where $0 \leq SF \leq 35$. Note that the maximum value of SF is governed by the value of FL.

FL field - field length of the binary number, including the sign; where $1 \leq FL \leq 36$. The maximum field length for the magnitude is 35 bits.

BP field - bit position of the left-most bit of the binary field (including the sign, when used). The most significant bit in a word is bit 35 and least significant bit is 0.

SD field - contains one of the following:

00 - conversion of a signed number and included decimal point. The positive sign (+) is replaced by a space.

01 - conversion of a signed number without decimal point. The positive sign (+) is replaced by a space.

02 - conversion of an unsigned number and included decimal point.

03 - conversion of an unsigned number without decimal point.

FIGURE 1: NOTES

(4) This operation is a transfer of 3 bits from memory, where (n) is the position of the first character, and

n = 1 for bits 35-33
n = 12 for bits 2-0

(5) This operation is a transfer of FIELDATA characters from memory, where (n) is the position of the first character, and

n = 1 for bits 35-30
n = 6 for bits 5-0

(6) This operation is a transfer of 4 bits from memory, where (n) is the position of the first character, and

n = 1 for bits 35-32
n = 9 for bits 3-0

(7) Skipped characters are filled with spaces.

(8) The Repeat Cycle is used when it is desired to repeat successive operations in an image. This function must precede the operations to be repeated. It may refer to any operation other than Op codes 08 and 09. (y) is the number of times a sequence of operations is to be applied. (p) is the number of operations in the repeat cycle.

(9) This operation transfers the output image to the buffer. This function must appear last in the parameter group.

For card image only the a-field is used.

For printer image: (y) is the number of lines to be spaced before the image is printed. This spacing is in lieu of the normal spacing. If (y) = 0, normal line spacing is implied.

If (t) is set, a new page is started before printing the image. (y) lines are spaced before printing. Normal line spacing does not apply when (t) is set.

FIGURE 1: NOTES

FIELDATA characters to be formed (except where otherwise noted). The j-field indicates the beginning position of the binary field. The a-field contains the desired OP code.

The output character fields are considered to begin at character position one for each image. This is the leftmost character position in the image. The user specifies the operation parameters so that the output character fields are defined sequentially across the output image from left to right.

If more than one (IMGOUT) is used to define a single card or printer image, the second and following IMGOUT macros define fields in the image sequentially following the previous IMGOUT. In this case the total character count specified in all IMGOUT macros for a specific card or printer image must not exceed the character count of the image (80 or 128).

3. Close Reel, External Format Output File: There is no Close Reel in the Output subroutine for External Tape Format. When the amount of tape remaining on a reel is less than the amount necessary to write a block, a close reel procedure is executed. This procedure involves three steps:
 - (1) Write End-of-Reel sentinel on terminated tape
 - (2) Rewind terminated tape with interlock
 - (3) Write label block on alternate tape
4. Close File, External Format Output File: The Close File (EXOEND is used to terminate writing of a particular output file. The EXOEND macro must be included in the coding for each output file. The EXOEND macro generates a set of instructions which reference the file control routine. There are no restrictions on the number of EXOEND macros which may be coded to reference a single output file, but only one of these may be selected by the program to perform the actual function. Each EXOEND macro must contain the unique file name.

When the EXOEND macro is executed, any images remaining in the output buffer are recorded. Then the End-of-File sentinel is recorded.

The EXOEND macro allows the user three options for the last output tape: rewind, rewind with interlock, no rewind. When the no rewind option is used, the last output tape is positioned at the End-of-File sentinel.

The EXOEND macro must be included in the instructions executed when the output file is terminated. If an IMGOUT is executed subsequent to the execution of an EXOEND, the File Control Exit is used.

The format for the EXOEND macro is:

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37
	t			E	X	O	E	ND	(pa) (px)

where:

t = Macro Location Tag

pa = File Name (unique to file - same as used in EOCON)

px = Positioning procedure for last output media, with parameters:

REWIND = rewind last tape

REWINT = rewind last tape with interlock

NOREWI = do not reposition last tape

B. Input Subroutine

The Input subroutine reads 80 column card images or 128 character printer images from the input tape. The card or printer images are in FIELDATA code. The subroutine provides for advance reading of blocks of images into the core input buffers. The images are disassembled (and converted if specified) according to parameter specifications. The subroutine maintains the images in the core input buffers until they are disassembled and stored.

The functions provided by this subroutine include Open File, Image Read and Close File. These functions are provided via macros which are incorporated into the coding each time they are used.

1. Open Input File, External Format: Two macros are required for each input file which is opened in the object program. One macro, EICON, generates the input file control routine table. The second macro, EOPIN, is used in the instruction section of the program.
 - a. The External Input Control (EICON) macro generates the file control routine for the input file. It is normally used in the DBANK of the program. Only one EICON macro may be contained in the program for each input file. A unique name must be assigned each file. This name is used in associating the various macros which reference the file.

Two input buffers are allocated for the input images. The standby mode of buffer operation is used in reading the input tape.

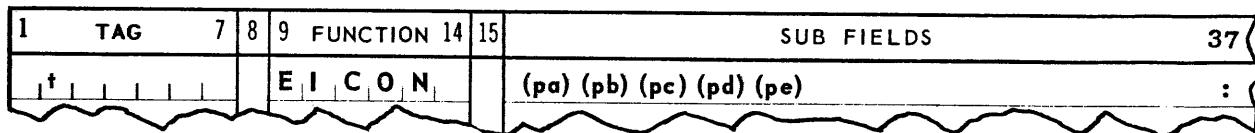
The functions of the file control routine include the reading and checking of the label block for each input file as well as for each input tape reel. The block count and the amount of usable tape are stored for future reference. These items are updated each time a data block is read from the input tape.

It is the responsibility of the user to insure that the input tape is properly positioned at a label block. Any error in positioning the input tape results in the use of the File Control Error Exit.

The label block is checked for image type and the proper reading mode is set up. The reading modes provide for 80 or 128 FIELDATA character images. The reading for External Format tapes is always in the forward direction.

The EOPIN, Image Read and Close File macros generate instructions which reference the file control routine. The file control routine contains checks which insure that each input file is opened only once. An attempt to read or close a closed file results in the use of the File Control Error Exit.

The format for the EICON macro is:



where:

t = Macro Location Tag

pa = File Name (unique to file)

pb = Input Medium No. 1, in form:

g, h

g = coded digit to indicate equipment type

= 2 for UNISERVO IIA tape unit

= 6 for magnetic drum

h = Input tape unit or drum table (EXEC ROC I/O tag)

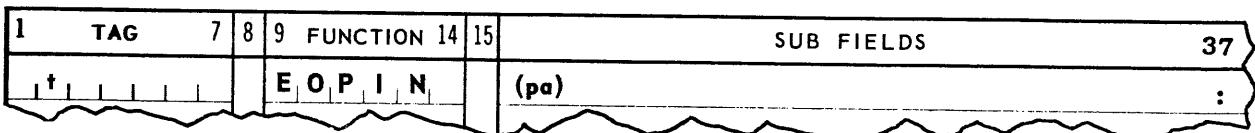
pc = Input Medium No. 2, in same form as item pb

pd = File Control Error Exit - when closed file is read or closed or when opened file is opened

pe = Read Error Exit - when unrecoverable read error occurs:

b. The Open File (EOPIN) macro generates instructions for opening the input file. These instructions, which are generally in the IBANK, reference the file control routine. There are no restrictions on the number of EOPIN macros which may be coded to reference an input file, but only one of them can be selected by the program to perform the actual function. Each macro must specify the unique name assigned to the file.

The format for the EOPIN macro is:



where:

t = Macro Location tag

pa = File Name (unique to this file - same as used in EICON)

2. Image Read, External Format: The Image Read macro (IMGIN) generates a set of instructions which reference the file control routine. There are no restrictions on the number of IMGIN macros which reference a single input file. Each IMGIN must contain the unique file name.

The instructions generated by the IMGIN macro provide for the conversion of a field of FIELDATA characters into a binary number. The storage of the binary number is also provided. The length of the Fieldata field as well as the conversions to be performed are specified by operation parameters. The location of the operation parameters is specified as a parameter.

The operation parameters specified in any IMGIN macro may not define more than one input image. However, more than one IMGIN macro may be used to define a single input image. If the character count specified in any IMGIN macro (or group of IMGIN macros) exceeds the character count of the input image, the File Control Error Exit is used.

The item advance technique is employed in the implementation of the IMGIN macro. The file control routine initiates a "tape read" each time an input buffer is emptied. As the blocks are read from the input tape, the block count and amount of usable tape are updated.

All tape reading is performed in the forward direction. If a read error (either parity or character count errors) is discovered, all possible means of recovery have already been initiated by the Executive System I/O Equipment Handler. This type of error results in the use of the Read Error Exit.

Each IMGIN macro must specify an exit for End-of-File sentinels. This exit is used when the IMGIN references an End-of-File sentinel.

The format for the IMGIN macro is:

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37
					IMGIN			(pa) (pf) (pg)	:

where:

t = Macro Location Tag
pa = File Name (unique to file, same as used in EICON)
pf = Operation parameter location in form s, t:
s = Number of parameter words
t = Core address of first parameter
pg = End-of-File Exit - when End-of-File is encountered.

- a. The format for operation parameters is the same as for instructions. Operation parameters may be written in Assembly System form. The order in which the parameter word is written is:

f, a, u, b, j

An asterisk (*) in the u-field sets the i-field to one (1), while an asterisk (*) in the b-field sets the h-field to one (1).

- b. The Table of Operations, Figure 2, gives a complete explanation of the conversions and functions that may be performed with conversion parameters. The f-field specifies the number of FIELDATA characters to be converted (except where otherwise noted). The j-field indicates the beginning position into which the converted number is stored. The a-field contains the desired Op code.

The FIELDATA character fields are considered to start at character position one (1) for each input image. This is the left-most character position in the input image. The user specifies the operation so that the FIELDATA character fields are defined sequentially across the input image from left to right.

If more than one IMGIN is used to define a single card or printer image, the second and following IMGIN macros define fields in the image sequentially following the previous IMGIN. The total character count specified in all IMGIN macros for a specific card or printer image must not exceed the character count of the image.

		35	29	25	21	17	16	15	0	
CONVERSIONS	Operations	f	j	a	b	h	i	u	Footnote	
		Floating Point Decimal to Floating Point Binary	$6 \leq y \leq 13$	-	01				1	
		Fixed Point Decimal to Floating Point Binary	$3 \leq y \leq 10$	-	02				2	
	STORE	Fixed Point Decimal to Fixed Point Binary	$3 \leq y \leq 12$	-	03				2,3	
		Octal Store	$1 \leq y \leq 63$	$1 \leq n \leq 12$	04				4	
	FUNCTION	FIELDATA Store	$1 \leq y \leq 63$	$1 \leq n < \underline{6}$	05				5	
		BCD Store	$1 \leq y \leq 63$	$1 \leq n \leq 9$	06				6	
	Operations	Character Skip	$1 \leq y \leq 63$	-	07	-	-	-	-	
		Repeat Cycle	$1 \leq y \leq 63$	-	08	-	-	-	r	7
		Terminate Image Transfer	-	-	09	-	-	-	-	8

NOTE: The effective U is the location(s) at which the converted characters are stored.

FIGURE 2: Table of Operations, IMGIN Macro

Notes to Figure 2.

A space may be used in lieu of the plus sign for positive numbers; however, a minus sign will always be used with negative numbers.

(1) The floating point input format is:

±.XXXXXXXXMM

where:

XXXXXXXX is the mantissa

and,

MM is the power to the base 10 such that $-37 \leq MM \leq 38$

The character count, (y) includes:

sign of the mantissa
decimal point
sign of the characteristic
two digit characteristic

Inasmuch as this adds up to 5, a count of 6 indicates a 1-digit mantissa. A count of 13 indicates a 8-digit mantissa.

(2) The fixed point input format is:

±XXXXXXXXX.X

Note: The number to be converted must be $10^{-10} \leq \text{number} < 10^{10}$

The character count, (y) includes the sign and decimal point. A character count of 3 indicates a sign, decimal point, and one digit. A count of 12 indicates a sign, decimal point, and 10 digits.

(3) A parameter containing the operation 03 must be followed by another parameter with the following format. It is generally coded in SLEUTH language as an S (Sixth word generative) instruction.

35	29	23	17	0
SF	FL	BP		NOT USED

FIGURE 2: NOTES

The unscaled binary point is located to the left of the most significant bit of the magnitude of the binary number.

SF field - number of bit positions within the specified field the binary point is moved to the right, where, $0 \leq SF \leq 35$. Note that the maximum value of SF is governed by the value of FL.

FL field - field length of the converted number, including the sign, where $1 \leq FL \leq 36$. The maximum field length of the magnitude is 35 bits.

BP field - bit position of the leftmost bit of the binary field (including the sign, when used). The most significant bit in a word is bit 35 and the least significant bit is 0.

(4) This operation is a transfer of 3 bits into memory, where (n) is the position at which the first character is stored, and

$$\begin{aligned} n &= 1 \text{ for bits } 35-33 \\ n &= 12 \text{ for bits } 2-0 \end{aligned}$$

(5) This operation is a transfer of FIELDATA characters to memory intact, where (n) is the position at which the first character is stored, and

$$\begin{aligned} n &= 1 \text{ for bits } 35-30 \\ n &= 6 \text{ for bits } 5-0 \end{aligned}$$

(6) This operation is a transfer of 4 bits into memory, where (n) is the position at which the first character is stored, and

$$\begin{aligned} n &= 1 \text{ for bits } 35-32 \\ n &= 9 \text{ for bits } 3-0 \end{aligned}$$

(7) The Repeat Cycle is used when it is desired to repeat successive operations in an image. The Repeat Cycle must immediately precede the operations that are to be repeated. It may refer to any operations other than Op codes 08 and 09.

FIGURE 2: NOTES

(y) is the number of times a sequence of operations are to be applied.

(r) is the number of parameter words in repeat cycle.

(8) The Terminate Image Transfer is used to terminate the transfer of data from an input image. This function must be last in the parameter group for the complete input image.

FIGURE 2: NOTES

3. Close Reel, External Format Input File: The Close Reel function is not available to the user in the Input subroutine for External Tape Format. When an IMGIN references an End-of-Reel sentinel, a Close Reel procedure is performed. This procedure involves three steps:

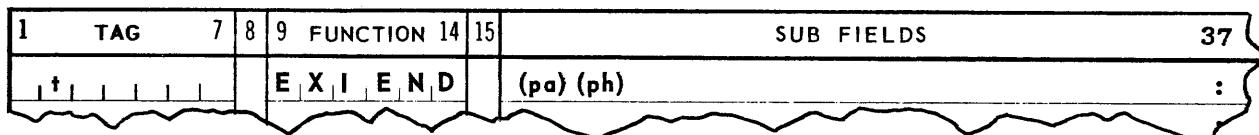
- (1) Rewind terminated tape (without interlock)
- (2) Read and check label block on alternate tape
- (3) Fill input buffers from alternate tape

4. Close File, External Format Input File: The Close File (EXIEND) macro is used to terminate the reading of an input file. The EXIEND macro must be included in the coding for all input files. The EXIEND macro generates a set of instructions which reference the file control routine. There are no restrictions on the number of EXIEND macros which may be coded to reference an input file, but only one of them can be selected by the program to perform the actual function. Each EXIEND macro must contain the unique file name.

