UNIVAC **DATA FOR COPYRIGHT REGISTRATION**

TITLE

UNIVAC 9200/9300 Series 8410 Disc Subsystem Sort Programmer Reference, UP-7651 Rev. 1

NAME AND HOME ADDRESS OF AUTHOR

Edward R. Kelly 732 W. Wingohocking St. Phila., Pa.

CITIZENSHIP OF AUTHOR

USA

DATE OF PUBLICATION (RELEASE)

January 1973

NEW WORK

No

REVISION

Yes

TITLE, AUTHOR AND DATE OF ORIGINAL WORK AND/OR PRIOR REVISIONS

UNIVAC 9200/9300 Systems Disc Sort Programmers Reference, UP-7651

CHANGES FROM ORIGINAL AND/OR PRIOR REVISIONS

This document now provides the programmer with linking procedures for the sort/merge program; also, changes have been made to the description in Section 1 for the organization of the record sort, tag sort, and merge card decks. Other minor corrections and additions have been made throughout the manual.

NAME AND ADDRESS OF PHOTOENGRAVER

N/A

NAME AND ADDRESS OF TYPESETTER

UNIVAC P. O. Box 500 Blue Bell, Pa. 19422

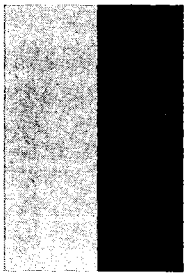
NAME AND ADDRESS OF PRINTER

UNIVAC P. O. Box 500 Blue Bell, Pa. 19422

NAME AND ADDRESS OF BINDER

N/A

UNIVAC
SYSTEMS
PUBLICATIONS



UNIVAC 9200/9300 Series

REVISION

8410 Disc Subsystem
Sort Programmer Reference
UP-7651 Rev. 1

E. K.

This UNIVAC 9200/9300 Series Library Memo announces the release and availability of "UNIVAC 9200/9300 Series 8410 Disc Subsystem Sort Programmer Reference", UP-7651 Rev. 1. This is a Standard Library Item (SLI).

This revision contains additions and corrections to the "UNIVAC 9200/9300 Systems Disc Sort Programmers Reference", UP-7651. This document now provides the programmer with linking procedures for the sort/merge program; also, changes have been made to the description in Section 1 for the organization of the record sort, tag sort, and merge card decks. Other minor corrections and additions have been made throughout the manual.

Destruction Notice: This revision supersedes and replaces "UNIVAC 9200/9300 Systems Disc Sort Programmers Reference", UP-7651 released on Library Memo 26 dated March 31, 1969. Please destroy all copies of UP-7651 and/or Library Memo 26.

Request additional copies through your local Univac Manager via a Sales Help Requisition Form. These forms are submitted to:

Customer Information Distribution Center (CIDC)
Univac Division Sperry Rand Corporation
P.O. Box 448
King of Prussia, Pa. 19406

H. MASTERS
Group Manager
Systems Publications

LISTS	ATTACHMENTS	THIS SHEET IS
217, 630 and 692 Library Memo only	UP-7651 Rev. 1 (covers and 54 pages) plus Library Memo to Univac Library Mailing Lists 51D, 52, 53D, 54D, 55D, and 56	Library Memo for UP-7651 Rev. 1

DATE:
January 1973

SPERRY RAND

SPERRY RAND

UNIVAC

9200/9300
SERIES

8410 DISC SUBSYSTEM SORT

PROGRAMMER REFERENCE

UP-7651
Rev. 1

This document contains the latest information available at the time of publication. However, the Univac Division reserves the right to modify or revise its contents. To ensure that you have the most recent information, contact your local Univac Representative.

UNIVAC is a registered trademark of the Sperry Rand Corporation.

PRINTED IN U.S.A.

PAGE STATUS SUMMARY

ISSUE: UP-7651 Rev. 1

The following table lists the status of each page in this document and indicates the update package (if applicable).

Section	Page Number	Page Status	Update Package
Cover/ Disclaimer		Orig.	
PSS	1	Orig.	
Contents	1 thru 3	Orig.	
1	1 thru 7	Orig.	
2	1 thru 7	Orig.	
3	1 thru 7	Orig.	
4	1 thru 7	Orig.	
5	1 thru 17	Orig.	

Section	Page Number	Page Status	Update Package

CONTENTS

FIGURES

TABLES

1. INTRODUCTION

1.1. GENERAL

This manual describes the sort/merge routine provided for use with the UNIVAC 8410 Disc Sort Package and the UNIVAC 9200/9200 II/9300/9300 II Systems. The manual was written on the premise that the reader has a thorough understanding of the following:

- *UNIVAC 9200/9200 II/9300/9300 II Systems Tape/Disc Assembler Programmer Reference, UP-7508* (current version);
- *UNIVAC 9200 II/9300/9300 II Systems Magnetic Tape Subsystems IOCS Programmer Reference, UP-4135* (current version);
- *UNIVAC 9200/9200 II/9300/9300 II System Operating System Programmer Reference, UP-7531* (current version);
- *UNIVAC 9200/9200 II/9300/9300 II Systems Minimum Operating System Programmer Reference, UP-7547* (current version).

The minimum equipment configuration for using the Sort package is:

- One UNIVAC 9200/9300 Central Processor
- 12,288 bytes of storage
- One 400/600 CPM card reader
- One 8410 Disc File Control
- Two 8410 Disc File Handlers

The Sort can use up to 32,768 bytes of storage with a maximum of eight disc file handlers.

On card systems, the disc sort operates in conjunction with the Minimum Operating System and the disc dispatcher. On tape or disc systems, the Sort can be loaded from a library file.

1.2. SORT PACKAGE PROGRAMS

The Sort package contains a record sort program, a tag sort program, and a merge program. (The merge is a separate program distinct from merge coding, which is a part of a sort program.)

1.2.1. Record Sort

A record sort results in a file of records arranged in a physical sequence according to the collating order of a key field in the records.

A record sort operation requires, three discs: input, output, and work. If only two disc drives are available, one drive is used first as an input disc and subsequently as an output disc. The operator is informed by a display when to replace the input disc. The Sort operation continues using an output and a work disc.

1.2.2. Tag Sort

A tag sort results in a file consisting of disc addresses only, disc addresses and their associated key fields, or an entire record.

The tag sort requires a minimum of two disc handlers to produce a file of disc addresses or when user own code is included; three disc handlers are required if the output file is to contain complete records.

1.2.3. Merge Program

The Merge program combines two or more sorted files into one sorted file. Files that are segmented because of time or hardware limitations can be merged into one file at a convenient time.

Each merge input file is assumed to have identical characteristics. A record that is out of sequence in an input file results in a program halt and a display indicating the logical unit number of the input file containing the record that is out of sequence.

1.3. PROCESSING CAPABILITIES

The Sort and the Merge programs process sequential or direct-access files. The operator is alerted by a display before a direct-access file is processed. Indexed sequential files can be processed by using own code to access records from the indexed sequential file and delivering the record to the Sort program.

Fixed-length blocked or unblocked records, and variable-length unblocked records are sorted. The maximum record length is 160 bytes. The first four bytes of a variable-length record must contain record length information.

The field used for the sorting procedure can be made up of a maximum of 12 key fields in a record. Key fields can be alphanumeric, unpacked decimal, packed decimal, or signed binary data; key fields can be sorted in either ascending or descending sequence. The size limitations of a key field are specified in the FIELD statement (see 2.7).

1.4. INFORMATION SUPPLIED TO SORT

The programmer provides statement cards for the Sort and the Merge programs. Statement cards contain information required to direct the Sort or the Merge operation. The information describes input and output file requirements, label specifications, available work (scratch) discs, and the key fields.

Optionally, the programmer provides own code routines to perform input and output operations and functions otherwise controlled by the Sort or the Merge program. Own code routines perform the following functions in addition to the normal Sort operations: file reduction (combining information from two records containing the same key fields), read error control, and sequencing of records.

The method of transferring program control between own code and Sort program is described in the following sections.

1.5. SORT AND MERGE PROGRAM ORGANIZATION

The organization of the record sort, tag sort, and merge sort card decks is as follows:

Record or Tag Sort

CARD
DECK

A = Phase 0 (PH0L or PH0)
B = Statement cards with a blank card following
C = Relocator phase (PHRL or PHL) – optional
D = Own code modules with /* END card and a blank card following (optional)
E = Phase 1
F = Phase 2
G = Phase 3

Merge

CARD
DECK

A = Phase 0 (PH0L or PH0)
B = Statement cards with a blank card following
C = Relocator phase (PHRL or PHL) – optional
D = Own code modules with /* END card and a blank card following (optional)
H = Phase 4 (merge only)

Card decks A, C, E, F, G, and H constitute the self-loading sort setup and operating coding. However, this does not preclude the necessity of loading the Minimum Operating System (MOS) supervisor with the Disc Dispatcher before the Sort is loaded.

Figures 1–1 and 1–2 show where the statement and own code decks are inserted in the Sort and the Merge programs.

