PHILCO
A SUBSIDIARY OF Ford Motor Company
COMPUTER DIVISION

ELECTRONIC DATA PROCESSING SYSTEMS

PHILCO 2000

SORT SYSTEM

TM-21

PHILCO 2000
SORT SYSTEM

February 1963

PHILCO CORPORATION

A SUBSIDIARY OF Ford Motor Company

Computer Division • 3900 Welsh Road

Willow Grove, Pennsylvania

PREFACE

This manual describes the use of the Philco 2000 Sort System, and includes tape format, parameters, examples and error recovery procedures.

The user should be familiar with basic sort techniques, with programming on Philco 2000 Electronic Data Processing Systems, and with Philco's Translator-Assembler-Compiler (TAC).

The Sort System is not a revision of the Philco Sort Generator released in November, 1961 (Philco Manual TM-17), but is a completely new system of major subroutines and a generator, designed to provide a wide range of sort capabilities.

Because the present sort system represents an advancement in technique beyond the original Sort Generator, it is recommended that the new sort be used.

CONTENTS

CONTENTS (Continued)

SECTION I

GENERAL DESCRIPTION

The Philco 2000 Sort System is a combination of TAC subroutines and a generator, designed to provide programmers with a versatile, convenient tool for producing a wide range of sort programs for Philco 2000 Electronic Data Processing Systems.

SORT SYSTEM FEATURES

The following features are incorporated into the Philco 2000 Sort System.

- Items of any size from one to 3840 words can be processed.
- Any number of keys from one to 255 may be specified.
- Wide choice of input and output data formats is permitted without supplementary own coding.
- Backward read capabilities of Philco 2000 Computer Systems eliminate tape rewinding time during merge passes.
- Supplementary coding may be included in first and last passes and prior to each merge pass.
- Modular construction of the sort system facilitates installation replacement of subroutines.
- Programmer option is provided for supplementing or overriding compile time parameters with new parameters at run time.
- Packing of items on working tapes is automatically optimized at run time.
- String lengths are maximized by allocating core memory at run time.
- Two or more independent sort calls may be given within a single program.
- Replacement-Selection technique is used.

SORT OPERATION

The Philco 2000 Sort System provides a generated sort consisting of (1) an initial sorting pass (prepass, or first pass) that performs the internal sort described below, and (2) subsequent merge passes that merge the strings produced by the internal sort until one single sorted string is produced.

Provision is made for inclusion of Own Code in the sort pass, prior to each merge pass, and in the final merge pass.

Internal Sort

Internal sorting is performed by the replacement-selection method (otherwise known as the tournament or tree method), which provides the following advantages:

- Internal processing time is reduced to a minimum.*
- There are no inherent limits on string length, although the combined length of items to be sorted may not exceed the length of a single merge tape. A string extends until no item in the working area can continue it.**

Merge

The power of the merge is determined at run time by the number of working tapes available. As few as three merge tapes and as many as 16 may be specifed. Half the total number of tapes available are used for ascending strings, while the remainder of the tapes are for descending strings. If the total number of tapes is odd, one more tape is used to hold ascending strings than to hold descending strings. The input and output tape(s) may be used as merge tapes.

An optimizing routine automatically selects the appropriate size for groups of items and tape assignments in order to reduce non-data tape areas to a minimum. Moreover, if this routine determines that the sort cannot be accomplished as described by the programmer (because of insufficient memory space for input or output working tape buffer areas), the routine automatically reduces the number of working tapes, and hence the power of the merge, to the point at which the sort can be accomplished.

MEMORY ALLOCATION

Working memory is automatically assigned throughout sort operations by an allocation subroutine. The subroutine assigns portions of memory, accepts their return from other subroutines, and maintains a continuous record of available memory. At the conclusion of the sort, all working memory used by the sort system is released for programmer use.

SORT PARAMETERS

Sort parameters, selected by the programmer, specify the type of sort program to be produced by the sort system. Ten of 16 possible parameters within the sort system describe the particular arrangement of input data and specify the desired arrangement

---

* As determined in the following manner: If n items can be contained in the memory work area, a new item may be collated among n-1 other items, and an output item may be selected after $1 + [\log_2 n]$ comparisons.

** With randomly-ordered input, the average string length is $2n$.

and format of output data. The remaining six parameters are links by which the sort system calls upon programmer's own subroutines.

Prior to the start of operations, the sort system creates a parameter table for storage of parameter values. Information provided by the various combinations of the 16 parameters at compile time is stored in the Base Segment of this table. If the programmer does not supply information for certain parameters, the sort system will insert preset values for them into the segment.

The programmer may also override or add additional parameters to his initial program specifications submitted at compile time by providing new parameters at run time. The new information is placed into the parameter table in one or more supplementary parameter segments. Run time parameters supersede equivalent compile time parameters.

If the same sort call is issued several times within a program, the sort will be automatically reset to the information specified by the Base Segment. If run time parameter information is also required, the supplementary table segments must be set each time the sort is called.

COMPATIBILITY

The generator supplied with the sort system also accepts legitimate calls for the Philco 2000 Sort Generator, released November, 1961 (Philco Manual TM-17). It is intended, however, that the new sort system obsolete the 1961 Sort Generator described therein.

ERROR RECOVERY

In general, errors that are detected by the sort system may be classified into two groups — those from which recovery may be made by transferring control to an earlier phase of the sort, and those from which recovery is not possible. In those cases where recovery is possible, an option is given that permits the operator to attempt recovery or to transfer control to an error exit.

SECTION II

INPUT-OUTPUT TAPE FORMAT

The input-output tape format used by the Sort System consists of a label block, data items, two sentinels and a checksum. Any of these may be absent.

The general input-output tape format is illustrated as follows:

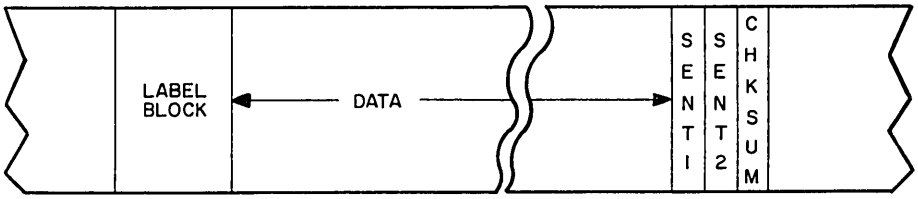


Figure 2–A. General Input–Output Tape Format

LABEL BLOCK

The label block processed by the sort is a 128-word block whose first word contains a label identifying the data that follows. During label checking by the sort, the label specified by the programmer is compared with the label within the first word of the label block; if the two are not equal, an error will be indicated. If label checking is to be performed by own code rather than by the sort, any label processing procedure may be used. Because label checking is optional, the label block may be omitted.

DATA GROUPS

The tape containing the data to be sorted, together with terminating sentinels and checksum, is considered by the sort to be divided into two types of alternating areas — effective and residue. The effective area, which contains the actual data to be sorted, is called a group. Groups range in size from one to 3840 words, and always begin at the first word of a physical block on tape. The residue area that follows a group contains from zero to 127 non-data words used to fill the last physical block associated with the preceding group. The first word of the next physical block is then the beginning of a new group of effective words. Within a given file, all groups are the same size and hence are always separated by the same number of residue words.

Thus, the sort processes only the data that appears within the effective area of tape as illustrated below:

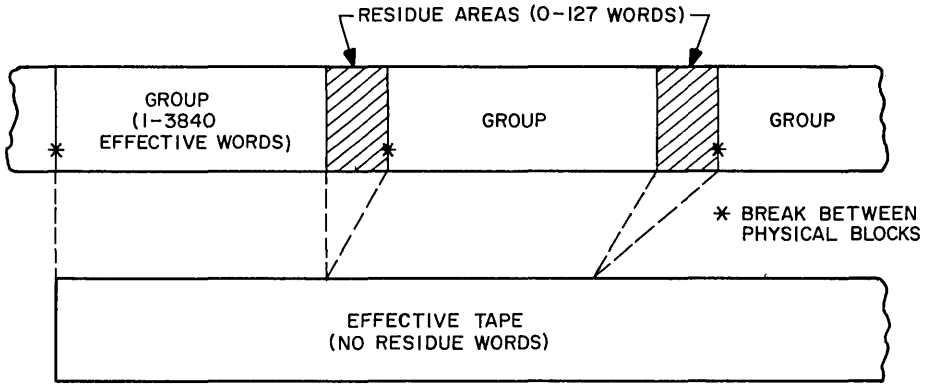


Figure 2-B. Effective Area of Tape

Residue words on an input tape are never processed by the sort, but are written on the output tapes as words of filler characters (32_8).

DATA ITEMS

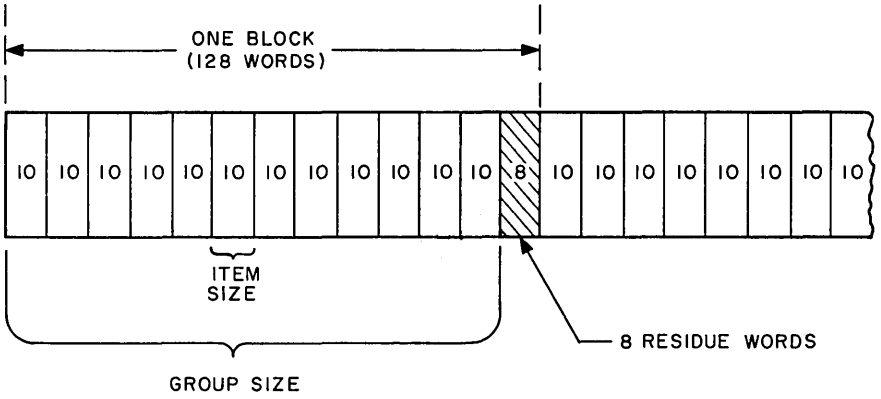
The data to be sorted is arranged in uniformly-sized units called items, each of which completely describes one member of a class of data. Multiple items of the same class comprise a file.

Items appear on tape only in those effective areas specified by groups. Items may be from one to 3840 words in size, and may be smaller than, equal to, or larger than their associated groups.

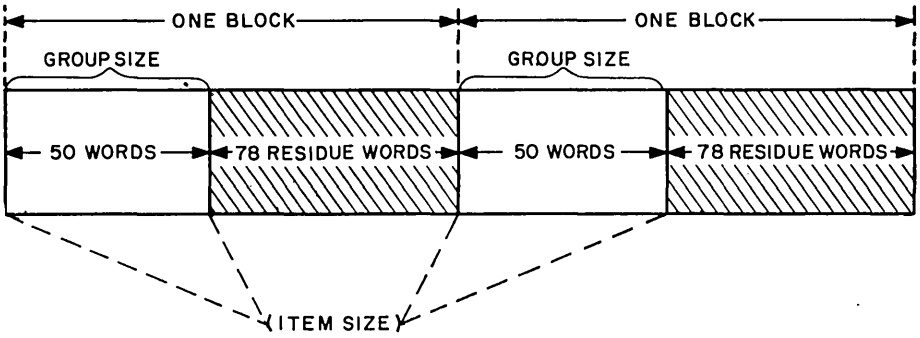
Items are sorted according to one or more words or parts of words within the item called keys. Corresponding keys must be of the same size and in the same relative location in each item.

