7090/7094 DATA PROCESSING SYSTEMS BULLETIN

IBM 7090/7094 GENERALIZED SORTING PROGRAM
7090/7094 SORT (729-FIXED LENGTH)

This bulletin describes the fixed-length version of the IBM 7090/7094 Generalized Sorting Program (7090/7094 Sort). Part I, "Introduction and General Principles," discusses the organization and structure of the program, including a description of the sorting and merging techniques used. Part II, "7090/7094 Sort Operations," gives detailed information for using the program, including general specifications, control card formats, tape record format and file structure, and user modification procedures.

The sort program will be extended to incorporate facilities for handling variable-length records at a later time and new documentation will be provided at that time.

It is assumed that the reader has a basic understanding of the 7090 or 7094, especially as regards input/output devices and magnetic tape records; no knowledge of symbolic programming is required except in those cases where it is desired to introduce program modifications. The reader may also need to refer to the following IBM publications:

7090 Operating Systems: Basic Monitor (IBSYS), Form J28-8086

IBM 709/7090 Input/Output Control System, Form C28-6100-2

Minimum Machine Configuration

32K 7090 or 7094

Two channels

Minimum of five 729 tape units (II, IV, or VI)

An on-line printer

(An on-line card reader is desirable but not necessary.)

Address comments regarding this publication to
Programming Systems Publications, IBM Corporation, 1271 Avenue of the Americas, New York 20, N.Y.

Table of Contents

PART I. INTRODUCTION AND GENERAL PRINCIPLES

The IBM 7090/7094 Generalized Sorting Program (7090/7094 Sort) is a modular program designed to offer the user the greatest possible flexibility. The fixed-length version of Sort accepts, as input to be sorted, fixed-length records written in either signed or unsigned binary or BCD mode. These records are sorted using either the commercial or scientific collating sequences, in ascending or descending order. The records may be sorted on any number of control fields, and these control fields may be up to 2,000 words in length (i.e., the maximum size of a tape record allowed by the program).

7090/7094 Sort is designed to operate under the Basic Monitor (IBSYS), which handles the assignment of all input/output components. Specifically, 7090/7094 Sort uses IBNUC, IBSUP, and IOEX; however, the sort program uses its own read/write routines and buffering schemes. (See the IBSYS bulletin.)

PROGRAM CHARACTERISTICS

7090/7094 Sort is composed of separate subprograms (modules), supplied as a series of blocked card images on the Library Tape. Only those subprograms needed for a particular sort run are loaded from the Library Tape at execution time; the others are not loaded. For this reason, each of the subprograms is relocatable.

The program determines which subprograms are required for a given sort run from information supplied in the form of control card parameters. These control cards are described in full in Part II of this manual.

The modular structure of Sort allows the user great flexibility in introducing routines into the sort program to permit such operations as summarization, deletion, lengthening, and shortening of records. These routines must be located on the Library Tape and are loaded and handled in the same manner as the subprograms supplied as 7090/7094 Sort. (See "User Modifications to Sort," page 34.)

The Merge Program

Sort incorporates a complete merge program, thus making a separate merge program unnecessary. When a merge run is executed, only those subroutines that constitute the merge program are loaded from the Library Tape. These same subroutines are used during a sort run, as part of the sort program, to merge strings of sorted records. Furthermore, it is possible in the course of a sort run to merge previously sorted records with records currently being sorted.

The order of merge to be used (i.e., the number of sequences merged concurrently) is determined by the user, usually on the basis of tape unit availability. An increase in the order of merge is advantageous when it serves to reduce the number of merge passes necessary to complete the sort.

Tape Usage

Sort requires 2M+2 tapes, where M is the order of merge. The two additional tape drives are for the Library Tape (SYSLBx), which is necessary, and for the Unreadable Records Tape (SYSCK2), which is optional. If no SYSCK2 is attached, unreadable records and dictionaries, if any, will be deleted.

Tape Labels

Sort uses the standard labeling conventions of 709/7090 IOCS (see the IOCS reference manual) but provides the option of using non-standard labeling procedures. (See "File Structure and Tape Record Format," page 26.)

Optional Program Features

The following options, which are specified by parameters in a control card (see page 23), are available to the user during execution of a sort and/or merge run:

1. Checksums. If desired, a checksum will be computed for each logical record of a sort and carried for the entire sort. It is then dropped before the final output is written on tape.

2. Short tape blocks. If this option is specified, short tape blocks will be accommodated by the program provided their length is a multiple of the length of the logical input record.

3. Maintaining equal records. If desired, any equal records encountered by the sort will be written in the output file in the same order as they appeared in the input file.

4. No checkpoints. Checkpoints are normally written by Sort (see page 32), but can be omitted if desired.

5. Buffers. Normally, fixed-length records are "scatter read" into core storage; this option allows the reading of records into buffers instead, to provide access to the records for modification.

6. Relocate COMMON. This option provides for relocating in core storage the area reserved for the common parameters of the program.

7. Deletions. If desired, records may be selected and deleted from the file on the basis of control card information.

7090/7094 SORT PHASES

Sort is divided into four distinct phases -- the Edit Phase, the Internal Sort Phase, the Merge Phase, and the Final Merge Phase. They are described below.

In addition to the four phases, the program contains the Sort Monitor, which communicates with IBSYS and initializes the sort program, and the Post-Processor, which prints out information on the completed sort run and then returns control to the Sort Monitor.

The Edit Phase

The Edit Phase is the first phase of Sort. It is called by the Sort Monitor, and performs the following functions:

1. Calculation of Internal Parameters

   The internal parameters for use by the other three phases are calculated by Edit Phase and placed into relocatable tables (the COMMON area, see page 41) for later use.

2. Building of Program List

   Edit Phase determines which of the subprograms on the Library Tape will be required in subsequent phases, and the names of these subprograms are placed in respective program lists for the Internal Sort, Merge, and Final Merge Phases. These program lists will later be used by each phase to call the necessary subprograms for that phase.

3. Card Analysis

   During Edit Phase, the control cards are loaded into storage and the parameters are analyzed. Card analysis consists of the following:

   a. The control cards are checked for any errors that may have resulted from mispunching.

   b. The control cards are tested to see that all required parameters are supplied and in correct form.

   c. Messages are printed on the on-line printer to inform the operator of any errors or possible errors in the control cards.

d. Once checked for corrections, the information is placed in relocatable tables.

The Internal Sort Phase

The Internal Sort Phase is a one-pass phase. Its function is to arrange the input records in sequenced strings one or several output blocks long. (The output block is as large as can be accommodated by the Merge Phase.) The output of this phase is divided among the number of tapes represented by the order of merge.

The method used by this phase in producing the sequenced strings is a modified continuous merge. A number of records is read in to fill a record storage area, and these records are sorted in this area. Then a smaller number of records is read into an auxiliary area, sorted, and merged with the sorted records in the record storage area. A comparable number of sorted records is now selected and written as output, and another small number of records is read in to continue the process.

Unless the option of buffering is requested, fixed-length records are read into the record storage area by a Scatter Read procedure and are not moved during the internal sort processing. Instead, the sorting and merging is done by use of tag tables. When they are selected to be output in a string, the records are written out by a Scatter Write procedure.

The selection of records to be written out is made in such a way as to produce the longest possible strings. This internal sort method takes advantage of any sequences already existing in the file to produce unusually long strings. For random records, strings about twice the size of the record storage area can be obtained. Since the number of strings formed in a given file varies inversely with the length of the strings, longer strings definitely reduce the number of merge passes necessary to complete the sort.

The Merge and Final Merge Phases

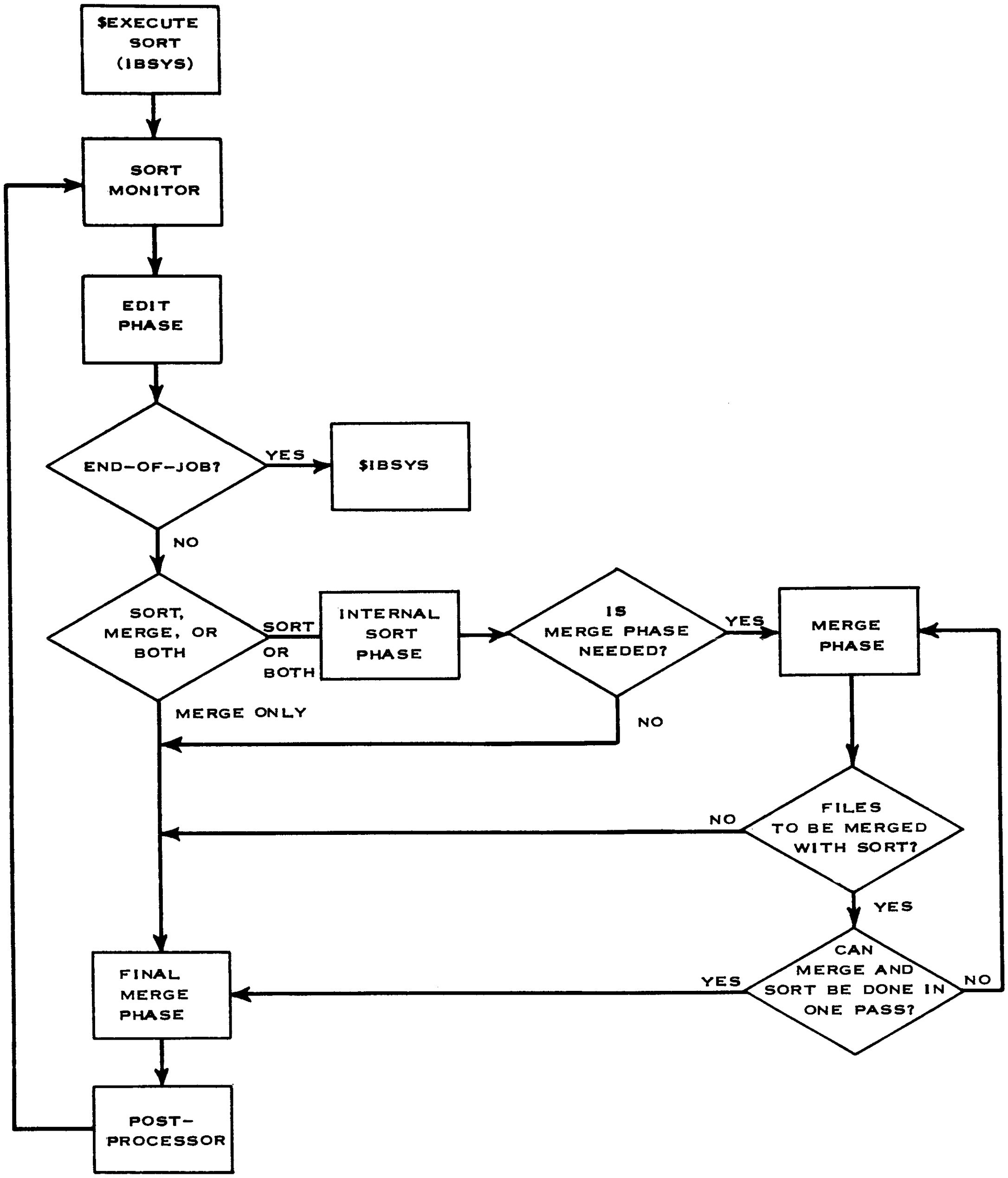
The Merge and Final Merge Phases are designed to handle as high an order of merge as the hardware configuration will allow. Generally, the higher the order of merge, the fewer the merge passes required. An increase in the order of merge, therefore, frequently reduces the total execution time of a sort or merge run. On the other hand, increasing the order of merge reduces the size of the tape block for the same available memory locations. If the size of a tape block is too small, the input/output operation is slowed due to the inter-record gap time. The block size is calculated by the Edit Phase in consideration of available storage and order of merge.