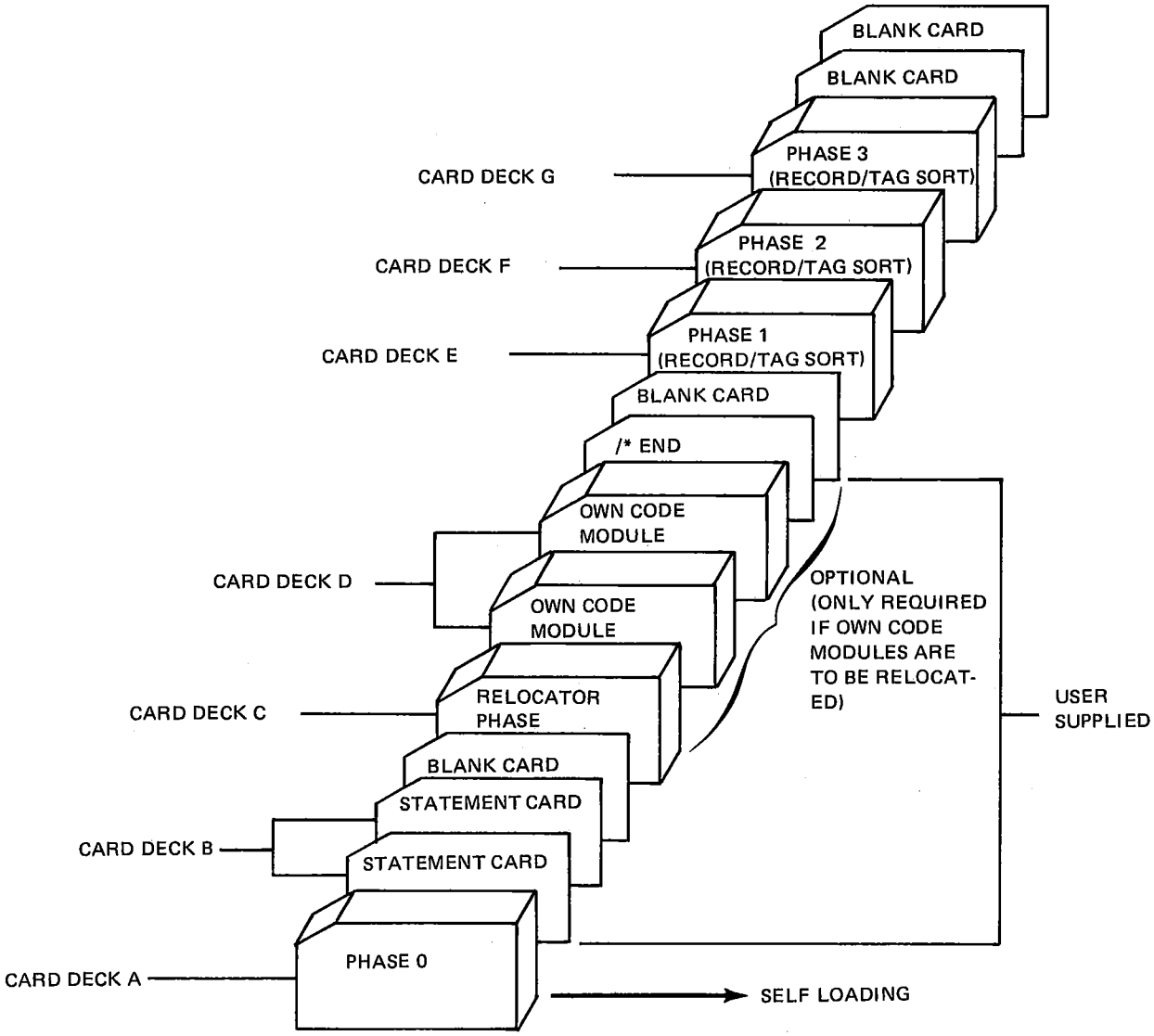


Figure 1–1. Record or Tag Sort, Statement and Own Code Decks

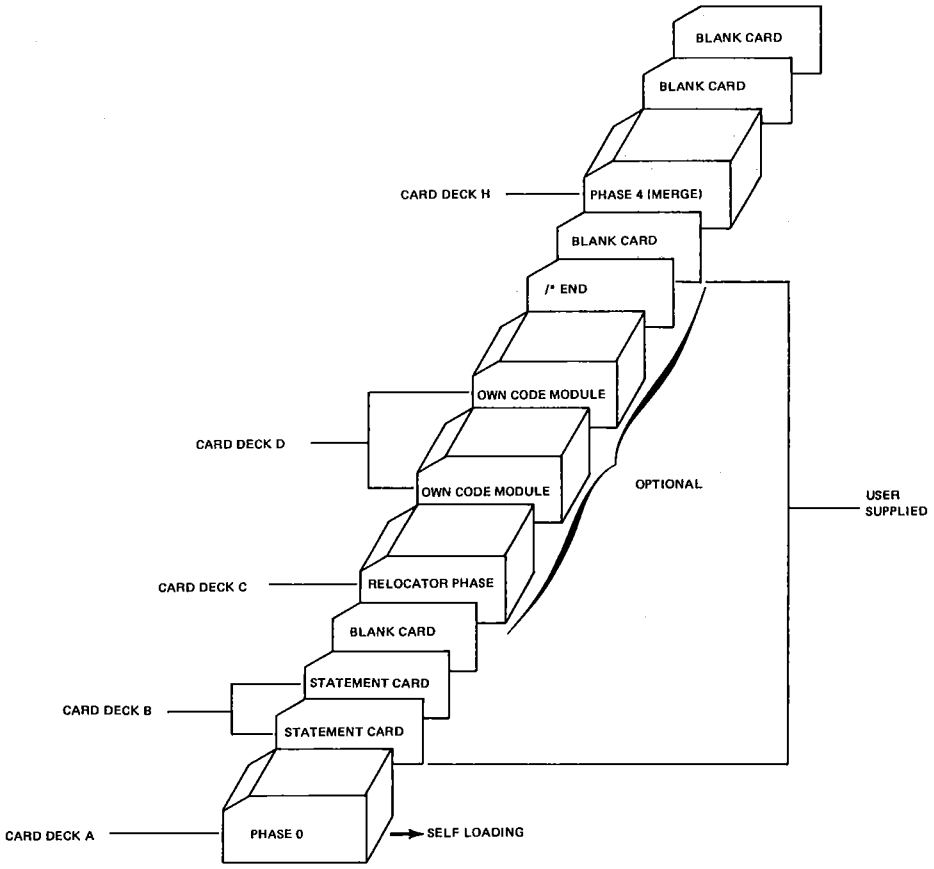


Figure 1–2. Merge, Statement and Own Code Decks

1.6. RESTART FEATURE

The record and tag sort programs can be interrupted anytime after the conclusion of the Input/Internal Sort Phase and subsequently restarted. The Sort program determines the restart point. A record on the first work disc is created by the Sort program and then updated each time a checkpoint is passed. Checkpoints occur at the end of any phase and after a pass within a phase.

Restart is effected by mounting the discs where they were when the Sort was interrupted and assigning them the same logical unit numbers. The original statement deck is resubmitted to the Sort with an additional statement card, RSTRT.

1.7. MULTICYCLE SORT FEATURE

During an input phase, the Sort determines if one-half of the available work area is used before the end of file is detected. If this is the case, the Sort automatically terminates the input phase but continues the Sort operation, using only those records that were accepted prior to the termination of the input phase. (The Sort records information required for the next cycle on the disc checkpoint sectors.) The Sort can then be completed in multicycle operation by resubmitting the deck with a MCYCL statement in the statement deck.

At least three disc units are required for the merge operation following a multicycle sort if the Merge controls the output. If own code controls the output, two disc units are required.

1.8. LINKING PROCEDURES

When the sort/merge program is to be executed under the control of the Nonconcurrent Operating System (NCOS) or higher level operating system, each phase of the program must be linked from the library file using the tape linker. The cards needed to accomplish this are illustrated by Figure 1–3.

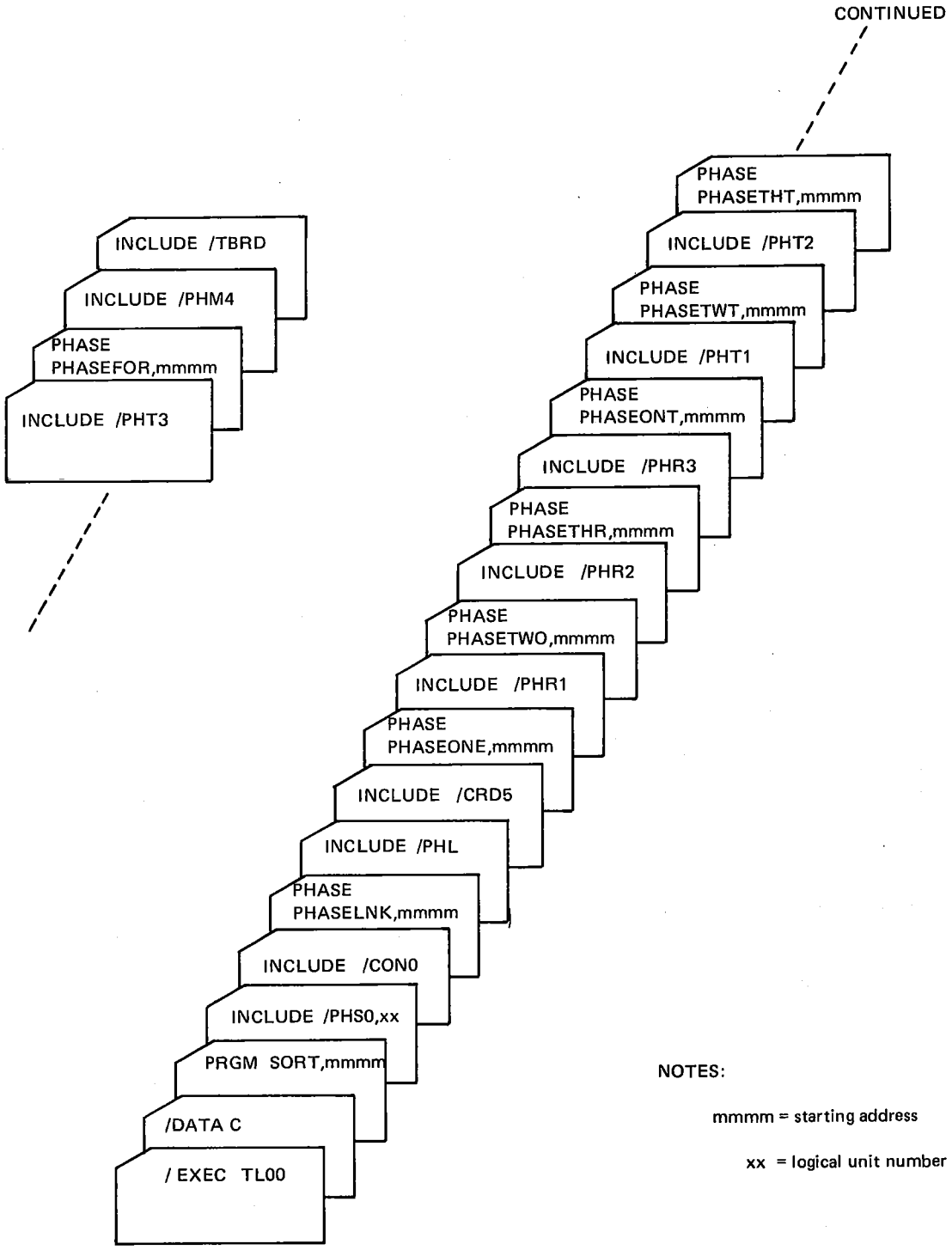


Figure 1–3. NCOS Disc Linker Input

To execute the sort or merge operation after the linking, place the disc pack containing the newly constructed file on logical unit zero. Place any other discs required by the program on their respective units and load the supervisor. Execution can then be accomplished by using the job stream illustrated in Figure 1–4.