DATA FORMAT EXAMPLES

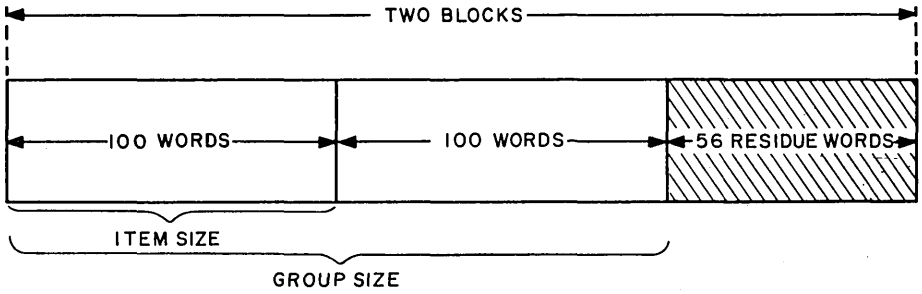
Card-to-tape conversion of ten words per card, 12 cards per block, results in a tape format of 120 words of data and eight words of residue. The item size is 10 words and the group size is 120 words to indicate that the data is arranged as follows:



An item size of 100 words and a group size of 50 words would indicate that the data is arranged as follows:



An item size of 100 words and a group size of 200 words would indicate that data is arranged as follows:



SENTINELS

Two sentinel areas, SENT1 and SENT2, are both of item size and are used in combination by the sort to signal end-of-tape and end-of-file conditions on the input tape. The areas are located in the two item-sized areas following the last item.

If input data is terminated on a sentinel basis (described within the INSENT parameter, page 18), the sort recognizes the SENT1 and SENT2 areas in the following manner:

- A possible end-of-data signal is indicated if identical words of a specified sentinel are encountered within an item area.
- A definite end-of-data signal is indicated if identical words of a second specified sentinel are encountered in the second item area immediately following the first. In certain cases, both sentinels may be the same.

If input data is terminated on an item or block count basis, both sentinels are disregarded.

SENT1 and SENT2 are always written by the sort in the two item-sized areas following the last item on the output tape.

CHECKSUM

The checksum appears in the first word of the item area following the last sentinel. Checksum verification by the sort is optional for input tapes; a checksum is always calculated and written by the sort on the output tapes. The block in which the checksum appears is the final block written by the sort. The checksum is the algebraic sum of all data words processed. Any overflow that may occur is ignored.

SECTION III

SORT PARAMETERS

INTRODUCTION

The generator of the Philco 2000 Sort System produces the TAC coding and associated subroutine calls for the sort specified by a particular sort statement.

Sort Statement

The sort statement, written by the programmer, consists of the call word SORT, and various combinations of 16 possible parameters. SORT in the Command field of a TAC card (Columns 17-24) is a call for the Sort System generator; the parameters in the Address and Remarks field of the card (Columns 25-80) each indicate one or more characteristics of the particular sort program to be generated. The statement is terminated by a dollar sign ($).

Sort Parameters

Each of the 16 parameters consists of an identifier, which labels the parameter, and one or more associated quantifiers, which directly or indirectly indicate the specific values required by the parameter. The parameter is written as:

$$\text{IDENTIFIER}(\text{quantifier}_1,\text{quantifier}_2,\ldots,\text{quantifier}_n)$$

As an example, the parameter which specifies information for input tapes has the identifier INTAPE. One of its quantifiers, which specifies the first input tape, could be written as the number 9 to indicate that Tape Unit 9 is to be the first input tape for sort operations.

Quantifiers of each parameter are separated by commas, and the entire group of quantifiers is enclosed within parentheses. The right parenthesis provides an unambiguous end-of-parameter indicator. No characters of any kind may appear between parameters, and the comma is the only character which may appear between parameter quantifiers.

Parameters may be listed in any order. The order of quantifiers within a parameter is fixed, however, and if a quantifier is omitted, a separating comma must be present to maintain quantifier sequence.

Return From Sort

The normal return upon completion of the sort is to the instruction following the sort statement. Should the sort be terminated because of an error, control will be transferred to the programmer's

error exit specified within the ERRORSUB parameter (page 31). Illegal parameters and quantifiers will cause error indications to be listed on the Code-Edit when detected at compile time, and will cause control to be transferred to the ERRORSUB exit when detected at run time.

PARAMETER FORMAT

As indicated, the word SORT is to appear in the Command field of a card, and the parameters are to appear in the Address and Remarks field. To simplify explanation of the Identifiers and Quantifiers, all capitalized words that appear within the illustration of the parameter format below are to appear on the card exactly as shown. All italicized words are to be replaced by quantifiers required to perform a specific sort.

For example, the parameter INSENT (page 18) is described as:

Address and Remarks
INSENT(*endfile,endtape*)

The written coding for specifying SENTINEL and XRAYXRAY as sentinels for the input sentinel parameter, INSENT, should then appear as:

INSENT(W/SENTINEL,W/XRAYXRAY)

Refer to Section IV, page 39, for an example of a complete sort statement.

PARAMETER FUNCTIONS

Sort System parameters are divided into two groups — 10 descriptive parameters for which the sort system generates all required coding, and six linkage parameters that provide linkage between the sort and programmer own code subroutines. The functions of the 16 parameters are listed as follows:

Descriptive Parameters

PARAMETER	FUNCTION
INFORM	Specifies the size of the items to be sorted and the arrangement of data on the input tape.
INLAB	Specifies one or more words, each of which is to be compared with the first word of the label blocks of successive input tapes.
INSENT	Specifies the input sentinel words.
INTAPE	Specifies one or two input tape units, disposition of the last input tape, and the method of recognizing the end of input.
KEY	Specifies the basis on which the items are to be sorted.
OUTFORM	Specifies the size of the items and the arrangement of data to be written onto output tapes.
OUTLAB	Specifies one or more words, each of which is to be written in the label block of successive output tapes.
OUTSENT	Specifies the output sentinel words.
OUTTAPE	Specifies one or two output tape units, disposition of the last tape, and the amount of information to be written on each output tape.
WKTAPES	Specifies the working tape units.

Linkage Parameters

PARAMETER	FUNCTION
ERRORSUB	Specifies the location of the programmer's own error subroutine to which control is to be transferred should an error, from which recovery is not possible, be detected.
INSUB	Specifies the location of the programmer's own input subroutine.
LABSUB	Specifies the location of the programmer's own label checking and/or label writing subroutines.
MERGESUB	Specifies the location of the programmer's subroutine to be entered before each merge pass begins.
OUTSUB	Specifies the location of the programmer's own output and wrapup subroutines.
PARSUB	Specifies the location of the programmer's parameter table generating subroutine, which provides parameter information at run time.

PRESET PARAMETER VALUES

Preset values will be inserted by the sort system into appropriate quantifier positions of five parameters, if the quantifiers are omitted by the programmer. These parameters are INFORM, INTAPE, KEY, OUTFORM, and OUTTAPE.

In addition, a complete parameter containing preset quantifier values will be provided by the sort system for any or all of three parameters entirely omitted by the programmer. These are INFORM, INSENT and ERRORSUB.

DESCRIPTIVE PARAMETERS

The first ten parameters to be described are those within the descriptive group — parameters for which the sort system generates all necessary coding. These parameters permit the programmer to specify a wide variety of sort programs without the need for own coding.

INFORM

Specifies the size of the items to be sorted and the arrangement of data on the input tapes.

FORMAT

Address and Remarks
INFORM(*item,group,check*)

QUANTIFIERS

item The size of the input items,* expressed in number of words as one of the following:

1. A decimal number, 1 through 3840.
2. A TAC constant of the form O/, H/, N/, D/, F/, or W/. The number of words is to be specified in Bits 1 through 15.
3. A symbolic address or M/ address. The number of words is to be specified in the address field of the indicated half-word.

If *item* is omitted, 10 words per item is assumed.

group The number of contiguous words in an input group,** written in the same manner as for *item*. The maximum number of words that may be specified for *group* is 3840. If *group* is omitted, 120 words per group is assumed.

check Checksum verification of input reels. One of the following may be selected:

CHKSUM Indicates that a checksum verification is to to be made.

NOCHKSUM Indicates that no checksum verification is to be made.

If *check* is omitted, NOCHKSUM is assumed. If input data is terminated by ITEMCNT or BLKCNT (refer to INTAPE, page 20) and checksum verification is required, two item-sized areas will be bypassed (representing the normal sentinels) and the next available word will be considered to contain the checksum.***

PRESET VALUES

The preset values for the three quantifiers of INFORM are 10, 120, and NOCHKSUM respectively. If the entire INFORM parameter is omitted, the generator supplies a parameter in the form: INFORM(10,120).

---

* Refer to page 6 for a description of items.
** Refer to page 5 for a description of groups.
*** Refer to page 8 for a description of the checksum word.

ACTION

Information specified by the INFORM parameter causes the sort program to process input data in the form designated by the *item* and *group* quantifiers. Checksum verification is dependent upon the status of the *check* quantifier. If checksum verification is requested and the checksum on the tape does not match the computed checksum, an error indication will be typed on the Console Typewriter. (Refer to page 47.)

The INFORM parameter must be used even if the INSUB parameter (page 32) is used, as the size of the items to be sorted is defined by *item*.

EXAMPLE

INFORM(8,128,CHKSUM)

Explanation: This parameter causes the sort program to process input data as 8-word items in groups of 128 words (16 items per block). Checksum verification is to be made.

INLAB

Specifies one or more words, each of which is to be compared with the first word of the label blocks of successive input tapes.

FORMAT

Address and Remarks
INLAB(lab_1,lab_2,...,lab_n)

QUANTIFIERS

lab_1 The one-word label that is to be compared with the label in the first input tape reel. It may be written in one of the following forms:

1. A TAC constant of the form O/, D/, F/, N/, H/, or W/.
2. A symbolic, decimal, or M/ address that specifies the location of a word that contains the label.

lab_2,...,lab_n One-word labels to be compared with labels of subsequent tape reels, written in the same manner as that for *lab_1*.

NOTE: Labels for those quantifier positions that are omitted (as designated by a comma only) are assumed to be the same as the last designated label configuration. If there are more input reels than labels supplied in the parameter, the last designated label specifies the labels for subsequent reels.

PRESET VALUES

None

ACTION

Information specified by the INLAB parameter causes the sort program to use the labels given within *lab_1, lab_2,. . . ,lab_n* as a comparison quantity for its label checking procedure for the respective input tapes. Only the first word of the label block must match the indicated label word. If the two labels do not match, an error indication will be typed on the Console Typewriter. (Refer to page 46.)