The EXIEND macro allows the user three options for the last input tape; rewind, rewind with interlock, no rewind. When the no rewind option is used, the last input tape is left as positioned by the last read.

The EXIEND macro must be included in the instructions which are executed following the sensing of an End-of-File sentinel. If an IMGIN is executed following the sensing of an End-of-File sentinel, the File Control Error Exit is used.

The format for the EXIEND macro is:



where:

t = Macro Location Tag

pa = File Name (unique to file - same as used in EICON)

ph = Positioning procedure for input tape unit, with parameters:

REWIND = rewind last input tape
REWINT = rewind last input tape with
 interlock
NOREWI = do not reposition last input tape

V. SECONDARY INPUT/OUTPUT SUBROUTINES

The Secondary Input/Output subroutines provide for reading or writing of data on the secondary peripheral equipment: Card Reader, Card Punch, and High-Speed Printer. These subroutines consist of Auxiliary Routines and Service Routines.

A. Auxiliary Routines

Auxiliary Routines provide for input/output on secondary peripheral equipment during the execution of the object program with which they are associated. They are analogous to "on-line" operations.

1. Card Read Subroutine

The functions of the Card Read subroutine are: to accept the card image from the Card Reader, convert the input data as specified, and transfer the converted input data to core memory.

The conversions, if specified, provided by this subroutine are from decimal, either fixed point or floating point, to binary, either fixed point or floating point.

2. Card Punch Subroutine

The functions of the Card Punch subroutine are: to accept the output data from core memory, convert parts of this data as specified, assemble the card image, and punch the card.

The conversions, if specified, provided by this subroutine are from binary, either fixed point or floating point, to decimal, either fixed point or floating point.

3. Print Subroutine

The functions of the Print Subroutine are: to accept the output data from core memory, convert parts of this data as specified, assemble the printer image, and print a line.

The conversions are the same as for the Card Punch subroutine.

B. Service Routines

The Service routines contained in the Input/Output Library are independent operating programs. These routines provide the necessary control for transferring required images between the magnetic tape and the card equipment or the High-Speed Printer. They are analogous to "off-line" operations.

These routines are designed so that they require no external parameter information. All necessary parameter information is contained in the Start Card (for card read operation) or in the File Label (for card punch or printer operation).

Included in the Service routines are a Card-to-Tape Conversion routine, a Tape-to-Card Conversion routine, and a Tape-to-Printer Conversion routine. For any of these, the magnetic drum can be substituted for tape units.

1. Card-to-Tape Conversion Routine

The function of the Card-to-Tape Conversion routine is to transfer card images from the card reader to magnetic tape. This routine is capable of transferring images in FIELDATA format, row-binary format, or column-binary format.

Certain conventions apply to the card decks handled by this routine. Each card deck is headed by a Start Card. This card contains three fields, all punched in Hollerith format. One field contains the label for the card deck (this label is recorded in the File Label on the tape). A second field indicates the format in which the cards that follow are punched. The third field indicates the ending procedure to follow when the end of the deck is sensed. The ending procedure allows three options: rewind, rewind with interlock or no rewind.

The end of a card deck is signalled by a sentinel card. The three cards which follow this sentinel card are skipped. Hence these three cards must not contain useful information. The end of card deck signal results in the following action:

- a. The output buffers are emptied.
- b. An End-of-File sentinel is recorded on the output tape.
- c. The rewind procedure specified on the Start Card is followed.
- d. A card read is initiated to obtain the Start Card of the next card deck to process.

The end of a card deck may also be signalled from the Keyboard. This signal is given in response to a request to place more cards in the reader. A signal of this nature results in the first three steps in the action outlined above. The card read for a Start Card is not initiated.

All recording by this routine is in the External Format. The block count and amount of usable tape are updated each time a tape write is executed. When the amount of usable tape remaining on a reel is less than the amount of tape necessary to write the output buffer, a Close Reel procedure is instituted. This procedure is identical to the procedure described in the Image Write External Format section.

2. Tape-to-Card Conversion Routine

The function of the Tape to-Card Conversion is to transfer card images from magnetic tape to the card punch. This routine is capable of transferring images in FIELDATA format, row-binary format and column-binary format.

The label block on the input tape provides two fields of information which are punched on the Start Card. One field is the label, the second field describes the format of the card deck which follows.

Input tapes for this routine are recorded in the External Tape Format. Standby buffer operation is used in reading the input tape. When an End-of-Reel sentinel is encountered, a Close Reel procedure is instituted. This procedure is identical to that described in the Image Read External Format section.

When an End-of-File sentinel is encountered, all card images on the input tape have been punched. The end card is punched, followed by three blank cards. The input tape is then read for another label block. When the end-of-recording signal is received from the tape unit, the input tape is rewound. The operator is signalled to remove the terminated tape and to mount a new input tape.

3. Tape-to-Printer Conversion Routine

The function of the Tape-to-Printer Conversion routine is to transfer printer images from magnetic tape to the High-Speed Printer. This routine is capable of transferring FIELDATA images of 128 characters each.

The label block on the input tape serves to confirm the tape format. This routine will also print a tape prepared with FIELDATA card images (80 characters per input image). Since this latter image does not contain line spacing information, the output page is prepared with 62 printed lines per page. The first and last two lines on each page are skipped. No special printing, such as page numbers, appears on the output page(s).

Input tapes for this routine are recorded in the External Tape Format. Standby buffer operation is used in reading the input tape. When an End-of-Reel sentinel is encountered, a Close Reel procedure is instituted.

When an End-of-File sentinel is encountered all printer images have been printed. The routine resets the paper in the printer to the next page. A special notation to indicate the termination of the file of printer images is printed. The paper is then repositioned to the first line of the next page. Thus a page with a special printed sentinel separates each output obtained from the tape.

After the paper has been marked and positioned as indicated above, the input tape is read for a subsequent label block. When the end-of-recording signal is received from the tape unit, the input tape is re-wound. The operator is signalled to remove the terminated tape and to load a new input tape.

VI. SPECIAL FUNCTIONS

The special functions provide miscellaneous operations associated with input/output on tape files. They are not applicable to drum files.

The Continue Write macro makes it possible to initiate writing of output data on a previously closed tape file. The Locate File macro is used to locate a file on a multi-file reel. The Tape Check macro determines the usable length of the tape and writes a header block.

A. Continue Write

The Continue Write macro (CONWRI) is used to continue the writing of a tape file. It may be used indiscriminately on output files with data blocks longer than 240 words. For other files, it can only be used to overwrite the End-of-File sentinel.

The control routine generated by the CONWRI macro transfers the amount of usable tape from the input file to the output file. A check is made for an End-of-File sentinel or a label block in the block preceding the tape position. If a sentinel exists, the sentinel or label block is overwritten. If a data block precedes the tape position, the data block is overwritten.

The sequence of events by which the macro is used between input and output files is as follows:

1. Execute CONWRI macro
2. Close Input File
3. Open Output File

The format for the CONWRI macro is:

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37
	t			C	O	N	W	R	I

where:

t = Macro Location Tag

pa = Input/Output tape unit (EXEC ROC I/O Tag)

pb = Output File Name - file which is to furnish the output data

pc = Input File Name - file which positioned tape

Parameter pc may be omitted. This implies the tape is positioned at an End-of-File sentinel or at a label block. If pc is omitted, and the tape is not positioned following a sentinel block, the File Control Error Exit specified for pb is used.

B. Locate File

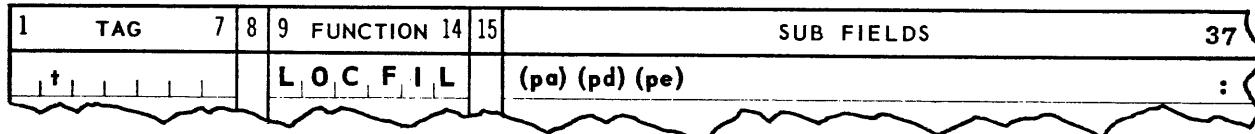
The Locate File macro (LOCFIL) provides for the location of a file on a multi-file reel. This function positions the tape to the label block of the specified file.

The LOCFIL macro allows two modes of operation. In one mode, the tape is positioned to the next label block on the tape. For this mode, item pd is omitted.

The second mode of operation provides for the location of a sequentially numbered file on the tape. For this mode of operation the files on a multi-file reel are considered as numbered sequentially with the label block for file one as the first block on tape. In this mode, the tape is rewound and then searched for the sequentially numbered file.

The LOCFIL macro has an error exit. This exit is used when the file search moves the tape beyond the recorded area. This exit is used as the result of an end of recording indication from the Tape Unit.

The format for the LOCFIL macro is:



where:

t = Macro Location Tag

pa = Input tape unit (EXEC ROC I/O Tag)

pd = File Position on Tape

pe = Error Exit - when end of recording is encountered

For the first mode of operation, pd is omitted. This causes the tape to be positioned to the next label block on the tape.

C. Tape Check Routine

The function of the Tape Check routine is to determine the amount of usable tape on a reel. It also records a header block on the tape which has been checked.

This routine writes as many blocks of 12,450 words as are necessary to determine the number of words which may be written on the subject tape. Each block written on the tape is twenty-five (25) feet in length (including interblock space). A cumulative sum of the words (and inter-block spaces) written is generated. When the end-of-tape signal is received from the tape unit this sum represents the amount of usable tape on the reel. The tape is then rewound and check read. The check reading verifies the amount of usable tape which was previously calculated as well as checking the tape for proper recording. The tape is rewound and the header block is recorded.

In calculating the amount of usable tape, only full 12,450 word blocks are accumulated. If the last block is not fully written on the tape, it is not included in the cumulative sum. Before this value is recorded on the subject tape, 12,500 words are subtracted from the amount of usable tape. This figure represents a 25-foot trailer.

After the header block has been recorded the tape is rewound. The operator is informed that the tape is to be removed from the tape unit and another tape is to be loaded for checking.

VII. SENTINEL BLOCK FORMATS

A. General Description

The operation of the Input/Output Library is such that no more than one sentinel block is recorded on each tape. (The writing of the By-pass Sentinel Blocks is a function of the object program). This sentinel block is either a Header block, an End-of-Reel sentinel block, or an End-of-File sentinel block. All other blocks are either Data blocks or Label blocks.

The Header block is recorded only when tape is checked by the Tape Check routine. It is replaced by a Label block which is written at the first actual use of the tape.

The End-of-Reel sentinel block is recorded at the end of a reel, and is not normally overwritten, except when the tape is completely rewritten.

The End-of-File sentinel block is recorded to signify the end of a file, and is replaced by the next Label block recorded on the tape.

The Drum End-of-File Sentinel is the only sentinel used on the magnetic drum. It consists of a single word containing the sentinel identification 747574757475 (octal). It may be replaced by the first word of the Label block of the succeeding file.

Identical sentinels and sentinel blocks are used for both Internal and External Formats. Hence multi-file reels may contain both Internal Format files and External Format files. However, the blocks within a file may not be mixed.

All sentinel blocks have essentially the same format. A tape sentinel block is fifty words in length, with the first word (00) and the last word (49) containing the sentinel identification 747574757475 (octal). Figure 3 illustrates a sentinel block:

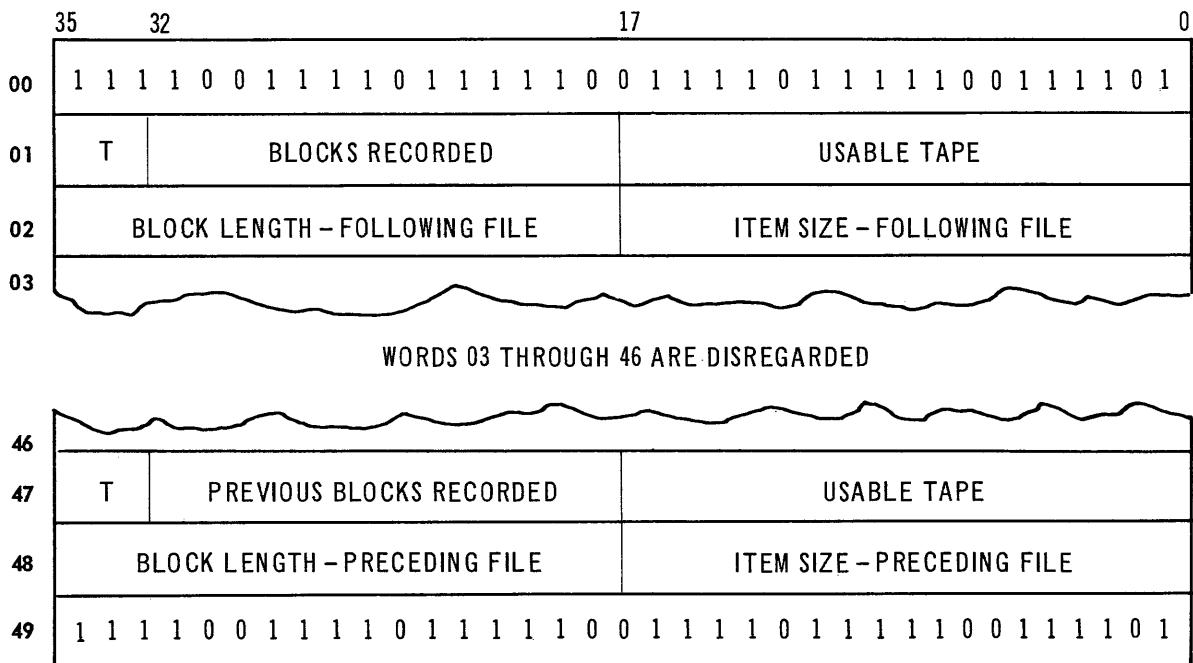


FIGURE 3: Sentinel Block

Words 01 and 47 are used to describe the tape preceding the point at which the sentinel is recorded. The T field, bits 35-33, identifies the type of sentinel block as follows:

T-field	Sentinel Type
000	Header block
000	End-of-File sentinel
010	End-of-Reel sentinel

Bits 32-18 indicate the number of blocks recorded on the tape previous to the sentinel block. The usable tape field, bits 17-00, indicates the amount of tape available on the reel following the data block which precedes the sentinel.

Words 02 and 48 describe the file immediately preceding the sentinel block. Bits 35-18 indicate the maximum block length recorded in the preceding file, and bits 17-00 indicate the size of the items in the preceding file.

B. Usable Tape Field

Each sentinel block contains the 18-bit usable tape field in bits 17-00 of words 01 and 47. This field contains a value which represents the amount of tape still available on the tape. The unit of measurement in the field is a group of eight words. When a sentinel is read, the usable tape figure is removed from the sentinel and shifted left 3 bits (multiplied by 8). The value so generated represents the number of words which may yet be written. Thus, if enough tape is available to write 24000 words, the value entered in the sentinel would be $24000 \div 8 = 3000$, or 5670 in octal.