The timing of the merge phases is usually the read/write or tape movement time, since processing (the internal merging of the records) can usually by done well within the tape time. The phases are designed so that the reading and writing of records is continuous, with no time loss due to waiting.

The records in the input blocks have already been arranged in sequence by the Internal Sort Phase. During the Merge Phase, the sequenced strings are merged together in succeeding passes until the number of strings is equal to or less than the order of merge. Then the Final Merge Phase is loaded and the last merge pass is performed. The final output is written out according to the format specified by the control information.

The Final Merge Phase also handles the merging of previously sorted files, if any, with the records being sorted.

System Flow of Control

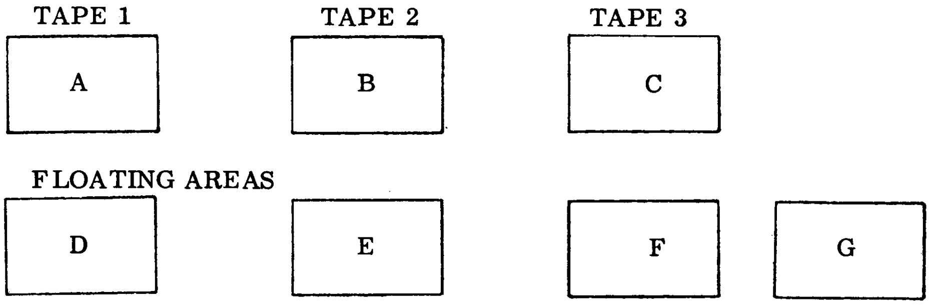


Storage Allocation in Merge Phases

In order to obtain a maximum overlap of reading, computing, and writing operations in the merge phases, the following technique of assigning record storage areas is used:

For a balanced merge (i.e., with one channel used for input and the other for output) of merge order M, 2M+1 storage areas are needed, each equal in length to the maximum number of words in a tape record on the merge tapes. At least one of these areas is assigned to each tape at any given time, with the extra or "floating" areas assigned as needed to the tape with the highest reading priority, as discussed below. An area becomes a floating area as soon as it is released from its previous assignment.

Consider the case of a 3-way balanced merge. The number of areas needed is 7. If these are designated by the letters A-G, the Merge Phase starts with the following assignments:

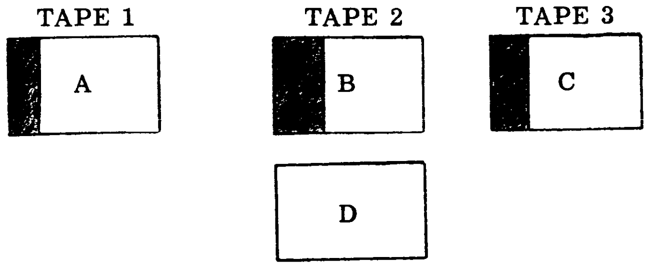


Initially, one tape record is read from each of the three merge input tapes, filling areas A, B, and C. (At this point, areas D-G are floating.)

A comparison is made among the control fields of the last records in A, B, and C. Reading is started into area D from the tape associated with the area having the lowest control field in its last record. In the case of equal control fields, the tape with the lower number is given reading priority.

Merging of the records in A, B, and C is begun and continues until enough have been merged to fill an output block, at which time writing is begun. If necessary, processing is then delayed until the previous read is completed (D is filled).

At this point, the record storage areas in use might look like this:



The shaded areas indicate the records already merged and now being written out.

Assume that area D was assigned to tape 2. A test is made to see if there is a stepdown between areas B and D (indicating that a new sequence has been started in D). If not, the reading priority is established between the last records of A, D, and C, and reading is started into area E from the tape with highest priority.

Merging the records in A, B, and C continues until there are enough to fill another output block. The previous write is checked for completion and, when complete, a new write is begun.

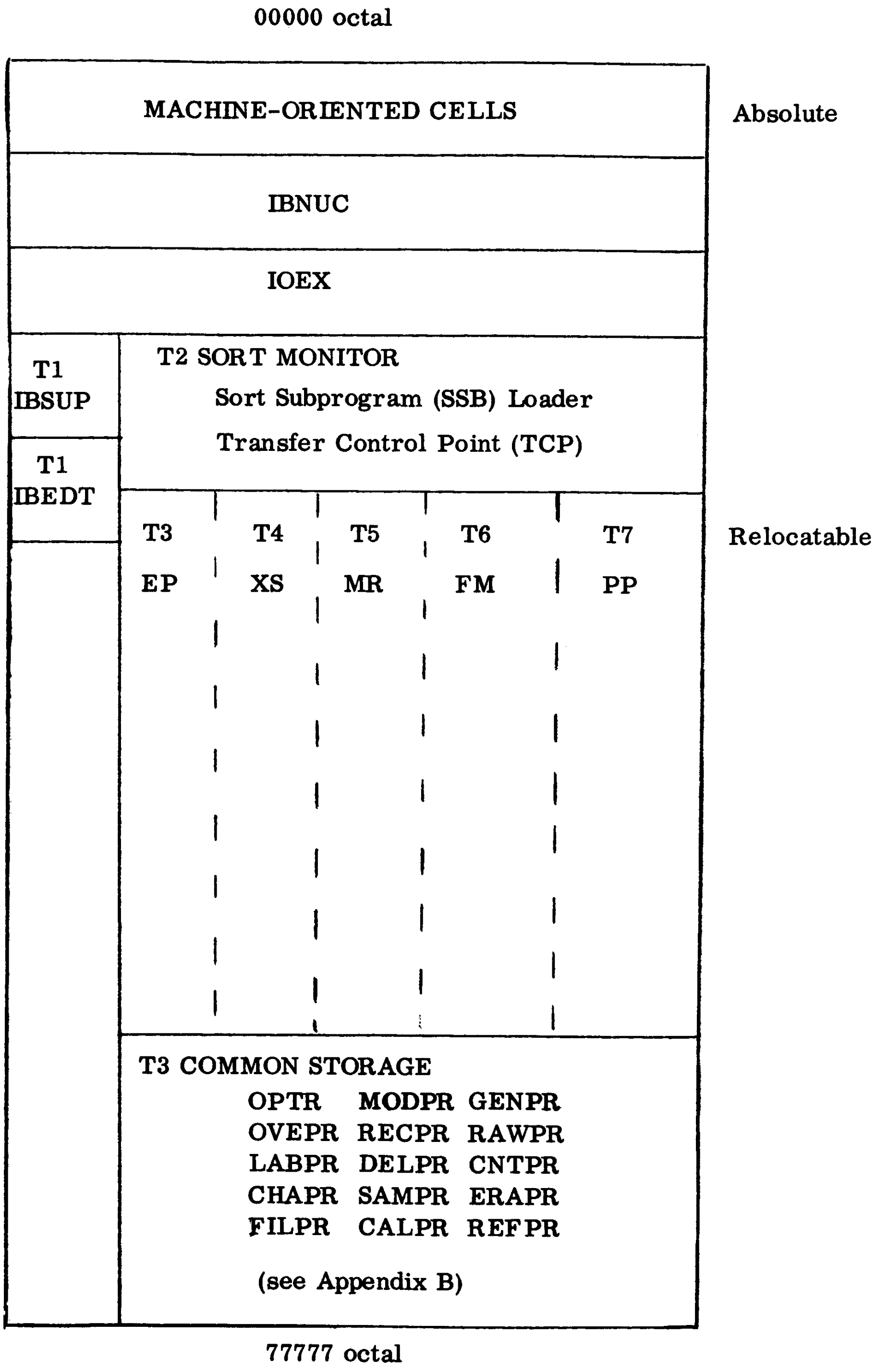
Eventually there will be a stepdown in one of the files. In this case, the area with the new sequence is not used for merging and simply stands by until all files have had a stepdown. There will be three areas in use, each with a block from the new sequence, and the entire process is repeated.

In the final merge pass, the number of record storage areas is modified, if necessary, to accommodate any output blocking size without a special re-blocking pass. When the output blocking is equal to or less than the merge blocking, the same 2M+1 areas are used. If, however, the output blocking is greater than the merge blocking, the number of areas can be represented by the expression

$$N = 2M + 1 + \frac{2(OB-MB)}{MB}$$

where N is the number of record storage areas,
M is the order of merge,
OB is the output blocking, and
MB is the merge blocking.

CORE STORAGE LAYOUT



00000 octal

MACHINE-ORIENTED CELLS

Absolute

IBNUC

IOEX

T1 IBSUP

T1 IBEDT

T2 SORT MONITOR

Sort Subprogram (SSB) Loader

Transfer Control Point (TCP)

T3 EP

T4 XS

T5 MR

T6 FM

T7 PP

Relocatable

T3 COMMON STORAGE

OPTR MODPR GENPR
OVEPR RECPR RAWPR
LABPR DELPR CNTPR
CHAPR SAMPR ERAPR
FILPR CALPR REFPR

(see Appendix B)

77777 octal

Note: Origin at upper limit is fixed, but may be changed using the OPTION Card.

