SY33-8556-3
File No. S370/S4300-31

Systems

DOS/VSE Linkage Editor Logic

Program Number 5745-SC-LNK

IBM

Summary of Amendments

Edition SY33-8556-3 documents:

Support of Fixed Block Architecture (FBA) disk devices

Fourth Edition (February, 1979)

This is a major revision of, and obsoletes SY33-8556-2 and Technical Newsletters SN33-8785 and SN33-9218.

This edition applies to the IBM Disk Operating System/Virtual Storage Extended, DOS/VSE, and to all subsequent releases until otherwise indicated. Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 Bibliography, GC20-0001, for the editions that are applicable and current.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Laboratory, Publications Department, Schoenaicher Strasse 220, D-7030 Boeblingen, Germany. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

TABLE OF CONTENTS

PREFACE

This Program Logic Manual (PLM) is a guide to the IBM Disk Operating System/Virtual Storage Extended (DOS/VSE), Linkage Editor program. It supplements the program listings by providing descriptive texts and flowcharts.

For overall system control logic description, this PLM is to be used with six other PLMs:

DOS/VSE Supervisor Logic, SY33-8551

DOS/VSE Error Recovery and Recording Transients Logic, SY33-8552

DOS/VSE Logical Transients and Dump Phases Logic, SY33-8553

DOS/VSE System Serviceability Aids Logic, SY33-8554

DOS/VSE Initial Program Load and Job Control Logic SY33-8555

DOS/VSE Librarian Logic, SY33-8557

Publications that aid in the use of this manual are:

OS/VS, DOS/VSE and VM/370 Assembler Language, GC33-4010

Guide to DOS/VSE Assembler, GC33-4024

DOS/VSE System Control Statements, GC33-5376

DOS/VSE Operating Procedures, GC33-5378

Publications related to the subject of this manual are:

DOS/VSE System Management Guide, GC33-5371

DOS/VSE Data Management Concepts, GC24-5138

DOS/VSE Macro User's Guide, GC24-5139

DOS/VSE Macro Reference, GC24-5140

DOS/VSE System Generation, GC33-5377

DOS/VSE Messages, GC33-5379

DOS/VSE LIOCS Volume 1, SY33-8559

Titles and abstracts of the other related publications are listed in the IBM System/370 Bibliography, GC20-0001.

PUBLICATION ORGANIZATION

This manual consists of five major sections:

- Introduction to the Linkage Editor.
- Method of Operation, describing the program function, the structure of object modules as input, I/O flow and storage layout.
- Program Organization, describing in detail the library record formats, the control flow and various features of the program.
- General and detailed charts showing the logic flow of the linkage editor program. General charts are identified by two-digit numerals such as 01, detailed flowcharts by letters such as AA through ZZ.
- Appendixes which include a label list, phase-to-module and message cross references for use in analyzing program errors, a brief description of the system residence organization, and the linkage editor External Symbol Dictionary (ESD) processing and map.

In this publication, system and component names as listed below should be read as indicated:

System/component name	To be read as
DOS/VS	DOS/VSE (see Note below)

Note: Unless that name explicitly refers to DOS/VS Release 34 or an earlier DOS/VS release.

FIGURES

CHARTS

General Charts

Detail Charts

INTRODUCTION

All programs to be executed in the DOS/VS environment must be link-edited and stored in the core image library before they can be executed. The core image library is either on SYSRES (the system core image library) or on SYSCLB (a private core image library). The linkage editor program accomplishes the link edit function operating in one of three modes:

1. Catalog mode. An object module is link-edited and permanently stored in the core image library. The core image directory of cataloged phases is updated in this mode of operation.

2. Load and execute mode. An object module is link-edited for temporary storage in the core image library and is immediately executed.

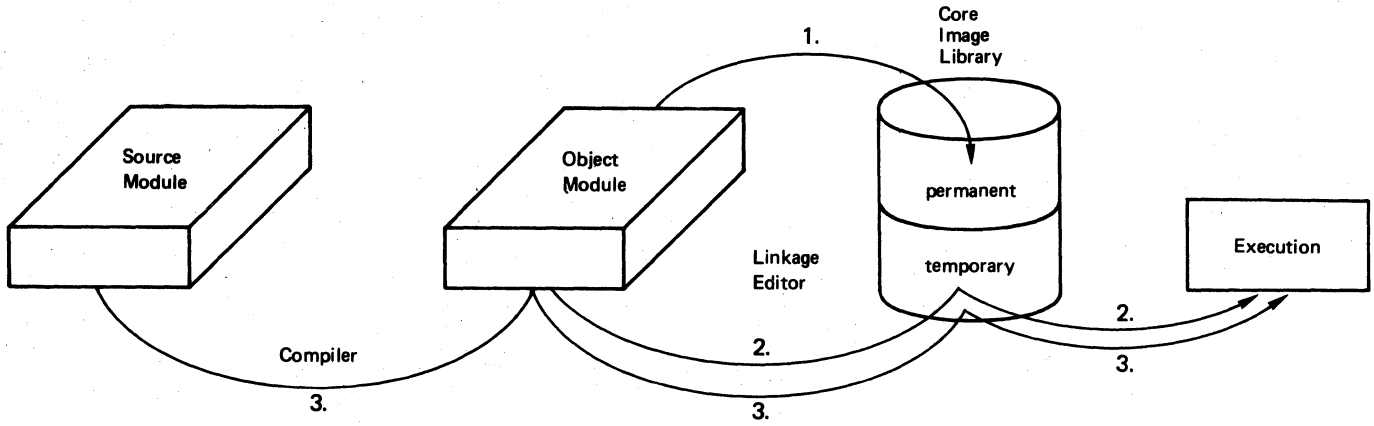
3. Assemble and execute mode. A source module is assembled or compiled. The object module (output) is link-edited for temporary storage in the core image library and is immediately executed.

Note: When operating in catalog mode, the core image directory for linked phases is updated. A reenterable program can be cataloged as eligible to be loaded into the shared virtual area (SVA eligible). Phases resident in the SVA can be shared concurrently by programs running in different partitions.

Job control calls the linkage editor program when a // EXEC LNKEDT control statement is read.

For updating the core image library directory, control is given to $MAINDIR. If OPTION CATAL was specified, the directory for cataloged phases is updated. If OPTION LINK was specified, the link area of the directory is updated. After either one of these functions is completed, control is returned to the linkage editor and then passed on to job control.

The linkage editor program can run in either the background or a foreground partition. If it runs in the foreground, you must assign a private core image library (SYSCLB). In the background partition, the linkage editor defaults to the system core image library if no private core image library is assigned. The linkage editor issues a diagnostic message and terminates abnormally when you assign the private core image library across partitions.



PROGRAM FUNCTION

The linkage editor prepares programs for execution on DOS/VS. Input for the linkage editor are the relocatable modules produced by the language translators. The linkage editor processes these modules into program phases which may be immediately executed or cataloged into the core image library.

The linkage editor control cards direct the program to read an input module(s) and tc form phases from the control section within the modules. Figure 1 shows hcw phases can be formed. The linkage editor relocates the origin of each control section in the phase, assigns each phase an area of storage and a transfer address, and modifies the contents of the address constants in the phase.

Sample of a 2-module input resulting in a 3-phase output

Language Translator Output	Linkage Editor Output
Module A ESDs TXT-CSECTA TXT-CSECTB TXT-CSECTC RLDs	Phase 1 CSECTA CSECTB
Module B ESDs TXT-CSECTD TXT-CSECTE TXT-CSECTF TXT-CSECTG RLDs	Phase 2 CSECTC CSECTD CSECTE
	Phase 3 CSECTF CSECTG

Figure 1. Example of a Module-Phase Relationship

The relocation factor for each control section is determined and saved by building a table called the control dictionary. This table contains the linkage editor phase definitions and the module ESD items. When complete, the table provides sufficient information for determining the location of each control section and for resolving any references between control sections.

The module TXT items are then built into phase blocks. The RLD items (address constants) are modified and inserted into the text. A transfer address is determined for each phase. Unresolved address constants will appear as zero RLD items in relocatable phases.

The linkage editor will also accept as input phases retrieved by the CSERV program from the core image library. The purpose of this function is to allow recataloging of an already link-edited phase to a different core image library.

OBJECT MODULES AS INPUT

The input to the linkage editor consists of object modules and linkage editor control cards. Each module is the output of a complete language translator run. It consists of dictionaries and text for one or more control sections.

The dictionaries contain the information necessary for the linkage editor to resolve references between different modules. The text consists of the actual instructions and data fields of the module.

Six card types can be produced, by the language translators or by the programmer, to form a module. They appear in the following order:

Card Type	Definition
ESD	External symbol dictionary
SYM	Ignored by linkage editor
TXT	Text
RLD	Relocation list dictionary
REP	Replacement to text made by the programmer
END	End of module

The *external symbol dictionary* contains control section definitions and intermodule references. When the linkage editor has the ESDs from all modules, it can relocate the sections and resolve the references. Five types of entries are defined in the control dictionary.

ESD Type	*Definition*
SD	Section definition: provides control section name, assembled origin and length. Generated by a named START or a named CSECT in a source module.
WX	Generated by Weak External Reference (WXTRN), which has a function similar to EXTERN, except that WXTRN suppresses AUTOLINK. The linkage editor treats WX as an ER, NOAUTO.
PC	Private code: provides assembled origin and length for an unnamed control section.
LD/LR	Label definition: specifies the assembled address and the associated SD of a label that may be referred to by another module. The LD entry is termed LR (Label Reference) when the entry is matched to an ER entry.
ER	External reference: specifies the location of a reference made to another module. ER is generated by EXTRN or a V-type address constant in a source module.
CM	Common: indicates the amount of main storage to be reserved for common use by different phases. CM is generated by CCM in a source module.

The *relocation list dictionary* identifies portions of text that must be modified on relocation (address constants).

When the linkage editor reads a module, it stores ESDs in its control dictionary, writes TXT and REP items in core image blocks in the library, and writes RLD items on an RLD file. Each item that is identified by the language translators with an ESID number is re-identified by the linkage editor with a control dictionary number to avoid duplication of identification between modules. For the ESD processing, see *Appendix E*.

I/O FLOW

The I/O flow for the linkage editor program consists of:

- Input from:

  SYSLNK - the system link library
  SYSRLB - a private relocatable library
  SYSRES - the system relocatable
           library on SYSRES
  SYS001 - the I/O Workfile

- Output to:

  SYSLST - the list device
  SYSLOG - the logging device
  SYSRES - the system CIL on SYSRES
  SYSCLB - a private core image library

Figure 2 shows the I/O flow of the linkage editor.

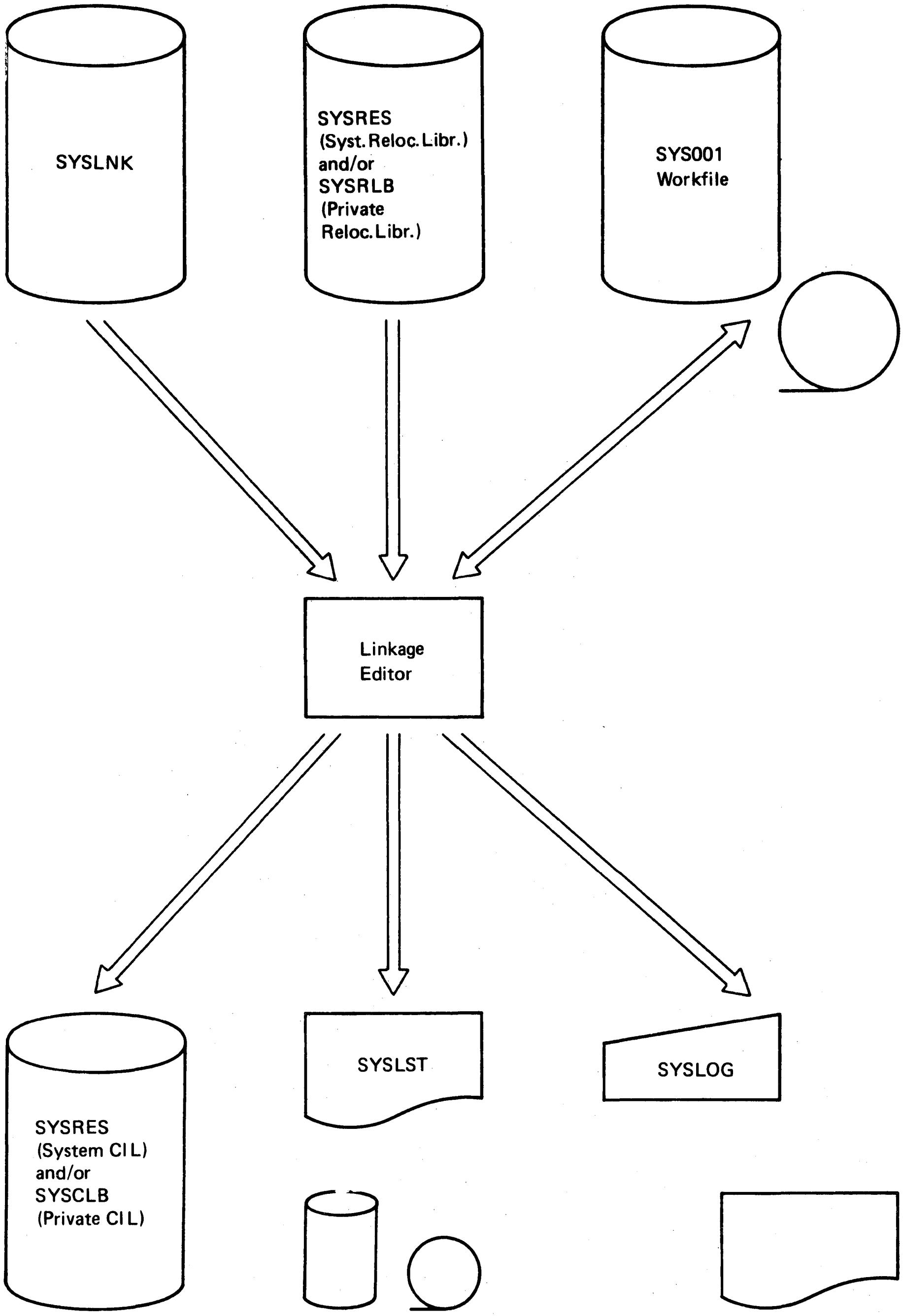


Figure 2. Linkage Editor I/O Flow

STORAGE LAYOUT

Addresses in Col. "Start at approx." are relative to the beginning of partition.

Start at approx.	Addressed by	Description	Length approx.
0	IJBLNK	Part of linkage editor permanent in storage	19K
19K	IJBINL VCATND	INL-CSECT, used only during initialization, thereafter overlaid by I/O-areas	4K
		Aligned on 1K	
19K	AWKARE	I/O-area for core image lib	1K
		Aligned on 2K (page-boundary)	
20K 22K	RLDIRAR1 RLDIRAR2	2 I/O-areas for reading directory of relocatable library (used alternately)	2K 2K
24K	IOAREA1	I/O-area for reading from SYS001	0,5K
		Aligned on 2K (page-boundary)	
26K-320 bytes		Save-area for 1 record of a member in the relocatable library (handling spanned records: already read part of the record is moved before the part being read by a new read-command)	320 bytes
26K	FRMBUF	I/O-area for reading member of relocatable library Also used as I/O-area during initialization (reading library descriptor records)	2K
28K	FLNBUF	I/O-area for reading from SYSLNK (default-size=1K, may be changed by user)	1K or ?K
29K	LTMINE	Workarea (called Linkage Table) for processing ESD-numbers during processing an object module (i.e. only till the next END card)	1200 bytes
		Will be overlaid by status-table when calling $LIBSTAT	13 bytes
30K +176 bytes	CDENT1	Workarea (called control dictionary) containing information about all phases and ESD-items. Size will vary, as big as necessary 20 bytes for each ESD-item 40 bytes for each phase	?
30K+?	CTLDAD	Workarea (called stowtable) containing information about all phases as interface table for $MAINDIR or $MAINDIF. Size will vary, max. 2K 12 bytes for control-information 30 bytes for each phase	?
30K+?	WRKMAIN	Workarea used by $MAINDIR or $MAINDIF (approx. 24K, for exact value consult $MAINDIR or $MAINDIF)	24K
54K+?		END OF USED STORAGE	

When calling $LIBSTAT overlaying at

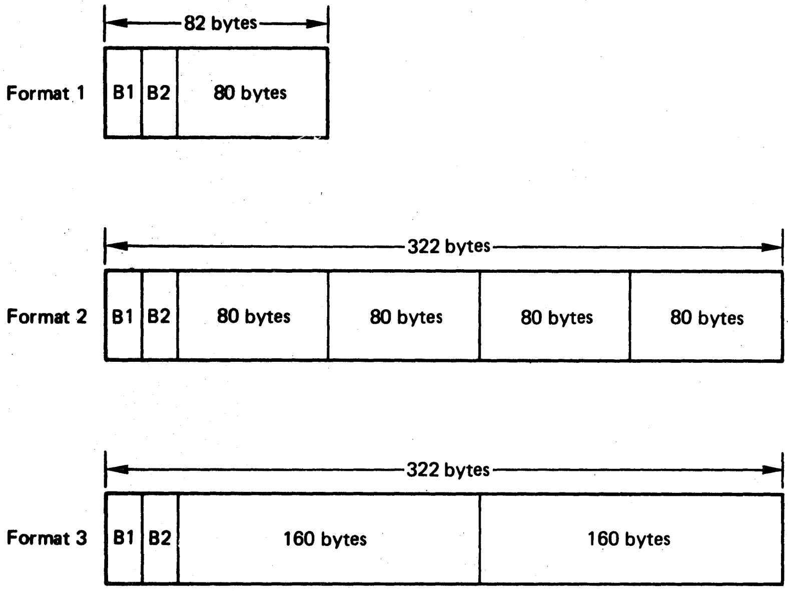
29K	LTMINE	Workarea (called status-table) as interface for $LIBSTAT	13 bytes
19K+ 16 bytes		Workarea used by $LIBSTAT (approx. 4K, for exact value consult $LIBSTAT)	4K

PROGRAM ORGANIZATION

LIBRARY FORMATS

SYSLNK AND RELOCATABLE LIBRARIES

The Linkage Editor reads data from SYSLNK in three different formats (format 1, 2 and 3) and from the Relocatable Library in a unique format (format 3).



B1 (1 byte) – Number of records per block (either 1, 2 or 4)
B2 (1 byte) – Record length (either 80 or 160)

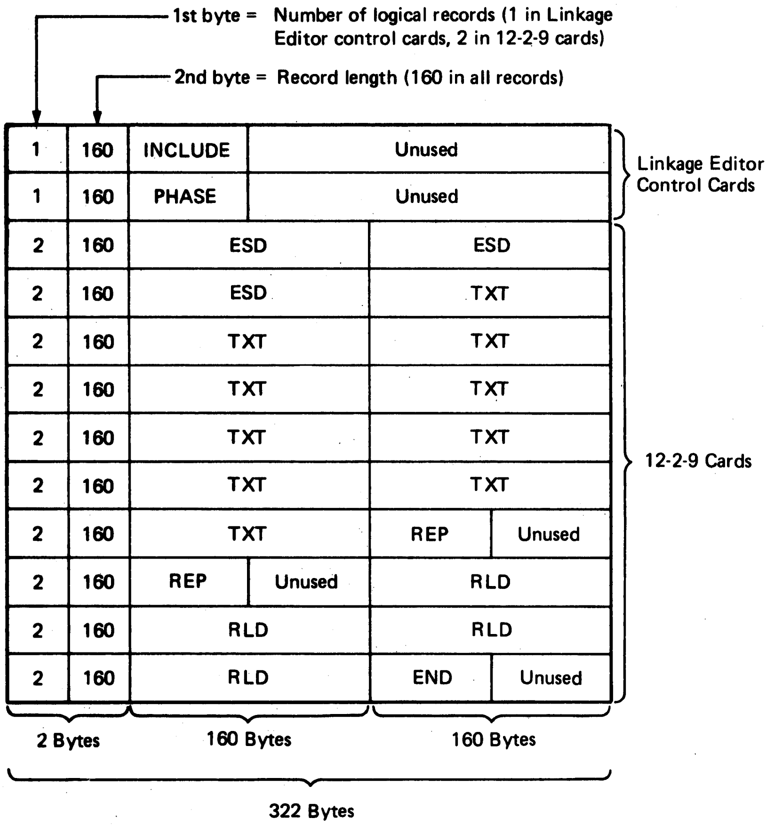
Figure 3. Block Format on SYSLNK and Relocatable Library

Format 3 is the only one accepted from the Relocatable Library. (Refer to the DOS/VS Librarian Logic for a detailed description.)

The Linkage Editor recognizes different item types (ESD, TXT, RLD, REP, END). Only one item type can be contained in each individual record.

The Linkage Editor control statements ACTION, INCLUDE, ENTRY and PHASE each occupy a complete block, i.e. only one record per block will be accepted.

For a private relocatable library, Figure 4 shows an example of a logical block consisting of a set of physical blocks.


1st byte = Number of logical records (1 in Linkage Editor control cards, 2 in 12-2-9 cards)
2nd byte = Record length (160 in all records)
Linkage Editor Control Cards
12-2-9 Cards
2 Bytes
160 Bytes
160 Bytes
322 Bytes

1	160	INCLUDE	Unused		
1	160	PHASE	Unused		
2	160	ESD		ESD	
2	160	ESD		TXT	
2	160	TXT		TXT	
2	160	TXT		TXT	
2	160	TXT		TXT	
2	160	TXT		TXT	
2	160	TXT		REP	Unused
2	160	REP	Unused	RLD	
2	160	RLD		RLD	
2	160	RLD		END	Unused

Figure 4. Example of a logical block on SYSRLB

Physical Description of SYSLNK and Relocatable Libraries

For CKD devices, the physical record is identical with the logical block. For FBA devices, this is not true. For the FBA records of SYSLNK, the SAM format is used. The control interval is 1K by default. The user may change this value by specifying a different size in the EXTENT card.

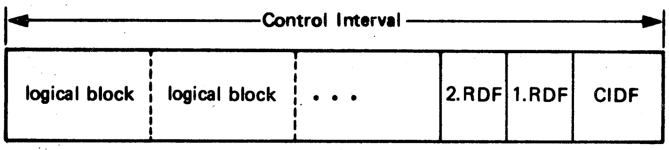


Figure 5. SAM-Format for SYSLNK on FBA

CIDF Control interval definition field (4 bytes; first two bytes containing address of free space, last two bytes containing length of free space)

RDF Record definition field (3 bytes)

1. RDF: 1. Byte: bit 1=0 only 1 block
   bit 1=1 (X'40') more than one block in this C.I.
   last two bytes: length of block

2. RDF: last two bytes: number of blocks in this C.I.

For the relocatable library on an FBA device, no special format is used. The first logical block of a member starts on a physical block boundary. The next logical blocks are written sequentially, regardless of 'physical block' boundaries ("spanned records", i.e. part of a logical may be in one physical block and the rest of it in the next physical block).

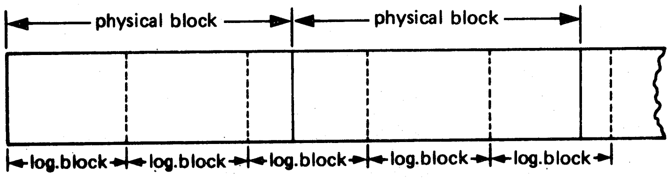


Figure 6. Spanned records for SYSRLB on FBA

CORE IMAGE LIBRARY

The logical record of the core image library has a size of 1024 bytes. It contains executable code and at the end of the phase some RLD information, whenever the phase is relocatable. For CKD devices, the records are unblocked and the physical record is the same as the logical record. For FBA devices with a blocksize of 512 bytes, the unit-of-transfer will always be two physical blocks = 1024 = one logical record.

WORKFILE SYS001

The logical record for the workfile SYS001 has a size of 240 bytes. In the first 16 bytes, it contains the standard information of a RLD card, in the rest, it contains the variable information of four RLD cards. For CKD devices, the records are unblocked and the physical record is the same as the logical record. For FBA devices, two logical records are blocked together in one physical block.

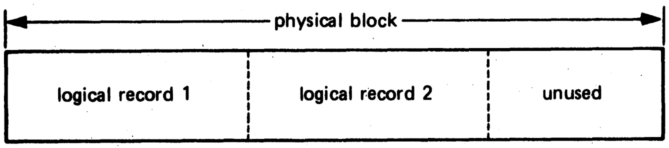


Figure 7. Blocked records for SYS001 on FBA

CONTROL FLOW

The linkage editor is a single phase program divided into ten CSECTs. The CSECT names and their functions are the following:

IJBINL Entry point for calling program, used for initialization and processing ACTION cards (afterwards overlaid by I/O-areas), (Chart 01).

IJBLNK Contains subroutines and constants, besides subroutine ALNKOF, which reads input and gives control to the appropriate CSECT (Chart 02). See also the chapter on subroutines.

IJBFIN Reads input, if SYSLNK or a Relocatable Library is on FBA (Chart 03).

IJBESD Processes ESD-cards (Charts 04, 05).

IJBOTH Processes TXT, REP, RLD, END-cards (Chart 06).

IJBSCN Processes INCLUDE, PHASE, ENTRY cards (Chart 07).

IJBCTL Post-processes PHASE, ENTRY cards (Chart 08).

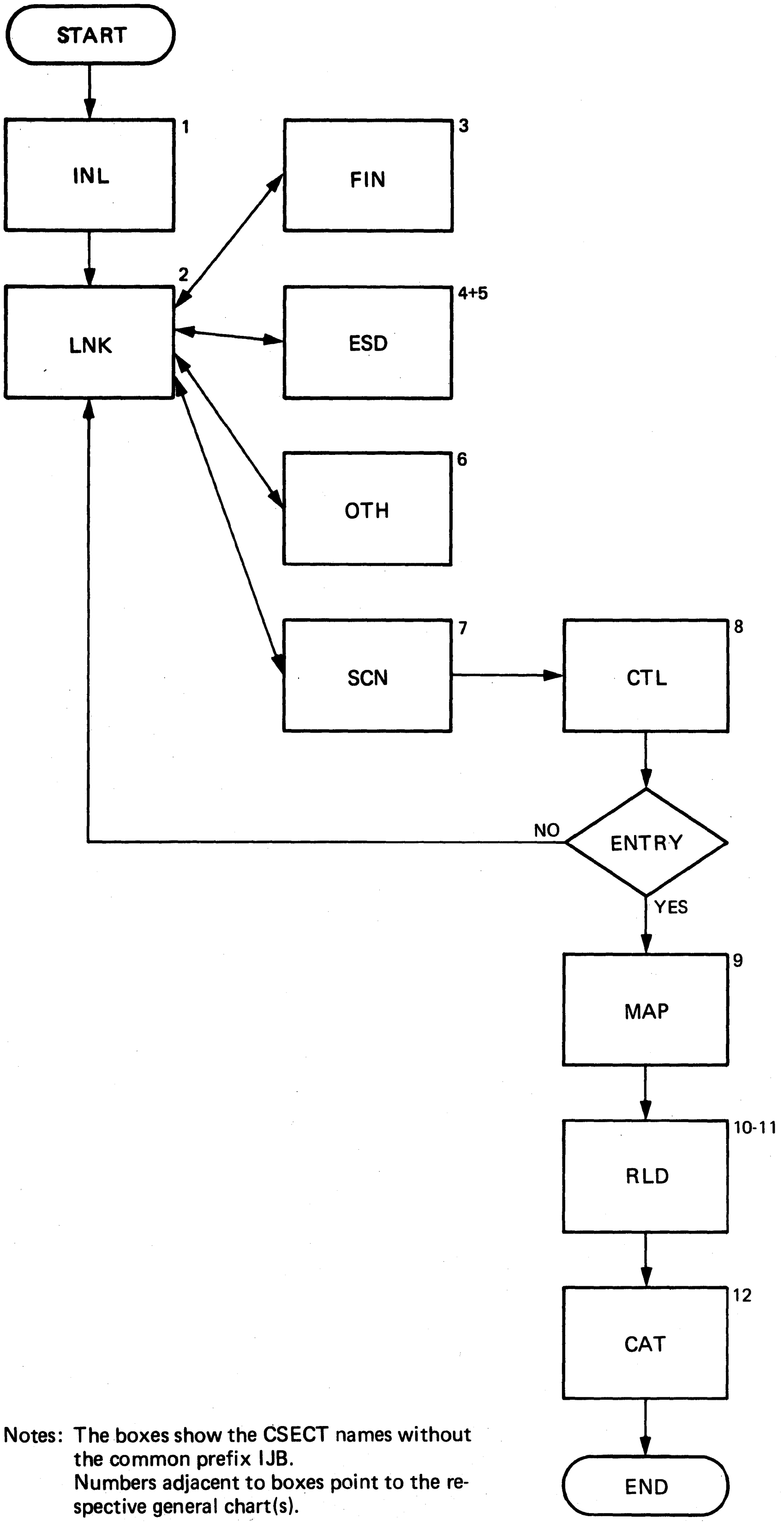
After ENTRY-card:

IJBMAP Prints linkage editor map (Chart 09) (Figure 19).

IJBRLD Processes RLD-items read from SYS001 (written by IJBOTH) (Charts 10, 11).

IJBCAT Updates directory of Core Image Library (calling $MAINDIR or MAINDIF). Prints status report (calling $LIBSTAT), returns to caller (Chart 12).

Figure 8 shows how these CSECTS are connected in the program.



Notes: The boxes show the CSECT names without the common prefix IJB.
Numbers adjacent to boxes point to the respective general chart(s).

Figure 8. Linkage Editor Control Flow

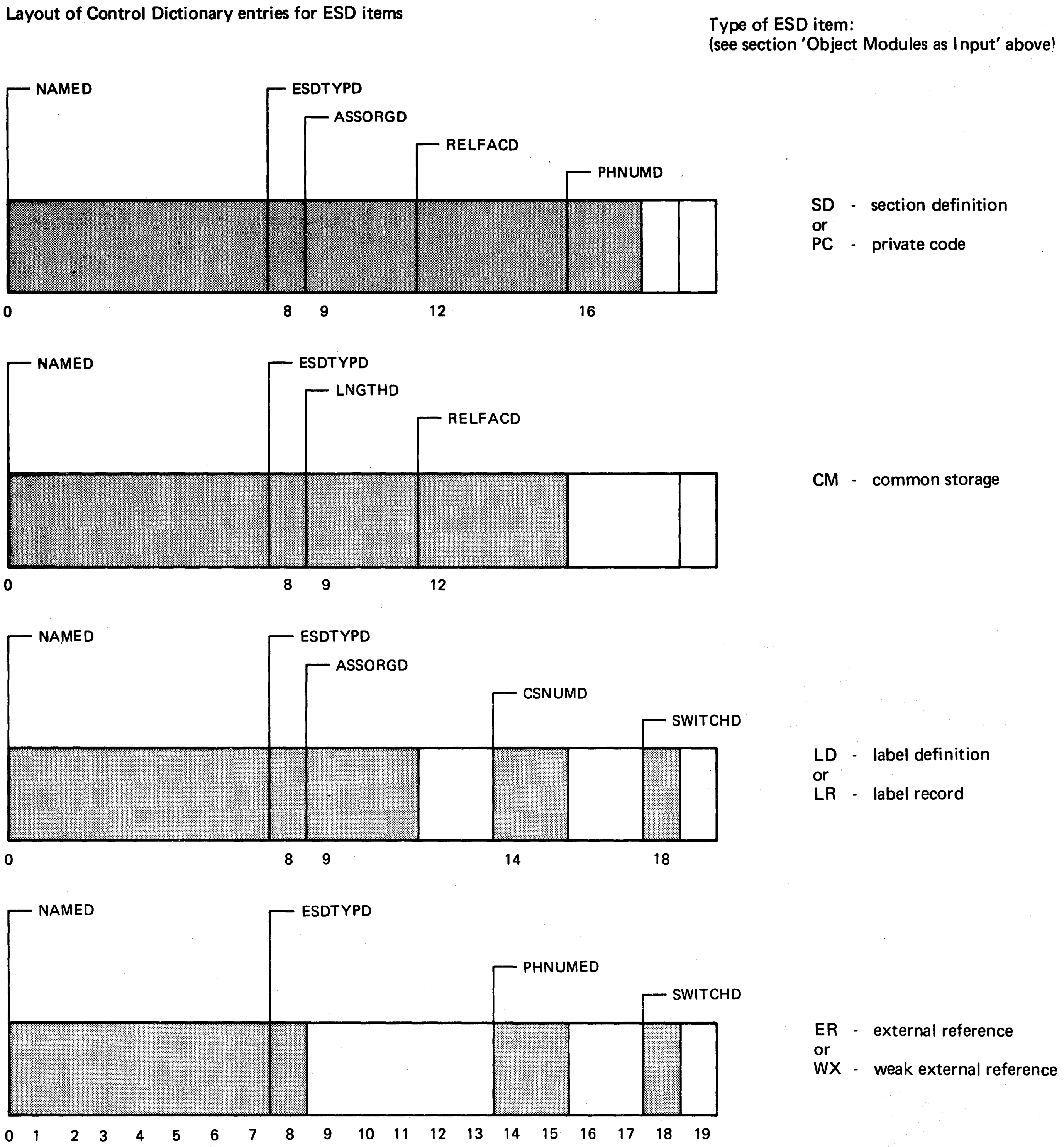
CONTROL DICTIONARY (CD)

The control dictionary is an internal linkage editor table used to store phase and external symbol dictionary (ESD) information. All information necessary for relocation is contained in the control dictionary.

The control dictionary is composed of a variable number of 20- and 40-byte entries. Each phase entry is 40 bytes long and is handled as if it consisted of two 20-byte entries by the routines scanning the control dictionary (except for when it is scanned for a section definition entry). All other types of entries are 20 bytes long.

The control dictionary starts on the first fullword boundary after the linkage table. Location CDENTI contains the address of the first entry. Location CTLDAD contains the address of the last entry in the control dictionary. Refer to Figure 9 for the format of the control dictionary entries.

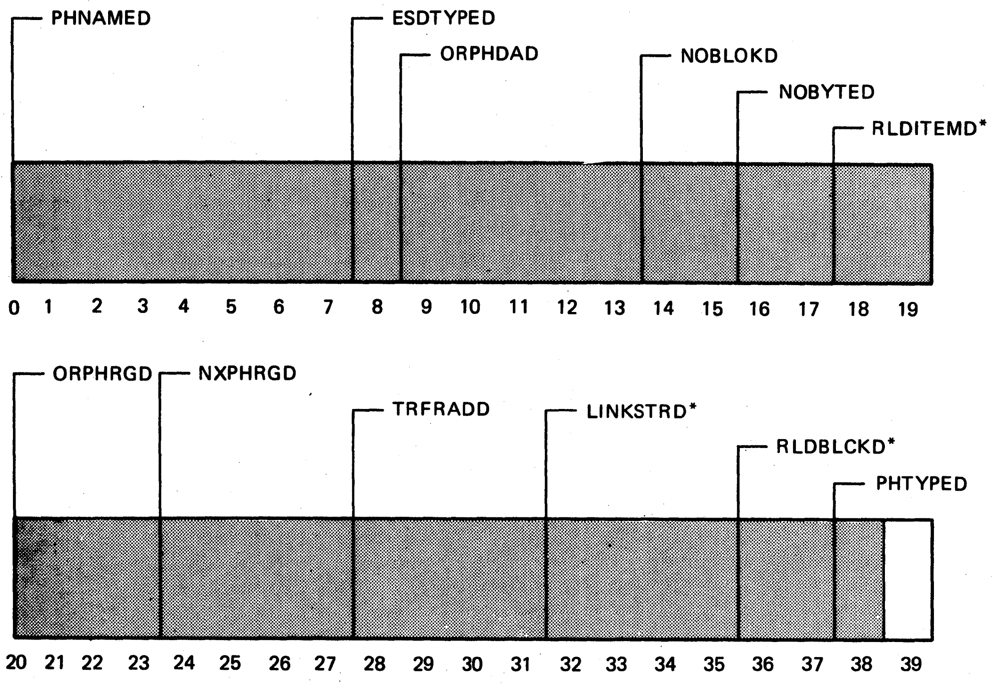
FORMAT OF CD ENTRIES



Byte 19 is reserved for future use.

Figure 9. Format of Control Dictionary Entries (Part 1 of 2)

Layout of Control Dictionary entries for a phase



Byte 39 is reserved for future use.
*Only used for relocatable phases. Otherwise the content is zero.

Figure 9. Format of Control Dictionary Entries (Part 2 of 2)

USE OF CD ENTRY FIELDS

Each control dictionary entry is first built in a fixed location in IJBLNK and is called "current entry". The current phase control dictionary entry starts at label CPHENT and the current ESD entry at label CESDENT. The current ESD entry is added to the control dictionary at label ELBINT in the ESD processor. The current phase entry is moved to the control dictionary at label MOVENTRY after the next phase card or the ENTRY card has been read.

In IJBMAP the location CPHENT is used to save a phase entry during a search for ESD entries belonging to the phase that is being processed. References to control dictionary information are sometimes to the current entry and sometimes to the control dictionary itself. Therefore, in the following description of the control dictionary fields, the name of the corresponding current field between parentheses is added to the control dictionary name of the field.

This CD Field	Contains
NAMED (NAME)	the name of the ESD item.
ESDTYPD (ESDTYP)	the type of ESD item. The representation of different ESD items is shown below.
ASSORGD (ASSORG)	the assembled origin of SD, PC, LD, or LR.
LNGTHD (LNGTH)	the length of the CM.
RELFACD (RELFAC)	• in the case of an SD or PC the relocation factor. The relocation factor is calculated by subtracting the assembled origin from the next possible phase origin NXPHRGD (NXPHRG). • in the case of a CM the absolute address of the next start of the CM.
CSNUMD (CSNUM)	a pointer to the SD or CM

containing the entry symbol of the LD or LR. If the SD or CM pointed to by an LD or LR has already been processed the LD or LR is called "assigned". If not, LD or LR is called "unassigned". In the case of an unassigned LD or LR this field contains the ESID number of the SD or CM as a pointer. In the case of an assigned LD or LR this field contains the control dictionary number of the SD or CM.

ESD Item	Representation
Section Definition (SD)	X'00'
Private Code (PC)	X'04'
Common (CM)	X'05'
Label Definition (LD)	X'01'
Label Reference (LR)	X'03'
External Reference (ER)	X'02'
Weak External (WX)	X'02'

PHNUMED (PHNUME) the phase number of the phase in which the ER or WX was encountered.

PHNUMD (PHNUM) the phase number of the phase containing the SD or PC.

SWITCHD (CSWITCH)

Bit	On	Off
0-4	Unused	
5*	No AUTOLINK necessary	AUTOLINK necessary
6	ER is a weak external	ER is a normal external reference
7	LD/LR unassigned	LD/LR assigned

*This bit is always set if bit 6 is on and also after AUTOLINK was not successful.

PHNAMED (PHNAMEC) the phase name.

ESDTYPED (ESDTYPC) X'07' to classify the control dictionary entry as a phase entry

ORPHDAD (ORPHDA) the disk address of the first text block of the phase in the format CCHHR for CKD devices. For FBA devices, the first four bytes contain the physical block number (relative to the device, not to the library); the fifth byte is unused.

NOBLOKD (NOBLOK) the number of text blocks.

NOBYTED (NOBYTE) the number of bytes in the last text block.

RLDITEMD (RLDITEMS) the number of RLD items.

ORPHRGD (ORPHRG) the load address of the phase. The first byte must be X'00' (see note below).

NXPHRGD (NXPHRG) the highest phase address. This field is initialized with the contents of ORPHRG. Every time an SD or PC is encountered the field NXPHRGD is aligned on a double word boundary and the length of the control section is added to it.

TRFRADD (TRFRAD) the transfer address of the phase. The first byte must be X'00' (see note below).

LINKSTRD (LINKSTRT) the starting address of the partition into which the program is link edited.

RLDBLCKD (RLDBLCKS) the number of extra RLD blocks. Extra RLD blocks must be added if, after the text of a relocatable phase, RLD information is stored for use by the relocating loader and the remaining space in the last text block is not sufficient.

PHTYPED (PHTYPE) an indication as to the type of phase.

Type of Phase	Representation
Self relocating	X'80'
Relocatable	X'40'
SVA eligible	X'20'
Not relocatable	X'00'

Note: Most of the routines scanning the control dictionary handle it as if it consisted of fixed length (20-byte) entries. The value X'00' in the first bytes of the ORPHRGD and TRFRADD fields, prevents the second half of a phase entry to be interpreted as an ESD entry since the displacements of ORPHRGD and TRFRACD correspond to those of NAMED and ESDTYPC. Because of the type mask X'00' for an SD, an exception has been made for routines searching the control dictionary for an SD.

LINKAGE TABLE

The linkage table is an internal linkage editor mechanism used to link the ESIC number supplied by the language translator output to the corresponding control dictionary number that belongs to a control dictionary entry.

The linkage table is composed of a variable number of fixed 3-byte entries up to a maximum of 400. It is built separately for each object module. When an END card is processed, signalling the end of a module, the table is reset to zeros. Location LTMINE contains the address of the first item in the linkage table minus 3 bytes. LNKTAD contains the address of the last item in the linkage table plus 3 bytes.

Linkage Table

Control Dictionary Number	ESD Type
2 Bytes	1 Eyte

USE OF THE LINKAGE TABLE ANC CONTROL DICTIONARY

The linkage table is designed to associate text and RLD information with the proper relocation attribute from the control dictionary. The following steps are taken in processing text:

1. Get the ESID number and calculate the linkage table entry.

2. Go to the linkage table.

3. Extract the control dictionary number field of the linkage table, and calculate the control dictionary entry location.

4. Go to the control dictionary entry.

5. Extract the relocation factor.

6. Add the relocation factor to the assembled origin of the text to be loaded.

7. Substitute the result of the calculation in step 6 (the load origin) for the language translator supplied assembled origin (for the text).

8. Calculate the block of the core image library to which this text belongs (next available block).

9. Get the proper core image block.

10. Put the text into the core image block.

Note: If a TXT card on P-pointer points to a negative control dictionary number, that control section is skipped. If the R-pointer points to a negative control dictionary number, that control section is needed (CSECT is not in this phase in real storage).

THE AUTOLINK FEATURE

This feature tries to locate a module in the private (if assigned) and system relocatable libraries for any unresolved ERs found in the preceding phase. The signal that indicates a phase has finished processing is either a new phase card or an ENTRY card. When the signal is detected, autolink is attempted unless the feature has been suppressed by a NOAUTO phase card, action card option, or by WXTRN.

Examples of Autolink with LIOCS

Whenever a DTF macro is expanded during a language translator run, an ER is generated with a label corresponding to a label of a LIOCS module. The label of the ER is used as the search argument in autolink. The autolink processing searches first the private (if assigned), then the system relocatable directories for the corresponding label. The directory entry contains the disk address of the module in the relocatable libraries. The module is the macro expansion, and is then treated as an include statement.

Linkage Editor Fundamental Calculations: For the examples in this presentation:

- The symbol A/O represents the assembled origin.

- The symbol R/F represents the relocation factor.

- The symbol L/O represents the load origin.

- The symbol P/O represents the phase origin.

Example 1: The A/O provided by the language translator is added to an R/F determined by the phase origin information. If the phase is not relocated when it is loaded into main storage the result, the L/O, is the main storage address that is the physical location of this text, RLD item, or control section.

A/O + R/F = L/O

Example 2: The assembled origin of the CSECT being processed is subtracted from the address that is the next possible phase origin. This results in the relocation factor for that control section.

P/O - A/O = R/F

Example 3: The address of the next available control dictionary entry is calculated by adding the length of the last entry to the address of the last entry.

Example 4: The current linkage table entry plus 3 equals the next linkage table entry.

SUBROUTINES IN IJBLNK

The first CSECT (IJBLNK) of the linkage editor program contains most of the subroutines used by other linkage editor CSECTs. After processing any of these subroutines, control is returned to the calling routine if not indicated otherwise. A list containing the subroutine names of this CSECT, the main entry points, descriptions of the routines' functions, and the appropriate flowcharts is shown below:

Sub-routine	Entry point	Function	Chart
RDS000	RDS000	Reads input from SYSLNK or the relocatable library.	AA
LTESID	LTESID	Note: Input to this routine is an ESID number supplied by the language translators. Inspects the control dictionary by taking the following actions: • If the control dictionary number is zero, the ESID number has not yet been processed. The routine then returns to the address in the link register. • If the control dictionary number is negative, the ESD item is bypassed and the routine returns to link register plus 4. Addresses of the linkage table entry and the control dictionary number are supplied. • If the control dictionary number is positive, the routine returns to register plus 8. Relocation factor (for SD/PC), control dictionary number and address of the control dictionary are supplied.	AB
SRCHCD	SRCHCD	Searches the control dictionary for a matching label.	AC
	SRPCOD	Entry at SRPCOD continues the search after a matching label has been found.	
CNVHEX	CNVHEX	Converts EBCDIC input to hexadecimal output.	AC
PRINT	PRINT	Prints messages and map onto SYSLST.	AD
LOGMSG	LOGMSG	Prints error message onto SYSLOG.	AD
PRTLST	PRTLST	Prepares for printing the linkage editor diagnostics of input.	AD
SPACE1	SPACE1	Spaces one line on SYSLST.	AD
AD1DSK	AD1DSK	Updates the disk address to the next record.	AE
UPDSKAD	UPDSKAD	Updates the disk address to the first record on the next track.	
XTPHNO	XTPHNO	Extracts the phase number from SD, PC, LD, or LR control dictionary entries.	AF
	XTPHGT	Entry XTPHGT is used if the entry is known to be a SD or PC.	

ABTERR	ABTERR	Gives control to IJBRLD for abnormal termination error handling.	AF
CDSIZE	CDSIZE	Checks for control dictionary overflow.	AF
WRITE	WRITE READC1	Reads or writes core image blocks.	AG
	DISKRDWR DISKIO	Executes the I/O by means of EXCP.	
	FDISKIO	Entry if I/O for FBA.	
ALNKPR	ALNKPR	Initializes for the scanning of the relocatable directory for AUTOLINK. Extracts unresolved ERs from the control dictionary in collating sequence. Gives control to the INCLUDE processor and, after the modules have been included, passes control to ALNKOF.	AG
ALNKOF	ALNKOF RDNEXT	Reads the input stream and diagnoses the type of card to pass control to the appropriate CSECT.	AH
	RDEXEC EXLOAD	Entry at RDEXEC skips reading of card.	
CANCL	CANCL	Cancel routine. If necessary, sets the new supervisor cataloged bit in COMREG off.	AH
ERROR	ERROR	Sets up to print non-termination error messages. If the calling routine sets the RETRN bit in ERRSW, the ERROR routine returns to the calling routine. If the RETRN bit is off, return is to RDNEXT or to ALNKPR if the error occurred during AUTOLINK.	AI
NOTCTL	NOTCTL	Converts input cards (containing 12-2-9 in the first column) to machine printable format.	AI
DERDAD	DERDAD	Provides a core image block containing a specified storage address for a work area. If the space in the work area is required for a next block, the current block is written back to the CIL.	AK

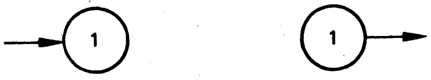
General Charts Conventions:

1. A unit of programming, routine, CSECT, or phase, is contained in one box like this:

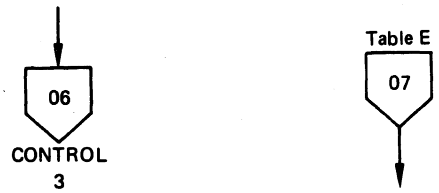
```
   xxx
 ┌──────────────────────────────────────┐
 | yyy                              zzz |
 |--------------------------------------|
 | 1.                                   |
 | 2.     step members                  |
 | 3.                                   |
 |                                      |
```

where: xxx marks the label and routine name
yyy says briefly what the routine does
zzz is the reference to the detail chart(s).
The step numbers are given from 1 to n within this routine only.

2. On-page connectors are such:

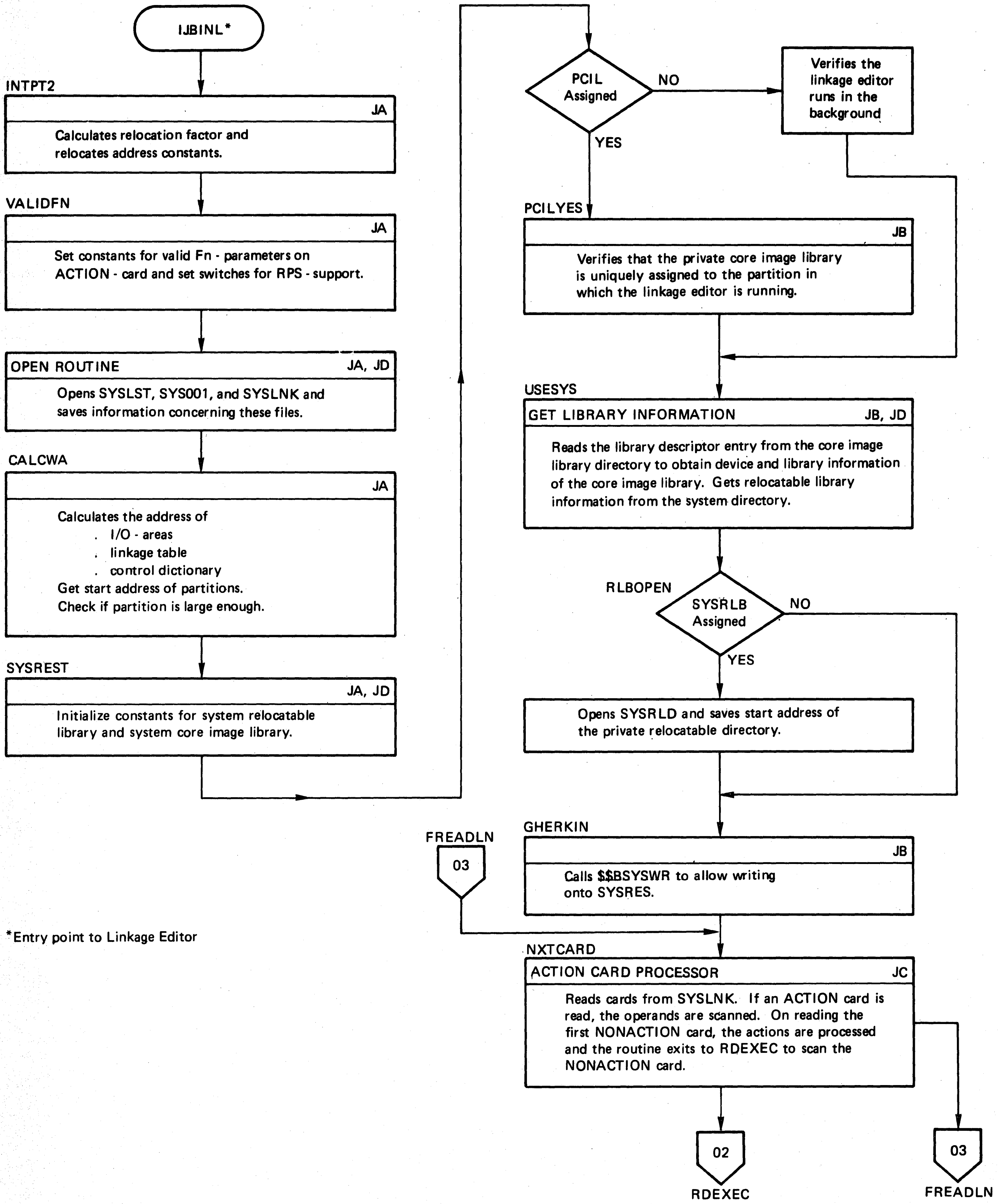


3. Off-page connectors are such:



where: the number in the frame marks the chart from which we come or to which we go,
the word above (incoming) or below (outgoing) marks the label (routine) on that chart,
the number under the word marks the step within the routine to which we go if it is not step 1.

Chart 01. IJBINL and FBAINL Initialization



Calculates relocation factor and relocates address constants.

Set constants for valid Fn - parameters on ACTION - card and set switches for RPS - support.

Opens SYSLST, SYS001, and SYSLNK and saves information concerning these files.

Calculates the address of
. I/O - areas
. linkage table
. control dictionary
Get start address of partitions.
Check if partition is large enough.

Initialize constants for system relocatable library and system core image library.

Verifies the linkage editor runs in the background

Verifies that the private core image library is uniquely assigned to the partition in which the linkage editor is running.

Reads the library descriptor entry from the core image library directory to obtain device and library information of the core image library. Gets relocatable library information from the system directory.

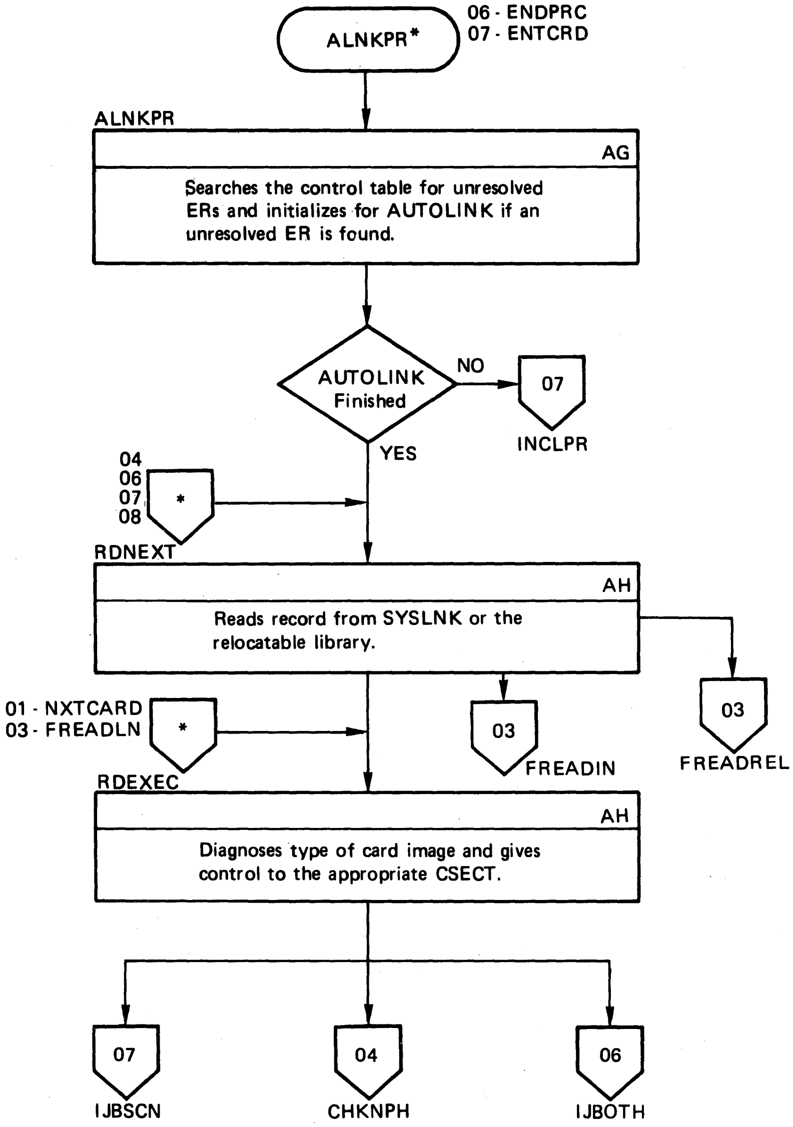
Opens SYSRLD and saves start address of the private relocatable directory.

Calls $$BSYSWR to allow writing onto SYSRES.

Reads cards from SYSLNK. If an ACTION card is read, the operands are scanned. On reading the first NONACTION card, the actions are processed and the routine exits to RDEXEC to scan the NONACTION card.

*Entry point to Linkage Editor

Chart 02. IJBLNK - Subroutines


ALNKPR*
06 - ENDPRC
07 - ENTCRD
ALNKPR
AG
Searches the control table for unresolved ERs and initializes for AUTOLINK if an unresolved ER is found.
AUTOLINK Finished
NO
07
INCLPR
YES
04
06
07
08
*
RDNEXT
AH
Reads record from SYSLNK or the relocatable library.
01 - NXTCARD
03 - FREADLN
*
03
FREADIN
03
FREADREL
RDEXEC
AH
Diagnoses type of card image and gives control to the appropriate CSECT.
07
IJBSCN
04
CHKNPH
06
IJBOTH

Note: The flowcharts AA-AF and AJ-AK describe the less important subroutines.