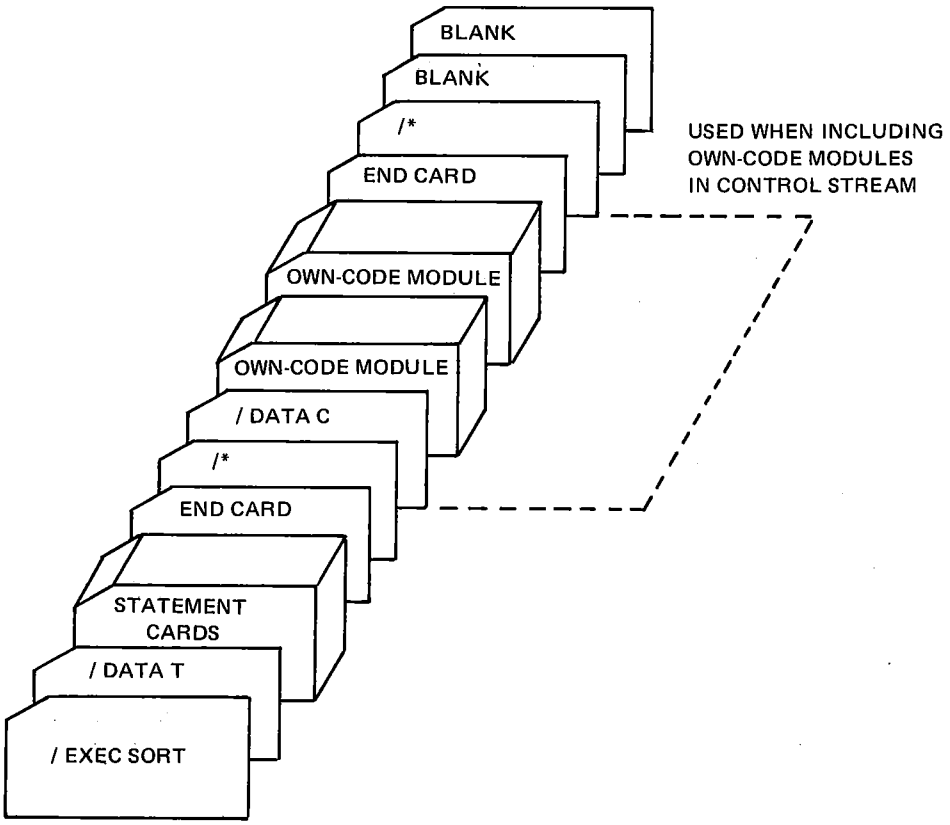


Figure 1–4. Loading Program From Disc Under NCOS

2. STATEMENT CARDS

2.1. GENERAL

The statement names and their associated functions are:

STATEMENT NAME	FUNCTION
DISCS	Describes the available work discs.
IN	Describes an input file.
ILB	Specifies input file label information.
OUT	Describes an output file.
OLB	Specifies output file label information.
FIELD	Describes key fields.
RSTRT	Indicates restart of a sort.
MCYCL	Indicates the Sort continues using multicycle operation.
SORT	Describes the characteristics of the Sort operation.
OWNCD	Specifies the own code routines included in the Sort operation.
END	Indicates the end of the statement deck.

The statement name is specified in the operation field. The label field is blank. A statement can contain parameters in the operand field. The first character of the first parameter must start in column 16. A comma must be specified for an omitted parameter; trailing commas can be omitted.

The following rules apply in creating a statement deck:

- Statement cards can be in any order except for END, which must be the last statement in the statement deck.
- Parameters associated with a given statement must be specified in the sequence shown in the format description.
- Commas must separate parameters; a parameter can be omitted; however, the comma must be specified.
- A blank following a parameter indicates the end of the parameter list in that card.
- Each parameter must conform to the length requirements specified in a given format description.

2.2. DISCS

Use: Indicates the number of work discs available for a sort operation and the logical disc unit addresses.

Format: DISCS $\underline{w}_1$, $\underline{w}_2$, $\underline{w}_3$, $\underline{w}_4$, $\underline{w}_5$, $\underline{w}_6$, $\underline{w}_7$, $\underline{w}_8$.

Legend: Each $\underline{w}$ is a unique two-digit logical unit number of a work disc. A logical number can be in the range 00 through 07.

Rules: The position of a logical number in a statement indicates the sequence in which the work discs are accepted for use by the Sort program.

The value of the logical number assigned to a disc drive must not exceed the number of available disc drives.

A Sort operation requires at least one work disc for each input disc; however, a faster Sort is executed if at least two work discs are available.

A merge operation does not require a work disc unless an own code option is used. The work disc in this case need only be online during the initialization of the merge; it can then be replaced with an input disc if desired.

Notes: The logical unit number of a work disc can be the same as that specified for an input disc (see 2.3) if an input disc is to be used as an additional work disc at the conclusion of the Input/Internal Sort Phase.

For a record sort operation, the programmer can specify the same logical number to both work discs and input discs. The Sort will halt at the end of the input phase allowing the operator to remove an input disc and replace it with a work disc.

For a tag sort operation, the input, output, and work disc units must be online for the entire Sort operation when specifying an entire record as output; discs for output can be omitted if the output device is provided for by the own code routine which can specify any I/O device.

The maximum number of records that can be sorted can be determined from the following formulas:

OPERATION	MAXIMUM NUMBER OF RECORDS	
	NO OWN CODE PRESENT	OWN CODE PRESENT
Record Sort (Fixed-Length Record)	$\left\lfloor \frac{9994(D)}{2} \right\rfloor \times \left\lfloor \frac{160}{RL} \right\rfloor$	$\left\lfloor \frac{9900(D)}{2} \right\rfloor \times \left\lfloor \frac{160}{RL} \right\rfloor$
Record Sort (Variable-Length Record)	$\left\lfloor \frac{9994(D)}{2} \right\rfloor \times \left\lfloor \frac{160}{RL_{MAX}} \right\rfloor$	$\left\lfloor \frac{9900(D)}{2} \right\rfloor \times \left\lfloor \frac{160}{RL_{MAX}} \right\rfloor$
Tag Sort	$\left\lfloor \frac{9994(D)}{2} \right\rfloor \times \left\lfloor \frac{160}{KL+7} \right\rfloor$	$\left\lfloor \frac{9900(D)}{2} \right\rfloor \times \left\lfloor \frac{160}{KL+7} \right\rfloor$

where: D = number of work discs available to Sort program.
RL = record length.
RL_{MAX} = maximum record length (variable-length records).
KL = total length of all key fields
[ ] = round integer to next higher value if quotient contains a decimal fraction.
⌊ ⌋ = truncate decimal fraction of quotient.

Each work disc must have been prepared by the VTOC Recorder. A single file, with file identification of SORTWORK and one extent from TT=00, SS=55 to TT=99, SS=99 should be created by the user; the Sort rejects any work disc volume not meeting these requirements.

An example of DISC statements selecting work discs on logical units 01 and 03 is:

LABEL	OPERATION	OPERAND
1	10	16
	DISCS	01,03

2.3. IN

Use: Describes the characteristics of an input file.

Format: IN b, i_1, i_2, e, f, s, g, v, n

Legend: b is a two-digit value indicating the number of records per block. The value must be in the range 01 through 99.

i_1 is the first logical disc unit number of the input file; the number must consist of two digits in the range 00 through 07.

i_2 is the second logical disc unit number of the input file; the number must consist of two digits in the range 00 through 07.

e can be any one of the following:
C — cancel sort upon detecting a read error.
D — ignore read error.
S — skip sector containing read error.

f is the record length in bytes for fixed-length records. The record length must consist of three digits in the range 001 through 160. This parameter must be present when processing fixed-length records.

s is a three-digit value specifying the minimum number of bytes contained in a variable-length record. The value must be in the range 005 through 160.

g is a three-digit value specifying the maximum number of bytes contained in a variable-length record. The value must be in the range 005 through 160.

v is a two-digit value specifying the number of input volumes. The value must be in the range 01 through 99.

n is an eight-byte input file name. It consists of eight alphanumeric characters.

Rules: Block size must be in the range 2 through 160 bytes; block size is equal to the number of records per block (b) multiplied by the record length (f).

For variable-length records, the parameters s and g must always be specified even if the own code routine controls the input.

The parameter v does not have to be specified if input is controlled by own code.

If i_1 and i_2 are omitted, the Sort program assumes that the input is handled by the own code routine. Otherwise, both i_1 and i_2 must be specified. If $i_1 = i_2$, the input file or files are processed by the same disc drive assigned to the logical number specified.

Notes: If a multivolume file is specified, a program halt occurs upon detection of an end of volume condition; the next volume can then be mounted.

If $i_1 \neq i_2$, the Sort automatically swaps disc drives upon detecting an end of volume condition. However, if the programmer specifies two logical units for a multivolume input file, the Sort halts upon detection of an end of volume condition to alert the operator that an automatic swap is to be performed. When START is pressed, the swap is executed and the file label of the next volume is checked. A display occurs if the label check fails. At this point there are several courses of action which are described under the ILB (see 2.4) statement in the parameter specification section.