In the diagram above, the symbols T1-T7 indicate the time sequence of the various stages of Sort. The symbols are explained as follows:

T1 - Calling of 7090/7094 Sort

Sort is called by the Basic Monitor (IBSYS) through a $EXECUTE SORT control card (see page 14). IBSYS defines the computer and provides the necessary information on the availability of input/output components. All I/O assignments are a function of IBSYS.

T2 - Loading of Sort Monitor

IBSYS loads the Sort Monitor. At this point, control passes to Sort through the Sort Monitor. The Monitor includes the Sort Subprograms Binary (SSB) Loader, which loads the subprograms proper to each phase.

T3 - Loading of Control Cards and Execution of Edit Phase (EP)

The control cards are always loaded from SYSIN1, the symbolic designation of the IBSYS input unit. Edit Phase is then executed. The parameters needed to execute the sort are generated from the control information and placed in COMMON storage starting at location 77777_8. The subprograms loaded in this phase are as follows:

ASSIGN
EP001
SOP
RB01
RELEAS
RESTAR
SK001

T4 - Loading and Execution of Internal Sort Phase

The Internal Sort (XS) subprogram is loaded over Edit Phase, along with the subprograms required by this phase. The area in storage used for sorting extends from the first word after the last subprogram loaded to the last cell before COMMON storage. The subprograms loaded in this phase are as follows:

IOBS*
CKPT
CKSUM*
BCONV**
DEBLK
DELETE*
DEPAD
EQUALS*
FXMOV
LABEL
LOCATE**
MOVE**
SOP
RESTAR
IOSS*
WTFIX
WRSEL
XSLOG
XSMIX

*optional
**variable-length records only

T5 - Loading and Execution of Merge Phase

The Merge Phase (MR) subprogram is loaded over the Internal Sort Phase, along with the subprograms necessary to this phase. These subprograms are as follows:

BTD
IOBS
CKPT
CKSUM*
DEBLK
LEQ
XTR
LABEL
MRL
MR
SOP
RESTAR
WRSEL

*optional

T6 - Loading and Execution of Final Merge Phase

The Final Merge Phase (FM) subprogram is loaded over the Merge Phase. This phase is identical to the Merge Phase except for its initialization procedures. During this phase, which is effectively a separate merge program, previously sorted files may be merged with the file currently being sorted. The subprograms loaded in this phase are as follows:

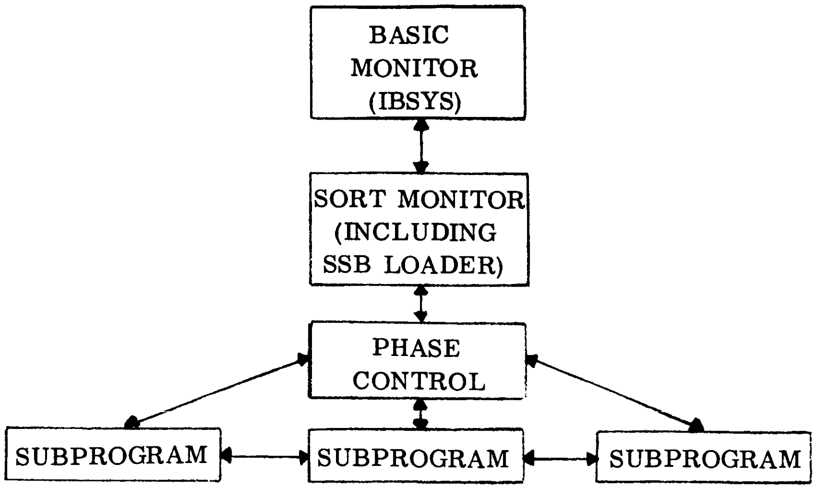
BTD
IOBS
CKPT
CKSUM*
DEBLK
DELETE*
LEQ
XTR
FM
LABEL
MRL
SOP
RESTAR
WRSEL

*optional

T7 - The Post-Processor (PP)

At this stage the sort run is completed and control passes from the Final Merge Phase to the Post-Processor. The Post-Processor releases all tape assignments; prints counts of records sorted, records deleted, records dumped, padding records added, reserve units used for inter-system communication; and then returns control to the Sort Monitor. The Sort Monitor reloads the Edit Phase and the next control card is read. If the next control card is a $IBSYS card, control is returned to IBSYS. If another set of Sort control cards is read, sorting is resumed.

PROGRAM COMMUNICATION



1. Inter-job control passes to 7090/7094 Sort through the Sort Monitor.

2. Inter-phase control is a function of the Sort Monitor and the control program for a particular phase.

3. Intra-phase (or inter-subprogram) control is a function of the Transfer Vector technique of 7090/7094 Sort. Subprograms can communicate with each other and with the control program for any of the Sort phases. No distinction is made between a user subprogram and a Sort subprogram.

PART II. 7090/7094 SORT OPERATIONS

GENERAL SPECIFICATIONS

Input Records

Input to 7090/7094 Sort (fixed-length version) is fixed-length records written in either BCD or binary mode, either signed or unsigned. The maximum allowable size of the input records or blocks is 2, 000 words. Minimum size is 3 words, in keeping with tape record error recovery conventions now in use on the 7090. The records must be grouped into blocks of equal size except where the VARIABLE BLOCKING option is specified (see page 23).

High or low padding may be used to fill out any blocks of an input file. The output file may also be padded either with high padding at the beginning or low padding at the end, or left unpadded, as desired.

Control Fields

The fields of a record and control field information may be expressed in either bits or characters. If control fields are to be sorted in accordance with the commercial collating sequence, the control fields may be expressed in either characters or bits, but if in bits, the number specified must be a multiple of six.

It is possible to intermix, in the same record, control fields to be sorted in ascending sequence with fields to be sorted in descending sequence. For this reason, the following convention must be observed:

> Corresponding control fields (i.e., the same field from record to record) must all be signed or all be unsigned, and must be sorted in the same sequence.

There is no restriction on the number of control fields or their length. However, the number has been effectively set at 200 for the distributed version of the program. If necessary, this number can be increased by changing, in the Edit Phase, BES 200 to BES xxxx, where xxxx is the number of control fields desired, at the following symbolic locations:

FILEN	FILLB
FIEUS	FILNW
FILSW	FILLM
FILSB	FILRM

BASIC MONITOR CONTROL CARDS

Two Basic Monitor (IBSYS) Control Cards are required to use Sort, as follows:

1. $EXECUTE SORT

   This card calls Sort and causes control to be transferred to it from IBSYS.

2. $IBSYS

   This card transfers control from Sort to IBSYS and causes the Basic Monitor Supervisor (IBSUP) to be restored from tape.

For complete information on these cards, see the IBSYS reference manual.

SORT CONTROL CARDS

In order for a file to be sorted, certain information must be supplied to the sort program. This includes a description of the logical records, of the input and output files, and the control fields on which the records are to be sorted. Other information for use in options, labels, and modifications may also be needed. Thirteen control cards have been designed with this functional division in mind, as follows:

Card	Description
RECORD	Defines logical record(s)
FILE	Defines file characteristics
SORT	Specifies sort
MERGE	Specifies merge
CHANNELS	Specifies input, merge, and output channels
LABEL	For use with non-standard labels
OPTION	Lists options desired
MODIFICATION	Specifies names of modification programs
DELETE	Specifies records to be deleted from a file
OVERFLOW	For use with an overflow sort
RESTART	Initializes restart procedure
REMARK	A remarks card
END	Indicates end of control card deck

In a normal sort run, only the RECORD, FILE, SORT, CHANNELS, and END Cards are needed.

For maximum legibility and to eliminate the necessity for punching those fields not actually required in a particular sort run, the control cards have a variable-field format with field identifiers. The fields are separated by commas. Information may be punched anywhere in columns 7-72; if a statement is too long to fit on one card, it may be continued on as many additional cards as necessary. Continuation cards are indicated by an "X" punched in column 6; thus, the first card of each statement must not be punched in column 6. Columns 1-5 and 73-80 are ignored by the program and may be punched as desired. In all fields except those for which a character count is given, i.e., Hollerith information, blanks are ignored by the program and may be included as desired for legibility.

It is not necessary that the control statements (i.e., a control card and its continuation cards, if any) be arranged in any specific order, except that the END Card must be the last control card. In like manner, the parameters within a control statement may be supplied in any convenient order, provided the first word of the statement is the statement identifier.

To simplify punching, only the first three characters of any field or statement identifier need be punched. For example, RECORD may also be punched as REC.

Detailed descriptions of the control cards follow:

FILE Cards

There are FILE Cards in the Sort program for both input and output functions. The FILE Card for input contains the descriptive information about the physical characteristics and organization of the data which is necessary to read the file into core storage from the input unit. The FILE Card for output contains the information from which the processed data will be organized into a file on the output unit.

Input FILE Card

The complete format for the Input FILE Card is as follows:

```
FILE, INPUT/#, REELS/n, MODE/B or D, DENSITY/H or L,
                                                    S
   RECORDS/n, BLOCKSIZE/n, PADDING/H or L, LABEL/ or
   H                                                N
   or, SERIAL/#, RLSEQ/#, NAME/xxH----, CKPṪ/S or N,
   L
```

CKSUMS, BLKSEQ, DICT

where

INPUT/# = Logical number assigned to the input file, to be used whenever this file is referred to, as in the SORT statement.

REELS/n = Number of reels contained in the file. May be omitted in the case of a one-reel file.

MODE/B or D = Mode of the file, binary or BCD. If a file carries a standard label, this field may be omitted.

DENSITY/H or L = Density of the file, high or low. If a file carries a standard label, this field may be omitted.

RECORDS/n = Number of logical records in the file. If this field is omitted, full reels are assumed in calculating the capacity of the sort.

BLOCKSIZE/n = The number of words per tape block, or the maximum tape block size in words if variable-length records. NOTE: This number does not include the count of the checksum or block sequence word in the block.

PADDING/H or L = Padding, high or low, in the case of fixed-length records only.

LABEL/$\overset{S}{\underset{N}{\text{or}}}$ $\overset{H}{\underset{L}{\text{or}}}$ = The file carries a standard label in high or low density, or a non-standard label in high or low density. The four possibilities are SH, SL, NH, and NL. If the file carries no label, the field LABEL should be omitted.

SERIAL/# = File serial number. This field is optional and is omitted if the file is not labeled.

RLSEQ/# = Reel sequence number of the file. This field is optional and is omitted if the file is not labeled.

NAME/xxH---- = Name of the file. The xx represents the number of Hollerith characters, including blanks, in the name. This number must not exceed 18. The name of the file immediately follows the character H.