REMARKS

1. The use of INLAB implies that label checking is to be performed by the sort program (except as described in Remark 2).

2. If INLAB is used in the same sort statement with either INSUB or the *inent* quantifier of LABSUB, INLAB will be disregarded.

3. If neither INLAB, INSUB, or the *inent* quantifier of LABSUB is specified, it is assumed that:
   (1) no label checking is to be performed,
   (2) the first input tape is positioned at the start of the first data block, and
   (3) subsequent tapes contain no input label blocks.

EXAMPLE

INLAB(W/INLABELA,W/INLABELB,W/INLABELC)

Explanation: Action of this parameter causes the sort program to compare INLABELA with the first word of the label block of the first input reel. INLABELB is to be compared with the label of the second input reel, and INLABELC is to be compared with the label of the third input reel.

INSENT

INSENT Specifies the input sentinel words.

FORMAT

Address and Remarks
INSENT(*endfile,endtape*)

QUANTIFIERS

endfile A TAC constant that specifies the end-of-file sentinel word (SENT2*) on the input tape. The constant must appear as one of the following: O/, D/, F/, N/, H/, or W/. A symbolic, decimal, or M/ address may also be used to specify the location of a word that contains the sentinel configuration.

NOTE: The quantifier *endfile* is always required with INSENT.

endtape A TAC constant that specifies the end-of-tape sentinel word (SENT1*) on the input tape. The *endtape* sentinel is written in the same format as that for *endfile*. If *endtape* is omitted, the *endfile* sentinel is assumed for both *endfile* and *endtape*.

PRESET VALUES

None. If the entire INSENT parameter is omitted, however, the generator will supply a parameter in the form: INSENT(W/SENTINEL).

ACTION

If the input data is terminated on a sentinel basis (refer to INTAPE, page 20), the action of the INSENT parameter causes reading of an input reel to stop whenever the words in a SENT1 area match the *endtape* word and the words in a SENT2 area match either the *endtape* or *endfile* word. Input for the file is terminated when the words in the SENT2 area match the *endfile* word. If input data is terminated on a tape count basis, the action is the same as that for the sentinel except that input for the file is terminated when the required number of tapes is reached. If input data is terminated on an item count or block count basis, the sentinels specified by INSENT are disregarded.

EXAMPLE ONE

INSENT(EOF)

Explanation: Action of this parameter causes the sort program to cease reading the input tapes when the sentinel in location EOF is encountered in both the SENT1 and SENT2 areas.

---

* Refer to page 7 for a description of SENT1 and SENT2 sentinel areas.

EXAMPLE TWO

INSENT(W/SENTINEL,W/XRAYXRAY)

Explanation: This parameter causes the sort program to cease reading each input tape whenever the sentinel XRAYXRAY is encountered in the SENT1 area and either XRAYXRAY or SENTINEL is encountered in the SENT2 area. If SENTINEL is encountered in the SENT2 area, and SENT is specified as the *signal* quantifier of INTAPE, reading of the input file will be terminated.

INTAPE

Specifies one or two input tape units, disposition of the last input tape, and the method of recognizing the end of input.

FORMAT

Address and Remarks
INTAPE(t_1,*disp*,*signal*,*cnt*,t_2)

QUANTIFIERS

t_1 The first input tape unit, specified as one of the following:

1. A decimal number.
2. A TAC constant written as O/, D/, F/, N/, H/, or W/. The unit number is specified in Bits 20 through 23.
3. A symbolic address or M/ address. The unit number is specified in Bits 20 through 23 of the word at the indicated address.

If data items are contained on two or more tape reels, all odd-numbered reels (1st, 3rd, 5th) are assumed to be mounted on the tape unit specified as t_1. If t_1 is omitted, Tape 0 is assumed.

disp The disposition (rewind status) of the last tape reel, specified as:

RWDLO The reel is to be rewound with lockout.

RWD The reel is to be rewound without lockout.

NORWD The reel is not to be rewound.

If the quantifier is omitted, RWDLO is assumed. For multiple reel files, all tapes other than the last are unconditionally rewound with lockout.

signal The termination signal that designates the end of input data. Depending upon the type of signal used, one of the following may be written:

SENT End-of-File sentinel

ITEMCNT Item count*

BLKCNT Block count*

TAPECNT Reel count*

If *signal* is omitted, SENT is assumed. Whenever the physical end-of-tape is sensed, reading is to be continued from another reel.

---

* The number of items, blocks or reels to be processed is specified by the *cnt* quantifier.

cnt A decimal number that specifies the number of items, blocks, or reels to be counted for ITEMCNT, BLKCNT, or TAPECNT, respectively, as designated by *signal* above, to determine when end-of-data is reached. The quantifier *cnt* may also be the symbolic address of a word that contains the number in Bits 1 through 47.

t_2 The second input tape unit, which is identical in format to t_1. If data items are contained on two or more tape reels, however, all even-numbered reels (2nd, 4th, 6th) are mounted on the tape unit specified as t_2. If t_2 is omitted, all input is assumed to be on t_1, even though multiple reels may be required.

PRESET VALUES

The preset values for the first three quantifiers of INTAPE are 0, RWDLO, and SENT, respectively.

ACTION

If there is only one input tape, information specified by the INTAPE parameter causes the sort program to read input data from the tape unit designated by t_1 until the sentinel specified by the INSENT parameter (page 18) or the quantity designated within the *cnt* quantifier of INTAPE is reached. If there are two or more input tapes, input data will be read first from the tape unit designated as t_1, second from t_2, third (if present) from t_1 and so forth, alternating between the two units whenever the INSENT sentinel or physical end-of-tape is encountered. After the end-of-file signal is encountered, the last tape reel is either rewound with lockout, rewound without lockout, or is not rewound, depending upon *disp*. All other reels are rewound with lockout.

EXAMPLE

INTAPE(6,RWDLO,TAPECNT,3,7)

Explanation: This parameter causes the sort program to read input data from the tape reels mounted on Units 6 (t_1) and 7 (t_2) until the sentinel on the third reel is encountered. The last input tape is to be rewound with lockout.

KEY

KEY

Specifies the basis on which the items are to be sorted.

FORMAT

Address and Remarks
KEY(*n,type,word,mask*)

QUANTIFIERS

n A decimal number, one through 255, that indicates the sequence number of the key. If *n* is omitted, the value 1 is assumed.

type The type of key comparison, indicated by one of the following:

A Alphanumeric

B Signed binary

F Floating-point

If *type* is omitted, B is assumed.

word A decimal number which indicates the word, within the item, that contains the key. The first word of the item is defined as Word 1.

If *word* is omitted, the value 1 is assumed.

mask A TAC constant that specifies the mask by which the key word is to be extracted. The constant may be in the form O/, D/, F/, N/, H/, or W/. A symbolic, decimal, or M/ address may also be used to indicate the location of the mask.

If *mask* is omitted, a full-word key is assumed. A mask of all zeros produces no comparison code and in effect deletes the key.

PRESET VALUES

The preset values for the first three quantifiers of KEY are 1, B, and 1, respectively.

ACTION

Information provided by one or more KEY parameters causes input items to be sorted according to the particular specifications of each parameter, in the order listed by key number. The primary sequence is based upon the parameter identified as Key One, while supplementary sequencing is based upon information supplied by subsequent KEY parameters, identified as Key 2, Key 3, etc. A single sort may contain a maximum of 255 keys, each of which must be specified by a different KEY parameter. The parameter table (page A-1) is searched at run time for the most recent

definition of Key One, then Key Two, and so on until a key definition cannot be found for the next number. For example, if the parameter table contains entries for Key Numbers 1, 2, 3, and 5 (but not 4) only Keys 1, 2, and 3 will be used.

EXAMPLE ONE

KEY(1,A,5)

Explanation: This parameter, assuming no other KEY parameter is specified, causes the sort program to sort alphanumerically on the full fifth word of each item.

EXAMPLE TWO

KEY(3,B,4,N/1T23)

Explanation: This parameter causes the sort program to sort on Bit 23 of the fourth word of each item as the third key of the sort.

OUTFORM

Specifies the size of the items and the arrangement of data to be written onto output tapes.

FORMAT

Address and Remarks
OUTFORM(*item,group*)

QUANTIFIERS

item The size of the output items,* expressed in number of words and written in one of the following forms:

1. A decimal number, one through 3840.
2. A TAC constant of the form O/, H/, N/, D/, F/, or W/. The number of words is specified in Bits 1 through 15.
3. A symbolic address or M/ address. The number of words is specified in the address field of the indicated half-word.

group The number of contiguous words in an output group,** written in the same manner as for *item*. The maximum number of words which may be specified for *group* is 3840. If *group* is omitted, 120 words per group is assumed.

PRESET VALUES

The preset values for the two quantifiers of OUTFORM are 10 and 120, respectively. If the entire OUTFORM parameter is omitted, and OUTSUB (page 35) is also omitted, the same values specified for INFORM will be assumed for OUTFORM.

ACTION

Information specified by the OUTFORM parameter causes the sort program to write data on the output tapes in the form designated by the *item* and *group* quantifiers. If neither the OUTFORM nor the OUTSUB parameter (page 35) is specified, the output item size and group size will be assumed to be the same as the corresponding input sizes.

REMARKS

1. If the value of *item* specified in the OUTFORM parameter is less than that specified in the INFORM parameter, the number of words by which they differ will be deleted from the end of each item when they are written onto the output tapes.
2. If the value of *item* specified in the OUTFORM parameter is greater than that specified in the INFORM parameter, the necessary number of filler words*** are added to the end of each sorted item prior to output.

EXAMPLE

OUTFORM(5,125)

Explanation: This parameter causes the sort program to write output data as 5-word items in groups of 125.

* Refer to page 6 for a description of items.
** Refer to page 5 for a description of groups.
***Filler words are composed of eight filler characters (octal 32).

OUTLAB

Specifies one or more words, each of which is to be written in the label block of successive output tapes.

FORMAT

Address and Remarks
OUTLAB(lab_1,lab_2,...,lab_n)

QUANTIFIERS

lab_1 The one-word label that is to be written in each of the 128 words of the label block on the first output tape. It may be written as one of the following:

1. A TAC constant of the form O/, D/, F/, N/, H/, or W/.
2. A symbolic, decimal, or M/ address that specifies the location of a word that contains the label.

lab_2,..,lab_n One-word labels to be written on subsequent tape reels, specified in the same manner as *lab_1*.

NOTE: Labels for those quantifier positions that are omitted (as designated by a comma only) are assumed to be the same as the last designated label configuration. If there are more output reels than labels, the last designated label will be written on subsequent reels.

PRESET VALUES

None

ACTION

Information specified by the OUTLAB parameter causes the sort program to write the labels designated by lab_1,lab_2,...,lab_n in each of the 128 words of the label blocks on each of the respective output tapes.