An IN statement must be present for every input file specified. For a Sort operation, one IN statement can be specified; for a merge operation, the number of IN statements depends on the number of input disc drives available (not to exceed eight).

If the e parameter is either D or S, the Sort halts on a read error to allow the operator the option of cancelling the Sort operation.

If the e parameter is omitted, an own code routine must be included to handle the error conditions.

2.4. ILB

Use: Describes the input file label.

Format: ILB f, c, g, v

Legend:
- f is the file identification which must be identical to the file name specified in the corresponding IN statement.
- c is the file creation date which must consist of six characters. The leading character may be blank.
- g is the four-character file generation number.
- v is a two-digit value that is one less than the number of the first volume of the file.

Rules: The label check must be performed by own code when own code controls the input operation; an ILB statement is not required if own code handles the input procedure.

Notes: The Sort performs a comparison between the creation date, generation number, and volume number in the label and the corresponding information in the ILB statement. If any of the values do not correspond, the program halts and a display occurs. The operator has the option of either mounting a new volume and reinitiating a label check, or continuing the Sort using the volume that does not check. In this case, the values specified in the ILB statement are replaced in the Sort by the values in the label; these values are then used to check subsequent volumes in a multivolume file.

2.5. OUT

Use: Describes the characteristics of the output file.

Format: OUT b, o_1, o_2

Legend:
- b is a two-digit value indicating the number of records per block. The value must be in the range 01 through 99 bytes. (Block size must be in the range 2 through 160 bytes; block size is equal to records per block multiplied by the record length.)
- o_1 is a two-digit value specifying the first output file to be written.
- o_2 is a two-digit value specifying the second output file to be written.

Rules: An OUT statement must appear in the statement deck unless own code is to control the output operation.

For a merge the parameters o_1 and o_2 must be specified unless own code is to control the output operation.

Both o_1 and o_2 parameters must be specified or both omitted. If $o_1 = o_2$ both output volumes are processed using the same disc drive. If $o_1 \neq o_2$ the Sort program performs an automatic disc swap upon detecting an end of volume condition.

If both o_1 and o_2 are omitted, the Sort program determines the logical unit to be used for output. A program halt is executed and a display notifies the operator of the logical output unit selected.

For a record sort or a tag sort the parameters o_1 and o_2 must not be the same as a logical unit specified for a work disc. For a tag sort, the values specified for parameters o_1 and o_2 must not be the same as that specified for a work or an input disc.

2.6. OLB

Use: Describes the file label to be created for output discs.

Format: OUT f, c, g, v, x

Legend:
- f is a file identifier consisting of eight alphanumeric characters.
- c is the creation date consisting of six characters.
- g is the four-character file generation number.
- v is a two-digit value that is one less than the number of the first volume of the file.
- x is the six-character expiration date.

Notes: The Sort creates the labels unless own code controls the output operations; own code must then create output labels.

2.7. FIELD

Use: Describes a maximum of 12 key fields.

Format: FIELD fspn, fspn, fspn, fspn, fspn, ... fspn

Legend: Each fspn describes one key field in a record; a maximum of 12 key fields may be specified. (The first field is called a major key field; the subsequent fields are called minor key fields.) The characters fspn represent the following:

- f indicates the following formats:
  - C indicates a character field; maximum field length is 160 bytes.
  - B indicates a signed binary field; maximum field length is 160 bytes.
  - D indicates a packed decimal field; maximum field length is 16 bytes.
  - Z indicates a zoned decimal field; maximum field length is 16 bytes.
- s indicates the required order of the following sorted output:
  - A indicates an ascending sequence.
  - D indicates a descending sequence.
- p indicates the position in a record of the most significant byte of the key field; p must be a three-digit number in the range 001 through 160.
- n indicates the number of bytes in the field; n must be a three-digit number in the range 001 through 160.

Rules: If own code performs the record sequencing operations, a FIELD statement must not appear in the statement deck.

At least one FIELD statement must appear in a statement deck; a maximum of 12 fspn specifications can be made.

The fields are entered on the FIELD statement card in order of their significance, that is, the leftmost field is the most significant field in a statement, or, if more than one card is used, the first card contains the most significant field information.

For variable-length records, the number of bytes in a key field must not exceed the minimum record length specified in the s parameter of the IN statement (see 2.3).

For a tag sort, the total key length in bytes must be in the range 1 through 153.

2.8. RSTRT

Use: Specifies restart points that are to be established during the execution of a Sort operation.

Format: RSTRT

Notes: The RSTRT statement does not contain parameters.

The RSTRT statement is required by the Sort operation if a restart is to be executed.

Checkpoints occur at the end of any phase and at the end of any pass within the phase.

The Sort may be interrupted at any time after completion of the Input/Internal Sort Phase (Phase 1). Subsequent restart is effected by mounting the discs where they were when the Sort was interrupted and assigning them the same logical unit numbers. The original statement deck is resubmitted to the Sort with an additional statement card (RSTRT).

2.9. MCYCL

Use: Indicates that the Sort operation is to continue in the multicycle mode if the work area becomes inadequate for sorting in the normal mode.

Format: MCYCL

Rules: If the Sort determines the logical number of the disc drive for output (o_1 and o_2 in the OUT statement are omitted), a program halt and a display occur. The operator must then place a work disc on the drive associated with the logical unit number.

Notes: The MCYCL statement does not contain parameters.

A halt display notifies the operator that a multicycle operation is required.

To restart the Sort process, the programmer must reload the Sort program, supply all the original statement cards and include a MCYCL statement card.

2.10. SORT

Use: Indicates the required characteristics of the Sort.

Format: SORT n, t, d, f, c, w

Legend: n specifies the highest memory address available to the Sort. The address must be equal to or less than that specified in the boundary table of the Minimum Operating System. If the value is omitted, the Sort uses the value in the boundary table.

t specifies the type of sort operation:
- T= tag sort
- R= record sort
- M= merge

d specifies a data reduction procedure (if this parameter is omitted, own code must be supplied):
- R indicates that the Sort is to retain records with equal key fields.
- D indicates the Sort is to delete one of two records with equal key fields.

f specifies the form of the final output for a tag sort:
- A= disc addresses only
- K= disc addresses and associated key field
- R= entire record

c specifies the number of bytes of memory to be reserved by the Sort program as common to an own code input and own code output procedure. The number of bytes must be expressed by three digits in the range 001 through 160. (The first byte of this area can be addressed by specifying the label associated with a RES instruction.)

w indicates whether a write check operation is to be performed:
- 0 does not perform a write check
- 1 write check to be performed during output of final sequenced file
- 2 write check to be performed during all phases of a record or a tag sort

Rule: When data reduction own code is used, the d parameter must be omitted.

2.11. OWNCD

Use: Specifies the type of own code routines provided by the programmer. (The use and requirements of own code routines are described in Section 3.)

Format: OWNCD a, b, c, d, e, f, g

Legend:
- a is 1 if input procedure own code is included; omit if input own code is not included.
- b is 1 if analyzation own code is included; omit if own code is not included.
- c is 1 if read error own code is included; omit if own code is not included.
- d is 1 if record sequence own code is included; omit if own code is not included.
- e is 1 if data reduction own code is included; omit if own code is not included.
- f is 1 if output analyzation own code is included; omit if own code is not included.
- g is 1 if output procedure own code is included; omit if own code is not included.

Notes: Own code exits are provided for input, merge, and output functions of the Sort program.

The OWNCD statement must include commas for the parameters that are omitted. For example:

```
OWNCD   ,,,1,,,1
```

indicates that only record sequence own code and output procedure own code are included in the Sort program.

2.12. END

Use: Indicates the end of the statement deck.

Format: END

2.13. EXAMPLE OF SORT STATEMENT DECK

The following example illustrates the statement cards required to sort a file. (Own code options are not used. No data reduction or write check is performed.)

LABEL	OPERATION	OPERAND	COMMENTS
	DISCS	01,02,03	Three work discs are available.
	IN	03,01,01,S,050,,,,02,MSTRPYRL	(03) records per block, (01,01) the first and second input discs are assigned to logical drive 01, (S) skip sector containing read error, (050) number of characters per record, (02) number of input volumes, (MSTRPYRL) file ID.
	ILB	MSTRPYRL,022768,0001,00	(MSTRPYRL) file ID, (022768) creation date, (0001) generation number, (00) volume number.
	OUT	03	Three records per output block (blocked output records).
	OLB	MSTRPYRL,022868,0002,00,032868	(MSTRPYRL) file ID, (022868) creation date, (0002) generation number, (00) volume number, (032868) expiration date.
	FIELD	CA00200 5,CA00900 2	First field: (C) character field, (A) sort on ascending key sequence, (002) position of most significant byte, (005) number of bytes in field. Second field: (C) character field, (A) sort on ascending key sequence, (009) position of most significant byte, (002) number of bytes in field.
	SORT	R,R	(R) record sort, (R) retain records with equal key fields.
	END		End of statement deck.

3. OWN CODE ROUTINES

3.1. GENERAL

Own code options available to the programmer and the requirements for linking these routines with the Sort programs are explained in this section.

The first instruction of an own code routine must specify a label provided by the Sort package. The label names and the own code functions associated with each label are as follows:

RSOC – Record Sequencing
DROC – Data Reduction
IPRO – Input Procedure
ERRO – Input Error Handling
ALYI – Analysis of Input Record
OPRO – Output Procedure
ALYO – Analysis of Output Record

The following rules must be considered when using own code routines:

- Own code routines must contain ENTRY statements for names referenced in own code by the Sort program.
- Own code must include EXTRN statements for names referenced by own code that are contained in the Sort program.
- Register 14 must not be used by own code. The address of the next sequential instruction in the Sort is stored in register 14 (by a BAL). Register 14 is the means by which own code returns program control to the Sort (by a BC).
- Own code routines can be supplied to the Sort or merge programs on cards in relocatable format. If own code is assembled using the tape or the disc assembler, only one CSECT can be specified and no COMMON can be specified.