CKPT/S or N = If S (for Standard), the file carries a checkpoint on every reel except the first; if N (for Non-standard), the file carries a checkpoint on every reel including the first.

CKSUMS = When specified, a half-word checksum is carried in each tape block. It is permitted in binary files only..

BLKSEQ = When specified, a half-word block sequence is carried in each tape block; for binary files only.

DICT = When specified, a dictionary is carried in the file.

If the standard labeling conventions of IOCS are used, some of the above information is given in the label and need not be included in the FILE statement, i.e., MODE, DENSITY, CKPT, CKSUMS. In addition, SERIAL number, NAME, and RLSEQ number are given in the label and are checked by Sort only if they are included in the FILE statement. If they are not included, they are not checked.

Thus, the Input FILE statement may be considerably condensed. For a one-reel file, it may be as concise as follows:

```
                                           H
FILE, INPUT/#, BLOCKSIZE/n, LABEL/S  or
                                           L
```

or even,

```
                          H
FIL, INP/#, LAB/S  or, BLO/n
                          L
```

Output FILE Card

The complete format for the Output FILE Card is as follows:

```
FILE, OUTPUT, MODE/B or D, DENSITY/H or L, BLOCKSIZE/n,
                                      S    H
      PADDING/H or L, LABEL/  or   or , SERIAL/#, RLSEQ/#,
                                      N    L

      NAME/xxH----, RETAIN/n, CKSUMS, BLKSEQ  DICT
```

where

MODE/B or D = Mode, binary or decimal, in which the output file will be written.

DENSITY/H or L = Density, high or low, of the output file.

BLOCKSIZE/n = Words per tape record.

PADDING/H or L = For fixed-length files only; the first or last tape record may be padded, high or low, if desired; if this field is omitted, no padding will be added to the output file by the program.

LABEL/ $\begin{matrix}S\\ \text{or}\\ N\end{matrix}\;\begin{matrix}H\\ \text{or}\\ L\end{matrix}$ = To indicate that a file is to have a standard or non-standard label, in high or low density. The four possibilities are SH, SL, NH, and NL. If no label is desired, this field may be omitted.

SERIAL/# = File serial number. If standard label is specified, this field should be included.

RLSEQ/# = Reel sequence number. If standard label is specified, this number is used as the initial reel sequence number. If this field is omitted, the reels are sequenced starting with the number 1.

NAME/xxH---- = Name of the output file. The xx represents the number of characters, including blanks, in the name. This number must not exceed 18. The name of the file immediately follows the character H. This field is only for files with standard labels.

RETAIN/n = Number of days the file is to be retained. This field is only for files with standard labels.

CKSUMS = When specified, a half-word checksum is carried in each tape block; for binary files only.

BLKSEQ = When specified, a half-word block sequence word is carried in each tape block; for binary files only.

DICT = When specified, a dictionary is to be carried in the output file; permitted only if input file carries a dictionary.

If the output file is not to be labeled, the Output File statement may be reduced to the following:

```
FIL, OUT, MOD/B or D, DEN/H  or L, BLO/n
```

RECORD Card

The general format of the RECORD Card is as follows:

```
RECORD, TYPE/F or V or xx, LENGTH/(nL1, nL2, nL3),
            B  U     B  U          B  U
    FIELD/(n1 or  or, n2 or  or,...,ni or  or )
            C  S     C  S          C  S
```

where

TYPE/F or V or xx = Fixed- or variable-length record, or record classified by type. The last mentioned pertains to a file which has more than one record type, the records being fixed within types but not between types. (In the fixed-length version, only the F is acceptable.)

LENGTH/(nL1, nL2, nL3) = Length of logical records in words; where nL1 is input record length, nL2 is the record length of the input to the Merge Phase, and nL3 is the record length of the output from the Final Merge Phase (nL2 and nL3 are used when the length of records will be modified during the sort).

```
        B  U     B  U         B  U
FIELD/(n1 or  or, n2 or  or,...,ni or  or ) = The fields of a
        C  S     C  S         C  S
```

logical record; each field is expressed in bits or characters, signed or unsigned.

The following is a description of a fixed-length record:

```
RECORD, TYPE/F, LENGTH/12, FIELDS/(12, 2, 12,
    4, 5, 6, 7, 8)
```

This statement identifies a fixed-length record of 12 words whose first eight fields are 12, 2, 12, 4, 5, 6, 7, and 8 unsigned characters, respectively.

Characters are assumed to be the standard units of measurement. If neither B nor C is given, C is assumed.

If a field is unsigned, it is not necessary to write the "U" character in the field description of a RECORD Card. All signed fields, however, must carry an "S" designation after the length of the field.

Note that it is necessary to specify fields in the RECORD statement only until all control fields and any fields that are to be used as a basis for deleting records (see DELETE Card) have been included within the parentheses.

Example:

Suppose that it is desired to sort on only the first and eighth fields, in that order, and both of the fields are signed. The RECORD statement above can then be written

```
REC, TYP/F, LEN/12, FIE/(12S, 36, 8S)
```

It is not necessary to write "U" or "S" in the description of fields which will not be used as sort keys. When such fields are consecutive, it is desirable to specify them as one field, as in the above example.

SORT and MERGE Cards

The general format of the SORT statement is

SORT, FILE/#, SEQUENCE/C or S, ORDER/n,

FIELDS/($\#1^{A}_{D}$, . . . , $\#i^{A}_{D}$)

where

FILE/# = The number of the file, as given in the FILE statement.

SEQUENCE/C or S = The collating sequence to be used: C (for Commercial, EAM, 705) or S (for Scientific, 7090). If this parameter is omitted, S is assumed.

ORDER/n = The order of merge.

FIELDS/($\#1^{A}_{D}$, . . . , $\#i^{A}_{D}$) = Specifies the control fields, or sort keys, by which the records are to be sorted, arranged from high to low order. The A and D indicate that a particular field is to be sorted in Ascending or Descending order. If this designation is not included, A is assumed.

The general format of the MERGE statement is

MERGE, FILES/###, SEQUENCE/C or S, ORDER/n,

FIELDS/($\#1^{A}_{D}$, . . . , $\#i^{A}_{D}$)

If both sort and merge are required in a single run, then both SORT and MERGE statements must be used, except that the MERGE statement can be abbreviated by writing the file numbers only, and eliminating the rest of the fields. For example,

```
SORT, FILE/12, SEQUENCE/S, ORDER/5, FIELDS/(5,1,4,3,2)
MERGE, ORDER/5, FILES/(2,3)
```

Note: The same field may serve as a control field more than once. In the above SORT and/or MERGE statement, for example, it is permissible to specify the control fields as

```
FIELDS/(5,1,4,5,2)
```

or any desired combination thereof.

CHANNELS Card

The CHANNELS Card is used to specify the channel on which the input to be sorted or merged is mounted and the channels that are to be used for merging, and may be used to specify the output channel. Actual physical units are determined through the monitor system and its availability table. The form of the card is as follows:

CHANNELS, INPUT/X, MERGE/(Y_1 , Y_2), OUTPUT/Z

where

INPUT/X = Channel on which input is mounted. Only one channel may be specified. If it is specified as one of the eight characters A-H, that channel is selected. If it is specified as one of the characters J-Q, the input channel is obtained through the Basic Monitor.

MERGE/(Y_1, Y_2) = Channels to be used for merging. One of the Y_i may be the same as the input channel, if desired.

OUTPUT/Z = Channel on which output is written. If this parameter is used, and Z is specified as one of the eight characters A-H, output is written on that channel and the output tapes are rewound. (Note that if Z is not the channel on which output would normally be written, time will be lost in the execution of a copy pass.) If Z is specified as one of the characters J-Q, the output tapes are rewound and the symbolic designation of the tape unit is stored in the address portion of the first unit control word of the unit control block for that tape unit. This designation will then be used by the next system. If this parameter is omitted, output is written on the channel where it would fall in the course of execution and the output tapes are rewound and unloaded.

For example, the card

CHANNELS, INPUT/A, MERGE/(A,B)

means that the input to the sort is on Channel A, the merge channels are A and B, and the output tapes will be rewound and unloaded.

If a SORT is being run, the Sort Monitor assigns one or two drives as the input drives on the channel specified on this card, the number depending on whether the input file is on one reel or is multi-reel; if multi-reel, two drives are always assigned. A message is printed telling the operator the physical tape unit assignment that has been made. If a MERGE is being run, the monitor assigns as many drives on the input channel as there are input files and prints a message to this effect. If the run is a sort and merge run, only the selection of the channel for the sort input file can be made on the CHANNELS Card. The channel for the merge input files is decided by the program just prior to the final merge pass, and a message will be printed.

Unreadable records will be written out on the unit designated by SYSCK2 in the Basic Monitor system unit table.

LABEL Card

The LABEL Card has the following format:

LABEL, IDENT/xxH---- $(18 \leq xx \leq 84)$

where

IDENT/xxH----= indicates the length in characters of the label to be written on the output tape and must be a multiple of six. The actual label is written following the H. If the number of characters in the label is greater than the xx specified, then the extra characters are truncated. If the number of actual characters is less than that specified, then the rest of the label is filled to the right with blanks.

Use of the LABEL Card is permitted only when a non-standard label is specified on the Output FILE Card. If a non-standard label is specified and the LABEL Card is not present, it is assumed that the same label that is on the input file is to be written on the output file.

OVERFLOW Card

The format for the OVERFLOW Card is the following:

```
OVERFLOW, BLOCKS/n
```

where

BLOCKS/n = The number of tape records of the current input tape that have been processed. This number does not include any label, checkpoint, or dictionary records. The number of records processed is expressed in decimal.

Example:

```
OVERFLOW, BLOCKS/521
```

This statement indicates to the sort program that 521 tape records of sort information were read before the overflow occurred.

OPTION Card

The following is the format for the OPTION card:

```
OPTION, CKSUMS, NOCKPT, EQUALS, RELCOM/n, BUFFER,
     VARIABLE BLOCKING
```

where

CKSUMS = This option, when specified, causes a checksum to be computed for each logical record and carried along for the entire sort. The checksum is dropped before the final output is written on tape.

NOCKPT = When NOCKPT is specified it indicates that no checkpoints are to be written during the sort; therefore, Sort will not write the normal checkpoints. If there is no tape assigned as SYSCK2, the NOCKPT option will be assumed. Note that if no checkpoints are taken, restart is impossible except from the beginning of job.

EQUALS = This option, when specified, will instruct Sort to keep all equal records in the same order as they appear in the input file.