Chart 03. IJBFIN - Processing FBA device input

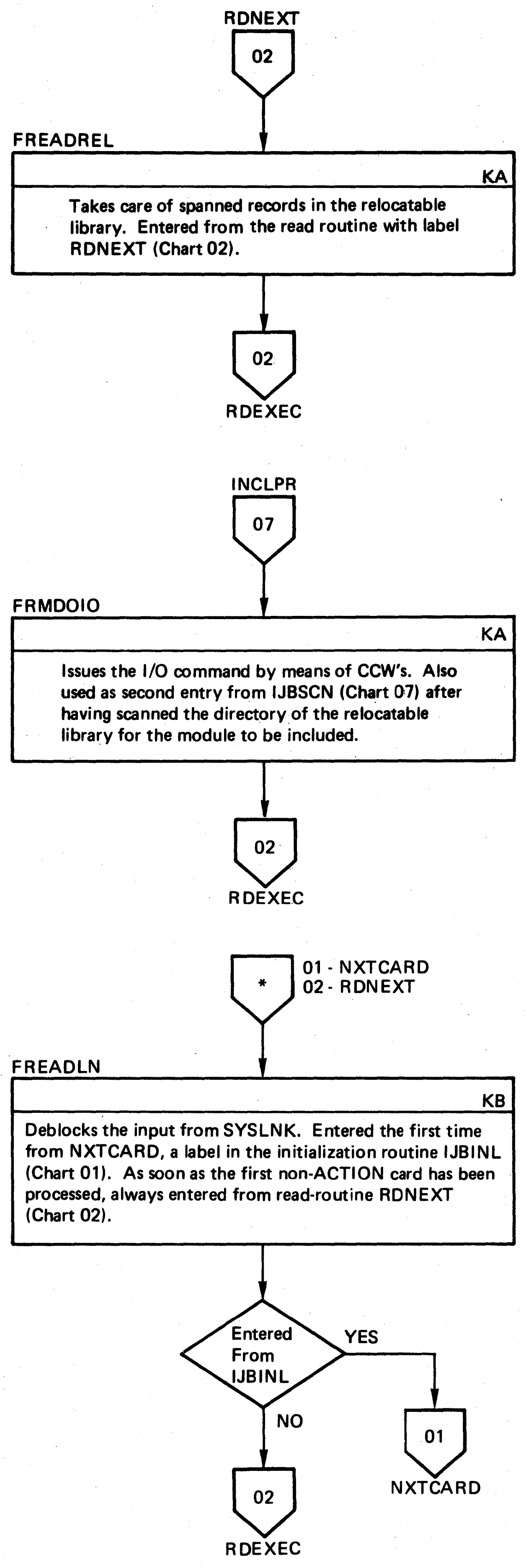

RDNEXT
02
FREADREL
KA
Takes care of spanned records in the relocatable library. Entered from the read routine with label RDNEXT (Chart 02).
02
RDEXEC
INCLPR
07
FRMDOIO
KA
Issues the I/O command by means of CCW's. Also used as second entry from IJBSCN (Chart 07) after having scanned the directory of the relocatable library for the module to be included.
02
RDEXEC
*
01 - NXTCARD
02 - RDNEXT
FREADLN
KB
Deblocks the input from SYSLNK. Entered the first time from NXTCARD, a label in the initialization routine IJBINL (Chart 01). As soon as the first non-ACTION card has been processed, always entered from read-routine RDNEXT (Chart 02).
Entered From IJBINL
YES
NO
01
NXTCARD
02
RDEXEC

Chart 04. IJBESD - ESD Processing

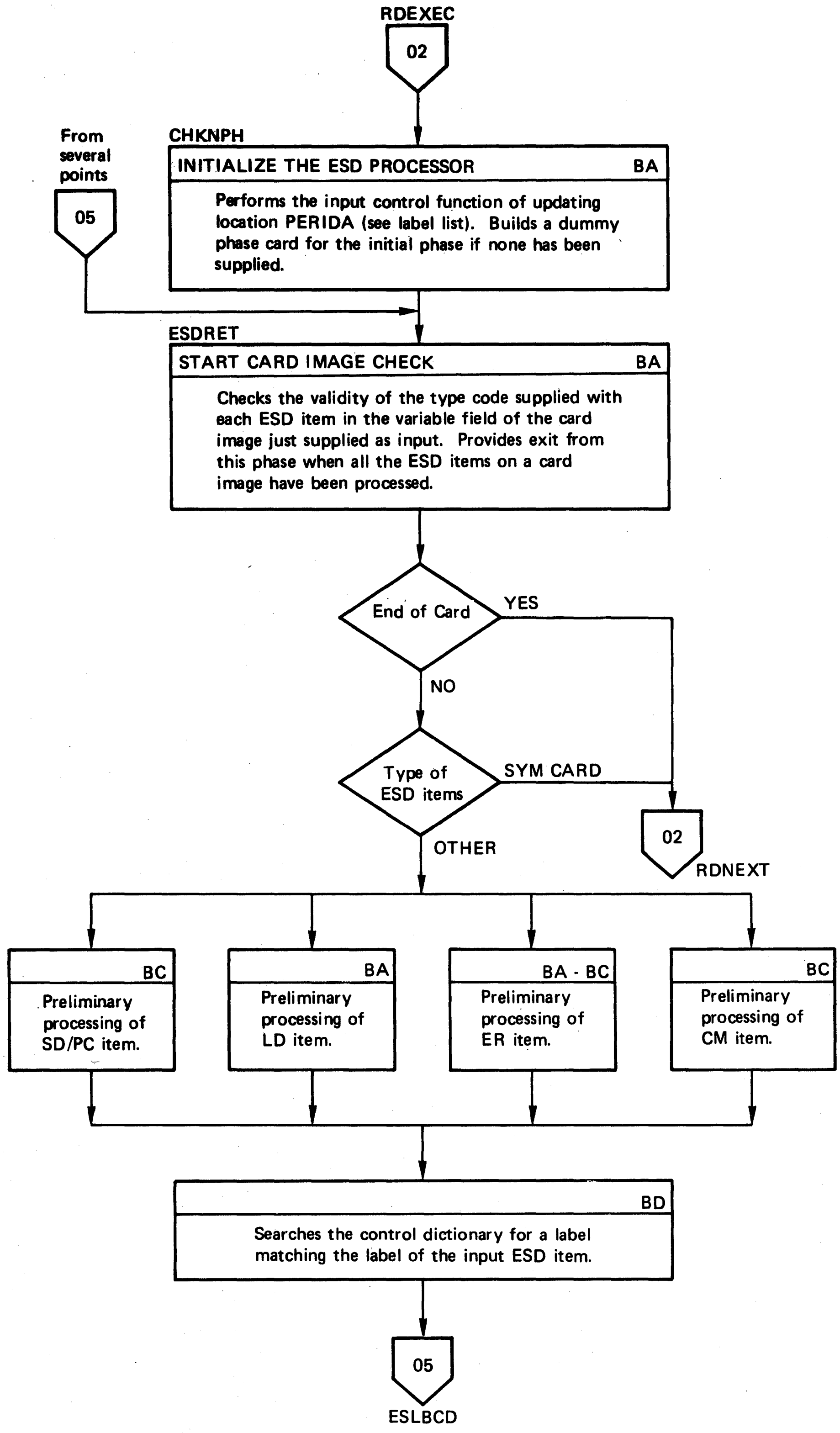

RDEXEC
02
From several points
05
CHKNPH
INITIALIZE THE ESD PROCESSOR
BA
Performs the input control function of updating location PERIDA (see label list). Builds a dummy phase card for the initial phase if none has been supplied.
ESDRET
START CARD IMAGE CHECK
BA
Checks the validity of the type code supplied with each ESD item in the variable field of the card image just supplied as input. Provides exit from this phase when all the ESD items on a card image have been processed.
End of Card
YES
NO
Type of ESD items
SYM CARD
02
RDNEXT
OTHER
BC
Preliminary processing of SD/PC item.
BA
Preliminary processing of LD item.
BA - BC
Preliminary processing of ER item.
BC
Preliminary processing of CM item.
BD
Searches the control dictionary for a label matching the label of the input ESD item.
05
ESLBCD

Chart 05. IJBESD - ESD Processing

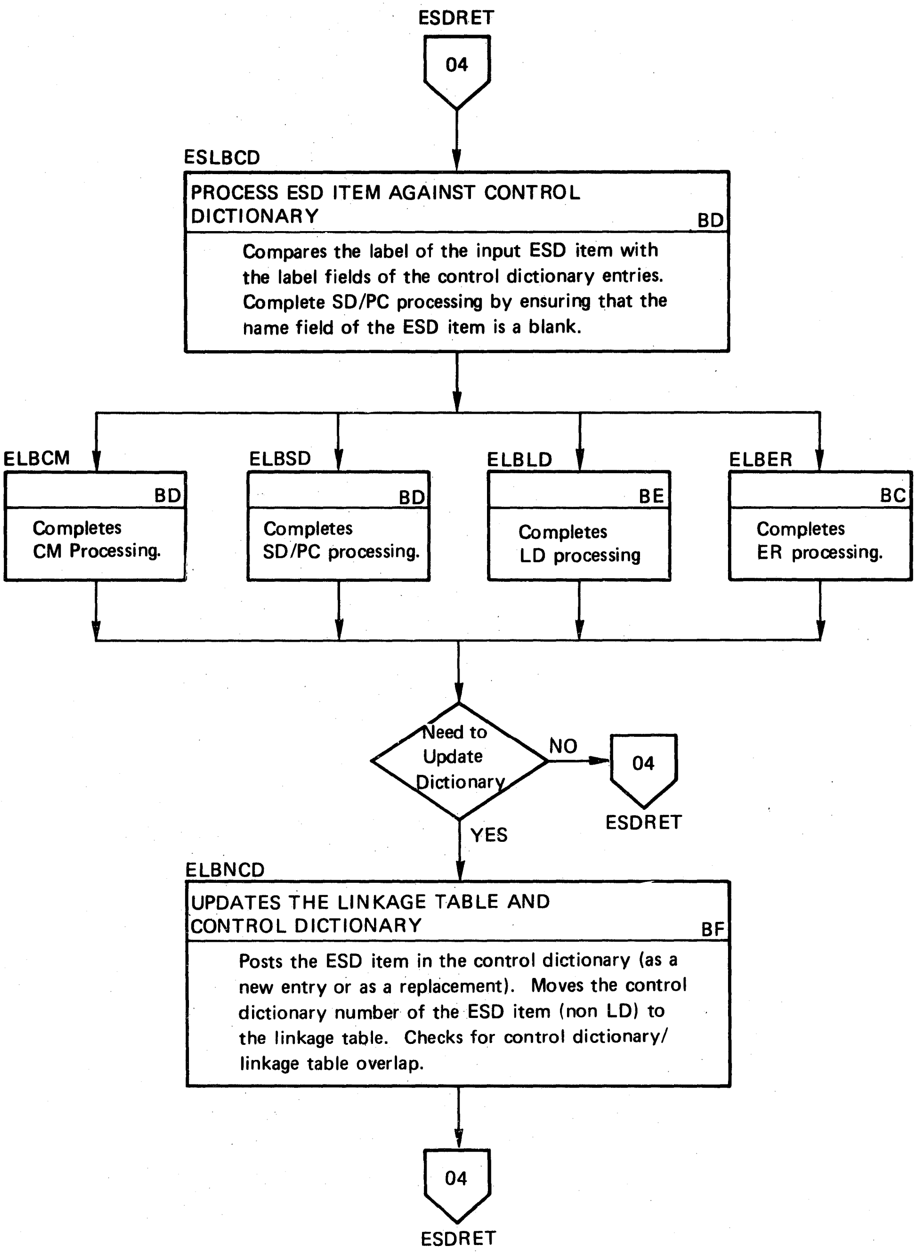

ESDRET
04
ESLBCD
PROCESS ESD ITEM AGAINST CONTROL DICTIONARY
BD
Compares the label of the input ESD item with the label fields of the control dictionary entries. Complete SD/PC processing by ensuring that the name field of the ESD item is a blank.
ELBCM
BD
Completes CM Processing.
ELBSD
BD
Completes SD/PC processing.
ELBLD
BE
Completes LD processing
ELBER
BC
Completes ER processing.
Need to Update Dictionary
NO
04
ESDRET
YES
ELBNCD
UPDATES THE LINKAGE TABLE AND CONTROL DICTIONARY
BF
Posts the ESD item in the control dictionary (as a new entry or as a replacement). Moves the control dictionary number of the ESD item (non LD) to the linkage table. Checks for control dictionary/ linkage table overlap.
04
ESDRET

Chart 06. IJBOTH - TXT, REP, RLD, and END Processing

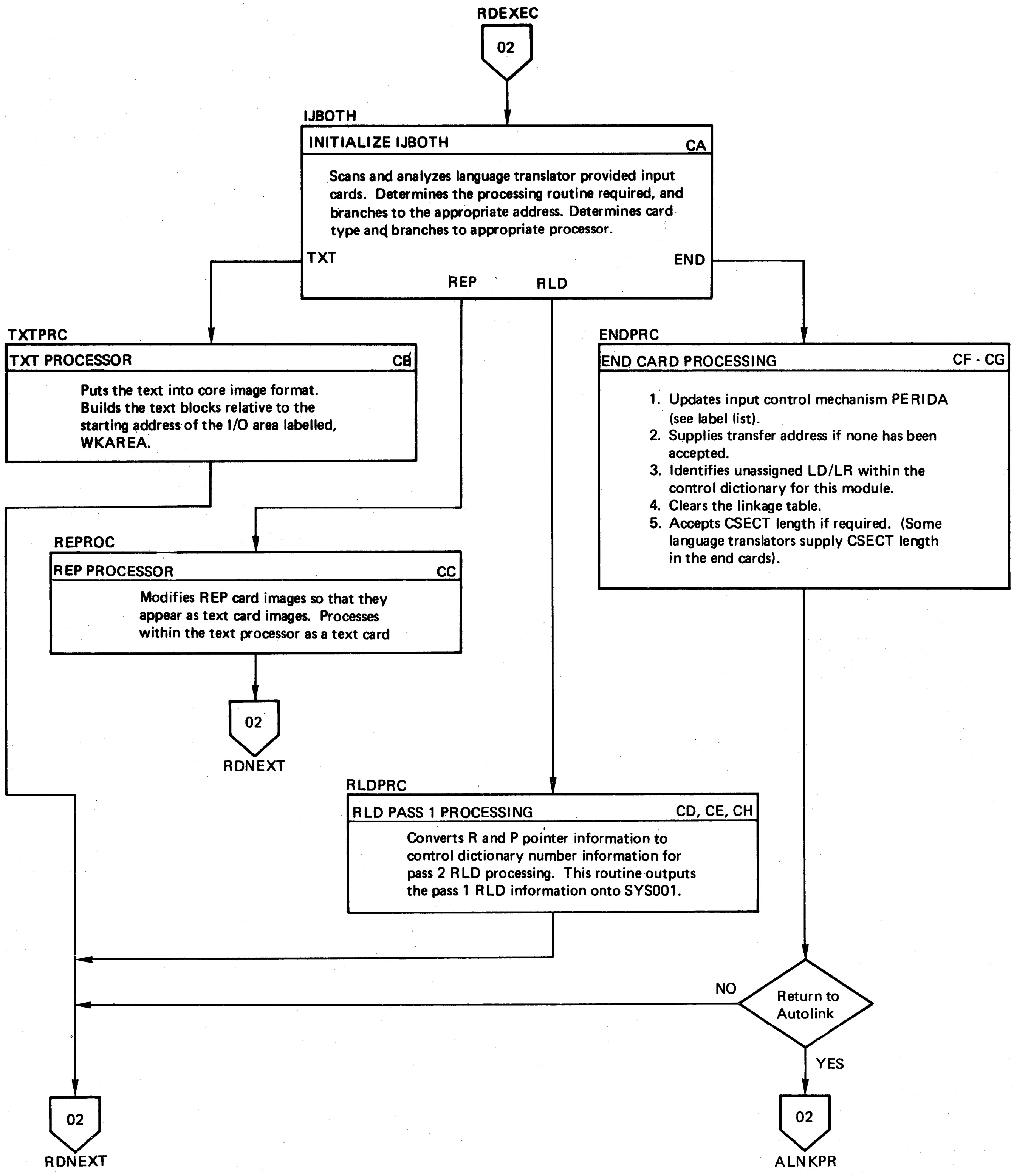
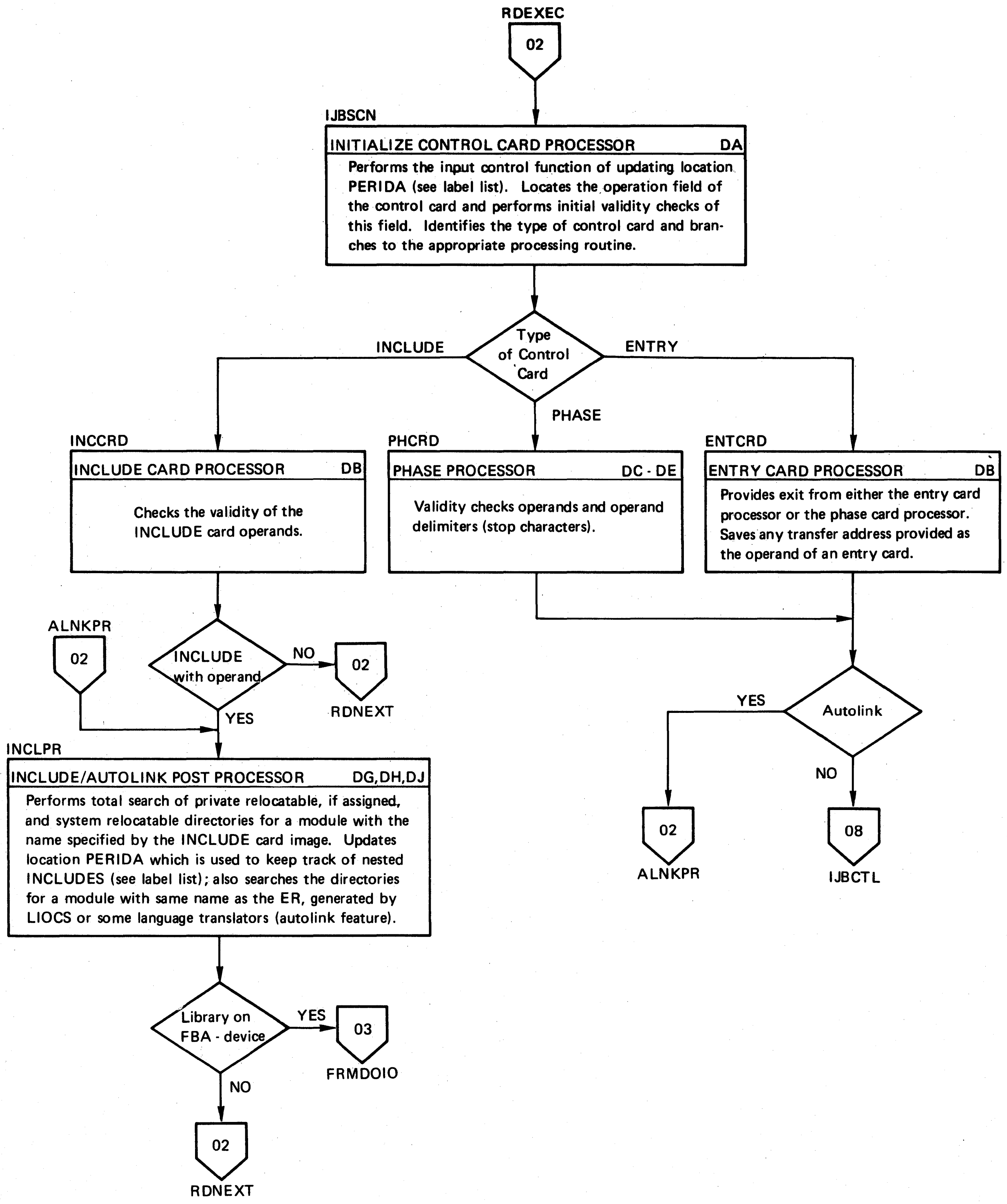

RDEXEC
02
IJBOTH
INITIALIZE IJBOTH
CA
Scans and analyzes language translator provided input cards. Determines the processing routine required, and branches to the appropriate address. Determines card type and branches to appropriate processor.
TXT
END
REP
RLD
TXTPRC
TXT PROCESSOR
CB
Puts the text into core image format. Builds the text blocks relative to the starting address of the I/O area labelled, WKAREA.
ENDPRC
END CARD PROCESSING
CF - CG
1. Updates input control mechanism PERIDA (see label list).
2. Supplies transfer address if none has been accepted.
3. Identifies unassigned LD/LR within the control dictionary for this module.
4. Clears the linkage table.
5. Accepts CSECT length if required. (Some language translators supply CSECT length in the end cards).
REPROC
REP PROCESSOR
CC
Modifies REP card images so that they appear as text card images. Processes within the text processor as a text card
02
RDNEXT
RLDPRC
RLD PASS 1 PROCESSING
CD, CE, CH
Converts R and P pointer information to control dictionary number information for pass 2 RLD processing. This routine outputs the pass 1 RLD information onto SYS001.
NO
Return to Autolink
YES
02
RDNEXT
02
ALNKPR

Chart 07. IJBSCN - Control Statement (INCLUDE, PHASE, and ENTRY) and Scan Processing



INITIALIZE CONTROL CARD PROCESSOR: Performs the input control function of updating location PERIDA (see label list). Locates the operation field of the control card and performs initial validity checks of this field. Identifies the type of control card and branches to the appropriate processing routine.

INCLUDE CARD PROCESSOR: Checks the validity of the INCLUDE card operands.

PHASE PROCESSOR: Validity checks operands and operand delimiters (stop characters).

ENTRY CARD PROCESSOR: Provides exit from either the entry card processor or the phase card processor. Saves any transfer address provided as the operand of an entry card.

INCLUDE/AUTOLINK POST PROCESSOR: Performs total search of private relocatable, if assigned, and system relocatable directories for a module with the name specified by the INCLUDE card image. Updates location PERIDA which is used to keep track of nested INCLUDES (see label list); also searches the directories for a module with same name as the ER, generated by LIOCS or some language translators (autolink feature).

Chart 08. IJBCTL - PHASE and ENTRY Statement Processing

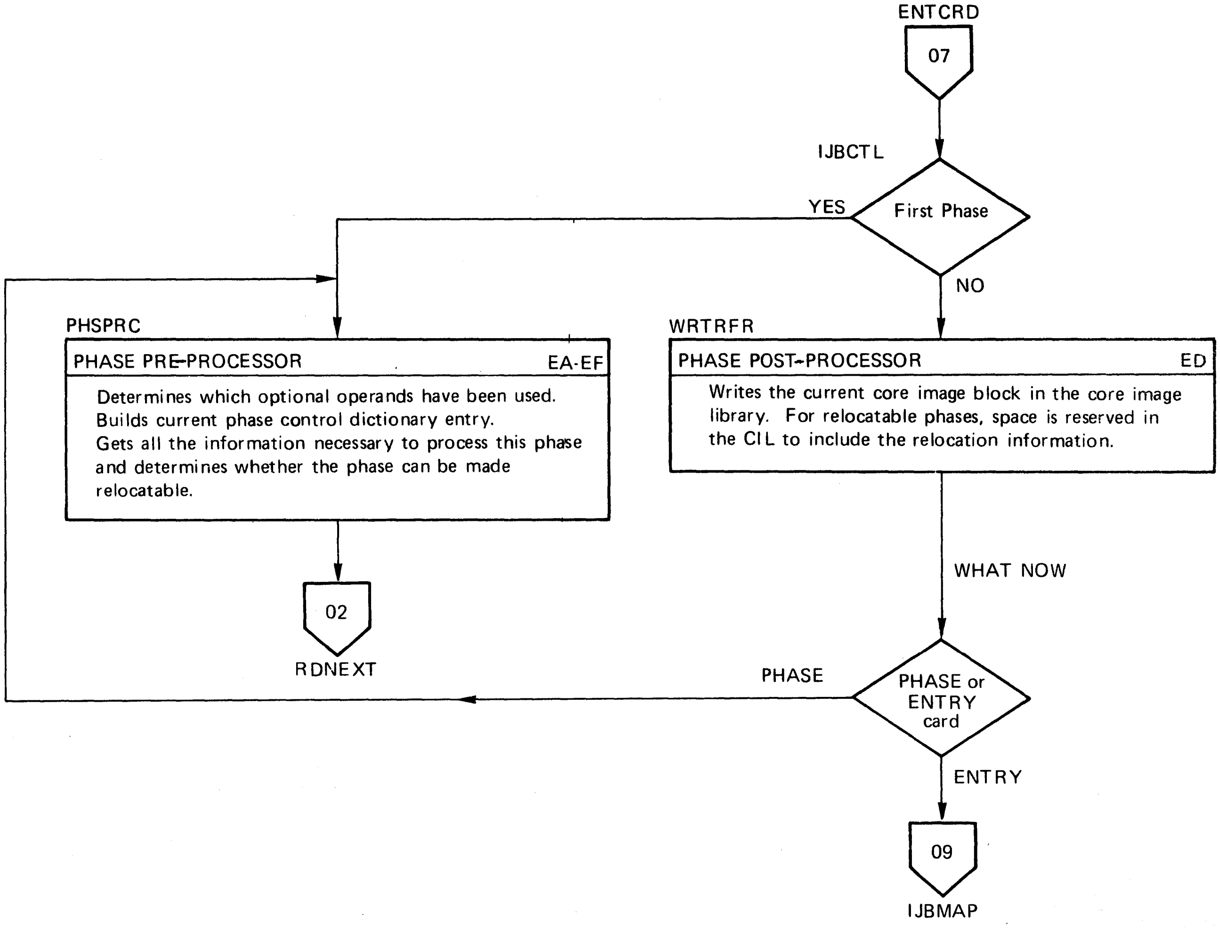

ENTCRD
07
IJBCTL
First Phase
YES
NO
PHSPRC
PHASE PRE-PROCESSOR
EA-EF
Determines which optional operands have been used. Builds current phase control dictionary entry. Gets all the information necessary to process this phase and determines whether the phase can be made relocatable.
WRTRFR
PHASE POST-PROCESSOR
ED
Writes the current core image block in the core image library. For relocatable phases, space is reserved in the CIL to include the relocation information.
02
RDNEXT
WHAT NOW
PHASE
PHASE or ENTRY card
ENTRY
09
IJBMAP

Chart 09. IJBMAP - Print Map

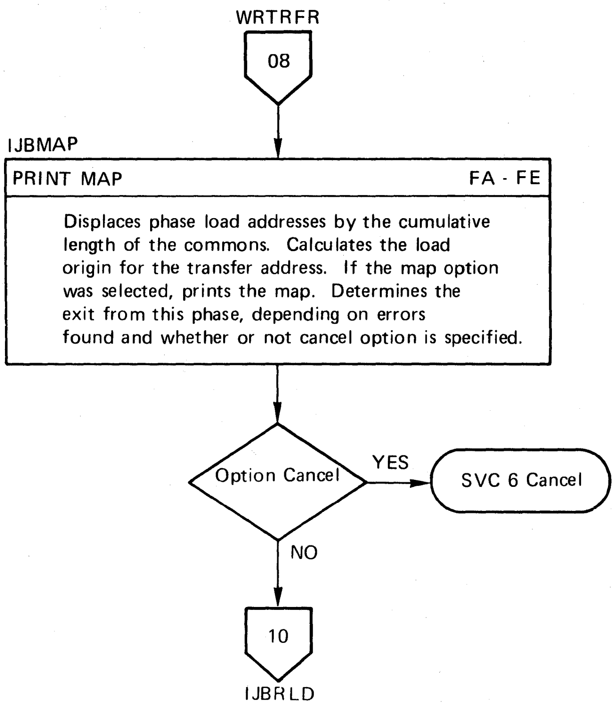
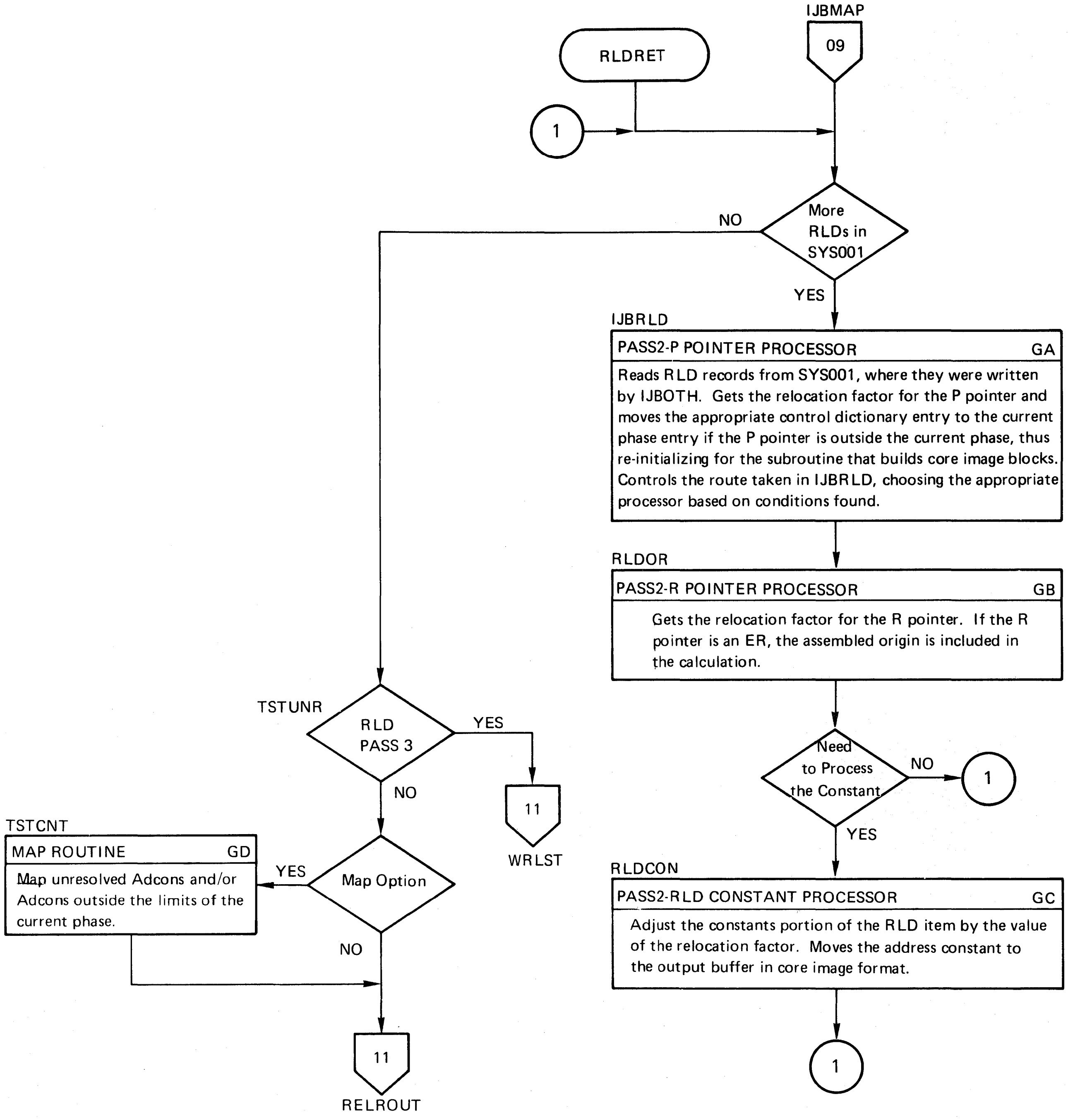

WRTRFR
08
IJBMAP
PRINT MAP
FA - FE
Displaces phase load addresses by the cumulative length of the commons. Calculates the load origin for the transfer address. If the map option was selected, prints the map. Determines the exit from this phase, depending on errors found and whether or not cancel option is specified.
Option Cancel
YES
SVC 6 Cancel
NO
10
IJBRLD

Chart 10. IJBRLD - RLD Processing



Reads RLD records from SYS001, where they were written by IJBOTH. Gets the relocation factor for the P pointer and moves the appropriate control dictionary entry to the current phase entry if the P pointer is outside the current phase, thus re-initializing for the subroutine that builds core image blocks. Controls the route taken in IJBRLD, choosing the appropriate processor based on conditions found.

Gets the relocation factor for the R pointer. If the R pointer is an ER, the assembled origin is included in the calculation.

Adjust the constants portion of the RLD item by the value of the relocation factor. Moves the address constant to the output buffer in core image format.

Map unresolved Adcons and/or Adcons outside the limits of the current phase.

Chart 11. IJBRLD - RLD Processing

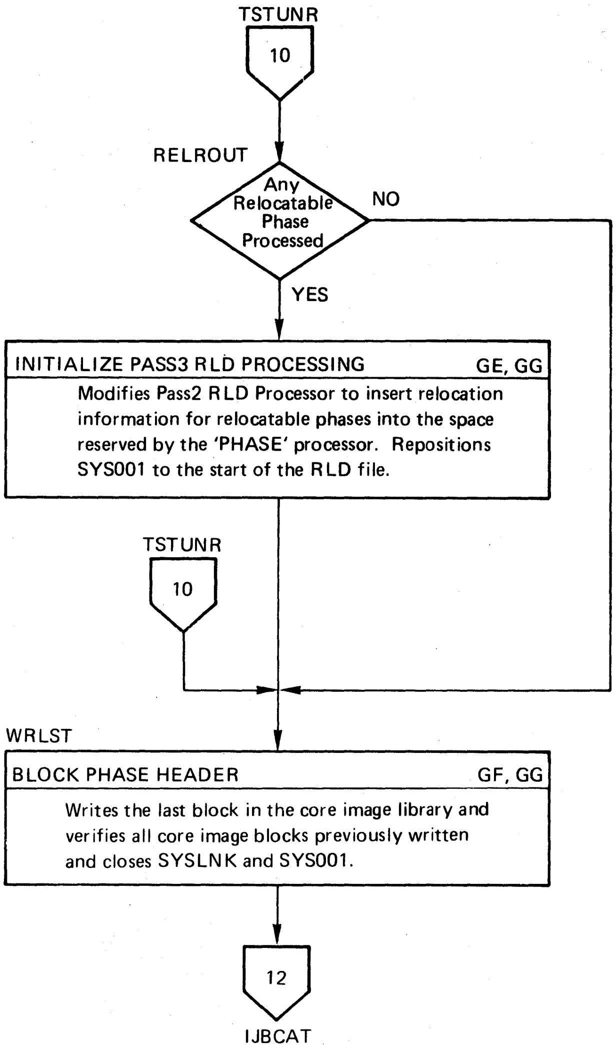
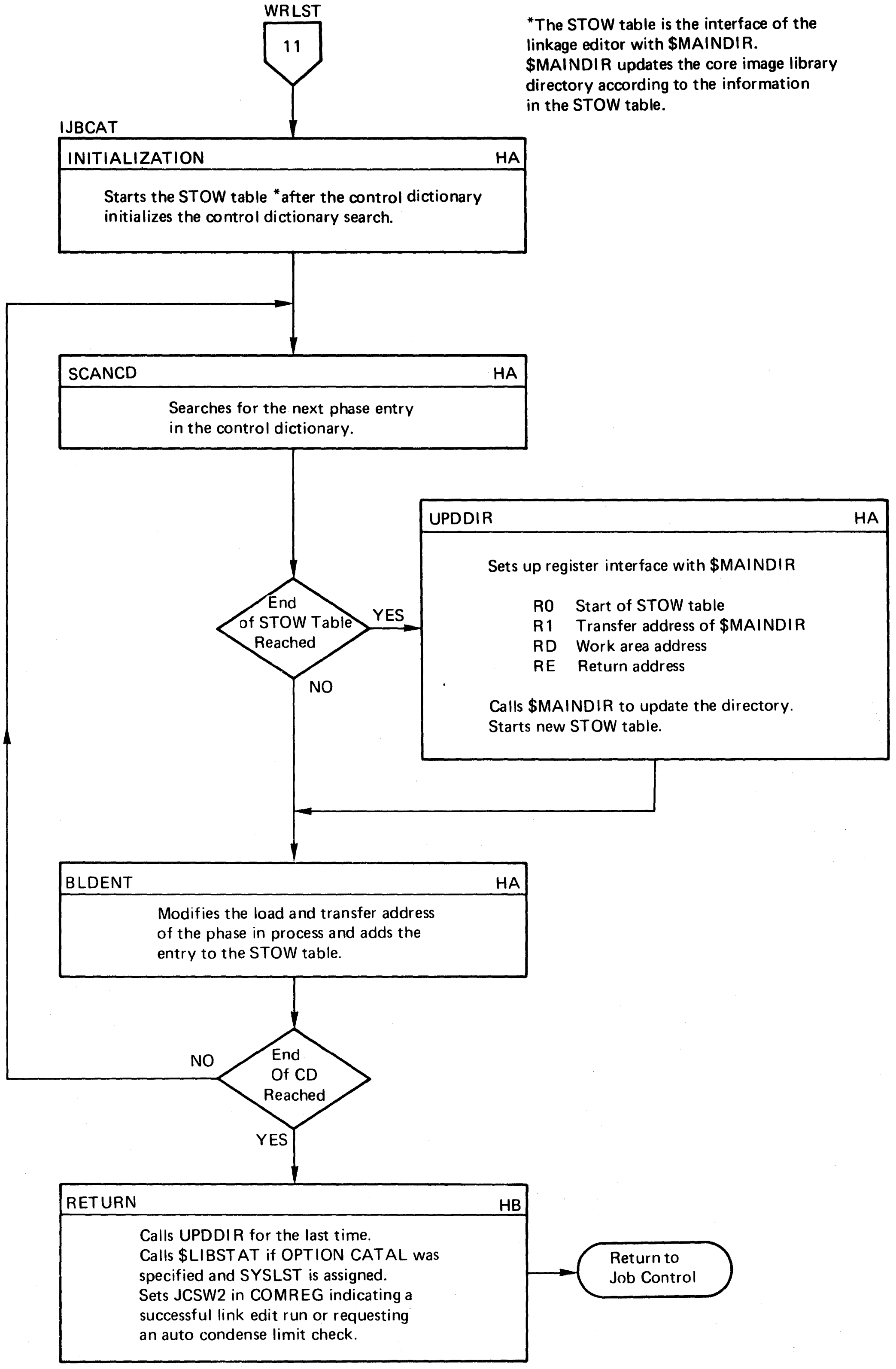

TSTUNR
10
RELROUT
Any Relocatable Phase Processed
NO
YES
INITIALIZE PASS3 RLD PROCESSING
GE, GG
Modifies Pass2 RLD Processor to insert relocation information for relocatable phases into the space reserved by the 'PHASE' processor. Repositions SYS001 to the start of the RLD file.
TSTUNR
10
WRLST
BLOCK PHASE HEADER
GF, GG
Writes the last block in the core image library and verifies all core image blocks previously written and closes SYSLNK and SYS001.
12
IJBCAT

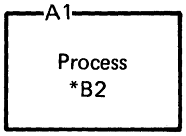
Chart 12. IJBCAT - Catalog Routine


WRLST
11
IJBCAT
INITIALIZATION
HA
Starts the STOW table *after the control dictionary initializes the control dictionary search.
SCANCD
HA
Searches for the next phase entry in the control dictionary.
End of STOW Table Reached
YES
NO
UPDDIR
HA
Sets up register interface with $MAINDIR
R0 Start of STOW table
R1 Transfer address of $MAINDIR
RD Work area address
RE Return address
Calls $MAINDIR to update the directory.
Starts new STOW table.
BLDENT
HA
Modifies the load and transfer address of the phase in process and adds the entry to the STOW table.
End Of CD Reached
NO
YES
RETURN
HB
Calls UPDDIR for the last time.
Calls $LIBSTAT if OPTION CATAL was specified and SYSLST is assigned.
Sets JCSW2 in COMREG indicating a successful link edit run or requesting an auto condense limit check.
Return to Job Control

*The STOW table is the interface of the linkage editor with $MAINDIR. $MAINDIR updates the core image library directory according to the information in the STOW table.

DETAIL CHARTS

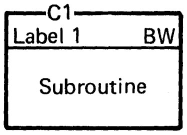
DESCRIPTION



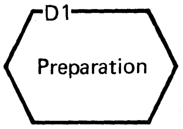
A group of program instructions that perform a processing function of the program. The label, if any, is shown above the block.

*B2

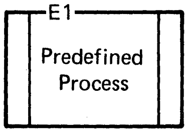
If any additional explanation is required, its location on the chart is identified by an asterisk and the block ID.



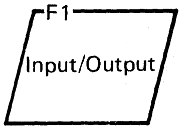
Description of a subroutine. The starting label of the routine appears above the stripe. If the subroutine is documented in detail on another flowchart, the ID of this flowchart is also shown.



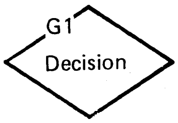
An instruction, or group of instructions, that changes portions of a routine or initializes a routine for given conditions.



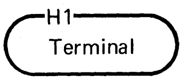
A group of operations not detailed in the flowcharts in this manual, such as user's routines.



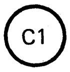
Any function of an input/output device or program, usually branching to an I/O routine to perform the function stated in the block.



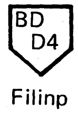
Points where the program branches to alternate processing, based upon variable conditions such as program switch settings and test results.



The beginning, end or point of interruption in a program.



On-page connector. An entry from or an exit to another function on the same flowchart. The number in the connector identifies the corresponding entry or exit on the chart.



Off-page connector, an entry from, or an exit to, a given point on another flowchart. The characters in the connector identify the chart and block. The corresponding label, if any, is placed outside the connector. For multiple entries and exits, an asterisk appears in the connector and the characters are listed nearby.

EXAMPLE

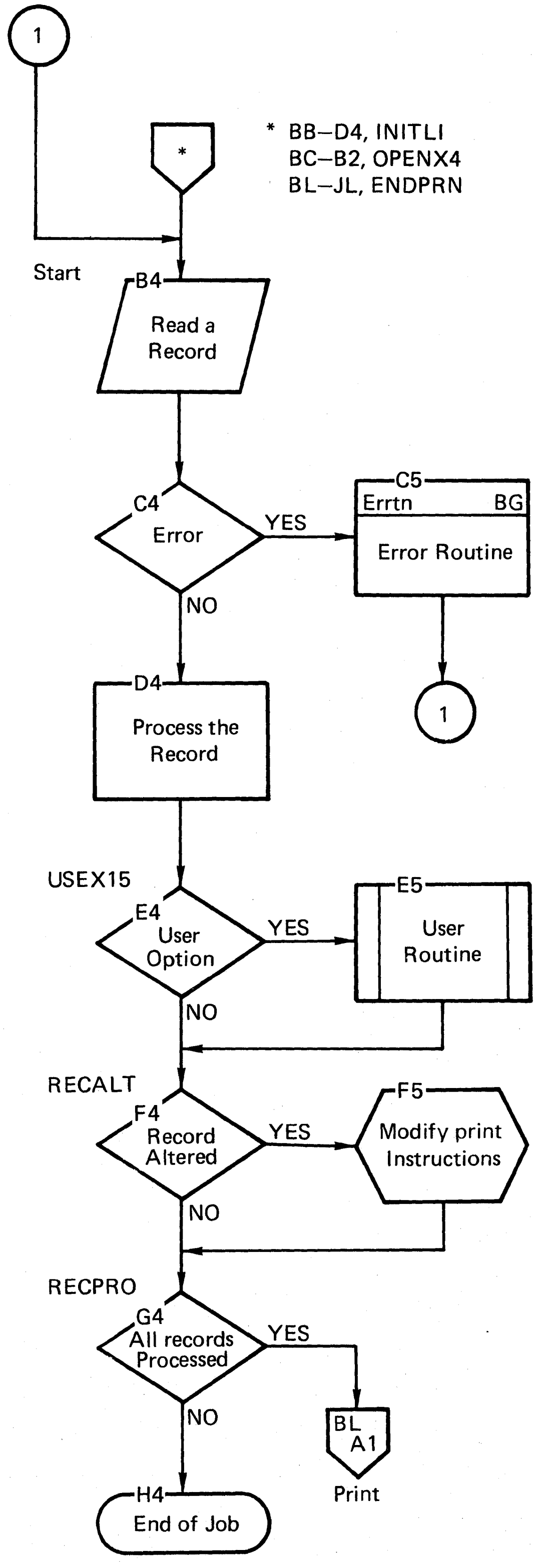
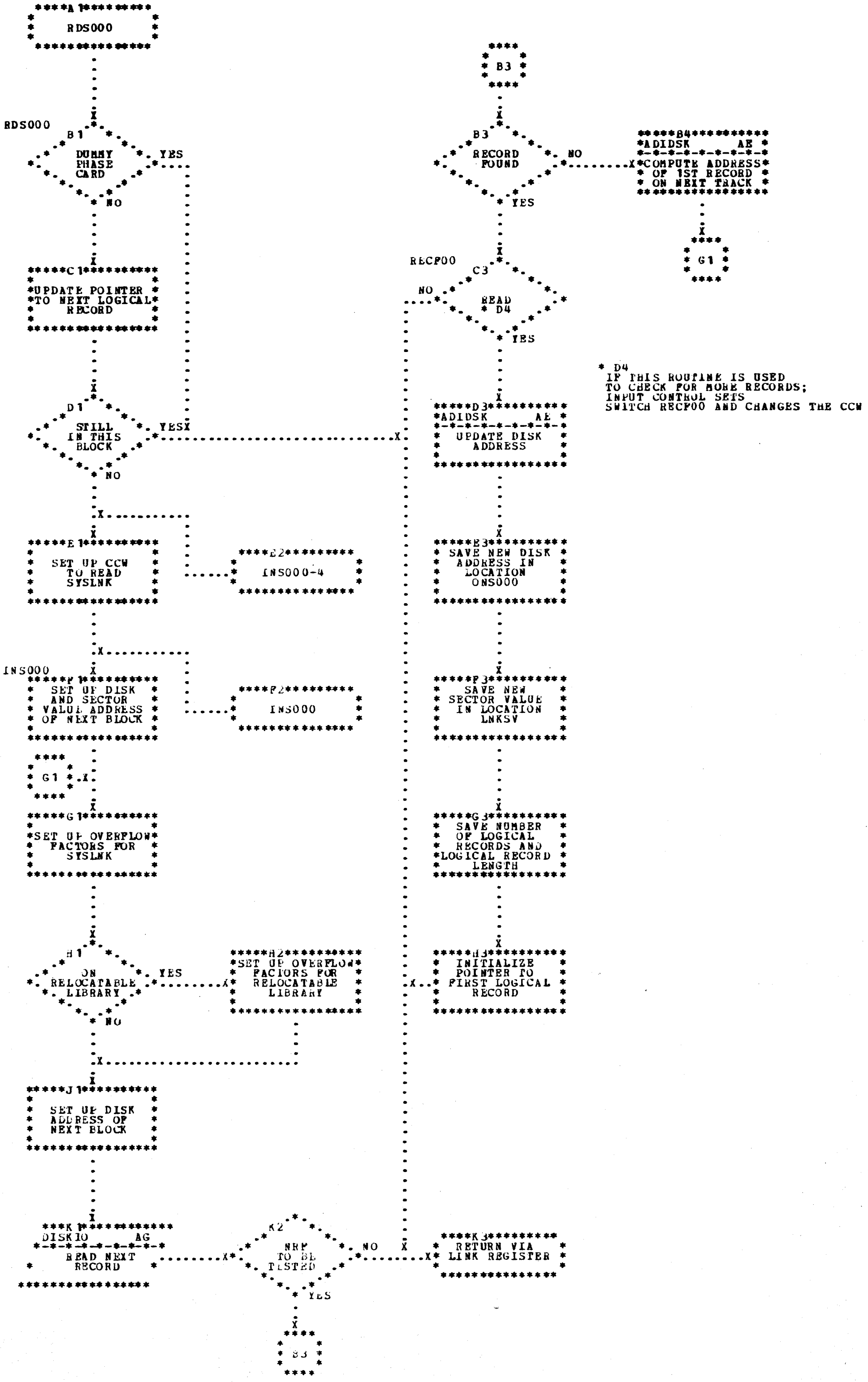


Chart AA. IJBLNK - Read SYSLNK Subroutine. Refer to Chart 02


A1
RDS000
B3
RDS000
B1
DUMMY PHASE CARD
YES
NO
B3
RECORD FOUND
NO
YES
B4
ADIDSK AE
COMPUTE ADDRESS OF 1ST RECORD ON NEXT TRACK
G1
C1
UPDATE POINTER TO NEXT LOGICAL RECORD
RECF00
C3
NO
READ
D4
YES
D1
STILL IN THIS BLOCK
YES
NO
D3
ADIDSK AE
UPDATE DISK ADDRESS
E1
SET UP CCW TO READ SYSLNK
E2
INS000-4
E3
SAVE NEW DISK ADDRESS IN LOCATION ONS000
INS000
F1
SET UP DISK AND SECTOR VALUE ADDRESS OF NEXT BLOCK
F2
INS000
F3
SAVE NEW SECTOR VALUE IN LOCATION LNKSV
G1
G1
SET UP OVERFLOW FACTORS FOR SYSLNK
G3
SAVE NUMBER OF LOGICAL RECORDS AND LOGICAL RECORD LENGTH
H1
ON RELOCATABLE LIBRARY
YES
NO
H2
SET UP OVERFLOW FACTORS FOR RELOCATABLE LIBRARY
H3
INITIALIZE POINTER TO FIRST LOGICAL RECORD
J1
SET UP DISK ADDRESS OF NEXT BLOCK
K1
DISKIO AG
READ NEXT RECORD
K2
NRF TO BE TESTED
NO
YES
K3
RETURN VIA LINK REGISTER
B3

* D4
IF THIS ROUTINE IS USED
TO CHECK FOR MORE RECORDS;
INPUT CONTROL SETS
SWITCH RECF00 AND CHANGES THE CCW

Chart AB. IJBLNK - Control Dictionary Search Subroutine. Refer to Chart 02

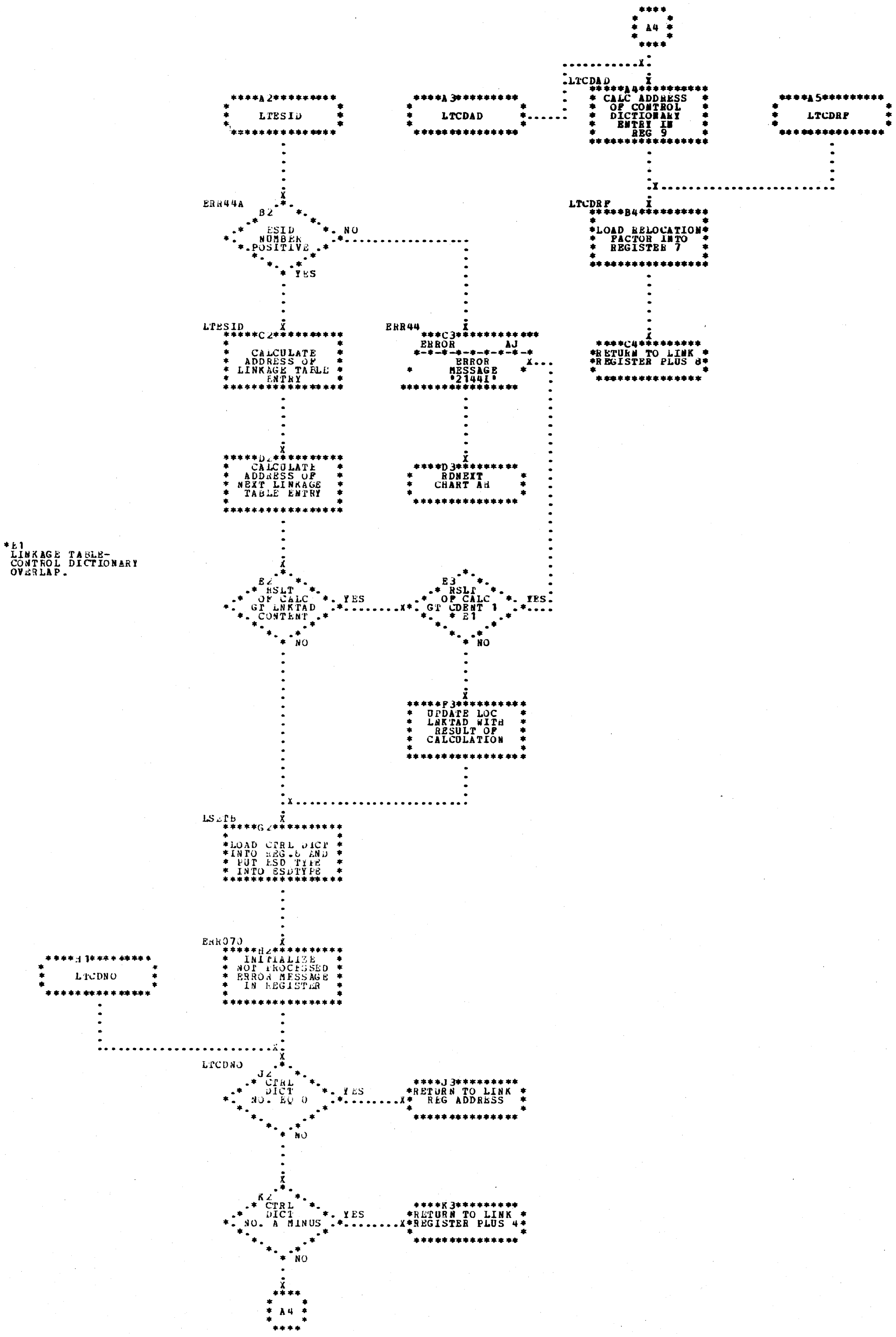

A4
LTCDAD
A4
CALC ADDRESS OF CONTROL DICTIONARY ENTRY IN REG 9
A2
LTESID
A3
LTCDAD
A5
LTCDRF
ERR44A
B2
ESID NUMBER POSITIVE
NO
YES
LTCDRF
B4
LOAD RELOCATION FACTOR INTO REGISTER 7
LTESID
C2
CALCULATE ADDRESS OF LINKAGE TABLE ENTRY
ERR44
C3
ERROR
AJ
ERROR MESSAGE '2144I'
C4
RETURN TO LINK REGISTER PLUS 8
D2
CALCULATE ADDRESS OF NEXT LINKAGE TABLE ENTRY
D3
RDNEXT CHART AH
*E1
LINKAGE TABLE-
CONTROL DICTIONARY
OVERLAP.
E2
RSLT OF CALC GT LNKTAD CONTENT
YES
NO
E3
RSLT OF CALC GT CDENT 1 * E1
YES
NO
F3
UPDATE LOC LNKTAD WITH RESULT OF CALCULATION
LSETB
G2
LOAD CTRL DICT INTO REG.6 AND PUT ESD TYPE INTO ESDTYPE
ERR070
H2
INITIALIZE NOT PROCESSED ERROR MESSAGE IN REGISTER
H1
LTCDNO
LTCDNO
J2
CTRL DICT NO. EQ 0
YES
NO
J3
RETURN TO LINK REG ADDRESS
K2
CTRL DICT NO. A MINUS
YES
NO
K3
RETURN TO LINK REGISTER PLUS 4
A4

Chart AC. IJBLNK - Label Search and Convert to Binary Subroutines. Refer to Chart 02

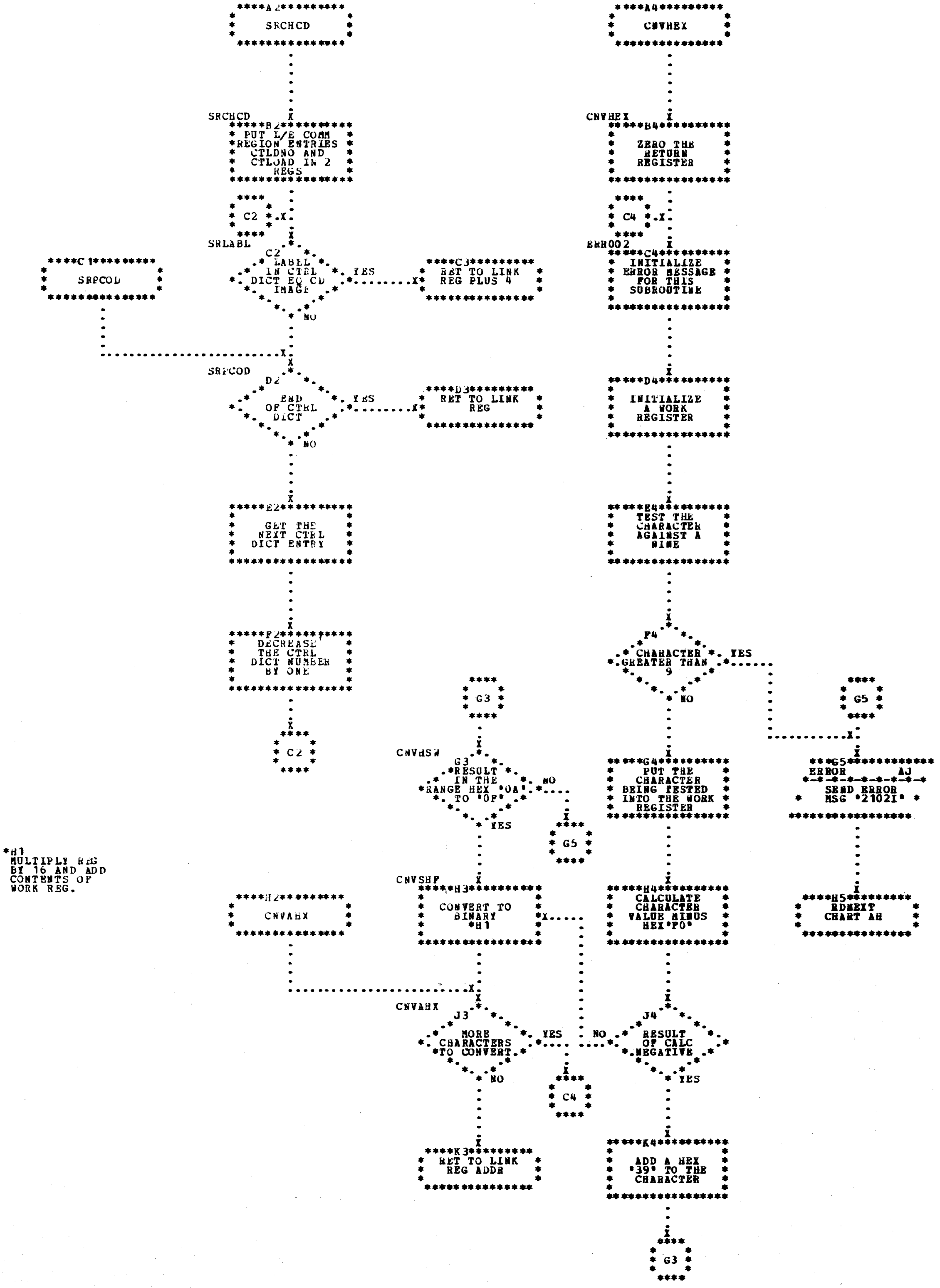
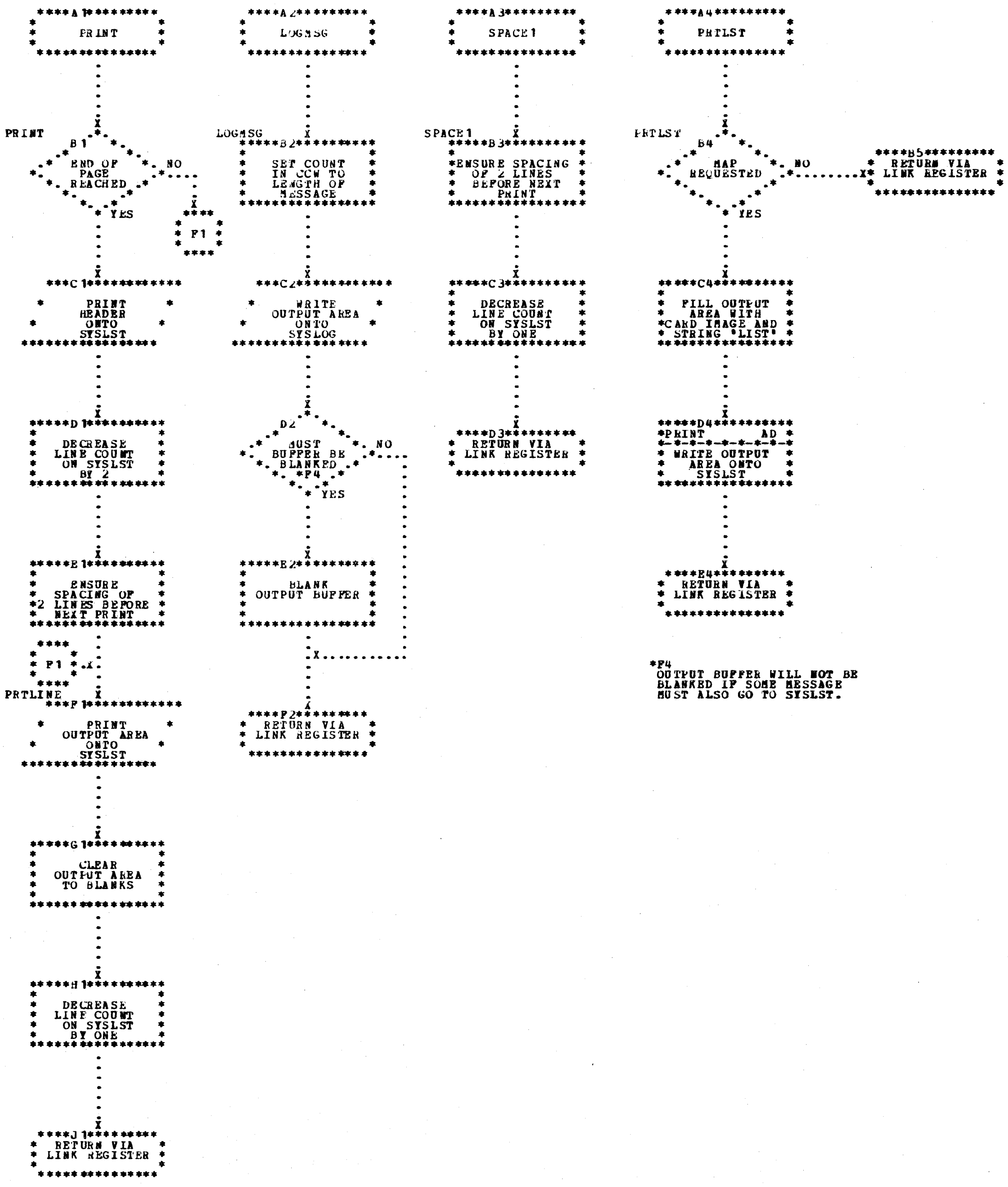

A2 SRCHCD
SRCHCD
B2 PUT L/E COMM REGION ENTRIES CTLDNO AND CTLOAD IN 2 REGS
C2
SRLABL
C2 LABEL IN CTRL DICT EQ CD IMAGE
YES
NO
C1 SRPCOD
C3 RET TO LINK REG PLUS 4
SRPCOD
D2 END OF CTRL DICT
YES
NO
D3 RET TO LINK REG
E2 GET THE NEXT CTRL DICT ENTRY
F2 DECREASE THE CTRL DICT NUMBER BY ONE
C2
A4 CNVHEX
CNVHEX
B4 ZERO THE RETURN REGISTER
C4
ERR002
C4 INITIALIZE ERROR MESSAGE FOR THIS SUBROUTINE
D4 INITIALIZE A WORK REGISTER
E4 TEST THE CHARACTER AGAINST A NINE
F4 CHARACTER GREATER THAN 9
YES
NO
G3
CNVHSW
G3 RESULT IN THE RANGE HEX 'OA' TO 'OF'
NO
YES
G5
G4 PUT THE CHARACTER BEING TESTED INTO THE WORK REGISTER
G5
G5 ERROR AJ
SEND ERROR MSG '2102I'
*H1
MULTIPLY REG BY 16 AND ADD CONTENTS OF WORK REG.
H2 CNVABX
CNVSHF
H3 CONVERT TO BINARY *H1
H4 CALCULATE CHARACTER VALUE MINUS HEX'F0'
H5 RDNEXT CHART AH
CNVABX
J3 MORE CHARACTERS TO CONVERT
YES
NO
C4
J4 RESULT OF CALC NEGATIVE
NO
YES
K3 RET TO LINK REG ADDR
K4 ADD A HEX '39' TO THE CHARACTER
G3

Chart AD. IJBLNK - Print Subroutines. Refer to Chart 02


A1 PRINT
PRINT
B1 END OF PAGE REACHED
NO
YES
F1
C1 PRINT HEADER ONTO SYSLST
D1 DECREASE LINE COUNT ON SYSLST BY 2
E1 ENSURE SPACING OF 2 LINES BEFORE NEXT PRINT
F1
PRTLINE
F1 PRINT OUTPUT AREA ONTO SYSLST
G1 CLEAR OUTPUT AREA TO BLANKS
H1 DECREASE LINE COUNT ON SYSLST BY ONE
J1 RETURN VIA LINK REGISTER
A2 LOGMSG
LOGMSG
B2 SET COUNT IN CCW TO LENGTH OF MESSAGE
C2 WRITE OUTPUT AREA ONTO SYSLOG
D2 MUST BUFFER BE BLANKED *F4
NO
YES
E2 BLANK OUTPUT BUFFER
F2 RETURN VIA LINK REGISTER
A3 SPACE1
SPACE1
B3 ENSURE SPACING OF 2 LINES BEFORE NEXT PRINT
C3 DECREASE LINE COUNT ON SYSLST BY ONE
D3 RETURN VIA LINK REGISTER
A4 PRTLST
PRTLST
B4 MAP REQUESTED
NO
YES
B5 RETURN VIA LINK REGISTER
C4 FILL OUTPUT AREA WITH CARD IMAGE AND STRING 'LIST'
D4 PRINT AD
WRITE OUTPUT AREA ONTO SYSLST
E4 RETURN VIA LINK REGISTER

*F4
OUTPUT BUFFER WILL NOT BE BLANKED IF SOME MESSAGE MUST ALSO GO TO SYSLST.