REMARKS

1. The use of OUTLAB implies that label writing is to be performed by the sort program (except as described in Remark 2).
2. If OUTLAB is used in the same sort statement with either OUTSUB or the *outent* quantifier of LABSUB, OUTLAB will be disregarded.
3. If neither OUTLAB, OUTSUB, or the *outent* quantifier of LABSUB is specified, it is assumed that no label writing is to be performed.

EXAMPLE

OUTLAB(W/XXXXXXXX,W/YYYYYYYY,W/ZZZZZZZZ)

Explanation: This parameter causes the sort program to write XXXXXXXX as the output label for the first output reel, YYYYYYYY as the label for the second output reel, and ZZZZZZZZ as the label for all subsequent reels.

OUTSENT

Specifies the output sentinel words.

FORMAT

Address and Remarks
OUTSENT(*endfile,endtape*)

QUANTIFIERS

endfile A TAC constant that specifies the end-of-file sentinel word (SENT2*) to be written on the final output tape. The constant must be one of the following: O/, D/, F/, N/, H/, or W/. A symbolic, decimal, or M/ address may also be used to specify the location of a word that contains the sentinel configuration.

NOTE: The quantifier *endfile* is always required with OUTSENT.

endtape A TAC constant that specifies the end-of-tape sentinel word (SENT1*) to be written on the output tapes. The *endtape* quantifier is written in the same format as that for *endfile*. If *endtape* is omitted, the *endfile* sentinel is assumed for both *endfile* and *endtape*.

PRESET VALUES

None. If the entire OUTSENT parameter is omitted, however, the sort system will assume sentinels equivalent to the input sentinels (see INSENT, page 18).

ACTION

Information specified by the OUTSENT parameter causes the endtape sentinel to be written in the SENT1 area whenever writing of data on an output tape is complete, either through exhaustion of data or fulfilling the count specified by the *cnt* quantifier of OUTTAPE (page 27).

If writing is terminated because the count specified in OUTTAPE is fulfilled, the *endtape* sentinel will also be used to write a SENT2 area. When data is exhausted, the SENT2 area is written with *endfile* sentinels.

EXAMPLE ONE

OUTSENT(W/SENTINEL,W/XRAYXRAY)

Explanation: This parameter causes the sort program to write XRAYXRAY in both the SENT1 and SENT2 areas of all output tapes other than the last. In the last output tape, XRAYXRAY will be written in the SENT1 area and SENTINEL in the SENT2 area.

EXAMPLE TWO

OUTSENT(EOF)

Explanation: This parameter causes the sort program to write the word at location EOF in both the SENT1 and SENT2 areas of all output tapes.

* Refer to page 7 for a description of SENT1 and SENT2 sentinel areas.

OUTTAPE

Specifies one or two output tape units, disposition of the last tape, and the amount of information to be written on each output tape.

FORMAT

Address and Remarks
OUTTAPE(t_1,*disp*,*signal*,*cnt*,t_2)

QUANTIFIERS

The quantifiers t_1 , *disp* , and t_2 are the same as those for INTAPE (page 20), except that they refer to the output tape(s). If t_1 is omitted, Tape Unit 9 is assumed to be the output tape.

signal The condition for switching output tape units between t_1 and t_2 . Either of the following may be selected:

EOT Indicates that units are to be switched when the physical end-of-tape is detected.

BLKCNT Indicates that units are to be switched after the number of blocks specified by *cnt* have been written on the reel.

If *signal* is omitted, EOT is assumed.

cnt A decimal number that specifies the number of blocks for BLKCNT in the *signal* quantifier. The quantifier *cnt* may also be the symbolic address of a word that contains the number in Bits 1 through 47.

PRESET VALUES

The preset values for the first three quantifiers of OUTTAPE are 9, RWDLO, and EOT, respectively.

ACTION

Information specified by the OUTTAPE parameter causes the sort program to write output data onto the tape unit(s) designated by t_1 and possibly t_2 switching the units whenever the conditions specified by *signal* (and possibly *cnt*) have been fulfilled. After all output has been written, the last tape reel is either rewound with lockout, rewound without lockout, or is not rewound, depending upon *disp*. All other reels are rewound with lockout.

EXAMPLE ONE

OUTTAPE(,RWD,BLKCNT,CNT1)

Explanation: This parameter causes the sort program to write output data on Tape Unit 9 until the number of blocks specified in symbolic location CNT1 are written. If more data is to be written, a different reel must be mounted on Unit 9 after the initial reel has been rewound with lockout. The last output reel is to be rewound without lockout.

EXAMPLE TWO

OUTTAPE(5,NORWD,EOT,,6)

Explanation: This parameter causes the sort program to write output data onto Tape Unit 5 until end-of-tape, and then continue writing on Tape Unit 6. Writing is to continue in this manner on both tapes until all data is written. The final output tape is not to be rewound.

WKTAPES

Specifies the working tape units.

FORMAT

Address and Remarks
WKTAPES($t_1,t_2,\ldots,t_n$)

QUANTIFIERS

$t_1,t_2,\ldots,t_n$ The working tape units, each of which must be specified as one of the following:

1. A decimal number.
2. A TAC constant written as O/, D/, F/, N/, H/, or W/. The unit number is specified in Bits 20 through 23.
3. A symbolic address or M/ address. The unit number is specified in Bits 20 through 23 of the word at the indicated address.

A minimum of three tapes must be available as working tapes. These may include (without being specified) those selected as output tapes under the OUTTAPE parameter (page 27). Thus, if both OUTTAPE tape quantifiers are specified a minimum of one tape must be specified as a WKTAPES quantifier. If the OUTTAPE parameter is omitted, however, three tapes must be specified as WKTAPES quantifiers. Input tapes designated under the INTAPE parameter (page 20) may also be specified as working tapes, but they must be written as WKTAPES quantifiers. If only three working tapes are available (including the two output tapes), only one input tape may be specified as a working tape.

PRESET VALUES

None

ACTION

Information specified by the WKTAPES parameter causes the sort program to use those tapes specified by $t_1,t_2,\ldots,t_n$, plus any tapes specified as OUTTAPE quantifiers, as working tapes for sort operations.

EXAMPLE ONE

WKTAPES(7,8,9)

Explanation: This parameter causes the sort program to assign Tape Units 7, 8, and 9 as the working tape units.

EXAMPLE TWO

WKTAPES(9)

Explanation: This parameter causes the sort program to assign Tape Unit 9 and the output units specified by the OUTTAPE parameter as working tape units. In order for this sort to run, two output units must have been assigned within the OUTTAPE parameter.

LINKAGE PARAMETERS

The second parameter group provides linkage between the programmer's own subroutines and the sort.

All identifiers of the six parameters within this group end with the letters "SUB," denoting that the programmer is to provide his own subroutine for each parameter selected. In all cases except for ERRORSUB, return is made to the sort by the TJM-JMP method, and any index registers other than 1 or 2 used by the programmer's code must be restored prior to return. In the case of ERRORSUB, no return is made to the sort because use of ERRORSUB indicates the detection of a non-recoverable error.

ERRORSUB

Specifies the location of the programmer's own error subroutine to which control is to be transferred, should an error, from which recovery is not possible, be detected. (Refer to Section V, page 53, for a listing of sort error indications.)

FORMAT

Address and Remarks
ERRORSUB(*ent*)

QUANTIFIER

ent The entrance address of the error subroutine, specified as a decimal, symbolic,* or M/ address.

PRESET VALUES

None. If the entire ERRORSUB parameter is omitted, the generator supplies a parameter in the form: ERRORSUB(A36ERR). (Refer to page B-2.)

ACTION

Information specified by the ERRORSUB parameter causes the sort program to transfer control to the address specified by *ent* whenever an error, from which recovery is not possible, is detected.

REMARKS

In general, errors may be classified into two groups — those in which recovery may be made by transferring control to an earlier phase of the sort, and those in which recovery is not possible. In cases where recovery is possible, an option is generally given that permits the operator to attempt recovery, or to transfer control to an error exit.

EXAMPLE ONE

ERRORSUB(ERRS)

Explanation: This parameter causes the sort program, whenever an error from which recovery is not possible is detected, to transfer control to location ERRS.ERRS, the starting address of the programmer's own error subroutine.

EXAMPLE TWO

ERRORSUB(MAS.ERRS)

Explanation: This parameter causes the sort program to transfer control to location MAS.ERRS if an error, from which recovery is not possible, is detected.

---

* If a single symbol is written for this quantifier, such as ABC, the quantifier will be assumed to be referencing a subroutine in TAC format. Hence, in this case, control will be transferred to location ABC.ABC. Any legitimate name.symbol, however, such as XRAY.ZEBRA, may be written.

INSUB

Specifies the location of the programmer's own input subroutine.

FORMAT

Address and Remarks
INSUB(*ent*)

QUANTIFIER

ent The entrance address of the input subroutine, specified as a decimal, symbolic,* or M/ address.

PRESET VALUES

None

ACTION

Information specified by the INSUB parameter causes the sort program to transfer control to the address specified by *ent*, whenever input data is to be transferred from the programmer's routine to the sort routine.

When the programmer's input subroutine has an item for the sort, a return is made to the sort by the TJM-JMP method. Index Register 1 must contain the address of the first word of the item. Any index registers other than 1 or 2 which are used by the programmer's code must be restored prior to return. When no further items are available, the return to the sort must be one full word beyond the normal return.

REMARKS

1. When an INSUB parameter is present, the INLAB (page 16) parameter and the *inent* quantifier of the LABSUB (page 33) parameter are disregarded. Hence, the subroutine specified by INSUB must provide for label checking if required.

2. Even though an INSUB parameter is specified, an INFORM parameter (page 14) with the *item* quantifier must be supplied to specify the size of the items to be sorted.

3. INSUB, if used, must be specified at compile time.

EXAMPLE

INSUB(IN)

Explanation: This parameter causes the sort program to transfer control to symbolic address IN.IN whenever an item is to be transferred from the programmer's routine to the sort routine.

---

* Refer to footnote, page 31.

LABSUB

Specifies the location of the programmer's own label checking and/or label writing subroutines.

FORMAT

Address and Remarks
LABSUB(*inent,outent*)

QUANTIFIERS

inent The entrance address of the programmer's own label checking subroutine, specified as a decimal, symbolic,* or M/ address.

outent The entrance address of the programmer's own label writing subroutine, specified in the same manner as for *inent*.

PRESET VALUES

None

ACTION

Information specified by the LABSUB parameter causes the sort program to transfer control to the address specified by *inent* whenever input labels are to be checked. Whenever output labels are to be written, the sort program transfers control to the address specified by *outent*. Upon entry to the address specified by *inent* or *outent* , the A Register will contain in Bits 20 through 23 the number of the tape involved. Return to the sort is accomplished via the TJM-JMP method. Any index registers used other than 1 or 2 must be restored prior to return.