3.2. IPRO

Use: Allows input devices other than disc to be used in a sort operation.

Operation: After reading a record into storage, own code transfers an input record to the work area defined by the Sort program then transfers control to the address of the next instruction stored in register 14. The label of the work area is WORK.

Rules: Own code must perform label checking procedures.

After processing the entire file, own code must transfer control to the instruction labeled EOI in the Sort program.

ALYI (see 3.3) of ERRO (see 3.4) own code must not be used when IPRO is used.

3.3. ALYI

Use: For own code analysis of each input record to determine whether the Sort is to retain or delete the record.

Operation: The Sort program reads a record and transfers control to the label ALYI in own code. Register 11 contains the address of first byte of the record. If the record is to be retained, own code transfers control to the Sort program at the address in register 14. If the record is to be deleted from the file, own code transfers control to the label DLET in the Sort program.

Rules: ALYI is used only if the Sort program controls the file input procedure IPRO (see 3.2) cannot be present.

3.4. ERRO

Use: For own code determination of whether a record that generated a read error is to be accepted or rejected by the Sort or merge programs.

Operation: If a read error is detected, the Sort and merge programs transfer control to the label ERRO in own code.

Rules: Register 11 contains the address of the first byte of the data containing an error.

If the data is to be processed, own code transfers control to the Sort or the merge program at the address in register 14; if the data is to be by-passed, own code transfers control to the instruction labeled RJCT in the Sort or the merge program.

ERRO must not be used if IPRO is used.

3.5. RSOC

Use: To sequence records using own code.

Operation: After records are read, the address of the first byte of one record is placed in register 11 and the address of the first byte of the other record is placed in register 12. Control is then transferred from the Sort program or the merge program to the RSOC routine where the key fields in the records are compared. Using the address in register 14, the RSOC routine then returns control to the Sort program or the merge program where the condition code is examined to determine which of the two records is to be written.

When the value of the condition code indicates that operand 1 of a compare or a test instruction is less than operand 2 of that instruction, the record referenced by register 11 is selected by the Sort program or the merge program. When the value of the condition code indicates that operand 2 is less than operand 1, the record referenced by register 12 is selected. (If the value of the condition code indicates that the key fields compared are of equal value, a record is selected arbitrarily.)

To create a file based on an ascending order of key fields, the programmer must use operand 1 of a compare instruction to reference the key field of the record associated with register 11. Conversely, a descending order of record key fields is created by using operand 1 to reference the key field of the record associated with register 12.

Rules: Own code operations must not alter either the records or the contents of registers 11 and 12.

3.6. DROC

Use: For data reduction by combining information in two or more records containing equal key fields.

Operation: The Sort program enters the own code at the label DROC to permit data reduction when either of the following conditions exist:

- the records contain key fields that are equal, or
- the condition code indicates an equal state upon returning from the RSOC (see 3.5) own code.

When a data reduction procedure is to be performed, the Sort transfers control to own code label DROC; register 12 contains the address of the first byte of the record that can be eliminated and register 11 contains the address of the first byte of the other record.

Rules: After the data in two records have been combined, own code transfers control to the label DELE in the Sort program; the Sort program eliminates the record pointed to by register 12.

When data reduction is not required, own code transfers control to the address in register 14 which returns program control to the Sort program.

Own code must not change the contents of the key fields of a record to be retained.

Own code must not change the contents of registers 11 and 12.

The d parameter in the SORT statement (see 2.10) must be omitted if the data reduction own code is included in the Sort program.

3.7. ALYO

Use: For analysis of a record prior to transfer to an output file.

Operation: Before writing a record, the Sort and merge programs enter own code at the label ALYO and load register 12 with the address of the first byte of the record. Own code determines whether the record will be written in the output file or will be deleted.

Rules: If a record is to be retained, own code transfers control to the Sort or the merge program at the address in register 14.

If the record is to be deleted, own code transfers control to the instruction label DROP in the Sort or the merge program.

If ALYO own code is included in the program, the Sort or merge program must control the output operation.

3.8. OPRO

Use: For controlling the output of the final sorted file.

Operation: Before a record is written in an output file, the Sort and merge programs load the address of the first byte of the record into register 11 and transfer control to the label OPRO in own code. To request another record, own code transfers control to the Sort or the merge program at the address in register 14.

Upon detecting an end of file condition, the Sort or merge program transfers control to the label EOFO in own code. After completing end of file procedures, own code transfers control to the Sort or merge programs at the address in register 14.

The Sort program provides a record count for use by own code. The label of the count field is RCNT and consists of a four-byte packed decimal field.

3.9. SUMMARY OF OWN CODE OPTIONS AND LINKAGES

Table 3–1 shows the own code functions that are allowed in the Sort and merge operations and indicates the positional parameters specified in the OWNCD statement (see 2.11).

OWN CODE OPTION	LABEL	POSITIONAL PARAMETER IN OWNCD STATEMENT	OWN CODE PERMITTED FOR:		
			RECORD SORT	TAG SORT	MERGE
Input procedure	IPRO	a	X		
Analyze input record	ALYI	b	X	X	
Read error procedure	ERRO	c	X	X	X
Record sequencing	RSOC	d	X		X
Data reduction	DROC	e	X		X
Analyze output record	ALYO	f	X	X	X
Output procedure	OPRO	g	X	X	X

Table 3–1. Own Code Options

Table 3–2 shows the allowable EXTRN cards in each own code routine. The X indicates allowable usage.

EXTRN NAMES	DESCRIPTION	OWN CODE ROUTINES						
		RSOC	DROC	IPRO	ERRO	ALYI	OPRO	ALYO
ILBF	Input file ID (8 bytes)			X			X	
ILBC	Input file creation date (6 bytes)			X			X	
ILBG	Input file generation number (4 bytes)			X			X	
ILBV	Input file volume number (2 bytes)			X			X	
OLBF	Output file ID (8 bytes)			X			X	
OLBC	Output file creation date (6 bytes)			X			X	
OLBG	Output file generation date (4 bytes)			X			X	
OLBV	Output file volume number (2 bytes)			X			X	
OLBX	Output file expiration date (6 bytes)			X			X	
WORK	Sort work area (up to 160 bytes)			X				
EOI	End of input when own code controls input			X				
DLET	Delete logical record					X		
RJCT	Delete physical record				X			
DELE	Record deletion for data reduction		X					
RES	Base of common area for input and output own code procedures (up to 160 bytes)			X			X	
DROP	Deletion of logical record in final output phase							X
RCNT	Record count (4 bytes)						X	

Table 3–2. EXTRN Names

Table 3–3 shows the allowable ENTRY cards in each own code routine. The X indicates allowable usage.

ENTRY NAMES	OWN CODE ROUTINES						
	RSOC	DROC	IPRO	ERRO	ALYI	OPRO	ALYO
RSOC	X						
DROC		X					
ERRO				X			
IPRO			X				
OPRO						X	
EOFO						X	
ALYO							X
ALYI					X		
EOFI					X		

Table 3–3. ENTRY Names

3.10. DISC RESIDENT OWN CODE

If own code modules are to be loaded from a disc library then control cards are required which will direct the Relocator Phase of the Sort or merge to extract and relocate the own code modules for subsequent execution. There are two control card formats, primary and secondary.

The primary control card must appear first in the control card deck and signals to the Relocator Phase that own code modules reside in a disc library. The following chart illustrates the format of the primary control card.

PRIMARY CONTROL CARD		
CARD COLUMNS	DEFINITION	DESCRIPTION
10–14	Control indicator	EBCDIC. Must contain DSKOC.
16–23	File name	EBCDIC. Eight alphanumeric characters uniquely identifying the disc library file. If these columns are blank then the name assumed is SYSFILE.
25–26	Logical unit number	EBCDIC. Two-digit number (00–63) indicating the logical unit number of the file containing disc library.

The logical unit number of the disc on which the disc library resides must not be the same as any work disc specified on the DISCS statement card.

Each secondary control card defines one own code module. The secondary control cards *must* be read in the same order as that required for own code modules, which is: RSOC, DROC, IPRO, ERRO, ALYI, OPRO, and ALYO. Each secondary control card *must* contain the module name of the desired own code module. Group name specification is optional. If only the module name is specified, the Relocator Phase of the Sort or merge will search the disc library directory from the beginning for a match on the module name. On a match condition the module will be extracted and relocated. A no-match condition results in an error display. When both group name and module name are specified, the disc library directory is first searched for a match on group name. On a match condition the search continues until there is a match on module name or an End of Group (EOG) demarcation record is encountered. Own code modules should *not* reside in nested groups.

The following chart illustrates the format of the secondary control cards.

SECONDARY CONTROL CARD		
CARD COLUMNS	**DEFINITION**	**DESCRIPTION**
16-23	Group name	EBCDIC. Eight alphanumeric characters uniquely defining the group in which the own code module will be found.
25-32	Module name	EBCDIC. Eight alphanumeric characters uniquely defining the module name of the own code module to be extracted and relocated.

The primary and secondary control cards must be followed by an END card with /* in Columns 1 and 2 and END in Columns 10–12.

All own code modules which are to be loaded from a disc library *must* be in relocatable format. If any other format is encountered during execution of the Relocator Phase, the Sort or merge will be cancelled following notification of the error condition.

4. TIMING CONSIDERATIONS

4.1. GENERAL

This section contains timing formulas which may be used to estimate sorting times for files contained on 8410 DAS Disc Units.

4.2. RECORD SORT

To estimate sorting time for a record sort, it is first necessary to calculate how many records will be sorted internally during Phase 1. The number of records sorted internally (RSI) allows calculation of the number of strings (S) produced by Phase 1, which in turn can be used to determine the number of merge passes (P) required by the Sort.

To determine RSI and S the following calculations should be performed.