Chart AE. IJBLNK - Input Pointer to Address to be Updated. Refer to Chart 02

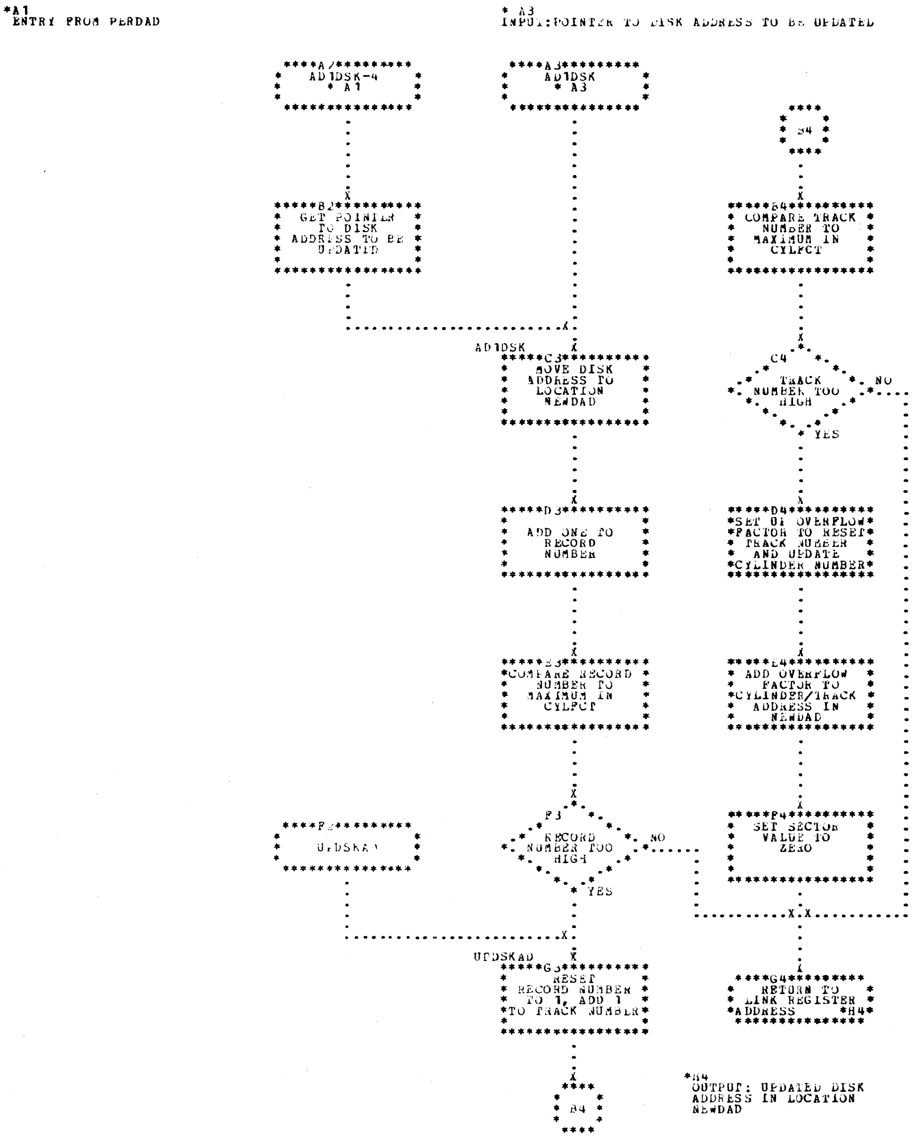
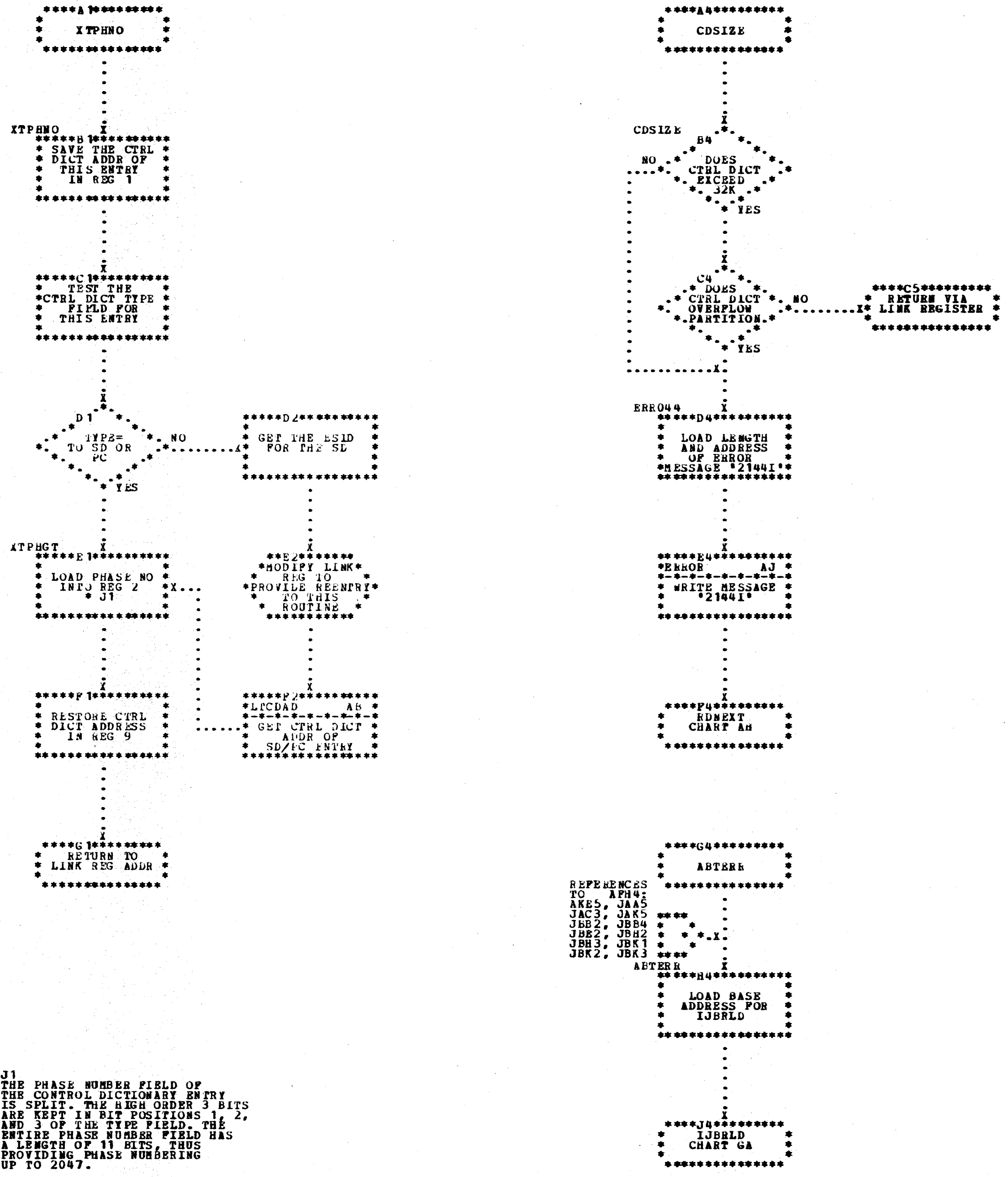
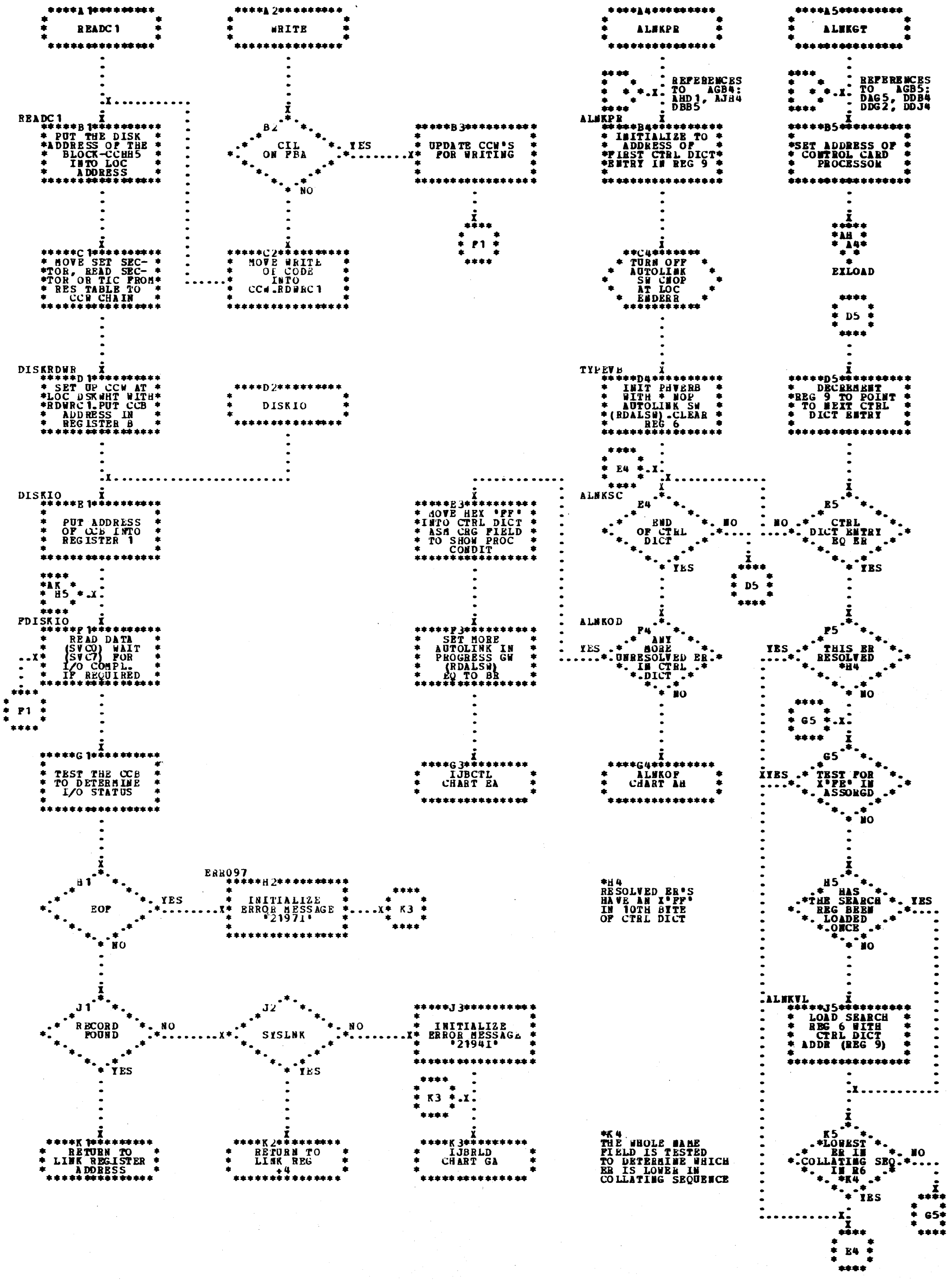

*A1
ENTRY FROM PERDAD
* A3
INPUT:POINTER TO DISK ADDRESS TO BE UPDATED
A2
AD1DSK-4
* A1
A3
AD1DSK
* A3
B4
B2
GET POINTER TO DISK ADDRESS TO BE UPDATED
B4
COMPARE TRACK NUMBER TO MAXIMUM IN CYLFCT
AD1DSK
C3
MOVE DISK ADDRESS TO LOCATION NEWDAD
C4
TRACK NUMBER TOO HIGH
NO
YES
D3
ADD ONE TO RECORD NUMBER
D4
SET UP OVERFLOW FACTOR TO RESET TRACK NUMBER AND UPDATE CYLINDER NUMBER
E3
COMPARE RECORD NUMBER TO MAXIMUM IN CYLFCT
E4
ADD OVERFLOW FACTOR TO CYLINDER/TRACK ADDRESS IN NEWDAD
F2
UPDSKAD
F3
RECORD NUMBER TOO HIGH
NO
YES
F4
SET SECTOR VALUE TO ZERO
UPDSKAD
G3
RESET RECORD NUMBER TO 1, ADD 1 TO TRACK NUMBER
G4
RETURN TO LINK REGISTER ADDRESS
*H4
B4
*H4
OUTPUT: UPDATED DISK ADDRESS IN LOCATION NEWDAD

Chart AF. IJBLNK - Extract Phase Number, Overflow Test, and Phase Load Subroutines
Refer to Chart 02


A1
XTPHNO
XTPHNO
B1
SAVE THE CTRL DICT ADDR OF THIS ENTRY IN REG 1
C1
TEST THE CTRL DICT TYPE FIELD FOR THIS ENTRY
D1
TYPE= TO SD OR PC
NO
YES
D2
GET THE ESID FOR THE SD
XTPHGT
E1
LOAD PHASE NO INTO REG 2
* J1
E2
MODIFY LINK REG TO PROVIDE REENTRY TO THIS ROUTINE
F1
RESTORE CTRL DICT ADDRESS IN REG 9
F2
LTCDAD AB
GET CTRL DICT ADDR OF SD/PC ENTRY
G1
RETURN TO LINK REG ADDR
A4
CDSIZE
CDSIZE
B4
DOES CTRL DICT EXCEED 32K
NO
YES
C4
DOES CTRL DICT OVERFLOW PARTITION
NO
YES
C5
RETURN VIA LINK REGISTER
ERR044
D4
LOAD LENGTH AND ADDRESS OF ERROR MESSAGE '2144I'
E4
ERROR AJ
WRITE MESSAGE '2144I'
F4
RDNEXT CHART AH
G4
ABTERR
REFERENCES TO AFH4: AKE5, JAA5 JAC3, JAK5 JBB2, JBB4 JBE2, JBH2 JBH3, JBK1 JBK2, JBK3
ABTERR
H4
LOAD BASE ADDRESS FOR IJBRLD
J4
IJBRLD CHART GA

*J1
THE PHASE NUMBER FIELD OF THE CONTROL DICTIONARY ENTRY IS SPLIT. THE HIGH ORDER 3 BITS ARE KEPT IN BIT POSITIONS 1, 2, AND 3 OF THE TYPE FIELD. THE ENTIRE PHASE NUMBER FIELD HAS A LENGTH OF 11 BITS, THUS PROVIDING PHASE NUMBERING UP TO 2047.

Chart AG. IJBLNK - Read/Write and Autolink Processing Subroutines. Refer to Chart 02


A1
READC1
A2
WRITE
A4
ALNKPR
A5
ALNKGT
REFERENCES TO AGB4: AHD1, AJH4 DBB5
REFERENCES TO AGB5: DAG5, DDB4 DDG2, DDJ4
READC1
B1
PUT THE DISK ADDRESS OF THE BLOCK-CCHH5 INTO LOC ADDRESS
B2
CIL ON FBA
YES
NO
B3
UPDATE CCW'S FOR WRITING
P1
ALNKPR
B4
INITIALIZE TO ADDRESS OF FIRST CTRL DICT ENTRY IN REG 9
B5
SET ADDRESS OF CONTROL CARD PROCESSOR
C1
MOVE SET SECTOR, READ SECTOR OR TIC FROM RES TABLE TO CCW CHAIN
C2
MOVE WRITE OF CODE INTO CCW.RDWRC1
C4
TURN OFF AUTOLINK SW CNOP AT LOC ENDERR
AH A4
EXLOAD
D5
DISKRDWR
D1
SET UP CCW AT LOC DSKWHT WITH RDWRC1.PUT CCB ADDRESS IN REGISTER B
D2
DISKIO
TYPEVB
D4
INIT PHVERB WITH * NOP AUTOLINK SW (RDALSW).CLEAR REG 6
D5
DECREMENT REG 9 TO POINT TO NEXT CTRL DICT ENTRY
E4
DISKIO
E1
PUT ADDRESS OF CCB INTO REGISTER 1
E3
MOVE HEX 'FF' INTO CTRL DICT ASM CHG FIELD TO SHOW PROC CONDIT
ALNKSC
E4
END OF CTRL DICT
NO
YES
D5
NO
E5
CTRL DICT ENTRY EQ ER
YES
AK H5
FDISKIO
F1
READ DATA (SVC0) WAIT (SVC7) FOR I/O COMPL. IF REQUIRED
P1
F3
SET MORE AUTOLINK IN PROGRESS GW (RDALSW) EQ TO ER
ALNKOD
F4
ANY MORE UNRESOLVED ER IN CTRL DICT
YES
NO
F5
THIS ER RESOLVED *H4
YES
NO
G5
G1
TEST THE CCB TO DETERMINE I/O STATUS
G3
IJBCTL CHART EA
G4
ALNKOP CHART AH
G5
TEST FOR X'FF' IN ASSORGD
YES
NO
H1
EOF
YES
NO
ERR097
H2
INITIALIZE ERROR MESSAGE '2197I'
K3
H5
HAS THE SEARCH REG BEEN LOADED ONCE
YES
NO
J1
RECORD FOUND
NO
YES
J2
SYSLNK
NO
YES
J3
INITIALIZE ERROR MESSAGE '2194I'
ALNKVL
J5
LOAD SEARCH REG 6 WITH CTRL DICT ADDR (REG 9)
K3
K1
RETURN TO LINK REGISTER ADDRESS
K2
RETURN TO LINK REG +4
K3
IJBRLD CHART GA
K5
LOWEST ER IN COLLATING SEQ IN R6 *K4
NO
YES
G5
E4

*H4
RESOLVED ER'S
HAVE AN X'FF'
IN 10TH BYTE
OF CTRL DICT

*K4
THE WHOLE NAME
FIELD IS TESTED
TO DETERMINE WHICH
ER IS LOWER IN
COLLATING SEQUENCE

Chart AH. IJBLNK - Read Input Stream. Refer to Chart 02

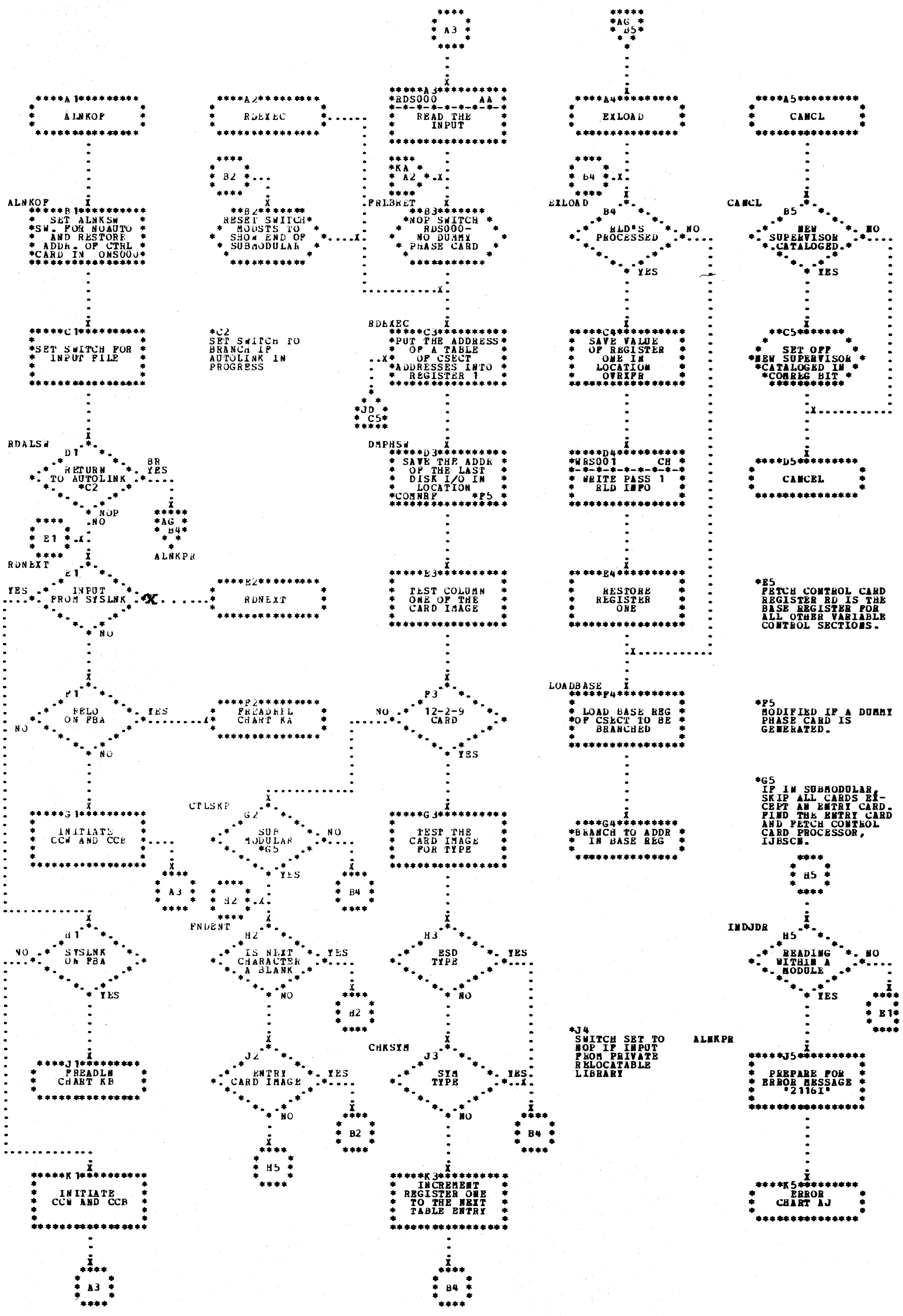

A3
AG B5
A1 ALNKOF
A2 RDEXEC
A3 RDS000 AA READ THE INPUT
A4 EXLOAD
A5 CANCL
B2
KA A2
B4
ALNKOF
B1 SET ALNKSW SW. FOR NOAUTO AND RESTORE ADDR. OF CTRL CARD IN ONS000
B2 RESET SWITCH MODSTS TO SHOW END OF SUBMODULAR
FRLBRET
B3 NOP SWITCH RDS000- NO DUMMY PHASE CARD
EXLOAD
B4 RLD'S PROCESSED
NO
YES
CANCL
B5 NEW SUPERVISOR CATALOGED
NO
YES
C1 SET SWITCH FOR INPUT FILE
*C2 SET SWITCH TO BRANCH IF AUTOLINK IN PROGRESS
RDEXEC
C3 PUT THE ADDRESS OF A TABLE OF CSECT ADDRESSES INTO REGISTER 1
JD C5
C4 SAVE VALUE OF REGISTER ONE IN LOCATION OVRXPR
C5 SET OFF NEW SUPERVISOR CATALOGED IN COMREG BIT
RDALSW
D1 RETURN TO AUTOLINK *C2
BR YES
NOP NO
AG B4
ALNKPR
E1
DMPHSW
D3 SAVE THE ADDR OF THE LAST DISK I/O IN LOCATION COMNRF *F5
D4 WRS001 CH WRITE PASS 1 RLD INFO
D5 CANCEL
RDNEXT
E1 INPUT FROM SYSLNK
YES
NO
E2 RDNEXT
E3 TEST COLUMN ONE OF THE CARD IMAGE
E4 RESTORE REGISTER ONE
*E5 FETCH CONTROL CARD REGISTER RD IS THE BASE REGISTER FOR ALL OTHER VARIABLE CONTROL SECTIONS.
F1 RELO ON FBA
YES
NO
NO
F2 FREADREL CHART KA
F3 12-2-9 CARD
NO
YES
LOADBASE
F4 LOAD BASE REG OF CSECT TO BE BRANCHED
*F5 MODIFIED IF A DUMMY PHASE CARD IS GENERATED.
G1 INITIATE CCW AND CCB
A3
CTLSKP
G2 SUB MODULAR *G5
NO
YES
B4
H2
G3 TEST THE CARD IMAGE FOR TYPE
G4 BRANCH TO ADDR IN BASE REG
*G5 IF IN SUBMODULAR, SKIP ALL CARDS EXCEPT AN ENTRY CARD. FIND THE ENTRY CARD AND FETCH CONTROL CARD PROCESSOR, IJBSCH.
H5
H1 SYSLNK ON FBA
NO
YES
FNDENT
H2 IS NEXT CHARACTER A BLANK
YES
NO
H2
H3 ESD TYPE
YES
NO
INDJDR
H5 READING WITHIN A MODULE
NO
YES
E1
*J4 SWITCH SET TO NOP IF INPUT FROM PRIVATE RELOCATABLE LIBRARY
J1 FREADLN CHART KB
J2 ENTRY CARD IMAGE
YES
NO
B2
CHKSYM
J3 SYM TYPE
YES
NO
B4
H5
ALNKPR
J5 PREPARE FOR ERROR MESSAGE '2116I'
K1 INITIATE CCW AND CCB
A3
K3 INCREMENT REGISTER ONE TO THE NEXT TABLE ENTRY
B4
K5 ERROR CHART AJ

Chart AJ. IJBLNK - Non-Abort Error and Cverlay Subroutines. Refer to Chart C2

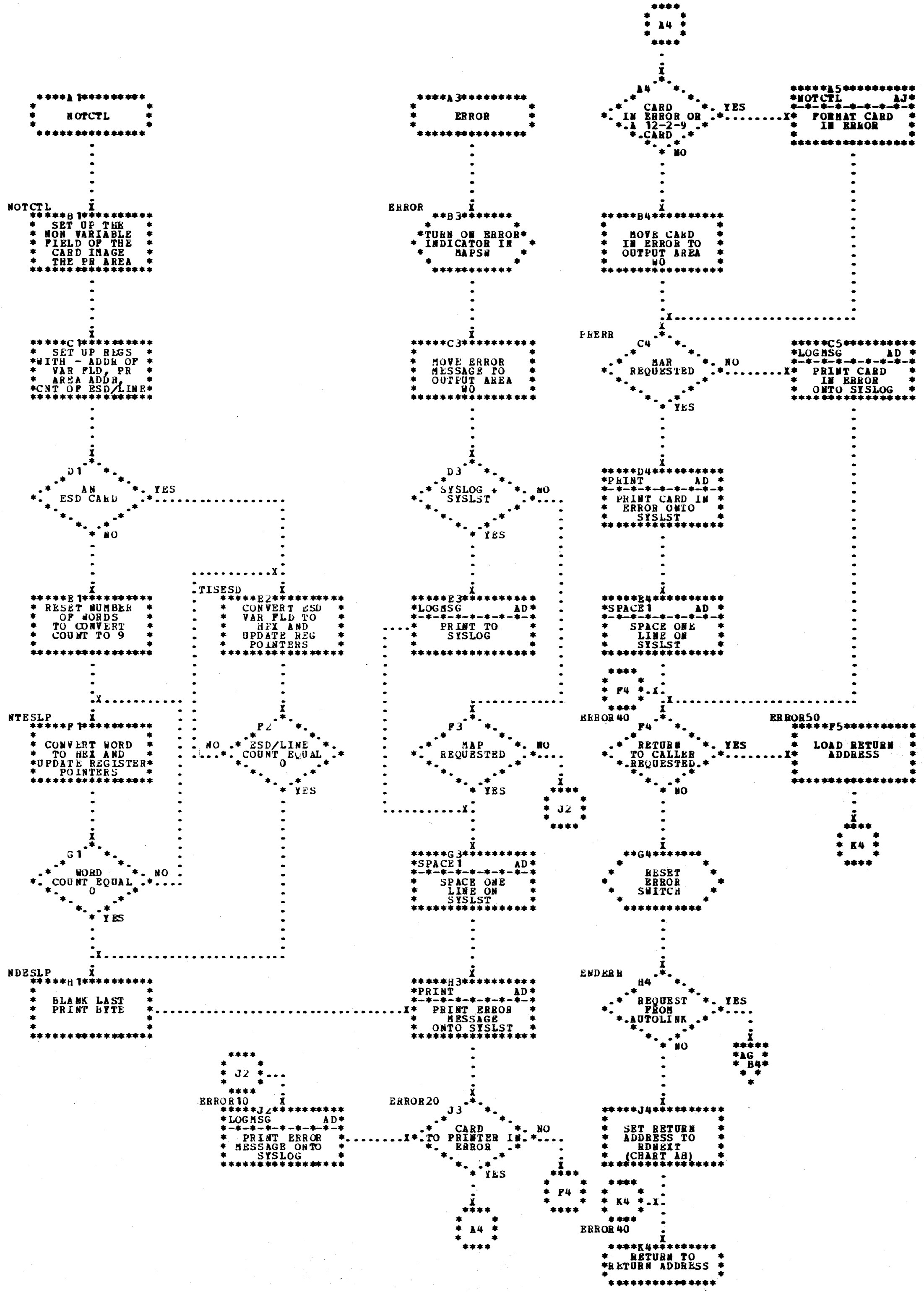


Chart AK. IJBLNK - Core Image Block Building Subroutine. Refer to Chart 02

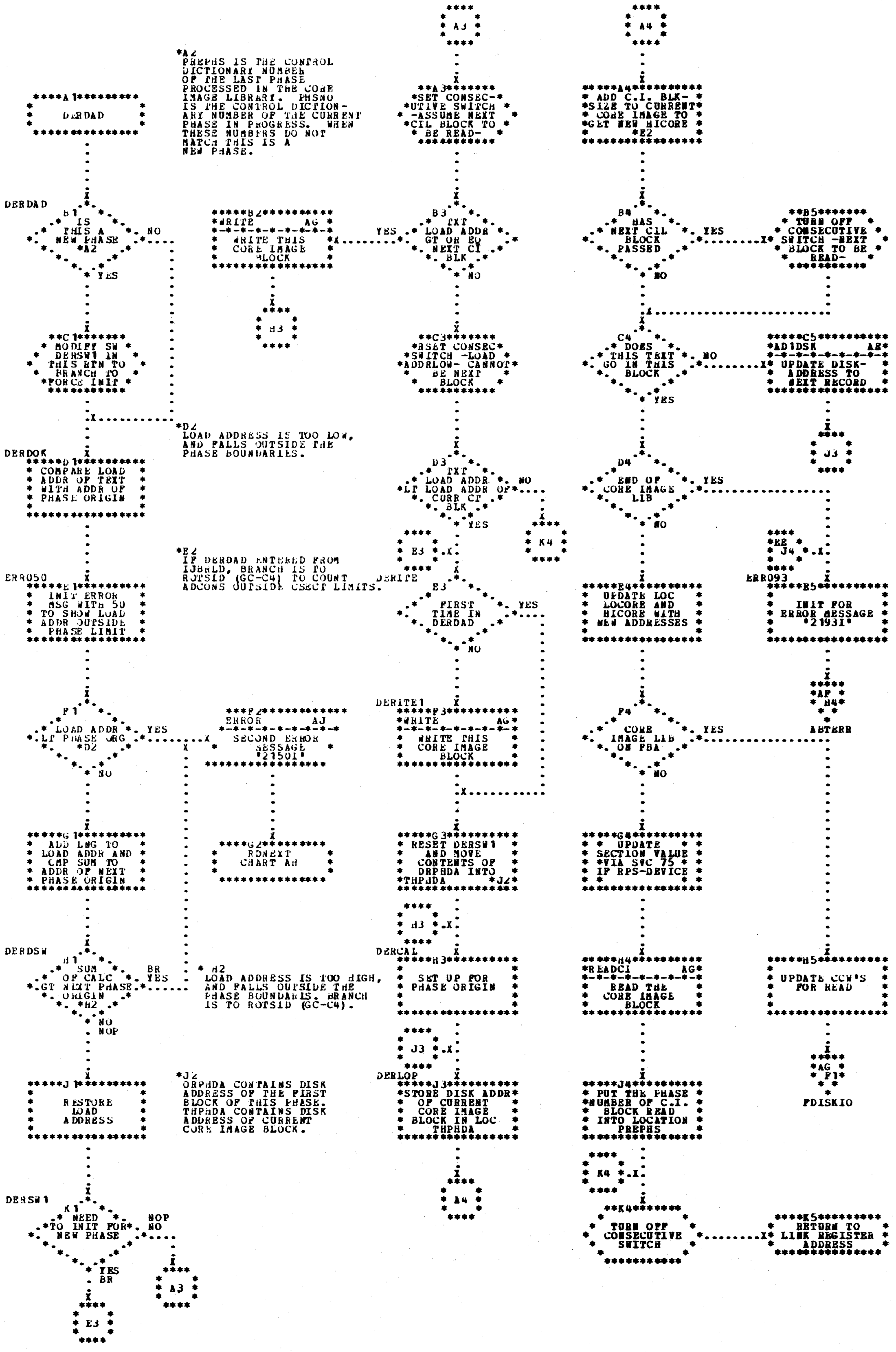
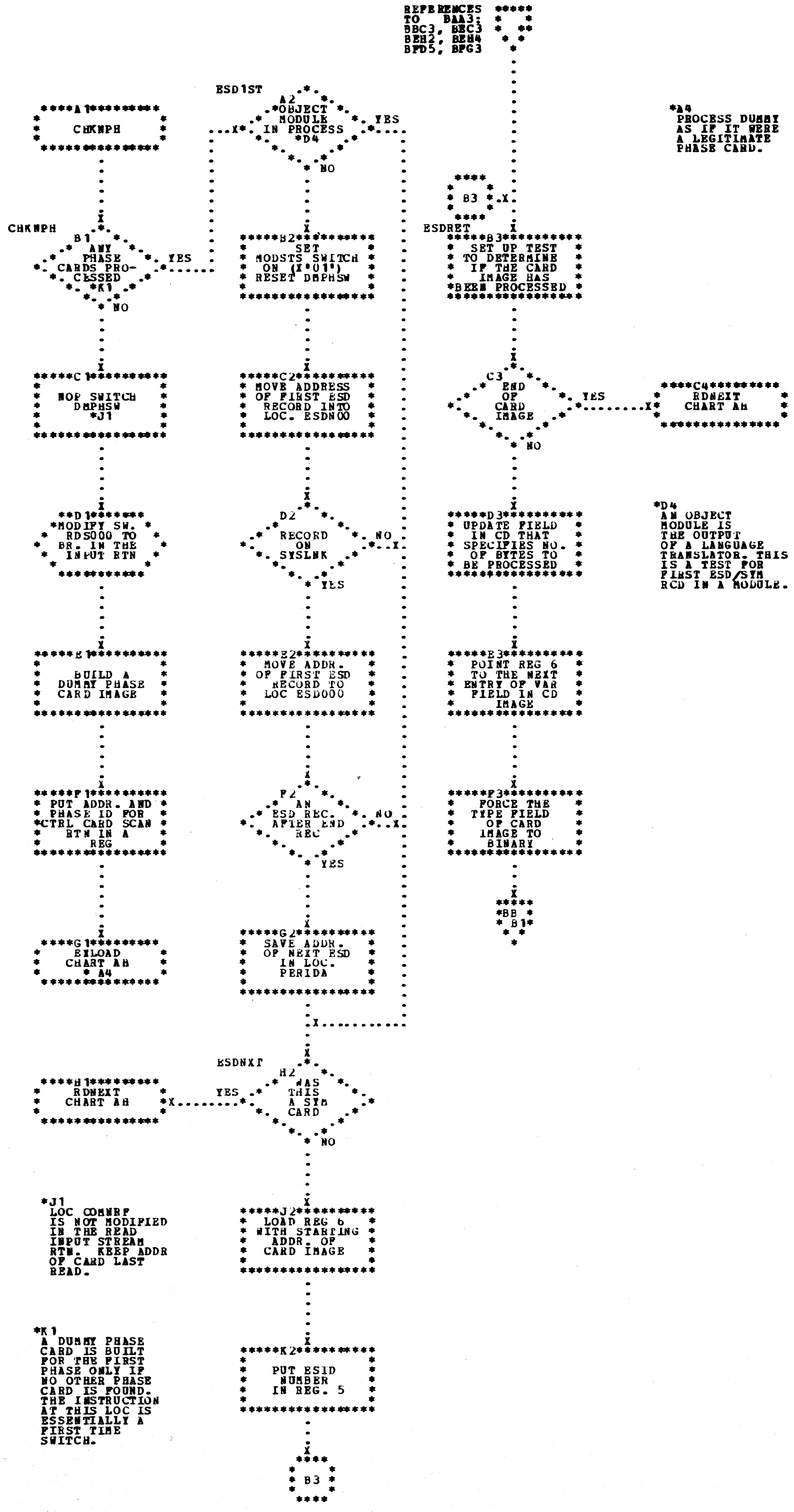

A1
DERDAD
DERDAD
B1
IS THIS A NEW PHASE *A2
YES
NO
C1
MODIFY SW DERSW1 IN THIS RTN TO BRANCH TO FORCE INIT
DERDOK
D1
COMPARE LOAD ADDR OF TEXT WITH ADDR OF PHASE ORIGIN
ERR050
E1
INIT ERROR MSG WITH 50 TO SHOW LOAD ADDR OUTSIDE PHASE LIMIT
F1
LOAD ADDR LT PHASE ORG *D2
YES
NO
G1
ADD LNG TO LOAD ADDR AND CMP SUM TO ADDR OF NEXT PHASE ORIGIN
DERDSW
H1
SUM OF CALC GT NEXT PHASE ORIGIN *H2
BR YES
NO NOP
J1
RESTORE LOAD ADDRESS
DERSW1
K1
NEED TO INIT FOR NEW PHASE
NOP NO
YES BR
E3
A3
*A2
PREPHS IS THE CONTROL DICTIONARY NUMBER OF THE LAST PHASE PROCESSED IN THE CORE IMAGE LIBRARY. PHSNO IS THE CONTROL DICTIONARY NUMBER OF THE CURRENT PHASE IN PROGRESS. WHEN THESE NUMBERS DO NOT MATCH THIS IS A NEW PHASE.
B2
WRITE AG
WRITE THIS CORE IMAGE BLOCK
H3
*D2
LOAD ADDRESS IS TOO LOW, AND FALLS OUTSIDE THE PHASE BOUNDARIES.
*E2
IF DERDAD ENTERED FROM IJBRLD, BRANCH IS TO ROTSID (GC-C4) TO COUNT ADCONS OUTSIDE CSECT LIMITS.
F2
ERROR AJ
SECOND ERROR MESSAGE '21501'
G2
RDNEXT CHART AH
* H2
LOAD ADDRESS IS TOO HIGH, AND FALLS OUTSIDE THE PHASE BOUNDARIS. BRANCH IS TO ROTSID (GC-C4).
*J2
ORPHDA CONTAINS DISK ADDRESS OF THE FIRST BLOCK OF THIS PHASE. THPHDA CONTAINS DISK ADDRESS OF CURRENT CORE IMAGE BLOCK.
A3
A3
SET CONSECUTIVE SWITCH -ASSUME NEXT CIL BLOCK TO BE READ-
B3
TXT LOAD ADDR GT OR EQ NEXT CI BLK
YES
NO
C3
RSET CONSEC SWITCH -LOAD ADDRLOW- CANNOT BE NEXT BLOCK
D3
TXT LOAD ADDR LT LOAD ADDR OF CURR CI BLK
NO
YES
E3
K4
DERITE
E3
FIRST TIME IN DERDAD
YES
NO
DERITE1
F3
WRITE AG
WRITE THIS CORE IMAGE BLOCK
G3
RESET DERSW1 AND MOVE CONTENTS OF DRPHDA INTO THPHDA *J2
H3
DERCAL
H3
SET UP FOR PHASE ORIGIN
J3
DERLOP
J3
STORE DISK ADDR OF CURRENT CORE IMAGE BLOCK IN LOC THPHDA
A4
A4
A4
ADD C.I. BLK-SIZE TO CURRENT CORE IMAGE TO GET NEW HICORE *E2
B4
HAS NEXT CIL BLOCK PASSED
YES
NO
C4
DOES THIS TEXT GO IN THIS BLOCK
NO
YES
D4
END OF CORE IMAGE LIB
YES
NO
E4
UPDATE LOC LOCORE AND HICORE WITH NEW ADDRESSES
F4
CORE IMAGE LIB ON FBA
YES
NO
G4
UPDATE SECTION VALUE VIA SVC 75 IF RPS-DEVICE
H4
READCI AG
READ THE CORE IMAGE BLOCK
J4
PUT THE PHASE NUMBER OF C.I. BLOCK READ INTO LOCATION PREPHS
K4
K4
TURN OFF CONSECUTIVE SWITCH
B5
TURN OFF CONSECUTIVE SWITCH -NEXT BLOCK TO BE READ-
C5
AD1DSK AE
UPDATE DISK-ADDRESS TO NEXT RECORD
J3
AE
J4
ERR093
E5
INIT FOR ERROR MESSAGE '21931'
AF
H4
ABTERR
H5
UPDATE CCW'S FOR READ
AG
F1
FDISKIO
K5
RETURN TO LINK REGISTER ADDRESS

Chart BA. IJBESD - Initialize ESD Processor and ESD Processor Card Image Check (Part 1 of 3). Refer to Charts 04 and 05


REFERENCES TO BAA3; BBC3, BEC3 BEH2, BEH4 BFD5, BFG3
A1
CHKNPH
CHKNPH
B1
ANY PHASE CARDS PROCESSED
*K1
YES
NO
C1
NOP SWITCH DMPHSW
*J1
D1
MODIFY SW. RDS000 TO BR. IN THE INPUT RTN
E1
BUILD A DUMMY PHASE CARD IMAGE
F1
PUT ADDR. AND PHASE ID FOR CTRL CARD SCAN RTN IN A REG
G1
EXLOAD CHART AH
* A4
ESD1ST
A2
OBJECT MODULE IN PROCESS
*D4
YES
NO
B2
SET MODSTS SWITCH ON (X'01') RESET DMPHSW
C2
MOVE ADDRESS OF FIRST ESD RECORD INTO LOC. ESDN00
D2
RECORD ON SYSLNK
NO
YES
E2
MOVE ADDR. OF FIRST ESD RECORD TO LOC ESD000
F2
AN ESD REC. AFTER END REC
NO
YES
G2
SAVE ADDR. OF NEXT ESD IN LOC. PERIDA
ESDNXT
H2
WAS THIS A SYM CARD
YES
NO
H1
RDNEXT CHART AH
J2
LOAD REG 6 WITH STARTING ADDR. OF CARD IMAGE
K2
PUT ESID NUMBER IN REG. 5
B3
B3
ESDRET
B3
SET UP TEST TO DETERMINE IF THE CARD IMAGE HAS BEEN PROCESSED
C3
END OF CARD IMAGE
YES
NO
C4
RDNEXT CHART AH
D3
UPDATE FIELD IN CD THAT SPECIFIES NO. OF BYTES TO BE PROCESSED
E3
POINT REG 6 TO THE NEXT ENTRY OF VAR FIELD IN CD IMAGE
F3
FORCE THE TYPE FIELD OF CARD IMAGE TO BINARY
BB
B1

*A4
PROCESS DUMMY AS IF IT WERE A LEGITIMATE PHASE CARD.

*D4
AN OBJECT MODULE IS THE OUTPUT OF A LANGUAGE TRANSLATOR. THIS IS A TEST FOR FIRST ESD/SYM RCD IN A MODULE.

*J1
LOC COMNRF IS NOT MODIFIED IN THE READ INPUT STREAM RTN. KEEP ADDR OF CARD LAST READ.

*K1
A DUMMY PHASE CARD IS BUILT FOR THE FIRST PHASE ONLY IF NO OTHER PHASE CARD IS FOUND. THE INSTRUCTION AT THIS LOC IS ESSENTIALLY A FIRST TIME SWITCH.

Chart BB. IJBESD - ESD Processor Card Image Check (Part 2 of 3)
Refer to Charts 04 and 05

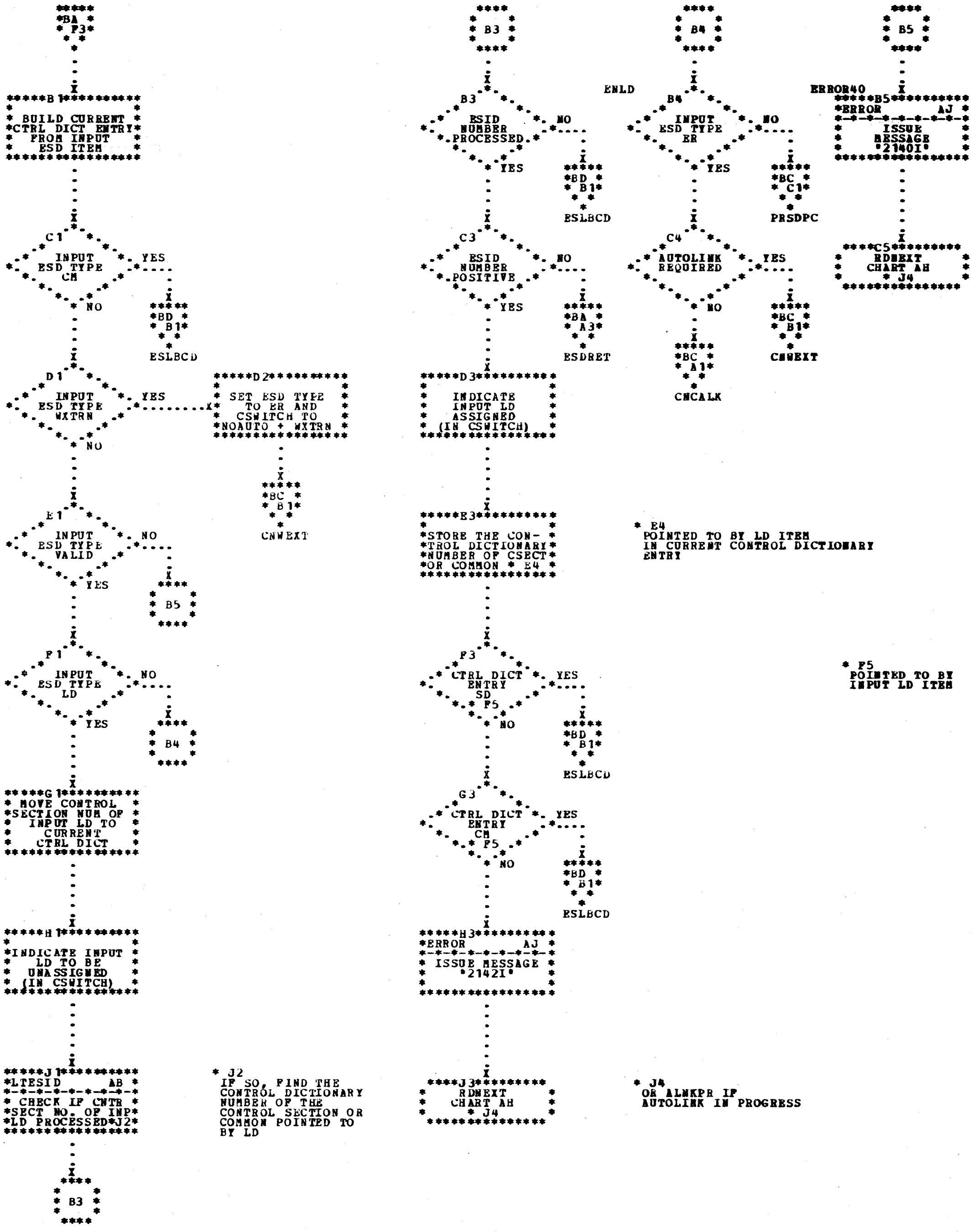

BA F3
B1 BUILD CURRENT CTRL DICT ENTRY FROM INPUT ESD ITEM
C1 INPUT ESD TYPE CM
YES
NO
BD B1
ESLBCD
D1 INPUT ESD TYPE WXTRN
YES
NO
D2 SET ESD TYPE TO ER AND CSWITCH TO NOAUTO + WXTRN
BC B1
CNWEXT
E1 INPUT ESD TYPE VALID
NO
YES
B5
F1 INPUT ESD TYPE LD
NO
YES
B4
G1 MOVE CONTROL SECTION NUM OF INPUT LD TO CURRENT CTRL DICT
H1 INDICATE INPUT LD TO BE UNASSIGNED (IN CSWITCH)
J1 LTESID AB
CHECK IF CNTR SECT NO. OF INP LD PROCESSED * J2
B3
* J2
IF SO, FIND THE CONTROL DICTIONARY NUMBER OF THE CONTROL SECTION OR COMMON POINTED TO BY LD
B3
B3 ESID NUMBER PROCESSED
NO
YES
BD B1
ESLBCD
C3 ESID NUMBER POSITIVE
NO
YES
BA A3
ESDRET
D3 INDICATE INPUT LD ASSIGNED (IN CSWITCH)
E3 STORE THE CONTROL DICTIONARY NUMBER OF CSECT OR COMMON * E4
F3 CTRL DICT ENTRY SD * F5
YES
NO
BD B1
ESLBCD
G3 CTRL DICT ENTRY CM * F5
YES
NO
BD B1
ESLBCD
H3 ERROR AJ
ISSUE MESSAGE '2142I'
J3 RDNEXT CHART AH * J4
B4
ENLD
B4 INPUT ESD TYPE ER
NO
YES
BC C1
PRSDPC
C4 AUTOLINK REQUIRED
YES
NO
BC B1
CNWEXT
BC A1
CNCALK
* E4
POINTED TO BY LD ITEM IN CURRENT CONTROL DICTIONARY ENTRY
* J4
OR ALNKPR IF AUTOLINK IN PROGRESS
B5
ERROR40
B5 ERROR AJ
ISSUE MESSAGE '2140I'
C5 RDNEXT CHART AH * J4
* F5
POINTED TO BY INPUT LD ITEM

Chart BC. IJBESD - ESD Processor Card Image Check (Part 3 of 3)
Refer to Charts 04 and 05

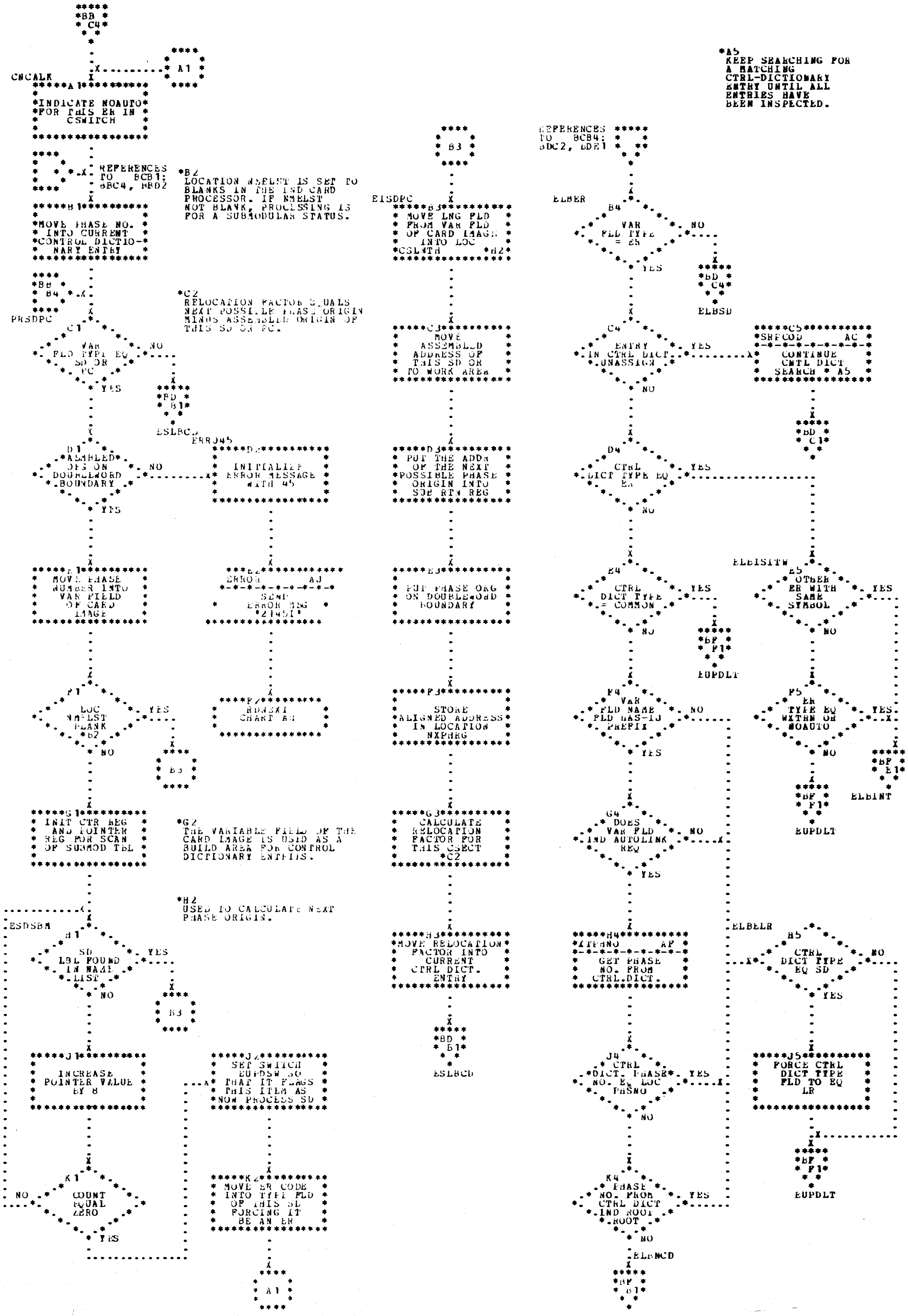

BB C4
A1
CNCALK A1
INDICATE NOAUTO FOR THIS ER IN CSWITCH
REFERENCES TO BCB1: BBC4, BBD2
*B2 LOCATION NMELST IS SET TO BLANKS IN THE END CARD PROCESSOR. IF NMELST NOT BLANK, PROCESSING IS FOR A SUBMODULAR STATUS.
B1 MOVE PHASE NO. INTO CURRENT CONTROL DICTIONARY ENTRY
BB B4
*C2 RELOCATION FACTOR EQUALS NEXT POSSIBLE PHASE ORIGIN MINUS ASSEMBLED ORIGIN OF THIS SD OR PC.
PRSDPC C1 VAR FLD TYPE EQ SD OR PC
NO
YES
BD B1
ESLBCD
D1 ASMBLED OFS ON DOUBLEWORD BOUNDARY
ERR045 D2 INITIALIZE ERROR MESSAGE WITH 45
E1 MOVE PHASE NUMBER INTO VAR FIELD OF CARD IMAGE
E2 ERROR AJ SEND ERROR MSG '2145I'
F1 LOC NMELST BLANK *B2
F2 RONEXT CHART A8
B3
G1 INIT CTR REG AND POINTER REG FOR SCAN OF SUBMOD TBL
*G2 THE VARIABLE FIELD OF THE CARD IMAGE IS USED AS A BUILD AREA FOR CONTROL DICTIONARY ENTRIES.
*H2 USED TO CALCULATE NEXT PHASE ORIGIN.
ESDSBM H1 SD LBL FOUND IN NAME LIST
B3
J1 INCREASE POINTER VALUE BY 8
J2 SET SWITCH EURDSW SO THAT IT FLAGS THIS ITEM AS NOW PROCESS SD
K1 COUNT EQUAL ZERO
K2 MOVE ER CODE INTO TYPE FLD OF THIS SD FORCING IT BE AN ER
A1
B3
EISDPC B3 MOVE LNG FLD FROM VAR FLD OF CARD IMAGE INTO LOC CSLNTH *H2
C3 MOVE ASSEMBLED ADDRESS OF THIS SD OR PC TO WORK AREA
D3 PUT THE ADDR OF THE NEXT POSSIBLE PHASE ORIGIN INTO SUB RTN REG
E3 PUT PHASE ORG ON DOUBLEWORD BOUNDARY
F3 STORE ALIGNED ADDRESS IN LOCATION NXPHRG
G3 CALCULATE RELOCATION FACTOR FOR THIS CSECT *C2
H3 MOVE RELOCATION FACTOR INTO CURRENT CTRL DICT. ENTRY
BD B1
ESLBCD
*A5 KEEP SEARCHING FOR A MATCHING CTRL-DICTIONARY ENTRY UNTIL ALL ENTRIES HAVE BEEN INSPECTED.
REFERENCES TO BCB4: BDC2, BDE1
ELBER B4 VAR FLD TYPE = ER
BD C4
ELBSD
C4 ENTRY IN CTRL DICT UNASSIGN
C5 SRFCOD AC CONTINUE CNTL DICT SEARCH * A5
BD C1
D4 CTRL DICT TYPE EQ ER
E4 CTRL DICT TYPE = COMMON
BF F1
EUPDLT
ELBISITW E5 OTHER ER WITH SAME SYMBOL
F4 VAR FLD NAME FLD HAS-IJ PREFIX
F5 ER TYPE EQ WXTRN OR NOAUTO
BF E1
ELBINT
BF F1
EUPDLT
G4 DOES VAR FLD IND AUTOLINK REQ
H4 XTFHNO AF GET PHASE NO. FROM CTRL.DICT.
ELBELR H5 CTRL DICT TYPE EQ SD
J4 CTRL DICT. PHASE NO. EQ LOC PHSNO
J5 FORCE CTRL DICT TYPE FLD TO EQ LR
BF F1
EUPDLT
K4 PHASE NO. FROM CTRL DICT IND ROOT
ELBNCD
BF B1
YES
NO

Chart BD. IJBESD - Process ESD Item against Control Dictionary and SD Processor
Refer to Charts 04 and 05

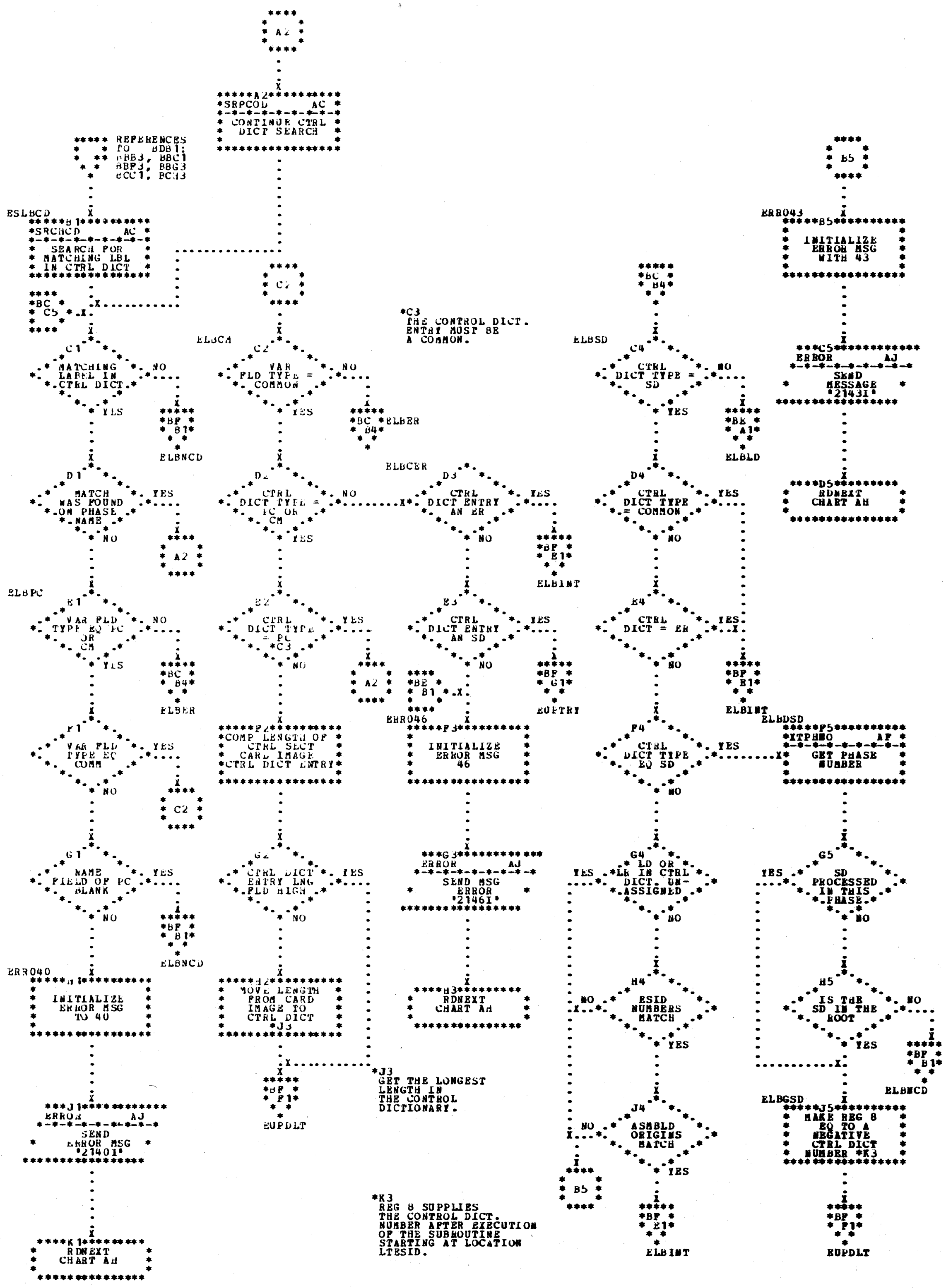
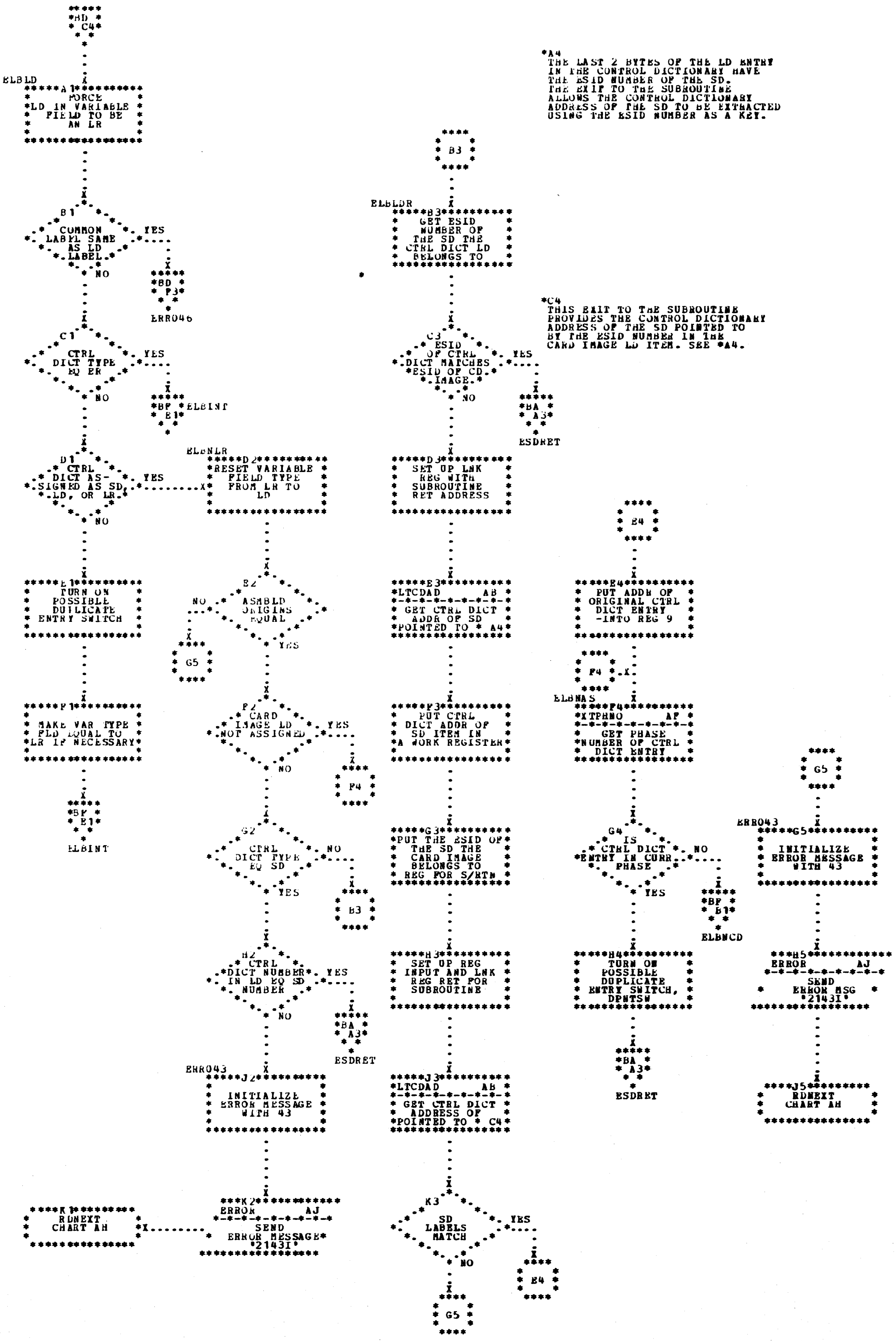


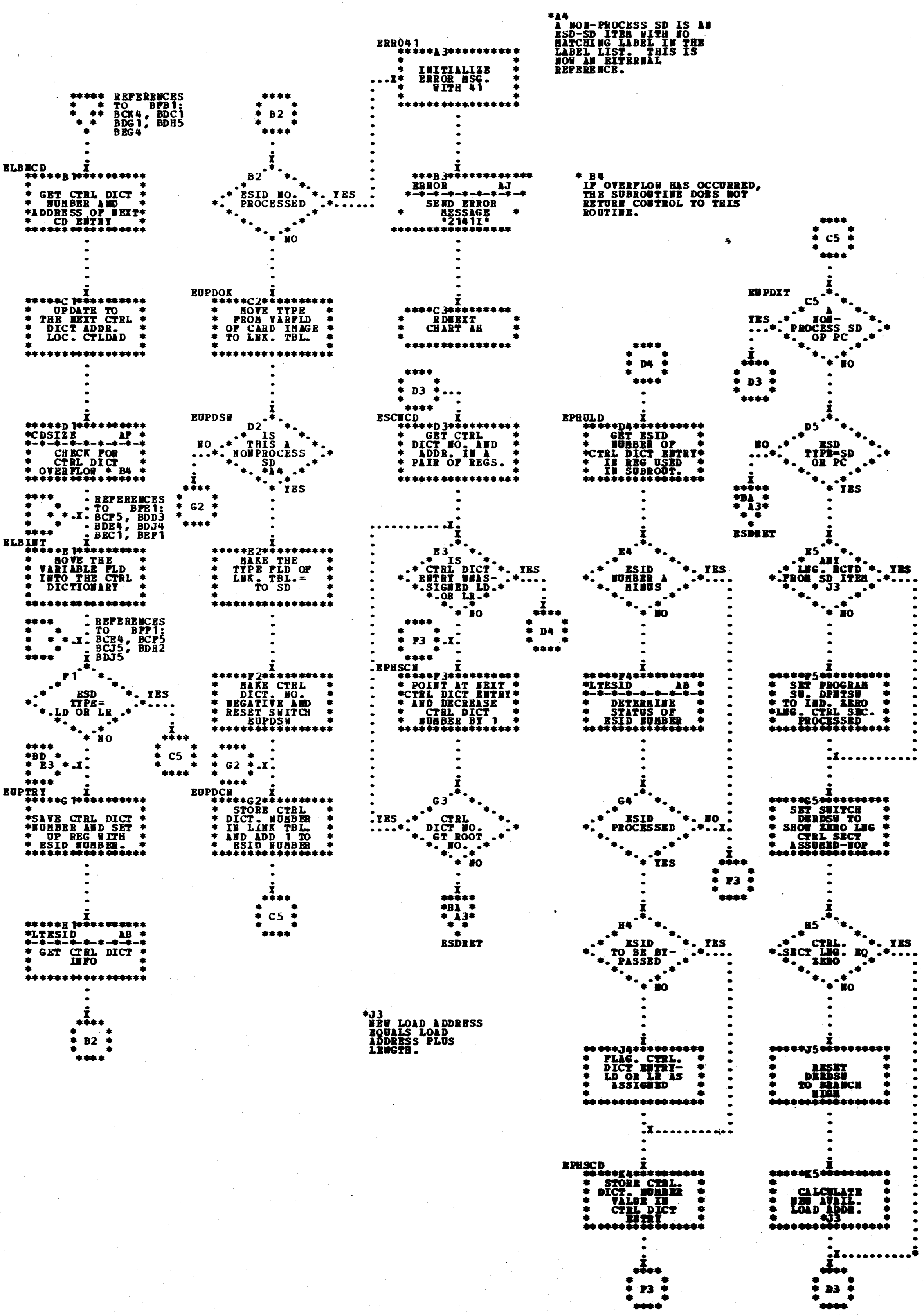
Chart BE. IJBESD - ESD Processor (LD/LR). Refer to Charts 04 and 05


BD C4
ELBLD
A1 PORCE LD IN VARIABLE FIELD TO BE AN LR
B1 COMMON LABEL SAME AS LD LABEL
YES
NO
BD F3
ERR046
C1 CTRL DICT TYPE EQ ER
YES
NO
BF E1 ELBINT
D1 CTRL DICT AS- SIGNED AS SD, LD, OR LR
YES
NO
ELBNLR
D2 RESET VARIABLE FIELD TYPE FROM LR TO LD
E1 TURN ON POSSIBLE DUILICATE ENTRY SWITCH
E2 ASMBLD ORIGINS EQUAL
NO
YES
G5
F1 MAKE VAR TYPE FLD EQUAL TO LR IF NECESSARY
BF E1
ELBINT
F2 CARD IMAGE LD NOT ASSIGNED
YES
NO
F4
G2 CTRL DICT TYPE EQ SD
NO
YES
B3
H2 CTRL DICT NUMBER IN LD EQ SD NUMBER
YES
NO
BA A3
ESDRET
ERR043
J2 INITIALIZE ERROR MESSAGE WITH 43
K2 ERROR AJ SEND ERROR MESSAGE '2143I'
K1 RDNEXT CHART AH
B3
ELBLDR
B3 GET ESID NUMBER OF THE SD THE CTRL DICT LD BELONGS TO
C3 ESID OF CTRL DICT MATCHES ESID OF CD. IMAGE
YES
NO
BA A3
ESDRET
D3 SET UP LNK REG WITH SUBROUTINE RET ADDRESS
E3 LTCDAD AB GET CTRL DICT ADDR OF SD POINTED TO * A4
F3 PUT CTRL DICT ADDR OF SD ITEM IN A WORK REGISTER
G3 PUT THE ESID OF THE SD THE CARD IMAGE BELONGS TO REG FOR S/RTN
H3 SET UP REG INPUT AND LNK REG RET FOR SUBROUTINE
J3 LTCDAD AB GET CTRL DICT ADDRESS OF POINTED TO * C4
K3 SD LABELS MATCH
YES
NO
E4
G5
E4
E4 PUT ADDR OF ORIGINAL CTRL DICT ENTRY -INTO REG 9
F4
ELBNAS
F4 XTPHNO AF GET PHASE NUMBER OF CTRL DICT ENTRY
G4 IS CTRL DICT ENTRY IN CURR. PHASE
NO
YES
BF E1
ELBNCD
H4 TURN ON POSSIBLE DUPLICATE ENTRY SWITCH, DPNTSW
BA A3
ESDRET
G5
ERR043
G5 INITIALIZE ERROR MESSAGE WITH 43
H5 ERROR AJ SEND ERROR MSG '2143I'
J5 RDNEXT CHART AH

*A4
THE LAST 2 BYTES OF THE LD ENTRY IN THE CONTROL DICTIONARY HAVE THE ESID NUMBER OF THE SD. THE EXIT TO THE SUBROUTINE ALLOWS THE CONTROL DICTIONARY ADDRESS OF THE SD TO BE EXTRACTED USING THE ESID NUMBER AS A KEY.