RELCOM/n = This option allows for the possibility of reducing the size of core by the number of locations specified. Memory would be reduced by taking the locations away from the top of core and relocating COMMON downward. (See COMMON, page 41 .)

BUFFERS = The buffer option allows for fixed-length records to be read into buffers instead of being scatter read into core storage. This option could be used if some work were to be done on the records before they are sorted.

VARIABLE BLOCKING = The option, when specified, allows a short block on the input tape, provided its actual length is a multiple of the length of the logical record.

Example:

Suppose that a user wishes to reserve 1200 locations at the top of core storage for his own use and that he also desires to keep all equal records in the same order as they appear in the input file. In this case, the OPTION statement is

```
OPTION, RELCOM/1200, EQUALS
```

MODIFICATION Card

The following is the format for the MODIFICATION Card:

```
MODIFICATION, PROGRAM/xxxxx, CELLS/n
```

where

PROGRAM/xxxxx = This designates the modification. The xxxxx portion must conform to the modification names defined by Sort. 25 names are available for modifications by this card: 10 names can be used in the Internal Sort Phase, 5 names in the Merge Phase, and 10 names in the Final Merge Phase. The names will have the following format:

Internal Sort Phase	XSM01 through XSM10
Merge Phase	MPM01 through MPM05

Final Merge Phase	FMM01 through FMM10

CELLS/n = This is used to designate the number of core locations the program and its parameters will occupy; n must be a decimal integer.

Example:

MOD, PROGRAM/XSM03, CELLS/1218

This statement indicates to the program that there is a modification in the Internal Sort Phase which has the name XSM03. When this name is given, the Edit Phase sets an indicator that XSM03 is to be entered at the appropriate time. The modification occupies 1218 cells. (See the section entitled "User Modifications to Sort.")

DELETE Card

The DELETE Card has the following format:

DELETE, TYPE/xx, FIELD/#, IDENT/n$\overset{O}{\underset{H}{B}}$----

where

TYPE/xx = This is to specify the particular record type number to be deleted. It can also contain an F for fixed-length records or a V for variable-length records. If TYPE is either F or V, this parameter may be omitted. (In the fixed-length version, only the F is acceptable.)

FIELD/#= This is used to specify the field in the record in which the bit pattern will be compared with that given in the IDENT field. Upon an eoual comparison, the record will be deleted.

IDENT/n$\overset{O}{\underset{H}{B}}$---- = This is the specific information used to determine if a record will be deleted; n is the number of bits, characters, or octal numbers that follow; O, B, and H stand for octal, binary, and Hollerith information, respectively. If the given information is found in the specified field in the given record type, then that particular record is deleted from the sort. The given information must cover the full field. The field length, of course, is described in the RECORD Card.

Example:

```
DELETE, TYPE/F, FIELD/2, IDENT/1HZ

DELETE, TYPE/05, FIELD/4, IDENT/1HA
```

The first statement directs the sort to delete all fixed-length records having a Z in field 2. The second statement directs the sort to delete all type 05 records having an A in field 4.

Note: If only the record type is given, then the records of that type are deleted whenever they are encountered. For example, the statement
DELETE, TYPE/09
directs the sort to delete all type 09 records from the file.

RESTART Card

The RESTART Card has the following format:

RESTART

The RESTART Card initializes a restart procedure (see "Checkpoint and Restart Procedure," page 32).

FILE STRUCTURE AND TAPE RECORD FORMAT

7090/7094 Sort accepts as input both signed and unsigned binary or BCD files. Minimum size of a record in words is 3; maximum is 2,000.

The output file of 7090/7094 Sort need not be in the same mode or density as the input file. (Note the implications of recording random binary information in the BCD mode as given in the IBM Reference Manual, 7090 Data Processing System, Form A22-6528, section entitled "Character Alteration in the BCD Mode.")

Tape labels must be in the BCD mode and must be not less than 3 and not more than 14 words in length. The program uses the standard labeling conventions of 709/7090 IOCS (see the IOCS reference manual) as pertains to mode, density, and the EOF of header labels, data blocks, and trailer labels. The program also accepts non-standard labels (see LABEL Card, page 22).

The following conventions apply to all header and trailer labels, the optional checkpoint of the input file, the data blocks of the input file, and the EOF's associated with all of these:

Record	Mode	Density	Remarks
Header Label and EOF	BCD	High or Low	Label and EOF must have same density; density and mode may differ from that of file.
Checkpoint and EOF	Binary or BCD	High or Low	Must have same density and mode as data.
Data Blocks and EOF	Binary or BCD	High or Low	Data and EOF must have same density.
Trailer Label and EOF	BCD	High or Low	Must have same density as data.

The above conventions apply to both input and output files, with the exception that there is no checkpoint file in the output file.

The number of words per tape record must equal the number given in the Input FILE Card (see BLOCKSIZE parameter, page 15). The only exception to this rule occurs when VARIABLE BLOCKING is specified in the OPTION Card (see page 23). In this case, the tape record length may be shorter than specified, provided it is a multiple of the length of the logical record (see LENGTH parameter of the RECORD Card, page 19).

In a binary file only, the actual tape record length may be one word longer than that specified under BLOCKSIZE. This is to accommodate the 18-bit folded checksum (left half of word) and the block sequence number (right half of word) permitted in accordance with the specifications of 709/7090 IOCS. The presence of a checksum and block sequence number is indicated by the CKSUMS and BLKSEQ parameters of the Input FILE Card.

If CKSUMS and BLKSEQ are specified on the Output FILE Card (see page 17), the program computes an 18-bit folded checksum and a block sequence number and substitutes them in place of those values which existed on the input file.

For internal purposes only, if either or both the EQUALS and CKSUMS options are specified on the OPTION Card (as distinguished from the CKSUMS which may be specified on the Input and Output FILE Cards), the program modifies the record in the following manner: A number is assigned to every logical record in the file and is carried immediately following the last word of the logical record. The checksum word follows the sequence number, if it is requested, or the last word of the logical input record if the sequence number is not requested.

Clockword

On all tapes prepared by the Internal Sort Phase and by all passes of the Merge Phase, two words are added to the output blocks -- one at the beginning and one at the end. These are called clockwords.

In the first half of the first clockword and the last half of the last clockword is placed a special bit structure that is used for checking for character shifting.

In the last half of the first clockword and the first half of the last clockword is entered a block sequence number. It is used for checking that the blocks are in proper sequence on the tape and assures that the tape is positioned correctly at all times. These numbers should, in all cases, agree with the physical record count.

SORT SUBPROGRAM BINARY LOADER

The subprograms necessary for each phase of Sort are loaded prior to the initiation of the phase by the Sort Subprogram Binary (SSB) Loader. Thus, storage is not taken up by subprograms used by another phase and the number of core locations used is kept to a minimum for each phase.

The SSB Loader loads the main program of each phase along with whatever subprograms that phase must use. The SSB Loader must therefore know how many programs are to be loaded, the names of these programs, and most importantly, on which input devices these programs are situated. This information is supplied to the Loader by its calling sequence and the Program List.

Calling Sequence

The SSB Loader is always in core storage as part of the Sort Monitor.

Contact is made with the Loader by a three-instruction calling sequence,

```
TSX   LD001,4
PZE   ORIGIN,,N
BCI   1,RETURN
```

where

ORIGIN contains the initial relocation value for each phase, as supplied by Edit Phase. Loading for a particular phase starts at this location.

N indicates how many cells immediately follow ORIGIN in memory.

RETURN is the name of the program to which control is passed when the loading for a particular phase has been completed. This program must be one of those in the Program List.

Program List

The Program List is part of the Sort Monitor. It contains the beginning load address and the names of the programs to be loaded for a particular phase. It also contains the addresses of the input devices on which these programs are situated. Edit Phase supplies the addresses and names of the programs required by each phase, defined as follows:

```
ORIGIN     PZE     xxxxx
           PZE     yyyyy,,M
           BCI     1,NAME1
           BCI     1,NAME2
            .
            .
            .
           BCI     1,NAMEM
           PZE     yyyyy,,M
           BCI     1,NAME1
           BCI     1,NAME2
            .
            .
            .
```

The SSB Loader commences loading at location xxxxx and also uses xxxxx as the initial relocation value, augmented each time by the length of the program just loaded. The address yyyyy signals the Loader as to where the following M programs are situated. This address can specify any of 80 tape units or 8 card readers (Channels A-H).

When the M programs from yyyyy are loaded, the Loader continues to the next set of M programs specified by the next yyyyy.

Library Tape

The Sort Library Tape contains all the programs for the sort operations along with the Sort Monitor.

The Library Tape is composed of blocked card-images of the sort program in column binary form. In this discussion any references made to "card" signify a card-image on the Library Tape. The sub-programs which the Loader must load are placed on the Library Tape in the following manner:

Program Card
-Program-
Transfer Card
Program Card
-Program-
Transfer Card
.
.
.
Program Card
-Program-
Transfer Card

Each program must be preceded by a Program Card and followed by a Transfer Card (one of the three types discussed below).

1. Program Card: This is the first card punched out by 709/7090 FAP for every relocatable program. No program card is punched for an absolute assembly -- a hand-punched program card must precede an absolute program. The Program Card, distinguished by a 12-punch in column 1 (of a column binary card) supplies the SSB Loader with the name of the program (columns 13-15) and its associated entry point (columns 16-18). This card also contains the length of the program, the length of the transfer vector, and the COMMON break, which is zero if no reference is made to COMMON.

2. Transfer Card: The presence of a Transfer Card in the deck signals the Loader that the end of a particular program has been reached. The Loader can recognize three types of Transfer Cards:

   a. FORTRAN Transfer Card - This card contains only a 12-punch in column 1.

   b. Absolute Transfer Card - There is no column 1 prefix and no word count -- only an address in columns 2 and 3.

   c. Relocatable Transfer Card - This card is not punched by the assembly program -- must be hand-punched. It need only consist of 11-, 7-, and 9-punches in column 1.

3. Program: The cards of the program can be either absolute or relocatable binary. Cards to be loaded via the card reader are row binary; all others are blocked column binary and placed on tape.

Load List

The SSB Loader maintains a list of 101 words called the Load List. The first word is the TCP (Transfer Control Point, see below). The next 100 words store information on fifty programs in a two-word scheme as follows:

Word 1: Program name in BCD
Word 2: Decrement - number of entries in transfer vector
Address - SSB-given load address

When the programs have been loaded, the Load List is searched to find the load address of the TCP program and the entries in the transfer vectors of the individual programs are changed to TTR's to the load address or, in case the program has not been loaded, the prefix of the entry in the transfer vector is changed to an STR.