(1) $\left\lfloor \frac{\text{MS-8K-OC}}{160} \right\rfloor$ = TSC

(2) $\left\lceil \frac{\text{TSC}}{40} \right\rceil$ = ST

(3) If ST>4, SET ST =4 $\left\lfloor \frac{\text{TSC}}{\text{ST}} \right\rfloor$ = SST

(4) Then: RSI = ST x SST x SB

where: SB equals sort blocking. Sort blocking is determined in the following manner:

$$\text{SB} = \left\lfloor \frac{160}{\text{RL}} \right\rfloor \quad \text{or} \quad \left\lfloor \frac{160}{\text{RL}_{max}} \right\rfloor$$

where: RL = record length (bytes)
RL_{max} = maximum record length in variable-length input file.

(5) And: $\text{S} = \left\lceil \frac{\text{N}}{\text{RSI}} \right\rceil$

where:
- MS = Available memory (number of bytes)
- K = 1000
- OC = Number of bytes used by own code
- TSC = Total working sectors
- ST = Number of internal substrings
- SST = Number of sectors per internal substring
- S = Number of strings produced by internal sort
- N = Total number of records to be sorted
- RSI = Number of records sorted internally
- $\lceil\ \rceil$ = Round integer to next higher value if quotient contains a decimal fraction
- $\lfloor\ \rfloor$ = Truncate decimal fraction of quotient.

Then it is necessary to determine the order of the merge (M). This is done in the following manner:

If MS is in the range $12K \leq MS < 16K$ then M = 5.
If MS is in the range $16K \leq MS < 20K$ then M = 6.
If MS is > 20K then M = 8.

Having determined S and M, P can be obtained by referring to the following charts:

Chart 1

M = 5	
Strings (S)	Passes (P)
1–5	1
6–25	2
26–125	3
126–625	4
626–3125	5
3126–15625	6
15626–78125	7

Chart 2

M = 6	
Strings (S)	Passes (P)
1–6	1
7–36	2
37–216	3
217–1296	4
1297–7776	5
7777–46656	6

Chart 3

M = 8	
Strings (S)	Passes (P)
1–8	1
9–64	2
65–512	3
513–4096	4
4097–32768	5
32769–262144	6

Having determined M, P, and S, sort time for each of the three phases of the Sort can now be estimated using the formulas given below.

Phase 1 (T1)

$$T1 = \left\lceil \frac{N}{BI} \right\rceil 55 + \left\lceil \frac{\left\lceil \frac{N}{BI} \right\rceil}{100} \right\rceil 47 + \left\lceil \frac{N}{SB} \right\rceil 25 + S \left\lceil \frac{ST \times SST \times 5}{100} \right\rceil 50$$

in milliseconds.

Phase 2 (T2)

$$T2 = (P-1) \left[2 \left\lceil \frac{N}{SB} \right\rceil 27.5 + 2 \left\lceil \frac{\left\lceil \frac{N}{SB} \right\rceil}{100} \right\rceil 235 \right]$$ in milliseconds.

Phase 3 (T3)

$$T3 = \left\lceil \frac{N}{SB} \right\rceil 30 + \left\lceil \frac{\left\lceil \frac{N}{SB} \right\rceil}{100} \right\rceil 235 + \left\lceil \frac{N}{BO} \right\rceil 50 + \left\lceil \frac{\left\lceil \frac{N}{BO} \right\rceil}{100} \right\rceil 47$$

in milliseconds.

where: BI = Input blocking, that is, number of records contained in one sector.
BO = Output blocking.

Therefore, total record sort time (TT) becomes:

$$TT = \frac{T1 + T2 + T3}{60000} \text{ in minutes.}$$

4.2.1. Record Sort Timing Example

Assume a file with the following characteristics is to be sorted:

- Bytes in the record (RL) = 80
- Available memory (MS) = 16,000 bytes
- Input blocking (BI) = 1
- Output blocking (BO) = 2
- Number of records to be sorted (N) = 5,000

Then:

$$TSC = \left\lfloor \frac{16000 - 8000}{160} \right\rfloor = 50$$

$$ST = \left\lceil \frac{50}{40} \right\rceil = 2$$

$$SST = \left\lfloor \frac{50}{2} \right\rfloor = 25$$

$$SB = \left\lfloor \frac{160}{80} \right\rfloor = 2$$

$$RSI = 2 \times 25 \times 2 = 100$$

$$S = \left\lceil \frac{5000}{100} \right\rceil = 50$$

Since MS equals 16000, M = 6; therefore, P = 3 is obtained from Chart 2.

Then: $T1 = (5000)55 + \left\lceil \frac{5000}{100} \right\rceil 47 + (2500)25 + 50\left\lceil \frac{2x25x5}{100} \right\rceil 58 = 348550$

$$T2 = 2\left[2(2500)27.5+2\left\lceil \frac{2500}{100} \right\rceil 235 \right] = 298500$$

$$T3 = (2500)30 + \left\lceil \frac{2500}{100} \right\rceil 235 + (2500)50 + \left\lceil \frac{2500}{100} \right\rceil 47 = 207150$$

and total sort time becomes:

$$\frac{348550 + 298500 + 207150}{60000} = 14.2 \text{ minutes}$$

4.3. TAG SORT

To estimate sorting time for a tag sort the formulas given under record sort apply with the following exceptions:

SB should be determined by $\left\lfloor \frac{160}{KL+7} \right\rfloor$

where: KL = total length of all key fields to be compared.

If the sorted output consists of disc address only, or disc address plus key information only, then T3 is calculated in the same fashion as described under record sort. However, if the entire record is to be written, use the following formula to calculate T3 time:

$$T3 = \left\lceil \frac{N}{SB} \right\rceil 30 + \left\lceil \frac{\left\lceil \frac{N}{SB} \right\rceil}{100} \right\rceil 235 + \left\lceil \frac{N}{BO} \right\rceil 50 + \left\lceil \frac{\left\lceil \frac{N}{BO} \right\rceil}{100} \right\rceil 47 + AS(N)$$

where: $AS = \left\lceil \frac{\left\lceil \frac{N}{BI} \right\rceil}{100} \right\rceil + 47$

NOTE: If AS > 147, set AS = 147.

4.3.1. Tag Sort Timing Example 1

When necessary to determine sorting time for a file which has the following characteristics:

- Bytes in the record (RL) = 80
- Available memory (MS) = 16,000 bytes
- Input blocking (BI) = 1
- Output blocking (BO) = 2, and the entire record is to be written
- Number of records to be sorted (N) = 5,000
- Total length of all key fields (KL) = 3 bytes

Then: $TSC = \left\lfloor \frac{16000 - 8000}{160} \right\rfloor = 50$

$ST = \left\lceil \frac{50}{40} \right\rceil = 2$

$SST = \left\lfloor \frac{50}{2} \right\rfloor = 25$

$SB = \left\lfloor \frac{160}{10} \right\rfloor = 16$

$RSI = 2 \times 25 \times 16 = 800$

$S = \left\lceil \frac{5000}{800} \right\rceil = 7$

Since MS equals 16000, M = 6, and P = 2 is obtained from Chart 2.

So: $T1 = (5000)\,55 + \left\lceil \frac{5000}{100} \right\rceil 47 + \left\lceil \frac{5000}{16} \right\rceil 25 + 7 \left\lceil \frac{2 \times 25 \times 5}{100} \right\rceil 58 = 286393$

$T2 = 1 \left[2(313)27.5 + 2 \left\lceil \frac{313}{100} \right\rceil 235 \right] = 19070$

$T3 = (313)30 + \left\lceil \frac{313}{100} \right\rceil 235 + (2500)50 + \left\lceil \frac{2500}{100} \right\rceil 47 + 97(5000) = 621483$

and total sort time becomes:

$$\frac{286393 + 19070 + 621483}{60000} = 15.5 \text{ minutes}$$

4.3.2. Tag Sort Timing Example 2

If the file described in the preceding example is to be sorted, but instead of the entire record, only the disc addresses are required as final output, and they are to be blocked twenty-two (22) per sector, the values computed in the previous example apply and

$$T1 = (5000)55 + \left\lceil \frac{5000}{100} \right\rceil 47 + \left\lceil \frac{5000}{16} \right\rceil 25 + 7 \left\lceil \frac{2 \times 25 \times 5}{100} \right\rceil 58 = 286393$$

$$T2 = 1 \left[2(313)27.5 + 2 \left\lceil \frac{313}{100} \right\rceil 235 \right] = 19070$$

$$T3 = 313(30) + \left\lceil \frac{313}{100} \right\rceil 235 + \left\lceil \frac{5000}{22} \right\rceil 50 + \left\lceil \frac{228}{100} \right\rceil 47 = 21871$$

So total sort time becomes:

$$\frac{286393 + 19070 + 21871}{60000} = 5.5 \text{ minutes}$$

4.4. DISC WRITE CHECK OPERATIONS

Whenever the disc write check operation is requested, sort times are increased considerably. The formulas below should be used in estimating sort times if the write check operation has been specified for any phase.

Phase 1 (Record or Tag Sort)

$$T1 = \left\lceil \frac{N}{BI} \right\rceil 55 + \left\lceil \left\lceil \frac{N}{BI} \right\rceil \frac{}{100} \right\rceil 47 + \left\lceil \frac{N}{SB} \right\rceil 75 + S \left\lceil \frac{ST \times SST \times 5}{100} \right\rceil 58$$

in milliseconds.

Phase 2 (Record or Tag Sort)

$$T2 = (P-1) \left[2 \left\lceil \frac{N}{SB} \right\rceil 52.5 + 2 \left\lceil \left\lceil \frac{N}{SB} \right\rceil \frac{}{100} \right\rceil 235 \right]$$ in milliseconds.

Phase 3 (Record Sort, Tag Sort w/ disc address or tag output)

$$T3 = \left\lceil \frac{N}{SB} \right\rceil 30 + \left\lceil \left\lceil \frac{N}{SB} \right\rceil \frac{}{100} \right\rceil 235 + \left\lceil \frac{N}{BO} \right\rceil 100 + \left\lceil \left\lceil \frac{N}{BO} \right\rceil \frac{}{100} \right\rceil 47$$

in milliseconds.