*C4
THIS EXIT TO THE SUBROUTINE PROVIDES THE CONTROL DICTIONARY ADDRESS OF THE SD POINTED TO BY THE ESID NUMBER IN THE CARD IMAGE LD ITEM. SEE *A4.

Chart BF. IJBESD - ESD Processor Update Linkage Table and Control Dictionary
Refer to Charts C4 and 05


REFERENCES TO BFB1: BCK4, BDC1 BDG1, BDH5 BEG4
ELBNCD
B1
GET CTRL DICT NUMBER AND ADDRESS OF NEXT CD ENTRY
C1
UPDATE TO THE NEXT CTRL DICT ADDR. LOC. CTLDAD
D1
CDSIZE AF
CHECK FOR CTRL DICT OVERFLOW * B4
REFERENCES TO BFE1: BCF5, BDD3 BDE4, BDJ4 BEC1, BEF1
ELBINT
E1
MOVE THE VARIABLE FLD INTO THE CTRL DICTIONARY
REFERENCES TO BFF1: BCE4, BCF5 BCJ5, BDH2 BDJ5
F1
ESD TYPE= LD OR LR
YES
NO
C5
BD E3
EUPTRY
G1
SAVE CTRL DICT NUMBER AND SET UP REG WITH ESID NUMBER.
H1
LTESID AB
GET CTRL DICT INFO
B2
B2
B2
ESID NO. PROCESSED
YES
NO
EUPDOK
C2
MOVE TYPE FROM VARFLD OF CARD IMAGE TO LNK. TBL.
EUPDSW
D2
IS THIS A NONPROCESS SD *A4
NO
YES
G2
E2
MAKE THE TYPE FLD OF LNK. TBL.= TO SD
F2
MAKE CTRL DICT. NO. NEGATIVE AND RESET SWITCH EUPDSW
G2
EUPDCN
G2
STORE CTRL DICT. NUMBER IN LINK TBL. AND ADD 1 TO ESID NUMBER
C5
ERR041
A3
INITIALIZE ERROR MSG. WITH 41
B3
ERROR AJ
SEND ERROR MESSAGE '2141I'
C3
RDNEXT CHART AH
D3
ESCNCD
D3
GET CTRL DICT NO. AND ADDR. IN A PAIR OF REGS.
E3
IS CTRL DICT ENTRY UNASSIGNED LD OR LR
YES
NO
D4
F3
EPHSCN
F3
POINT AT NEXT CTRL DICT ENTRY AND DECREASE CTRL DICT NUMBER BY 1
G3
CTRL DICT NO. GT ROOT NO.
YES
NO
BA A3
ESDRET
D4
EPHULD
D4
GET ESID NUMBER OF CTRL DICT ENTRY IN REG USED IN SUBROUT.
E4
ESID NUMBER A MINUS
YES
NO
F4
LTESID AB
DETERMINE STATUS OF ESID NUMBER
G4
ESID PROCESSED
NO
YES
F3
H4
ESID TO BE BY-PASSED
YES
NO
J4
FLAG. CTRL. DICT ENTRY- LD OR LR AS ASSIGNED
EPHSCD
K4
STORE CTRL. DICT. NUMBER VALUE IN CTRL DICT ENTRY
F3
C5
EUPDXT
C5
A NON-PROCESS SD OF PC
YES
NO
D3
D5
ESD TYPE=SD OR PC
NO
YES
DA A3
ESDRET
E5
ANY LNG. RCVD FROM SD ITEM J3
YES
NO
F5
SET PROGRAM SW. DPNTSW TO IND. ZERO LNG. CTRL SEC. PROCESSED
G5
SET SWITCH DERDSW TO SHOW ZERO LNG CTRL SECT ASSUMED-NOP
H5
CTRL. SECT LNG. EQ ZERO
YES
NO
J5
RESET DERDSW TO BRANCH HIGH
K5
CALCULATE NEW AVAIL. LOAD ADDR. *J3
D3

*A4
A NON-PROCESS SD IS AN ESD-SD ITEM WITH NO MATCHING LABEL IN THE LABEL LIST. THIS IS NOW AN EXTERNAL REFERENCE.

* B4
IF OVERFLOW HAS OCCURRED, THE SUBROUTINE DOES NOT RETURN CONTROL TO THIS ROUTINE.

*J3
NEW LOAD ADDRESS EQUALS LOAD ADDRESS PLUS LENGTH.

Chart CA. IJBOTH - Initialization Refer to Chart 06

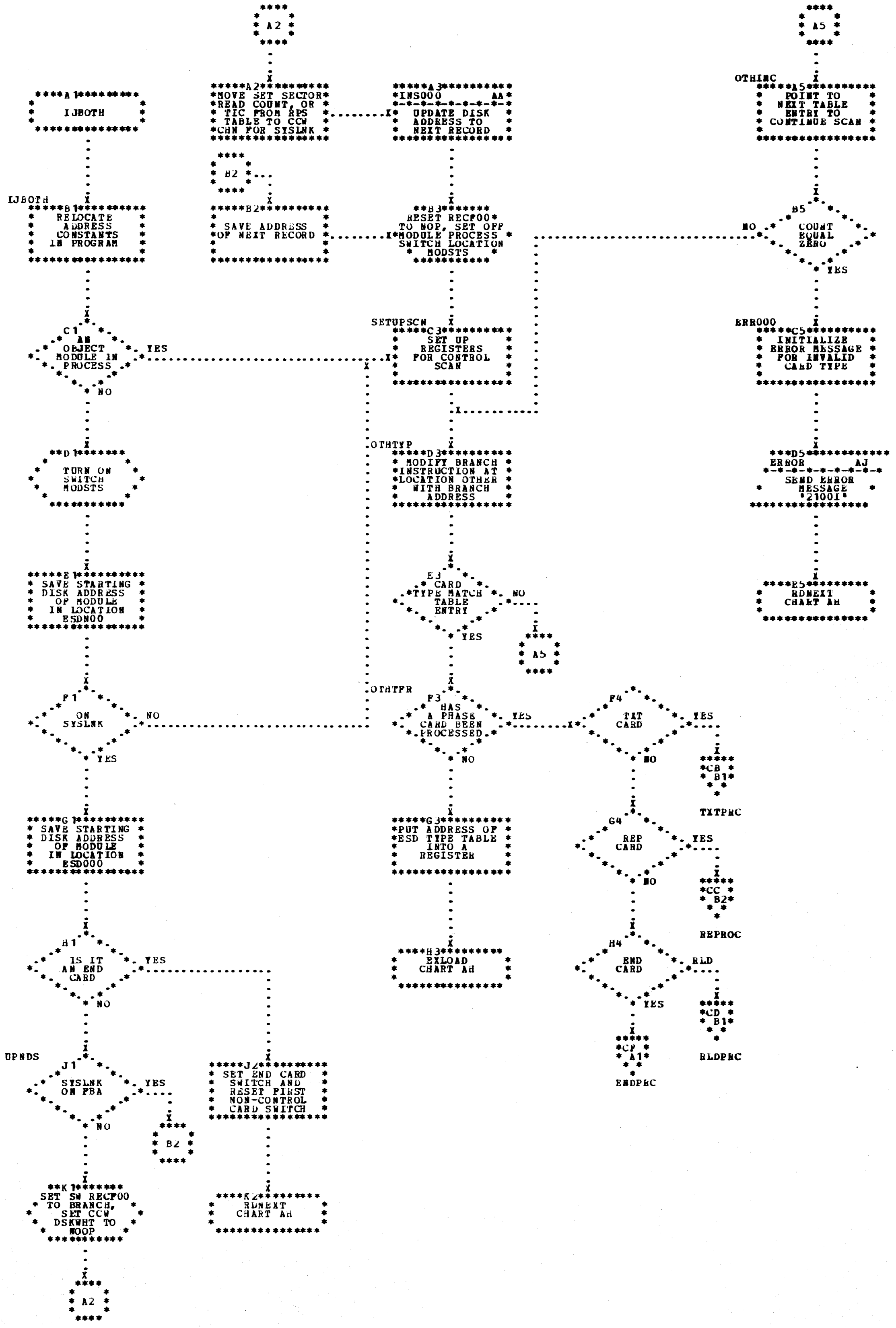
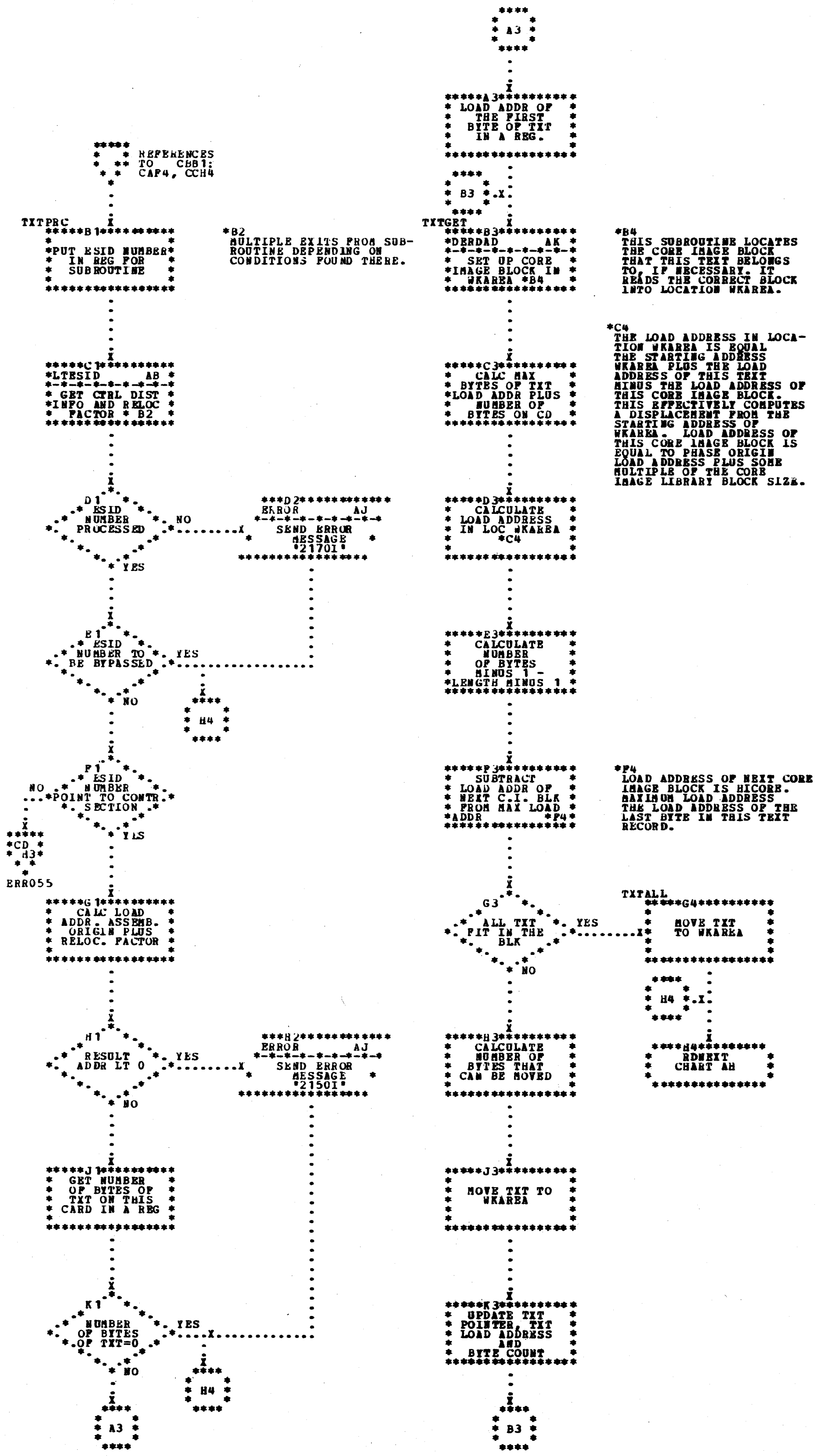

A2
A5
A1
IJBOTH
A2
MOVE SET SECTOR READ COUNT, OR TIC FROM RPS TABLE TO CCW CHN FOR SYSLNK
A3
INS000 AA
UPDATE DISK ADDRESS TO NEXT RECORD
OTHINC
A5
POINT TO NEXT TABLE ENTRY TO CONTINUE SCAN
B2
IJBOTH
B1
RELOCATE ADDRESS CONSTANTS IN PROGRAM
B2
SAVE ADDRESS OF NEXT RECORD
B3
RESET RECP00 TO NOP, SET OFF MODULE PROCESS SWITCH LOCATION MODSTS
B5
COUNT EQUAL ZERO
NO
YES
C1
AN OBJECT MODULE IN PROCESS
YES
NO
SETUPSCN
C3
SET UP REGISTERS FOR CONTROL SCAN
ERR000
C5
INITIALIZE ERROR MESSAGE FOR INVALID CARD TYPE
D1
TURN ON SWITCH MODSTS
OTHTYP
D3
MODIFY BRANCH INSTRUCTION AT LOCATION OTHER WITH BRANCH ADDRESS
D5
ERROR AJ
SEND ERROR MESSAGE '2100I'
E1
SAVE STARTING DISK ADDRESS OF MODULE IN LOCATION ESDN00
E3
CARD TYPE MATCH TABLE ENTRY
NO
YES
A5
E5
RDNEXT CHART AH
F1
ON SYSLNK
NO
YES
OTHTPR
F3
HAS A PHASE CARD BEEN PROCESSED
YES
NO
F4
TXT CARD
YES
NO
CB B1
TXTPRC
G1
SAVE STARTING DISK ADDRESS OF MODULE IN LOCATION ESD000
G3
PUT ADDRESS OF ESD TYPE TABLE INTO A REGISTER
G4
REP CARD
YES
NO
CC B2
REPROC
H1
IS IT AN END CARD
YES
NO
H3
EXLOAD CHART AH
H4
END CARD
RLD
YES
CD B1
CF A1
ENDPRC
RLDPRC
UPNDS
J1
SYSLNK ON FBA
YES
NO
B2
J2
SET END CARD SWITCH AND RESET FIRST NON-CONTROL CARD SWITCH
K1
SET SW RECP00 TO BRANCH, SET CCW DSKWHT TO NOOP
K2
RDNEXT CHART AH
A2

Chart CB. IJBOTH - Text Processor Reter to Chart 06


A3
A3
LOAD ADDR OF THE FIRST BYTE OF TXT IN A REG.
B3
TXTGET
B3
DERDAD AK
SET UP CORE IMAGE BLOCK IN WKAREA *B4
C3
CALC MAX BYTES OF TXT LOAD ADDR PLUS NUMBER OF BYTES ON CD
D3
CALCULATE LOAD ADDRESS IN LOC WKAREA *C4
E3
CALCULATE NUMBER OF BYTES MINUS 1 - LENGTH MINUS 1
F3
SUBTRACT LOAD ADDR OF NEXT C.I. BLK FROM MAX LOAD ADDR *F4
G3
ALL TXT FIT IN THE BLK
YES
NO
TXTALL
G4
MOVE TXT TO WKAREA
H4
H4
RDNEXT CHART AH
H3
CALCULATE NUMBER OF BYTES THAT CAN BE MOVED
J3
MOVE TXT TO WKAREA
K3
UPDATE TXT POINTER, TXT LOAD ADDRESS AND BYTE COUNT
B3
REFERENCES TO CBB1: CAF4, CCH4
TXTPRC
B1
PUT ESID NUMBER IN REG FOR SUBROUTINE
C1
LTESID AB
GET CTRL DIST INFO AND RELOC FACTOR * B2
D1
ESID NUMBER PROCESSED
NO
YES
D2
ERROR AJ
SEND ERROR MESSAGE '21701'
E1
ESID NUMBER TO BE BYPASSED
YES
NO
H4
F1
ESID NUMBER POINT TO CONTR. SECTION
NO
YES
CD H3
ERR055
G1
CALC LOAD ADDR. ASSEMB. ORIGIN PLUS RELOC. FACTOR
H1
RESULT ADDR LT 0
YES
NO
H2
ERROR AJ
SEND ERROR MESSAGE '21501'
J1
GET NUMBER OF BYTES OF TXT ON THIS CARD IN A REG
K1
NUMBER OF BYTES OF TXT=0
YES
NO
H4
A3

*B2
MULTIPLE EXITS FROM SUBROUTINE DEPENDING ON CONDITIONS FOUND THERE.

*B4
THIS SUBROUTINE LOCATES THE CORE IMAGE BLOCK THAT THIS TEXT BELONGS TO, IF NECESSARY. IT READS THE CORRECT BLOCK INTO LOCATION WKAREA.

*C4
THE LOAD ADDRESS IN LOCATION WKAREA IS EQUAL THE STARTING ADDRESS WKAREA PLUS THE LOAD ADDRESS OF THIS TEXT MINUS THE LOAD ADDRESS OF THIS CORE IMAGE BLOCK. THIS EFFECTIVELY COMPUTES A DISPLACEMENT FROM THE STARTING ADDRESS OF WKAREA. LOAD ADDRESS OF THIS CORE IMAGE BLOCK IS EQUAL TO PHASE ORIGIN LOAD ADDRESS PLUS SOME MULTIPLE OF THE CORE IMAGE LIBRARY BLOCK SIZE.

*F4
LOAD ADDRESS OF NEXT CORE IMAGE BLOCK IS HICORE. MAXIMUM LOAD ADDRESS THE LOAD ADDRESS OF THE LAST BYTE IN THIS TEXT RECORD.

Chart CC. IJBOTH - REP Processor. Refer to Chart 06

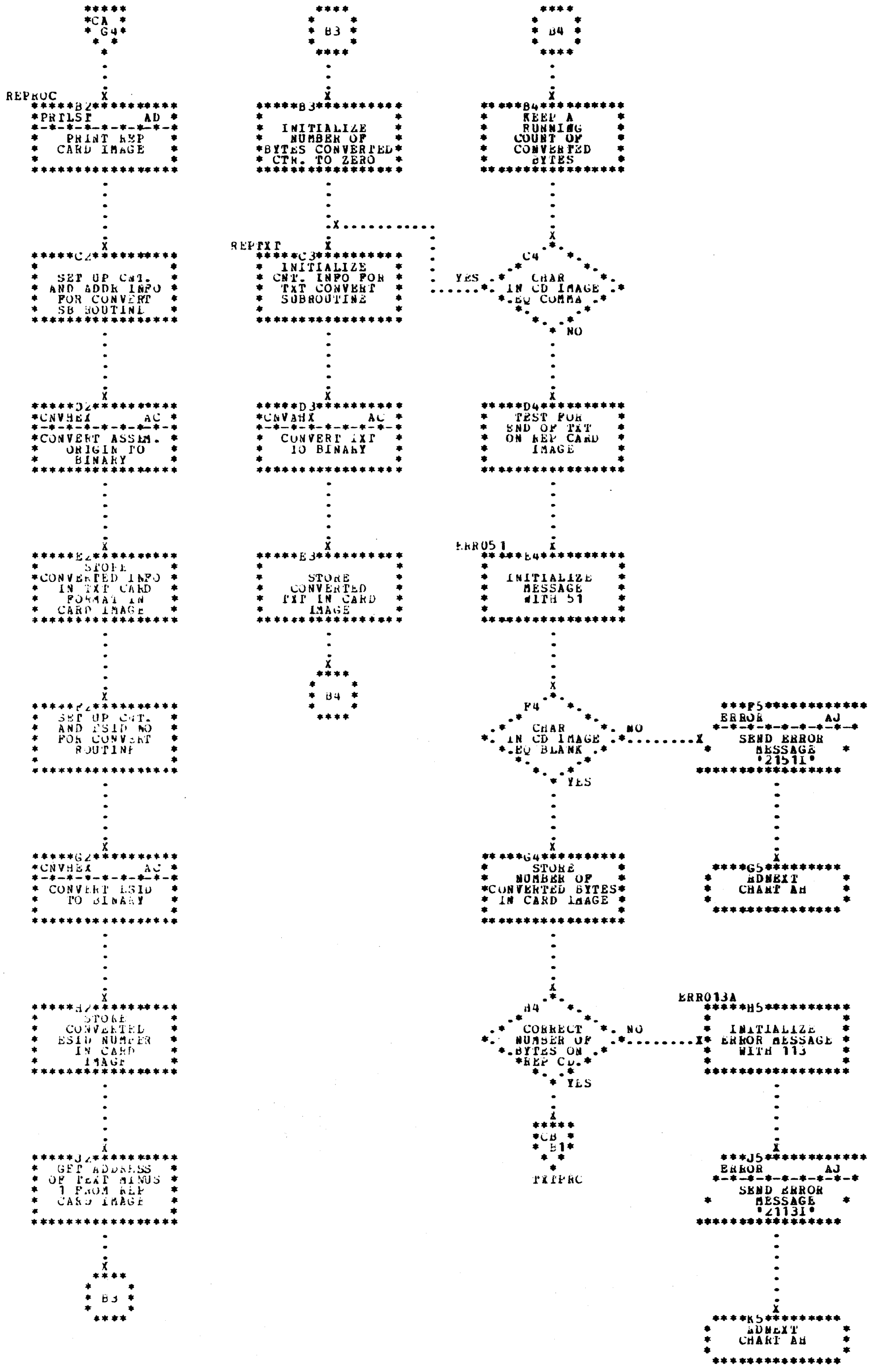
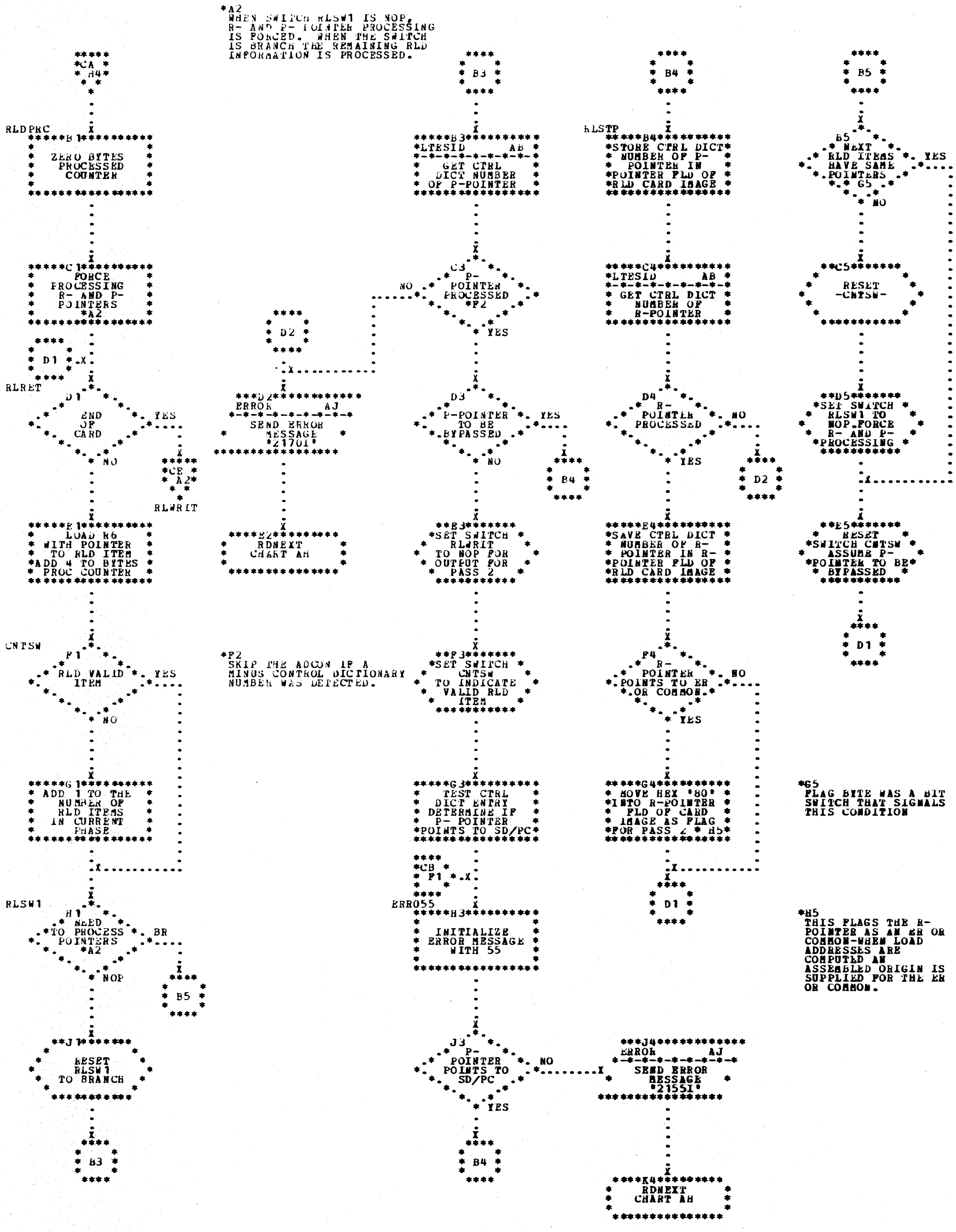

CA G4
REPROC
B2
PRILST AD
PRINT REP CARD IMAGE
C2
SET UP CNT. AND ADDR INFO FOR CONVERT SB ROUTINE
D2
CNVHEX AC
CONVERT ASSM. ORIGIN TO BINARY
E2
STORE CONVERTED INFO IN TXT CARD FORMAT IN CARD IMAGE
F2
SET UP CNT. AND ESID NO FOR CONVERT ROUTINE
G2
CNVHEX AC
CONVERT ESID TO BINARY
H2
STORE CONVERTED ESID NUMBER IN CARD IMAGE
J2
GET ADDRESS OF TEXT MINUS 1 FROM REP CARD IMAGE
B3
B3
INITIALIZE NUMBER OF BYTES CONVERTED CTR. TO ZERO
REPTXT
C3
INITIALIZE CNT. INFO FOR TXT CONVERT SUBROUTINE
D3
CNVAHX AC
CONVERT TXT TO BINARY
E3
STORE CONVERTED TXT IN CARD IMAGE
B4
B4
B4
KEEP A RUNNING COUNT OF CONVERTED BYTES
C4
CHAR IN CD IMAGE EQ COMMA
YES
NO
D4
TEST FOR END OF TXT ON REP CARD IMAGE
ERR051
E4
INITIALIZE MESSAGE WITH 51
F4
CHAR IN CD IMAGE EQ BLANK
NO
YES
F5
ERROR AJ
SEND ERROR MESSAGE '21511'
G5
ADNEXT CHART AH
G4
STORE NUMBER OF CONVERTED BYTES IN CARD IMAGE
H4
CORRECT NUMBER OF BYTES ON REP CD.
NO
YES
ERR013A
H5
INITIALIZE ERROR MESSAGE WITH 113
CB B1
TXTPRC
J5
ERROR AJ
SEND ERROR MESSAGE '21131'
K5
ADNEXT CHART AH

Chart CD. IJBOTH - RLD Pass 1 Processing (Part 1 of 2). Refer to Chart 06


CA H4
RLDPRC
B1 ZERO BYTES PROCESSED COUNTER
C1 FORCE PROCESSING R- AND P- POINTERS *A2
D1
RLRET
D1 END OF CARD
YES
NO
CE A2
RLWRIT
E1 LOAD R6 WITH POINTER TO RLD ITEM ADD 4 TO BYTES PROC COUNTER
CNTSW
F1 RLD VALID ITEM
YES
NO
G1 ADD 1 TO THE NUMBER OF RLD ITEMS IN CURRENT PHASE
RLSW1
H1 NEED TO PROCESS POINTERS *A2
BR
NOP
B5
J1 RESET RLSW1 TO BRANCH
B3
D2
D2 ERROR AJ SEND ERROR MESSAGE '21701'
E2 RDNEXT CHART AH
B3
B3 LTESID AB GET CTRL DICT NUMBER OF P-POINTER
C3 P- POINTER PROCESSED *F2
NO
YES
D3 P-POINTER TO BE BYPASSED
YES
NO
B4
E3 SET SWITCH RLWRIT TO NOP FOR OUTPUT FOR PASS 2
F3 SET SWITCH CNTSW TO INDICATE VALID RLD ITEM
G3 TEST CTRL DICT ENTRY DETERMINE IF P- POINTER POINTS TO SD/PC
CB F1
ERR055
H3 INITIALIZE ERROR MESSAGE WITH 55
J3 P- POINTER POINTS TO SD/PC
NO
YES
B4
B4
RLSTP
B4 STORE CTRL DICT NUMBER OF P- POINTER IN POINTER FLD OF RLD CARD IMAGE
C4 LTESID AB GET CTRL DICT NUMBER OF R-POINTER
D4 R- POINTER PROCESSED
NO
YES
D2
E4 SAVE CTRL DICT NUMBER OF R- POINTER IN R- POINTER FLD OF RLD CARD IMAGE
F4 R- POINTER POINTS TO ER OR COMMON
NO
YES
G4 MOVE HEX '80' INTO R-POINTER FLD OF CARD IMAGE AS FLAG FOR PASS 2 * H5
D1
J4 ERROR AJ SEND ERROR MESSAGE '21551'
K4 RDNEXT CHART AH
B5
B5 NEXT RLD ITEMS HAVE SAME POINTERS * G5
YES
NO
C5 RESET -CNTSW-
D5 SET SWITCH RLSW1 TO NOP.FORCE R- AND P- PROCESSING
E5 RESET SWITCH CNTSW ASSUME P- POINTER TO BE BYPASSED
D1

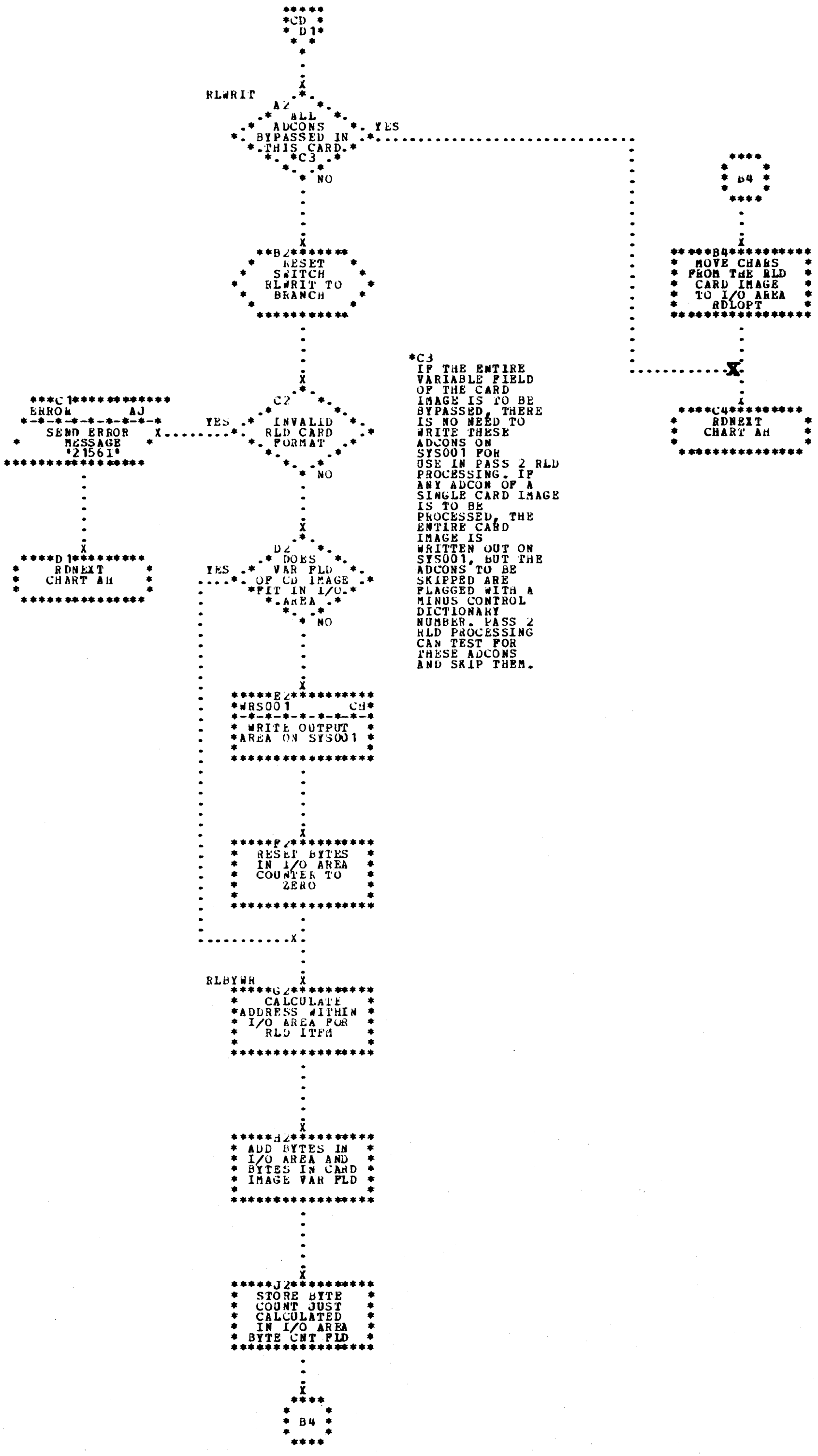
*A2
WHEN SWITCH RLSW1 IS NOP, R- AND P- POINTER PROCESSING IS FORCED. WHEN THE SWITCH IS BRANCH THE REMAINING RLD INFORMATION IS PROCESSED.

*F2
SKIP THE ADCON IF A MINUS CONTROL DICTIONARY NUMBER WAS DETECTED.

*G5
FLAG BYTE WAS A BIT SWITCH THAT SIGNALS THIS CONDITION

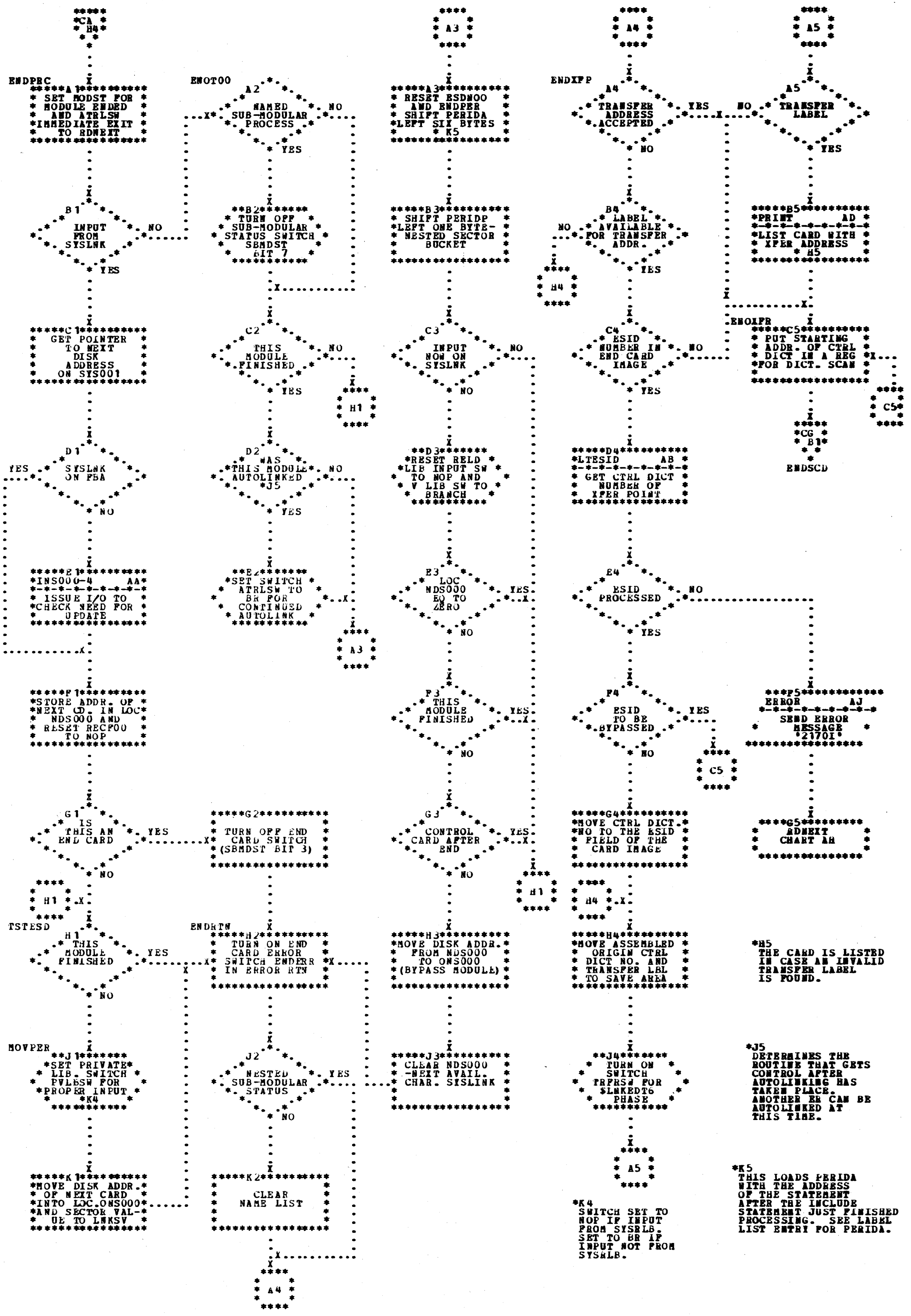
*H5
THIS FLAGS THE R-POINTER AS AN ER OR COMMON-WHEN LOAD ADDRESSES ARE COMPUTED AN ASSEMBLED ORIGIN IS SUPPLIED FOR THE ER OR COMMON.

Chart CE. IJBOTH - RLD Pass 1 Processing (Part 2 of 2). Refer to Chart 06


CD D1
RLWRIT
A2
ALL ADCONS BYPASSED IN THIS CARD
C3
YES
NO
B2
RESET SWITCH RLWRIT TO BRANCH
C2
INVALID RLD CARD FORMAT
YES
NO
C1
ERROR AJ
SEND ERROR MESSAGE '21561'
D1
RDNEXT CHART AH
D2
DOES VAR FLD OF CD IMAGE FIT IN I/O AREA
YES
NO
E2
WRS001 CH
WRITE OUTPUT AREA ON SYS001
F2
RESET BYTES IN I/O AREA COUNTER TO ZERO
RLBYWR
G2
CALCULATE ADDRESS WITHIN I/O AREA FOR RLD ITEM
H2
ADD BYTES IN I/O AREA AND BYTES IN CARD IMAGE VAR FLD
J2
STORE BYTE COUNT JUST CALCULATED IN I/O AREA BYTE CNT FLD
B4
B4
B4
MOVE CHARS FROM THE RLD CARD IMAGE TO I/O AREA RDLOPT
C4
RDNEXT CHART AH

*C3
IF THE ENTIRE VARIABLE FIELD OF THE CARD IMAGE IS TO BE BYPASSED, THERE IS NO NEED TO WRITE THESE ADCONS ON SYS001 FOR USE IN PASS 2 RLD PROCESSING. IF ANY ADCON OF A SINGLE CARD IMAGE IS TO BE PROCESSED, THE ENTIRE CARD IMAGE IS WRITTEN OUT ON SYS001, BUT THE ADCONS TO BE SKIPPED ARE FLAGGED WITH A MINUS CONTROL DICTIONARY NUMBER. PASS 2 RLD PROCESSING CAN TEST FOR THESE ADCONS AND SKIP THEM.

Chart CF. IJBOTH - End Processor (Part 1 of 2). Refer to Chart 06


CA H4
ENDPRC
A1 SET MODST FOR MODULE ENDED AND ATRLSW IMMEDIATE EXIT TO RDNEXT
B1 INPUT FROM SYSLNK
YES
NO
C1 GET POINTER TO NEXT DISK ADDRESS ON SYS001
D1 SYSLNK ON FBA
YES
NO
E1 INS000-4 AA
ISSUE I/O TO CHECK NEED FOR UPDATE
F1 STORE ADDR. OF NEXT CD. IN LOC NDS000 AND RESET RECF00 TO NOP
G1 IS THIS AN END CARD
YES
NO
H1
TSTESD
H1 THIS MODULE FINISHED
YES
NO
MOVPER
J1 SET PRIVATE LIB. SWITCH PVLBSW FOR PROPER INPUT *K4
K1 MOVE DISK ADDR. OF NEXT CARD INTO LOC.ONS000 AND SECTOR VALUE TO LNKSV
ENOT00
A2 NAMED SUB-MODULAR PROCESS
NO
YES
B2 TURN OFF SUB-MODULAR STATUS SWITCH SBMDST BIT 7
C2 THIS MODULE FINISHED
NO
YES
H1
D2 WAS THIS MODULE AUTOLINKED *J5
NO
YES
E2 SET SWITCH ATRLSW TO BR FOR CONTINUED AUTOLINK
A3
G2 TURN OFF END CARD SWITCH (SBMDST BIT 3)
ENDRTN
H2 TURN ON END CARD ERROR SWITCH ENDERR IN ERROR RTN
J2 NESTED SUB-MODULAR STATUS
YES
NO
K2 CLEAR NAME LIST
A4
A3
A3 RESET ESDN00 AND ENDPER SHIFT PERIDA LEFT SIX BYTES * K5
B3 SHIFT PERIDP LEFT ONE BYTE- NESTED SECTOR BUCKET
C3 INPUT NOW ON SYSLNK
NO
NO
D3 RESET RELD LIB INPUT SW TO NOP AND V LIB SW TO BRANCH
E3 LOC NDS000 EQ TO ZERO
YES
NO
F3 THIS MODULE FINISHED
YES
NO
G3 CONTROL CARD AFTER END
YES
NO
H1
H3 MOVE DISK ADDR. FROM NDS000 TO ONS000 (BYPASS MODULE)
J3 CLEAR NDS000 -NEXT AVAIL. CHAR. SYSLINK
A4
ENDXFP
A4 TRANSFER ADDRESS ACCEPTED
YES
NO
B4 LABEL AVAILABLE FOR TRANSFER ADDR.
NO
YES
H4
C4 ESID NUMBER IN END CARD IMAGE
NO
YES
D4 LTESID AB
GET CTRL DICT NUMBER OF XFER POINT
E4 ESID PROCESSED
NO
YES
F4 ESID TO BE BYPASSED
YES
NO
C5
G4 MOVE CTRL DICT. NO TO THE ESID FIELD OF THE CARD IMAGE
H4
H4 MOVE ASSEMBLED ORIGIN CTRL DICT NO. AND TRANSFER LBL TO SAVE AREA
J4 TURN ON SWITCH TRFRSW FOR $LNKEDT6 PHASE
A5
A5
A5 TRANSFER LABEL
NO
YES
B5 PRINT AD
LIST CARD WITH XFER ADDRESS * H5
ENOXFR
C5 PUT STARTING ADDR. OF CTRL DICT IN A REG FOR DICT. SCAN
C5
CG B1
ENDSCD
F5 ERROR AJ
SEND ERROR MESSAGE '2170I'
G5 RDNEXT CHART AH

*H5
THE CARD IS LISTED IN CASE AN INVALID TRANSFER LABEL IS FOUND.

*J5
DETERMINES THE ROUTINE THAT GETS CONTROL AFTER AUTOLINKING HAS TAKEN PLACE. ANOTHER ER CAN BE AUTOLINKED AT THIS TIME.

*K4
SWITCH SET TO NOP IF INPUT FROM SYSRLB. SET TO BR IF INPUT NOT FROM SYSRLB.

*K5
THIS LOADS PERIDA WITH THE ADDRESS OF THE STATEMENT AFTER THE INCLUDE STATEMENT JUST FINISHED PROCESSING. SEE LABEL LIST ENTRY FOR PERIDA.

Chart CG. IJBOTH - End Processor (Part 2 of 2). Refer to Chart 06

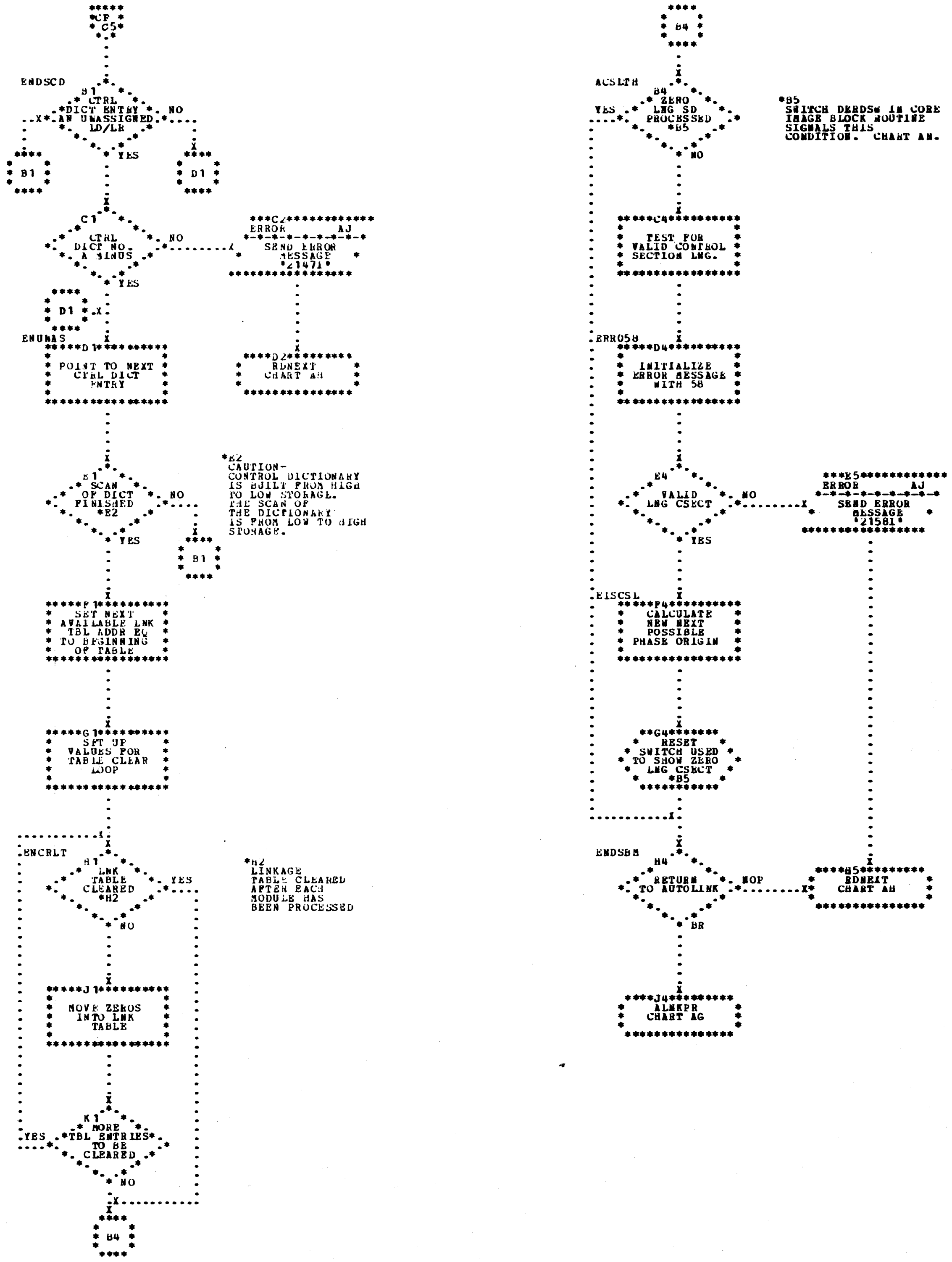

CF C5
ENDSCD
B1 CTRL DICT ENTRY AN UNASSIGNED LD/LR
NO
YES
B1
D1
C1 CTRL DICT NO. A MINUS
NO
YES
C2 ERROR AJ SEND ERROR MESSAGE '21471'
D1
ENUMAS
D1 POINT TO NEXT CTRL DICT ENTRY
D2 RDNEXT CHART AH
E1 SCAN OF DICT FINISHED *E2
NO
YES
B1
*E2 CAUTION- CONTROL DICTIONARY IS BUILT FROM HIGH TO LOW STORAGE. THE SCAN OF THE DICTIONARY IS FROM LOW TO HIGH STORAGE.
F1 SET NEXT AVAILABLE LNK TBL ADDR EQ TO BEGINNING OF TABLE
G1 SET UP VALUES FOR TABLE CLEAR LOOP
ENCRLT
H1 LNK TABLE CLEARED *H2
YES
NO
*H2 LINKAGE TABLE CLEARED AFTER EACH MODULE HAS BEEN PROCESSED
J1 MOVE ZEROS INTO LNK TABLE
K1 MORE TBL ENTRIES TO BE CLEARED
YES
NO
B4
B4
ACSLTH
B4 ZERO LNG SD PROCESSED *B5
YES
NO
*B5 SWITCH DERDSW IN CORE IMAGE BLOCK ROUTINE SIGNALS THIS CONDITION. CHART AN.
C4 TEST FOR VALID CONTROL SECTION LNG.
ERR058
D4 INITIALIZE ERROR MESSAGE WITH 58
E4 VALID LNG CSECT
NO
YES
E5 ERROR AJ SEND ERROR MESSAGE '21581'
EISCSL
F4 CALCULATE NEW NEXT POSSIBLE PHASE ORIGIN
G4 RESET SWITCH USED TO SHOW ZERO LNG CSECT *B5
ENDSBR
H4 RETURN TO AUTOLINK
NOP
BR
H5 RDNEXT CHART AH
J4 ALNKPR CHART AG

Chart CH. IJBOTH - Write SYSC01 Subroutines. Refer to Chart 06

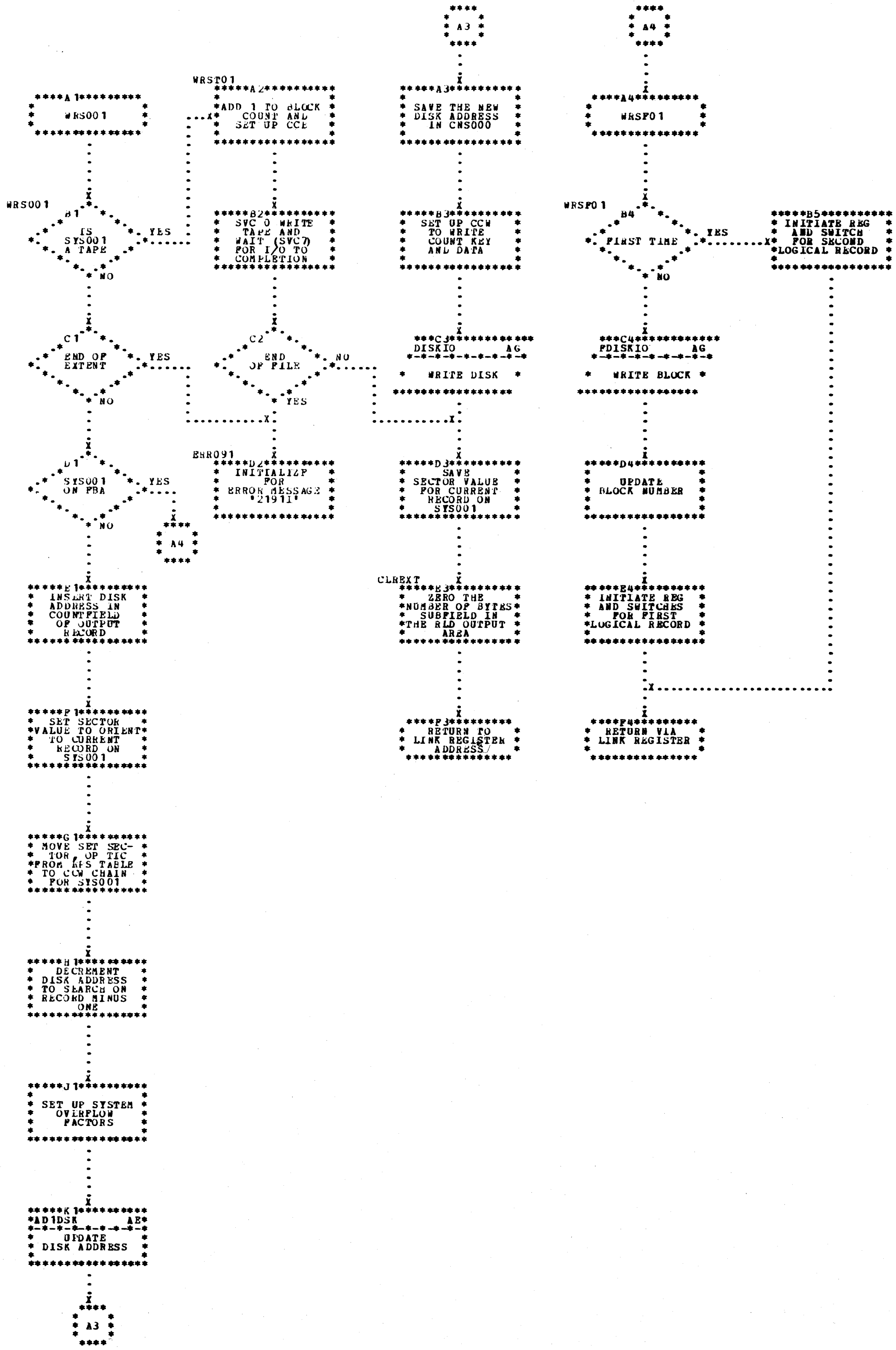

A3
A4
A1
WRS001
WRST01
A2
ADD 1 TO BLOCK COUNT AND SET UP CCE
A3
SAVE THE NEW DISK ADDRESS IN CNS000
A4
WRSF01
WRS001
B1
IS SYS001 A TAPE
YES
NO
B2
SVC 0 WRITE TAPE AND WAIT (SVC7) FOR I/O TO COMPLETION
B3
SET UP CCW TO WRITE COUNT KEY AND DATA
WRSF01
B4
FIRST TIME
YES
NO
B5
INITIATE REG AND SWITCH FOR SECOND LOGICAL RECORD
C1
END OF EXTENT
YES
NO
C2
END OF FILE
NO
YES
C3
DISKIO AG
WRITE DISK
C4
FDISKIO AG
WRITE BLOCK
D1
SYS001 ON FBA
YES
NO
A4
ERR091
D2
INITIALIZE FOR ERROR MESSAGE '2191I'
D3
SAVE SECTOR VALUE FOR CURRENT RECORD ON SYS001
D4
UPDATE BLOCK NUMBER
E1
INSERT DISK ADDRESS IN COUNTFIELD OF OUTPUT RECORD
CLREXT
E3
ZERO THE NUMBER OF BYTES SUBFIELD IN THE RLD OUTPUT AREA
E4
INITIATE REG AND SWITCHES FOR FIRST LOGICAL RECORD
F1
SET SECTOR VALUE TO ORIENT TO CURRENT RECORD ON SYS001
F3
RETURN TO LINK REGISTER ADDRESS
F4
RETURN VIA LINK REGISTER
G1
MOVE SET SECTOR, OP TIC FROM LFS TABLE TO CCW CHAIN FOR SYS001
H1
DECREMENT DISK ADDRESS TO SEARCH ON RECORD MINUS ONE
J1
SET UP SYSTEM OVERFLOW FACTORS
K1
AD1DSK AE
UPDATE DISK ADDRESS
A3

Chart DA. IJBSCN - Initialize Control Card Processor. Refer to Chart 07

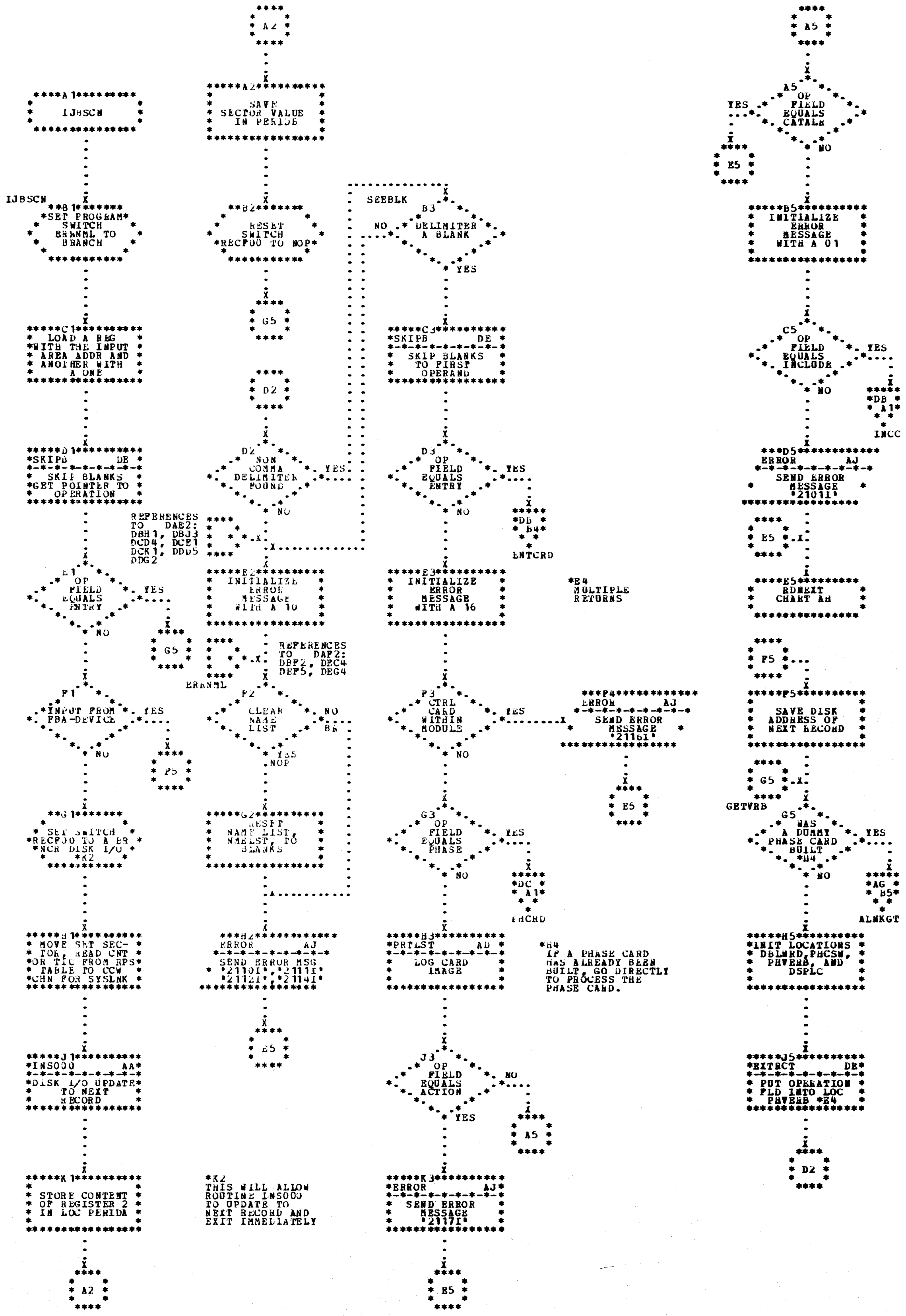

A2
A1
IJBSCN
A2
SAVE SECTOR VALUE IN PERIDE
IJBSCN
B1
SET PROGRAM SWITCH BRNML TO BRANCH
B2
RESET SWITCH RECP00 TO NOP
G5
C1
LOAD A REG WITH THE INPUT AREA ADDR AND ANOTHER WITH A ONE
D2
D1
SKIPB DE
SKIP BLANKS GET POINTER TO OPERATION
D2
NON COMMA DELIMITER FOUND
YES
NO
REFERENCES TO DAE2: DBH1, DBJ3 DCD4, DCE1 DCK1, DDD5 DDG2
E1
OP FIELD EQUALS ENTRY
YES
NO
G5
E2
INITIALIZE ERROR MESSAGE WITH A 10
REFERENCES TO DAF2: DBF2, DEC4 DEF5, DEG4
BRNML
F1
INPUT FROM FBA-DEVICE
YES
NO
F5
F2
CLEAR NAME LIST
NO
BR
YES
NOP
G1
SET SWITCH RECP00 TO A BR NCH DISK I/O *K2
G2
RESET NAME LIST, NMELST, TO BLANKS
H1
MOVE SET SECTOR, READ CNT OR TIC FROM RPS TABLE TO CCW CHN FOR SYSLNK
H2
ERROR AJ
SEND ERROR MSG '2110I','2111I' '2112I','2114I'
E5
J1
INS000 AA
DISK I/O UPDATE TO NEXT RECORD
K1
STORE CONTENT OF REGISTER 2 IN LOC PERIDA
A2
*K2
THIS WILL ALLOW ROUTINE INS000 TO UPDATE TO NEXT RECORD AND EXIT IMMEDIATELY
SEEBLK
B3
DELIMITER A BLANK
NO
YES
C3
SKIPB DE
SKIP BLANKS TO FIRST OPERAND
D3
OP FIELD EQUALS ENTRY
YES
NO
DB B4
ENTCRD
E3
INITIALIZE ERROR MESSAGE WITH A 16
*E4
MULTIPLE RETURNS
F3
CTRL CARD WITHIN MODULE
YES
NO
F4
ERROR AJ
SEND ERROR MESSAGE '2116I'
E5
G3
OP FIELD EQUALS PHASE
YES
NO
DC A1
PHCRD
H3
PRTLST AD
LOG CARD IMAGE
*H4
IF A PHASE CARD HAS ALREADY BEEN BUILT, GO DIRECTLY TO PROCESS THE PHASE CARD.
J3
OP FIELD EQUALS ACTION
NO
YES
A5
K3
ERROR AJ
SEND ERROR MESSAGE '2117I'
E5
A5
A5
OP FIELD EQUALS CATALR
YES
NO
E5
B5
INITIALIZE ERROR MESSAGE WITH A 01
C5
OP FIELD EQUALS INCLUDE
YES
NO
DB A1
INCC
D5
ERROR AJ
SEND ERROR MESSAGE '2101I'
E5
E5
RDNEXT CHART AH
F5
F5
SAVE DISK ADDRESS OF NEXT RECORD
G5
GETVRB
G5
WAS A DUMMY PHASE CARD BUILT *H4
YES
NO
AG B5
ALNKGT
H5
INIT LOCATIONS DBLWRD, PHCSW, PHVERB, AND DSPLC
J5
EXTRCT DE
PUT OPERATION FLD INTO LOC PHVERB *E4
D2

Chart DB. IJBSCN - INCLUDE Card Processor and Entry Card Processor, Refer to Chart 07

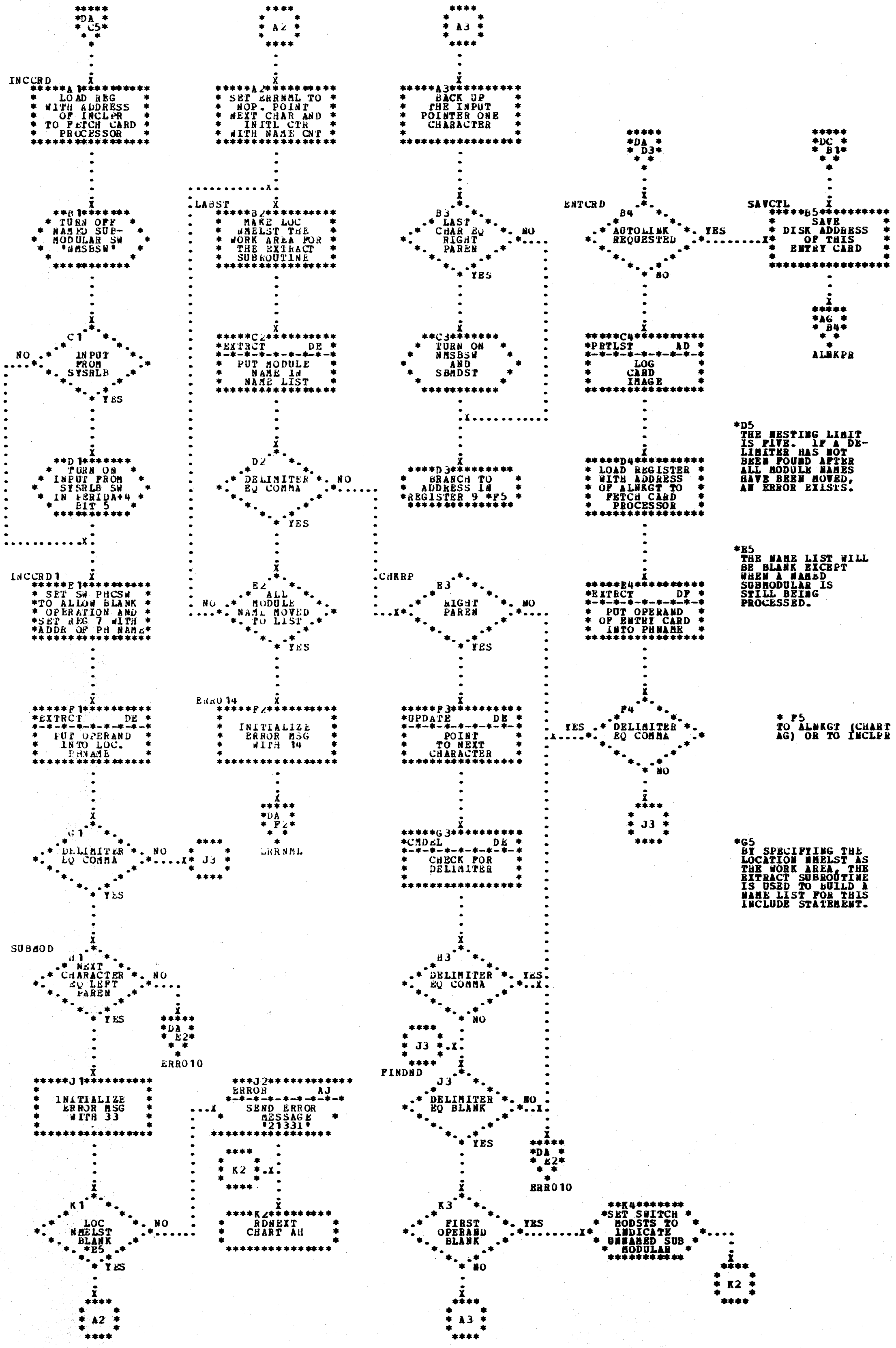

DA C5
INCCRD A1 LOAD REG WITH ADDRESS OF INCLPR TO FETCH CARD PROCESSOR
B1 TURN OFF NAMED SUB-MODULAR SW 'NMSBSW'
C1 INPUT FROM SYSRLB NO YES
D1 TURN ON INPUT FROM SYSRLB SW IN FERIDA+4 BIT 5
INCCRD1 E1 SET SW PHCSW TO ALLOW BLANK OPERATION AND SET REG 7 WITH ADDR OF PH NAME
F1 EXTRCT DE PUT OPERAND INTO LOC. PHNAME
G1 DELIMITER EQ COMMA NO YES
J3
SUBMOD H1 NEXT CHARACTER EQ LEFT PAREN NO YES
DA E2 ERRO10
J1 INITIALIZE ERROR MSG WITH 33
K1 LOC NMELST BLANK *B5 NO YES
A2
A2
A2 SET ERRNML TO NOP. POINT NEXT CHAR AND INITL CTR WITH NAME CNT
LABST B2 MAKE LOC NMELST THE WORK AREA FOR THE EXTRACT SUBROUTINE
C2 EXTRCT DE PUT MODULE NAME IN NAME LIST
D2 DELIMITER EQ COMMA NO YES
E2 ALL MODULE NAME MOVED TO LIST NO YES
ERRO14 F2 INITIALIZE ERROR MSG WITH 14
DA F2 ERRNML
J2 ERROR AJ SEND ERROR MESSAGE '21331'
K2
K2 RDNEXT CHART AH
A3
A3 BACK UP THE INPUT POINTER ONE CHARACTER
B3 LAST CHAR EQ RIGHT PAREN NO YES
C3 TURN ON NMSBSW AND SBMDST
D3 BRANCH TO ADDRESS IN REGISTER 9 *F5
CHKRP E3 RIGHT PAREN NO YES
F3 UPDATE DE POINT TO NEXT CHARACTER
G3 CMDEL DE CHECK FOR DELIMITER
H3 DELIMITER EQ COMMA YES NO
J3
FINDND J3 DELIMITER EQ BLANK NO YES
DA E2 ERRO10
K3 FIRST OPERAND BLANK YES NO
K4 SET SWITCH MODSTS TO INDICATE UNNAMED SUB MODULAR
K2
A3
DA D3
ENTCRD B4 AUTOLINK REQUESTED YES NO
C4 PRTLST AD LOG CARD IMAGE
D4 LOAD REGISTER WITH ADDRESS OF ALNKGT TO FETCH CARD PROCESSOR
E4 EXTRCT DF PUT OPERAND OF ENTRY CARD INTO PHNAME
F4 DELIMITER EQ COMMA YES NO
J3
DC B1
SAVCTL B5 SAVE DISK ADDRESS OF THIS ENTRY CARD
AG B4 ALNKPR
*D5 THE NESTING LIMIT IS FIVE. IF A DELIMITER HAS NOT BEEN FOUND AFTER ALL MODULE NAMES HAVE BEEN MOVED, AN ERROR EXISTS.
*E5 THE NAME LIST WILL BE BLANK EXCEPT WHEN A NAMED SUBMODULAR IS STILL BEING PROCESSED.
* F5 TO ALNKGT (CHART AG) OR TO INCLPR
*G5 BY SPECIFYING THE LOCATION NMELST AS THE WORK AREA, THE EXTRACT SUBROUTINE IS USED TO BUILD A NAME LIST FOR THIS INCLUDE STATEMENT.