Use of TTR and STR in Transfer Vectors

When loading is complete, pass 2 is begun. Each of the loaded programs contains a transfer list of the entry points to any other subprograms called on by the program: During pass 2, each BCD subprogram name in these transfer lists is replaced by a TTR ***** if that subprogram was also loaded, or an STR prefix if the subprogram was not loaded. The STR prefix (-1) becomes the first character of the BCD name.

Transfer Control Point

The third word in the calling sequence to SSB Loader is of the type

```
BCI    1,RETURN
```

The BCD name RETURN is placed in the TCP word, the first word of the Load List, as one of the Loader's first functions. When pass 2 is initiated, the name in the TCP word is compared with the names of the loaded programs. When its equal is found, its load address is placed in a TTR instruction. When pass 2 is complete, this TTR instruction is executed, and control is passed to the TCP program.

CHECKPOINT AND RESTART PROCEDURE

Checkpoints are written at the following points in the execution of a sort, except in those cases where the NOCKPT option is specified in the OPTION Card (see page 23):

1. After each reel of input of the Internal Sort Phase
2. At the end of the Internal Sort Phase
3. After each pass of the Merge Phase
4. After each reel of output of the Final Merge Phase (EOT)

When a checkpoint is taken, core storage is written on the checkpoint tape (SYSCK2) and preserved, and all tape assignments are recorded along with the number of records on each, to allow the repositioning of tapes on restart.

To restart the sort run, an $EXECUTE SORT Control Card is used, followed by a RESTART Control Card. The Edit Phase of 7090 Sort reads the RESTART Card and transfers control to the Sort Monitor which initiates the restart procedure. SYSCK2, the checkpoint tape, is positioned and the checkpoint file is read in; core storage is restored; and the tapes are repositioned.

ASSIGN AND RELEASE

ASSIGN is an Edit Phase subroutine used for assigning units from the

Availability Chain for use by the sort program. The user may also use this routine to request the assignment of specific units.

The calling sequence to the ASSIGN subroutine is

```
          CALL      ASSIGN
          PZE       LIST,,LENGTH
              Return
```

where

LIST = location of the first entry in LIST, i.e., the first of those consecutive locations in core storage containing the information necessary to make desired assignments (see below).

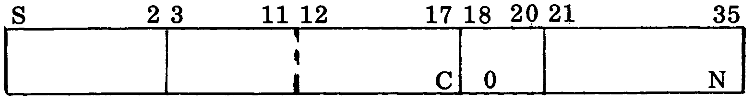
LENGTH = length of LIST in number of words.

Return 2,4 is used by ASSIGN if more units are requested than are available. All available units have been assigned. The number of units still needed is in the accumulator.

Return 3,4 is used by ASSIGN when the request is for an inter-system unit (symbolic channels J-Q) and no such unit is found in the scan of the Unit Control Blocks.

Return 4,4 is the normal return.

Each of the entries in LIST must have the following format:



C is the BCD representation of the channel being requested. Where there is an N in the address portion of the entry, the Nth unit in the Availability Chain will be assigned. Otherwise, the first unit in the Availability Chain will be assigned.

If an inter-system communication unit is sought, C must be one of the symbolic channel designations J-Q. The address portion (bits 21-35) of the LIST entry must contain the same bits as found in the address portion of the first Unit Control Word of that unit. (See the IBSYS bulletin.)

If the unit to be assigned is not in ready status, a message is printed to read if the unit and the machine halts.

When ASSIGN returns, each entry in LIST corresponding to an assigned unit contains the location of the Unit Control Block of that unit in its address portion.

A unit which has been assigned at the request of the user must also be released by the user. The calling sequence to the RELEASE subroutine is

```
CALL        RELEAS
PZE         LIST,,LENGTH
  Normal return
```

where LIST and LENGTH are the same as in the calling sequence to the ASSIGN subroutine above.

Each of the entries in LIST must have the following format:

S	1	18 20	21 35
	*	0	N

N is the location of the Unit Control Block of the unit to be released.

If a unit has already been released and placed in the Availability Chain, the request will be ignored.

*If a unit was reserved for inter-system communication and is now to be release, the 1-bit must be set to one in the LIST entry. Otherwise, the unit is not released.

USER MODIFICATIONS TO SORT

In general, modifications to the sort program are simply subprograms added to the programs on the Library Tape and requested at execution time by control cards (see page 24). In this way, modifications can usually be incorporated in the program without reassembly. Switches have been included in 7090/7094 Sort at those points where modifications are assumed to be most likely. The request for a modification causes the SSB Loader to load the modification along with the appropriate phase and set the necessary switch to use it.

Like the other Sort subprograms, the modifications must be assembled in relocatable form using FAP (the FORTRAN Assembly Program). In order that they be recognized by the SSB Loader, their names are restricted. The name indicates the phase in which a modification is to be loaded and the point at which it is to be used. The names are as follows:

XSM01 . . . XSM10	Internal Sort Phase
MPM01 . . . MPM05	Merge Phase
FMM01 . . . FMM10	Final Merge Phase

Note: The Edit Phase modifications (EPM01 through EPM05) are treated in a slightly different manner (see below).

The modification switches may be of one of the following forms:

```
ZET   INDICATOR
CALL  MODIFICATION
```

or

```
NZT   INDICATOR
TRA   next
STL   GENPR-4
CALL  MODIFICATION
```

Switches of the second type are used in a program as follows:

```
Main Program                                          Subprogram
             .                           MODIFICATION    .
             .                                           .
             .                                           .
             NZT     INDICATOR                           STL   GENPR-5
             TRA     next                                TRA   1,4
             STL     GENPR-4
             CALL    MODIFICATION
next         .
             .
             .
```

GENPR-4 and GENPR-5 are two locations in COMMON. A call to a subprogram is preceded by storing, in location GENPR-4, the location at which control left the main program. A return to the main program is preceded by storing the last location of the subprogram in location GENPR-5. This permits a method of tracing in case of any type of difficulty.

Each of these switches, or transfer points, has a permanent indicator, such as IND5, and a modification name, such as XSM05. Initially, the indicator is set to zero; however, if the modification is requested, the indicator is set to non-zero. Although the modification name always appears in the transfer vector of the phase in which the above instructions are written, the loader processes the transfer vector in different ways depending on whether or not the modification has been requested and loaded from the Library Tape.

Thirty indicators have been provided and more can be included, if desired, by reassembling the sort program.

One MODIFICATION Control Card (see page 24) is needed for each modification requested. When the control card is read, the Edit Phase adds the name of the modification to the Program List and sets the appropriate indicator to non-zero.

It should be noted that modification programs may obtain their parameters from COMMON storage, assuming that COMMON has been defined as in each of the other sort subprograms. If additional COMMON storage is needed, a reassembly of the sort will be necessary. The modification programs may communicate with each other or with any of the subprograms through their transfer vectors.

It is assumed that modifications to the Edit Phase will be rare. Consequently, a different method of requesting them is provided. After IBSYS reads the $EXECUTE SORT Control Card and control is transferred to the Sort Monitor, the machine halts and prints

```
SET ENTRY KEYS (31-35) FOR EDIT PHASE MODIFICATIONS
```

After the desired keys, if any, are set, the user presses START to continue and the Sort Monitor adds the requested Edit Phase modification names to the Program List.

Entry keys 31 through 35 are assigned as follows:

Key	Modification Name
31	EPM01
32	EPM02
33	EPM03
34	EPM04
35	EPM05

USE OF COMMON FOR CODING PARAMETER ENTRIES

Since many of the sort programs and subprograms make reference to one or more of the parameter tables, these tables are placed in COMMON using the FAP (FORTRAN Assembly Program) COMMON statements. Any reference to a parameter entry results in the decrement or address of an instruction getting either a 10- or 11-bit configuration in columns 7-12 (the relocation bits; see "Relocation Scheme," below) depending on whether the decrement or address was above or below the Program Break.

These parameter tables, which are consecutive, are placed in COMMON as follows:

```
77461   OPTPR   COMMON   8
77451   OVEPR   COMMON   2
77447   LABPR   COMMON   5
77442   CHAPR   COMMON   6
77434   FILPR   COMMON   14
77416   MODPR   COMMON   30
77360   RECPR   COMMON   10
77346   DELPR   COMMON   2
77344   SAMPR   COMMON   7
77335   CALPR   COMMON   50
77253   GENPR   COMMON   10
77241   RAWPR   COMMON   20
77215   CNTPR   COMMON   20
77171   ERAPR   COMMON   30
77133   REFPR   COMMON   1
```

Assume program XSM02 makes reference to the parameter tables FILPR, SAMPR, MODPR, and ERAPR. Typical coding and assembly might be

Coding		Assembly	Loader Given Address
CLA	FILPR+3	0500 77437	77755
ADD	SAMPR+4	0400 77340	77666
SUB	MODPR-10	0402 77404	77722
STO	ERAPR+29	0601 77226	77544

In essence, SSB would add 316 to each of the above addresses. The resulting sum is the desired address and replaces the assembled address.

Relocation Scheme

When the SSB Loader encounters a 0 relocation bit in columns 7-12 of a relocatable card, it does not relocate the decrement or address. When a 10- or 11-bit configuration is encountered, however, the following relocation scheme is used:

1. 10 bits - below the COMMON break (which is the last location not used by the program): relocate the address or decrement relative to the value of the current increment.

   - above the COMMON break: relocate the address or decrement relative to the parameter table load address.

2. 11 bits - below the COMMON break: relocate the address or decrement relative to the parameter table load address.

   - above the COMMON break: relocate the address or decrement relative to the value of the current increment.

APPENDIX A. COLLATING SEQUENCES

Listed below are the two collating sequences which may be used in 7090/7094 Sort.