Each time a block is recorded on the tape, the block word count plus 50 words is subtracted from the amount of usable tape. The 50 words represent interblock space.

C. Header Block

As a result of the Tape Check Subroutine, bits 35-18 of words 01 and 47 and of words 02 and 48 are initially set to zero. Bits 17-00 of words 01 and 47 indicate the total amount of usable tape on the reel. Words 00 and 49 contain the sentinel identification 747574757475 (octal). See Figure 4.

The Header block is not applicable to the magnetic drum.

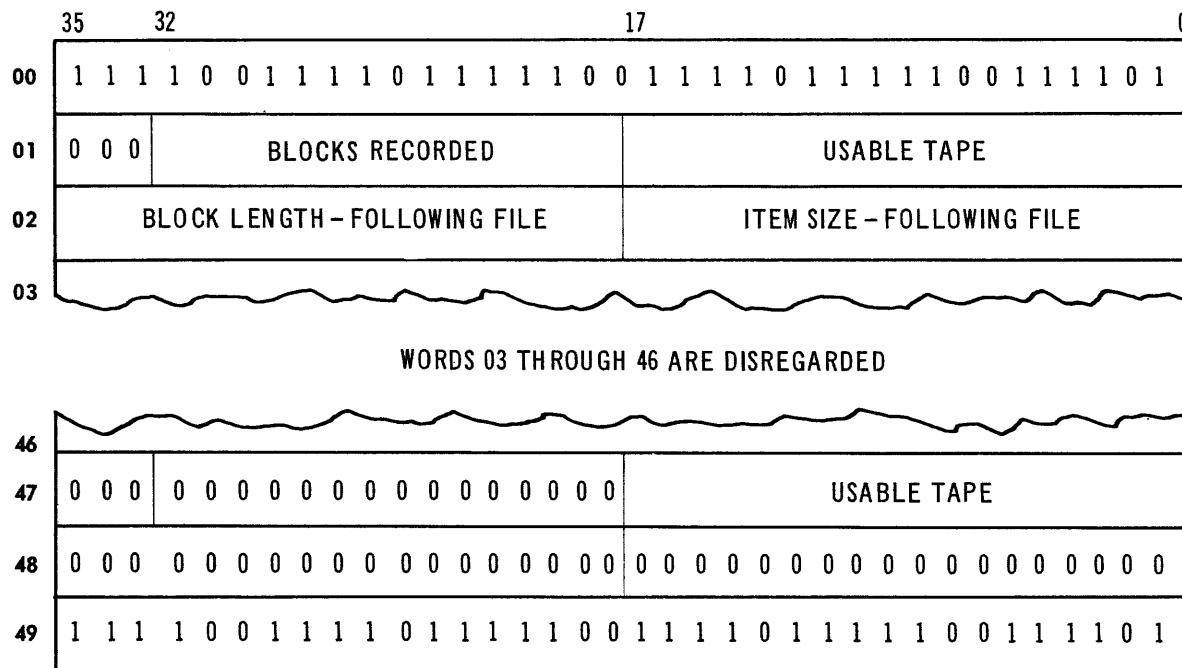


FIGURE 4: Header Block

D. End-of-File Sentinel Block.

All words in the sentinel block are filled in as indicated in figure 1 for tape files. For the magnetic drum the sentinel identifier 747574757475 serves as the End-of-File sentinel. See Figure 5.

	35	32	17	0
00	1 1 1 1 0 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 0 1			
01	0 0 0	BLOCKS RECORDED		USABLE TAPE
02		BLOCK LENGTH - FOLLOWING FILE		ITEM SIZE - FOLLOWING FILE
03		WORDS 03 THROUGH 46 ARE DISREGARDED		
46				
47	0 0 0	PREVIOUS BLOCKS RECORDED		USABLE TAPE
48		BLOCK LENGTH - PRECEDING FILE		ITEM SIZE - PRECEDING FILE
49	1 1 1 1 0 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 0 1			

FIGURE 5: End-of-File Sentinel Block

E. End-of-Reel Sentinel Block.

For tape files, the usable tape field is cleared to zero, and the remainder of the block is filled in as indicated in figure 6.

The End-of-Reel Sentinel is not applicable to the magnetic drum.

	35	32	17	0
00	1 1 1 1 0 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 0 1			
01	0 1 0	0 0	0 0	
02				
03		WORDS 03 THROUGH 46 ARE DISREGARDED		
46				
47	0 1 0	PREVIOUS BLOCKS RECORDED	0 0	
48		BLOCK LENGTH - PRECEDING FILE		ITEM SIZE - PRECEDING FILE
49	1 1 1 1 0 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 0 1			

FIGURE 6: End-of-Reel Sentinel Block

VIII. INTERNAL TAPE FORMAT

A. General Description

The Internal Tape Format provides for multi-reel data files as well as multi-file reels. Label blocks, Data blocks, and Bypass sentinel blocks are provided in this format. The minimum length for a data block is 120 words, and may be any number greater than 120. The length of Label blocks and Bypass sentinel blocks is determined by the length of the data blocks in accordance with the following table:

Data Block Length (words)	Label or Bypass Sentinel Length
$120 \leq \text{length} < 240$	120
$240 \leq \text{length}$	240

Files with Data blocks less than 240 words in length impose certain other restrictions. In case of a write error, the rewrite procedure is not used, and the Write Error Exit is used immediately. The Continue Write (CONWRI) macro-instruction can be used only when the overwritten block is an End-of-File Sentinel.

Output files with data blocks less than 240 words in length cannot replace existing files, except where the existing file is the first file on the tape. Thus, a 120 word Label block may replace an End-of-File sentinel only for the second and following files on a multi-file reel.

None of the above restrictions apply to to tape files with data blocks of 240 words or longer.

B. Label Blocks

A label block is the first block on each reel of tape. A Label block also heads each file on a multi-file reel. When a Label block is recorded on a tape, it replaces a Header block, an End-of-file sentinel block, or another Label block. Thus, the first Label block recorded on a tape replaces either an existing Label block or a Header block.

The format of a Label block is shown in Figure 7.

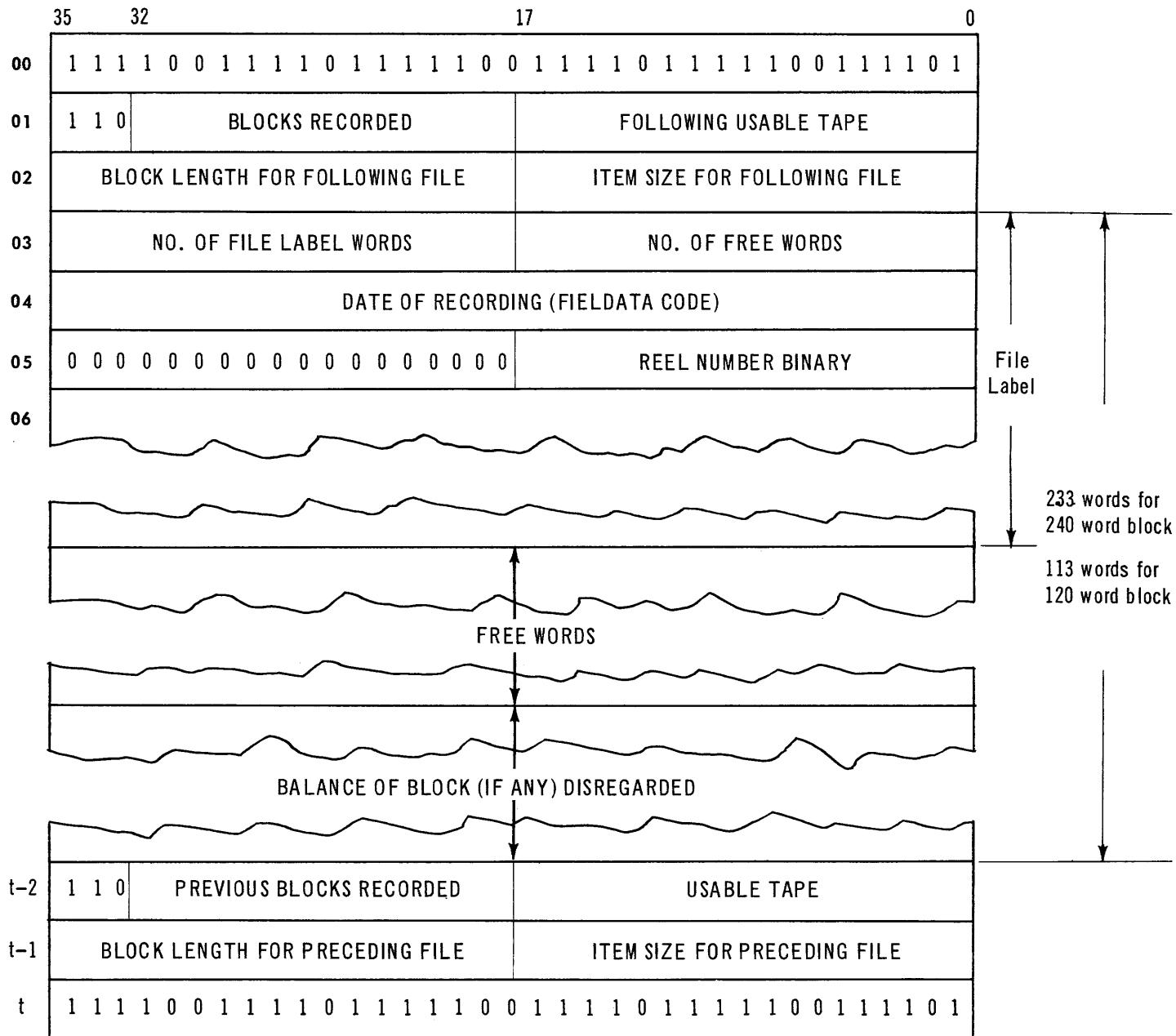


FIGURE 7: Label Block

The first word (00) and the last word (t) contain the sentinel identification 747574757475 (octal). Words 01 and t-2 are tape descriptive words. The bit formation of 110 (binary) in bits 35-33 identify the block as a Label block. Bits 32-18 indicate the number of blocks previously recorded. Bits 17-00 indicate the amount of usable tape; in word 01 the Label block is included as part of the usable tape, while word t-2 indicates the amount of usable tape following the Label block.

Words 02 and t-1 are file descriptive words. Bits 35-18 indicate block lengths, and bits 17-00 indicate the item size. Word 02 describes the file following the Label block, while word t-1 describes the file immediately preceding the Label block. The item size for the appropriate file is cleared to zero for variable length items; for fixed length items and actual item size appears in bits 17-00.

Word 3 describes the parts of the Label Image. Bits 35-18 indicate the number of words in the File Label. Bits 17-00 indicate the number of Free Words.

The File Label is variable in length, and is specified by the user. Word 4 of the Label block is the first word of the File Label, and contains the Date of Recording, in six FIELDATA characters. The second word of the File Label (word 5 of the Label block) contains the reel number, in binary format, in bits 17-00; bits 35-18 are disregarded.

The Free Words section of the Label block is variable in length, and is specified by the user. It may be omitted. The Free Words follow immediately after the File Label. The remainder of the Label block, up to t-3 inclusive, is disregarded.

C. Data Blocks

Data blocks contain one or more data items. Both the block size and the item size may be variable within the same data file. Figure 8 illustrates a variable-length item block, and figure 9 illustrates a fixed-length item block. The two formats are basically identical, the only difference being that in the variable-length item block the first word of each item is an item-count word specifying the length of the preceding item (bits 35-18), and of the following item (bits 17-00). Thus bits 35-18 of the first item-count word, and bits 17-00 of the last item-count word are cleared to zero.

The body of fixed-length item blocks is composed entirely of data words.

In either format, block size is determined by the first word or the last word, which is identical. Bits 35-18 specify the number of items in the block and bits 17-00 specify the number of words in the block.

The minimum length for data blocks is 12 words. However, if the maximum block size in a variable-length block file is equal to or greater than 120 words, the minimum block size recorded on tape will be 240 words.

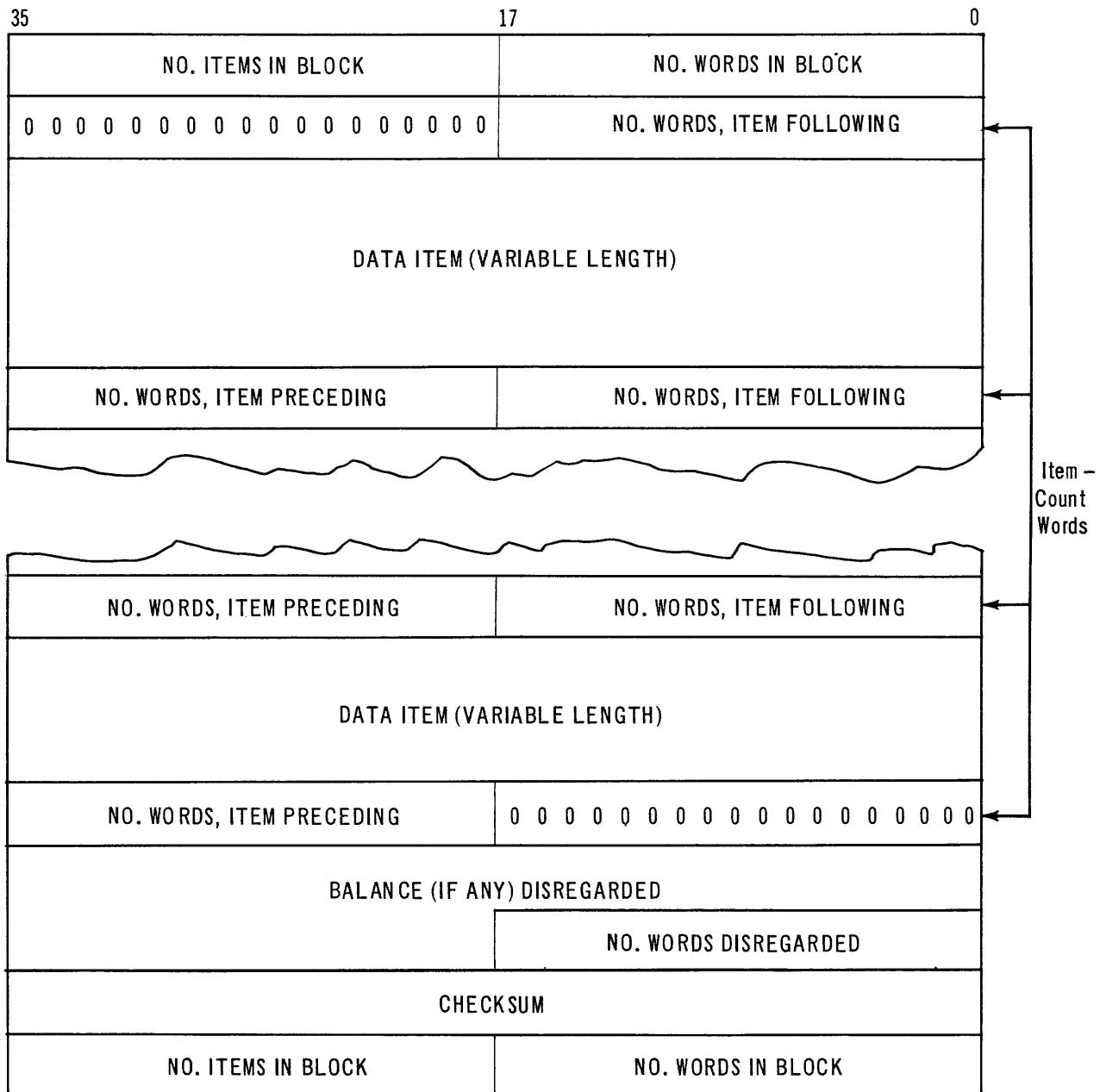


FIGURE 8: Variable-Length Item Data Block, Internal Format

Writing in either fixed or variable-length item size format is specified by a parameter in the Internal Format Output Control (IOCON) macro-instruction. When reading, identification of a block as either a fixed or variable-length item block is determined by bits 17-00, word 02 of the Label block. See Figure 7.