Chart DC. IJBSCN - Phase Card Processor (Part 1 of 2). Refer to Chart 07

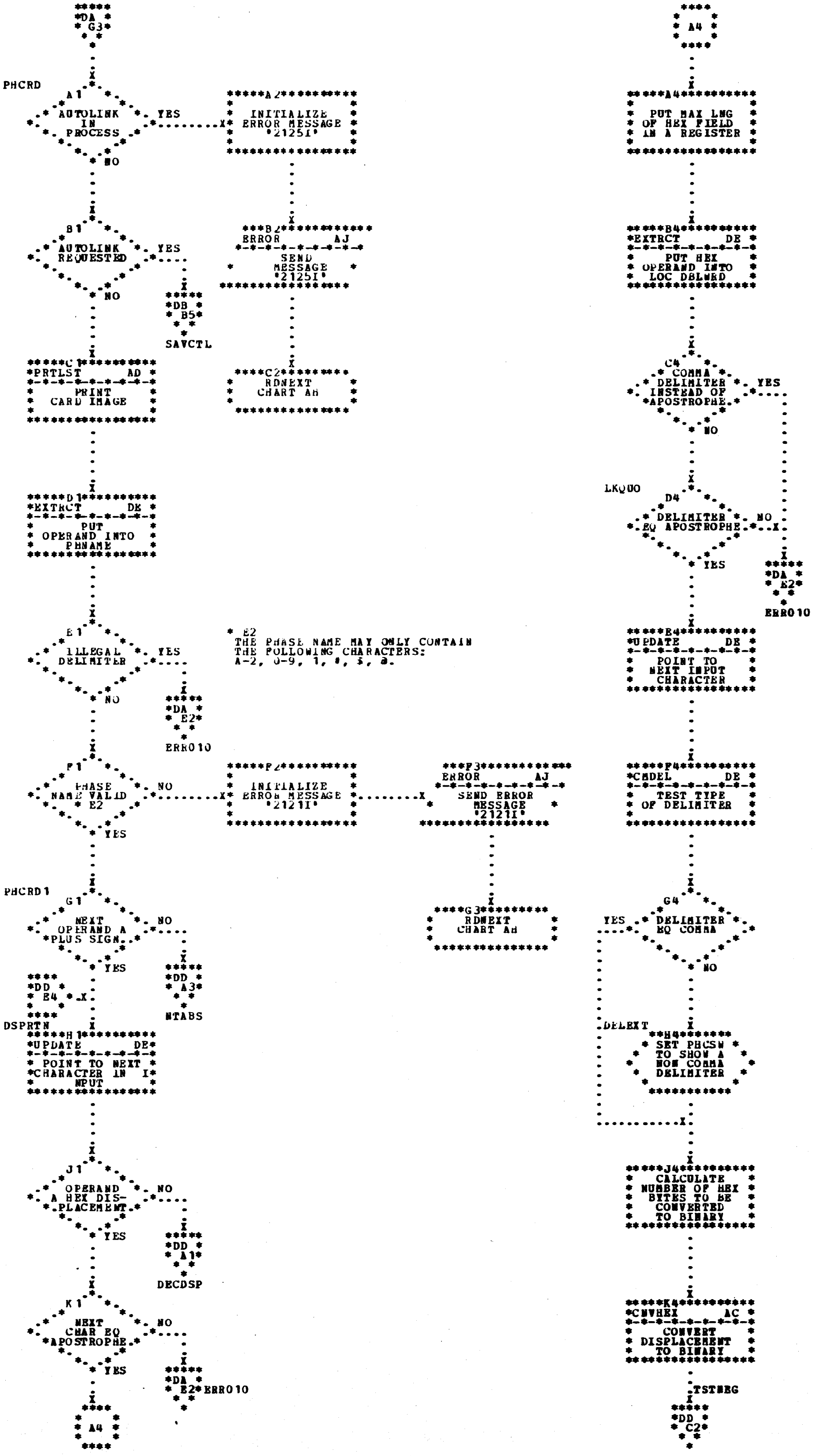


Chart DD. IJBSCN - Phase Card Processcr (Part 2 of 2). Refer to Chart 07

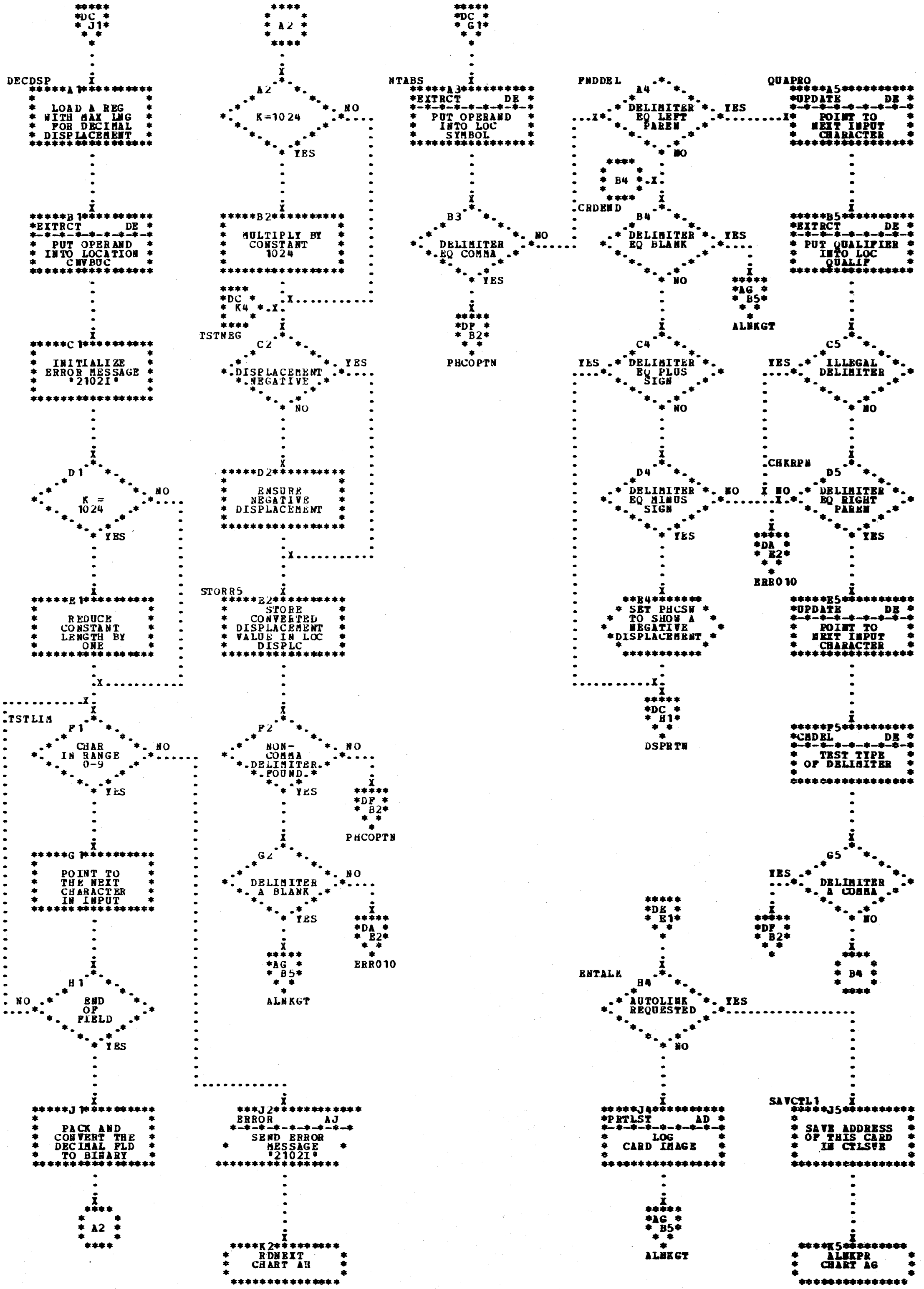

DC J1
DECDSP
A1 LOAD A REG WITH MAX LNG FOR DECIMAL DISPLACEMENT
B1 EXTRCT DE PUT OPERAND INTO LOCATION CNVBUC
C1 INITIALIZE ERROR MESSAGE '2102I'
D1 K = 1024
NO
YES
E1 REDUCE CONSTANT LENGTH BY ONE
TSTLIM
F1 CHAR IN RANGE 0-9
G1 POINT TO THE NEXT CHARACTER IN INPUT
H1 END OF FIELD
J1 PACK AND CONVERT THE DECIMAL FLD TO BINARY
A2
A2 K=1024
B2 MULTIPLY BY CONSTANT 1024
DC K4
TSTNEG
C2 DISPLACEMENT NEGATIVE
D2 ENSURE NEGATIVE DISPLACEMENT
STORR5
E2 STORE CONVERTED DISPLACEMENT VALUE IN LOC DISPLC
F2 NON-COMMA DELIMITER FOUND
DF B2 PHCOPTN
G2 DELIMITER A BLANK
AG B5 ALNKGT
DA E2 ERR010
J2 ERROR AJ SEND ERROR MESSAGE '2102I'
K2 RDNEXT CHART AH
DC G1
NTABS
A3 EXTRCT DE PUT OPERAND INTO LOC SYMBOL
B3 DELIMITER EQ COMMA
DF B2 PHCOPTN
FNDDEL
A4 DELIMITER EQ LEFT PAREN
B4
CHDEND
B4 DELIMITER EQ BLANK
AG B5 ALNKGT
C4 DELIMITER EQ PLUS SIGN
D4 DELIMITER EQ MINUS SIGN
E4 SET PHCSW TO SHOW A NEGATIVE DISPLACEMENT
DC H1 DSPRTN
DE E1
ENTALK
H4 AUTOLINK REQUESTED
J4 PRTLST AD LOG CARD IMAGE
AG B5 ALNKGT
QUAPRO
A5 UPDATE DE POINT TO NEXT INPUT CHARACTER
B5 EXTRCT DE PUT QUALIFIER INTO LOC QUALIF
C5 ILLEGAL DELIMITER
CHKRPN
D5 DELIMITER EQ RIGHT PAREN
DA E2 ERR010
E5 UPDATE DE POINT TO NEXT INPUT CHARACTER
F5 CHDEL DE TEST TYPE OF DELIMITER
G5 DELIMITER A COMMA
DF B2
B4
SAVCTL1
J5 SAVE ADDRESS OF THIS CARD IN CTLSVE
K5 ALNKPR CHART AG

Chart DE. IJBSCN - Skip Blanks and Extract Field Subroutines (Part 1 of 2) Refer to Chart 07

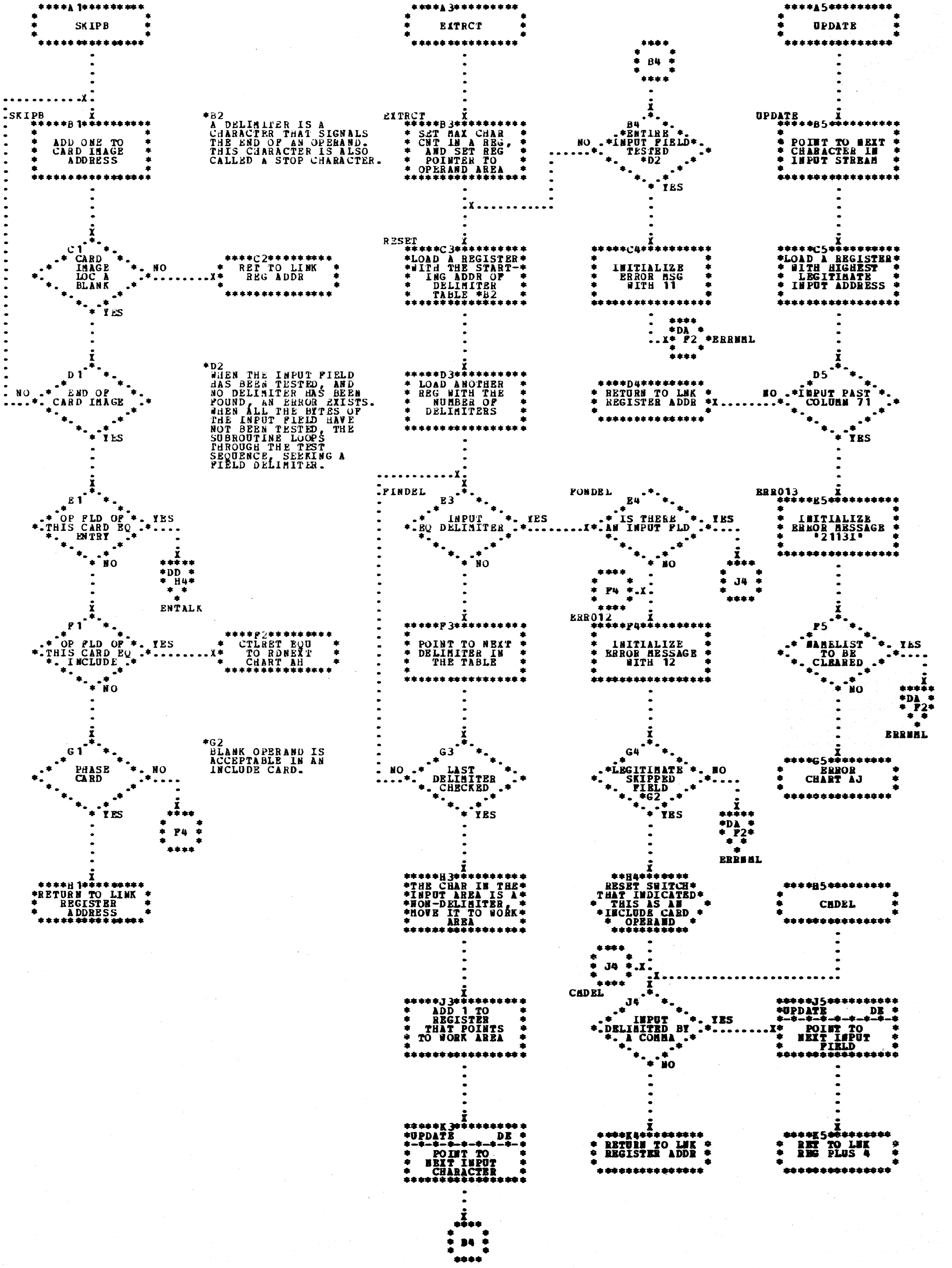

A1 SKIPB
SKIPB
B1 ADD ONE TO CARD IMAGE ADDRESS
C1 CARD IMAGE LOC A BLANK
NO
YES
C2 REF TO LINK REG ADDR
D1 END OF CARD IMAGE
NO
YES
E1 OP FLD OF THIS CARD EQ ENTRY
YES
NO
DD H4
ENTALK
F1 OP FLD OF THIS CARD EQ INCLUDE
YES
NO
F2 CTLRET EQU TO RDNEXT CHART AH
G1 PHASE CARD
NO
YES
F4
H1 RETURN TO LINK REGISTER ADDRESS
*B2 A DELIMITER IS A CHARACTER THAT SIGNALS THE END OF AN OPERAND. THIS CHARACTER IS ALSO CALLED A STOP CHARACTER.
*D2 WHEN THE INPUT FIELD HAS BEEN TESTED, AND NO DELIMITER HAS BEEN FOUND, AN ERROR EXISTS. WHEN ALL THE BYTES OF THE INPUT FIELD HAVE NOT BEEN TESTED, THE SUBROUTINE LOOPS THROUGH THE TEST SEQUENCE, SEEKING A FIELD DELIMITER.
*G2 BLANK OPERAND IS ACCEPTABLE IN AN INCLUDE CARD.
A3 EXTRCT
EXTRCT
B3 SET MAX CHAR CNT IN A REG, AND SET REG POINTER TO OPERAND AREA
RESET
C3 LOAD A REGISTER WITH THE START-ING ADDR OF DELIMITER TABLE *B2
D3 LOAD ANOTHER REG WITH THE NUMBER OF DELIMITERS
FINDEL
E3 INPUT EQ DELIMITER
YES
NO
F3 POINT TO NEXT DELIMITER IN THE TABLE
G3 LAST DELIMITER CHECKED
NO
YES
H3 THE CHAR IN THE INPUT AREA IS A NON-DELIMITER, MOVE IT TO WORK AREA
J3 ADD 1 TO REGISTER THAT POINTS TO WORK AREA
K3 UPDATE DE POINT TO NEXT INPUT CHARACTER
B4
B4
B4 ENTIRE INPUT FIELD TESTED *D2
NO
YES
C4 INITIALIZE ERROR MSG WITH 11
DA F2 ERRNML
D4 RETURN TO LNK REGISTER ADDR
FONDEL
E4 IS THERE AN INPUT FLD
YES
NO
F4
J4
ERR012
F4 INITIALIZE ERROR MESSAGE WITH 12
G4 LEGITIMATE SKIPPED FIELD *G2
NO
YES
DA F2
ERRNML
H4 RESET SWITCH THAT INDICATED THIS AS AN INCLUDE CARD OPERAND
J4
CMDEL
J4 INPUT DELIMITED BY A COMMA
YES
NO
K4 RETURN TO LNK REGISTER ADDR
A5 UPDATE
UPDATE
B5 POINT TO NEXT CHARACTER IN INPUT STREAM
C5 LOAD A REGISTER WITH HIGHEST LEGITIMATE INPUT ADDRESS
D5 INPUT PAST COLUMN 71
NO
YES
ERR013
E5 INITIALIZE ERROR MESSAGE '21131'
F5 NAMELIST TO BE CLEARED
YES
NO
DA F2
ERRNML
G5 ERROR CHART AJ
H5 CMDEL
J5 UPDATE DE POINT TO NEXT INPUT FIELD
K5 RET TO LNK REG PLUS 4

Chart DF. IJBSCN - Skip Blanks and Extract Field Subroutines (Part 2 of 2)
Refer to Chart 07

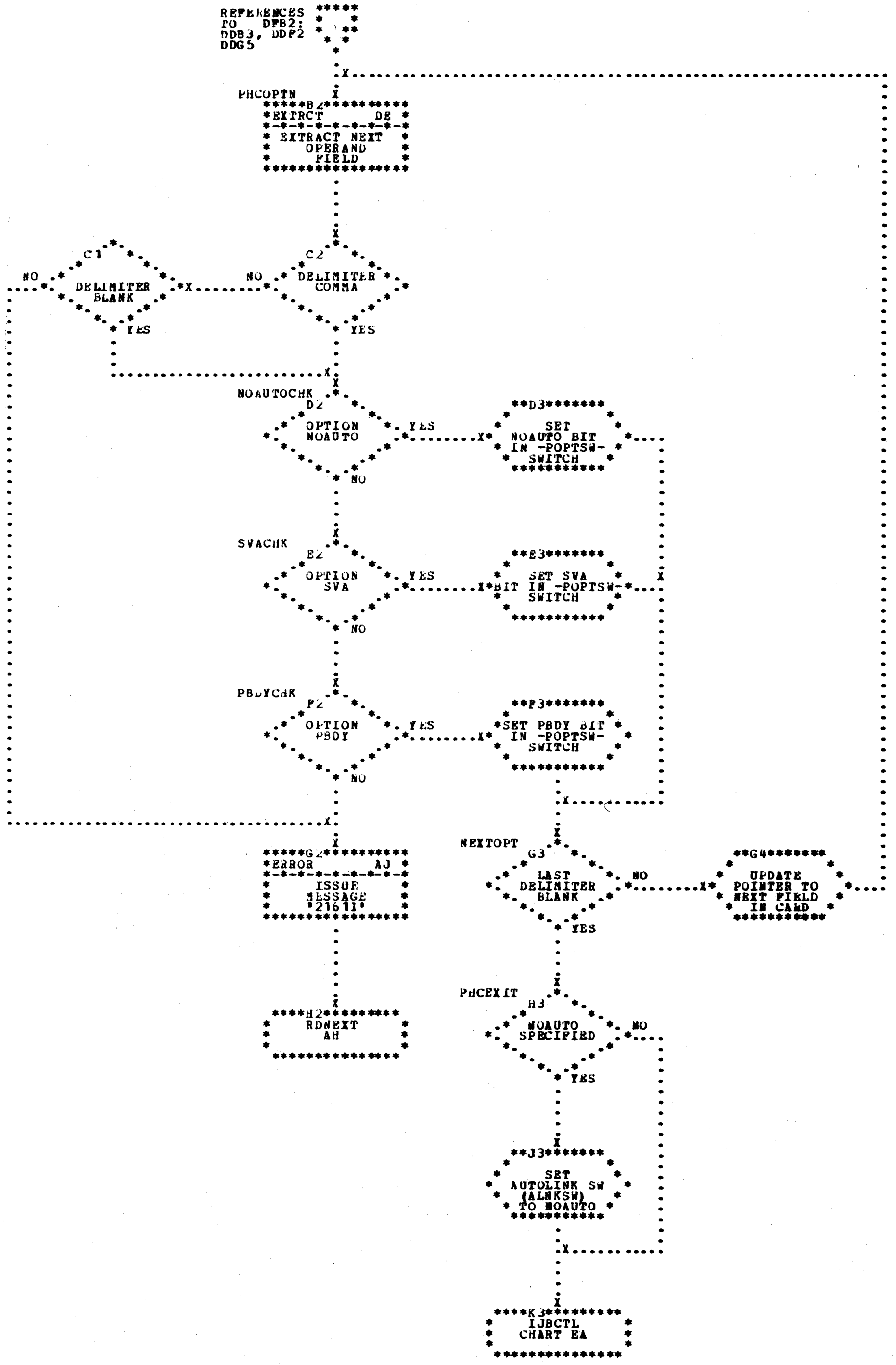
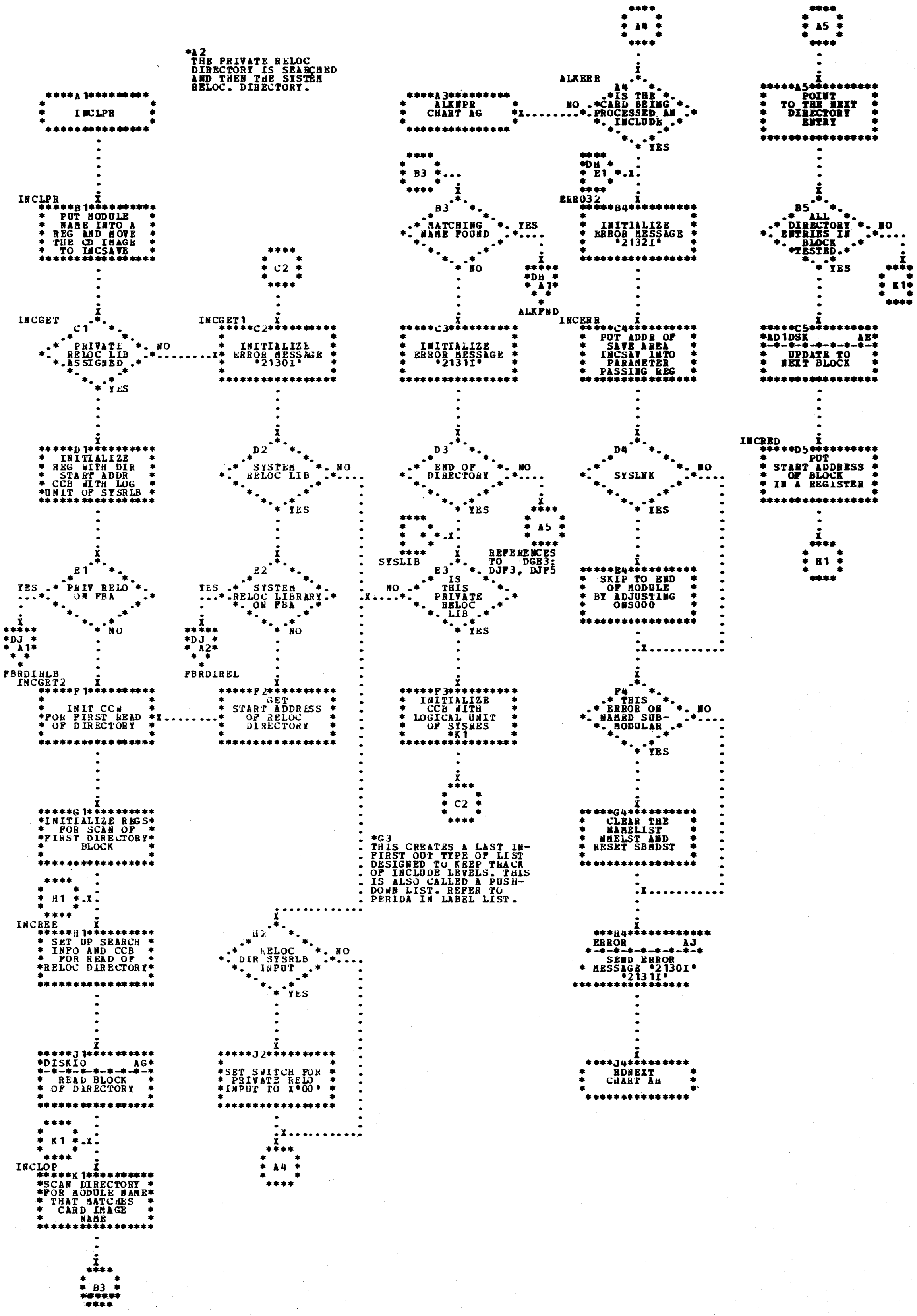


Chart DG. IJBSCN - Phase Card Option Frocessor (Part 1 ot 2). Refer to Chart 07


A1
INCLPR
INCLPR
B1
PUT MODULE NAME INTO A REG AND MOVE THE CD IMAGE TO INCSAVE
INCGET
C1
PRIVATE RELOC LIB ASSIGNED
NO
YES
D1
INITIALIZE REG WITH DIR START ADDR CCB WITH LOG UNIT OF SYSRLB
E1
PRIV RELO ON FBA
YES
NO
DJ A1
FBRDIRLB
INCGET2
F1
INIT CCW FOR FIRST READ OF DIRECTORY
G1
INITIALIZE REGS FOR SCAN OF FIRST DIRECTORY BLOCK
H1
INCREE
H1
SET UP SEARCH INFO AND CCB FOR READ OF RELOC DIRECTORY
J1
DISKIO AG
READ BLOCK OF DIRECTORY
K1
INCLOP
K1
SCAN DIRECTORY FOR MODULE NAME THAT MATCHES CARD IMAGE NAME
B3
C2
INCGET1
C2
INITIALIZE ERROR MESSAGE '21301'
D2
SYSTEM RELOC LIB
NO
YES
E2
SYSTEM RELOC LIBRARY ON FBA
YES
NO
DJ A2
FBRDIREL
F2
GET START ADDRESS OF RELOC DIRECTORY
H2
RELOC DIR SYSRLB INPUT
NO
YES
J2
SET SWITCH FOR PRIVATE RELO INPUT TO X'00'
A4
A3
ALKNPR CHART AG
B3
B3
MATCHING NAME FOUND
YES
NO
DH A1
ALKFND
C3
INITIALIZE ERROR MESSAGE '21311'
D3
END OF DIRECTORY
NO
YES
A5
SYSLIB
E3
IS THIS PRIVATE RELOC LIB
NO
YES
REFERENCES TO DGB3: DJF3, DJF5
F3
INITIALIZE CCB WITH LOGICAL UNIT OF SYSRES
K1
C2
A4
ALKERR
A4
IS THE CARD BEING PROCESSED AN INCLUDE
NO
YES
DH E1
ERR032
B4
INITIALIZE ERROR MESSAGE '21321'
INCERR
C4
PUT ADDR OF SAVE AREA INCSAV INTO PARAMETER PASSING REG
D4
SYSLNK
NO
YES
E4
SKIP TO END OF MODULE BY ADJUSTING OMS000
F4
THIS ERROR ON NAMED SUB-MODULAR
NO
YES
G4
CLEAR THE NAMELIST NMELST AND RESET SBMDST
H4
ERROR AJ
SEND ERROR MESSAGE '21301' '21311'
J4
RDNEXT CHART AH
A5
A5
POINT TO THE NEXT DIRECTORY ENTRY
B5
ALL DIRECTORY ENTRIES IN BLOCK TESTED
NO
YES
K1
C5
AD1DSK AE
UPDATE TO NEXT BLOCK
INCRED
D5
PUT START ADDRESS OF BLOCK IN A REGISTER
H1

*A2
THE PRIVATE RELOC DIRECTORY IS SEARCHED AND THEN THE SYSTEM RELOC. DIRECTORY.

*G3
THIS CREATES A LAST IN-FIRST OUT TYPE OF LIST DESIGNED TO KEEP TRACK OF INCLUDE LEVELS. THIS IS ALSO CALLED A PUSH-DOWN LIST. REFER TO PERIDA IN LABEL LIST.

Chart DH. IJBSCN - Phase Card Option Frocessor (Fart 2 of 2). Refer to Chart 07

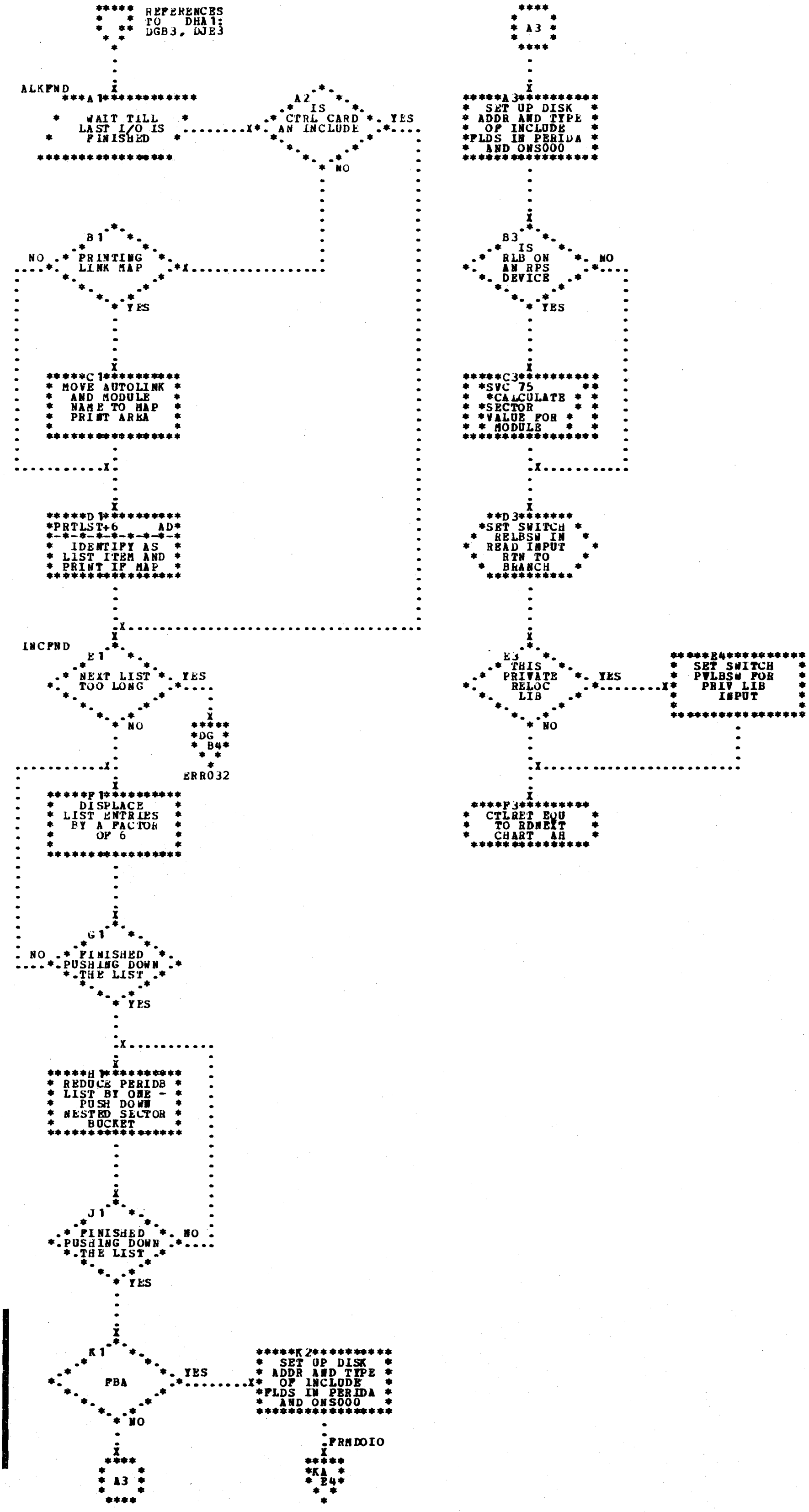

REFERENCES TO DHA1: DGB3, DJE3
ALKFND
A1
WAIT TILL LAST I/O IS FINISHED
A2
IS CTRL CARD AN INCLUDE
YES
NO
B1
PRINTING LINK MAP
NO
YES
C1
MOVE AUTOLINK AND MODULE NAME TO MAP PRINT AREA
D1
PRTLST+6 AD
IDENTIFY AS LIST ITEM AND PRINT IF MAP
INCFND
E1
NEXT LIST TOO LONG
YES
NO
DG B4
ERR032
F1
DISPLACE LIST ENTRIES BY A FACTOR OF 6
G1
FINISHED PUSHING DOWN THE LIST
NO
YES
H1
REDUCE PERIDB LIST BY ONE - PUSH DOWN NESTED SECTOR BUCKET
J1
FINISHED PUSHING DOWN THE LIST
NO
YES
K1
FBA
YES
NO
K2
SET UP DISK ADDR AND TYPE OF INCLUDE FLDS IN PERIDA AND ONS000
PRMDOIO
KA E4
A3
A3
SET UP DISK ADDR AND TYPE OF INCLUDE FLDS IN PERIDA AND ONS000
B3
IS RLB ON AN RPS DEVICE
NO
YES
C3
SVC 75
CALCULATE SECTOR VALUE FOR MODULE
D3
SET SWITCH RELBSW IN READ INPUT RTN TO BRANCH
E3
THIS PRIVATE RELOC LIB
YES
NO
E4
SET SWITCH PVLBSW FOR PRIV LIB INPUT
F3
CTLRET EQU TO RDNEXT CHART AH

Chart DJ. IJBSCN - Scanning Directory if Relocatable Library on FBA. Refer to Chart 07

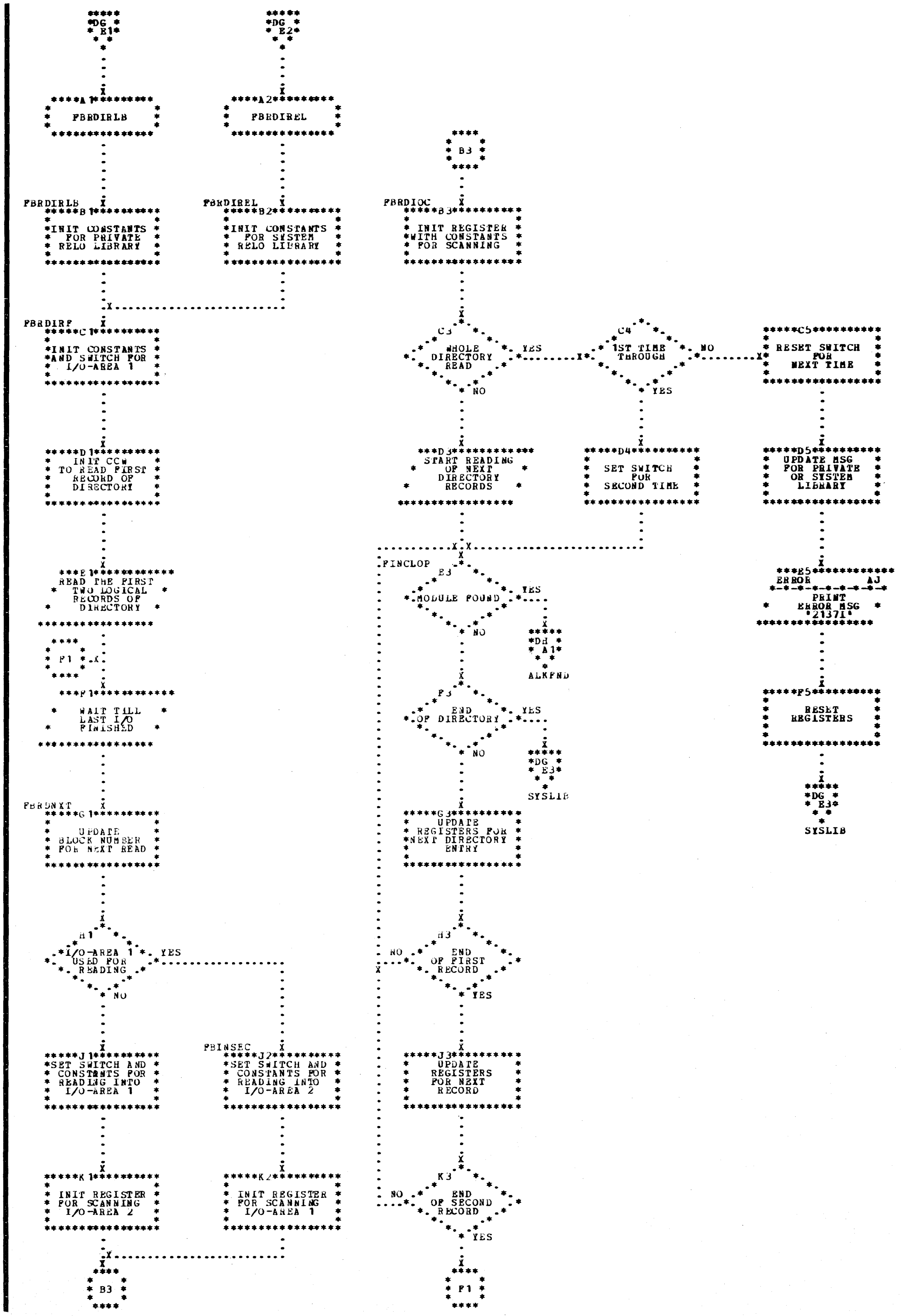
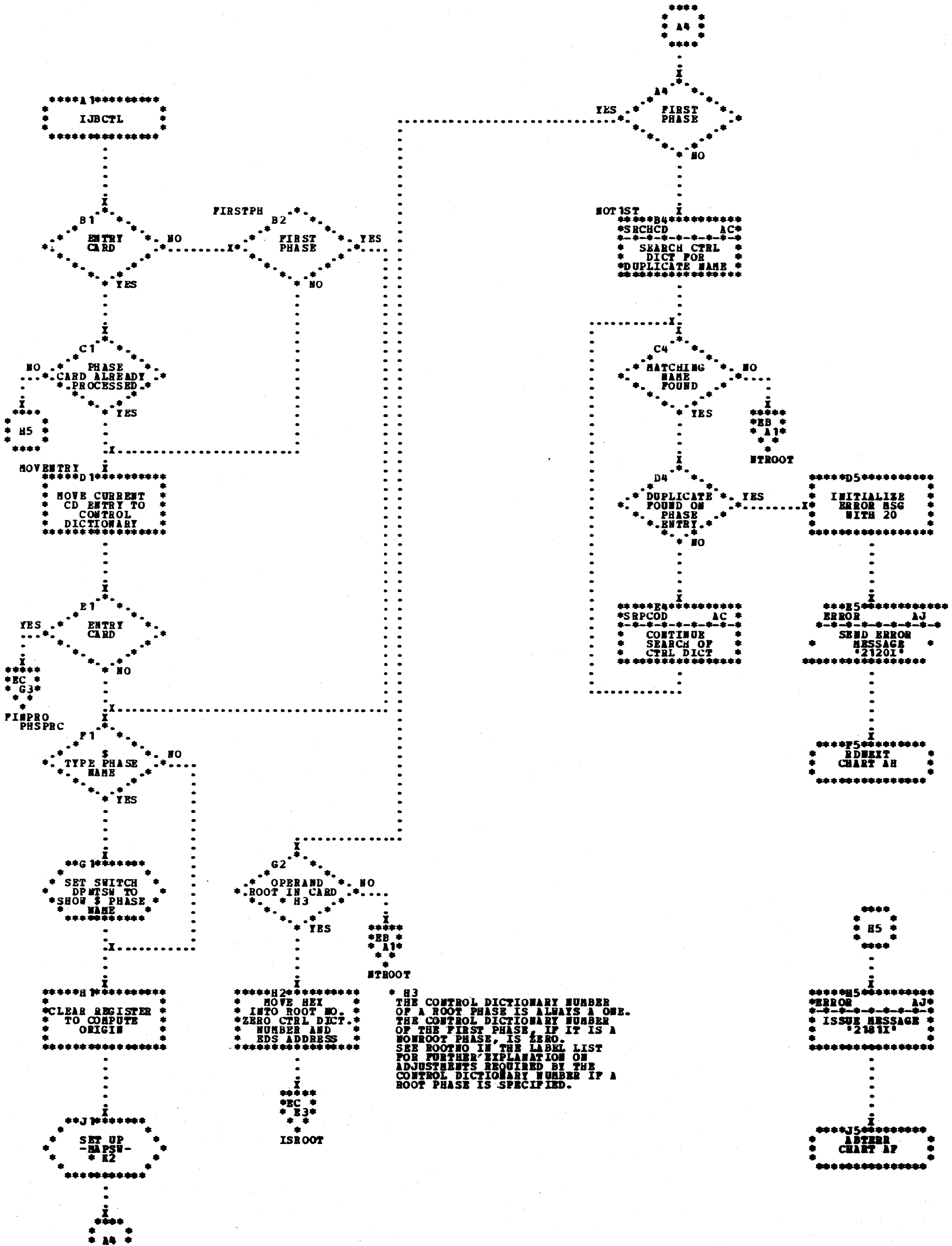

DG E1
DG E2
A1
FBRDIRLB
A2
FBRDIREL
B3
FBRDIRLB
B1
INIT CONSTANTS FOR PRIVATE RELO LIBRARY
FBRDIREL
B2
INIT CONSTANTS FOR SYSTEM RELO LIBRARY
FBRDIOC
B3
INIT REGISTER WITH CONSTANTS FOR SCANNING
FBRDIRF
C1
INIT CONSTANTS AND SWITCH FOR I/O-AREA 1
D1
INIT CCW TO READ FIRST RECORD OF DIRECTORY
E1
READ THE FIRST TWO LOGICAL RECORDS OF DIRECTORY
F1
F1
WAIT TILL LAST I/O FINISHED
FBRDNXT
G1
UPDATE BLOCK NUMBER FOR NEXT READ
H1
I/O-AREA 1 USED FOR READING
YES
NO
J1
SET SWITCH AND CONSTANTS FOR READING INTO I/O-AREA 1
FBINSEC
J2
SET SWITCH AND CONSTANTS FOR READING INTO I/O-AREA 2
K1
INIT REGISTER FOR SCANNING I/O-AREA 2
K2
INIT REGISTER FOR SCANNING I/O-AREA 1
B3
C3
WHOLE DIRECTORY READ
YES
NO
D3
START READING OF NEXT DIRECTORY RECORDS
FINCLOP
E3
MODULE FOUND
YES
NO
DH A1
ALKFND
F3
END OF DIRECTORY
YES
NO
DG E3
SYSLIB
G3
UPDATE REGISTERS FOR NEXT DIRECTORY ENTRY
H3
NO
END OF FIRST RECORD
YES
J3
UPDATE REGISTERS FOR NEXT RECORD
K3
NO
END OF SECOND RECORD
YES
F1
C4
1ST TIME THROUGH
NO
YES
D4
SET SWITCH FOR SECOND TIME
C5
RESET SWITCH FOR NEXT TIME
D5
UPDATE MSG FOR PRIVATE OR SYSTEM LIBRARY
E5
ERROR
AJ
PRINT ERROR MSG '21371'
F5
RESET REGISTERS
DG E3
SYSLIB

Chart EA. IJBCTL - Phase/Entry Processor (Part 1 of 6). Refer to Chart 08


A1
IJBCTL
B1
ENTRY CARD
NO
YES
FIRSTPH
B2
FIRST PHASE
YES
NO
C1
PHASE CARD ALREADY PROCESSED
NO
YES
H5
MOVENTRY
D1
MOVE CURRENT CD ENTRY TO CONTROL DICTIONARY
E1
ENTRY CARD
YES
NO
EC G3
FINPRO
PHSPRC
F1
TYPE $ PHASE NAME
NO
YES
G1
SET SWITCH DPNTSW TO SHOW $ PHASE NAME
H1
CLEAR REGISTER TO COMPUTE ORIGIN
J1
SET UP -MAPSW- * K2
A4
G2
OPERAND ROOT IN CARD * H3
NO
YES
EB A1
NTROOT
H2
MOVE HEX INTO ROOT NO. ZERO CTRL DICT. NUMBER AND EDS ADDRESS
EC E3
ISROOT
A4
A4
FIRST PHASE
YES
NO
NOT1ST
B4
SRCHCD AC
SEARCH CTRL DICT FOR DUPLICATE NAME
C4
MATCHING NAME FOUND
NO
YES
EB A1
NTROOT
D4
DUPLICATE FOUND ON PHASE ENTRY
YES
NO
D5
INITIALIZE ERROR MSG WITH 20
E4
SRPCOD AC
CONTINUE SEARCH OF CTRL DICT
E5
ERROR AJ
SEND ERROR MESSAGE '21201'
F5
RDNEXT CHART AH
H5
H5
ERROR AJ
ISSUE MESSAGE '2141I'
J5
ABTERR CHART AF

* H3
THE CONTROL DICTIONARY NUMBER OF A ROOT PHASE IS ALWAYS A ONE. THE CONTROL DICTIONARY NUMBER OF THE FIRST PHASE, IF IT IS A NONROOT PHASE, IS ZERO. SEE ROOTNO IN THE LABEL LIST FOR FURTHER EXPLANATION ON ADJUSTMENTS REQUIRED BY THE CONTROL DICTIONARY NUMBER IF A ROOT PHASE IS SPECIFIED.