REMARKS

The *inent* and *outent* quantifiers override the INLAB and OUTLAB parameters respectively. The INSUB and OUTSUB parameters, in turn, override the *inent* and *outent* quantifiers, respectively, of the LABSUB parameter.

EXAMPLE ONE

LABSUB(IN,OUT)

Explanation: This parameter causes the sort program to transfer control, whenever input labels are to be checked, to location IN.IN, which should be the starting address of the programmer's own label checking subroutine. Whenever output labels are to be written, the sort program will transfer control to location OUT.OUT, which should be the starting address of the programmer's own label writing subroutine.

EXAMPLE TWO

LABSUB(,M/2210)

Explanation: This parameter causes the sort program to transfer control to memory location 2210_8 each time a label is to be written. Checking of input labels will be done by the sort if an INLAB parameter is present.

* Refer to footnote on page 31.

MERGESUB

MERGESUB

Specifies the location of the programmer's own subroutine to be entered before each merge pass begins.

FORMAT

Address and Remarks
MERGESUB(*ent*)

QUANTIFIER

ent The entrance address of the inter-merge subroutine, specified as a decimal, symbolic,* or M/ address.

PRESET VALUES

None

ACTION

Information specified by the MERGESUB parameter causes the sort program to transfer control to the address specified by *ent* prior to the start of each merge pass. Return to the sort is accomplished via the TJM-JMP method. Any index registers other than 1 or 2 which are used by the programmer's code must be restored prior to return.

EXAMPLE

MERGESUB(ABLE)

Explanation: This parameter causes the sort program to transfer control to location ABLE.ABLE just prior to the start of each merge pass.

* Refer to footnote on page 31.

OUTSUB

Specifies the location of the programmer's own output subroutine and wrapup coding.

FORMAT

Address and Remarks
OUTSUB(*itement,wrapent*)

QUANTIFIERS

itement The entrance address of the output subroutine, specified as a decimal, symbolic,* or M/ address.

wrapent The entrance address of the wrapup coding, specified in the same manner as *itement*. Control is transferred to *wrapent* when end-of-data has been encountered during the final merge pass.

PRESET VALUES

None

ACTION

Information specified by the OUTSUB parameter causes the sort program to transfer control to the address specified by *itement* each time an item is available for output. The sort program will transfer control to the address specified by *wrapent* when no more data is available.

When the sort transfers control to *itement,* Index Register 1 contains the address of the first word of the item for output. Return from the programmer's subroutine is accomplished via the TJM-JMP method. Any index registers other than 1 or 2 which are used by the programmer's code must be restored prior to return.

When the sort transfers control to *wrapent* , it has returned all allocated memory** and restored all index registers other than 1 and 2. Return to the sort from the wrapup coding is not necessary but, if desired, may be accomplished via the TJM-JMP method. The sort will then exit to the instruction immediately beyond the original sort statement.

REMARKS

1. OUTSUB, when used, must be specified at compile time.

2. When an OUTSUB parameter is present, the OUTLAB parameter (page 25) and the *outent* quantifier of the LABSUB parameter (page 33) are disregarded. Hence, the subroutine specified by OUTSUB must provide for label writing if required.

---

* Refer to footnote on page 31.
** Refer to Memory Allocation, page 2.

3. When OUTSUB is used, OUTFORM need not be used. If OUTFORM is specified when OUTSUB is used, any information provided by OUTFORM will be disregarded.

EXAMPLE

OUTSUB(ABLE,BAKER)

Explanation: This parameter causes the sort program to transfer control to location ABLE.ABLE each time an item is available for output, and to location BAKER BAKER when the end of data has been found.

PARSUB

Specifies the location of the programmer's subroutine for generating a parameter table segment. The table segment provides revised parameter information at run time.

FORMAT

Address and Remarks
PARSUB(*ent*)

QUANTIFIER

ent The entrance address of the segment generating subroutine specified as a decimal, symbolic,* or M/ address.

PRESET VALUES

None

ACTION

Information specified by the PARSUB parameter causes control to be transferred to the address specified by *ent* — during initialization of the sort at run time. The subroutine whose entrance address is to be located in *ent* is assumed to provide a supplementary segment containing run time parameter information for the parameter table. Any number of PARSUB parameters, each contributing one or more segments, may be specified in a single sort call. Return must be made to the sort via the TJM-JMP method with Index Register 1 set to the starting location of a parameter table segment (refer to Appendix A). Any index registers, other than 1 or 2, used by the programmer's code, must be restored prior to the return. If one or more PARSUB parameters are to be used, at least one must be specified at compile time.

EXAMPLE

PARSUB(SYSPAR)

Explanation: This parameter causes the sort program to transfer control to the SYSPAR subroutine, which is to provide run time parameter information for a supplementary parameter segment.

* Refer to footnote on page 31.

SECTION IV

SORT EXAMPLES

EXAMPLE ONE

The first sort example illustrates the use of the descriptive parameters — those which specify a wide variety of sort programs without the need for supplementary own coding.

The example concerns an inventory tape containing a label block, inventory items, and two item-sized sentinels. The tape was created by using card-to-tape equipment, processing five-word items with 24 items per block. The card deck consisted of one card containing a label (RECΔ0263) in its first word, followed by 23 blank cards; the inventory items; one card containing in each of its first five words the word SENTTAPE; and one card containing the word SENTFILE in each of its first five words.

Output from the sort, scheduled for further processing, is to be in two separate files — one by stock number and one by bin number.

It is necessary to sort these items first by stock number (the first word of each item) and then by bin number (contained in the first fifteen bits of the second word). The output tape of records sorted by stock number is to be used as input to the bin number sort operation.

The following coding illustrates one method by which this sorting may be accomplished.

PHILCO COMPUTER DIVISION

PHILCO CODING FORM

Page.....of.....

Program: | Programmer: | Date:

IDENTITY AND SEQUENCE	L	LOCATION	COMMAND	ADDRESS AND REMARKS
			SØRT	INTAPE(N/OT23,,RWDLØ,,SENT)
				ØUTAPE(N/9T23,,RWD,,EØT)
				WKTAPES(N/3T23,,N/6T23,,N/4T23)
				INFØRM(SIZE,,SIZE+IH)
				ØUTFØRM(SIZE,,SIZE+IH)
				INSENT(W/SENTFILE,,W/SENTTAPE)
				ØUTSENT(W/SENTFILE,,W/SENTTAPE)
				INLAB(W/RECΔ0263)
				ØUTLAB(W/STN00263)
				KEY(1,,A,,1)$
			SØRT	INTAPE(9,,RWDLØ,,SENT)
				ØUTTAPE(10,,RWDLØ,,EØT)
				WKTAPES(3,,6,,4)
				INFØRM(SIZE,,SIZE+IH)
				INSENT(W/SENTFILE,,W/SENTTAPE)
				INLAB(W/STN00263)
				ØUTLAB(W/BIN00263)
				KEY(1,,B,,2,,Ø/77777)$
		SIZE	N/5T15;N/12ØT39$	

*(Note that the parameters **OUTFORM** and **OUTSENT** need not be specified because the quantifiers are the same as those of **INFORM** and **INSENT**.)*

The same sort may also be accomplished by providing a subroutine for run time parameter table supplementation.

If SYSPAR (page B-3) were to be used, the sort statements would then be coded in the following way:

```
L  Location  Command  Address and Remarks
             SORT     PARSUB(SYSPAR)$
             SORT     PARSUB(SYSPAR)$
```

The parameters (in the format specified in Appendix C) would be supplied on the input tape at run time.

EXAMPLE TWO

The second sort example illustrates the use of the linkage parameters — those which provide linkage between the sort and programmer's own subroutines.

This example concerns a master file of employees which is required by a manufacturing firm with plants in twelve locations. The file is to be used for determining which employees may be transferred to openings in other plants within the next fiscal year.

The records of employees who will be of retirement age by the end of the next fiscal year are to be excluded from the master file.

The individual plants have supplied files of employee records that are to be sorted according to job category. Any records in the same job category are to be sorted by employee number.

All of the tapes have the same format, ten words per item, 200 words per group. The first word of each item contains the employee number, and the last twenty-four bits of the fifth word contain the job number. The first block of each tape is a label block that indicates the plant number.

All files except that of the central office contain two item-sized sentinels of the word ENDOTAPE. The file of the central office contains one item-sized sentinel of ENDOTAPE and one of ENDOFILE.

Because some employee records are to be excluded, an input routine must be supplied.

PHILCO
COMPUTER DIVISION

PHILCO CODING FORM

Page.....of.....

Program: | Programmer | Date:

IDENTITY AND SEQUENCE	L	LOCATION	COMMAND	ADDRESS AND REMARKS
			NAME	ØWNINS
		START	SØRT	INSUB(ØWNIN)INFØRM(IO,,200)ØUTTAPE(IO,RWDLØ,,EØT,,,I I)
				ØUTFØRM(IO,,200)ØUTSENT(W/ENDØFILE,W/ENDØTAPE)
				WKTAPES(3,,6,,IO,,I I)
				ØUTLAB(W/MAST1Δ63,W/MAST2Δ63,W/MAST3Δ63)
				KEY(I,,B,,5,,24/IT47)KEY(2,,A,,I)$
			•	
			•	
			•	
		ØWNIN	TJM	EXIT $
			•	
			•	
			•	
		EXIT	JMP	X $
		ENDATA	INCA	EXIT $
			JMP	EXIT $
			END	START$

(At exit time, Index Register 1 must contain the address of the first word of the item. Note that INFORM must still be used in the sort statement. Any label checking must be provided in this coding.)

(When input has been exhausted, return to the sort must be one word beyond the TJM–JMP return.)

If the exclusion of some employee records had not been necessary, the following parameters could have been used in place of INSUB:

L	Location	Command	Address and Remarks
			INTAPE(SOMTAPE,RWDLO,TAPECNT, 13,ANOTHER)
			INSENT(W/ENDOFILE,W/ENDOTAPE)
			LABSUB(OWNIN.CHKPLTS)

The input label checking subroutine CHKPLTS must use the TJM-JMP method for return to the sort.

SECTION V

ERROR INDICATIONS AND RECOVERY PROCEDURES

Errors detected by the Sort System during compilation are indicated on the Code-Edit, while errors detected at run time are indicated on the Console Typewriter.

RUN TIME ERRORS

Run time errors may be classified into two groups - those from which recovery may be made by returning to an earlier phase of the sort and those from which recovery is not possible.

Recoverable Errors

When error recovery is possible, an option is generally given that permits the operator to attempt recovery, or to indicate to the sort that control is to be transferred to the location specified by the ERRORSUB parameter (page 31). A complete description of these errors, with possible recovery procedures, is presented in Table 5-A (page 45).

Non-Recoverable Errors

When error recovery is not possible, control is automatically transferred to the error subroutine. A complete listing of these errors is contained within Table 5-B (page 53).

INDICATIONS UPON ERROR EXIT

In all cases when control is transferred to the error subroutine, Index Registers 1 and 2 will indicate the subroutine in which the error was detected and the type of error, respectively (refer to Table 5-B). The D Register will contain in its left address field the return address to the programmer's coding upon normal completion of sort operations.