If a block cannot be filled with data items, the balance of the block is disregarded. A count of the number of disregarded words is kept in bits 17-00 of the word preceding the checksum word. Disregarded words are used in calculating the checksum.

35

17

0

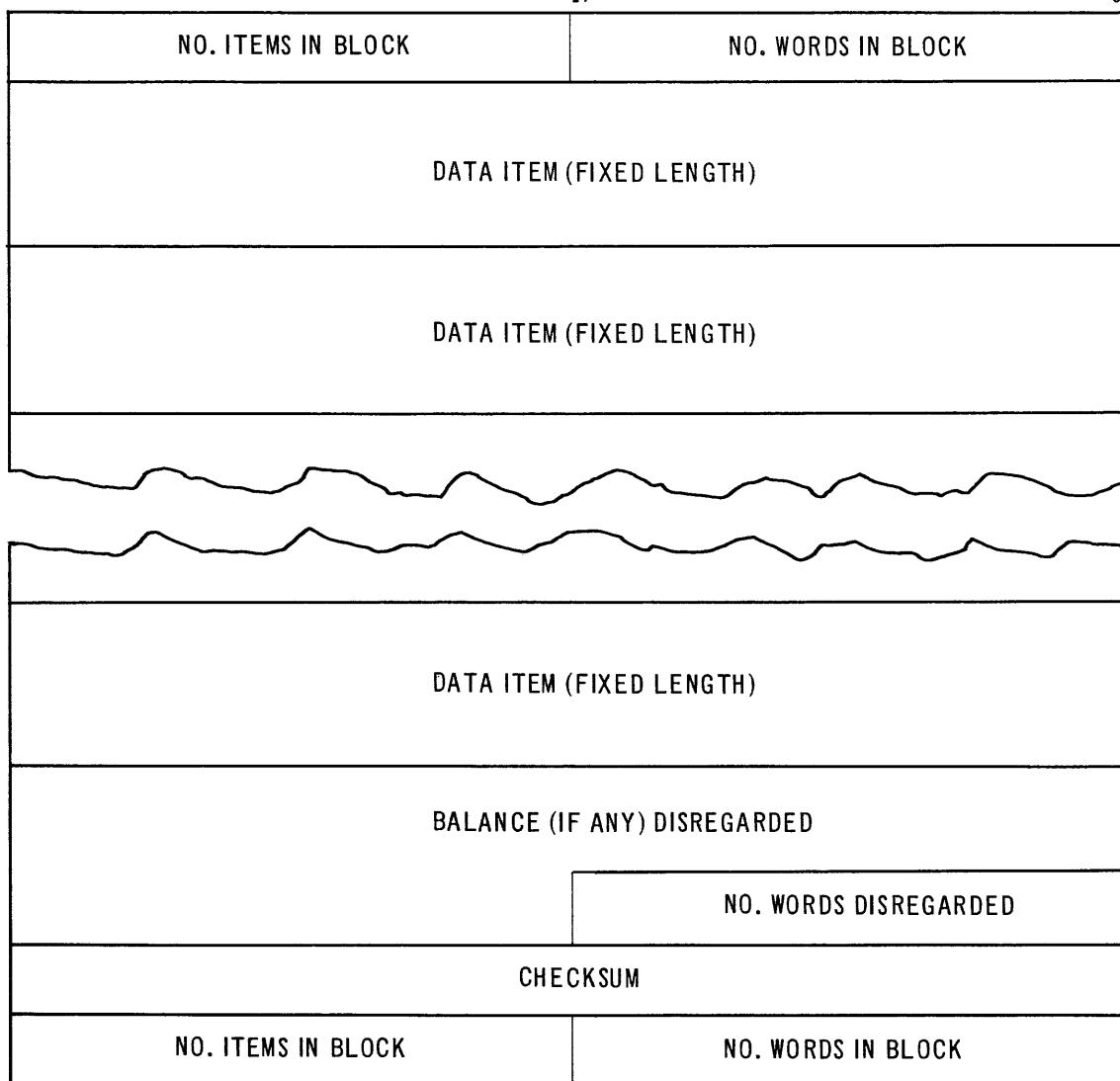


FIGURE 9: Fixed-Length Item Data Block, Internal Format

D. Bypass Sentinel Blocks.

Bypass sentinels are generally used in pairs to surround information blocks which are to be bypassed when reading the tape. The second Bypass sentinel block may be replaced by an End-of-File or End-of-Reel sentinel.

Bypass sentinels are used only on Internal Format tape files. The Bypass sentinel block is illustrated in Figure 10.

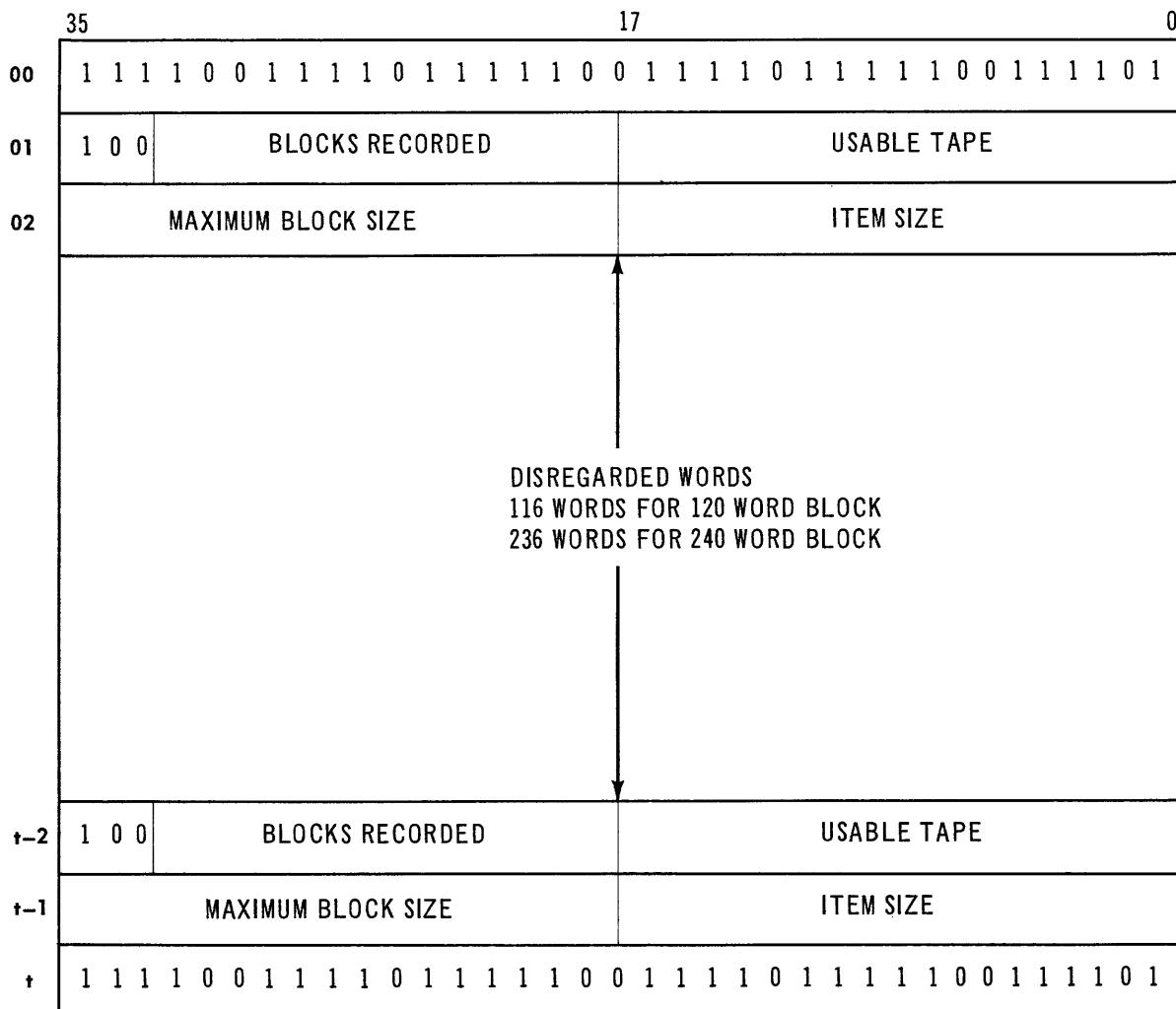


FIGURE 10: Bypass Sentinel Block

Words 00 and t contain the sentinel identification 747574757475 (octal). The words 01 and t-1 describe the file to which the Bypass sentinel refers. Bits 35-18 indicate the maximum block length, and bits 17-00 indicate the item size. These words are the same for both of the Bypass sentinel blocks involved.

Words 02 and t-1 describe the tape. The identifier field, bits 35-33, is set to 100 (binary). Bits 32-18 indicate the number of blocks recorded on the tape, including the sentinel block. Bits 17-00 indicate the amount of usable tape following the sentinel block, and therefore the values entered in this field will be different in the two sentinel blocks.

The Bypass sentinel block is either 120 or 240 words in length. Recording of the Bypass sentinel block is a function of the object program.

IX. EXTERNAL TAPE FORMAT

A. General Description

The External Tape Format provides for multi-reel data files as well as multi-file reels. Header blocks, End-of-File Sentinels and End-of-Reel sentinels as described in Section X are used with this format. Bypass sentinels are never used in the External Tape Format. Label blocks, different in format from the Label blocks for Internal format, are used to identify files. All other blocks are Data blocks.

B. Label Blocks

A Label block is the first block on each reel of tape. A Label block also heads each file on a multi-file reel. The Label block is the same size as the Data blocks in the subject file. It replaces a Header block, an End-of-File sentinel, or another Label block when it is recorded on a tape. Figure 11 illustrates an External Format Label block.

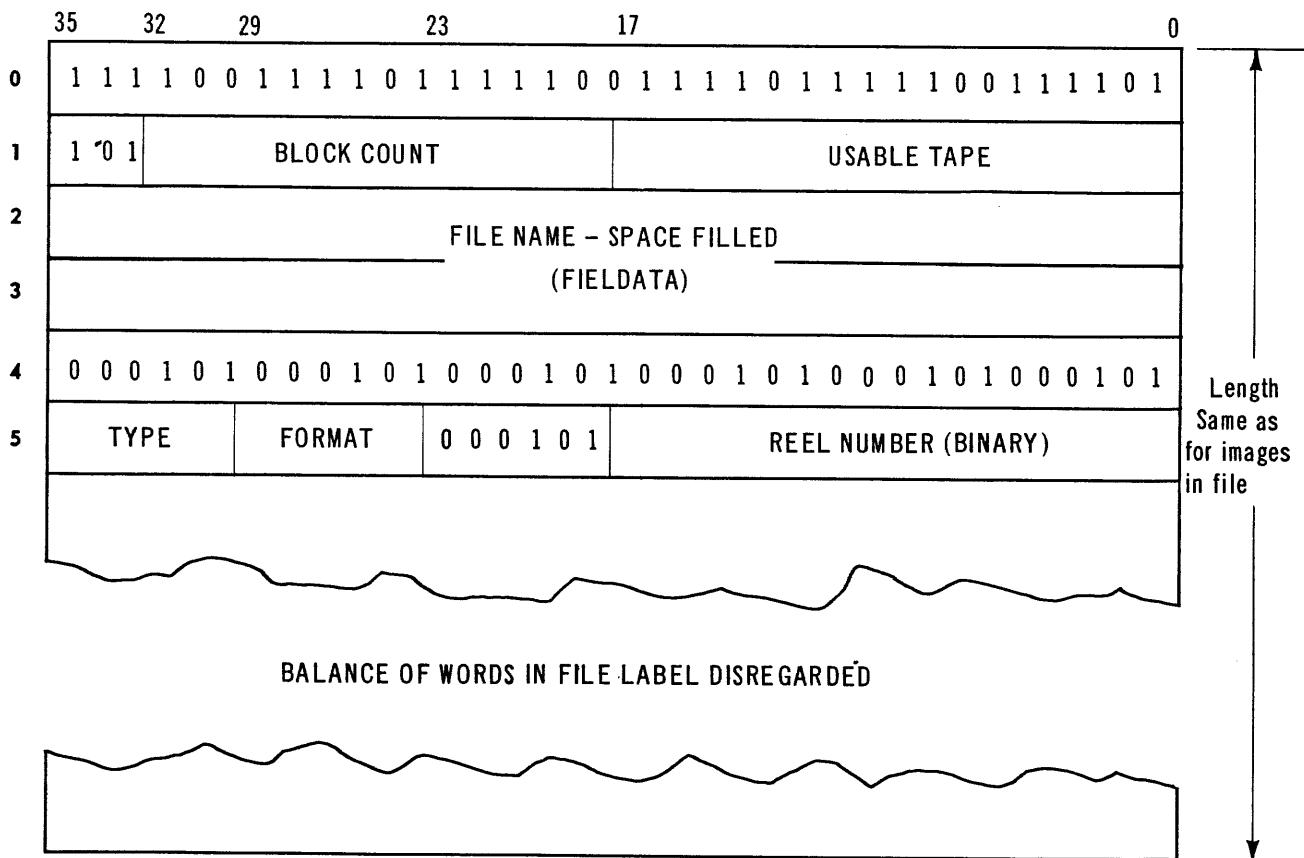


FIGURE 11: File Label, External Format

Word 0 is the sentinel identification 747574757475 (octal). Word 1 describes the tape. Bits 35-33 contain the binary digits 101. Bits 32-18 indicate the number of blocks already recorded on the tape, including the subject Label block. Bits 17-00 indicate the amount of usable tape following the Label block.

Words 2 and 3 contain the File Name. This FIELDATA field is left justified and all unused portions of the field are filled with blank space FIELDATA characters (05....). Word 4 is also filled in with blank space characters. The File Name is not available to the object program and is not checked by the External Format routines, but is primarily for external identification. The file Name comes either from the Start card on input tapes or from the File Name associated with the Open Output File, External Format (EOPOUT) routine on output files.