* K2
ASSUME PHASE TO BE RELOCATABLE

Chart EB. IJBCTL - Phase/Entry Processor (Part 2 of 6). Refer to Chart 08

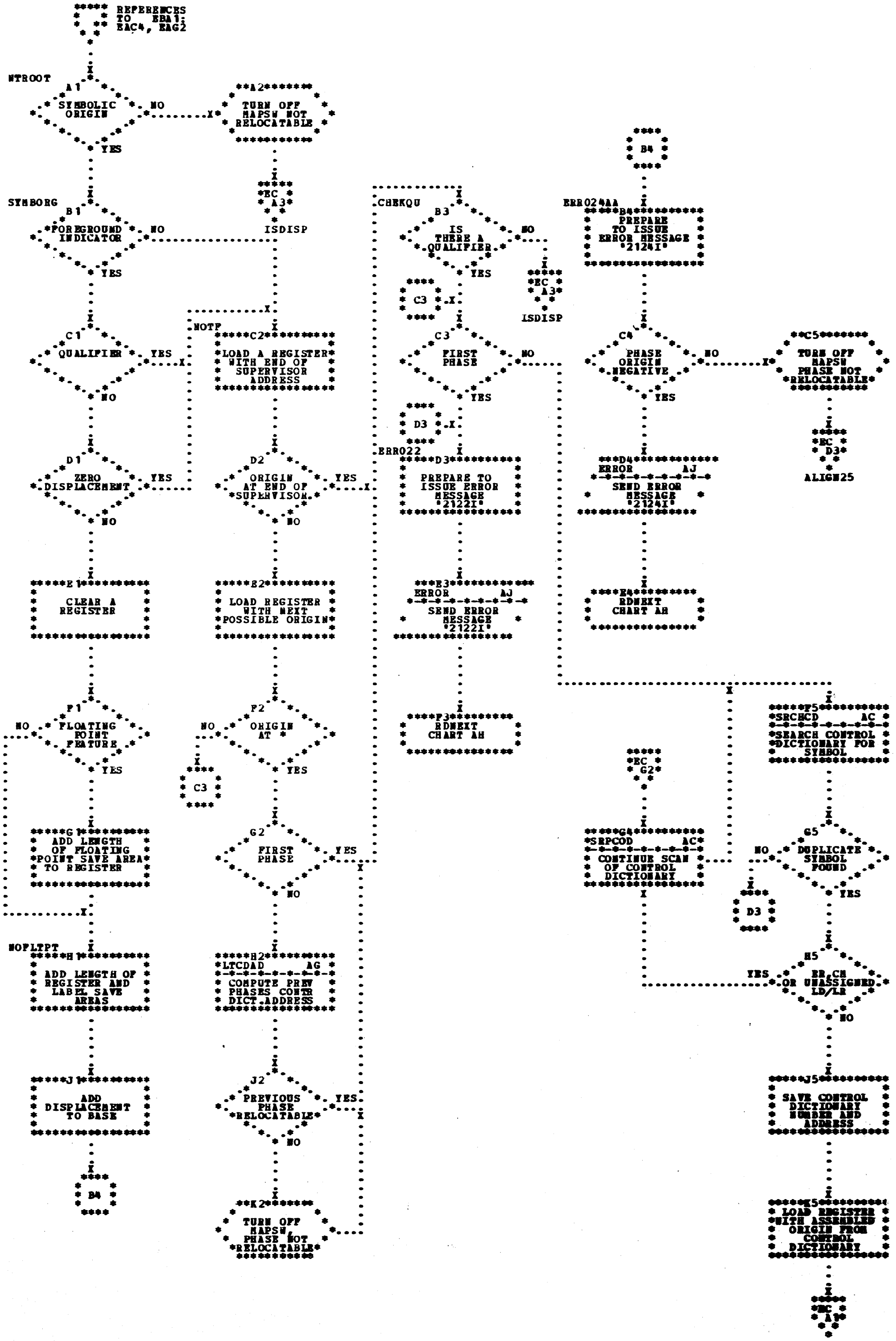
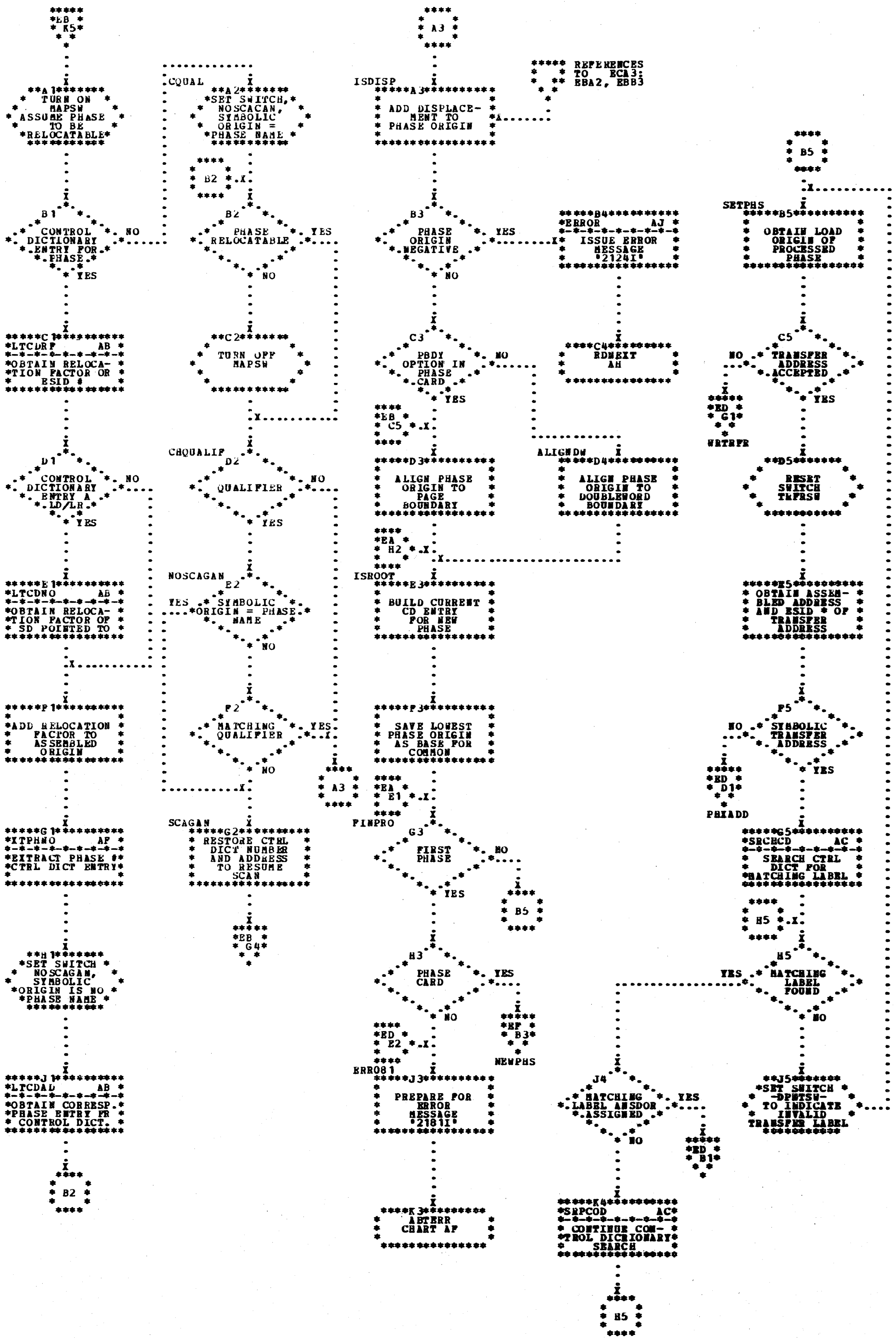

REFERENCES TO EBA1: EAC4, EAG2
NTROOT
A1
SYMBOLIC ORIGIN
NO
YES
A2
TURN OFF MAPSW NOT RELOCATABLE
EC A3
ISDISP
SYMBORG
B1
FOREGROUND INDICATOR
NO
YES
C1
QUALIFIER
YES
NO
NOTP
C2
LOAD A REGISTER WITH END OF SUPERVISOR ADDRESS
D1
ZERO DISPLACEMENT
YES
NO
D2
ORIGIN AT END OF SUPERVISOR
YES
NO
E1
CLEAR A REGISTER
E2
LOAD REGISTER WITH NEXT POSSIBLE ORIGIN
F1
FLOATING POINT FEATURE
NO
YES
F2
ORIGIN AT *
NO
YES
C3
G1
ADD LENGTH OF FLOATING POINT SAVE AREA TO REGISTER
G2
FIRST PHASE
YES
NO
NOFLTPT
H1
ADD LENGTH OF REGISTER AND LABEL SAVE AREAS
H2
LTCDAD AG
COMPUTE PREV PHASES CONTR DICT.ADDRESS
J1
ADD DISPLACEMENT TO BASE
B4
J2
PREVIOUS PHASE RELOCATABLE
YES
NO
K2
TURN OFF MAPSW, PHASE NOT RELOCATABLE
CHEKQU
B3
IS THERE A QUALIFIER
NO
YES
EC A3
ISDISP
C3
C3
FIRST PHASE
NO
YES
D3
ERR022
D3
PREPARE TO ISSUE ERROR MESSAGE "2122I"
E3
ERROR AJ
SEND ERROR MESSAGE "2122I"
F3
RDNEXT CHART AH
B4
ERR024AA
B4
PREPARE TO ISSUE ERROR MESSAGE "2124I"
C4
PHASE ORIGIN NEGATIVE
NO
YES
C5
TURN OFF MAPSW PHASE NOT RELOCATABLE
EC D3
ALIGN25
D4
ERROR AJ
SEND ERROR MESSAGE "2124I"
E4
RDNEXT CHART AH
F5
SRCHCD AC
SEARCH CONTROL DICTIONARY FOR SYMBOL
EC G2
G4
SRPCOD AC
CONTINUE SCAN OF CONTROL DICTIONARY
G5
DUPLICATE SYMBOL FOUND
NO
YES
D3
H5
ER,CR OR UNASSIGNED LD/LR
YES
NO
J5
SAVE CONTROL DICTIONARY NUMBER AND ADDRESS
K5
LOAD REGISTER WITH ASSEMBLED ORIGIN FROM CONTROL DICTIONARY
EC A1

Chart EC. IJBCTL - Phase/Entry Processor (Part 3 of 6). Refer to Chart 08



EB K5
A1 TURN ON MAPSW ASSUME PHASE TO BE RELOCATABLE
B1 CONTROL DICTIONARY ENTRY FOR PHASE
NO
YES
C1 LTCDRP AB OBTAIN RELOCATION FACTOR OR ESID #
D1 CONTROL DICTIONARY ENTRY A LD/LR
NO
YES
E1 LTCDNO AB OBTAIN RELOCATION FACTOR OF SD POINTED TO
F1 ADD RELOCATION FACTOR TO ASSEMBLED ORIGIN
G1 XTPHNO AF EXTRACT PHASE # CTRL DICT ENTRY
H1 SET SWITCH NOSCAGAN, SYMBOLIC ORIGIN IS NO PHASE NAME
J1 LTCDAD AB OBTAIN CORRESP. PHASE ENTRY FR CONTROL DICT.
B2
CQUAL
A2 SET SWITCH, NOSCACAN, SYMBOLIC ORIGIN = PHASE NAME
B2
B2 PHASE RELOCATABLE
YES
NO
C2 TURN OFF MAPSW
CHQUALIF
D2 QUALIFIER
NO
YES
NOSCAGAN
E2 SYMBOLIC ORIGIN = PHASE NAME
YES
NO
F2 MATCHING QUALIFIER
YES
NO
A3
SCAGAN
G2 RESTORE CTRL DICT NUMBER AND ADDRESS TO RESUME SCAN
EB G4
A3
REFERENCES TO ECA3: EBA2, EBB3
ISDISP
A3 ADD DISPLACEMENT TO PHASE ORIGIN
B3 PHASE ORIGIN NEGATIVE
YES
NO
B4 ERROR AJ ISSUE ERROR MESSAGE '2124I'
C3 PBDY OPTION IN PHASE CARD
NO
YES
C4 RDNEXT AH
EB C5
D3 ALIGN PHASE ORIGIN TO PAGE BOUNDARY
ALIGNDW
D4 ALIGN PHASE ORIGIN TO DOUBLEWORD BOUNDARY
EA H2
ISROOT
E3 BUILD CURRENT CD ENTRY FOR NEW PHASE
F3 SAVE LOWEST PHASE ORIGIN AS BASE FOR COMMON
EA E1
FINPRO
G3 FIRST PHASE
NO
YES
B5
H3 PHASE CARD
YES
NO
EF B3
NEWPHS
ED E2
ERR081
J3 PREPARE FOR ERROR MESSAGE '2181I'
K3 ABTERR CHART AF
B5
SETPHS
B5 OBTAIN LOAD ORIGIN OF PROCESSED PHASE
C5 TRANSFER ADDRESS ACCEPTED
NO
YES
ED G1
WRTRFR
D5 RESET SWITCH TRFRSW
E5 OBTAIN ASSEMBLED ADDRESS AND ESID # OF TRANSFER ADDRESS
F5 SYMBOLIC TRANSFER ADDRESS
NO
YES
ED D1
PHXADD
G5 SRCHCD AC SEARCH CTRL DICT FOR MATCHING LABEL
H5
H5 MATCHING LABEL FOUND
YES
NO
J4 MATCHING LABEL ANSDOR ASSIGNED
YES
NO
ED B1
J5 SET SWITCH -DPHTSW- TO INDICATE INVALID TRANSFER LABEL
K4 SRPCOD AC CONTINUE CONTROL DICRIONARY SEARCH
H5

Chart ED. IJBCTL - Phase/Entry Processor (Part 4 of 6). Refer to Chart 08

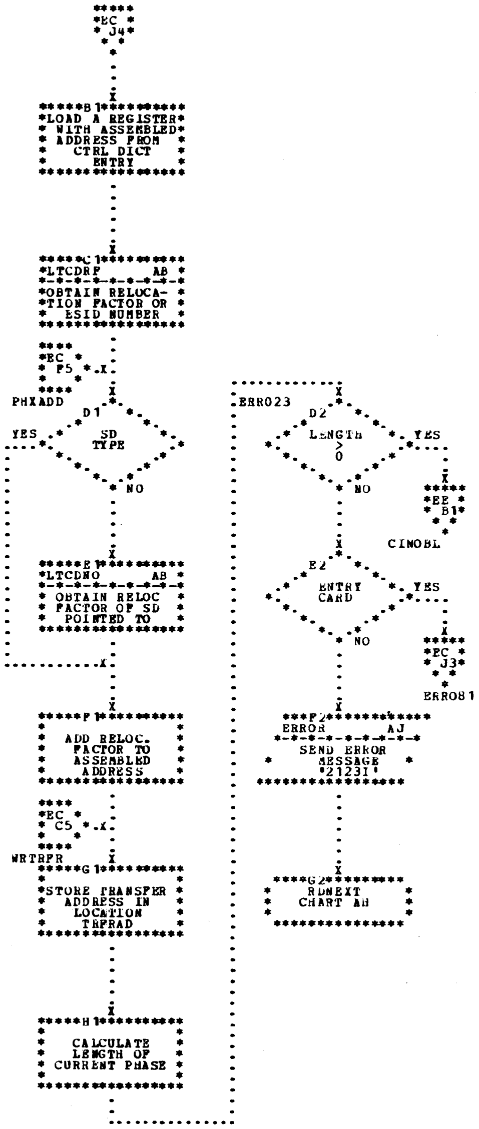
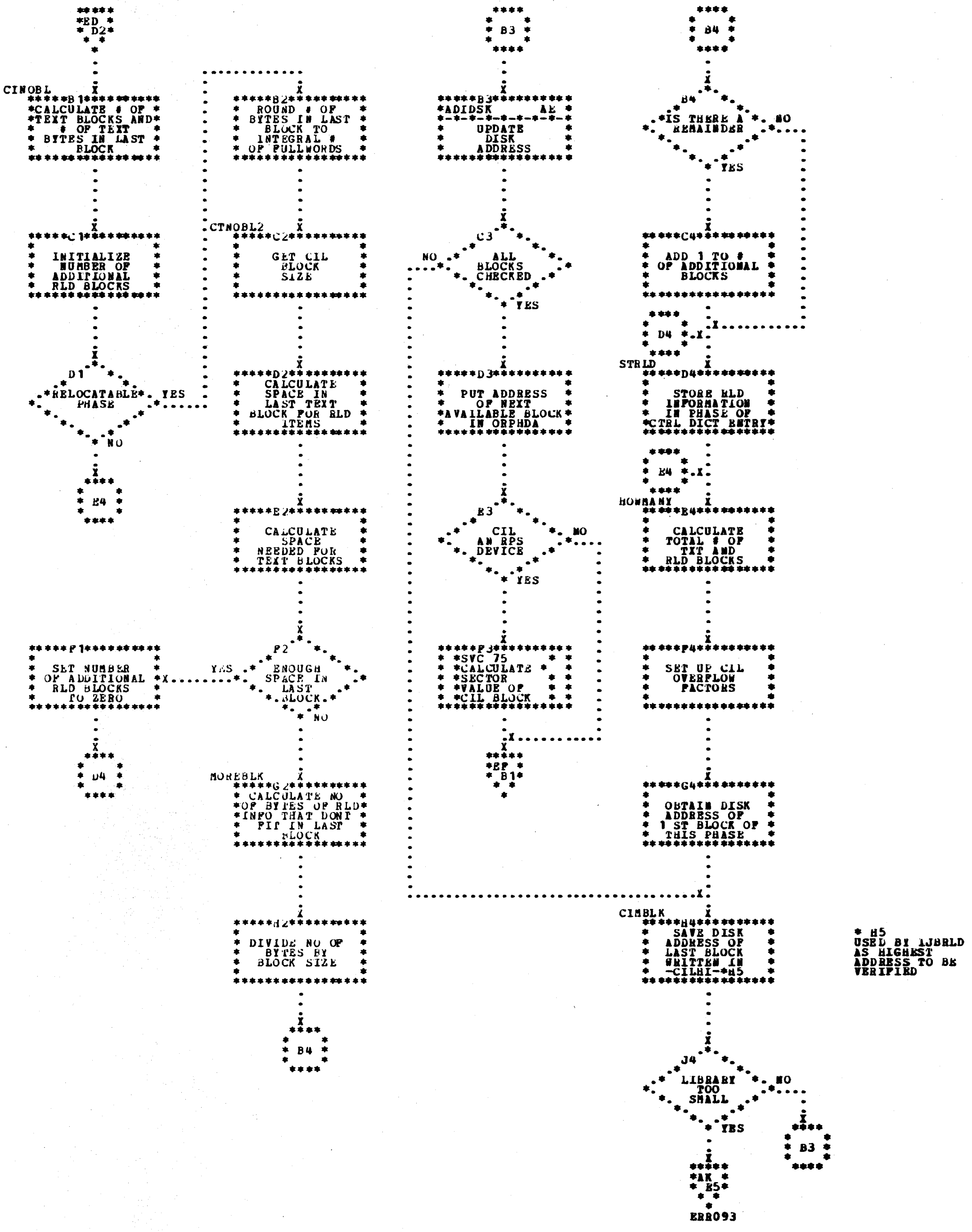

EC
J4
B1
LOAD A REGISTER WITH ASSEMBLED ADDRESS FROM CTRL DICT ENTRY
C1
LTCDRF AB
OBTAIN RELOCATION FACTOR OR ESID NUMBER
EC
F5
PHXADD
D1
SD TYPE
YES
NO
E1
LTCDNO AB
OBTAIN RELOC FACTOR OF SD POINTED TO
F1
ADD RELOC. FACTOR TO ASSEMBLED ADDRESS
EC
C5
WRTRFR
G1
STORE TRANSFER ADDRESS IN LOCATION TRFRAD
H1
CALCULATE LENGTH OF CURRENT PHASE
ERR023
D2
LENGTH > 0
YES
NO
EE
B1
CINOBL
E2
ENTRY CARD
YES
NO
EC
J3
ERR081
F2
ERROR AJ
SEND ERROR MESSAGE '2123I'
G2
RDNEXT CHART AH

Chart EE. IJBCTL - Phase/Entry Processor (Part 5 of 6) Refer to Chart 08



ED D2
CINOBL
B1
CALCULATE # OF TEXT BLOCKS AND # OF TEXT BYTES IN LAST BLOCK
C1
INITIALIZE NUMBER OF ADDITIONAL RLD BLOCKS
D1
RELOCATABLE PHASE
YES
NO
B4
F1
SET NUMBER OF ADDITIONAL RLD BLOCKS TO ZERO
D4
B2
ROUND # OF BYTES IN LAST BLOCK TO INTEGRAL # OF FULLWORDS
CTNOBL2
C2
GET CIL BLOCK SIZE
D2
CALCULATE SPACE IN LAST TEXT BLOCK FOR RLD ITEMS
E2
CALCULATE SPACE NEEDED FOR TEXT BLOCKS
F2
ENOUGH SPACE IN LAST BLOCK
YES
NO
MOREBLK
G2
CALCULATE NO OF BYTES OF RLD INFO THAT DONT FIT IN LAST BLOCK
H2
DIVIDE NO OF BYTES BY BLOCK SIZE
B4
B3
B3
ADIDSK AE
UPDATE DISK ADDRESS
C3
ALL BLOCKS CHECKED
NO
YES
D3
PUT ADDRESS OF NEXT AVAILABLE BLOCK IN ORPHDA
E3
CIL AN RPS DEVICE
NO
YES
F3
SVC 75 CALCULATE SECTOR VALUE OF CIL BLOCK
EF B1
B4
B4
IS THERE A REMAINDER
NO
YES
C4
ADD 1 TO # OF ADDITIONAL BLOCKS
D4
STRLD
D4
STORE RLD INFORMATION IN PHASE OF CTRL DICT ENTRY
E4
HOWMANY
E4
CALCULATE TOTAL # OF TXT AND RLD BLOCKS
F4
SET UP CIL OVERFLOW FACTORS
G4
OBTAIN DISK ADDRESS OF 1 ST BLOCK OF THIS PHASE
CINBLK
H4
SAVE DISK ADDRESS OF LAST BLOCK WRITTEN IN -CILHI-*H5
* H5 USED BY IJBRLD AS HIGHEST ADDRESS TO BE VERIFIED
J4
LIBRARY TOO SMALL
NO
YES
B3
AK E5
ERR093

Chart EF. IJBCTL - Phase/Entry Processor (Part 6 of 6). Refer to Chart 08

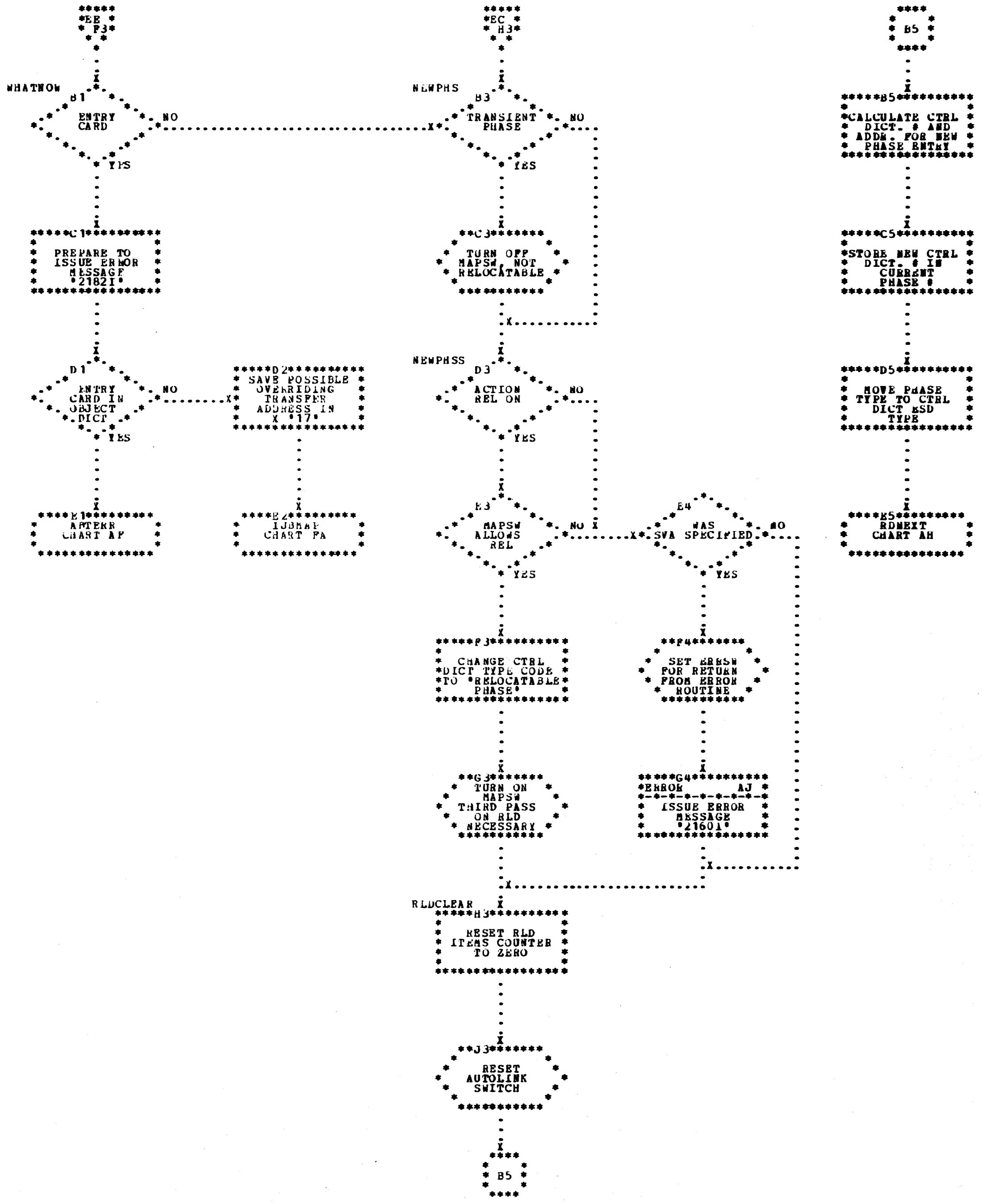

EE F3
WHATNOW
B1
ENTRY CARD
NO
YES
C1
PREPARE TO ISSUE ERROR MESSAGE '2182I'
D1
ENTRY CARD IN OBJECT DICT
NO
YES
D2
SAVE POSSIBLE OVERRIDING TRANSFER ADDRESS IN X '17'
E1
APTERR CHART AP
E2
IJBMAP CHART PA
EC H3
NEWPHS
B3
TRANSIENT PHASE
NO
YES
C3
TURN OFF MAPSW, NOT RELOCATABLE
NEWPHSS
D3
ACTION REL ON
NO
YES
E3
MAPSW ALLOWS REL
NO
YES
E4
WAS SVA SPECIFIED
NO
YES
F3
CHANGE CTRL DICT TYPE CODE TO 'RELOCATABLE PHASE'
F4
SET ERRSW FOR RETURN FROM ERROR ROUTINE
G3
TURN ON MAPSW THIRD PASS ON RLD NECESSARY
G4
ERROR AJ
ISSUE ERROR MESSAGE '2160I'
RLDCLEAR
H3
RESET RLD ITEMS COUNTER TO ZERO
J3
RESET AUTOLINK SWITCH
B5
B5
CALCULATE CTRL DICT. # AND ADDR. FOR NEW PHASE ENTRY
C5
STORE NEW CTRL DICT. # IN CURRENT PHASE #
D5
MOVE PHASE TYPE TO CTRL DICT ESD TYPE
E5
RDNEXT CHART AH

Chart FA. IJBMAP - Print Map (Part 1 cf 5). Refer to Chart 09

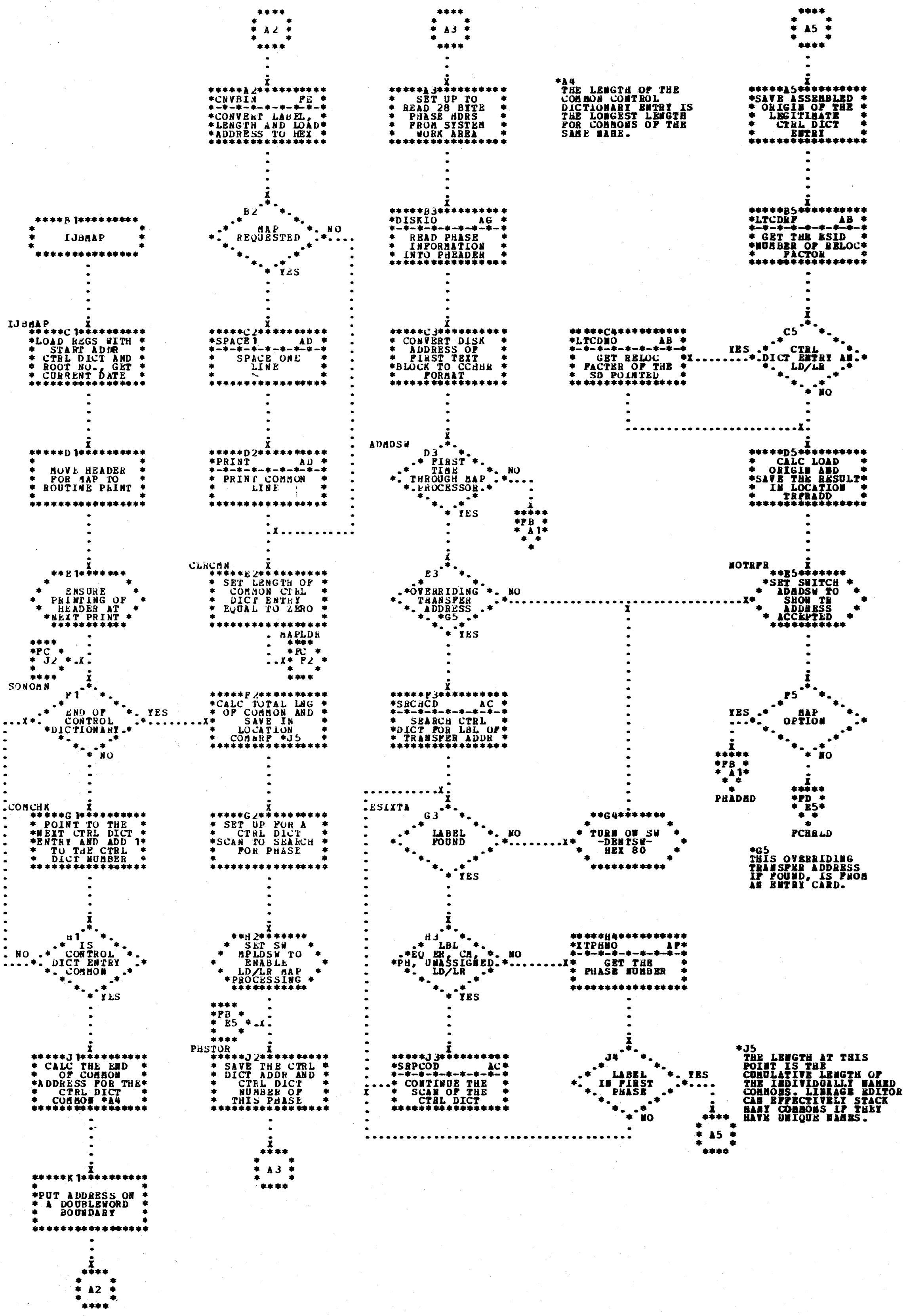
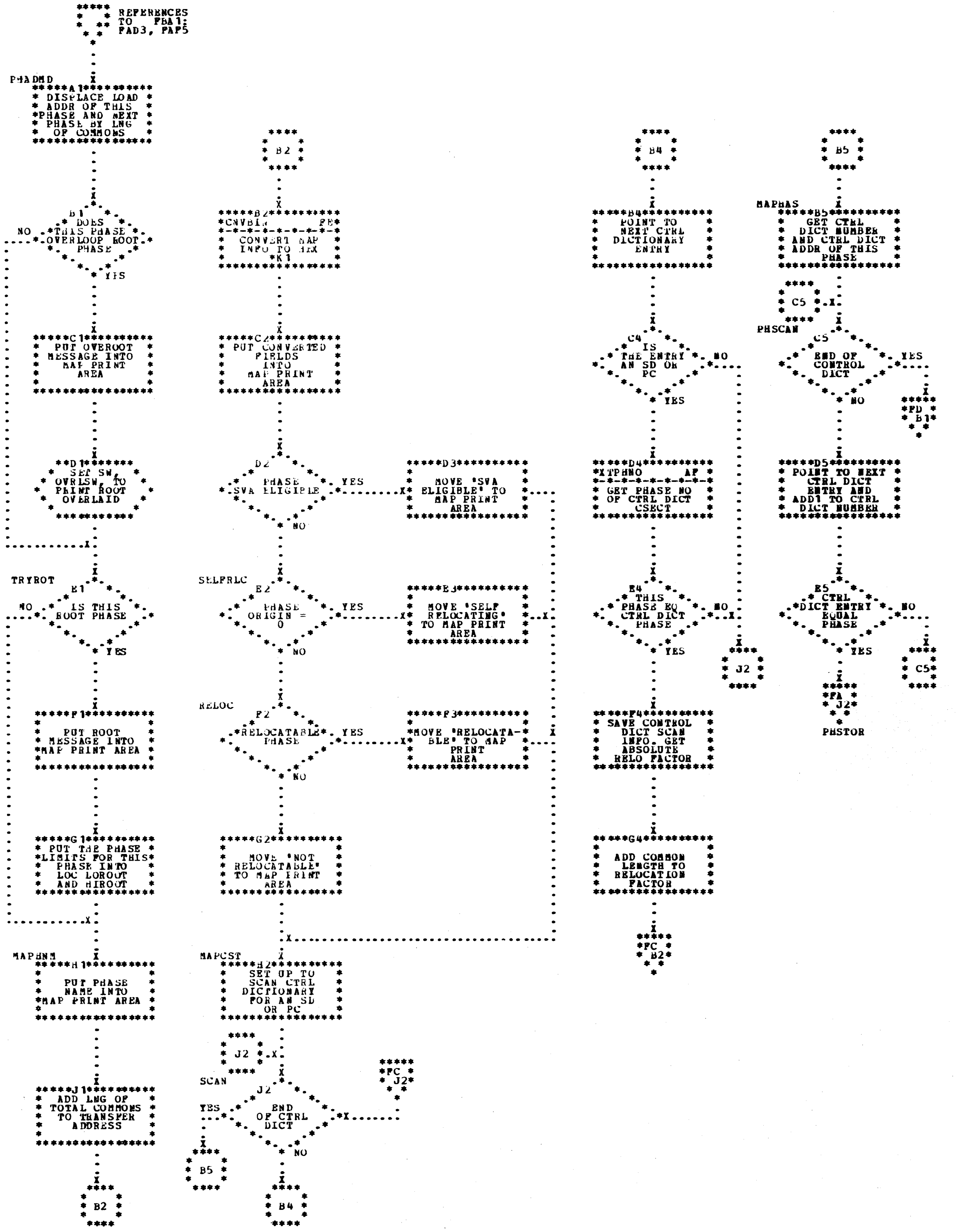
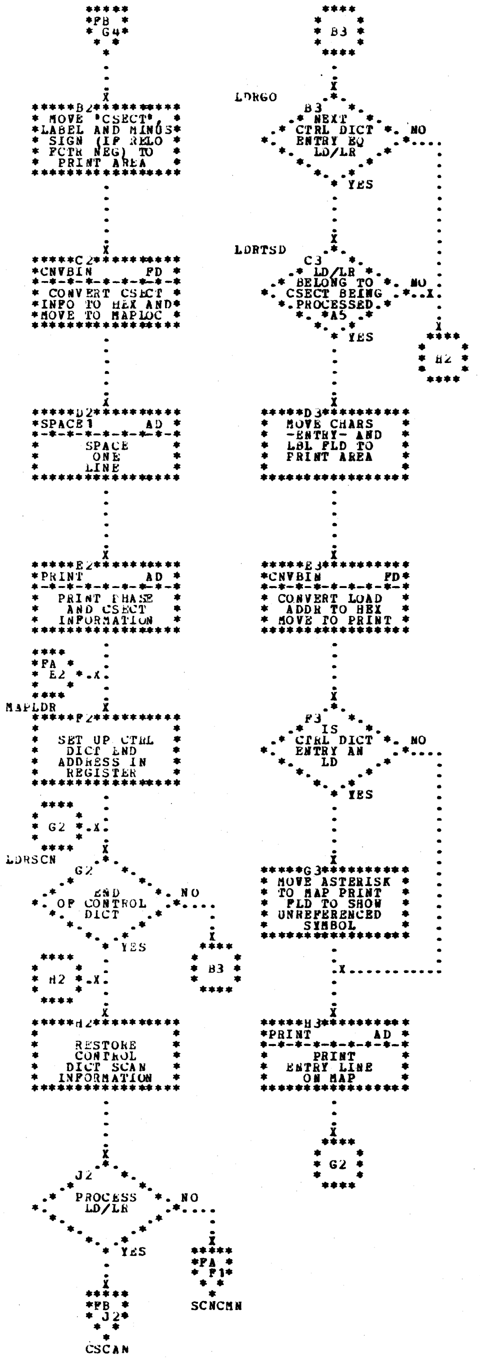

A2
A2 CNVBIN FE CONVERT LABEL, LENGTH AND LOAD ADDRESS TO HEX
B2 MAP REQUESTED NO YES
C2 SPACE1 AD SPACE ONE LINE
D2 PRINT AD PRINT COMMON LINE
CLRCMN E2 SET LENGTH OF COMMON CTRL DICT ENTRY EQUAL TO ZERO
MAPLDR FC F2
F2 CALC TOTAL LNG OF COMMON AND SAVE IN LOCATION COMMRP *J5
G2 SET UP FOR A CTRL DICT SCAN TO SEARCH FOR PHASE
H2 SET SW MPLDSW TO ENABLE LD/LR MAP PROCESSING
FB E5
PHSTOR J2 SAVE THE CTRL DICT ADDR AND CTRL DICT NUMBER OF THIS PHASE
A3
A3
A3 SET UP TO READ 28 BYTE PHASE HDRS FROM SYSTEM WORK AREA
B3 DISKIO AG READ PHASE INFORMATION INTO PHEADER
C3 CONVERT DISK ADDRESS OF FIRST TEXT BLOCK TO CCHHR FORMAT
ADMDSW D3 FIRST TIME THROUGH MAP PROCESSOR NO YES
FB A1
E3 OVERRIDING TRANSFER ADDRESS *G5 NO YES
F3 SRCHCD AC SEARCH CTRL DICT FOR LBL OF TRANSFER ADDR
ESIXTA G3 LABEL FOUND NO YES
H3 LBL EQ ER, CM, PH, UNASSIGNED LD/LR NO YES
J3 SRPCOD AC CONTINUE THE SCAN OF THE CTRL DICT
*A4 THE LENGTH OF THE COMMON CONTROL DICTIONARY ENTRY IS THE LONGEST LENGTH FOR COMMONS OF THE SAME NAME.
C4 LTCDNO AB GET RELOC FACTER OF THE SD POINTED
G4 TURN ON SW -DENTSW- HEX 80
H4 ITPHNO AP GET THE PHASE NUMBER
J4 LABEL IN FIRST PHASE YES NO
A5
A5
A5 SAVE ASSEMBLED ORIGIN OF THE LEGITIMATE CTRL DICT ENTRY
B5 LTCDMP AB GET THE ESID NUMBER OF RELOC FACTOR
C5 CTRL DICT ENTRY AN LD/LR YES NO
D5 CALC LOAD ORIGIN AND SAVE THE RESULT IN LOCATION TRFRADD
NOTRFR E5 SET SWITCH ADMDSW TO SHOW TR ADDRESS ACCEPTED
F5 MAP OPTION YES NO
FB A1 PHADMD
FD E5 FCHRLD
*G5 THIS OVERRIDING TRANSFER ADDRESS IF FOUND, IS FROM AN ENTRY CARD.
*J5 THE LENGTH AT THIS POINT IS THE CUMULATIVE LENGTH OF THE INDIVIDUALLY NAMED COMMONS. LINKAGE EDITOR CAN EFFECTIVELY STACK MANY COMMONS IF THEY HAVE UNIQUE NAMES.
B1 IJBMAP
IJBMAP C1 LOAD REGS WITH START ADDR CTRL DICT AND ROOT NO., GET CURRENT DATE
D1 MOVE HEADER FOR MAP TO ROUTINE PRINT
E1 ENSURE PRINTING OF HEADER AT NEXT PRINT
FC J2
SONOMN F1 END OF CONTROL DICTIONARY YES NO
COMCHK G1 POINT TO THE NEXT CTRL DICT ENTRY AND ADD 1 TO THE CTRL DICT NUMBER
H1 IS CONTROL DICT ENTRY COMMON NO YES
J1 CALC THE END OF COMMON ADDRESS FOR THE CTRL DICT COMMON *A4
K1 PUT ADDRESS ON A DOUBLEWORD BOUNDARY
A2

Chart FB. IJBMAP - Print Map (Part 2 of 5). Refer to Chart 09


REFERENCES TO FBA1: FAD3, FAP5
PHADMD
A1
DISPLACE LOAD ADDR OF THIS PHASE AND NEXT PHASE BY LNG OF COMMONS
B1
DOES THIS PHASE OVERLOOP ROOT PHASE
NO
YES
C1
PUT OVEROOT MESSAGE INTO MAP PRINT AREA
D1
SET SW, OVRLSW, TO PRINT ROOT OVERLAID
TRYROT
E1
IS THIS ROOT PHASE
NO
YES
F1
PUT ROOT MESSAGE INTO MAP PRINT AREA
G1
PUT THE PHASE LIMITS FOR THIS PHASE INTO LOC LOROOT AND HIROOT
MAPHNM
H1
PUT PHASE NAME INTO MAP PRINT AREA
J1
ADD LNG OF TOTAL COMMONS TO TRANSFER ADDRESS
B2
B2
CNVBIN FE
CONVERT MAP INFO TO HEX
*K1
C2
PUT CONVERTED FIELDS INTO MAP PRINT AREA
D2
PHASE SVA ELIGIBLE
YES
NO
D3
MOVE 'SVA ELIGIBLE' TO MAP PRINT AREA
SELFRLC
E2
PHASE ORIGIN = 0
YES
NO
E3
MOVE 'SELF RELOCATING' TO MAP PRINT AREA
RELOC
F2
RELOCATABLE PHASE
YES
NO
F3
MOVE 'RELOCATABLE' TO MAP PRINT AREA
G2
MOVE 'NOT RELOCATABLE' TO MAP PRINT AREA
MAPCST
H2
SET UP TO SCAN CTRL DICTIONARY FOR AN SD OR PC
J2
SCAN
J2
END OF CTRL DICT
YES
NO
FC J2
B5
B4
B4
B4
POINT TO NEXT CTRL DICTIONARY ENTRY
C4
IS THE ENTRY AN SD OR PC
NO
YES
D4
XTPHNO AP
GET PHASE NO OF CTRL DICT CSECT
E4
THIS PHASE EQ CTRL DICT PHASE
NO
YES
J2
F4
SAVE CONTROL DICT SCAN INFO. GET ABSOLUTE RELO FACTOR
G4
ADD COMMON LENGTH TO RELOCATION FACTOR
FC B2
B5
MAPHAS
B5
GET CTRL DICT NUMBER AND CTRL DICT ADDR OF THIS PHASE
C5
PHSCAN
C5
END OF CONTROL DICT
YES
NO
FD B1
D5
POINT TO NEXT CTRL DICT ENTRY AND ADD1 TO CTRL DICT NUMBER
E5
CTRL DICT ENTRY EQUAL PHASE
NO
YES
C5
FA J2
PHSTOR

*K1
THE FOLLOWING FIELDS ARE CONVERTED TO HEXADECIMAL:
1. TRANSFER ADDRESS - UPDATED BY THE LNG OF COMMONS.
2. PHASE ORIGIN LOAD ADDRESS - MAPPED AS LOCORE.
3. END OF PHASE LOAD ADDRESS - MAPPED AS HICORE.
4. CORE IMAGE LIBRARY DISK ADDRESS IN THE FORM CHR.
THIS 3-BYTE ADDRESS IS CONVERTED AT LOCATION -CNVLOP-.

Chart FC. IJBMAP - Print Map (Part 3 of 5). Refer to Chart 09


PB G4
B2 MOVE 'CSECT', LABEL AND MINUS SIGN (IF RELO FCTR NEG) TO PRINT AREA
C2 CNVBIN FD CONVERT CSECT INFO TO HEX AND MOVE TO MAPLOC
D2 SPACE1 AD SPACE ONE LINE
E2 PRINT AD PRINT PHASE AND CSECT INFORMATION
FA E2
MAPLDR
F2 SET UP CTRL DICT END ADDRESS IN REGISTER
G2
LDRSCN
G2 END OF CONTROL DICT
NO
YES
B3
H2
H2 RESTORE CONTROL DICT SCAN INFORMATION
J2 PROCESS LD/LR
NO
YES
PB J2
CSCAN
PA F1
SCNCMN
B3
LDRGO
B3 NEXT CTRL DICT ENTRY EQ LD/LR
NO
YES
LDRTSD
C3 LD/LR BELONG TO CSECT BEING PROCESSED *A5
NO
YES
H2
D3 MOVE CHARS -ENTRY- AND LBL FLD TO PRINT AREA
E3 CNVBIN FD CONVERT LOAD ADDR TO HEX MOVE TO PRINT
F3 IS CTRL DICT ENTRY AN LD
NO
YES
G3 MOVE ASTERISK TO MAP PRINT FLD TO SHOW UNREFERENCED SYMBOL
H3 PRINT AD PRINT ENTRY LINE ON MAP
G2

*A5
THE CONTROL DICTIONARY
NUMBERS OF THE CSECT BEING
PROCESSSED AND THE CSECT
POINTED TO BY THE LD/LR ARE
COMPARED. IF AN EQUAL CONDI
TION IS FOUND, THE LD/LR
BELONGS TO THE CSECT BEING
PROCESSED

Chart FD. IJBMAP - Print Map (Part 4 of 5). Refer to Chart 09

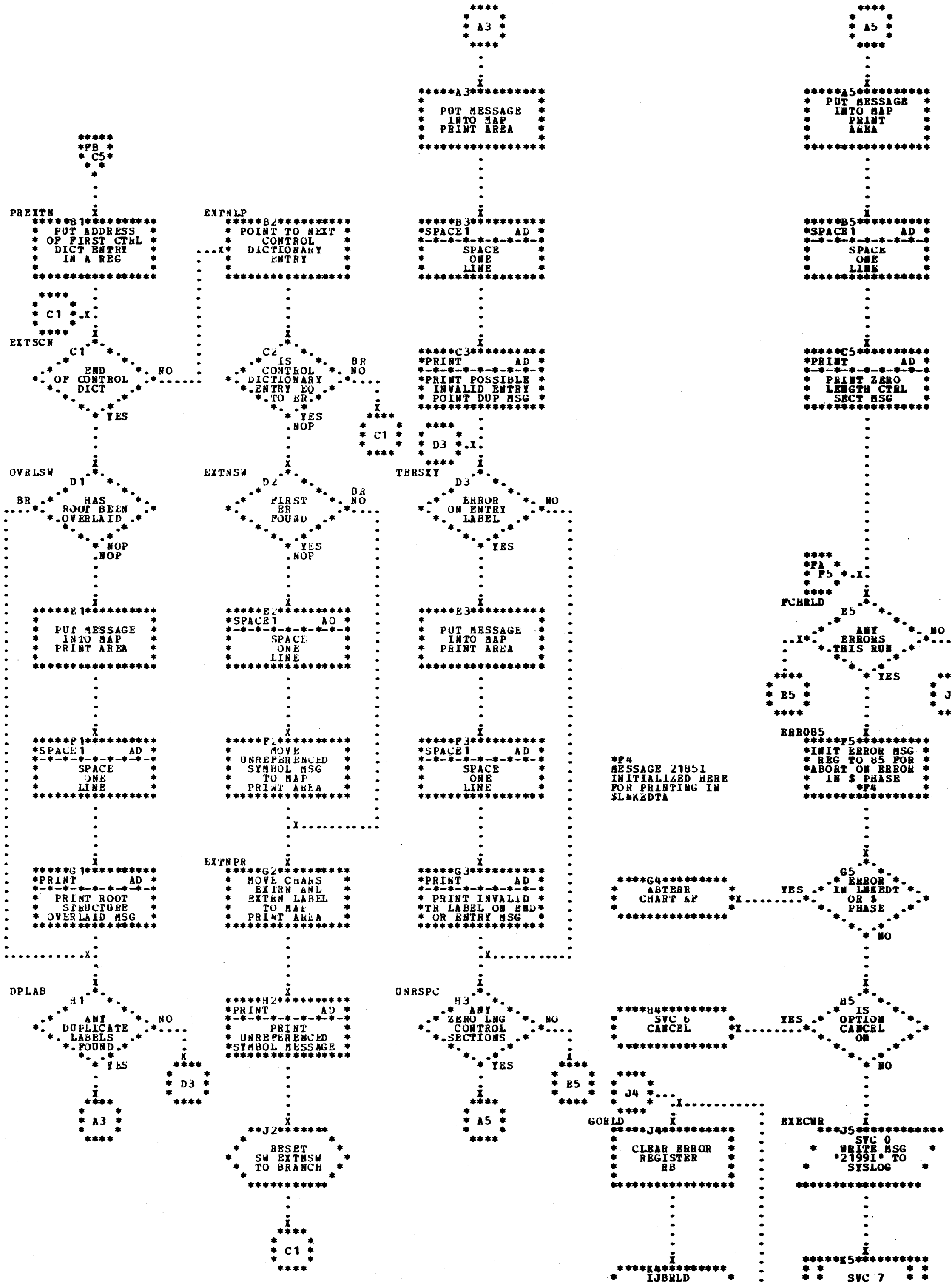

A3
A3 PUT MESSAGE INTO MAP PRINT AREA
A5
A5 PUT MESSAGE INTO MAP PRINT AREA
FB C5
PREXTN B1 PUT ADDRESS OF FIRST CTRL DICT ENTRY IN A REG
EXTNLP B2 POINT TO NEXT CONTROL DICTIONARY ENTRY
B3 SPACE1 AD SPACE ONE LINE
B5 SPACE1 AD SPACE ONE LINE
C1
EXTSCN C1 END OF CONTROL DICT
NO
YES
C2 IS CONTROL DICTIONARY ENTRY EQ TO ER
BR NO
YES NOP
C1
C3 PRINT AD PRINT POSSIBLE INVALID ENTRY POINT DUP MSG
C5 PRINT AD PRINT ZERO LENGTH CTRL SECT MSG
D3
OVRLSW D1 HAS ROOT BEEN OVERLAID
BR
NOP NOP
EXTNSW D2 FIRST ER FOUND
BR NO
YES NOP
TERSXY D3 ERROR ON ENTRY LABEL
NO
YES
FA F5
FCHRLD E5 ANY ERRORS THIS RUN
NO
YES
E5
J4
E1 PUT MESSAGE INTO MAP PRINT AREA
E2 SPACE1 AD SPACE ONE LINE
E3 PUT MESSAGE INTO MAP PRINT AREA
F1 SPACE1 AD SPACE ONE LINE
F2 MOVE UNREFERENCED SYMBOL MSG TO MAP PRINT AREA
F3 SPACE1 AD SPACE ONE LINE
*F4 MESSAGE 21851 INITIALIZED HERE FOR PRINTING IN $LNKEDTA
ERR085 F5 INIT ERROR MSG REG TO 85 FOR ABORT ON ERROR IN $ PHASE *F4
G1 PRINT AD PRINT ROOT STRUCTURE OVERLAID MSG
EXTNPR G2 MOVE CHARS EXTRN AND EXTRN LABEL TO MAP PRINT AREA
G3 PRINT AD PRINT INVALID TR LABEL OR END OR ENTRY MSG
G4 ABTERR CHART AF
YES
G5 ERROR IN LNKEDT OR $ PHASE
NO
DPLAB H1 ANY DUPLICATE LABELS FOUND
NO
YES
D3
H2 PRINT AD PRINT UNREFERENCED SYMBOL MESSAGE
UNRSPC H3 ANY ZERO LNG CONTROL SECTIONS
NO
YES
E5
H4 SVC 6 CANCEL
YES
H5 IS OPTION CANCEL ON
NO
A3
A5
J4
J2 RESET SW EXTNSW TO BRANCH
GORLD J4 CLEAR ERROR REGISTER RB
EXECWR J5 SVC 0 WRITE MSG '2199I' TO SYSLOG
C1
K4 IJBRLD CHART GA
K5 SVC 7 WAIT

Chart FE. IJBMAP - Print Map (Part 5 of 5). Refer to Chart 09

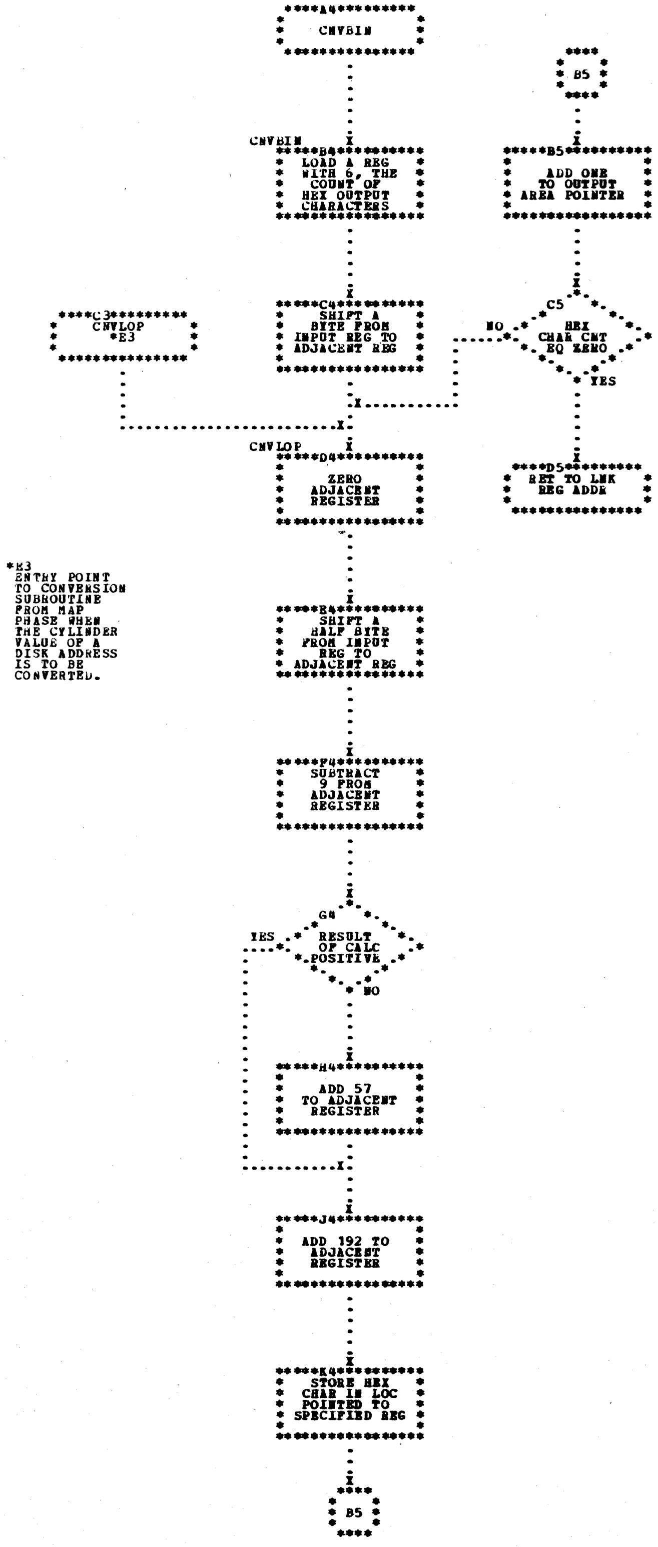
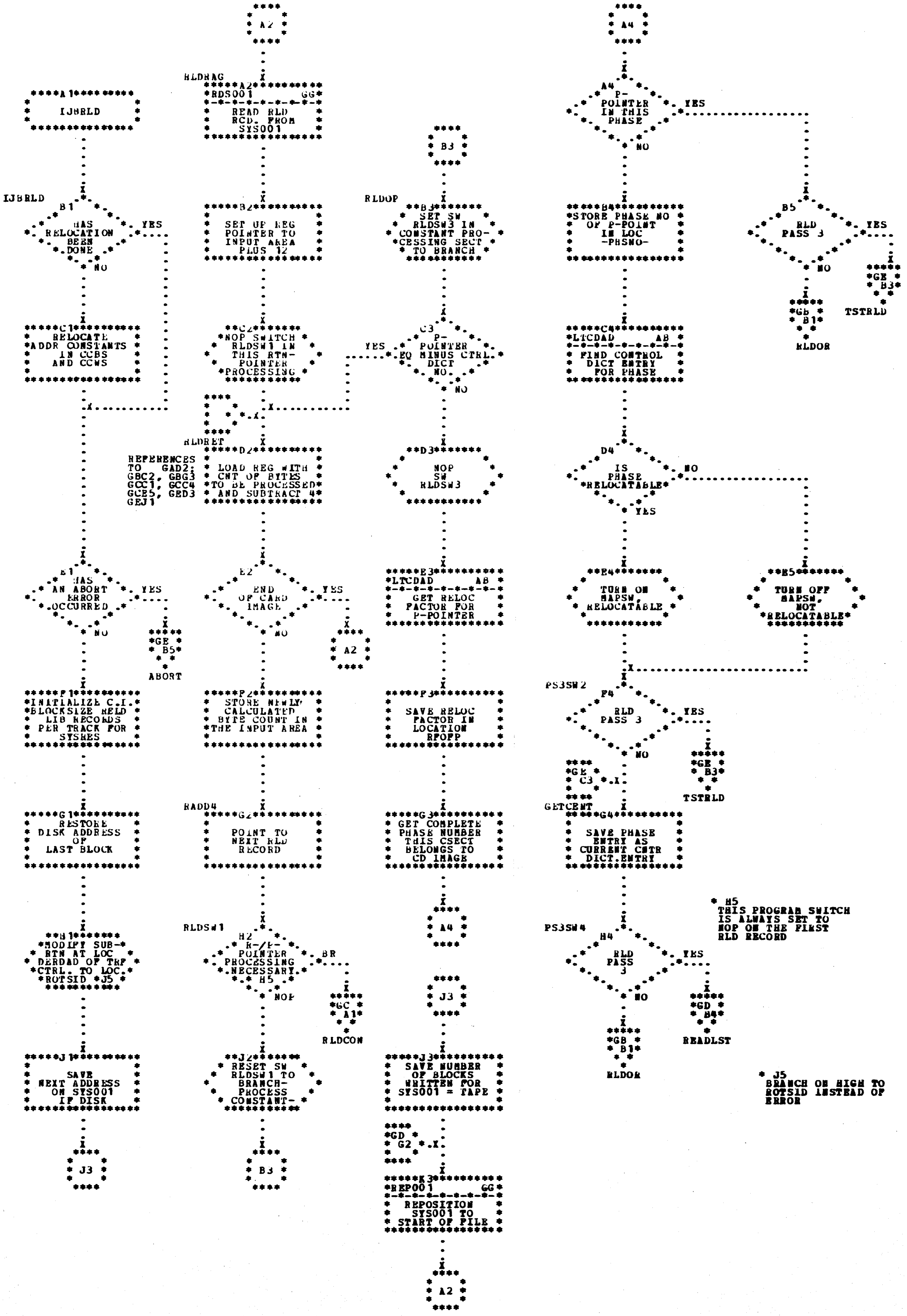

A4
CNVBIN
B5
CNVBIN
B4
LOAD A REG WITH 6, THE COUNT OF HEX OUTPUT CHARACTERS
B5
ADD ONE TO OUTPUT AREA POINTER
C3
CNVLOP *E3
C4
SHIFT A BYTE FROM INPUT REG TO ADJACENT REG
C5
HEX CHAR CNT EQ ZERO
NO
YES
CNVLOP
D4
ZERO ADJACENT REGISTER
D5
RET TO LNK REG ADDR
*E3
ENTRY POINT TO CONVERSION SUBROUTINE FROM MAP PHASE WHEN THE CYLINDER VALUE OF A DISK ADDRESS IS TO BE CONVERTED.
E4
SHIFT A HALF BYTE FROM INPUT REG TO ADJACENT REG
F4
SUBTRACT 9 FROM ADJACENT REGISTER
G4
RESULT OF CALC POSITIVE
YES
NO
H4
ADD 57 TO ADJACENT REGISTER
J4
ADD 192 TO ADJACENT REGISTER
K4
STORE HEX CHAR IN LOC POINTED TO SPECIFIED REG
B5

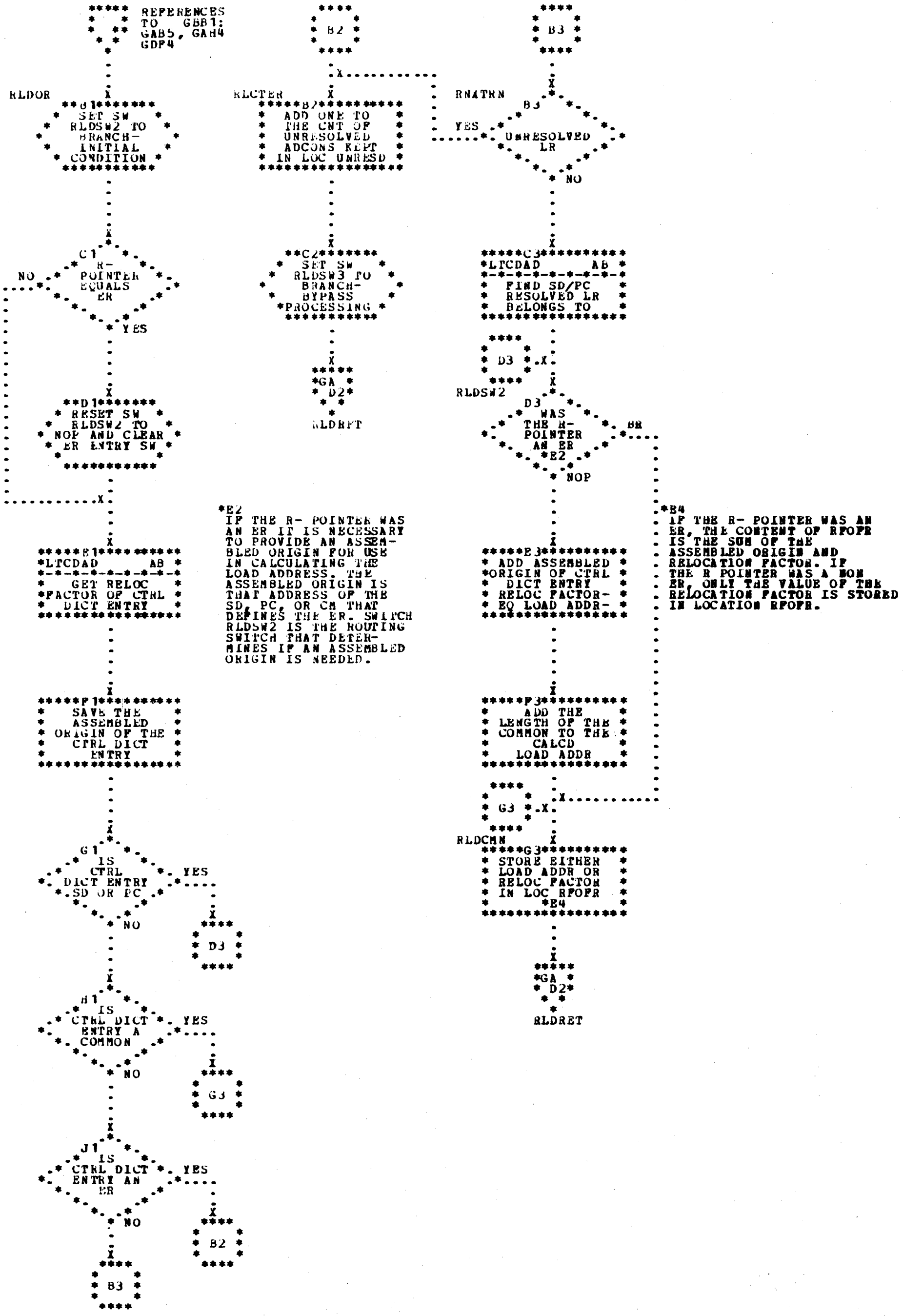
Chart GA. IJBRLD - Pass 2 P-Pointer Processing. Refer to Chart 10


A1
IJBRLD
IJBRLD
B1
HAS RELOCATION BEEN DONE
YES
NO
C1
RELOCATE ADDR CONSTANTS IN CCBS AND CCWS
E1
HAS AN ABORT ERROR OCCURRED
YES
NO
GE B5
ABORT
F1
INITIALIZE C.I. BLOCKSIZE RELD LIB RECORDS PER TRACK FOR SYSRES
G1
RESTORE DISK ADDRESS OF LAST BLOCK
H1
MODIFY SUB-RTN AT LOC DERDAD OF TRF CTRL. TO LOC. ROTSID *J5
J1
SAVE NEXT ADDRESS ON SYS001 IF DISK
J3
A2
RLDRAG
A2
RDS001 GG
READ RLD RCD. FROM SYS001
B2
SET UP REG POINTER TO INPUT AREA PLUS 12
C2
NOP SWITCH RLDSW1 IN THIS RTN-POINTER PROCESSING
B
RLDRET
D2
REFERENCES TO GAD2; GBC2, GBG3 GCC1, GCC4 GCE5, GED3 GEJ1
LOAD REG WITH CNT OF BYTES TO BE PROCESSED AND SUBTRACT 4
E2
END OF CARD IMAGE
YES
NO
A2
F2
STORE NEWLY CALCULATED BYTE COUNT IN THE INPUT AREA
RADD4
G2
POINT TO NEXT RLD RECORD
RLDSW1
H2
R-/P-POINTER PROCESSING NECESSARY *H5
BR
NOP
GC A1
RLDCON
J2
RESET SW RLDSW1 TO BRANCH-PROCESS CONSTANT-
B3
B3
RLDOP
B3
SET SW RLDSW3 IN CONSTANT PRO-CESSING SECT TO BRANCH
C3
P-POINTER EQ MINUS CTRL. DICT NO.
YES
NO
D3
NOP SW RLDSW3
E3
LTCDAD AB
GET RELOC FACTOR FOR P-POINTER
F3
SAVE RELOC FACTOR IN LOCATION RPOFP
G3
GET COMPLETE PHASE NUMBER THIS CSECT BELONGS TO CD IMAGE
A4
J3
J3
SAVE NUMBER OF BLOCKS WRITTEN FOR SYS001 = TAPE
GD G2
K3
REP001 GG
REPOSITION SYS001 TO START OF FILE
A2
A4
A4
P-POINTER IN THIS PHASE
YES
NO
B4
STORE PHASE NO OF P-POINT IN LOC -PHSNO-
C4
LTCDAD AB
FIND CONTROL DICT ENTRY FOR PHASE
D4
IS PHASE RELOCATABLE
NO
YES
E4
TURN ON MAPSW, RELOCATABLE
PS3SW2
F4
RLD PASS 3
YES
NO
GE C3
GE B3
TSTRLD
GETCENT
G4
SAVE PHASE ENTRY AS CURRENT CNTR DICT.ENTRY
PS3SW4
H4
RLD PASS 3
YES
NO
GD B4
READLST
GB B1
RLDOR
B5
RLD PASS 3
YES
NO
GE B3
TSTRLD
GB B1
RLDOR
E5
TURN OFF MAPSW, NOT RELOCATABLE

* H5
THIS PROGRAM SWITCH IS ALWAYS SET TO NOP ON THE FIRST RLD RECORD

* J5
BRANCH ON HIGH TO ROTSID INSTEAD OF ERROR

Chart GB. IJBRLD - Pass 2 R-Pointer Processing. Refer to Chart 10


REFERENCES TO GBB1: GAB5, GAH4 GDF4
RLDOR
B1
SET SW RLDSW2 TO BRANCH- INITIAL CONDITION
C1
R- POINTER EQUALS ER
NO
YES
D1
RESET SW RLDSW2 TO NOP AND CLEAR ER ENTRY SW
E1
LTCDAD AB
GET RELOC FACTOR OF CTRL DICT ENTRY
F1
SAVE THE ASSEMBLED ORIGIN OF THE CTRL DICT ENTRY
G1
IS CTRL DICT ENTRY SD OR PC
YES
NO
D3
H1
IS CTRL DICT ENTRY A COMMON
YES
NO
G3
J1
IS CTRL DICT ENTRY AN ER
YES
NO
B2
B3
B2
RLCTER
B2
ADD ONE TO THE CNT OF UNRESOLVED ADCONS KEPT IN LOC UNRESD
C2
SET SW RLDSW3 TO BRANCH- BYPASS PROCESSING
GA D2
RLDRPT
B3
RNATRN
B3
UNRESOLVED LR
YES
NO
C3
LTCDAD AB
FIND SD/PC RESOLVED LR BELONGS TO
D3
RLDSW2
D3
WAS THE R- POINTER AN ER
*E2
BR
NOP
E3
ADD ASSEMBLED ORIGIN OF CTRL DICT ENTRY RELOC FACTOR- EQ LOAD ADDR-
F3
ADD THE LENGTH OF THE COMMON TO THE CALCD LOAD ADDR
G3
RLDCMN
G3
STORE EITHER LOAD ADDR OR RELOC FACTOR IN LOC RFOPR
*B4
GA D2
RLDRET

*E2
IF THE R- POINTER WAS AN ER IT IS NECESSARY TO PROVIDE AN ASSEMBLED ORIGIN FOR USE IN CALCULATING THE LOAD ADDRESS. THE ASSEMBLED ORIGIN IS THAT ADDRESS OF THE SD, PC, OR CM THAT DEFINES THE ER. SWITCH RLDSW2 IS THE ROUTING SWITCH THAT DETERMINES IF AN ASSEMBLED ORIGIN IS NEEDED.

*B4
IF THE R- POINTER WAS AN ER, THE CONTENT OF RFOPR IS THE SUM OF THE ASSEMBLED ORIGIN AND RELOCATION FACTOR. IF THE R POINTER WAS A NON ER, ONLY THE VALUE OF THE RELOCATION FACTOR IS STORED IN LOCATION RFOPR.