An indication of the current sort phase is available during sort operations. (Refer to Table 5-C, page 54.) Table 5-D indicates the location of information associated with XORD errors (page 54).

RELATIONSHIP BETWEEN THE SORT AND XORD

Because XORD* is used by the Sort System for input and output operations, a SETXORD call is normally issued by the sort so that it may institute error recovery procedures upon detection of a tape error. In order to accommodate programs that use a tape reassignment table and/or process tape errors by own code, operation of the sort may be modified.

* XORD is a Philco subroutine for the execution of magnetic tape input-output orders (Program Description PD-11).

The word at A36SORT.SETXORD is checked by the sort at the start of operations; if this word is negative (Bit 0 is set to 1) the SETXORD call will not be issued. If the word is positive the sort will use it as the parameter of the SETXORD call. Initially, this word contains only the address of the preset XORD error processing routine in Bits 25-40.

If there is a tape reassignment table, either the address of this table may be placed in Bits 1-15 of A36SORT.SETXORD, or Bit 0 of this word may be set to one. If Bit 0 is set to 1, the programmer assumes the responsibility of processing tape error exits from XORD.

TABLES

INITIAL TYPE-OUT	SORT PHASE	CONDITION	INITIAL TYPE-OUT AND PROCESSING
LABEL RD ERR TAPE *t*	Label Check	Begin Tape, End Tape, S1 or S2 error detected (no *inent* quantifier of LABSUB parameter)	1. LABEL RD ERR TAPE *t* is typed out, followed by a statement of the error type(s). 2. QUIT,RETRY? is typed out, followed by a type-in request. a. If QUIT. is typed in, control will be transferred to the ERRORSUB exit with Index Register 1 set to 13_8 and Index Register 2 set to 4. b. If RETRY. is typed in, the tape will be rewound with lockout and a re-read of the label block will be attempted.
LABEL RD ERR TAPE *t*	Label Check	Begin Tape, End Tape, S1, or S2 error detected (*inent* quantifier of LABSUB parameter)	1. LABEL RD ERR TAPE *t* is typed out, followed by a statement of the error type(s). 2. Control is transferred to the ERRORSUB exit with Index Register 1 set to 13_8 and Index Register 2 set to 4.
LAB TAPE lab_n NOT lab_x	Label Check	Label not as specified by INLAB parameter	1. LAB TAPE lab_n NOT lab_x is typed out. 2. IGNORE, QUIT,RETRY? is typed out, followed by a type-in request. a. If IGNORE. is typed in, processing will continue as if the label check had not been in error. b. If QUIT. is typed in, control will be transferred to the ERRORSUB exit with Index Register 1 set to 6 and Index Register 2 set to 4. The label encountered will be placed in the A Register and the label specified by the INLAB parameter will be placed in the Q Register. c. If RETRY. is typed in, the tape will be rewound with lockout and a re-check of the label block will be attempted.

Table 5-A. Recoverable Errors and Recovery Procedures Continued

INITIAL TYPE-OUT	SORT PHASE	CONDITION	INITIAL TYPE-OUT AND PROCESSING
INPUT CKSMERR	Prepass	Checksum error detected	1. INPUT CKSMERR, IGNORE, QUIT? is typed out, followed by type-in request. a. If IGNORE. is typed in, processing will continue as if the checksums had matched. b. If QUIT. is typed in, control is transferred to the ERRORSUB exit with Index Register 1 set to 6 and Index Register 2 set to 5. The computed checksum is placed into the A Register and the expected checksum is placed into the Q Register.
PREPASS RD ERR TAPE *t*	Prepass	S1, S2, or Begin Tape error detected on input tape	1. PREPASS RD ERR TAPE *t* is typed out, followed by a statement of the error type. 2. Control is transferred to the ERRORSUB exit with Index Register 1 set to 13_8 and Index Register 2 set to 4.
PREPASS RD ERR TAPE *t*, EOT	Prepass	End of input tape detected (no INSUB parameter)	1. PREPASS RD ERR TAPE *t*, EOT is typed out. 2. Tape *t* is rewound with lockout. 3. Processing continues. Label checking on next reel is done if specified. If only one input unit was specified (no t_2 quantifier in INTAPE parameter), TAPE *t* IN LOCAL will be typed out and the computer will halt. The operator should press the ADVANCE bar after a new reel is mounted.
PREPASS RD ERR TAPE *t*, EOT	Prepass	End of input tape detected (INSUB parameter)	1. PREPASS RD ERR TAPE *t*, EOT is typed out. 2. Control is transferred to the ERRORSUB exit with Index Register 1 set to 13_8 and Index Register 2 set to 4.

Continued

Table 5-A. Recoverable Errors and Recovery Procedures

INITAL TYPE-OUT	SORT PHASE	CONDITION	INITIAL TYPE-OUT AND PROCESSING
PREPASS WR ERR TAPE *t*	Prepass	S1, S2, Begin Tape, or End Tape error detected on merge tape	1. PREPASS WR ERR TAPE *t* is typed out, followed by a statement of the error type. 2. Control is transferred to the ERRORSUB exit with Index Register 1 set to 13_8 and Index Register 2 set to 4.
MGPASS *n* RD ERR TAPE *t*	Merge	S1, S2, Begin Tape, End Tape, checksum or block count error detected on tape being read	1. MGPASS *n* RD ERR TAPE *t* is typed out, followed by a statement of the error type(s). 2. QUIT,RETRY? is typed out, followed by a request for a type-in. a. If QUIT. is typed, control is transferred to the ERRORSUB exit, with Index Register 1 set to 13_8 and Index Register 2 set to 4. b. If RETRY. is typed in, all merge tapes will be rewound and tape *t* will be rewound with lockout. NEW UNIT *t* MAY BE SELECTED, KEEP SAME REEL will then be typed out. All merge tapes will be positioned at the start of the current pass and the pass will be retried.
MGPASS *n* WR ERR TAPE *t*	Merge	S1, S2, Begin Tape or End Tape error detected on tape on which data is being written	1. MGPASS *n* WR ERR TAPE *t* is typed out, followed by a statement of the error type(s). 2. QUIT,RETRY? is typed out, followed by a request for a type-in. a. If QUIT. is typed in, control is transferred to the ERRORSUB exit

Continued

Table 5-A. Recoverable Errors and Recovery Procedures

INITIAL TYPE-OUT	SORT PHASE	CONDITION	INITIAL TYPE-OUT AND PROCESSING
MGPASS *n* WR ERR TAPE *t* (Continued)			with Index Register 1 set to 13_8 and Index Register 2 set to 4. b. If RETRY. is typed in, all tapes are rewound, and Tape *t* is rewound with lockout. NEW UNIT *t* MAY BE SELECTED, NEW REEL MAY BE USED. NEW, OLD? is typed out, followed by a request for a type-in. (1) If NEW. is typed in, Tape *t* is not repositioned before the current pass is restarted. All other merge tapes are repositioned before restarting. (2) If OLD. is typed in, the sort program reacts as in (1) above except that the reel on unit *t* is treated the same as the other merge output tapes.
MRGSENT WR ERR TAPE *t*	Merge	S1, S2, Begin Tape, or End Tape error detected on tape on which a beginning sentinel is being written	1. MRGSENT WR ERR TAPE *t* is typed out, followed by a statement of the error type(s). 2. QUIT,RETRY? is typed out, followed by a type-in request. a. If QUIT. is typed in, control is transferred to the ERRORSUB exit, with Index Register 1 set to 13_8 and Index Register 2 set to 4. b. If RETRY. is typed in, Tape *t* is rewound with lockout and NEW UNIT *t* MAY BE SELECTED, NEW REEL MAY BE USED. is typed out. Processing then continues with another attempt to write the sentinel.

Continued

Table 5-A. Recoverable Errors and Recovery Procedures

INITIAL TYPE-OUT	SORT PHASE	CONDITION	INITIAL TYPE-OUT AND PROCESSING
LASTPASS RD ERR TAPE *t*	Last Pass	S1, S2, Begin Tape, End Tape, checksum or block count error detected on tape being read (no OUTSUB parameter and no *outent* quantifier of LABSUB parameter)	1. LASTPASS RD ERR TAPE *t*, is typed out, followed by a statement of the error type(s). 2. Processing is the same as for this condition during the Merge phase except that in the Last Pass phase, all output tapes are rewound with lockout if RETRY. is typed in.
LASTPASS RD ERR TAPE *t* or LASTPASS WR ERR TAPE *t*	Last Pass	Any tape error other than End Tape detected on tape being written (OUTSUB parameter or *outent* quantifier of LABSUB parameter)	1. LASTPASS RD ERR TAPE *t* or LASTPASS WR ERR TAPE *t*, is typed out, followed by a statement of the error type(s). 2. Control is transferred to the ERRORSUB exit with Index Register 1 set to 13_8 and Index Register 2 set to 4.
LASTPASS WR ERR TAPE *t*	Last Pass	S1, S2, or Begin Tape error detected on tape being written (no OUTSUB parameter and no *outent* quantifier of LABSUB parameter)	1. LASTPASS WR ERR TAPE *t*, is typed out, followed by a statement of the error type(s). 2. Processing is the same as for this condition during the Merge phase except that in the Last Pass phase, all output tapes are rewound with lockout and the option NEW, OLD? is not typed out.
LASTPASS WR ERR TAPE *t*, EOT	Last Pass	End of Tape error detected on output tape (*signal* quantifier of OUTTAPE parameter set to EOT. No OUTSUB parameter)	1. LASTPASS WR ERR TAPE *t*, EOT is typed out. 2. Tape *t* is rewound with lockout. 3. Processing will continue on next output reel.

Continued

Table 5-A. Recoverable Errors and Recovery Procedures

INITIAL TYPE-OUT	SORT PHASE	CONDITION	INITIAL TYPE-OUT AND PROCESSING
LASTPASS WR ERR TAPE *t*, EOT	Last Pass	End of Tape error detected on output tape (*signal* quantifier of OUTTAPE parameter set to BLKCNT. No OUTSUB parameter)	1. LASTPASS WR ERR TAPE *t*, EOT is typed out. 2. Tape *t* is rewound with lockout. 3. IGNORE, QUIT, RETRY? is typed out, followed by a type-in request. (RETRY will not be typed out if an *outent* quantifier is specified.) a. If IGNORE. is typed in, processing will continue on the next output reel. A label will be written on the new reel if label writing is specified by the parameters. The count of blocks is reset to 0 unless a sentinel was being written when EOT was sensed. If a sentinel was being written, only the remainder of the sentinel is written before switching tapes. b. If QUIT. is typed in, control will be transferred to the ERRORSUB exit with Index Register 1 set to 13_8 and Index Register 2 set to 4. c. If RETRY. is typed in, processing is the same as for an EOT while writing during a merge pass. All output tapes are rewound with lockout at the start of the new pass.
REPOS RD ERR TAPE *t* or REPOS WR ERR TAPE *t*	Reposition for Retry	Begin Tape, End Tape, S1, or S2 error detected on any tape.	1. REPOS RD ERR TAPE *t* or REPOS WR ERR TAPE *t* is typed out as required, followed by a statement of the ERROR TYPE(s). 2. QUIT, RETRY? is typed out, followed by a type-in request. 3. Further processing is the same as for a Begin Tape, End Tape, S1 or S2 error while reading or writing tape.