Word 5 contains three fields. The reel number is in bits 17-00 in binary configuration. Bits 35-30 identify the input/output medium; the entry in this field is an octal 10 (=FIELDATA C) for cards, or an octal 25 (=FIELDATA P) for the printer. Bits 23-18 contain the binary bit configuration 00101. Bits 29-24 identify the format of the file as follows:

Octal	FIELDATA Equivalent	Format
13	F	FIELDATA
27	R	Row-binary
10	C	Column-binary

The rest of the Label block is disregarded.

C. Data Blocks

Data blocks are, in general, composed entirely of words which are images of the desired input or output. The image can be in FIELDATA, row-binary or column-binary format. The length of a Date block is fixed for each type of input or output according to the following table:

Image Type	Format	Words Per Image	Images Per Block	Words Per Block
Card	FIELDATA	14	18	252
Card	Row-binary	36	7	252
Card	Column-binary	27	9	243
Printer	FIELDATA	22	11	242

Since each computer word can contain six FIELDATA characters, a total of 13 full words plus a fraction, or actually 14 words, are required to create an 80-column card image. Similarly, 22 computer words are required to create a 128 character printer image.

The row-binary card image is contained in 36 computer words. For each row, one computer word is required for columns 1-36, one word for columns 37-72, and one word containing 8 bits of data in the most significant bit positions for columns 73-80. Thus a full card image contains 12 of these 3-word groups.

The column-binary card image is contained in 26 computer words plus one word containing 24 bits of data in the most significant bit positions, for a total of 27 words.

A Data block must consist of an integral number of images.

X. DRUM FORMATS

Each drum file under the control of LION is headed by a Label block, and is followed by either another Label block or an End-of-File sentinel. The End-of-File sentinel is the only sentinel used on the drum.

A. End-of-File Sentinel

The End-of-File sentinel consists of the single Sentinel Identification word 747574757475 (octal). This sentinel is used for both Internal and External Drum Formats.

B. Internal Drum Format

A drum file in the Internal Format consists of a Label block, data blocks, and either another Label block or an End-of-File Sentinel.

1. The Label block is variable in length, with a minimum length of five words. The maximum length is limited only by the size of the output buffer. Figure 12 illustrates the Drum Label Block.

The first word, word 00, contains the sentinel, 747574757475 (octal). The last word, t, is filled with binary ones. Word 01 is the descriptor word. Bits 35-32 contain the identifier field, 110. Bits 22-00 indicate the number of words allocated to the drum file.

Word 02 is the file descriptor, Bits 35-18 indicate the maximum block size in the file. Bits 17-00 are used to indicate the item size. For fixed item size files, this field contains the item size. This field is zero for variable item size files.

Word 03 describes the parts of the label image. Bits 35-18 indicate the number of words in the File Label. The number of Free Words is indicated in bits 17-00.

The File Label is variable in length and is specified by the user. The Date of Recording is the first word in the File Label. It is six FIELDATA characters in length, The second word in the File Label is cleared to zero.

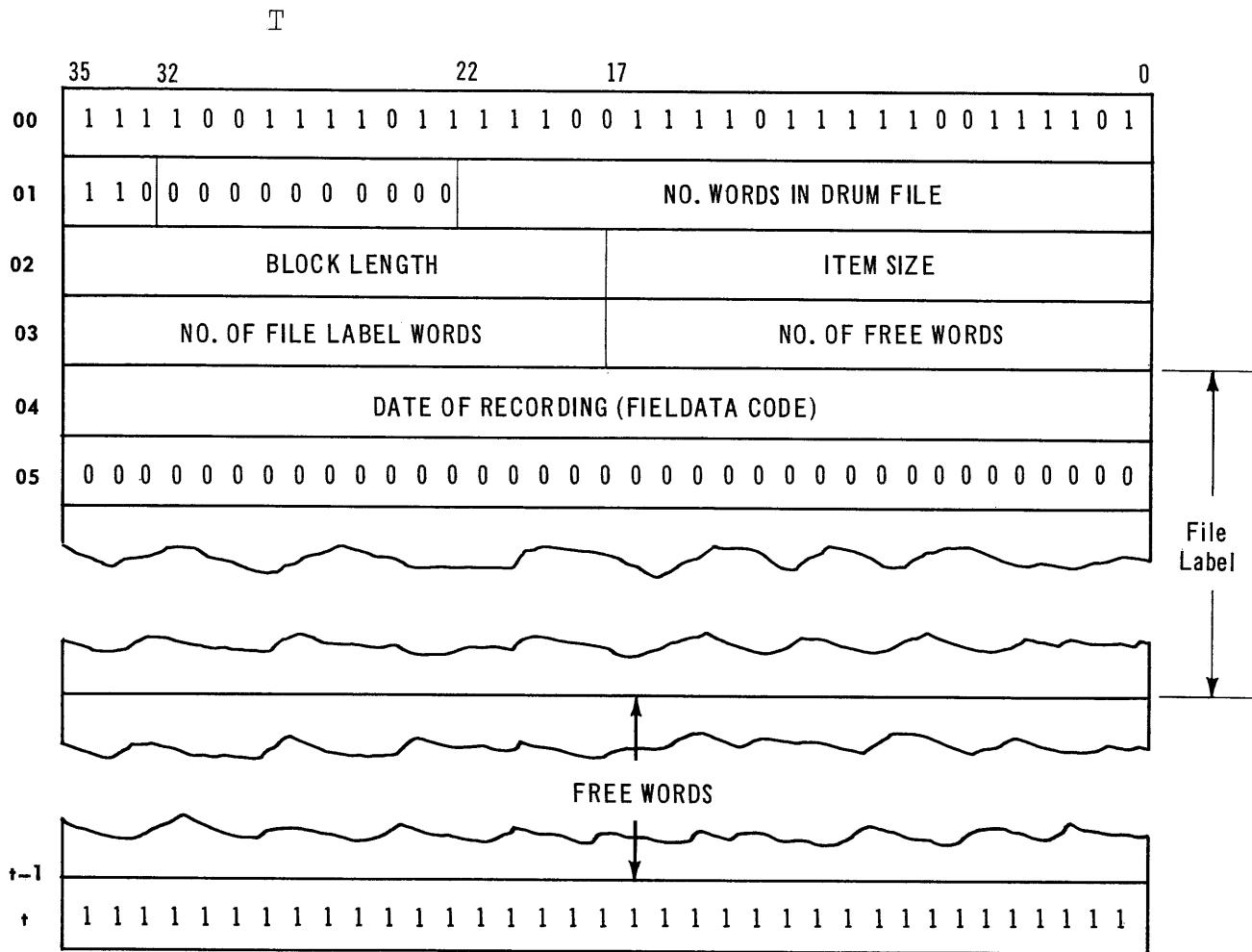


FIGURE 12: Label Block, Drum Internal Format

The contents of the Free Word section, if any, is specified by the user. It is variable in length.

2. Data blocks contain one or more data items. The block size as well as the item size may be variable within the same data file. All data items are contained in an integral number of computer words. Both fixed-length item data blocks and variable-length item data blocks are provided in the Internal Format.

The first and last words of the block are identical. The number of items in the block is in bits 29-18. Bits 17-00 indicate the number of words in the block. The item count is limited to 4095 items per block.

The next to the last word in the block contains the block checksum. The format for variable-length item blocks is shown in Figure 13. Figure 10 illustrates the format for fixed-length item blocks.

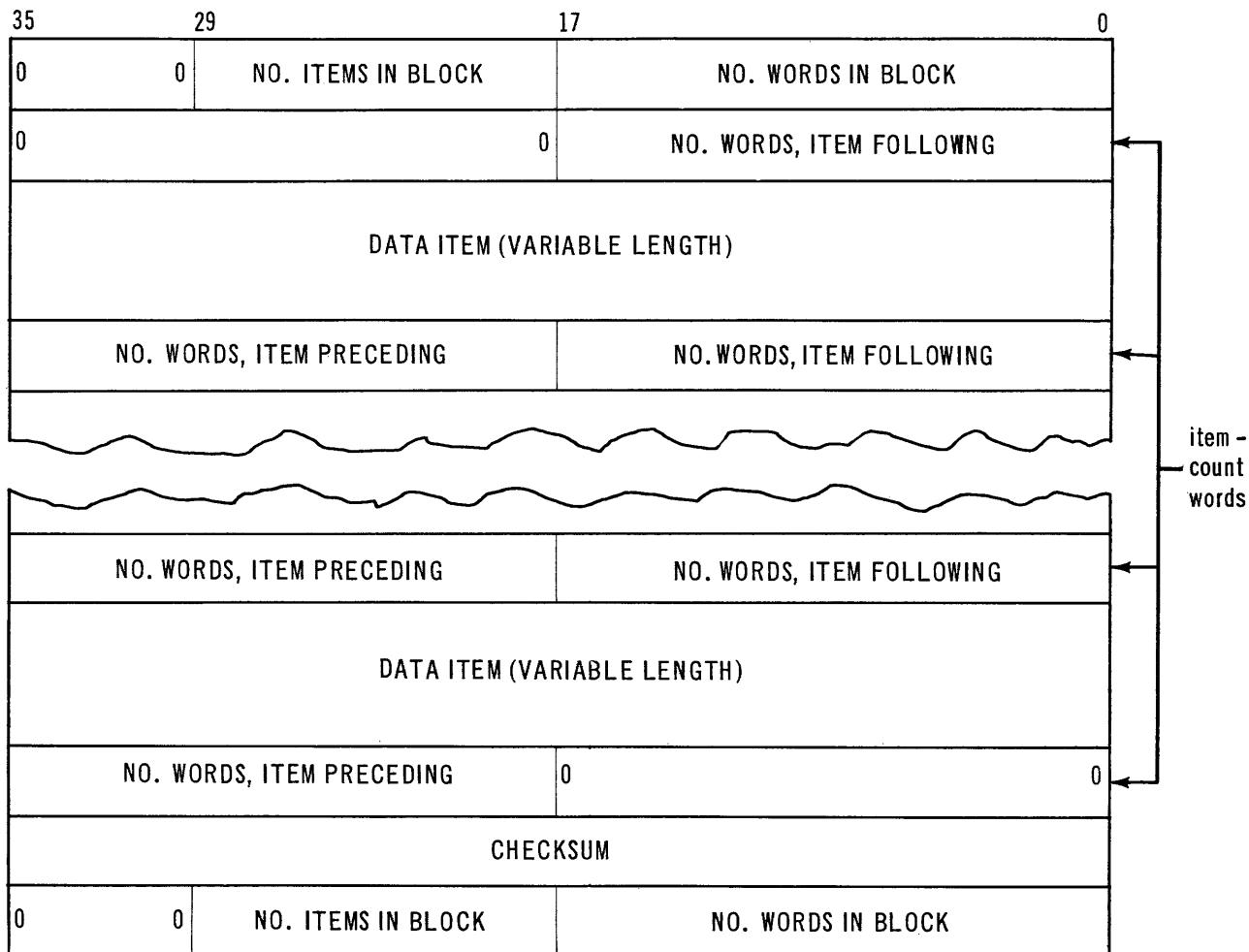


FIGURE 13: Variable-Length Item Data Block, Drum Internal Format

In the variable-length item blocks (Figure 13) the individual items are separated by item count words. The number of words in the preceding item is contained in bits 35-18. Bits 17-00 indicate the number of words in the item following. Hence bits 35-18 in the second word in the block are cleared to zero. Likewise bits 17-00 in the word preceding the checksum are cleared to zero.

The body of fixed-length item blocks, Figure 14, is composed only of data words. Only the first, last, and next to last words in the block contain control information.

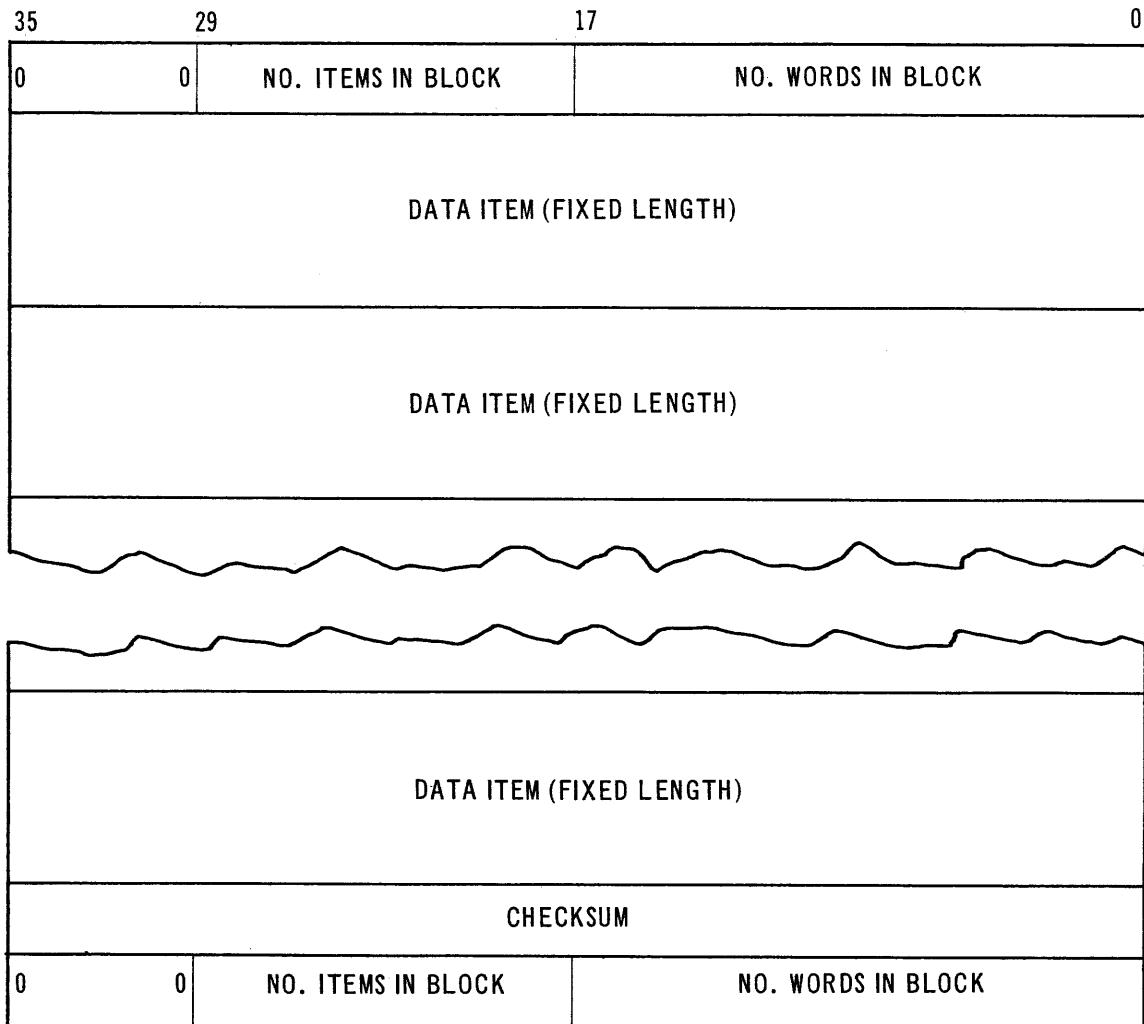


FIGURE 14: Fixed-Length Item Data Block, Drum Internal Format

C. External Drum Format

The External Drum Format is identical to the External Tape Format, with one exception. Unfilled images in the last data block of a file are not recorded on the drum. In the latter case, the image which follows the last image filled by the user program, is filled with the sentinel 747574757475 (octal). This block is then recorded on the drum including the specially filled sentinel image.