Chart GC. IJBRLD - Pass 2 RLD Constant Processing. Refer to Chart 11

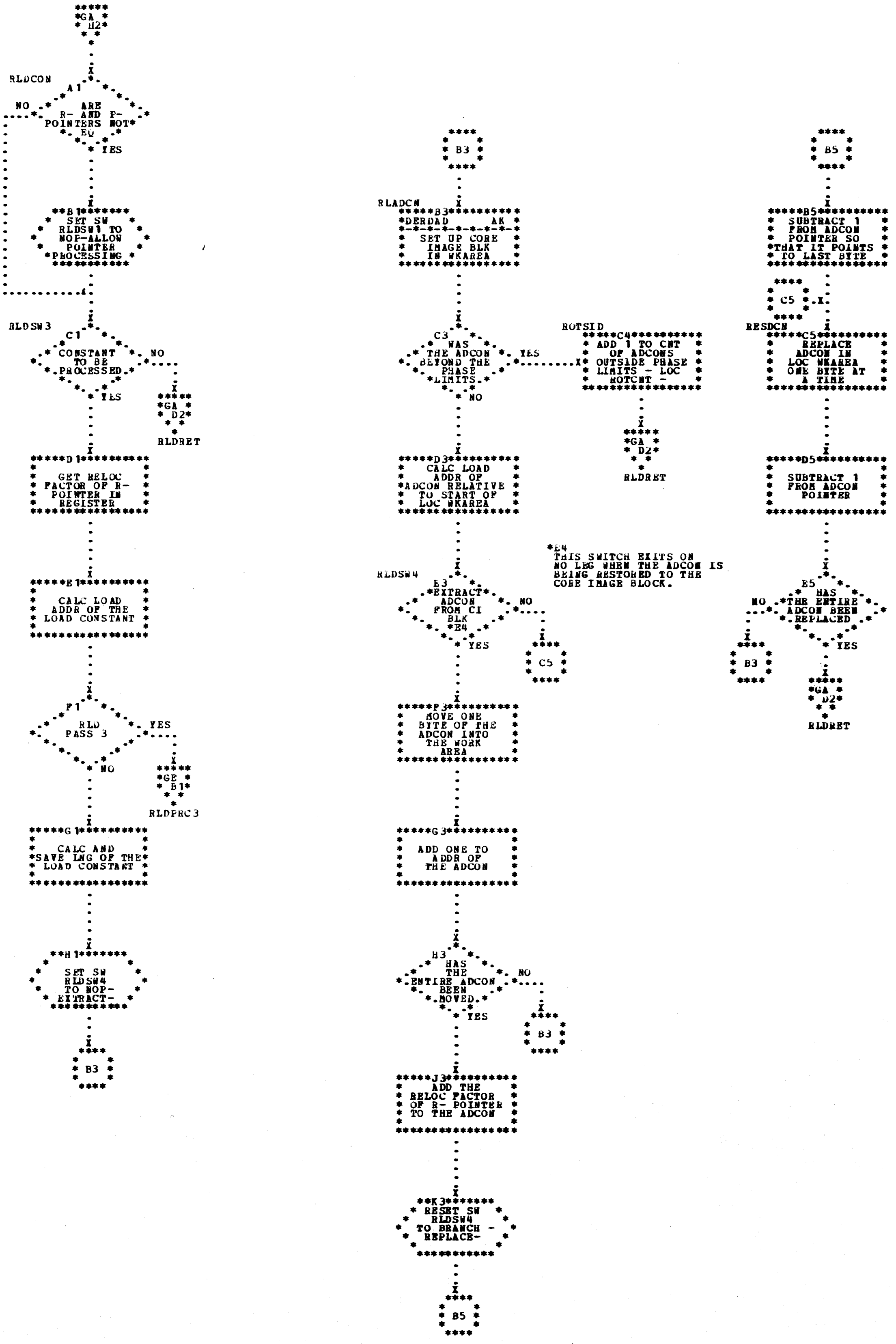
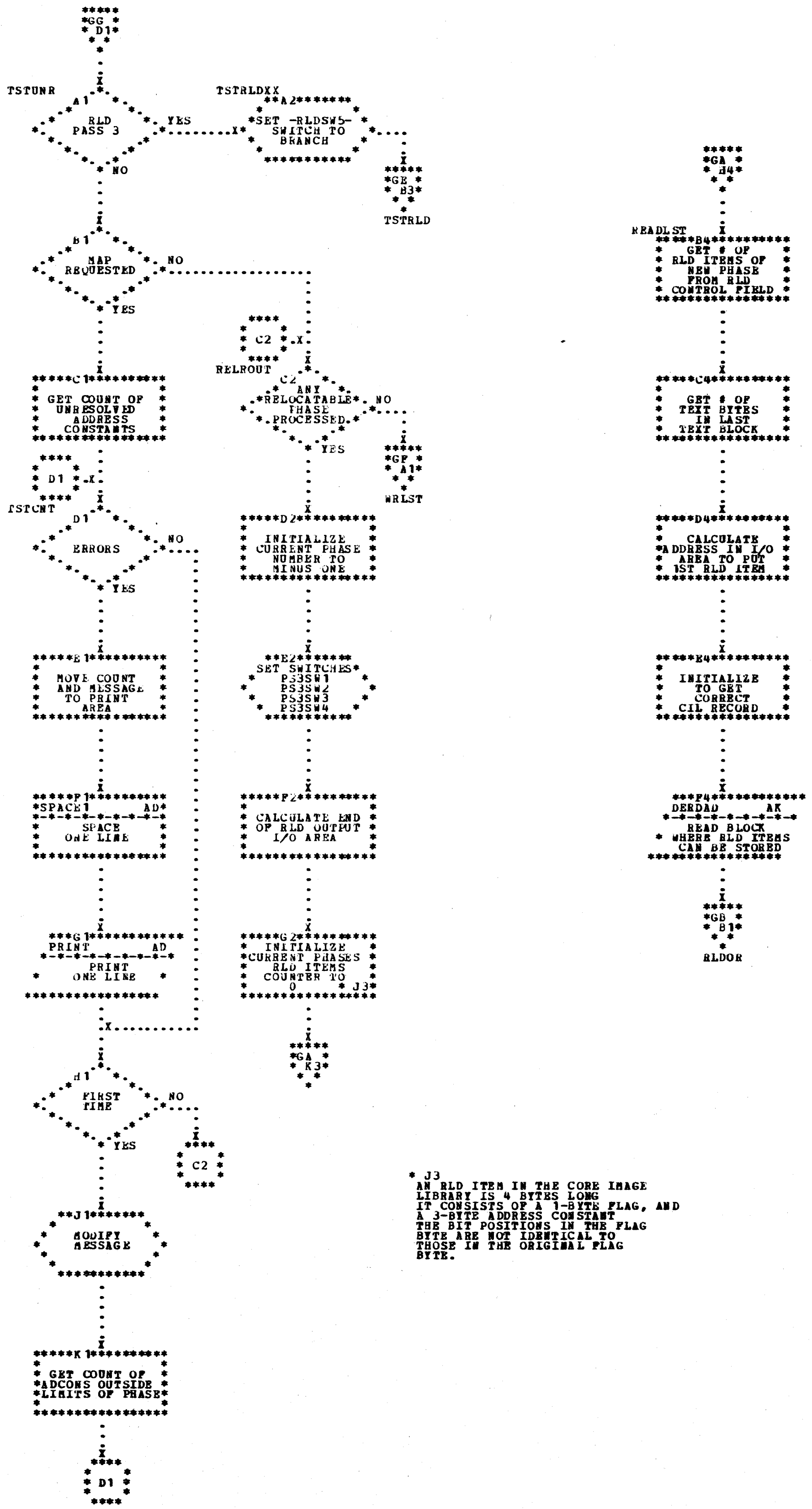
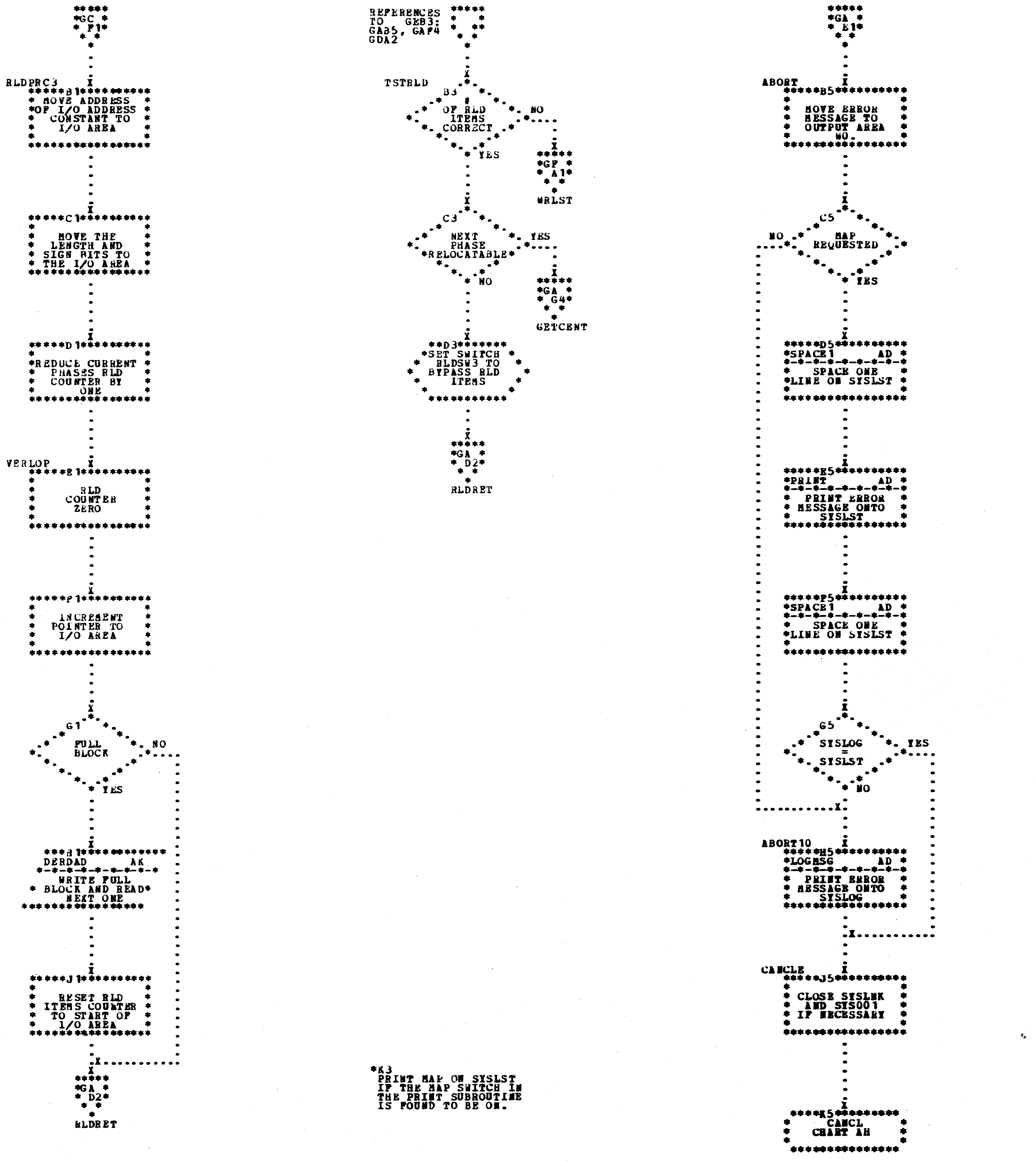

GA H2
RLDCON
A1
ARE R- AND P- POINTERS NOT EQ
NO
YES
B1
SET SW RLDSW1 TO NOP-ALLOW POINTER PROCESSING
RLDSW3
C1
CONSTANT TO BE PROCESSED
NO
YES
GA D2
RLDRET
D1
GET RELOC FACTOR OF R- POINTER IN REGISTER
E1
CALC LOAD ADDR OF THE LOAD CONSTANT
F1
RLD PASS 3
YES
NO
GE B1
RLDPRC3
G1
CALC AND SAVE LNG OF THE LOAD CONSTANT
H1
SET SW RLDSW4 TO NOP-EXTRACT-
B3
B3
RLADCN
B3
DERDAD AK
SET UP CORE IMAGE BLK IN WKAREA
C3
WAS THE ADCON BEYOND THE PHASE LIMITS
YES
NO
ROTSID
C4
ADD 1 TO CNT OF ADCONS OUTSIDE PHASE LIMITS - LOC ROTCNT -
GA D2
RLDRET
D3
CALC LOAD ADDR OF ADCON RELATIVE TO START OF LOC WKAREA
*E4
THIS SWITCH EXITS ON NO LEG WHEN THE ADCON IS BEING RESTORED TO THE CORE IMAGE BLOCK.
RLDSW4
E3
EXTRACT ADCON FROM CI BLK *E4
NO
YES
C5
F3
MOVE ONE BYTE OF THE ADCON INTO THE WORK AREA
G3
ADD ONE TO ADDR OF THE ADCON
H3
HAS THE ENTIRE ADCON BEEN MOVED
NO
YES
B3
J3
ADD THE RELOC FACTOR OF R- POINTER TO THE ADCON
K3
RESET SW RLDSW4 TO BRANCH - REPLACE-
B5
B5
B5
SUBTRACT 1 FROM ADCON POINTER SO THAT IT POINTS TO LAST BYTE
C5
RESDCN
C5
REPLACE ADCON IN LOC WKAREA ONE BYTE AT A TIME
D5
SUBTRACT 1 FROM ADCON POINTER
E5
HAS THE ENTIRE ADCON BEEN REPLACED
NO
YES
B3
GA D2
RLDRET

Chart GD. IJBRLD - Pass 3 RLD Constant Processing. Refer to Chart 11


GG D1
TSTUNR
A1
RLD PASS 3
YES
NO
TSTRLDXX
A2
SET -RLDSW5- SWITCH TO BRANCH
GE B3
TSTRLD
B1
MAP REQUESTED
NO
YES
C2
RELROUT
C2
ANY RELOCATABLE PHASE PROCESSED.
NO
YES
GF A1
WRLST
C1
GET COUNT OF UNRESOLVED ADDRESS CONSTANTS
D1
TSTCNT
D1
ERRORS
NO
YES
D2
INITIALIZE CURRENT PHASE NUMBER TO MINUS ONE
E1
MOVE COUNT AND MESSAGE TO PRINT AREA
E2
SET SWITCHES PS3SW1 PS3SW2 PS3SW3 PS3SW4
F1
SPACE1 AD
SPACE ONE LINE
F2
CALCULATE END OF RLD OUTPUT I/O AREA
G1
PRINT AD
PRINT ONE LINE
G2
INITIALIZE CURRENT PHASES RLD ITEMS COUNTER TO 0
J3
GA K3
H1
FIRST TIME
NO
YES
C2
J1
MODIFY MESSAGE
K1
GET COUNT OF ADCONS OUTSIDE LIMITS OF PHASE
D1
GA B4
READLST
B4
GET # OF RLD ITEMS OF NEW PHASE FROM RLD CONTROL FIELD
C4
GET # OF TEXT BYTES IN LAST TEXT BLOCK
D4
CALCULATE ADDRESS IN I/O AREA TO PUT 1ST RLD ITEM
E4
INITIALIZE TO GET CORRECT CIL RECORD
F4
DERDAD AK
READ BLOCK WHERE RLD ITEMS CAN BE STORED
GB B1
RLDOR

* J3
AN RLD ITEM IN THE CORE IMAGE
LIBRARY IS 4 BYTES LONG
IT CONSISTS OF A 1-BYTE FLAG, AND
A 3-BYTE ADDRESS CONSTANT
THE BIT POSITIONS IN THE FLAG
BYTE ARE NOT IDENTICAL TO
THOSE IN THE ORIGINAL FLAG
BYTE.

Chart GE. IJBRLD - Pass 2 Abort and Map Routines. Refer to Chart 11



GC F1
RLDPRC3
B1
MOVE ADDRESS OF I/O ADDRESS CONSTANT TO I/O AREA
C1
MOVE THE LENGTH AND SIGN BITS TO THE I/O AREA
D1
REDUCE CURRENT PHASES RLD COUNTER BY ONE
VERLOP
E1
RLD COUNTER ZERO
F1
INCREMENT POINTER TO I/O AREA
G1
FULL BLOCK
NO
YES
H1
DERDAD AK
WRITE FULL BLOCK AND READ NEXT ONE
J1
RESET RLD ITEMS COUNTER TO START OF I/O AREA
GA D2
RLDRET
REFERENCES TO GEB3: GAB5, GAF4 GDA2
TSTRLD
B3
OF RLD ITEMS CORRECT
NO
YES
GF A1
WRLST
C3
NEXT PHASE RELOCATABLE
YES
NO
GA G4
GETCENT
D3
SET SWITCH BLDSW3 TO BYPASS RLD ITEMS
GA D2
RLDRET
*K3
PRINT MAP ON SYSLST IF THE MAP SWITCH IN THE PRINT SUBROUTINE IS FOUND TO BE ON.
GA E1
ABORT
B5
MOVE ERROR MESSAGE TO OUTPUT AREA WO.
C5
MAP REQUESTED
NO
YES
D5
SPACE1 AD
SPACE ONE LINE ON SYSLST
E5
PRINT AD
PRINT ERROR MESSAGE ONTO SYSLST
F5
SPACE1 AD
SPACE ONE LINE ON SYSLST
G5
SYSLOG = SYSLST
YES
NO
ABORT10
H5
LOGMSG AD
PRINT ERROR MESSAGE ONTO SYSLOG
CANCLE
J5
CLOSE SYSLNK AND SYS001 IF NECESSARY
K5
CANCL CHART AH

Chart GF. IJBRLD - Pass 2 Block Phase Header. Refer to Chart 11

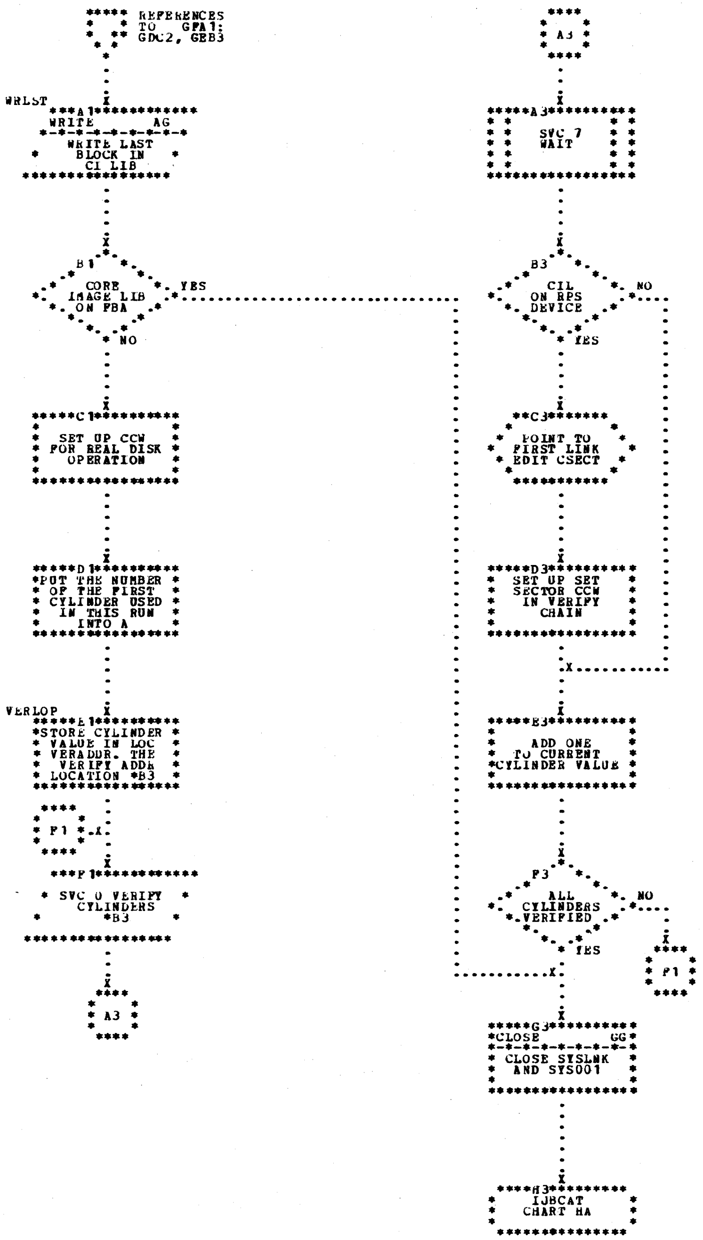
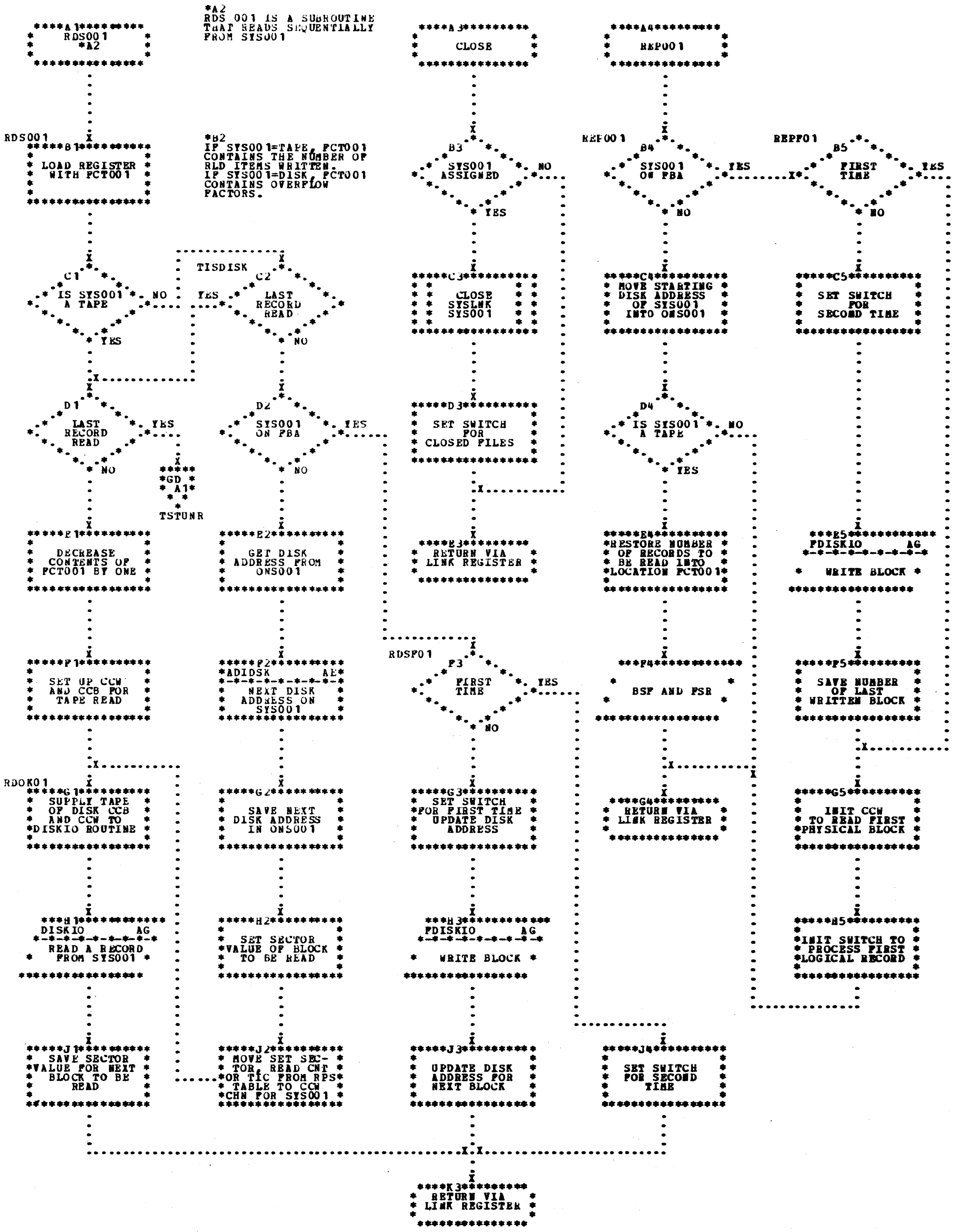

REFERENCES TO GFA1: GDC2, GEB3
WRLST
A1
WRITE AG
WRITE LAST BLOCK IN CI LIB
B1
CORE IMAGE LIB ON FBA
YES
NO
C1
SET UP CCW FOR REAL DISK OPERATION
D1
PUT THE NUMBER OF THE FIRST CYLINDER USED IN THIS RUN INTO A
VERLOP
E1
STORE CYLINDER VALUE IN LOC VERADDR. THE VERIFY ADDR LOCATION *B3
F1
F1
SVC 0 VERIFY CYLINDERS *B3
A3
A3
A3
SVC 7 WAIT
B3
CIL ON RPS DEVICE
NO
YES
C3
POINT TO FIRST LINK EDIT CSECT
D3
SET UP SET SECTOR CCW IN VERIFY CHAIN
E3
ADD ONE TO CURRENT CYLINDER VALUE
F3
ALL CYLINDERS VERIFIED
NO
YES
F1
G3
CLOSE GG
CLOSE SYSLNK AND SYS001
H3
IJBCAT CHART HA

Chart GG. IJBRLD - Pass 2 Subroutines. Refer to Chart 11

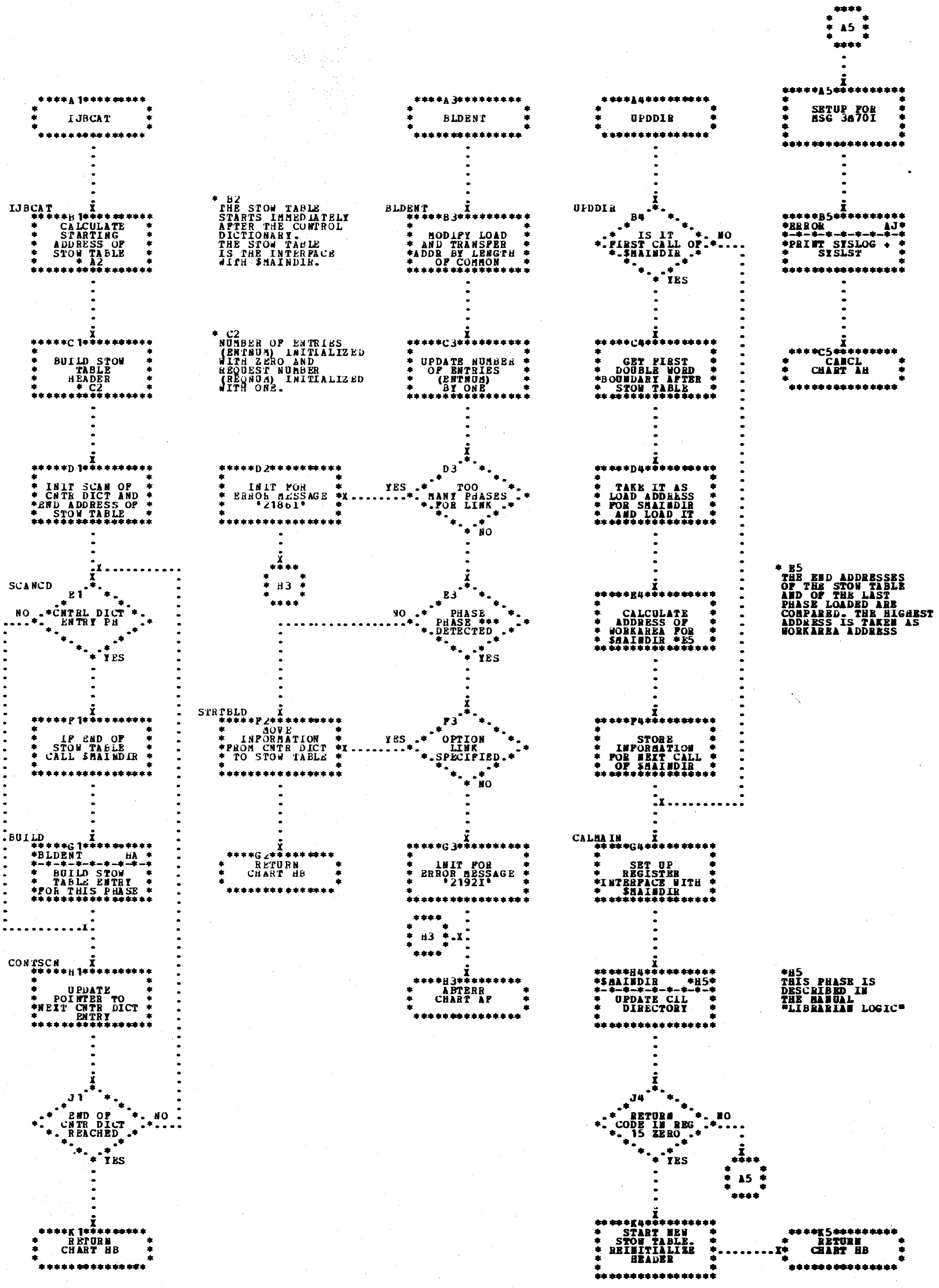


*A2
RDS 001 IS A SUBROUTINE THAT READS SEQUENTIALLY FROM SYS001

*B2
IF SYS001=TAPE, PCT001 CONTAINS THE NUMBER OF RLD ITEMS WRITTEN.
IF SYS001=DISK, PCT001 CONTAINS OVERFLOW FACTORS.

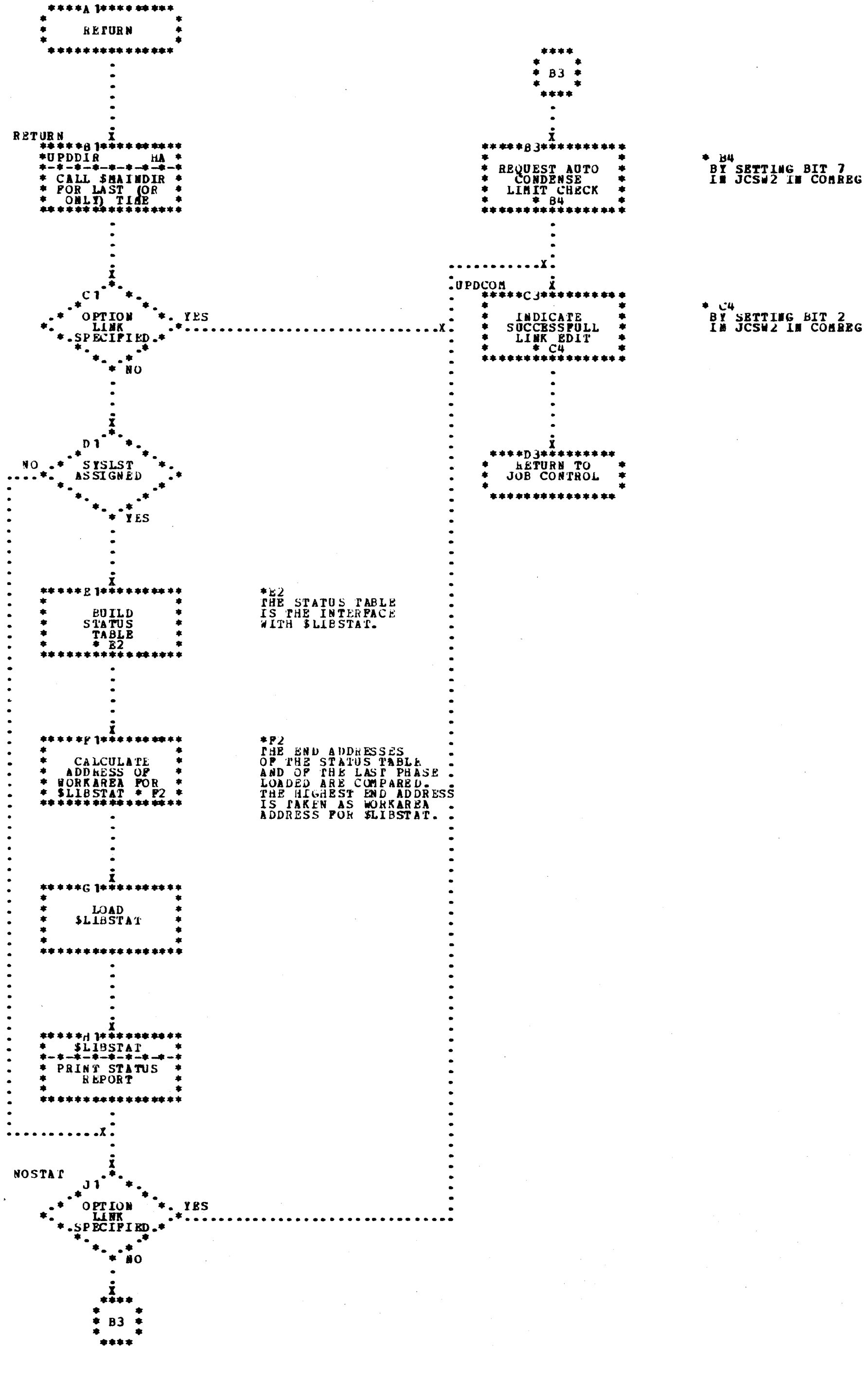
A1
RDS001 *A2
RDS001
B1
LOAD REGISTER WITH PCT001
C1
IS SYS001 A TAPE
NO
YES
TISDISK
C2
LAST RECORD READ
YES
NO
D1
LAST RECORD READ
YES
NO
GD A1
TSTUNR
D2
SYS001 ON PBA
YES
NO
E1
DECREASE CONTENTS OF PCT001 BY ONE
E2
GET DISK ADDRESS FROM ONS001
F1
SET UP CCW AND CCB FOR TAPE READ
F2
ADIDSK AE
NEXT DISK ADDRESS ON SYS001
RDOK01
G1
SUPPLY TAPE OF DISK CCB AND CCW TO DISKIO ROUTINE
G2
SAVE NEXT DISK ADDRESS IN ONS001
H1
DISKIO AG
READ A RECORD FROM SYS001
H2
SET SECTOR VALUE OF BLOCK TO BE READ
J1
SAVE SECTOR VALUE FOR NEXT BLOCK TO BE READ
J2
MOVE SET SEC- TOR, READ CNT OR TIC FROM RPS TABLE TO CCW CHN FOR SYS001
A3
CLOSE
B3
SYS001 ASSIGNED
NO
YES
C3
CLOSE SYSLNK SYS001
D3
SET SWITCH FOR CLOSED FILES
E3
RETURN VIA LINK REGISTER
RDSF01
F3
FIRST TIME
YES
NO
G3
SET SWITCH FOR FIRST TIME UPDATE DISK ADDRESS
H3
FDISKIO AG
WRITE BLOCK
J3
UPDATE DISK ADDRESS FOR NEXT BLOCK
K3
RETURN VIA LINK REGISTER
A4
REP001
REP001
B4
SYS001 ON PBA
YES
NO
C4
MOVE STARTING DISK ADDRESS OF SYS001 INTO ONS001
D4
IS SYS001 A TAPE
NO
YES
E4
RESTORE NUMBER OF RECORDS TO BE READ INTO LOCATION PCT001
F4
BSF AND FSR
G4
RETURN VIA LINK REGISTER
J4
SET SWITCH FOR SECOND TIME
REPF01
B5
FIRST TIME
YES
NO
C5
SET SWITCH FOR SECOND TIME
E5
FDISKIO AG
WRITE BLOCK
F5
SAVE NUMBER OF LAST WRITTEN BLOCK
G5
INIT CCW TO READ FIRST PHYSICAL BLOCK
H5
INIT SWITCH TO PROCESS FIRST LOGICAL RECORD

Chart HA. IJBCAT - Update CIL Directory (Part 1 of 2). Refer to Chart 12



A1
IJBCAT
IJBCAT
B1
CALCULATE STARTING ADDRESS OF STOW TABLE * A2
C1
BUILD STOW TABLE HEADER * C2
D1
INIT SCAN OF CNTR DICT AND END ADDRESS OF STOW TABLE
SCANCD
E1
CNTRL DICT ENTRY PH
NO
YES
F1
IF END OF STOW TABLE CALL $MAINDIR
BUILD
G1
BLDENT HA
BUILD STOW TABLE ENTRY FOR THIS PHASE
CONTSCN
H1
UPDATE POINTER TO NEXT CNTR DICT ENTRY
J1
END OF CNTR DICT REACHED
NO
YES
K1
RETURN CHART HB
* B2
THE STOW TABLE STARTS IMMEDIATELY AFTER THE CONTROL DICTIONARY. THE STOW TABLE IS THE INTERFACE WITH $MAINDIR.
* C2
NUMBER OF ENTRIES (ENTNUM) INITIALIZED WITH ZERO AND REQUEST NUMBER (REQNUM) INITIALIZED WITH ONE.
D2
INIT FOR ERROR MESSAGE '21861'
H3
STRTBLD
F2
MOVE INFORMATION FROM CNTR DICT TO STOW TABLE
G2
RETURN CHART HB
A3
BLDENT
BLDENT
B3
MODIFY LOAD AND TRANSFER ADDR BY LENGTH OF COMMON
C3
UPDATE NUMBER OF ENTRIES (ENTNUM) BY ONE
D3
TOO MANY PHASES FOR LINK
YES
NO
E3
PHASE PHASE *** DETECTED
NO
YES
F3
OPTION LINK SPECIFIED
YES
NO
G3
INIT FOR ERROR MESSAGE '21921'
H3
H3
ABTERR CHART AF
A4
UPDDIR
UPDDIR
B4
IS IT FIRST CALL OF $MAINDIR
NO
YES
C4
GET FIRST DOUBLE WORD BOUNDARY AFTER STOW TABLE
D4
TAKE IT AS LOAD ADDRESS FOR $MAINDIR AND LOAD IT
E4
CALCULATE ADDRESS OF WORKAREA FOR $MAINDIR *E5
F4
STORE INFORMATION FOR NEXT CALL OF $MAINDIR
CALMAIN
G4
SET UP REGISTER INTERFACE WITH $MAINDIR
H4
$MAINDIR *H5
UPDATE CIL DIRECTORY
J4
RETURN CODE IN REG 15 ZERO
NO
YES
A5
K4
START NEW STOW TABLE. REINITIALIZE HEADER
A5
A5
SETUP FOR MSG 3A70I
B5
ERROR AJ
PRINT SYSLOG + SYSLST
C5
CANCL CHART AH
* E5
THE END ADDRESSES OF THE STOW TABLE AND OF THE LAST PHASE LOADED ARE COMPARED. THE HIGHEST ADDRESS IS TAKEN AS WORKAREA ADDRESS
*H5
THIS PHASE IS DESCRIBED IN THE MANUAL "LIBRARIAN LOGIC"
K5
RETURN CHART HB

Chart HB. IJBCAT - Update CIL Directory (Part 2 of 2). Refer to Chart 12



* B4
BY SETTING BIT 7
IN JCSW2 IN COMREG

* C4
BY SETTING BIT 2
IN JCSW2 IN COMREG

*E2
THE STATUS TABLE
IS THE INTERFACE
WITH $LIBSTAT.

*F2
THE END ADDRESSES
OF THE STATUS TABLE
AND OF THE LAST PHASE
LOADED ARE COMPARED.
THE HIGHEST END ADDRESS
IS TAKEN AS WORKAREA
ADDRESS FOR $LIBSTAT.

Chart JA. IJBINL - Initialization (Part 1 of 2). Refer to Chart 01

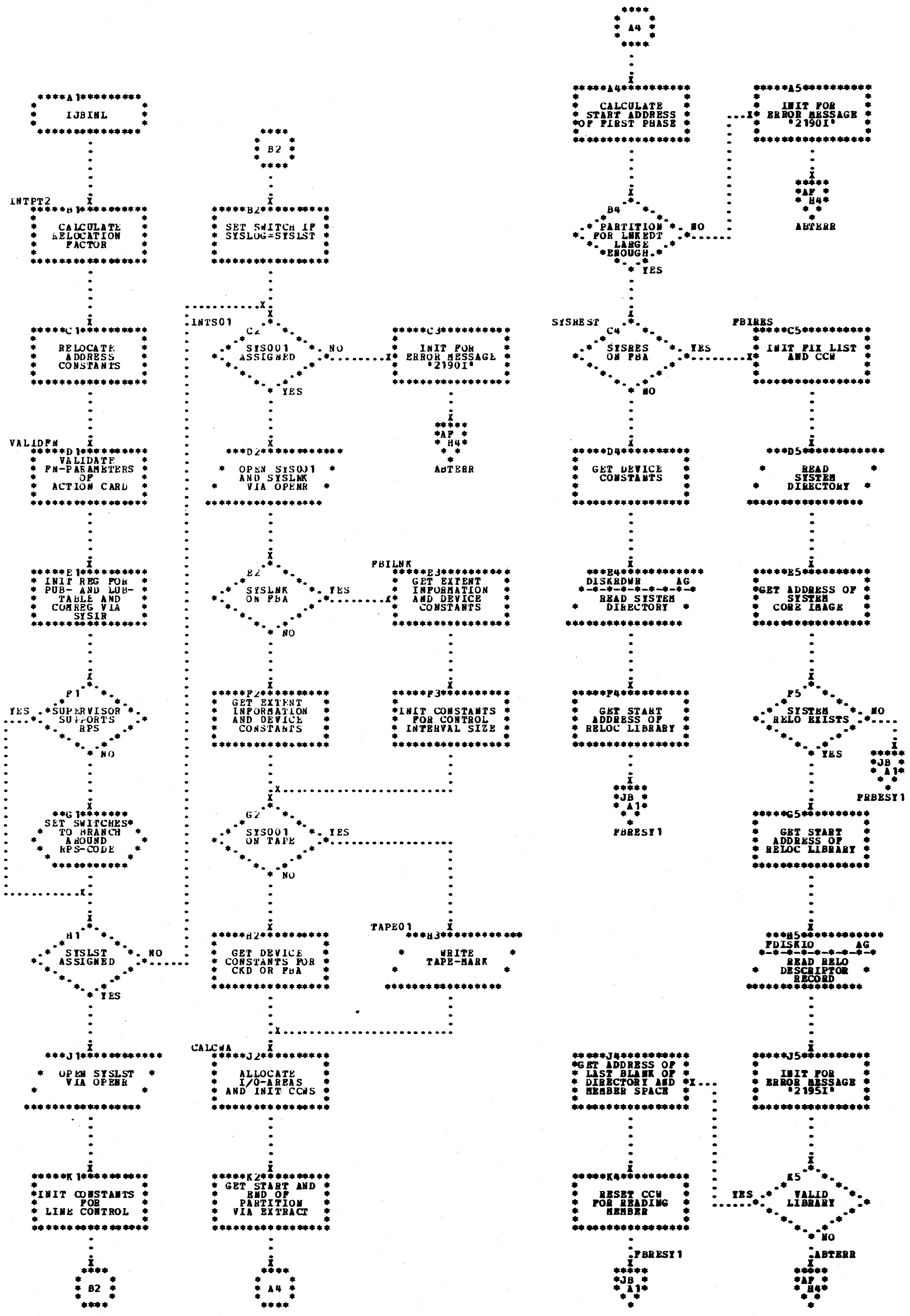

A1
IJBINL
B2
INTPT2
B1
CALCULATE RELOCATION FACTOR
C1
RELOCATE ADDRESS CONSTANTS
VALIDFN
D1
VALIDATE FN-PARAMETERS OF ACTION CARD
E1
INIT REG FOR PUB- AND LUB-TABLE AND COMREG VIA SYSIR
F1
SUPERVISOR SUPPORTS RPS
YES
NO
G1
SET SWITCHES TO BRANCH AROUND RPS-CODE
H1
SYSLST ASSIGNED
NO
YES
J1
OPEN SYSLST VIA OPENR
K1
INIT CONSTANTS FOR LINE CONTROL
B2
B2
SET SWITCH IF SYSLOG=SYSLST
INTS01
C2
SYS001 ASSIGNED
NO
YES
C3
INIT FOR ERROR MESSAGE '2190I'
AF H4
ABTERR
D2
OPEN SYS001 AND SYSLNK VIA OPENR
E2
SYSLNK ON FBA
YES
NO
FBILNK
E3
GET EXTENT INFORMATION AND DEVICE CONSTANTS
F2
GET EXTENT INFORMATION AND DEVICE CONSTANTS
F3
INIT CONSTANTS FOR CONTROL INTERVAL SIZE
G2
SYS001 ON TAPE
YES
NO
H2
GET DEVICE CONSTANTS FOR CKD OR FBA
TAPE01
H3
WRITE TAPE-MARK
CALCWA
J2
ALLOCATE I/O-AREAS AND INIT CCWS
K2
GET START AND END OF PARTITION VIA EXTRACT
A4
A4
A4
CALCULATE START ADDRESS OF FIRST PHASE
A5
INIT FOR ERROR MESSAGE '2190I'
AF H4
ABTERR
B4
PARTITION FOR LNKEDT LARGE ENOUGH
NO
YES
SYSREST
C4
SYSRES ON FBA
YES
NO
FBIRES
C5
INIT FIX LIST AND CCW
D4
GET DEVICE CONSTANTS
D5
READ SYSTEM DIRECTORY
E4
DISKRDWR AG
READ SYSTEM DIRECTORY
E5
GET ADDRESS OF SYSTEM CORE IMAGE
F4
GET START ADDRESS OF RELOC LIBRARY
JB A1
FBRESY1
F5
SYSTEM RELO EXISTS
NO
YES
JB A1
FBRESY1
G5
GET START ADDRESS OF RELOC LIBRARY
H5
FDISKIO AG
READ RELO DESCRIPTOR RECORD
J4
GET ADDRESS OF LAST BLANK OF DIRECTORY AND MEMBER SPACE
J5
INIT FOR ERROR MESSAGE '2195I'
K4
RESET CCW FOR READING MEMBER
FBRESY1
JB A1
K5
VALID LIBRARY
YES
NO
ABTERR
AF H4

Chart JB. IJBINL - Initialization (Part 2 of 2). Refer to Chart 01

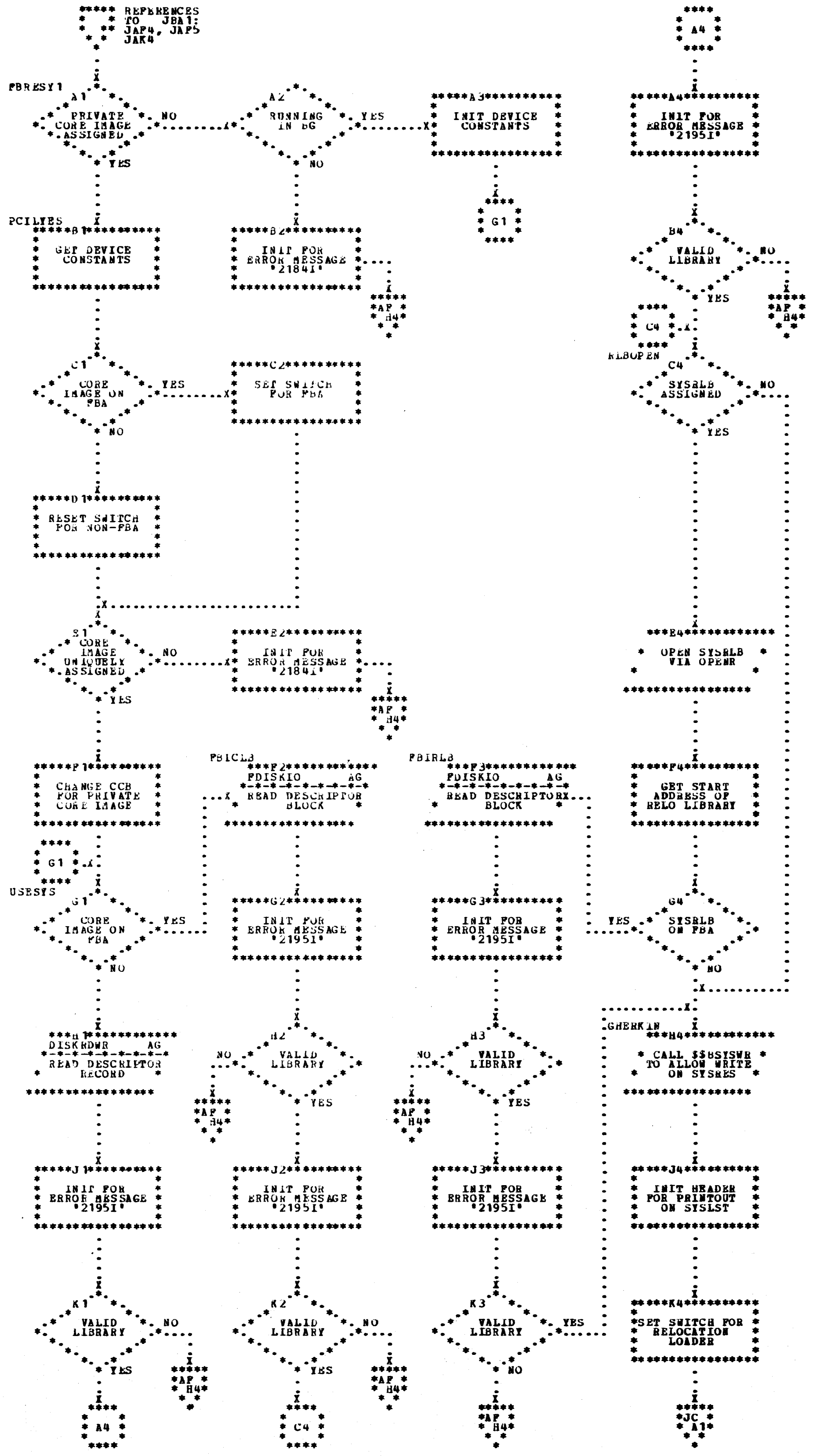


Chart JC. IJBINL - Action Processor (Part 1 of 2) Refer to Chart 01

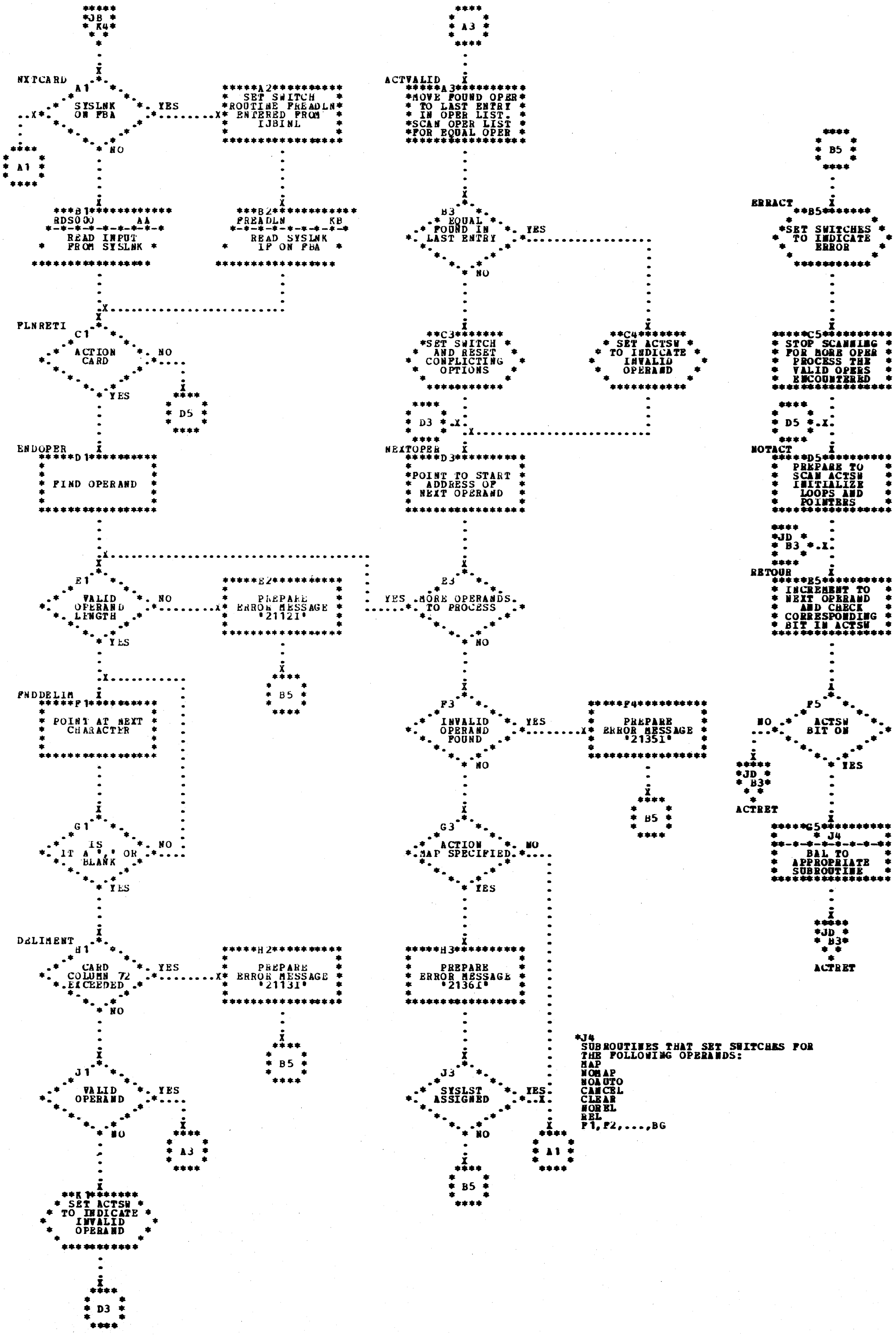

JB K4
NXTCARD
A1
SYSLNK ON FBA
YES
NO
A1
A2
SET SWITCH ROUTINE PREADLN ENTERED FROM IJBINL
B1
RDS000 AA
READ INPUT FROM SYSLNK
B2
PREADLN KB
READ SYSLNK IF ON FBA
PLNRETI
C1
ACTION CARD
NO
YES
D5
ENDOPER
D1
FIND OPERAND
E1
VALID OPERAND LENGTH
NO
YES
E2
PREPARE ERROR MESSAGE '2112I'
B5
FNDDELIM
F1
POINT AT NEXT CHARACTER
G1
IS IT A ',' OR BLANK
NO
YES
DELIMENT
H1
CARD COLUMN 72 EXCEEDED
YES
NO
H2
PREPARE ERROR MESSAGE '2113I'
B5
J1
VALID OPERAND
YES
NO
A3
K1
SET ACTSW TO INDICATE INVALID OPERAND
D3
A3
ACTVALID
A3
MOVE FOUND OPER TO LAST ENTRY IN OPER LIST. SCAN OPER LIST FOR EQUAL OPER
B3
EQUAL FOUND IN LAST ENTRY
YES
NO
C3
SET SWITCH AND RESET CONFLICTING OPTIONS
C4
SET ACTSW TO INDICATE INVALID OPERAND
D3
NEXTOPER
D3
POINT TO START ADDRESS OF NEXT OPERAND
E3
YES
MORE OPERANDS TO PROCESS
NO
F3
INVALID OPERAND FOUND
YES
NO
F4
PREPARE ERROR MESSAGE '2135I'
B5
G3
ACTION MAP SPECIFIED
NO
YES
H3
PREPARE ERROR MESSAGE '2136I'
J3
SYSLST ASSIGNED
YES
NO
B5
A1
B5
ERRACT
B5
SET SWITCHES TO INDICATE ERROR
C5
STOP SCANNING FOR MORE OPER PROCESS THE VALID OPERS ENCOUNTERED
D5
NOTACT
D5
PREPARE TO SCAN ACTSW INITIALIZE LOOPS AND POINTERS
JD B3
RETOUR
E5
INCREMENT TO NEXT OPERAND AND CHECK CORRESPONDING BIT IN ACTSW
F5
ACTSW BIT ON
NO
YES
JD B3
ACTRET
G5
J4
BAL TO APPROPRIATE SUBROUTINE
JD B3
ACTRET
*J4
SUBROUTINES THAT SET SWITCHES FOR THE FOLLOWING OPERANDS:
MAP
NOMAP
NOAUTO
CANCEL
CLEAR
NOREL
REL
F1,F2,...,BG

Chart JD. IJBINL - Action Processor (Part 2 of 2) Refer to Chart 02

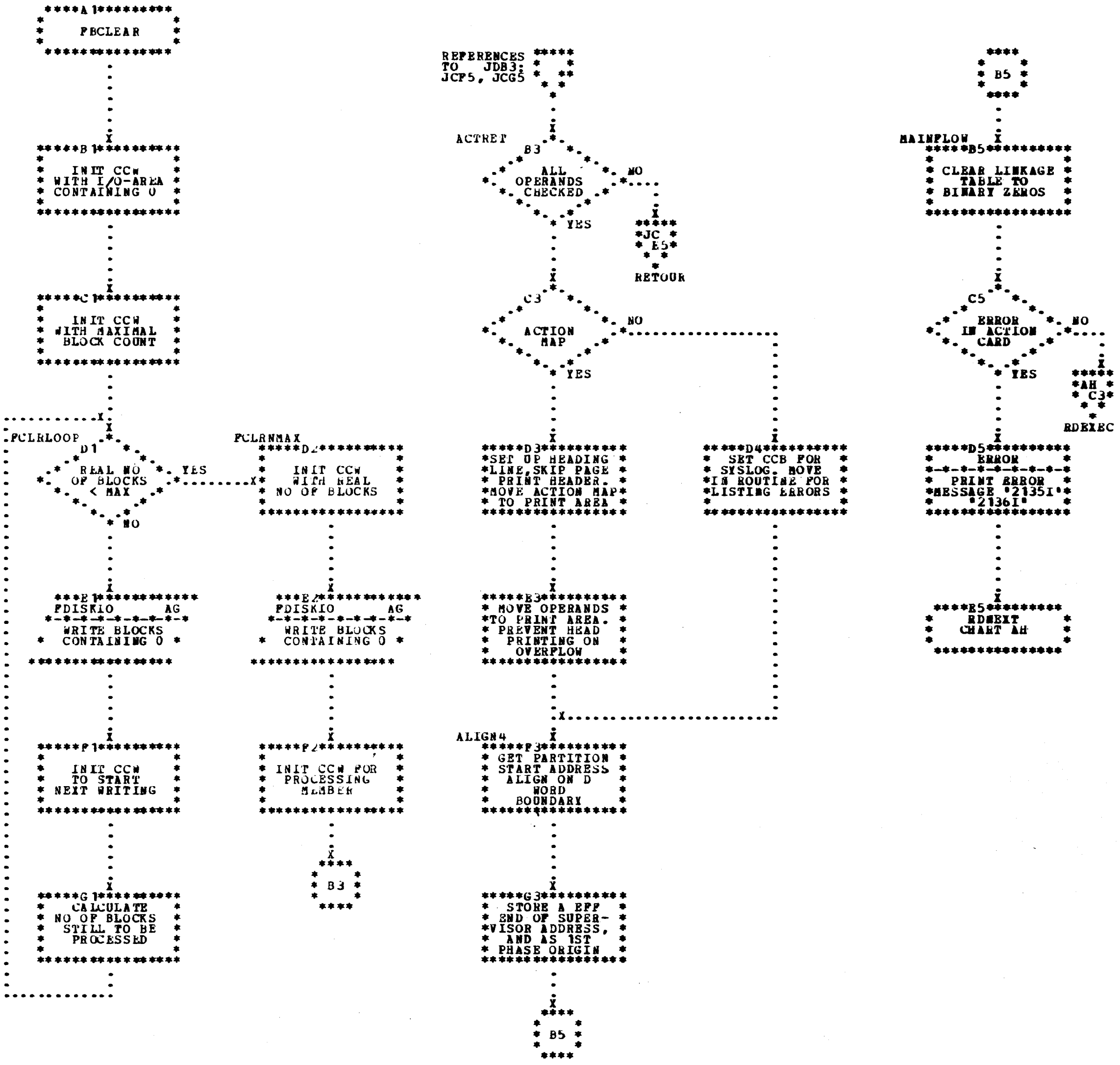

A1
FBCLEAR
B1
INIT CCW WITH I/O-AREA CONTAINING 0
C1
INIT CCW WITH MAXIMAL BLOCK COUNT
PCLRLOOP
D1
REAL NO OF BLOCKS < MAX
YES
NO
PCLRNMAX
D2
INIT CCW WITH REAL NO OF BLOCKS
E1
PDISKIO AG
WRITE BLOCKS CONTAINING 0
E2
PDISKIO AG
WRITE BLOCKS CONTAINING 0
F1
INIT CCW TO START NEXT WRITING
F2
INIT CCW FOR PROCESSING MEMBER
B3
G1
CALCULATE NO OF BLOCKS STILL TO BE PROCESSED
REFERENCES TO JDB3: JCF5, JCG5
ACTRET
B3
ALL OPERANDS CHECKED
NO
JC E5
RETOUR
YES
C3
ACTION MAP
NO
YES
D3
SET UP HEADING LINE, SKIP PAGE PRINT HEADER. MOVE ACTION MAP TO PRINT AREA
D4
SET CCB FOR SYSLOG. MOVE IN ROUTINE FOR LISTING ERRORS
E3
MOVE OPERANDS TO PRINT AREA. PREVENT HEAD PRINTING ON OVERFLOW
ALIGN4
F3
GET PARTITION START ADDRESS ALIGN ON D WORD BOUNDARY
G3
STORE A EFF END OF SUPER- VISOR ADDRESS, AND AS 1ST PHASE ORIGIN
B5
B5
MAINFLOW
B5
CLEAR LINKAGE TABLE TO BINARY ZEROS
C5
ERROR IN ACTION CARD
NO
AH C3
RDEXEC
YES
D5
ERROR
PRINT ERROR MESSAGE '2135I' '2136I'
E5
RDEXIT CHART AH

Chart KA. IJBFIN - Process Input from FBA-Relocatable Library. Refer to Chart 03

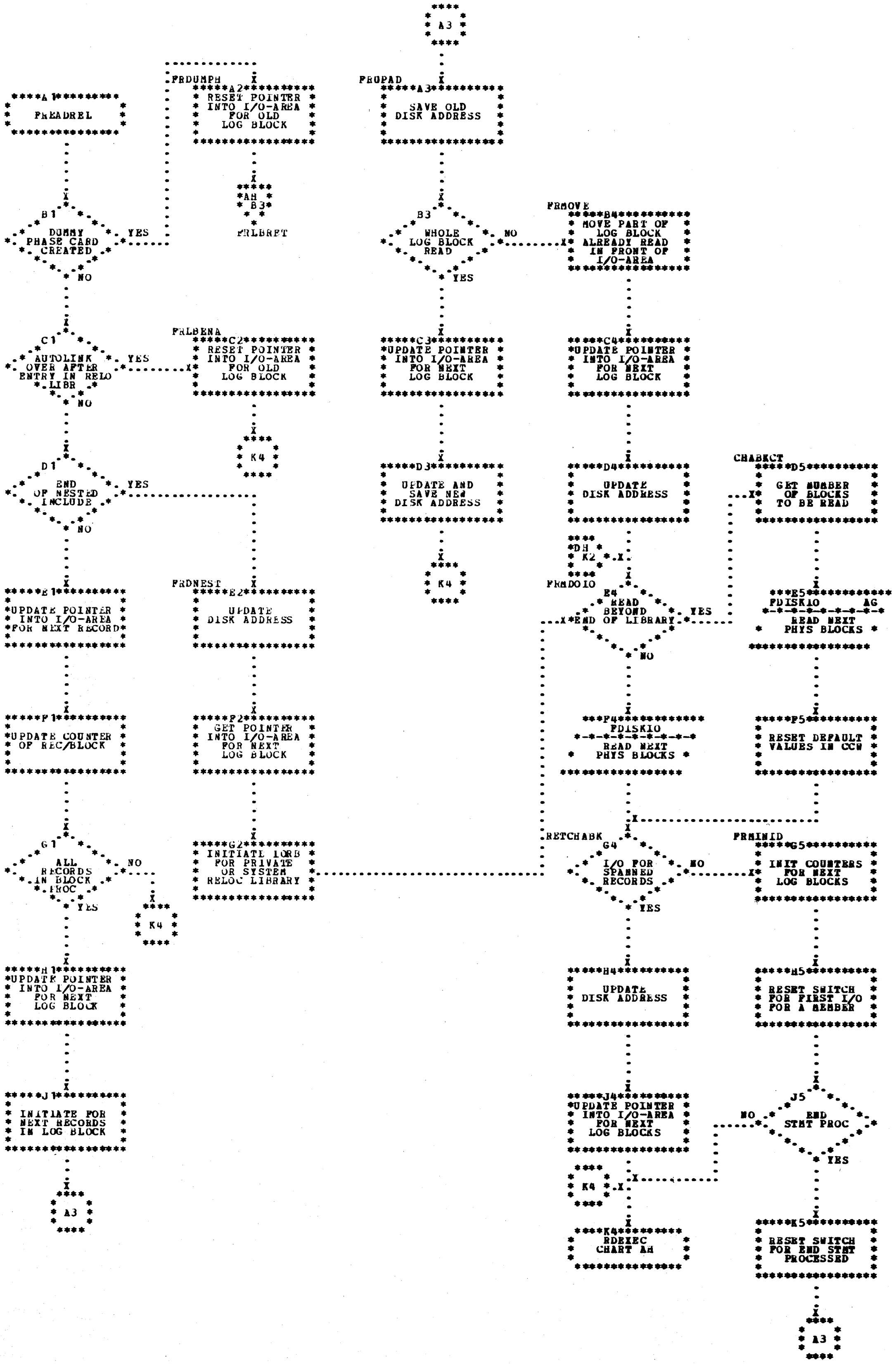
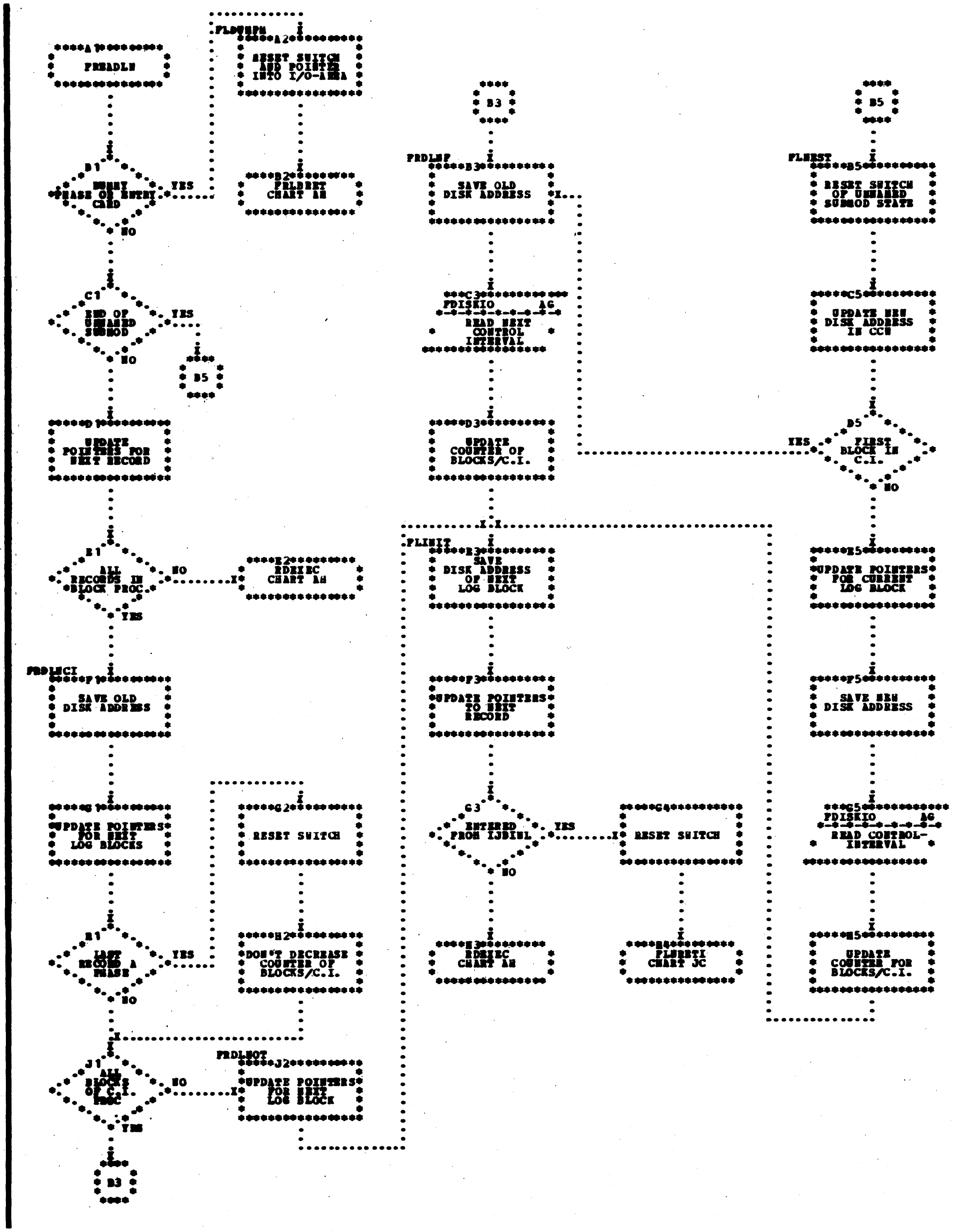

A3
A1
FREADREL
FRDUMPH
A2
RESET POINTER INTO I/O-AREA FOR OLD LOG BLOCK
AH B3
FRLBRPT
FROPAD
A3
SAVE OLD DISK ADDRESS
B1
DUMMY PHASE CARD CREATED
YES
NO
B3
WHOLE LOG BLOCK READ
NO
YES
FRMOVE
B4
MOVE PART OF LOG BLOCK ALREADY READ IN FRONT OF I/O-AREA
C1
AUTOLINK OVER AFTER ENTRY IN RELO LIBR
YES
NO
FRLBENA
C2
RESET POINTER INTO I/O-AREA FOR OLD LOG BLOCK
K4
C3
UPDATE POINTER INTO I/O-AREA FOR NEXT LOG BLOCK
C4
UPDATE POINTER INTO I/O-AREA FOR NEXT LOG BLOCK
D1
END OF NESTED INCLUDE
YES
NO
D3
UPDATE AND SAVE NEW DISK ADDRESS
K4
D4
UPDATE DISK ADDRESS
DH K2
CHABKCT
D5
GET NUMBER OF BLOCKS TO BE READ
E1
UPDATE POINTER INTO I/O-AREA FOR NEXT RECORD
FRDNEST
E2
UPDATE DISK ADDRESS
FRMDOIO
E4
READ BEYOND END OF LIBRARY
YES
NO
E5
FDISKIO AG
READ NEXT PHYS BLOCKS
F1
UPDATE COUNTER OF REC/BLOCK
F2
GET POINTER INTO I/O-AREA FOR NEXT LOG BLOCK
F4
FDISKIO
READ NEXT PHYS BLOCKS
F5
RESET DEFAULT VALUES IN CCW
G1
ALL RECORDS IN BLOCK PROC
NO
YES
K4
G2
INITIATE IORB FOR PRIVATE OR SYSTEM RELOC LIBRARY
RETCHABK
G4
I/O FOR SPANNED RECORDS
NO
YES
FRMINID
G5
INIT COUNTERS FOR NEXT LOG BLOCKS
H1
UPDATE POINTER INTO I/O-AREA FOR NEXT LOG BLOCK
H4
UPDATE DISK ADDRESS
H5
RESET SWITCH FOR FIRST I/O FOR A MEMBER
J1
INITIATE FOR NEXT RECORDS IN LOG BLOCK
A3
J4
UPDATE POINTER INTO I/O-AREA FOR NEXT LOG BLOCKS
J5
END STMT PROC
NO
YES
K4
K4
RDEXEC CHART AH
K5
RESET SWITCH FOR END STMT PROCESSED
A3

Chart KB. IJBFIN - Process Input from FEA-SYSLNK Refer to Chart 03


A1
PREADLN
B1
DUMMY PHASE OR ENTRY CARD
YES
NO
FLDUMPH
A2
RESET SWITCH AND POINTER INTO I/O-AREA
B2
FRLDRET CHART AH
C1
END OF UNNAMED SUBMOD
YES
NO
B5
D1
UPDATE POINTERS FOR NEXT RECORD
E1
ALL RECORDS IN BLOCK PROC.
NO
YES
E2
RDEXEC CHART AH
FRDLNCI
F1
SAVE OLD DISK ADDRESS
G1
UPDATE POINTERS FOR NEXT LOG BLOCKS
G2
RESET SWITCH
H1
LAST RECORD A PHASE
YES
NO
H2
DON'T DECREASE COUNTER OF BLOCKS/C.I.
J1
ALL BLOCKS OF C.I. PROC
NO
YES
FRDLNOT
J2
UPDATE POINTERS FOR NEXT LOG BLOCK
B3
B3
FRDLNP
B3
SAVE OLD DISK ADDRESS
C3
FDISKIO AG
READ NEXT CONTROL INTERVAL
D3
UPDATE COUNTER OF BLOCKS/C.I.
FLINIT
E3
SAVE DISK ADDRESS OF NEXT LOG BLOCK
F3
UPDATE POINTERS TO NEXT RECORD
G3
ENTERED FROM IJBINL
YES
NO
G4
RESET SWITCH
H3
RDEXEC CHART AH
H4
FLNRETI CHART JC
B5
FLNRST
B5
RESET SWITCH OF UNNAMED SUBMOD STATE
C5
UPDATE NEW DISK ADDRESS IN CCW
D5
FIRST BLOCK IN C.I.
YES
NO
E5
UPDATE POINTERS FOR CURRENT LOG BLOCK
F5
SAVE NEW DISK ADDRESS
G5
FDISKIO AG
READ CONTROL-INTERVAL
H5
UPDATE COUNTER FOR BLOCKS/C.I.