Commercial Sequence		Scientific Sequence	
Character	Card Code	Character	Card Code
blank		0	0
,	12-8-3	1	1
)	12-8-4	2	2
+	12	3	3
$	11-8-3	4	4
*	11-8-4	5	5
-	11	6	6
/	0-1	7	7
,	0-8-3	8	8
%	0-8-4	9	9
#	8-3	#	8-3
@	8-4	@	8-4
+0	12-0	+	12
A	12-1	A	12-1
B	12-2	B	12-2
C	12-3	C	12-3
D	12-4	D	12-4
E	12-5	E	12-5
F	12-6	F	12-6
G	12-7	G	12-7
H	12-8	H	12-8
I	12-9	I	12-9
-0	11-0	+0	12-0
J	11-1	.	12-8-3
K	11-2	)	12-8-4
L	11-3	-	11
M	11-4	J	11-1
N	11-5	K	11-2
O	11-6	L	11-3
P	11-7	M	11-4
Q	11-8	N	11-5
R	11-9	O	11-6
ǂ	0-8-2	P	11-7
S	0-2	Q	11-8
T	0-3	R	11-9
U	0-4	-0	11-0
V	0-5	$	11-8-3
W	0-6	*	11-8-4
X	0-7	blank	
Y	0-8	/	0-1
Z	0-9	S	0-2
0	0	T	0-3
1	1	U	0-4
2	2	V	0-5

Commercial Sequence		Scientific Sequence	
Character	Card Code	Character	Card Code
3	3	W	0-6
4	4	X	0-7
5	5	Y	0-8
6	6	Z	0-9
7	7	‡	0-8-2
8	8	,	0-8-3
9	9	%	0-8-4

APPENDIX B. ALLOCATION OF COMMON STORAGE

The COMMON area of core storage contains the parameters of the sort program as follows:

OPTPR	COMMON	8	Option Card Parameters
OVEPR	COMMON	2	Overflow Card Parameters
LABPR	COMMON	5	Label Card Parameters
CHAPR	COMMON	6	Channel Card Parameters
FILPR	COMMON	14	File Card Parameters
MODPR	COMMON	30	Modification Card Parameters
RECPR	COMMON	10	Record Card Parameters
DELPR	COMMON	2	Delete Card Parameters
SAMPR	COMMON	7	Sort and Merge Card Parameters
CALPR	COMMON	50	Calculated Parameters
GENPR	COMMON	10	General Parameters
RAWPR	COMMON	20	Read and Write Parameters
CNTPR	COMMON	20	Count Parameters
ERAPR	COMMON	30	Erasable Storage Parameters
REFPR	COMMON	1	Referenced Parameters Parameter

Allocation of storage within these areas is as follows:

OPTION Card Parameters

OPTPR-0 Number of cells COMMON relocated
-1 CKSUMS option
0 = no
1 = yes
-2 NOCKPT option
0 = no
1 = yes
-3 EQUALS option
0 = no
1 = yes
-4 BUFFER option
0 = no
1 = yes
-5 VARIABLE BLOCKING option
0 = no
1 = yes
-6
-7

OVERFLOW Card Parameters

OVEPR-0 Number of blocks for overflow
-1

LABEL Card Parameters

LABPR-0 Address of area for label construction; area is 14 words long

-1 Address of area for reading of label; area is 14 words long

-2 Address of LABEL Card if given

-3 Length of LABEL Card

-4 Address of labels used in Merge Phase

CHANNEL Card Parameters

CHAPR-0 Address of input channel

-1 Address of output channel UCW reference table

-2 Address of UCW table for output from Internal Sort Phase

-3 Address of UCW table for other merge channel

-4 Inter-system output mode

-5 Checkpoint tape attached
0 = yes
Non-zero = no

FILE Card Parameters

FILPR-0 Address of sort input file block

-1 Address of output file block

-2
.
. } Addresses of merge input file blocks
.
-11

-12

-13

MODIFICATION Card Parameters

MODPR-0 XSM01
. .
. .
. .
-9 XSM10
-10 MPM01
. .
. .
. .
-14 MPM05
-15 FMM01
. .
. .
. .
-24 FMM10
-25
-26
-27
-28
-29

RECORD Card Parameters

RECPR-0	Fixed record length to Internal Sort Phase
-1	Fixed record length of Merge Phase
-2	Fixed record length from Final Merge Phase
-3	
-4	
-5	Records to be sorted (in this version, always 0) 0 = fixed 1 = variable 2 = type
-6	
-7	
-8	
-9	

DELETE Card Parameter

DELPR-0	Address: address of DELETE control fields Decrement: number of DELETE statements
-1	

SORT and MERGE Card Parameters

SAMPR-0	Sort or merge sequence 0 - Scientific (709) 1 - Commercial (705)
-1	Merge order
-2	Number of files to be merged
-3	Address: address of control field data Decrement: number of control fields
-4	Sort - non-zero
-5	Merge - non-zero
-6	

Calculated Parameters

CALPR-0	Date-year, day
-1	Input blocking factor (number of logical records per block)
-2	Output blocking factor (number of logical records per block)
-3	Binary length (variable-length records)
-4	Size of G in words (XS)
-5	Size of tag table, L (XS)
-6	Size of partial sort list, M (XS)
-7	Ratio (L/M)
-8	Effective record length (XS)
-9	Effective record length (MR)
-10	Effective record length (FM)

-11	
-12	Number of words in all control fields
-13	Maximum merge blocking in words
-14	Maximum merge blocking in number of logical records
-15	Number of merge areas for Merge Phase
-16	Number of merge areas for Final Merge Phase
-17	Grouping factor for XS (number 1)
-18	Grouping factor for XS (number 2)
-19	Address: address of low padding record Decrement: record length of padding record
-20	Address: address of high padding record Decrement: record length of padding record
-21	
-22	Length of longest control field
-23	Available memory XS
-24	Available memory MR
-25	Available memory FM
-26	Number of words in input block (including trailer words, if expected)
-27	
.	
.	
.	
-49	

General Parameters

GENPR-0	Phase indicator
-1	Binary/BCD indicator 0 = BCD Non-zero = binary
-2	Pass number (Merge Phase)
-3	Complement of Index Register 4 for use of RESTART to return control to the main program
-4	
.	
.	
.	
-9	

Read and Write Parameters

RAWPR-0	Calling sequence Word 1 for read and write
-1	Calling sequence Word 2 for read and write
-2	Read completion indicator
-3	Read EOF indicator
-4	Write completion indicator
-5	Write EOT indicator

-6	Current buffer read grouping factor
-7	Address: logical file number of current read Decrement: logical file number of current write
-8	Sequence break and block length (second word of merge block)
-9	Write head word (sequence and block word)
-10	Location of file data block for current read
-11	Buffer write switch
-12	External output switch 0 = current output is internal output Non-zero = current output is external output
-13	External input switch 0 = current input is internal input Non-zero = current input is external input
-14	Location of table for read calling sequence word 2
-15	Location of table for write calling sequence word 2
-16	Current read mode 0 = BCD Non-zero = binary
-17	Current write mode 0 = BCD Non-zero = binary
-18	Location of first IOSP for current block in Scatter Read
-19	Address: number of logical records in current short block Decrement: number of words in last logical record of current short block

Count Parameters

CNTPR-0	Count of records (XS)
-1	Count of records (MR)
-2	Count of records (FM)
-3	Count of records dumped (XS)
-4	Count of records dumped (MR)
-5	Count of records dumped (FM)
-6	Count of padding records
-7	Records dumped this pass
-8	Records deleted in Internal Sort Phase
-9	Records deleted, this phase
-10	Record count, low padding, Final Merge Phase
-11	Record count, high padding, Final Merge Phase
-12	Write table fixer count
-13	Count of records to date for deblocking
-14	Count in records of total input to sort or merge
-15	
.	
.	
.	
-19	

Erasable COMMON Parameters for Internal Sort Phase

ERAPR -0	Address of word 1 of control field information
-1	Address of word 2 of control field information
-2	Address of word 3 of control field information
-3	Address of word 4 of control field information
-4	Address of word 5 of control field information
-5	Address of last record written out
-6	
-7	
-8	
-9	
-10	Address: location of first command in read table Decrement: M
-11	Address: location of first command in write table Decrement: number of record to be written out
-12	
.	
.	
.	
-19	
-20	File number (logical)
-21	
.	
.	
.	
-29	

Erasable COMMON Parameters for Merge Phase

ERAPR-0	Address of word 1 of control field information
-1	Address of word 2 of control field information
-2	Address of word 3 of control field information
-3	Address of word 4 of control field information
-4	Address of word 5 of control field information
-5	Communication cell between MR and FM EQUAL routines
-6	
-7	
-8	Compare Indicator 0 = logical ascending 1 = logical descending 2 = algebraic ascending 3 = algebraic descending
-9	Communication between MR and FM EXTRACT routines
-10	
-11	Write base for DELETE
-12	
.	
.	
.	
-19	

-20 Address to insert file number in internal label

-21 Address to insert pass number in internal label

-22 Communication cell between MR and FM and EQUALS

-23 Communication cell between MRL and LEQ

-24

.

.

.

-29

Referenced Parameters Parameter

REFPR-0 Address of last cell used by referenced parameters

APPENDIX C. FORMAT FOR CONTROL FIELD INFORMATION

Each of the control fields by which records are to be sorted or merged has a corresponding 5-word block of information in storage. This information is developed during the Edit Phase from the parameters of the SORT (or MERGE, in a merge run) Control Card.

The Internal Sort, the Merge, and the Final Merge Phases place these first five addresses in ERAPR to ERAPR-4 consecutively. Reference to the data of any control field can then be made via indirect addressing to ERAPR, ERAPR-1, etc. The format of these five words is as follows:

Word	Bits	Contents
1	S	0=ascending 1=descending
	3-17	Number of words control field extends over
	21-35	Starting word of control field
2	21-35	0=logical compare 1=algebraic compare 2=signed BCD compare
3		Left mask
4		Right mask
5	3-17	Ending bit of last word in control field
	21-35	Starting bit of first word in control field

Each DELETE control field also has a corresponding storage block of a minimum of 4 words, as follows:

Word	Bits	Contents
1	3-17	Number of words control field extends over
	21-35	Starting word of control field
2		Left mask
3		Right mask
4-n		Information to compare against DELETE control field.

APPENDIX D. INPUT AND OUTPUT FILE DATA BLOCK

Information on the input and output files of a sort or merge run is taken from the parameters of the FILE Control Cards of that run. This information is stored in a 25-word block, arranged as follows:

		Used	
Word	Contents	Input	Output
1	Mode 0 = decimal (may be omitted) Non-zero = binary	X	X
2	Density 0 = high (may be omitted) Non-zero = low	X	X
3	Padding 0 = no padding 1 = low padding 2 = high padding	X	X
4	Label (must be given) 0 = standard 1 = non-standard 2 = no label	X	X
5	Label density 0 = high (may be omitted) Non-zero = low	X	X
6	Blocksize (must be given)	X	X
7	Serial number (checked if given)	X	X
8	Reel sequence number (checked if given)	X	X
9	Checksums 0 = no Non-zero = yes	X	X
10	Block sequence number	X	X
11	Dictionary 0 = no Non-zero = yes	X	X
12	Name (checked if given)	X	X
13	Name	X	X
14	Name	X	X

		Used	
Word	Contents	Input	Output
15	Input number (must be given)	X	
16	Number of input reels (1 if not given)	X	
17	Number of input records (may be omitted)	X	
18	Checkpoint 0 = standard (may be omitted) 1 = non-standard 2 = no checkpoint	X	
19	Retain		X
20	Checksums or block sequence 0 = neither Non-zero = yes	X	X
21	Grouping factor	X	X
22	Number of reels processed to date	X	
23	Blocksize plus checksum or sequence word	X	X
24			
25			

APPENDIX E. THE SORT SUBPROGRAMS

Following is a list of the 7090/7094 Sort subprograms, giving their symbolic names, their functions, and the phases for which they will be loaded.