Phase 3 (Tag Sort with Record Output)

$$T3 = \left\lceil \frac{N}{SB} \right\rceil 30 + \left\lceil \frac{\left\lceil \frac{N}{SB} \right\rceil}{100} \right\rceil 235 + \left\lceil \frac{N}{BO} \right\rceil 100 + \left\lceil \frac{\left\lceil \frac{N}{BO} \right\rceil}{100} \right\rceil 47 + AS(N)$$

in milliseconds

Table 4–1 shows the differences in sort times for the three preceding examples when a write check operation has been specified.

	NO WRITE CHECK (MINUTES)	WRITE CHECK OUTPUT PHASE ONLY (MINUTES)	WRITE CHECK ALL PHASES (MINUTES)
RECORD SORT EXAMPLE	14.2	16.3	22.6
TAG SORT EXAMPLE 1	15.5	17.6	18.1
TAG SORT EXAMPLE 2	5.5	5.7	6.2

Table 4–1. Sort Timings

The time involved in mounting or dismounting discs is *not* reflected in the foregoing discussion of sort timing considerations.

Timing formulas are based on the assumption that at least two work discs are available to the Sort.

MINIMUM DISC REQUIREMENTS				
TYPE SORT	NO OWN CODE	OWN CODE IN	OWN CODE OUT	OWN CODE IN & OUT
Record	2†	2	2	1
Tag -				
Tags	2†	NA	2	NA
Address	2†	NA	2	NA
Records	3	NA	2	NA
Merge	3	NA	2	NA

† An input unit can be designated for output.

NA = Not applicable or not consistent.

5. HALTS AND DISPLAYS

5.1. GENERAL

This section describes the halts and displays which can occur during the execution of a Sort or a merge program.

5.2. HALTS - PHASE O

To continue control card validation after a display, press the START switch. If a control card error has occurred, the Sort is terminated at the completion of the card deck validation.

The Sort can be cancelled after any error display by keying a nonzero into location 4 and pressing the START switch.

HEXIDECIMAL DISPLAY	REASON	ACTION
1001	Illegal statement name detected.	Press START to ignore 2nd to last card in output hopper, or cancel.
1008	Output block size > 160 bytes.	Cancel.
1009	Work disc not available for own code used with merge program.	Cancel.
1010	Too many DISCS cards.	Continue validation or cancel.
1011	No work discs specified in the sort.	Cancel.
1012	Improper field length or value on a DISCS card.	Continue validation or cancel.
1013	To many work discs specified.	
1014	A work disc is specified more than once.	
1020	Not all required fields are on the IN card.	

HALT NUMBER	REASON	ACTION
1021	Input block size > 160 bytes.	Continue validation or cancel.
1022	Blocking factor on IN card is not $1 \leq$ I.B.F. ≤ 99.	
1023	Input unit designation on IN card is not valid.	
1024	Invalid 'e' field on IN card.	
1025	Record length not $0 <$ R-L- < 160 or field length $\neq$ 3 bytes.	
1026	Variable-length and fixed-length were both specified on the IN card, or the variable lengths were specified incorrectly.	
1027	Invalid "number of volumes" field on IN card.	
1028	File ID field length is $\neq$ 8 on the IN card.	
1029	Too many fields on the IN card.	
1030	Not all required fields are on the ILB card.	
1031	File ID field length on ILB card $\neq$ 8.	
1032	Creation date field length on ILB card $\neq$ 6.	
1033	Generation number field length $\neq$ 4.	
1034	Starting volume number field length on ILB card $\neq$ 2.	
1035	Too many fields on the ILB card.	
1036	Invalid blocking factor field on OUT card.	

HALT NUMBER	REASON	ACTION
1037	Invalid output disc designation on OUT card.	Continue validation or cancel.
1038	Too many fields on the OUT card.	
1039	All required fields not on the OUT card.	
1040	All required fields not on the OLB card.	
1041	File ID field length on OLB card $\neq$ 8.	
1042	Creation date field length on OLB card $\neq$ 6.	
1043	Generation number field length on OLB card $\neq$ 4.	
1044	Output volume number field length on OLB card $\neq$ 2.	
1045	Expiration date field length on OLB card $\neq$ 6.	
1046	Too many fields on OLB card.	
1047	Improper format of FIELD card.	
1048	Incorrect text on FIELD card.	
1049	Too many fields specified by FIELD cards.	
1050	Too many SORT cards.	
1051	Too many OWN CODE cards.	
1052	Too many ILB cards.	
1053	Too many OUT cards.	
1054	Too many OLB cards.	
1055	Too many IN cards, or none at all.	
1056	File ID's on IN and ILB cards do not match.	Cancel.

HALT NUMBER	REASON	ACTION
1057	All key fields are not within the bounds of the records.	Cancel.
1060	All required fields are not on the SORT card.	Continue validation or cancel.
1061	Highest memory byte noted on SORT card is not < value in MOS boundary table.	
1062	Invalid specification of type of sort.	
1063	Highest memory byte field on SORT is improper length.	
1064	Data reduction indicator on SORT card is invalid.	
1065	Invalid tag sort output field on SORT card.	
1066	Tag sort output indicator is present but a tag sort was not requested.	
1067	Invalid reserve field on the SORT card.	
1068	Invalid write check indicator on SORT card.	
1069	Too many fields on SORT card.	
1070	An invalid field is on the OWNCD card.	
1071	Too many fields on the OWNCD card.	
1072	The SORT or IN card is missing.	Cancel.
1073	Illegal combination of cards and/or own code options when requesting a merge.	
1074	Input and output disc unit are not unique in a merge.	

HALT NUMBER	REASON	ACTION
1075	Too many input discs specified.	Cancel.
1076	Illegal combination of own code and statement cards when doing a record sort.	
1077	Restart was illegally requested: either the input phase was not completed previously or the statement cards are not exactly the same as they were in the initial sort.	
1078	Combination of input and output units is > 8.	
1079	No input disc may contain more than one input file.	Continue validation or cancel.
1080	No data reduction was specified on either the OWNCD card or the SORT card, or it was specified on both.	Cancel.
1081	Both the input volume count on the IN card and IPRO on the SORT card are missing.	
1082	Input and work disc numbers may not overlap in a tag sort.	
1083	Own code input is not legal in a tag sort.	
1084	Input unit numbers and output unit number overlays in a tag sort with record output.	
1085	RSOC and DROC are illegal own code routines in a tag sort. The d field must be specified on the SORT card.	
1086	A DISCS, FIELD, or ILB card is missing.	
1087	The sum of the key field lengths is > 153.	

HALT NUMBER	REASON	ACTION
1088	MCYCL was illegally requested. Multicycle information is not on the checkpoint disc; or, the statement cards are not identical to those used in the initial sort.	Press START to cancel.
1090	MCYCL and RSTRT were illegally requested in the same run.	
1095	A specified output was also specified as a work disc.	
1096	The read error option on the IN card and the ERRO own code routine were either both specified or neither specified. One and only one must be specified.	
1097	When there are no OUT and/or OLB cards, the OPRO own code routine must be used and ALYO is not allowed.	
1098	The number of input discs is greater than the number of noninput work discs.	
1099	Due to a previous error the phase (and thus the sort) is being terminated. Fix the statement cards and reload Phase 0.	
11xx	The disc on logical unit xx does not contain a single file called SORTWORK, or a read error was detected when checking the file name. By pressing START the disc on unit xx will again be checked for the SORTWORK file name.	(1) Press START to check the same disc again. (2) Mount a new SORTWORK disc and press START. (3) Cancel.
12xx	Unable to read the checkpoint from logical unit xx when a RSTRT or MCYCL was requested.	(1) Press START to try read again. (2) Cancel.

5.3. RELOCATOR PHASE DISPLAYS

DISPLAY	REASON	ACTION
3D01	Card read not an A, J, K, H, Q, or Y.	Press START to read next card or ENTER a nonzero character into location 4 by means of the ALTER, and press START to cancel the sort.
3D02	Own code input not in proper sequence, that is, text before external definitions, etc.	Rerun sort. Press START to cancel.
3D03	Amount of own code will not fill in allotted area on disc checkpoint. A maximum of 14,601 bytes is allowed.	Reduce amount of own code. Rerun sort. Press START to cancel.
3D04	Disc write error.	Press START to display eeuu where ee = disc error, uu = logical unit number. Press START again to retry or enter a nonzero character into location 4 and press START to cancel.
3D05	Unrecoverable disc error.	Press START to display eeuu. Correct malfunction and rerun sort.
3D06	Own code missing (card loadable own code).	Clear reader and place missing own code into reader, feed a card and press START, or enter a nonzero character into location 4 and press START to cancel.
	(Disc loadable own code)	Clear reader and place missing secondary control card(s) into reader, followed by an END card.

HALT NUMBER	REASON	ACTION
3D07	Phase 1 cannot execute due to amount of own code present.	Reduce own code used. Press START to cancel. Rerun sort.
3D08	Phase 2 cannot execute due to amount of own code present.	
3D09	Phase 3 cannot execute due to amount of own code present.	
3D10	No A card (element definition card) or more than one A card for own code module.	Press START to cancel. Correct input to Relocator Phase. Rerun sort.
3D11	Merge cannot execute due to amount of own code present.	Reduce own code or number of input files. Press START to cancel. Rerun sort.
3D12	END card out of sequence (card loadable own code). END card encountered before transfer card of own code module.	Correct card input to Relocator Phase. Press START to cancel. Rerun sort.
3D13	Own code module contains invalid EXTRN.	Press START to cancel. Correct error condition and rerun sort.
3D14	Own code module contains invalid ENTRY.	
3D15	Own code module not relocatable code.	
3D16	Own code module present that was not specified on OWNCD statement card.	
3D17	An own code module was input more than once.	
3D18	Own code modules not in proper sequence.	
3D19	An own code module did not have all of the required ENTRIES.	
3D95	Invalid logical unit number of disc unit containing disc library file (this unit assigned to sort as input or output).	