XI. CARD FORMATS

This section describes the card formats used in conjunction with the 1107 Input/Output Library. Two specially punched cards are required for each card deck. The Start Card heads each card deck. Each card deck is followed by a Stop Card.

The Stop card is followed by three cards which are disregarded on reading the deck.

A. Start Card

This card is prepared in Hollerith code. It is illustrated in Figure 15. Columns 1 through 6 contain the sentinel `././.`. Column 7 contains an asterisk. Columns 8 through 12 are blank. Columns 13 through 24 contain the card deck name, left justified. Columns 25 through 30 are blank. Column 31 contains the letter C. Column 32 describes the deck format as follows:

Column 32	Format
F	Hollerith deck
R	Row-binary deck
C	Column-binary deck

Columns 37-72 are available for programmer use. On decks which are input to the Service routines, columns 73-78 contain the rewind procedure for the tape being produced as follows:

Columns 73-78	Procedure
REWIND	Rewind without interlock
REWINT	Rewind with interlock
NOREWI	Do not rewind

Columns 73-78 are blank on decks punched by the Service routine. Columns 79 and 80 are always blank.

The 14 computer words containing the image of the Start Card are transferred intact to the tape File Label by the Card-to-Tape routine. Likewise the entire File Label, with the exception of columns 73 through 80, is transferred to the Start Card by the Tape-to-Card routine.

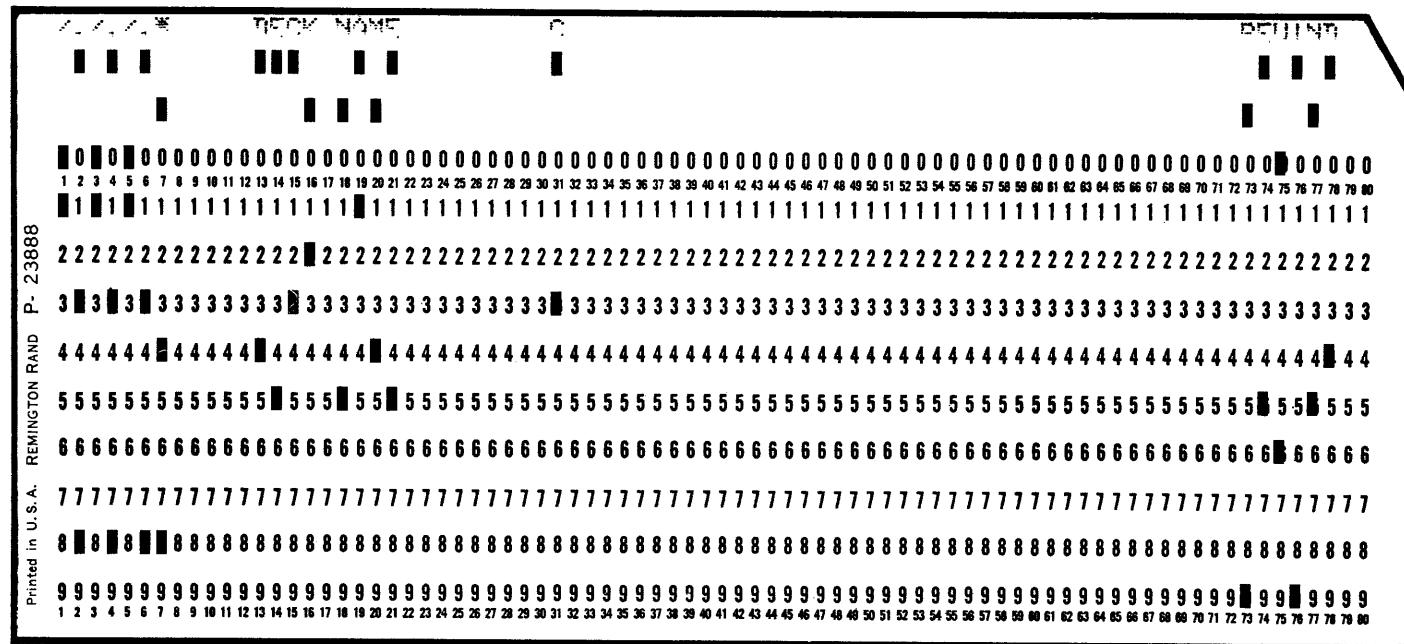


FIGURE 15: Start Card

B. Stop Card

This card is used to signal the end of a card deck. It is illustrated in Figure 16. Columns 1 through 8 and columns 73 through 80 are identical. Each column contains a 12-4-8 punch.

Columns 9 through 36 contain alternate spaces and ampersands. Odd-numbered columns are spaces. Ampersands appear in the even-numbered columns.

Columns 37 through 72 are blank.

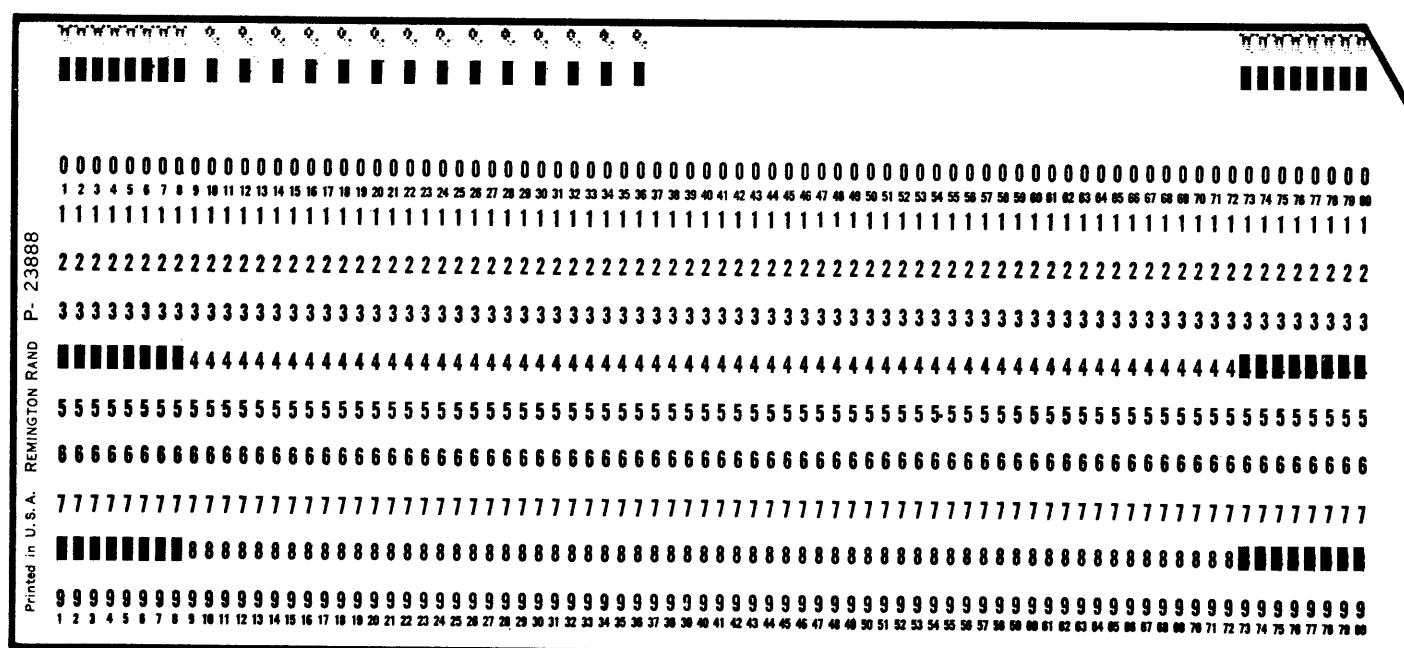


FIGURE 16: Stop Card

XII. PROGRAMMING EXAMPLES

A. Sample Program No. 1: Updating

In this example it is assumed that a Master File is to be updated. For maximum simplicity of the program logic, the Update File will contain the updated item in its entirety, and no updating of fields within an item is required. The Update File will not be used to create new items, therefore each item in the Update File must have a counterpart in the Master File.

Error routines are indicated, but are not programmed in detail. Fixed Length Data Items are assumed, and both input and output are in the Internal Format.

The numbered comments below are to be associated with the reference numbers at the right margin of the coded example, figure 18.

1. For definition of the program and I/O equipment, see the SLEUTH Assembly manual, UT-2574.
2. This line, and the three following lines, define, and place in the DBANK area, four locations which will be used for temporary storage of the data item control words. See reference 7.
3. The I/O control macros, in the DBANK area, will generate the file control routine tables for the input and output files. Note that unneeded parameters are coded as (\emptyset) or (\emptyset , \emptyset).
4. The OPNIN and OPNOUT macros open the input and output files.
5. The LOCFIL macro specifies where, on the TAPE1 reel, the Update File is to be found. As coded here, the third file on TAPE1 should be the desired Update File. JANDM is the address of the error routine to be followed if the File Label images on tape and in core do not agree. Programming of this error routine is not shown in the example.

6. The Update file is read. If the item read is the End-of-File Sentinel, program control is transferred to location WILMAR, which corresponds to B_2 on the flow chart, figure 17.
7. The ITMIN macro automatically stores in register $Q\emptyset$ a control word containing the number of words and core address of the item. From $Q\emptyset$ it is transferred, by programming, to locations ADDRU (17- $\emptyset\emptyset$) and ADDRU+1 (35- $\emptyset\emptyset$). Storing of the right half only in ADDRU is necessary because indirect addressing will be used later in the program. See also reference 2.
8. The Master File is read. The comments above on reading the Update File apply here also. An End-of-File exit (D_2 on the Flow chart) would cause an error print-out if the Update File had not previously reached an End-of-File. In either case, all files are closed and the program is ended.
9. The End-of-File exit at the time a file is read is automatic. If additional End-of-File tests are required, they must be programmed. In this problem the End-of-File exit for the Update File was WILMAR. At this location (B_2 on the flow chart), a flag of six binary bits (77_8) is stored in positions 35-30 of ADDRU. At reference 9, the TNZ instruction tests for the presence of this flag. On the flow chart, this is the test which leads to either E or F.
10. $Q\emptyset$ still contains the control word for the Master File, and since it is the same as parameter pb required for the ITMOUT macro, it is transferred directly to the second instruction generated by the ITMOUT macro, at location WRITEM+1.
11. The item from the Master File is written on the New Master File. Parameter pb, since it is provided as described in reference 10, is coded here as (\emptyset, \emptyset).
12. At location CHATER, which corresponds to E on the flow chart, the contents of the address contained in ADDRM (i.e. indirect addressing) are compared with the contents of the address contained in ADDRU. These words are assumed to be the Key words which identify each item.

13. The control word of the ITMIN macro located at READU had been placed in temporary storage at location ADDRU+1. It is now transferred to WRITEU+1. This step is similar to that at reference 10. An item from the Update File is now written.
14. The comparison of reference 12 is continued to determine whether the identification Key of the Update or Master File is greater. On the flow chart, this is the NO branch of the test at E.
15. If the Update File Item is lower than the Master File Item, an error is indicated, because a missing Master item has been detected. After the error printout, a new Update item is read, and program control is transferred to either location CHATER or WRITEM-1.
16. Close all files.
17. Transfer program control to EXEC.