Table 5-A. Recoverable Errors and Recovery Procedures Continued

INITIAL TYPE-OUT	SORT PHASE	CONDITION	INITIAL TYPE-OUT AND PROCESSING
TAPE *t* IN LOCAL	Any	Tape in Local	1. TAPE *t* IN LOCAL is typed out. 2. After removing the local condition, the operator may press the ADVANCE bar to continue.
TAPE *t* WR RING	Any	No Write Ring on tape to be written	1. TAPE *t* WR RING is typed out. 2. After installing a Write Ring, the operator may press the ADVANCE bar to continue.
TAPE *t* ROCKED 5	Any	Parity or sprocket error on any tape	1. TAPE *t* ROCKED 5 is typed out. 2. The operator may press the ADVANCE bar to repeat the attempt to read the block in error.

Table 5-A. Recoverable Errors and Recovery Procedures

INDEX REGISTER SETTING (OCTAL)		SUBROUTINE IN WHICH ERROR WAS DETECTED	ERROR TYPE
1X	2X		
1	1	A36KCG	Request for or return of memory not honored by subroutine MEMA
1	2	A36KCG	Generated code does not fit in available memory
1	3	A36KCG	No significant KEY entry in parameter table
1	4	A36KCG	Key word number greater than item size
2	1	A36TG	Request for or return of memory not honored by subroutine MEMA
2	2	A36TG	Generated code does not fit in available memory
3	3	A36SORT	At least one of the following entries is not in the parameter table: ERRORSUB, INSENT, or INFORM
4	1	A36OPT	Request for or return of memory not honored by subroutine MEMA
4	2	A36OPT	Insufficient work tapes or memory space for merge
5	1	A36MERG	Request for or return of memory not honored by subroutine MEMA
6	1	A36IN1	Request for or return of memory not honored by subroutine MEMA
6	3	A36IN1	No INTAPE1 or INTAPE2 entry in parameter table when required at run time
6	4	A36IN1	Label error during presort phase
6	5	A36IN1	Checksum error during presort phase
7	3	A36IN2	No INSUB entry in parameter table although specified at compile time
10	1	A36LP1	Request for or return of memory not honored by subroutine MEMA
10	3	A36LP1	No OUTTAPE2 entry in parameter table when a *signal* quantifier specified as BLKCNT is present
11	3	A36LP2	No OUTSUB entry in parameter table although specified at compile time
12	1	SYSPAR	Request for or return of memory not honored by subroutine MEMA
12	2	SYSPAR	Coding to be generated will exceed available memory
13	4	A36TED	Tape associated error from which recovery is not possible or is not requested.

Table 5-B. Indications to ERRORSUB

The word at location C.PASSIND indicates the current phase of the sort by the contents of its left address field as follows:

Left Address Field	Phase
1	Label Check
2	Prepass (other than label check)
3	Merge (while writing begin sentinels)
4	Merge (not when writing begin sentinels)
5	Last Pass

The right address field contains the number of the current merge pass (0 during Prepass).

Bit position 47 contains a 1 during repositioning of tapes during merge and last pass retries.

Table 5-C. Indication of Sort Phase

Location A36TED.SWORD1 contains the following five-word table of information obtained from XORD at the time of the most recent exit via SETXORD.

Location	Contents
SWORD1	WORD 1
SWORD1+1	WORD 2
SWORD1+2	WORD 3
SWORD1+3	A Register
SWORD1+4	Q Register

For interpretation of the contents, refer to the XORD Program Description (PD-11).

Table 5-D. XORD Information Provided on Error Exit

APPENDIX A

FORMAT OF PARAMETER TABLE SEGMENTS

Whenever parameter information is to be revised or added at run time, the new information must be provided in the form of a run time parameter table segment. Information for the segment is supplied to the sort system via the subroutine designated by the PARSUB parameter (page 37), with each of the new parameters specified as a one-word entry. The formats of the entries are described in Table A-1.* The starting location of this segment must be in Index Register 1 upon return to the sort.

RUN TIME PARAMETER TABLE SEGMENT SPECIFICATIONS

The parameter table segment to be submitted at run time must conform to the following specifications.

- Each parameter must appear as a one-word entry except that:
  1. The INTAPE and OUTTAPE parameters require two entry words if a count is specified.
  2. The WKTAPES, INLAB, and OUTLAB parameters may require several entries. The first entry of each set must be distinguished from the others as indicated in Table A-1.
- An END entry must appear following the last parameter entry. Whenever new segments are added by separate PARSUB entries, the END entry of the previous last segment is automatically changed by the sort system to a NEXTSEG entry. The fact that the END entry may be changed (if more than one segment is added) should be noted if the same table or segment is used for more than one sort in the same program.

A supplementary table segment may contain a PARSUB parameter entry. If this entry is used, the subroutine specified by the parameter will be entered when the sort processes the entry in that supplementary table segment. As this processing takes place before sorting begins, many parameter subroutines may be called during run time even though only one was specified at compile time. At least one PARSUB parameter must be specified at compile time, however, if supplementary parameter table segments are to be used.

The NULL entry provides a means for deleting other entries.

*The format of compile time and run time parameter table entries is exactly the same.

ENTRY	0 — 15	16 — 23	24 — 39	40	42	43 — 47
INFORM	Location of Half Word With Item-Size in Address Field		Location of Half Word With Group-Size in Address Field		A	9
INLAB			Location of Label Word	0	B	6
INSENT	Location of End-of-File Sentinel Word	0	Location of End-of-Tape Sentinel Word*	0		4
INTAPE1	Location of Word With 1st Input Tape Unit Number at T23	0 D E	Location of Word With 2nd Input Tape Unit Number at T23	0	C	17
INTAPE2**			Location of Word With Req'd Input Count at T47	0		18
KEY	Location of Word Containing Mask (See 1)	Key Number	Location of Half Word Which Indicates Word Number and Key Type.†			11
OUTFORM	Location of Half Word With Item-Size in Address Field		Location of Half Word With Group-Size in Address Field			10
OUTLAB			Location of Label Word	0	B	7
OUTSENT	Location of End-of-File Sentinel Word	0	Location of End-of-Tape Sentinel Word*	0		5
OUTTAPE1	Location of Word With 1st Output Tape Unit Number at T23	0 D F	Location of Word With 2nd Output Tape Unit Number at T23	0	C	19
OUTTAPE2**			Location of Word With Req'd Output Count at T47	0		20
WKTAPE		††	Location of Word With Tape Unit Number at T23.	0	B	3
ERRORSUB			Location of Programmer's Error Subroutine			15
INSUB			Location of Programmer's Input Subroutine			12
LABSUB	Location of Label Checking Subroutine		Location of Label Writing Subroutine		G C	8
MERGESUB			Location of Programmer's Pre-Merge Subroutine			14
OUTSUB	Location of Programmer's Output Subroutine		Location of Programmer's Wrapup Code			13
PARSUB			Location of Programmer's Parameter Subroutine			16
END			0			0
NEXTSEG			Location of Next Segment of Parameter Table	0		1
NULL						2

NOTES:
1. Bit positions specified by 1's and 0's must contain the designated number; otherwise an error will occur.
2. In reference to half-word indications, the F-Bit (i.e., Bit 16 or 40) will be checked by the sort to determine whether the left or right half of the word is indicated.
3. Bits in shaded areas are reserved for future use of the sort system.

EXPLANATION OF SYMBOLS

[A] 0 - NOCHKSUM; 1 - CHKSUM

[B] 0 - Not 1st in set; 1 - 1st in set

[C] 0 - Rt.Addr. field not used; 1 - Rt.Addr. field used

[D] 01 - NORWD; 10 - RWD; 00, 11 } RWDLO

[E] 00 - SENT; 01 - ITEMCNT; 10 - BLKCNT; 11 - TAPECNT

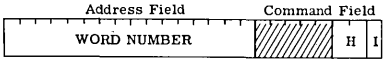
[F] 00 - EOT; 01 - BLKCNT

[G] 0 - Left Addr. field not used; 1 - Left Addr. field used

* If the *endtape* quantifier position is omitted, the right address field should be set to the same value as the left address field (i.e., the same sentinel word is used for both).

** Required only if *signal* is ITEMCNT, BLKCNT, or TAPECNT.

† The format of the key half word is illustrated as follows:



[H] 01 - Floating-Point Key; 10 - Binary Key; 11 - Alphanumeric Key

[I] 0 - No mask required; 1 - Mask required

†† The unit number of a working tape may appear in Bits 20-23 of the WKTAPE table entry itself if desired.

Table A-1. Format of Parameter Table Entries

APPENDIX B

SUBROUTINE DESCRIPTIONS

SORT SYSTEM SUBROUTINES

The following is a brief description of the subroutines within the Philco 2000 Sort System:

A36SORT

Subroutine A36SORT performs the executive function, calling on other subroutines as required. It also contains various constants and temporary parameter information that are accessible to the other subroutines.

A36TEXT

When PARSUB parameters are used, subroutine A36TEXT collects the resulting parameter table segments and appends them to the compile time segment.

A36TG

Subroutine A36TG generates a tree skeleton for the replacement-selection presort phase. The size of the skeleton is variable, depending on the item size and available memory.

A36KCG

Subroutine A36KCG generates a comparison chain from the KEY parameters. The presort and the merge have different generations in the same memory area.

A36IN1

Subroutine A36IN1 reads the input tapes and prepares items for presorting. It also checks labels (or enters the programmer's label subroutine) and verifies checksums when required. This subroutine is not called by the generator if an INSUB parameter is present at compile time.

A36IN2

Subroutine A36IN2 acts as a control buffer between the programmer's input code and other sort subroutines. It is called by the generator only if an INSUB parameter is present at compile time.

A36OUT

Subroutine A36OUT accepts items and writes them on working tapes during the presort pass and all merge passes except the last pass.

A36TREE Subroutine A36TREE controls the presort phase. It accepts items from either A36IN1 or the programmer's subroutine (via A36IN2), and delivers items, in sorted sequences, to A36OUT.

A36OPT Subroutine A36OPT establishes an optimum group size (i.e., minimum of residue words) for use during the merge passes within the limitations imposed by the number of working tapes, item size, and amount of available memory.

A36LP1 Subroutine A36LP1 accepts items and writes them on the output tapes during the last merge pass. It writes labels (or enters the programmer's label subroutine) if required and also writes the sentinel areas and checksums. This subroutine is not called by the generator if an OUTSUB parameter is present at compile time.