<table>
<tr><th>DISPLAY</th><th>REASON</th><th>ACTION</th></tr>
<tr><td>3D96</td><td>Own code module cannot be located in group specified on secondary control card.</td><td rowspan="4">Press START to read next control card or enter a nonzero character into location 4 and press START to cancel.</td></tr>
<tr><td>3D97</td><td>Own code Module whose name is specified on secondary control card cannot be located in disc library.</td></tr>
<tr><td>3D98</td><td>No module name specified on secondary control card.</td></tr>
<tr><td>3D99</td><td>Group name specified on secondary control card cannot be located in disc library.</td></tr>
<tr><td>4Exx</td><td>Disc library file cannot be located on logical unit xx.</td><td>Mount new volume and press START for new search or enter a nonzero character into location 4 and press START to cancel.</td></tr>
<tr><td>4Fxx</td><td>Disc library file on logical unit xx is on unit also specified to be a SORTWORK unit.</td><td>Press START to cancel. Correct error condition and rerun sort.</td></tr>
<tr><td>4De0</td><td>Disc error when reading disc library file.

e | Error
1 | Unrecoverable abnormal line
2 | Unrecoverable output bus check
5 | Catastrophic failure
7 | Nonoperational channel
8 | Invalid function
F | Unload buffer not successful</td><td>If START is pressed after errors 5, 7, or 8, sort is cancelled. If START is pressed after errors 1, 2, or F, disc operation will be retried. Enter a nonzero character into location 4 and press START to cancel.</td></tr>
</table>

e = error tape
xx = logical unit number

5.4. INTERNAL SORT PHASE HALTS

For any error halt, a nonzero keyin into location 4 will result in the execution of the cancel macro.

The x in position 1 of the halts listed below will be a 1 in the record sort program and a 2 in the tag sort program.

HALT NO.	REASON	ACTION
x301	Read error detected on input.	Record that an errored sector of record is being sorted. Press START.
x302	Read error detected on input.	Record that an errored sector of records was dropped. Press START.
x303	Read error detected on input.	Press START to cancel the sort.
x304	Illegal memory allocation on the SORT card or in the MOS boundary value table.	
x305	The file to be sorted is an indexed sequential file. This is an illegal file type.	
x306	The file type is other than sequential or indexed sequential.	Press START to sort the file. or Insert a nonzero keyin to cancel.
x307	Read error detected when reading own code information from the checkpoint. See Note 1.	Press START to retry the read. or Cancel.
x308	1308-Record sort is in memory and a tag sort was requested. 2308-Tag sort is in memory and a record sort was requested.	Cancel. Then, change either the parameter on the SORT card or the program to be used.
x309	Read error detected when reading own code from the checkpoint disc.	See Note 1. If error repeats, mount a new SORTWORK disc and restart the sort.
x310	Error when reading label or extent information from the input disc.	See Note 1.
x311	Read error detected when reading multicycle information from the input disc.	See Note 1.
x312	Error detected on write to disc.	See Note 1.
x313	Disc malfunction on data read or write check.	Press START to display NNXX. Then press START to cancel sort.

HALT NO.	REASON	ACTION
x314	Error when writing checkpoint to disc.	See Note 1.
x315	Multicycle indicator. The amount of input exceeded the available working storage. When this sort is ended, remount the input, work, and output discs on the same units and run the sort again with the same statement cards plus the MCYCL card. The remaining records on the input file will then be sorted. Use the merge program to merge the sorted output files into one file.	Press START to proceed.
x316	Wrong input units specified for MCYCL run or the wrong work discs have been mounted.	Press START to cancel.
x317	No input was received from IPRO.	Press START to cancel.

NOTE 1: Display format is nnxx. At some read and write error halts the error number (nn) and the unit number (xx) will be displayed by pressing START without a cancel keyin. Press START again to retry disc operation.

HALT NO.	REASON	ACTION
x4uu	The creation date or generation number on input unit uu does not match the information on the statement cards. See Note 2.	Options: (1) Disregard halt and press START. The new value from the present input disc will be saved for comparisons on ensuing volumes. (2) Mount the proper input disc, insert a nonzero keyin other than 01 and press START. (3) Insert a keyin of 01 to cancel.
x5uu	The next volume of input is about to be read from unit uu.	Mount the proper input and press START.
x6uu	Volume number not ascending order on unit uu.	Options: See x4uu.
x7uu	Improperly labeled input disc on unit uu.	Mount the proper input disc. Press START to reread. Press or cancel.

NOTE 2: All unit designations refer to logical unit numbers.

5.5. PHASE 2 DISPLAYS

DISPLAY		REASON	ACTION
Record Sort	Tag Sort		
1801	2801	Sort type indicated on checkpoint TAG SORT/RECORD SORT or MERGE. RECORD SORT/TAG SORT being run.	Rerun sort with correct Phase 2.
18x0	28x0	Disc error. x specifies error condition. X — Error 1 — Unrecoverable abnormal line 2 — Unrecoverable output bus check 5 — Catastrophic failure 7 — Nonoperational channel 8 — Invalid function F — Unload buffer not successful	If START is pressed after errors 5, 7, or 8, sort is cancelled. If START is pressed after errors 1, 2, or F, disc operation will be retried. Enter a nonzero character into location 4 by means of the ALTER switch and press START to cancel the sort.
19xx	29xx	Disc mounted on logical unit xx not a valid SORTWORK disc.	Mount a valid SORTWORK disc on logical unit spefied and press START; or, enter a nonzero character into location 4 by means of the ALTER switch and press START to cancel the sort.

5.6. PHASE 3 DISPLAYS

<table>
<tr><th colspan="2">DISPLAY</th><th>REASON</th><th>ACTION</th></tr>
<tr><th>RECORD SORT</th><th>TAG SORT</th><th></th><th></th></tr>
<tr><td>1A01</td><td>2A01</td><td>SORTWORK disc mounted on logical unit specified by halt 1Cxx, 1Dxx, or 1Exx is not a valid SORTWORK disc.</td><td>Mount a valid SORTWORK disc on logical unit specified by previous halt and press START; or, enter a nonzero character into location 4 by means of the ALTER switch and press START to cancel the sort.</td></tr>
<tr><td>1A02</td><td>2A02</td><td>Sort type indicated on checkpoint TAG SORT/RECORD SORT or MERGE. RECORD SORT/TAG SORT being run.</td><td>Rerun sort with correct Phase 3.</td></tr>
<tr><td>1Ax0</td><td>2Ax0</td><td>Disc error. x specifies error condition.

x | Error
1 | Unrecoverable abnormal line
2 | Unrecoverable output bus check
5 | Catastrophic failure
7 | Nonoperational channel
8 | Invalid function
F | Unload buffer not successful</td><td>If START is pressed after errors 5, 7, or 8, SORT is cancelled. If START is pressed after errors 1, 2, or F, disc operation will be retried. Enter a nonzero character into location 4 by means of the ALTER switch and press START to cancel the sort.</td></tr>
<tr><td>1Bxx</td><td>2Bxx</td><td>Disc on logical unit xx does not contain correct checkpoint information. This halt will only occur during a restart operation.</td><td>Mount the correct disc onto logical unit specified and press START, or enter a nonzero character into location 4 by means of the ALTER switch and press START to cancel the sort.</td></tr>
</table>

DISPLAY		REASON	ACTION
RECORD SORT	TAG SORT		
1Cxx	2Cxx	Sort ready to output final sequenced file on logical unit xx.	If a SORTWORK disc is on logical unit specified, press START; or mount a SORTWORK disc and press START.
1Dxx	2Dxx	Sort ready to output final sequenced file on logical unit xx. Dismounted disc becomes checkpoint disc for a multicycle or restart operation.	Specified disc must be removed. Mount a SORTWORK disc on which final output will be written on unit specified. Press START.
1Exx	2Exx	End of volume on logical unit xx when outputting final sorted file.	Dismount filled disc and mount a SORTWORK disc on unit specified for continuation of output processing. Press START.
1Fxx	2Fxx	End of file on logical unit xx when outputting final sorted file.	Dismount final output disc. Press START.
1FFF	2FFF	End of record Sort/Tag sort.	None required.

5.7. PHASE 4 DISPLAYS

DISPLAY	REASON	ACTION
31xx	Invalid output disc.	Load new disc to output unit, press START. Key in any nonzero to cancel job.
32xx	Invalid input label.	Load new disc to input unit, press START. Key in any nonzero to cancel job.
33xx	Invalid input creation date.	Press START to force label. Replace disc and key in any nonzero to test new disc.
34xx	Invalid input generation number.	
35xx	Invalid volume number.	
36xx	File loaded is Index Sequential or Sysfile.	Load new file to unit and press START. Key in any nonzero to cancel job.
37xx	File loaded is not defined or direct access.	Press START to merge file. Any nonzero keyin will cancel job.
38xx	Unrecoverable disc I/O error.	Press START to cancel job.
39xx	Recoverable disc I/O error.	Press START to try again. Any nonzero keyin will cancel job.
3A00	Invalid initialization (Phase 0 not properly executed or improper criteria used).	Press START to cancel job.
3A01	Cancel job requested, as result of sequence error.	Press START to cancel job.
3A02	Read error noted, automatic cancellation requested by criteria.	Press START to cancel job.
3A03	Read error noted, automatic ignore option requested by criteria.	Press START to ignore error. Any nonzero keyin will cancel job.

DISPLAY	REASON	ACTION
3A04	End of volume noted on output. No alternate available.	Load new work disc, press START to continue.
3A05	Read error noted, automatic skip option requested by criteria.	Press START to skip record. Any nonzero keyin will cancel job.
3Bxx	End of input volume noted. (Alternate unit available.)	Place next sequential volume on unit displayed. Press START to continue.
3Cxx	End of input volume, no alternate unit available.	Place next sequential volume on unit displayed and press START to continue.
3Exx	End of output volume noted. (Alternate unit is available.)	Place new work disc on unit displayed, press START to continue.
3Fxx	Sequence error noted on input file.	Enter FF to drop out of sequence record and continue. Press START to cancel.
3FFF	End of job.	Press START for supervisor EOJ return.