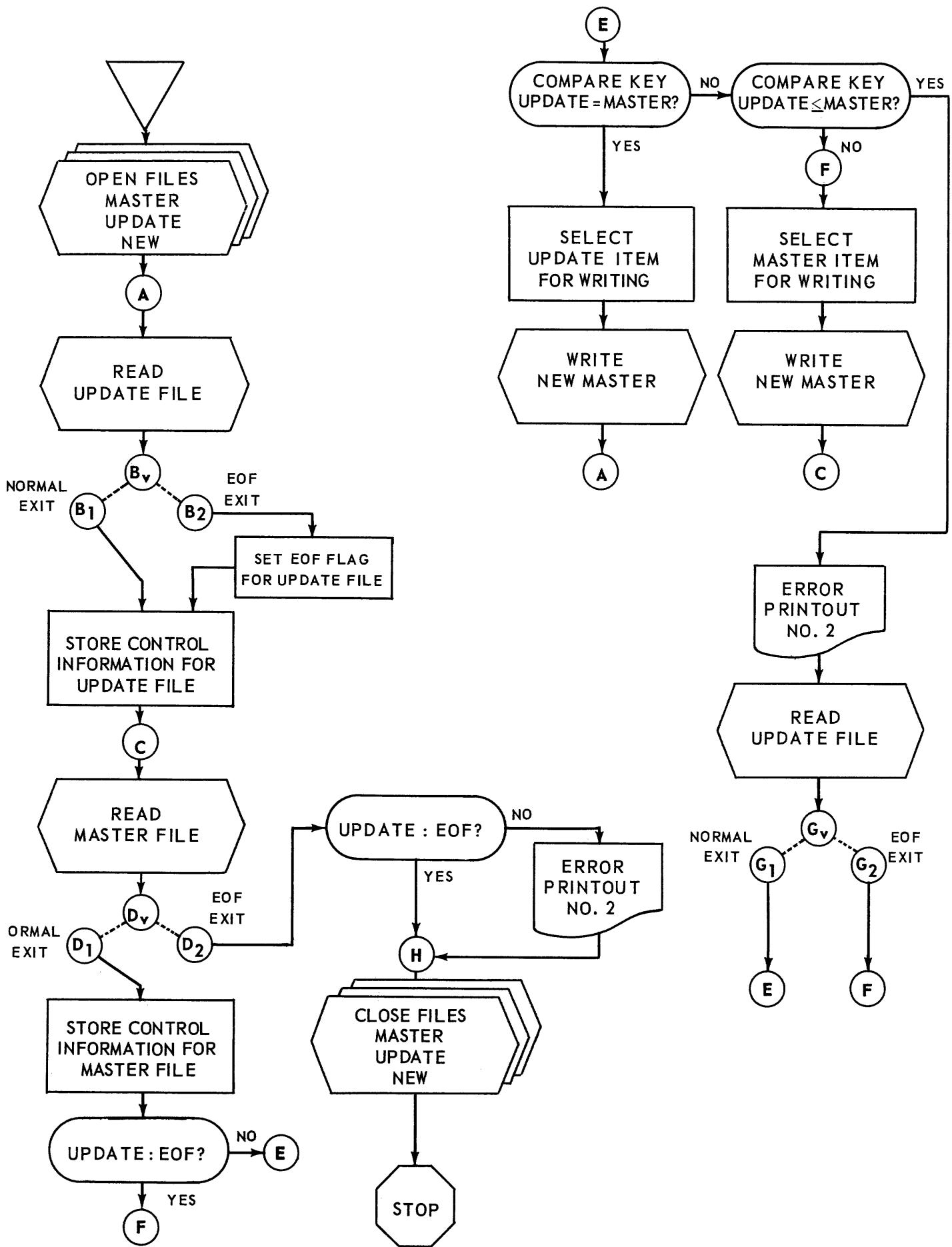


FIGURE 17: Flow Chart, Sample Program No. 1

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37	COMMENTS	REF
	H C H I A T			P R O				EXE	:		1
				M T C H					:		
	T A P E 1			I N M T A					:		
	T A P E 2			I N M T A					:		
	T A P E 3			I N M T A O					:		
				M T C H					:		
	T A P E 4			M T A					:		
	T A P E 5			M T A O					:		
				D B A N K					:		
	A D D R M			W				Ø	:		2
				W				Ø	:		
	A D D R U			W				Ø	:		
				W				Ø	:		
	F R A J O H			I I C O N				(MASTER) (LBLM,6) (FORW) (2,TAPE2) (2,TAPE3) (INBUF2) (INBUF3) (600) (Ø)			3
								(ERSKX1) (HENNX1) (STANX1)	:		
	H E L R I C			I I C O N				(UPDATE)(LBLU,4) (FORW) (2,TAPE1) (Ø,Ø) (INBUF1) (Ø) (900) (Ø) (ERSKX2) (HENNX2) (KOVX2) :			
	E L I T H O			I O C O N				(NEW) (LABLN,6) (Ø,Ø) (2,TAPE4) (2,TAPE5) (OUTBF4) (OUTBF5) (600) (Ø)			
								(ICHX3) (WRITEX) (20) (Ø)	:		
				I B A N K					:		
	O P E N			O P N I N				(MASTER)	:		4
				O P N I N				(UPDATE)	:		
				O P N O U T				(NEW)	:		
				L O C F I L				(TAPE1) (3) (JANDM)	:		5
	R E A D U			I T M I N				(UPDATE) (WILMAR)	:		6
				S T P				\$QØ, ADDRU,, \$H2	:		7
				S T P				\$QØ, ADDRU + 1	:		
	R E A D M			I T M I N				(MASTER) (PHYVIC)	:		8
				S T P				\$QØ, ADDR,, \$H2	:		
				S T P				\$QØ, ADDR + 1	:		
									:		

FIGURE 18

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37	COMMENTS	REF
					T N Z			, ADDRU,, \$S1	:		9
					J U M P			, CHATER	:	ZERO = NO EOF	
					S T P			\$Q0, WRITEM + 1	:	NON-ZERO = EOF	
W R I T E M					I T M O U T			(NEW) (0, 0)	:		10
					J U M P			, READM	:		11
C H A T E R					L D P			\$A15, *ADDRU	:		12
					T E Q			\$A15, *ADDRU	:		
					J U M P			, HARANN	:	NOT EQUAL	
					L D P			\$A15, ADDRU + 1	:	EQUAL	13
					S T P			\$A15, WRITEU + 1	:		
W R I T E U					I T M O U T			(NEW) (0, 0)	:		
					J U M P			, READU	:		
H A R A N N					T L E			\$A15, *ADDRU	:		14
					J U M P			, WRITEM - 1	:	U NOT < OR = M	
					R T J P				:	PRINT ERROR ROUTINE # 2	15
					I T M I N			(UPDATE) (FRANN)	:		
					S T P			\$Q0, ADDRU,, \$H2	:		
					S T P			\$Q0, ADDRU + 1	:		
					J U M P			, CHATER	:		
W I L M A R					L D P			\$A15, \$77,, \$UOP	:	SET EOF FLAG FOR UPDATE FILE	9
					S T P			\$A15, ADDRU,, \$S1	:		
					J U M P			, READM	:		
P H Y V I C					T N Z			, ADDRU,, \$S1	:		
					R T J P				:	ZERO = EOF. ERROR ROUTINE # 1	
					I E N F I L			(MASTER)(REWIND)	:	NON-ZERO = NO EOF	16
					I E N F I L			(UPDATE) (REWIND)	:		
					O E N F I L			(NEW) (REWINT)	:		
					L M J P			\$B1, \$END	:		17
F R A N N					L D P			\$Q0, ADDRU + 1	:		
					J U M P			, WRITEM - 1	:		

FIGURE 18

B. Sample Program No. 2: Listing

Assume a Master File, in Internal Tape Format, containing an identification key, class code, name, City and State, and a dollar amount. The format for each item is shown in figure 19.

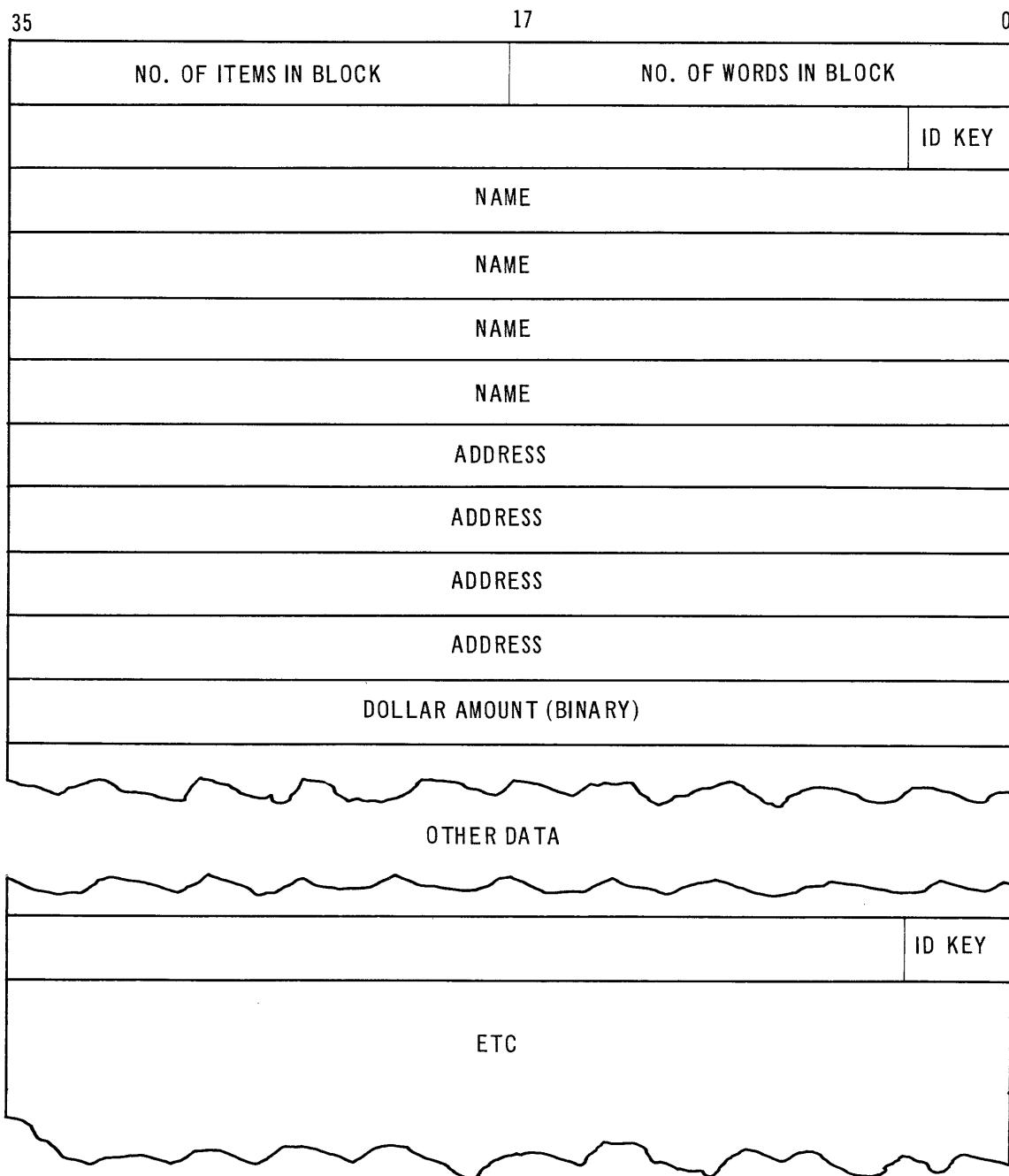


FIGURE 19

The identification key, name, and address are in FIELDATA form, and class and amount are in binary form.

A listing is to be prepared, showing identification, class, name, address, and amount by identification key. Items with zero class coding are to be ignored. Any number of items per key are possible, but only one line per key is to be printed and the amount is to be the total of all items for the subject key. Field headings are to be printed for each page of the listing, and a grand total is to be given at the end. The listing is to be double spaced. Figure 20 illustrates the desired format.

IDENT	CLASS	NAME	CITY STATE	AMOUNT
3348	4	J. JONES	ELMIRA N.Y.	4732.56
7955	12	J. SMITH	WAUKEGAN ILL.	988.52
12483	8	J. BROWN	SCRANTON PA.	13484.12
TOTAL				128415.06*

FIGURE 20

The numbered comments below are to be associated with the reference numbers at the right margin of the coded example, Figure 22. Reference will also be made to the Flow Chart, Figure 21.

1. OUTPAR is the address of the first parameter associated with the IMGOUT macro located at CARMEN + 2.
 - a. Six characters of the word located at the address contained in \$Q0 are transferred to the output buffer. This address is the location of the first word of the item. Since this word, the identification Key, was in FIELDATA format in the input file and was read in Internal Tape Format, it is stored in core in Internal Format. Operation code 5 in the a-field of the instruction will then give a FIELDATA to FIELDATA transfer to the output buffer. The j-field coding (1) means that a maximum of six characters can be formed from a single word, including blank spaces.

- b. The class, in binary format in the input file, will be transferred as a decimal to the output buffer. Class is a six-bit field located in bits $\emptyset 5-\emptyset 0$ of the second word of the item. The second parameter causes it to be listed as an unsigned number, without a decimal point.
 - c. Following the class field, three spaces will appear. Note that the spaces between the Key and class fields are formed naturally, since it is assumed that the class field will never exceed two digits.
- 2. Similarly, the heading and total line parameters are coded to produce the desired effect.
- 3. The SC instructions create the alphabetic constants which are to appear in the output.
- 4. The I/O control macros, in the DBANK area, will generate the file control routine tables for the input and output files. The input file uses the IICON macro for Internal Format, and the EOCON macro is used for the External Format output file. Similarly, the files are opened by the OPNIN and EOPOUT macros.
- 5. Programming for these four error exits is not shown.
- 6. The input item is read. Programming is straightforward and needs no explanation.
- 7. The data line is printed, using the OUTPAR parameters. Similarly, at the appropriate points in the program, the heading line and the grand total line are printed.
- 8. After the total line is printed, both files are closed.
- 9. This line of coding need not appear, as the transfer image will automatically complete the printed line with spaces.

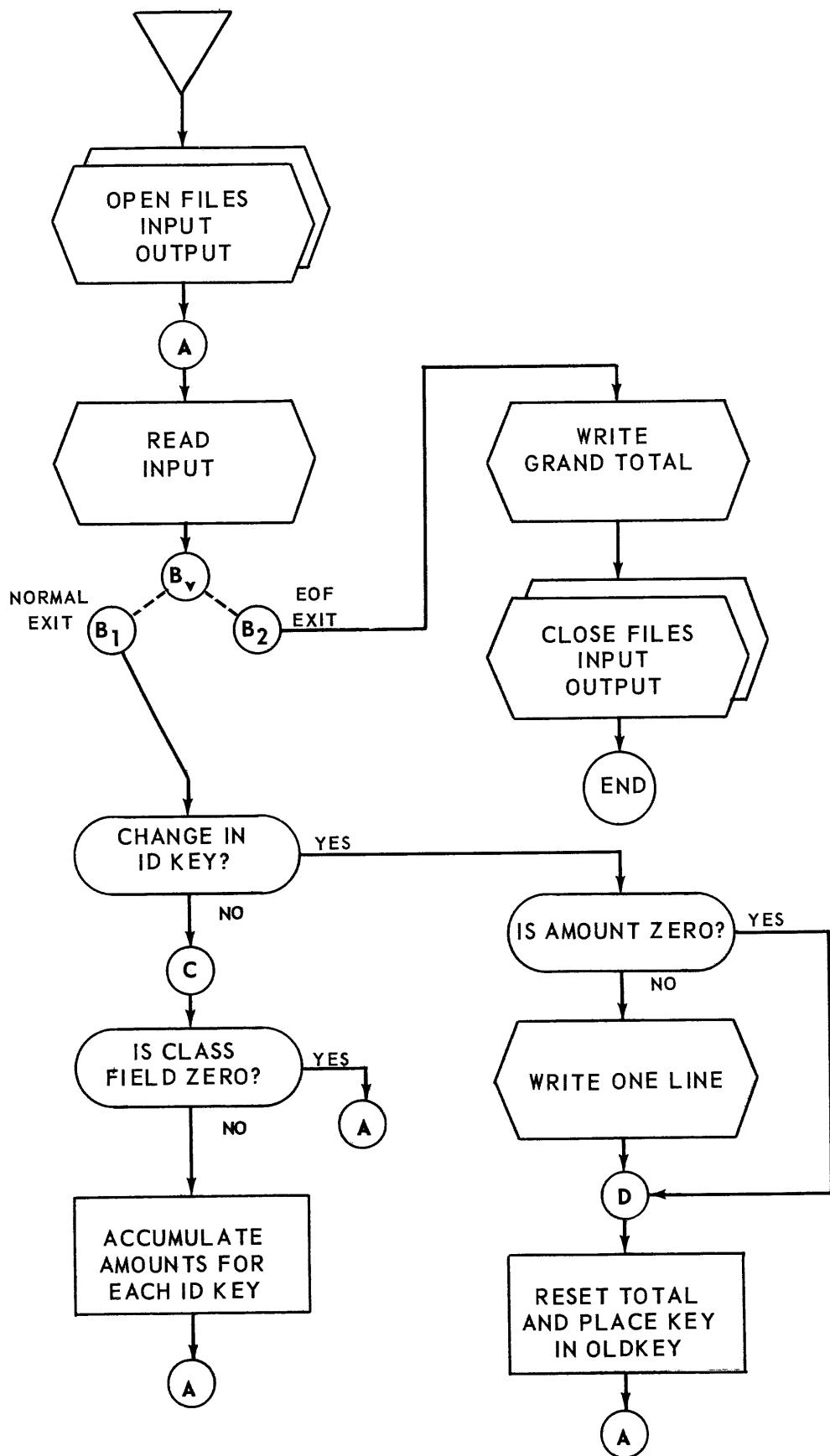


FIGURE 21: Flow Chart, Sample No. 2

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37	COMMENTS	REF
	L I S T N G			P R O				EXE	:		
				M T C H					:		
	I N T A P E			I N M T A					:		
				M T C H					:		
	O U T A P E			M T A					:		
				I B A N K					:		
	S T A R T			O P N I N				(IN)	:		
				E O P O U T				(OUT)	:		
	R E A D			I M G O U T				(OUT) (HEDPAR, 9)	:		
				I T M I N				(IN) (BARBRA)	:		
				S T P				\$QØ, KEY „\$H2	:		
				L D P				\$A6, * OLDKEY	:		
				T E Q				\$A6, * KEY	:		
				J U M P				, CARMEN	:		
				T N Z				, 1, \$QØ, \$S6	:		
				J U M P				, READ	:		
				L D P				\$A2, TOTAL	:		
				A D D				\$A2, 1Ø, \$QØ	:		
				S T P				\$A2, TOTAL	:		
				J U M P				, READ	:		
	C A R M E N			T N Z				, TOTAL	:		
				J U M P				, DELLA + 3	:		
				I M G O U T				(OUT) (OUTPAR, 12)	:		
	D E L L A			L D P				\$A3, TOTAL	:		
				A D D				\$A3, GRTOT	:		
				S T P				\$A3, GRTOT	:		
				S T Z				, TOTAL	:		
				L D P				\$A3, KEY	:		
				S T P				\$A3, OLDKEY	:		
				J U M P				, READ	:		