A36LP2 Subroutine A36LP2 acts as a control buffer between the programmer's output code and other sort subroutines. It is called by the generator only if an OUTSUB parameter is present at compile time.

A36MERG Subroutine A36MERG controls all merge passes. It reads the current input tapes and delivers items, in merged sequence, to A36OUT during passes other than the last, and to either A36LP1 or the programmer's subroutine (via A36LP2) during the last pass.

TBLSRCH Subroutine TBLSRCH provides a parameter table scanning function for any subroutine that requires it. It can also manipulate some of the entries when obtained from the table.

A36ERR Subroutine A36ERR provides Console Typewriter error typeouts and a transfer of control to 1SUBERR (memory location 3) when entered. The generator provides a parameter in the form ERRORSUB(A36ERR) if no ERRORSUB parameter is specified.

MEMA Subroutine MEMA provides run time memory allocation functions as follows: (1) it allocates memory segments to subroutines as required, maintaining a record of available memory; and (2) it accepts the return of memory segments from subroutines and adds it to its record of available memory.

NON-SORT SYSTEM SUBROUTINES The following TAC subroutines are also used, or are available for use, by the Sort System.

MXORD MXORD is a TAC library subroutine used to execute magnetic tape orders. All Sort System subroutines that process tape call upon MXORD.

MXORD is recommended for all programmer coding that processes tape. This conserves memory space and avoids errors which may arise if more than one tape processing subroutine is used.

SYSPAR

SYSPAR with its two nested subroutines, UNITNO and SYMTAPE, is a prototype run time parameter table segment generator that operates in conjunction with the Philco 2000 Operating System (SYS). It generates the table segment from parameters that are on the SYS input tape in card format.

The programmer may use the SYSPAR subroutine by writing its name as the quantifier of a PARSUB parameter (page 37). (Refer to Appendix C for a complete description of the use of SYSPAR.)

APPENDIX C

FORMAT OF RUN TIME PARAMETERS FOR USE WITH SUBROUTINE SYSPAR

To simplify the generation of a run time parameter segment table with appropriate parameter values, Subroutine SYSPAR (page B-3), in conjunction with the Philco Operating System (SYS), is available to the programmer.

SYSPAR accepts certain sort parameters that are in card format on the SYS input tape, and from this parameter information constructs an acceptable run time parameter table segment.

The parameters that may be specified using SYSPAR are INFORM, INLAB, INSENT, INTAPE, KEY, OUTFORM, OUTLAB, OUTSENT, OUTTAPE, and WKTAPES. Each parameter must appear on a separate card in the following format:

L	Location	Command	Address and Remarks
		IDENTIFIER	$quantifier_1, quantifier_2, \ldots \$$

The last parameter card must be followed by a card with the word END in the Command field.

Quantifier positions have the same significance for run time parameters as for compile time parameters. A comma must be inserted wherever a quantifier is omitted.

The form of the quantifiers at run time may be different, however, from the form of quantifiers at compile time. Table C-1 contains a complete listing of all possible run time quantifier settings for use with SYSPAR.

In general, illegal parameters or quantifiers will cause a SYS Control Line Error to occur, and control to be transferred to Location SYS.1XCONER.

PARAMETER	QUANTIFIER	ACCEPTABLE CONTENTS	MEANING
INTAPE and OUTTAPE	t_1	tape unit*	Tape Unit
	disp	R	Rewind
		L, omitted	Rewind with Lockout
		N	No Rewind
	signal (INTAPE)	ITEMCNT	Item Count
		BLKCNT	Block Count
		TAPECNT	Tape Reel Count
		SENT, omitted	Sentinel
	----or----		
	signal (OUTTAPE)	BLKCNT	Block Count
		EOT, omitted	End-of-Tape
	cnt	1 through 8 decimal digits	Required Count
	t_2	tape unit*	Tape Unit
WKTAPES	t_n	tape unit*	Tape Unit
INFORM and OUTFORM	*item*	1 through 4 decimal digits	Item Size
	group	1 through 4 decimal digits	Group Size
	check	CHECK	Verify Checksum
		omitted	Do Not Verify Checksum
INSENT and OUTSENT	*endfile*	1 through 8 non-break characters**	End-of-File Sentinel
	endtape	1 through 8 non-break characters**	End-of-Tape Sentinel
INLAB and OUTLAB	lab_n	1 through 8 non-break characters**	Label Word

Continued

Table C-1. Run Time Quantifier Settings for SYSPAR

PARAMETER	QUANTIFIER	ACCEPTABLE CONTENTS	MEANING
KEY	*n*	1 through 255 decimal	Key Number
	type	A	Alphanumeric
		B	Signed Binary
		F	Floating-Point
	word	1 through 4 decimal digits	Word in which Key is found
	mask	1 through 16 octal digits***	Mask For Extraction of Key
		omitted	No Mask; Full Word Key

*If Tape Unit is a decimal number from 0 through 15, it is used directly; otherwise, it must be entered via the Toggle Register when the Console Typewriter types:

TAPE *xxxxxxxx*=

where *xxxxxxxx* is the tape unit designation on the parameter card. A unit is assigned by reversing the setting of the toggle corresponding to the unit number.

**If less than eight characters are designated, the sentinel or label will be right-justified with leading zeroes.

***If less than 16 digits are designated, the mask will be left-justified with trailing zeroes.

Table C-1. Run Time Quantifier Settings for SYSPAR (Continued)

SYSPAR EXAMPLE

L	Location	Command	Address and Remarks	
		JOB	XRAY	$(1)
		ABS	4,SORTONE,GO	$(2)
		INTAPE	0,L,SENT	$(3)
		OUTTAPE	OT1,,BLKCNT,5000,OT2	$(4)
		WKTAPES	3,6,9,WT4	$(5)
		INFORM	5,3800,CHECK	$(6)
		OUTFORM	10,120	$(7)
		INSENT	A,B	$(8)
		OUTSENT	FSENT001,TSENT001	$(9)
		INLAB	TAPE1,TAPE2,TAPE3	$(10)
		OUTLAB	JOBONE	$(11)
		KEY	2,B,5,7774	$(12)
		END		$(13)

EXPLANATION

1. The job to be processed is identified as XRAY.

2. An ABS program with an ID of SORTONE is to be located on Tape 4 and run immediately. Revised parameter information supplied by the following SYS cards is to be submitted to the sort program via a PARSUB(SYSPAR) parameter which had been specified at compile time as part of program SORTONE. The information is to be used by SYSPAR to generate a run time parameter.

3. Input Tape Unit 0 is to be processed until an end-of-file sentinel is found, and then be rewound with lockout.

4. Output data is to be written alternately for 5000 blocks on each tape on the two tape units specified by the operator via the Toggle Register, until all output data is written. All units are to be rewound with lockout.

5. Tape Units 3, 6, and 9 and a unit to be specified by the operator are to be used as working tapes.

6. Items of five words, in groups of 3800 words, are to be processed from the input tape. The checksum is to be verified.

7. Items of 10 words, in groups of 120 words, are to be written on the output tape.

8. An input end-of-file sentinel having an A right-justified with leading zeroes, and an end-of-tape sentinel having a B right-justified with leading zeroes are to be used.

9. The output end-of-file sentinel and end-of-tape sentinel are to be FSENT001 and TSENT001, respectively.

10. The input labels are to be TAPE1, TAPE2, and TAPE3, each right-justified with leading zeroes.

11. The output label is to be JOBONE, right-justified with leading zeroes.

12. The second key is to be a signed binary comparison of the first 10 bits of Word 5. (Note that the mask is left-justified.)

13. The END card signifies to SYSPAR that the preceding KEY parameter is the last parameter card to be submitted within the current segment.

APPENDIX D

PARAMETER REFERENCE SUMMARY

PARAMETER REFERENCE SUMMARY

IDENTIFIER	QUANTIFIER	PRESET QUANTIFIER VALUES		PAGE NO.
		PARAMETER SPECIFIED	PARAMETER OMITTED	
INFORM	*(item,group,check)*	10, 120, NOCHKSUM	INFORM(10, 120)	14
INLAB	*(lab_1,lab_2,...,lab_n)*	None	None	16
INSENT	*(endfile,endtape)*	None	INSENT(W/SENTINEL)	18
INTAPE	*(t_1,disp,signal,cnt,t_2)*	0, RWDLO, SENT	None	20
KEY	*(n,type,word,mask)*	1, B, 1	None	22
OUTFORM	*(item,group)*	10, 120	Same as INFORM, if OUTSUB is also omitted	24
OUTLAB	*(lab_1,lab_2,...,lab_n)*	None	None	25
OUTSENT	*(endfile,endtape)*	None	Assumes sentinels equivalent to input sentinels of INSENT	26
OUTTAPE	*(t_1,disp,signal,cnt,t_2)*	9, RWDLO, EOT	None	27
WKTAPES	*(t_1,t_2,...,t_n)*	None	None	29
ERRORSUB	*(ent)*	None	ERRORSUB(A36ERR)	31
INSUB	*(ent)*	None	None	32
LABSUB	*(inent,outent)*	None	None	33
MERGESUB	*(ent)*	None	None	34
OUTSUB	*(itement,wrapent)*	None	None	35
PARSUB	*(ent)*	None	None	37

PARAMETER REFERENCE SUMMARY

IDENTIFIER	QUANTIFIER	PRESET QUANTIFIER VALUES		PAGE NO.
		PARAMETER SPECIFIED	PARAMETER OMITTED	
INFORM	*(item,group,check)*	10, 120, NOCHKSUM	INFORM(10, 120)	14
INLAB	*(lab_1,lab_2,...,lab_n)*	None	None	16
INSENT	*(endfile,endtape)*	None	INSENT(W/SENTINEL)	18
INTAPE	*(t_1,disp,signal,cnt,t_2)*	0, RWDLO, SENT	None	20
KEY	*(n,type,word,mask)*	1, B, 1	None	22
OUTFORM	*(item,group)*	10, 120	Same as INFORM, if OUTSUB is also omitted	24
OUTLAB	*(lab_1,lab_2,...,lab_n)*	None	None	25
OUTSENT	*(endfile,endtape)*	None	Assumes sentinels equivalent to input sentinels of INSENT	26
OUTTAPE	*(t_1,disp,signal,cnt,t_2)*	9, RWDLO, EOT	None	27
WKTAPES	*(t_1,t_2,...,t_n)*	None	None	29
ERRORSUB	*(ent)*	None	ERRORSUB(A36ERR)	31
INSUB	*(ent)*	None	None	32
LABSUB	*(inent,outent)*	None	None	33
MERGESUB	*(ent)*	None	None	34
OUTSUB	*(itement,wrapent)*	None	None	35
PARSUB	*(ent)*	None	None	37

PHILCO®
A SUBSIDIARY OF Ford Motor Company
COMPUTER DIVISION • WILLOW GROVE, PENNSYLVANIA