Symbolic Name	Function	Phase(s) Loaded
ASSIGN	ASSIGN	EP
BTD	BINARY TO DECIMAL	MR, FM
IOBS	BUFFER SCATTER	XS*, MR, FM
CKPT	CHECKPOINT	XS, MR, FM
CKSUM	CHECKSUM	XS*, MR*, FM*
BCONV	DECIMAL TO BINARY	XS**
DEBLK	DEBLOCK AND DUMP	XS, MR, FM
DELETE	DELETE	XS*, FM*
DEPAD	DEPADDING	XS
EQUALS	EQUALS	XS*
EP001	EDIT PHASE	EP
LEQ	EQUALS	MR, FM
XTR	EXTRACT	MR, FM
FM	FINAL MERGE	FM
FXMOV	FIXED MOVE	XS
LABEL	LABEL	XS, MR, FM
LOCATE	LOCATE	XS**
MRL	LITTLE SORT ROUTINE	MR, FM
MR	MERGE	MR
MOVE	MOVE	XS**
POST	POST-PROCESSOR	Post-Processor
SOP	OPEN-CLOSE	EP, XS, MR, FM
RB01	CARD-IMAGE TO BCD	EP
RELEAS	RELEASE	EP, Post-Processor
RESTAR	RESTART	EP, XS, MR, FM
SK001	SCAN	EP
IOSS	SCATTER-SCATTER	XS*
WTFIX	WRITE TABLE FIX	XS
WRSEL	WRITE SELECT	XS, MR, FM
	INTERNAL SORT	XS

* optional

** variable-length records only

APPENDIX F. LOADING THE SYSTEM

1. SYSLB1 - Systems Tape (IBSYS - all systems)

2. SYSIN1 - Input Unit

SSW1 Down	RDA Attached as SYSIN1	Control Card Loading Unit
Yes	Yes	IBSYS control cards and Sort control cards in card reader.
Yes	No	IBSYS control cards in card reader; Sort control cards on tape.
No	Yes	IBSYS control cards on tape; Sort control cards in card reader.
No	No	IBSYS control cards and Sort control cards on tape.

3. SSW6 Down - all on-line printing is suppressed except for tape assignments and error messages.

4. Press, Load Tape key to start.

Note: The Sort program must always be on tape.

IBM PUBLICATIONS

Following is a list of IBM publications which may be of interest to the reader:

REFERENCE MANUALS

Form Number	Title
A22-6528-1	IBM 7090 Data Processing System
A22-6616	7340 HYPERTAPE Drive
C28-6036	Generalized Sorting Program for the IBM 709 Sort 709
C28-6052	Generalized Merging Program for the IBM 709 Merge 709
C28-6100-2	IBM 709/7090 Input/Output Control System

GENERAL INFORMATION MANUALS

D22-6508-2	IBM 709/7090 Data Processing System
F28-8001	Sorting Methods for IBM Data Processing Systems

BULLETINS

G22-6505-1	IBM 7090 Data Processing System
G22-6634	7340 HYPERTAPE Drive
J28-6043-1	Sort 709: Sorting Times for the IBM 7090
J28-6059	Addenda and Errata to the Sort 709 Manual
J28-6061	Addenda to the Merge 709 Manual
J28-6098-1	FORTRAN Assembly Program (FAP) for the IBM 7090
J28-6138	Sort 709: Sorting Times for the IBM 7090 with IBM 729 VI Magnetic Tape Units
J28-6152	IBM 7090 with IBM 7340 HYPERTAPE Drives: Programs and Programming Systems
J28-6153	IBM HYPERTAPE Input/Output Control System for 7000 Series Data Processing Systems
J28-6156	IBM 7090 Generalized Sorting Program Using IBM 7340 HYPERTAPE Drives
J28-6184	IBM 7094 Programs and Programming Systems
J28-6186	FORTRAN Assembly Program (FAP) for the IBM 709/7090: Supplementary Information for the 32K Version
J28-6194	IBM 7090/7094 IBJOB Processor - Part 1: Monitor (IBJOB)
J28-6195	IBM 7090/7094 IBJOB Processor - Part 2: Loader (IBLDR)
J28-6196	IBM 7090/7094 IBJOB Processor - Part 3: Macro Assembly Program (IBMAP)
J28-6197	IBM 7090/7094 IBJOB Processor - Part 4: FORTRAN Compiler (IBFTC)
J28-8086	7090 Operating Systems: Basic Monitor (IBSYS)

IBM
International Business Machines Corporation
Data Processing Division, 112 East Post Road, White Plains, N. Y.

IBM Technical Newsletter

System 7090/7094
Re: Form No. J28-6217-0
This Newsletter No. N28-0023
Date August 3, 1962
Previous Newsletter Nos. None

ADDENDA TO FORM J28-6217-0: IBM 7090/7094 GENERALIZED SORTING PROGRAM - 7090/7094 SORT (729 - FIXED LENGTH) LOADING FROM IBM 1301 DISK STORAGE

The following information is supplied to describe the loading and editing of 7090/7094 Sort from tape and disk. This newsletter should be used in conjunction with the IBM bulletin, IBM 7090/7094 Generalized Sorting Program: 7090/7094 Sort (729 - Fixed Length), form J28-6217-0. Knowledge of the contents of this bulletin is assumed. The reader is also assumed to be familiar with IBSFAP operating under 7090/7094 IBSYS (729/1301 Version) and the contents of the IBM reference manual, IBM 7090/7094 Operating Systems: Basic Monitor (IBSYS), form C28-6248-0.

Loading the Sort Program

Upon recognition by IBSYS of a $EXECUTE SORT control card, the Sort Monitor is loaded and control is transferred to it. Upon receiving control, the Monitor tests to see if the program is to be loaded from tape or from disk storage. If the program is to be loaded from tape, the tape is positioned at the beginning of the second file (ASSIGN) of the sort program. The loader section of the Sort Monitor now uses its own Select Plus and Select Minus routines to read the several subprograms from the Library Tape. If the program is to be loaded from disk, the loader portion of the Sort Monitor uses the IBSYS system loader (SYSLDR) to position the disk at the beginning of the second file (ASSIGN) of the sort program. SYSLDR is then used to load each of the subprograms into core storage. For both disk and tape, all subprograms are loaded at 40000_8 and then are moved to their final position in storage.

Editing the Sort Program

The following procedure must be followed to update 7090/7094 Sort on either tape or disk:

1. Assemble those programs to be changed,using IBSFAP.

2. Execute 7090/7094 Sort with only Sense Switch 2 DOWN.

International Business Machines Corp., Programming Systems Publications, P. O. Box 390, Poughkeepsie, N. Y.

When the editing section of the Sort Monitor receives control, the same testing, positioning, and reading procedure is followed as explained in "Loading the Sort Program," above. The editor, at this point, reads one of the subprograms from tape or disk. The program just read is compared with the subprograms to be updated. If the two programs have the same name or identification, the newly-assembled program from SYSPP1 is blocked and written on SYSUT2. The old program from the Library Tape or Disk is then deleted. If the programs do not have the same name or identification, then the subprogram just read from the Library Tape or Disk is written on SYSUT2. This procedure is continued until all programs from SYSPP1 and the tape or disk are processed. SYSUT2 then contains all the subprograms of the second file (ASSIGN) of Sort, the old subprograms being replaced by the newly-assembled ones. Control is then returned to IBSYS. IBSYS control cards must then be provided for replacing the second file (ASSIGN) of the sort program.

This editing procedure is necessary, since all the subprograms of the ASSIGN file are relocatable and the IBSYS editor (IBEDT) does not provide for the editing of relocatable programs.

Since the first file of the sort program is absolute, it may be edited in the normal way as described in the IBSYS reference manual.

Restriction: SYSPP1 and SYSUT2, which are used for editing, must be assigned as tapes and cannot be assigned as areas in the disk file.

IBM Technical Newsletter

System 7090/7094
Re: Form No. J28-6217
This Newsletter No. N28-0036
Date December 6, 1962
Previous Newsletter Nos. N28-0023

IBM 7090/7094 GENERALIZED SORTING PROGRAM ADDENDA TO FORM J28-6217

The information in this newsletter provides a description of the variable length record format which will be handled by the 7090/7094 Generalized Sorting Program.

Variable Length Block Format

A block (tape record) may consist of one or more variable length logical records as long as the total number of words in the block is not more than 2000. The checksum-block sequence word which may be appended at the end of a binary tape block is not included in this total.

Variable Length Logical Record Format

Binary Mode Tapes

Binary logical records must be preceded by a control word written in the form IOCTN**,,N containing the length in words in the decrement. The length does not include the control word itself. The Sort will accept input files not in the IOCTN**,,N control word format providing the length is in the decrement portion of the control word. Binary output files will always be written by the Sort in the IOCTN**,,N control word format.

The minimum length of a variable length record is two words plus its associated control word. An exception is the ability to process one word logical records if the checksum-block sequence word is included as part of the tape block on input and output. This insures a three word minimum tape block.

BCD Mode Tapes

BCD logical records must contain as the first word of the record a control word containing the number of characters in the record. This number includes the six characters of the control word and is expressed as five BCD characters left-justified.

Example:

```
0 0 0 3 6 X
```

The sixth character in the control word may be any valid BCD character.

The record length must be a multiple of six characters and must be at least 18 characters.

9PAC Type Records

Logical records which are fixed within types but not between types will not be sorted by 7090/7094. These records are presently produced by the 7090 9PAC System. 9PAC files, however, can be prepared so that they will be sorted by 7090/7094. This preparation is described in the bulletin, IBM 7090 Programming Systems: Share 7090 9PAC-Supplement, Form J28-6211-1.