L10N 80

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37	COMMENTS	REF
	ANNE			IM	G	OUT		(OUT) (HEDPAR,9)	:		7
				J	U	M	P	, DELLA	:		
	BARBRA			IM	G	OUT		(OUT) (TOTPAR,7)	:		7
				I	E	N	F	(IN) (REWIND)	:		8
				E	X	O	E	(OUT) (REWINT)	:		
				L	M	J	P	\$B1, \$END	:		
				D	B	A	N	K	:		
	OUTPAR			6				5,0, \$Q0,1	:		1
				4				3,1, \$Q0	:		
				S				6,6,5,3,0,0	:		
				3				7	:		
				2	4			5,2,\$Q0,1	:		
				3				7	:		
				2	4			5,6,\$Q0,1	:		
				2				7	:		
				1	2			3, TOTAL,,2	:		
				S				27,36,35,0,0,0	:		
				5	0			7	:		9
				0				9	:		
	TOTPAR			4	0			7	:		2
				6				5, LAURA,,1	:		
				2	0			7	:		
				1	2			3, GRTOT,,2	:		
				S				27,36,35,0,0,0	:		
				6				5, STARR,,1	:		
				0				9	:		
	HEDPAR			6				5, MARGIE,,1	:		2
				6				5, NANCY,,1	:		
				1	2			7	:		
				6				5, OPAL,,1	:		

FIGURE 22

FIGURE 22

APPENDIX

Macro Definitions

A. Internal Format

The Internal Output and Internal Input Control macros (IOCON and IICON) generate the file control routine for the Internal Format output and input files. The macro definitions and word layout for these macros is shown in Figures 23 and 24. The generated words normally appear in the DBANK area.

All other Internal Format macros are in the IBANK area. These are shown in Figure 25.

B. External Format

The macro definitions for External Format are similar to those for Internal Format. The control macros EOCON and EICON are illustrated in Figures 26 and 27. The remaining External Format macro definitions are shown in Figure 28.

IICON MACRO DEFINITION				WORD CONTENTS									
WORD NUMBER	TAG	F	SUBFIELD										
00	(pa)	SC	1, (pa) :	File Name (in Fielddata)									
01	H	(pb)	:	Address of label			Number of label words						
02	H	(pc), Ø	:	F = 0, B = 1 ¹			Search	Read	Read Sent.				
03	F	\$, (pd)	:	In. file flag	Channel and unit #1								
04	F	\$, (pe)	:	In. file flag	Channel and unit #2								
05	H	(pj), (pk)	:	End-of-Reel Exit			File Control Error Exit						
06	H	(pl), (pm)	:	Read Error Exit			Label Error Exit						
07	RESV	2	:	Status word									
08				function	Channel and unit number								
09	W	(pf)	:	Number of words to read			Address of buffer #1						
10	RESV	2	:	Final access word									
11				Number of blocks read			Number of blocks to read						
12	W	\$747574757475:		Sentinel word									
13	RESV	2	:	Status word									
14				Read funct.	Channel and unit number								
15	W	(pg)	:	Number of words to read			Address of buffer #2						
16	RESV	3	:	Final access word									
17				Number of blocks read			Number of blocks to read						
18				Sentinel word									
19	H	(ph), Ø	:	Buffer length			Fixed item definition						
20	RESV	6	:	Block count (15 bits) ²			Amount of usable tape (21 bits) ³						
21													
22													
23				Item count									
24				Label tag + 13			Label tag + 7						
25				Label	Last item	CONWRI	End reel	Bypass	End file				
26	W	64	:	Block read	Not filled	No label	EOR Rewind	File end	Reel end				
27	W	Ø	:	Number of blocks used			Current buffer position						

1. F = Forward, B = Backward

2. For tape only; clear for drum

3. 23 bits for drum = number of words remaining

FIGURE 23: Word Layout From IICON Macro

Exec.
packet to
read
buffer 1
and rewind

Exec.
packet to
read
buffer 2

Exec.
packet for
timeouts

IOCON MACRO DEFINITION				WORD CONTENTS												
WORD NUMBER	TAG	F	SUBFIELD													
00	(pa)	S	1, (pa) :	File Name (in Fielddata)												
01		H	(pb) :	Address of label				Number of label words								
02		H	(pc) :	Address of Free Words				Number of Free Words								
03		F	\$, (pd) :	file flag	Channel and unit #1											
04		F	\$, (pe) :	file flag	Channel and unit #2											
05		H	(pj), (pk) :	End-of-Reel Exit				File Control Error Exit								
06		H	(pl), Ø :	Write Error Exit				Initial address current buffer								
07		RESV	2 :	Status word												
08		W	(pf) :	write funct.	Channel and unit number											
09		RESV	2 :	Number of words to write												
10		RESV	2 :	Final access word for read & write												
11		H	64, Ø :	Number of blocks read				(000001) ₈								
12		RESV	2 :	File opened	Header	Bypass EOF	EOR Rewind	End-of-Reel								
13		RESV	2 :	Status word												
14		W	(pg) :	function	Channel and unit number											
15		RESV	3 :	Number of words to write				Address of buffer #2								
16		RESV	3 :	Final access word for write												
17		H	(ph), (pm) :													
18		W	(pn) :	Buffer length				Current address of buffer								
19		RESV	4 :	Block count (15 bits) ⁴				Fixed item definition								
20		RESV	4 :					Amount of usable tape (21 bits) ⁵								
21		RESV	4 :													
22		RESV	4 :													
23		RESV	4 :													
24		RESV	4 :	Label tag + 13				Label tag + 7								

3. For tape only, cleared for drum

5. 23 bits for drum - number of words remaining

FIGURE 24: Word Layout From IOCON Macro

Exec. packet
 write buffer
 #1; write all
 sentinels, read
 write header

Exec. packet
 write buffer
 #2; rewind alt.
 tape on EOF

Exec. packet
for typeouts

FUNCTION	MACRO DEFINITIONS		
	TAG	F	SUBFIELD
Open output file	OPNOUT	MACRO LMJP H ENDMAC	1, IOSAV1 (pa), OPNOUT
Item for output file	ITMOUT	MACRO LMJP H H ENDMAC	1, IOSAV2 (pb) (pa), ITMOUT
Close reel on output file	OENREL	MACRO LMJP SC H ENDMAC	1, IOSAV2 1, (pb) (pa), OENREL
Close file on output file	OENFIL	MACRO LMJP SC H ENDMAC	1, IOSAV2 1, (pb) (pa), OENFIL
Prepare for bypass information	BYPASS	MACRO LMJP H ENDMAC	1, IOSAV1 (pa), BYPASS
Open input file	OPNIN	MACRO LMJP H ENDMAC	1, IOSAV1 (pa), OPNIN
Item for input file	ITMIN	MACRO LMJP H H ENDMAC	1, IOSAV2 0, (pb) (pa), ITMIN
Close reel on input file	IENREL	MACRO LMJP SC H ENDMAC	1, IOSAV2 1, (pb) (pa), IENREL
Close file on input file	IENFIL	MACRO LMJP SC H ENDMAC	1, IOSAV2 1, (pb) (pa), IENFIL

FIGURE 25: IBANK Macros for Internal Format

EICON MACRO DEFINITION				WORD CONTENTS												
WORD NUMBER	TAG	F	SUBFIELD													
00	(pa)	SC	1, (pa)	File Name (in Fielddata)												
01		H	Ø, Ø	Current image address												
02		G	Ø/6, Ø/12, Ø/18	0 0 0 0 0 0	Char. posit.	Final image	Search	Read	Read sent.							
03		F	\$4, (pb)	File flag												
04		F	\$4, (pc)	File flag												
05		H	Ø, (pd)	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	File Control Error Exit									
06		H	(pe), Ø	Read Error Exit												
07		RESV	2	Status word												
08		W	(pa) + 31	function	Channel and unit number											
09		RESV	5	Number of words read												
10	W	RESV	10	Address of buffer #1												
11		Final access word														
12		Number of blocks read						Number of blocks to read								
13		Sentinel word														
14		Status word														
15		function						Channel and unit number								
16		Number of words to read						Address of buffer #2								
17		Final access word														
18		Number of blocks read						Number of blocks to read								
19		Sentinel word														
20	RESV	Buffer length						images/block	words/image							
21		Block count (15 bits) ⁶						image count								
22		Amount of usable tape (21 bits) ⁷														
23																
24		Label tag + 13						Label tag + 7								
25																
26		W	64													
27		RESV	510	First address of current image												
28																
29		Char./image						char. count								
30																

6. For tape only, cleared for drum

7. 23 bits for drum - number of words remaining

FIGURE 26: Word Layout From EICON Macro

Exec. packet to read buffer #1

Exec. packet to read buffer #2

Exec. packet for timeout

8. For tape only, cleared for drum
9. 23 bits for drum - number of words remaining

FIGURE 27: Word Layout From EOCON Macro

FUNCTION	MACRO DEFINITIONS		
	TAG	F	SUBFIELD
Open input file	EOPIN	MACRO LMJP H ENDMAC	1, IOSAV1 (pa), EOPIN
Image read	IMGIN	MACRO LMJP H H H ENDMAC	1, IOSAV3 (pb) 0, (pc) (pa), IMGIN
Close input file	EXIEND	MACRO LMJP SC H ENDMAC	1, IOSAV2 1, (pb) (pa), EXIEND
Open output file	EOPOUT	MACRO LMJP H ENDMAC	1, IOSAV1 (pa), EOPOUT
Image write	IMGOUT	MACRO LMJP H H ENDMAC	1, IOSAV2 (pb) (pa), IMGOUT
Close output file	EXOEND	MACRO LMJP SC H ENDMAC	1, IOSAV2 1, (pb) (pa), EXOEND

FIGURE 28: IBANK Macros for External Format

INDEX

A

Auxiliary Subroutines, 43

B

BYPASS, 12

Bypass Sentinel Block, 58

C

Card Formats, 67

 Start Card, 67

 Stop Card, 68

Card Punch, 43

Card Read, 43

Card-to-Tape Conversion, 44

Close File, External Format Input, 41

 External Format Output, 31

 Internal Format Input, 19

 Internal Format Output, 11

Close Reel, Internal Format Input, 18

 Internal Format Output, 10

Continue Write, 47

Control, External Format Input, 33

 External Format Output, 22

 Internal Format Input, 13

 Internal Format Output, 5

CONWRI, 47

D

Data Block, External Drum Format, 66

 Internal Drum Format, 63

 External Tape Format, 61

 Internal Tape Format, 56

Drum Formats, 63

 End-of-File Sentinel, 63

 External Format, 66

 Internal Format, 63

E

EICON, 33

End-Of-File Sentinel, Magnetic Drum, 63

End-Of-File Sentinel, Magnetic Tape, 53

End-Of-Reel Procedure, External Format Input, 41

 External Format Output, 31

 Internal Format Input, 17

 Internal Format Output, 9

End-of-Reel Sentinel, 53

EOCON, 24
EOPIN, 34
EOPOUT, 25
EXIEND, 41
EXOEND, 31
External Drum Format, 66
External Format Subroutines, 21
External Format Input Subroutines 32
 Close File, 41
 Control, 33
 End-of-Reel Procedure, 41
 Image Read, 35
 Open File, 33,34
External Format Output Subroutines, 21
 Close File, 31
 Control, 22
 End-of-Reel Procedure, 31
 Image Write, 25
 Open File, 22,25
External Tape Formats, 60
 Data Blocks, 61
 Label Blocks, 60

H

Header Block, 52

I

IENFIL, 19
IENREL, 18
IICON, 13
Image Read, External Format, 35
Image Write, External Format, 25
IMGIN, 35
IMGOUT, 25
Input Subroutines, External Format, 32
Input Subroutines, Internal Format, 13
Internal Format Input Subroutines, 13
 Close File, 19
 Close Reel, 18
 Control, 13
 End-of-Reel Procedure, 17
 Open File, 13
 Read, 16
Internal Format, Magnetic Drum, 63
 Data Block, 64
 Label Block, 63
Internal Format, Magnetic Tape, 54
 Data Block, 56
 Label Block, 54
 Bypass Sentinel Block, 58

Internal Format Output Subroutines, 5
Close File, 11
Close Reel, 10
Control, 5
End-of-Reel Procedure, 9
Open File, 8
Write, 8
Internal Format Subroutines, 5, 13
IOCON, 5
ITMIN, 16
ITMOUT, 8

L

Label Block, External Tape Format, 60
Internal Tape Format, 54
Locate File, 48
LOCFIL, 48

M

Magnetic Drum Formats, 63
External Formats, 66
Internal Formats, 63
End-of-File Sentinel, 63
Magnetic Tape Formats, 50
External Formats, 60
Internal Formats, 54
Sentinel Blocks, 50

O

OENFIL, 11
OENREL, 10
Open File, External Format Input, 34
External Format Output, 22, 25
Internal Format Input, 13
Internal Format Output, 8
OPNIN, 16
OPNOUT, 8
Output Subroutines, External Format, 21
Internal Format, 5

P

Print Subroutines, External Format, 43
Programming Examples, 69, 75

R

Read, Internal Format, 16
Read, Image, External Format, 35

S

Sample Programs, 69, 75
Secondary Input/Output Subroutines, 43
Sentinel Block Formats, 50
Service Routines, 44
Special Functions, 47
Start Card, 67
Stop Card, 68

T

Tape Check Routine, 49
Tape-to-Card Conversion, 45
Tape-to-Printer Conversion, 46

U

Usable Tape Field, 52

W

Write, Internal Format, 8
Write, Image, External Format, 25

UNIVAC

DIVISION OF SPERRY RAND CORPORATION