APPENDIX A: LABEL LIST

This label list has two parts: One lists the entry labels and summarizes the function of the CSECTs the labels lead to; the other one lists all the labels occurring in the flowcharts.

Part 1: Entry Labels

Label	CSECT	Chart	
ABORT		GE	Beginning of non-recoverable error handling subroutine.
ABTERR		AF	Gives control to IJBRLD so that abort error processing can continue. Entered when non-recoverable errors are found.
ACSLTH		CG	Some compilers supply control section length in the END card. This routine processes the control section length for this special case.
ADMDSW	IJBMAP	FA	Program switch to indicate first time through the MAP processor.
ALNKPR	IJBLNK	AG	Starting address of the autolink processor, which is entered whenever a phase has finished processing and autolink has been requested.
CDSIZE	IJBLNK	AF	Starting label of a subroutine to check for control dictionary overflow.
CHKNPH	IJBESD	BA	Beginning of the ESD processor phase, IJBESD (see "Appendix E" for a detailed description of this phase).
CINOBL	IJBCTL	EE	Number of core image blocks required by this phase is equal to number of bytes loaded, divided by block size. Add one block for any remainder.
CNCALK	IJBESD	BC	The instructions starting at this label cancel autolink. The NOAUTCL switch in CSWITCH (current CID entry) is set to indicate that no autolink should be attempted on this ER, either because autolink has already been tried or because autolink should not be performed.
CNVHEX	IJBLNK	AC	Label at the beginning of a subroutine that converts hexadecimal to binary notation.
COMCHK	IJBMAP	FA	Beginning of common (CM) processing in the MAP processor. The cumulative length of discretely named commons is calculated at this point in the program.
DECDSP	INBSCN	DD	Beginning of decimal displacement processing for the operand of a phase card.
DEBDAD	IJBLNK	AK	Beginning of subroutine that builds core image blocks. Basically, this subroutine: 1. Ensures that the text is within the limits of the phase; 2. Finds the core image block the text belongs in; 3. Reads the core image block required by the text into the I/O area.
DEBDSW	INBLNK	AK	Program switch that forces continued processing when a zero length control section is found by the ESD processor (Chart BE).
DMPHSW	IJBLNK	AH	Program switch initialized as an MVC instruction. Modified to an effective NOP by the ESD processor (Chart BA) when a dummy phase card is to be built. By the NOP modification, the disk address of the ESD card not yet processed is retained in location COMNRF for use after the phase processing is finished. (Dummy phase cards are treated as actual phase cards.)
DSPRTN	IJBSCN	DC	Beginning of displacement operand

Label CSECT Chart

processing.

ELBCM IJBESD ED
Beginning of CM (common) processing. Whenever both the ESD item in the card image and the control dictionary entry are commons with matching labels, this routine puts the common with the longest length in the control dictionary. If commons with different labels are found, this routine posts the new common in the control dictionary. The map processor of the linkage editor calculates a total length of commons, which adjusts the phase origin.

ENDRTN IJBOTH CF
Tests SBMDST to determine if the name list should be cleared. (See SBMDST in label list.)

ENDSBM IJBOTH CG
Program switch providing an exit from the END card processor. The switch is assembled in the NOP state, initialized to branch by the END card processor, and reset to NOP by the END card processor if the module being processed has been autolinked.

ENDSCD IJBOTH CG
Searches control dictionary for unassigned LD/LRs, and ensures that the control dictionary number for such items is negative.

ENTALK IJBSCN DD
Provides an exit from the $LNKEDT4 phase of the linkage editor when a blank ENTRY card is found.

ENTCRD IJBSCN DB
Beginning of the entry card processor.

Note: At this time, a blank operand field would be detected in the skip blanks subroutine. (See SKIPB and ENTALK in the label list.)

ERRACT IJBINL JC
Common error exit from the action processor. Error messages are initialized in the action processor, but are actually issued during the execution of another initialization routine (see Chart AC).

ERRNML IJBSCN DA
Program switch initialized to branch. However, if an error occurs on an include card and the include card has created a name list, ERRNML is set to NOP. This allows the name list to be cleared before card error processing.

ESDRET IJBESD EA
Common entry point when the processing of an ESD item is finished. Instructions starting at this label determine if any more ESD items are in the variable field of the card image.

ESLBCD IJBESD ED
Beginning at this label, the ESD processor searches the control dictionary for a label that corresponds to the label of the ESD item in the current control dictionary entry.

EXLOAD IJBLNK AH

1. Entry point from the autolink routine (Chart AJ) when the control card processing phase is to be loaded.
2. Entry point from the ESD processor when a dummy phase card has been built and a control card processing phase is to be loaded.
3. Entry point from the initialize IJBOTH routine when the ESD processing phase is to be loaded.

FCHRLD IJBMAP FD
Entry point when NOMAP option is found.

INCLOP IJBSCN DG
Beginning of the relocatable directory scan for autolink, include, and terminal processing. The scan looks for a module with a name that matches the card image name field.

ISDISP IJBCTL EC
Processing when the phase card operand is a displacement.

ISROOT IJBCTL EC
Processing when the phase card operand is ROOT.

LABST IJBSCN DB
Builds the name list of control sections from the operand field of an INCLUDE card. These control sections subsequently build a phase (see NMELST in the label list).

LTCDNO IJBLNK AB
Entry point to the control dictionary search subroutine when a test for control dictionary number assignment is required by the calling

Label CSECT Chart

routine. The control dictionary number was available, but because its status was undetermined, entry is made to this subroutine.

LTESID IJBLNK AB

Beginning of the subroutine that finds control dictionary number, control dictionary address, or relocation factor, using the ESID number from the language translator and the linkage table constructed by the linkage editor. LTESID is the entry point when the ESID number is supplied.

MAPCST IJBMAP FB

Sets up for a scan of the control dictionary, searching for control sections that belong to the phase previously identified in this routine.

MAPHAS IJBMAP FB

Beginning of a control dictionary scan that searches for phase entries.

MAPLDR IJBMAP FC

Sets up for scan of control dictionary that searches for LD/LRs.

NMELST *

A list of control sections specified by the name list operand of an INCLUDE card. The list is blank except when a named submodule is still being processed.

NMSBSW *

Supplies the information in byte 4 of location PERIDA during INCLUDE card processing in the IJBSCN phase. Resets to zero during the execution of the control card scan in the IJBSCN phase. Bit 6 is set to one during initial INCLUDE card processing, and Bit 1 is set during phase post processing (autolink mode).

NOBLOK *

Label in control dictionary entry.

NOBYTE *

Label in control dictionary entry.

CRPHDA *

Label in control dictionary entry.

CRPHRG *

Label in control dictionary entry.

PERIDA *

The location labeled PERIDA is a 36-byte input control area used by the linkage editor program to:

- obtain the address of the next card image to be processed after the END card;
- determine the point at which processing is finished for an object module;
- maintain control over the nesting of include statements by functioning as last-in, first-out list to establish processing priorities.

Location PERIDA is used in conjunction with either location ESD000 or ESDN00 (see label list) depending on the input device being used at this time. ESD000 or ESDN00 is loaded with the disk address of the first ESD card image of the object module. PERIDA is loaded with the disk address of the card image that follows the control card image. The linkage editor program compares the disk address in location PERIDA with the address in either ESD000 or ESDN00. Input control is based on the result of the comparison made at END card time. Possible results and corresponding input control actions are:

- The address in PERIDA is equal to or higher than the address in ESD000. Process the card image sequentially following the END card.
- The address in PERIDA is lower than the address in ESD000. Get the address of next card image to be processed from PERIDA.
- The address in PERIDA is lower than the address in ESDN00. Get the address of the next card image to be processed from PERIDA.
- The address in PERIDA is equal to or higher than the address in ESDN00. Effectively shift PERIDA left six bytes. Get the address of the next card image to be processed from the updated PERIDA.

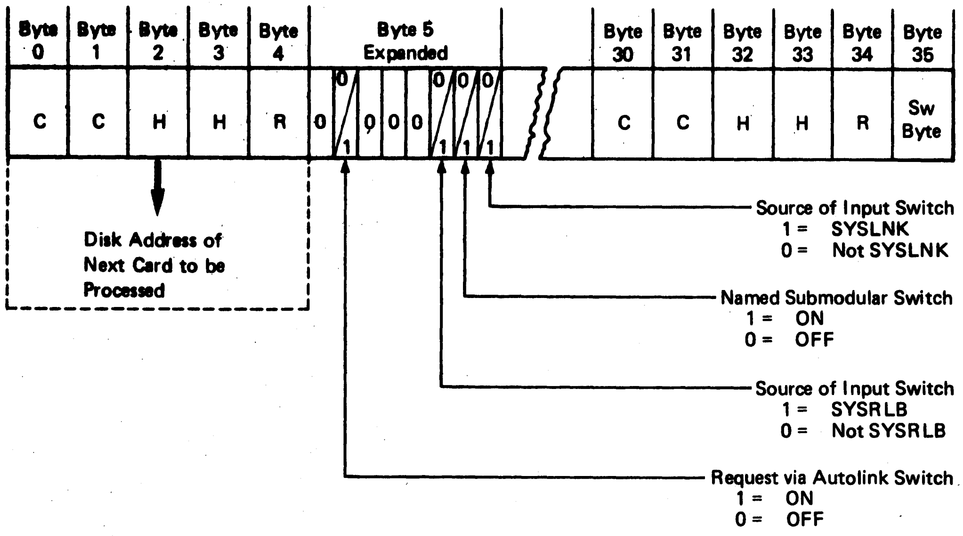
Before the comparison is made and the appropriate actions are taken at END card time, the linkage editor program ensures that a value is available for PERIDA (see RECF00 in this list). Location PERIDA establishes processing priority by functioning as a last-in, first-out list for up to five levels of include (nest depth). The list is built during

----

* Listing only

the execution of the include card processor (Chart DB). Figure 10 illustrates the physical structure of PERIDA and Figure 11 illustrates how this location functions as a last-in, first-out list.

Note: If all five levels of include are used, the last 6-byte segment of PERIDA contains the address of the card image following the first INCLUDE statement.



Note: Input from system relocatable library if byte 5 bits 5 and 7 are both 0.

Figure 10. PERIDA Layout

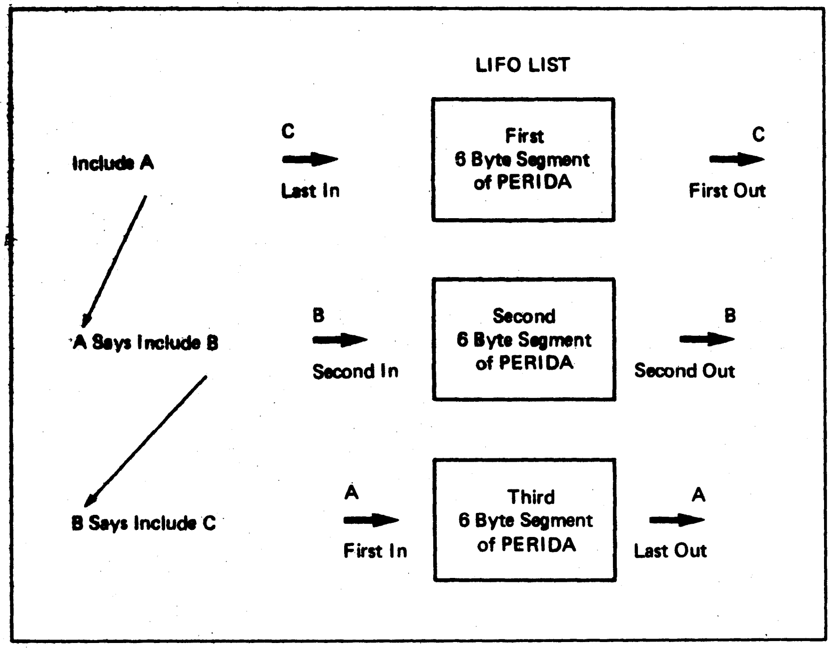


Figure 11. Last-In, First-Out List (LIFO)

Label CSECT Chart

PHADMD IJBMAP FB
Adjusts load address of the phase by the cumulative length of the discretely named commons.

PHCRD IJBSCN DC
Beginning of the phase processor in the IJBSCN phase. This part of the phase processor performs two basic functions:
1. Determines which optional operand has been used.
2. Validity checks the phase card image.

PHSTOR IJBMAP FA
Reinitializes the phase information in location PHEADR (see label list).

PVLBSW IJBLNK AH
Switch set to NOP by END and INCLUDE card processors if input is from SYSRLB.

QUAPRO IJBSCN DD
Beginning of qualifier processing for the operand of a phase card.

RDALSW IJBLNK AH
Switch set to branch if Autolink is in operation.

RECF00 IJBLNK AA
Program switch set to branch and reset to NOP by either the ESD processor (Chart BA) or the END card processor (Chart CF). The ESD processor ensures that the correct disk address is in location PERIDA (see label list) by branching to the read SYSLNK subroutine at location INS000 after the CCW has been set to NOP. The disk address of the next card image is located by updating the disk until a record is found. Register 2 supplies the correct disk address for location PERIDA. The END card processor used the same technique to locate the disk address of the next card to be processed (put into location NDS000). RECF00 is set to branch to prevent updating beyond the proper disk address.

RELBSW IJBLNK AH
Program switch that tests for input from the relocatable library. Sets (branch) in the include processor and resets (NOP) in the END card processor.

RLDPRC IJBOTH CD
Beginning of RLD pass 1 processing.

RLDRAG IJBRLD GA
Reads RLD information, supplied by pass 1, from SYS001.

RLDSW1 IJBRLD GA
Program switch set to branch within this phase, whenever pointer processing is finished.

RLDSW2 IJBRLD GB
Program switch initialized to branch, calculates load address (assembled origin of control dictionary entry plus relocation factor) when set to NOP in this phase.

RLDSW3 IJBRLD GC
Initialized to NOP. Set to branch within this phase whenever R and P pointers indicate wrong phase.

RLDSW4 IJBRLD GC
Program switch initialized to NOP, indicating the ADCON is to be extracted from the core image block. If the switch is set to branch, the ADCON is replaced in the core image block.

RLSW1 IJBOTH CD
Program switch set and reset within IJBOTH phase. Setting (NOP or branch) determines if the R and P pointers are to be processed. Several RLD items can habe the same R and P pointers. Only the first set of identical R and P pointers are processed.

RLWRIT IJBOTH CE
Program switch set to NOP in the pass 1 RLD processor when the RLD is to be processed in the pass 2 RLD processor. If the switch is set to branch (initial condition), the RLD is ignored.

ROOTNO *
This location contains a zero when the first phase is not specified root, and a one when it is a root. The value in ROOTNO is either added to or subtracted from the control dictionary number.
- Subtracted - when the control dictionary number is used to obtain a control dictionary entry address.

  (C/D NO - ROOTNO) X16 =
  DISPLACEMENT
  CDENT1 + DISPLACEMENT =
  ENTRY ADDRESS

----

* Listing only

Label CSECT Chart

- Added - when a root phase has been specified, a one is added for each control dictionary entry.

SBMDST *

A program switch that indicates when NMELST should be cleared (Bit 7) or an END card has been read (Bit 3). Bit 7 is turned on when a named submodule (INCLUDE NAME, (CSECT)) is found. At the same time, a bit switch in location PERIDA (see label list) is turned on. The apparent duplication of switches is necessary because the first 5-byte segment of PERIDA is a variable, which depends on nested levels of INCLUDEs. At END card time, a test is made of the bit switch in location PERIDA.

If bit 6 of byte 4 in PERIDA is on, turn off bit 7 in SBMDST. The switch in PERIDA can then be tested.

If bit 6 of byte 4 in PERIDA is off, do not change the status of SBMDST. The bit switch has already been moved to some other 5-byte segment of PERIDA. The linkage editor program can then test SBMDST to determine if the END card being processed is part of the module named in the INCLUDE statement (bit 7 in SBMDST off). The name list (NMELST) is cleared at END card time, except when a named submodule still being processed.

Label CSECT Chart

SRCHCD IJBLNK AC

Beginning of a subroutine that finds the last duplicate label in the control dictionary.

SRPCOD IJBLNK AC

Tests for end of control dictionary. Entry point for the label-search subroutine.

TRFRAD *

Label in control dictionary entry.

TSTCNT IJBRLD GD

Sets up MAP information (number of ADCONS outside the phase limits) in a test register. If the register is zeroed, there is no MAP information. If the content is nonzero, MAP information is to be printed.

UPDSKAD IJBLNK AE

Updates the disk address to the first record on the next track.

VERLOP IJBRLD GF

Beginning of a loop that reads and verifies all core image blocks written by linkage editor. All verification occurs at this point, rather than after each individual write operation.

XTPHNO IJBLNK AF

Beginning of a subroutine that extracts the phase number from the control dictionary.

Part 2: All Labels

As there is only one phase, $LNKEDT, in this program, "CSECT" here indicates the CSECT of the general phase.

Label	CSECT	Location	Label	CSECT	Location
ABORT	IJBRLD	GEB5	DERITE	IJBLNK	AKE3
ABORT10	IJBRLD	GEH5	DERITE1	IJBLNK	AKF3
ABTERR	IJBLNK	AFH4	DERLOP	IJBLNK	AKJ3
ACSLTH	IJBOTH	CGB4	DERSW1	IJBLNK	AKK1
ACTRET	IJBINL	JDB3	DISKIO	IJBLNK	AGE1
ACTVALID	IJBINL	JCA3	DISKRDWR	IJBLNK	AGD1
ADMDSW	IJBMAP	FAD3	DMPHSW	IJBLNK	AHD3
AD1DSK	IJBLNK	AEC3	DPLAB	IJBMAP	FDH1
ALIGNDW	IJBCTL	ECD4	DSPRTN	IJBSCN	DCH1
ALIGN4	IJBINL	JDF3			
ALKERR	IJBSCN	DGA4	EISCSL	IJBOTH	CGF4
ALKFND	IJBSCN	DHA1	EISDPC	IJBESD	BCB3
ALNKOD	IJBLNK	AGF4	ELBCER	IJBESD	BDD3
ALNKOF	IJBLNK	AHB1	ELBCM	IJBESD	BDC2
ALNKPR	IJBLNK	AGB4	ELBDSD	IJBESD	BDF5
ALNKSC	IJBLNK	AGE4	ELBELR	IJBESD	BCH5
ALNKVL	IJBLNK	AGJ5	ELBER	IJBESD	BCB4
			ELBGSD	IJBESD	BDJ5
BLDENT	IJBCAT	HAB3	ELBINT	IJBESD	BFE1
BUILD	IJBCAT	HAG1	ELBISITW	IJBESD	BCE5
			ELBLD	IJBESD	BEA1
CALCWA	IJBINL	JAJ2	ELBLDR	IJBESD	BEB3
CALMAIN	IJBCAT	HAG4	ELBNAS	IJBESD	BEF4
CANCL	IJBLNK	AHB5	ELBNCD	IJBESD	BFB1
CANCLE	IJBRLD	GEJ5	ELBNLR	IJBESD	BED2
CDSIZE	IJBLNK	AFB4	ELBPC	IJBESD	BDE1
CHABKCT	IJBFIN	KAD5	ELBSD	IJBESD	BDC4
CHEKQU	IJBCTL	EBB3	ENCRLT	IJBOTH	CGH1
CHKNPH	IJBESD	BAB1	ENDERR	IJBLNK	AJH4
CHKRP	IJBSCN	DBE3	ENDOPER	IJBINL	JCD1
CHKRPN	IJBSCN	DDD5	ENDPRC	IJBOTH	CFA1
CHKSYM	IJBLNK	AHJ3	ENDRTN	IJBOTH	CFH2
CHQUALIF	IJBCTL	ECD2	ENDSBM	IJBOTH	CGH4
CIMBLK	IJBCTL	EEH4	ENDSCD	IJBOTH	CGB1
CINOBL	IJBCTL	EEB1	ENDXFP	IJBOTH	CFA4
CLRCMN	IJBMAP	FAE2	ENLD	IJBESD	BBB4
CLREXT	IJBOTH	CHE3	ENOT00	IJBOTH	CFA2
CMDEL	IJBSCN	DEJ4	ENOXFR	IJBOTH	CFC5
CNCALK	IJBESD	BCA1	ENTALK	IJBSCN	DDH4
CNTSW	IJBOTH	CDF1	ENTCRD	IJBSCN	DBB4
CNVAHX	IJBLNK	ACJ3	ENUNAS	IJBOTH	CGD1
CNVBIN	IJBMAP	FEB4	EPHSCD	IJBESD	BFK4
CNVHEX	IJBLNK	ACB4	EPHSCN	IJBESD	BFF3
CNVHSW	IJBLNK	ACG3	EPHULD	IJBESD	BFD4
CNVLOP	IJBMAP	FED4	ERRACT	IJBINL	JCB5
CNVSHF	IJBLNK	ACH3	ERRNML	IJBSCN	DAF2
COMCHK	IJBMAP	FAG1	ERROR	IJBLNK	AJB3
CONTSCN	IJBCAT	HAH1	ERROR10	IJBLNK	AJJ2
CQUAL	IJBCTL	ECA2	ERROR20	IJBLNK	AJJ3
CRDEND	IJBSCN	DDB4	ERROR40	IJBESD	BBB5
CTLSKP	IJBLNK	AHG2	ERROR40	IJBLNK	AJF4
CTNOBL2	IJBCTL	EEC2	ERROR40	IJBLNK	AJK4
			ERROR50	IJBLNK	AJF5
DECDSP	IJBSCN	DDA1	ERR000	IJBOTH	CAC5
DELEXT	IJBSCN	DCH4	ERR002	IJBLNK	ACC4
DELIMENT	IJBINL	JCH1	ERR012	IJBSCN	DEF4
DERCAL	IJBLNK	AKH3	ERR013	IJBSCN	DEE5
DERDAD	IJBLNK	AKB1	ERR013A	IJBOTH	CCH5
DERDOK	IJBLNK	AKD1	ERR014	IJBSCN	DBF2
DERDSW	IJBLNK	AKH1	ERR022	IJBCTL	EBD3
			ERR023	IJBCTL	EDD2

Label	CSECT	Location	Label	CSECT	Location
ERR024AA	IJBCTL	EBB4	FNDDEL	IJBSCN	DDA4
ERR032	IJBSCN	DGB4	FNDDELIM	IJBINL	JCF1
ERR040	IJBESD	BDH1	FNDENT	IJBLNK	AHH2
ERR041	IJBESD	BFA3	FONDEL	IJBSCN	DEE4
ERR043	IJBESD	BDB5	FRDLNCI	IJBFIN	KBF1
ERR043	IJBESD	BEG5	FRDLNF	IJBFIN	KBB3
ERR043	IJBESD	BEJ2	FRDLNOT	IJBFIN	KBJ2
ERR044	IJBLNK	AFD4	FRDNEST	IJBFIN	KAE2
ERR045	IJBESD	BCD2	FRDUMPH	IJBFIN	KAA2
ERR046	IJBESD	BDF3	FRLBENA	IJBFIN	KAC2
ERR050	IJBLNK	AKE1	FRLBRET	IJBLNK	AHB3
ERR051	IJBOTH	CCE4	FRMDOIO	IJBFIN	KAE4
ERR055	IJBOTH	CDH3	FRMINID	IJBFIN	KAG5
ERR058	IJBOTH	CGD4	FRMOVE	IJBFIN	KAB4
ERR070	IJBLNK	ABH2	FRUPAD	IJBFIN	KAA3
ERR081	IJBCTL	ECJ3			
ERR085	IJBMAP	FDF5	GETCENT	IJBRLD	GAG4
ERR091	IJBOTH	CHD2	GETVRB	IJBSCN	DAG5
ERR093	IJBLNK	AKE5	GHERKIN	IJBINL	JBH4
ERR097	IJBLNK	AGH2	GORLD	IJBMAP	FDJ4
ERR44	IJBLNK	ABC3			
ERR44A	IJBLNK	ABB2	HOWMANY	IJBCTL	EEE4
ESCNCD	IJBESD	BFD3			
ESDNXT	IJBESD	BAH2	IJBCAT	IJBCAT	HAB1
ESDRET	IJBESD	BAB3	IJBMAP	IJBMAP	FAC1
ESDSBM	IJBESD	BCH1	IJBOTH	IJBOTH	CAB1
ESD1ST	IJBESD	BAA2	IJBRLD	IJBRLD	GAB1
ESIXTA	IJBMAP	FAG3	IJBSCN	IJBSCN	DAB1
ESLBCD	IJBESD	BDB1	INCCRD	IJBSCN	DBA1
EUPDCN	IJBESD	BFG2	INCCRD1	IJBSCN	DBE1
EUPDOK	IJBESD	BFC2	INCERR	IJBSCN	DGC4
EUPDSW	IJBESD	BFD2	INCFND	IJBSCN	DHE1
EUPDXT	IJBESD	BFC5	INCGET	IJBSCN	DGC1
EUPTRY	IJBESD	BFG1	INCGET1	IJBSCN	DGC2
EXECWR	IJBMAP	FDJ5	INCGET2	IJBSCN	DGF1
EXLOAD	IJBLNK	AHB4	INCLOP	IJBSCN	DGK1
EXTNLP	IJBMAP	FDB2	INCLPR	IJBSCN	DGB1
EXTNPR	IJBMAP	FDG2	INCRED	IJBSCN	DGD5
EXTNSW	IJBMAP	FDD2	INCREE	IJBSCN	DGH1
EXTRCT	IJBSCN	DEB3	INDJDR	IJBLNK	AHH5
EXTSCN	IJBMAP	FDC1	INS000	IJBLNK	AAF1
			INTPT2	IJBINL	JAB1
FBICLB	IJBINL	JBF2	INTS01	IJBINL	JAC2
FBILNK	IJBINL	JAE3	ISDISP	IJBCTL	ECA3
FBINSEC	IJBOTH	DJJ2	ISROOT	IJBCTL	ECE3
FBIRES	IJBINL	JAC5			
FBIRLB	IJBINL	JBF3	LABST	IJBSCN	DBB2
FBRDIOC	IJBOTH	DJB3	LDRGO	IJBMAP	FCB3
FBRDIREL	IJBOTH	DJB2	LDRSCN	IJBMAP	FCG2
FBRDIRF	IJBOTH	DJC1	LDRTSD	IJBMAP	FCC3
FBRDIRLB	IJBOTH	DJB1	LKQUO	IJBSCN	DCD4
FBRDNXT	IJBOTH	DJG1	LOADBASE	IJBLNK	AHF4
FBRESY1	IJBINL	JBA1	LOGMSG	IJBLNK	ADB2
FCHRLD	IJBMAP	FDE5	LSETB	IJBLNK	ABG2
FCLRLOOP	IJBINL	JDD1	LTCDAD	IJBLNK	ABA4
FCLRNMAX	IJBINL	JDD2	LTCDNO	IJBLNK	ABJ2
FDISKIO	IJBLNK	AGF1	LTCDRF	IJBLNK	ABB4
FINCLOP	IJBOTH	DJE3	LTESID	IJBLNK	ABC2
FINDEL	IJBSCN	DEE3			
FINDND	IJBSCN	DBJ3	MAINFLOW	IJBINL	JDB5
FINPRO	IJBCTL	ECG3	MAPCST	IJBMAP	FBH2
FIRSTPH	IJBCTL	EAB2	MAPHAS	IJBMAP	FBB5
FLDUMPH	IJBFIN	KBA2	MAPHNM	IJBMAP	FBH1
FLINIT	IJBFIN	KBE3	MAPLDR	IJBMAP	FCF2
FLNEST	IJBFIN	KBB5	MOREBLK	IJBCTL	EEG2
FLNRETI	IJBINL	JCC1	MOVENTRY	IJBCTL	EAD1

Label	CSECT	Location	Label	CSECT	Location
MOVPER	IJBCTH	CFJ1	RESET	IJBSCN	DEC3
			RETCHABK	IJBFIN	KAG4
NDESLP	IJBLNK	AJH1	RETOUR	IJBINL	JCE5
NEWPHS	IJBCTL	EFB3	RETURN	IJBCAT	HBB1
NEWPHSS	IJBCTL	EFD3	RLADCN	IJBRLD	GCB3
NEXTOPER	IJBINL	JCD3	RLBOPEN	IJBINL	JBC4
NEXTOPT	IJBSCN	DFG3	RLBYWR	IJBOTH	CEG2
NOAUTOCHK	IJBSCN	DFD2	RLCTER	IJBRLD	GBE2
NOFLTPT	IJBCTL	EBH1	RLDCLEAR	IJBCTL	EFH3
NOSCAGAN	IJBCTL	ECE2	RLDCMN	IJBRLD	GBG3
NOSTAT	IJBCAT	HBJ1	RLDCON	IJBRLD	GCA1
NOTACT	IJBINL	JCD5	RLDOP	IJBRLD	GAE3
NOTCTL	IJBLNK	AJB1	RLDOR	IJBRLD	GBB1
NOTF	IJBCTL	EBC2	RLDPRC	IJBOTH	CDB1
NOTRFR	IJBMAP	FAE5	RLDPRC3	IJBRLD	GEB1
NOT1ST	IJBCTL	EAB4	RLDRAG	IJBRLD	GAA2
NTABS	IJBSCN	DDA3	RLDRET	IJBRLD	GAD2
NTESLP	IJBLNK	AJF1	RLDSW1	IJBRLD	GAH2
NTROOT	IJBCTL	EBA1	RLDSW2	IJBRLD	GBD3
NXTCARD	IJBINL	JCA1	RLDSW3	IJBRLD	GCC1
			RLDSW4	IJBRLD	GCE3
OTHINC	IJBCTH	CAA5	RLRET	IJBOTH	CDD1
OTHTFR	IJBOTH	CAF3	RLSTP	IJBOTH	CDB4
OTHTYP	IJBCTH	CAD3	RLSW1	IJBOTH	CDH1
OVRLSW	IJBMAP	FDD1	RLWRIT	IJBOTH	CEA2
			RNXTRN	IJBRLD	GBB3
PBDYCHK	IJBSCN	DFF2	ROTSID	IJBRLD	GCC4
PCILYES	IJBINL	JBB1			
PHADMD	IJBMAP	FBA1	SAVCTL	IJBSCN	DBB5
PHCEXIT	IJBSCN	DFH3	SAVCTL1	IJBSCN	DDJ5
PHCOPTN	IJBSCN	DFB2	SCAGAN	IJBCTL	ECG2
PHCRD	IJBSCN	DCA1	SCAN	IJBMAP	FBJ2
PHCRD1	IJBSCN	DCG1	SCANCD	IJBCAT	HAE1
PHSCAN	IJBMAP	FBC5	SEEBLK	IJBSCN	DAE3
PHSPRC	IJBCTL	EAF1	SELFRLC	IJBMAP	FBE2
PHSTOR	IJBMAP	FAJ2	SETPHS	IJBCTL	ECB5
PHXADD	IJBCTL	EDD1	SETUPSCN	IJBOTH	CAC3
PRERR	IJBLNK	AJC4	SKIPB	IJBSCN	DEB1
PREXTN	IJBMAP	FDB1	SONOMN	IJBMAP	FAF1
PRINT	IJBLNK	ADB1	SPACE1	IJBLNK	ADB3
PRSDPC	IJBESD	BCC1	SRCHCD	IJBLNK	ACB2
PRTLINE	IJBLNK	ADF1	SRLABL	IJBLNK	ACC2
PRTLST	IJBLNK	ADB4	SRPCOD	IJBLNK	ACD2
PS3SW2	IJBRLD	GAF4	STORR5	IJBSCN	DDE2
PS3SW4	IJBRLD	GAH4	STRLD	IJBCTL	EED4
			STRTBLD	IJBCAT	HAF2
QUAPRO	IJBSCN	DDA5	SUBMOD	IJBSCN	DBH1
			SVACHK	IJBSCN	DFE2
RADD4	IJBRLD	GAG2	SYMBORG	IJBCTL	EBB1
RDALSW	IJBLNK	AHD1	SYSLIB	IJBSCN	DGE3
RDEXEC	IJBLNK	AHC3	SYSREST	IJBINL	JAC4
RDNEXT	IJBLNK	AHE1			
RDOK01	IJBRLD	GGG1	TAPE01	IJBINL	JAH3
RDSF01	IJBRLD	GGF3	TERSXY	IJBMAP	FDD3
RDS000	IJBLNK	AAB1	TISDISK	IJBRLD	GGC2
RDS001	IJBRLD	GGB1	TISESD	IJBLNK	AJE2
READC1	IJBLNK	AGB1	TRYROT	IJBMAP	FBE1
READLST	IJBRLD	GDB4	TSTCNT	IJBRLD	GDD1
RECF00	IJBLNK	AAC3	TSTESD	IJBOTH	CFH1
RELOC	IJBMAP	FBF2	TSTLIM	IJBSCN	DDF1
RELROUT	IJBRLD	GDC2	TSTNEG	IJBSCN	DDC2
REPF01	IJBRLD	GGB5	TSTRLD	IJBRLD	GEB3
REPROC	IJBCTH	CCB2	TSTRLDXX	IJBRLD	GDA2
REPTXT	IJBOTH	CCC3	TSTUNR	IJBRLD	GDA1
REP001	IJBRLD	GGB4	TXTALL	IJBOTH	CBG4
RESDCN	IJBRLD	GCC5	TXTGET	IJBOTH	CBB3

Label	CSECT	Location
TXTPRC	IJBOTH	CBB1
TYPEVB	IJBLNK	AGD4
UNRSPC	IJBMAP	FDH3
UPDATE	IJBSCN	DEB5
UPDCOM	IJBCAT	HBC3
UPDDIR	IJBCAT	HAB4
UPDSKAD	IJBLNK	AEG3
UPNDS	IJBOTH	CAJ1
USESYS	IJBINI	JBG1
VALIDFN	IJBINI	JAD1
VERLOP	IJBBLD	GEE1
VERLOP	IJBBLD	GFE1
WHATNOW	IJBCTL	EFB1
WRLST	IJBBLD	GFA1
WRSF01	IJBOTH	CHB4
WRST01	IJBOTH	CHA2
WRS001	IJBOTH	CHB1
WRTRFR	IJBCTL	EDG1
XTPHGT	IJBLNK	AFE1
XTPHNO	IJBLNK	AFB1

APPENDIX B: PHASE TO MODULE CROSS REFERENCE

Phase:	Module:
$LNKEDT	IJBLE1

Message	CSECT	Chart
2100I	IJBOTH	CA
2101I	IJBSCN	DA
2102I	IJBLNK	AC
	IJBSCN	DD
2110I	IJBSCN	DA
2111I	IJBSCN	DA
2112I	IJBSCN	DA
	IJBINL	JC
2113I	IJBOTH	CC
	IJBINL	JC
	IJBSCN	DE
2114I	IJBSCN	DA
2116I	IJBSCN	DA
	IJBLNK	AH
2117I	IJBSCN	DA
2120I	IJBCTL	EA
2121I	IJBSCN	DC
2122I	IJBCTL	EB
2123I	IJBCTL	ED
2124I	IJBCTL	EB, EC
2125I	IJBSCN	DC
2130I	IJBSCN	DG
2131I	IJBSCN	DG
2132I	IJBSCN	DG
2133I	IJBSCN	DB
2135I	IJBINL	JC, JD
2136I	IJBINL	JC, JD
2137I	IJBINL	JC, JD
2140I	IJBESD	BB, BD
2141I	IJBESD	BF

Message	CSECT	Chart
2142I	IJBESD	BB
2143I	IJBESD	BD, BE
2144I	IJBLNK	AB, AF
2145I	IJBESD	BC
2146I	IJBESD	BD
2147I	IJBOTH	CG
2150I	IJBLNK	AK
	IJBOTH	CB
2151I	IJBOTH	CC
2155I	IJBOTH	CD
2156I	IJBOTH	CE
2158I	IJBOTH	CG
2160I	IJBCTL	EF
2161I	IJBSCN	DF
2170I	IJBOTH	CB, CD, CF
2181I	IJBCTL	EA, EC
2182I	IJBCTL	EF
2184I	IJBINL	JB
2185I	IJBMAP	FD
2186I	IJBCAT	HA
2190I	IJBINL	JA
2191I	IJBOTH	CH
	IJBINL	JA
2192I	IJBCAT	HA
2193I	IJBLNK	AK
2194I	IJBLNK	AG
2195I	IJBINL	JB
2197I	IJBLNK	AG
2199I	IJBMAP	FD

APPENDIX D: SYSTEM RESIDENCE ORGANIZATION

Component		Starting Disk Address			Number of Tracks (Alloc.)	R = Required O = Optional
		CC	HH	R		
IPL Record	(Phase $$A$IPL1)	00	00	1	1	R
IPL Record		00	00	2		R
System Volume Label		00	00	3		R
User Volume Label		00	00	4		O
System Directory	Record 1	00	01	1	1	R
	Record 2	00	01	2		R
	Record 3	00	01	3		R
	Record 4	00	01	4		R
IPL Records (Phase $$A$PLBK)		00	01	5		R
Core Image Directory	Cataloged Phases	00	02		*	R
	Linked Phase					
Core Image Library Members		X	Y+1	1	*	R
Relocatable Directory		Z+1	00	1	*	O
Relocatable Library Members		X	Y+1	1	*	O
Source Statement Directory		Z+1	00	1	*	O
Source Statement Library Members		X	Y+1	1	*	O
Procedure Directory		Z+1	00	1	*	O
Procedure Library Members		X	Y+1	1	*	O
Label Information Area		Z+1	00	1	Device dependent	R

* Allocation Dependent on User Requirements
X = Ending CC of the Preceding Directory
Y = Ending HH of the Preceding Directory
Z = Ending CC of the Preceding Library

Figure 12. System Residence Organization on CKD

Component	Starting Disk Address Block Number	Number of Blocks	R=Required O=Optional
IPL Records (Phase $$A$IPL0)	0	1	R
System Volume Label[1]	1	1	R
System Directory	2	1	R
IPL Retrieval Program (Phase $$A$PLBF)	3	7	R
Core Image Directory	10	*	R
Core Image Library Members	X+1	*	R
Relocatable Directory	Y+1	*	O
Relocatable Library Members	X+1	*	O
Source Statement Directory	Y+1	*	O
Source Statement Library Members	X+1	*	O
Procedure Directory	Y+1	*	O
Procedure Library Members	X+1	*	O
Label Information Area	Y+1	200[2]	R

* = Allocation dependent on user requirements
X = Last block of preceding directory
Y = Last block of preceding library

[1] Optional user volume labels if written will be in the same block following the system volume label.

[2] Using the Restore program you may allocate a label information area different than the default size of 200 blocks.

Figure 13. System Residence Organization on FBA

Notes to Figure 12

The disk device can be an IBM 2314, an IBM 2319 or an IBM 3333/3330/3330-11/3340/3350. The organization of SYSRES is as follows:

IPL

This area contains the initial program load (IPL) bootstrap program, which causes the IPL retrieval program to be read from SYSRES and loaded into real storage.

System Volume Label

The volume label (VOL1 label) contains the address of the volume table of contents (VTOC) established when the pack was initialized.

User Volume Label

The user volume label area is provided for any additional standard volume labels (VOL2-VOL8 labels). This area can extend from record 4 through the end of track 0.

System Directory

This area contains the system (master) directory. Record 1 contains the location of the core image directory and the address of the label information area . Records 2, 3, and 4 contain the starting addresses of the relocatable directory, source statement directory, and procedure directory, respectively. Record 5 contains the IPL retrieval program.

Core Image Directory

This directory consists of two or more tracks, depending on the allocation specified by the user. The directory is in two parts: The first is the directory of cataloged phases; the second is the directory of linked phases.
Each directory entry describes one phase in the core image library and contains such information as the phase name, loading address, number of blocks, type of phase, entry point, starting disk address in the core image library, and the number of text bytes in the last block. The entries are sorted in alphanumeric sequence.
The first entry in the directory is called the library descriptor entry. This contains such informations as the number of directory tracks, library cylinders, active phases, directory blocks available, and library blocks available.
Thereafter, the entries have a length varying from 14 bytes to 34 bytes (depending on the specifications in the PHASE statement). Entries are grouped in blocks of 256 bytes, plus an 8-byte key for the highest phase name in the block.

Core Image Library

The core image library consists of one or more complete cylinders, depending on the allocation specified by the user.

Relocatable Directory

This directory consists of one or more tracks, depending on the allocation specified by the user. It contains two types of information:

1. System directory information for the relocatable directory and library. This information occupies the first five entries of the first record in the relocatable directory.

2. An entry that describes each module (the output of a complete language translator run) in the relocatable library and contains: the module name, total number of text-record blocks required to contain this module, starting disk address of the first text-record of this module, and change level identification.

Relocatable Library

The relocatable library consists of one or more complete cylinders, depending on the allocation specified by the user. The number of modules and the size of each module to be contained in this library dictate the number of tracks that must be allocated.

Source Statement Directory

This directory consists of one or more tracks, depending on the allocation specified by the user. It contains two types of information:

1. System directory information for the source statement directory and library. This information occupies the first five entries of the first record in the source statement directory.

2. An entry that describes each book (a sequence of source language statements in a compressed card image format, accessed by a single name) in the source statement library and contains: a sublibrary prefix, the book name, starting disk address of the first block of this book, total number of blocks required to contain this book in the source statement library, and change level information.

Source Statement Library

The source statement library consists of one or more complete cylinders depending on the allocation specified by the user. The number of blocks and the size of each book to be contained in this library dictates the number of tracks that must be allocated.

Procedure Directory

This directory consists of one or more tracks depending on the allocation specified by the user. It contains two types of information:

1. System directory information for the procedure directory and procedure library. This information occupies the first five entries of the first record in the procedure library.

2. An entry that describes each procedure (a set of control statements in card image format) catalogued in the procedure library and contains: the name of the procedure, the starting disk address of the procedure, the number of blocks occupied in the procedure library and a version and modification level.

Procedure Library

The procedure library consists of one or more complete cylinders, depending on the allocation specified by the user. Each procedure consists of one or more consecutive 80-byte blocks, containing control statements (one card image per block).

Label Information Area

The label information area contains standard, partition standard, and user label information for all partitions. This area is allocated 2 cylinders on the 3333/3330/3330-11, 2 cylinders on the 2314/2319, 3 cylinders on the 3340, or 1 cylinder on the 3350. Job Control stores label information found in job control statements here. The label information area follows the last library and ends the SYSRES file.

Volume Table of Contents

Following the label information area , the use of the remaining areas on the disk pack is left to the user's discretion. However, the volume table of contents (VTOC) must be contained on the same physical disk pack as the SYSRES file. (A VTOC is required on every disk pack.) The VTOC is most frequently the last cylinder before the alternate track area. The location and length of the VTOC are determined when the pack is initialized.

The VTOC is a file describing the organization of the disk pack. It contains the VTOC identifier (format 4 label) that contains the starting and ending addresses of the VTOC, a format 5 label that is not used by DOS/VS, and format 1, 2, and 3 labels that identify and describe all files on the pack. More specific information on label formats is contained in the DOS/VS DASD Labels, GC33-5375.

Alternate SYSRES Layout

The relocatable library, the source statement library, and the procedure library are shown as optional areas of the SYSRES file because these libraries are not essential for system operation. If desired, the relocatable and source statement libraries can be defined as private libraries; a private library for the procedure library is not supported. A private core image library can also be defined, but the system core image library must always be included on the SYSRES file.

PRE-PROCESSING

1. For each ESD item produced by a language translator, an input control dictionary entry is built at a fixed location in storage. In some cases this input control dictionary entry will be moved to the control dictionary during processing.

2. The input ESD type field is validated.

   - If it is a weak external, the ESD type field in the input control dictionary entry is set to ER and the NOAUTOL and WXTRN bits in CSWITCH are turned on.

   - If it is invalid, an error condition exists, the whole ESD card is ignored, and the next ESD card is processed.

3. Further pre-processing depends on the input ESD type.

   - For LD ESD input
     Each input LD ESD item has a pointer (ESID) to the linkage table control section. This pointer is used to determine whether an input ESD item has already been processed. The check is made by locating the corresponding linkage table entry and investigating the control dictionary number stored in this entry.
     If the number is:
     zero — the ESD item pointed to by the LD has not yet been processed. The LD is then marked unassigned in CSWITCH of the input control dictionary entry.
     negative — the entry is ignored and the next ESD item is processed.
     positive — an entry already exists for the ESD item pointed to by LD. If the existing entry is of the type SD or CM, the LD is marked assigned, and the control dictionary number of the SD or CM is stored in the input control dictionary entry.
     If the entry is not of the type SD or CM an error condition exists.

   - For ER ESD input
     If NOAUTO has been requested for the phase being processed, the NOAUTOL bit in CSWITCH is set on. Otherwise, CSWITCH remains off.

   - For SD or PC ESD input
     Requirements:
     (1) The assembled origin must be aligned on a double word boundary.
     (2) PC must be unnamed.

     The relocation factor is calculated by subtracting the assembled origin from the storage address (NXPHRG).

     For a normal INCLUDE the pre-processing is finished at this stage.

     For a submodular INCLUDE the name list of included CSECTs is scanned for a name identical to the name of the input control dictionary entry. If the names match, pre-processing is finished. If not, the ESD type field in the input control dictionary entry is changed to ER and a switch is set to ensure that the control dictionary number in the linkage table is given a negative value.

     Note: A negative control dictionary number in the linkage table entry is a signal for the ESD processor to ignore LDs belonging to this section definition, and for the text processor to ignore the corresponding text cards.

PROCESSING

1. The control dictionary is scanned for an entry with the same name as the input ESD item.

   This scan starts at the end of the control dictionary and proceeds towards the beginning until either a match occurs or the beginning of the control dictionary is reached. If a match occurs, the control dictionary entry is called a duplicate.

   The scan continues if the duplicate is a phase entry.

2. If no duplicate is found, the input control dictionary entry is added to the end of the control dictionary.

3. If the input ESD is an SD, PC, CM, or ER, an entry is made in the linkage table.

4. If a duplicate was found, the action taken by the ESD processor depends on the relationship between input and duplicate. Use Figures 14 - 18 to determine the different actions taken by the ESD processor.
A summary of all possible ESD processing actions is shown in Fig. 14. The actions are named A1 - Err-46.
To find out the appropriate action(s) taken during input processing of CM, ER, SD, or LD use Fig. 15, 16, 17, 18 respectively. The upper part of these figures shows the various conditions which exist (Y), not exist (N), or can be ignored (-), while the lower part indicates the actions taken (X).

Name	Description
A1	Ignore input control dictionary entry.
A2	Add input control dictionary entry to the end of the control dictionary.
A3	Replace duplicate with the input control dictionary entry.
A4*	Add the linkage table entry pointing to the last entry added to the control dictionary.
A5*	Add the linkage table entry pointing to the duplicate.
A6	Change duplicate LD to LR.
A7	Continue scan of the control dictionary.
A8	Save length of longest CM in the control dictionary.
A9	Give control dictionary number in linkage table a negative value.
A10	Change input LD to LR.
A11	Set 'Possible Duplicate Entry' switch.
Err-40	Print error message '2140I' and go to RDNEXT.
Err-43	Print error message '2143I' and go to RDNEXT.
Err-46	Print error message '2146I' and go to RDNEXT.

* If a submodular INCLUDE was used and the name list of included SDs does not contain an SD, the control dictionary number in the linkage table is given a negative value.

Figure 14. ESD Processing Actions

Duplicate = SD	Y	N	N	N	N
= PC	N	Y	N	N	N
= CM	N	N	Y	N	N
= LD/LR	N	N	N	N	Y
= ER	N	N	N	Y	N
A3	-	-	-	X	-
A4	-	-	-	-	-
A5	X	-	X	X	-
A7	-	X	-	-	-
A8	-	-	X	-	-
Err-46	-	-	-	-	X

Figure 15. Process Input CM

Duplicate = SD, LD, or LR	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	Y
= LD	-	-	N	N	N	N	Y	Y	Y	Y	N	N	N	N	-
= CM	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N
= ER	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	N
Duplicate unassigned *	N	N	N	N	N	N	N	N	N	N	-	-	-	-	Y
Name = 'IJ..' or 'IBM..'	Y	Y	Y	Y	Y	N	Y	Y	Y	N	-	-	-	-	-
Name = 'IBM..'	N	Y	-	-	N	-	-	-	N	-	-	-	-	-	-
NOAUTO for input	N	N	Y	N	N	-	Y	N	N	-	-	-	Y	N	-
Duplicate in current phase	N	N	-	Y	N	-	-	Y	N	-	-	N	Y	Y	-
Duplicate in ROOT phase	N	-	-	-	Y	-	-	-	Y	-	-	-	-	-	-
A2	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-
A3	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-
A4	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-
A5	-	-	X	X	X	X	X	X	X	X	X	X	X	X	-
A6	-	-	-	-	-	-	X	X	X	X	-	-	-	-	-
A7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X

* SD is to be considered assigned

Note: Weak externals are processed like ERs for which NCAUTO is requested.

Figure 16. Process Input ER

Duplicate = SD	N	N	N	N	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N
= CM	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
= LD or LR	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
= ER	N	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Duplicate unassigned	-	-	-	-	-	-	-	-	N	N	N	Y	Y	Y	Y	Y	Y	Y
Input and duplicate ESIDs agree	-	-	-	-	-	-	-	-	-	-	-	N	N	N	Y	Y	Y	Y
ASSORGs agree	-	-	-	-	-	-	-	-	-	-	-	-	-	-	N	N	N	Y
Duplicate in current phase	-	-	Y	N	Y	N	N	N	-	Y	N	-	Y	N	-	Y	N	-
Name = 'IBM..'	-	N	Y	Y	-	Y	N	N	N	Y	Y	N	Y	Y	N	Y	Y	-
Duplicate in ROOT phase	-	-	-	-	-	-	N	Y	-	-	-	-	-	-	-	-	-	-
A2	-	-	-	X	-	X	X	-	-	-	X	-	-	X	-	-	X	-
A3	X	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X
A4	-	-	-	X	-	X	X	-	-	-	X	-	-	X	-	-	X	-
A5	X	X	X	-	X	-	-	X	-	-	-	-	-	-	-	-	-	X
A9	-	-	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-	-
Err-43	-	-	-	-	-	-	-	-	X	X	-	X	X	-	X	X	-	-

Figure 17. Process Input SD

Duplicate = SD, LD, or LR	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
= CM	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
= LD or LR	N	N	N	N	-	-	-	-	-	Y	Y	Y	Y	Y	Y	Y	N	N	N	N
= ER	N	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Duplicate unassigned **	-	-	-	-	N	N	N	N	N	N	N	N	N	N	N	Y	-	-	-	-
Input unassigned	-	-	-	-	-	-	-	Y	Y	N	N	N	N	N	N	-	N	N	N	N
Input points to duplicate SD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	N	N	N
Input and duplicate point to same C/E entry	-	-	-	-	-	-	-	-	-	Y	N	N	N	N	N	-	-	-	-	-
Names of C/D entries agree	-	-	-	-	-	-	-	-	-	-	N	N	N	Y	Y	-	-	-	-	-
Name of Input and Dupl = 'IBM..'	-	N	Y	Y	N	Y	Y	-	-	-	N	Y	Y	-	-	-	-	N	Y	Y
Duplicate in current phase	-	-	Y	N	-	N	Y	N	Y	-	-	N	Y	N	Y	-	-	-	N	Y
Input and duplicate ASSORGs agree	-	-	-	-	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y

A1	-	-	-	-	-	-	-	-	X	X	-	-	-	-	X	-	X	-	-	-
A2	-	-	-	X	-	X	-	X	-	-	-	X	-	X	-	-	-	-	X	-
A3	-	X	X	-	-	-	-	-	-	-	-	-	-	-	-	X*	-	-	-	-
A10	-	X	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A11	-	-	-	-	-	-	-	-	X	-	-	-	-	-	X	X	-	-	-	-
Err-43	-	-	-	-	X	-	X	-	-	-	X	-	X	-	-	-	-	X	-	X
Err-46	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

* Action A3 is performed retaining the ESD type of the Duplicate
** SD is to be considered assigned

Figure 18. Process Input LD

POST-PROCESSING

- For ER, LD/LR, or CM ESD input the next ESD item is selected for processing.

- For SD or PC ESD input

  1. The control dictionary is scanned for unassigned LDs or LRs pointing to the input item.

  2. The control dictionary entries found during the previous scan are updated. This is done by storing in the control dictionary entry the control dictionary number found in the linkage table entry (that corresponds to the input item).

  3. The storage address (NXPHRG) is updated by adding the length of the control section to it.

     Note: If the length of the control section is provided in the END card, CSECT IJBOTH performs action 3.

For a detailed description of the Linkage Editor Map refer to:
DOS/VS Serviceability Aids and Debugging Procedures, GC33-5380.

Input List

```
JOB  NO NAME   06/28/77                            LINKAGE EDITOR DIAGNOSTIC OF INPUT

ACTION TAKEN  MAP REL
LIST   INCLUDE IJBSL1
LIST   PHASE DSERV,*,NOAUTO
LIST   INCLUDE  FBAEXCP
LIST   INCLUDE  IJJCPDV1
LIST   INCLUDE  ,(IJBDS050)
LIST   PHASE DSERVC,*,NOAUTO
LIST   INCLUDE  ,(IJBDSC)
LIST   PHASE DSERVF,*,NOAUTO
LIST   INCLUDE  ,(IJBDSF)
LIST   PHASE DSERV1,*,NOAUTO
LIST   INCLUDE  ,(IJBDS141)
LIST   PHASE DSERV2,DSERV1,NOAUTO
LIST   INCLUDE  ,(IJBDS250)
LIST   PHASE DSERV2F,DSERV1,NOAUTO
LIST   INCLUDE  ,(IJBDS25F)
LIST   PHASE DSERV3,DSERV1,NOAUTO
LIST   INCLUDE  ,(IJBDS350)
LIST   PHASE DSERV4,DSERV1,NOAUTO
LIST   INCLUDE  ,(IJBDS450)
LIST   PHASE DSERV5,DSERV1,NOAUTO
LIST   INCLUDE  ,(IJBDS550)
LIST   PHASE DSERV3F,DSERV1,NOAUTO
LIST   INCLUDE  ,(IJBDS35F)
LIST   PHASE DSERV6,DSERV1,NOAUTO
LIST   INCLUDE  ,(IJBDS650)
LIST   ENTRY
  |
  |
  |
  V
```

Figure 19. Linkage Editor Map (Part 1 of 3)

Format A

```
   PHASE  XFR-AD  LOCORE  HICORE  DSK-AD    ESD TYPE  LABEL     LOADED  REL-FR

DSERV     01DEEC  01C078  01E877  011 05 09 CSECT     FBA       01C078  01C078
                                            *  ENTRY  FBAEXCP   01C088
                                            *  ENTRY  FBAOPEN   01C994
                                            *  ENTRY  FBACLOSE  01CB78
                                            *  ENTRY  LASTADDR  01C988
                                            *  ENTRY  REGSV06   01CD84
                                            *  ENTRY  CCBADDR   01CDE4
                                            *  ENTRY  CCWADDR   01CDE8
                                            *  ENTRY  REGSV7F   01CE04

                                            CSECT     IJJCPDV1  01DBE0  01DBE0
                                               ENTRY  IJJCPDV2  01DBE0

                                            CSECT     IJBDS050  01DEA8  003808
                                            *  ENTRY  R3564     01DEB2
                                               ENTRY  IJJCPD3   01E360

DSERVC    01E878  01E878  01EBD9  011 06 09 CSECT     IJBDSC    01E878  003808

DSERVF    01EBE0  01EBE0  01EE73  011 06 0A CSECT     IJBDSF    01EBE0  003808

DSERV1    01EE78  01EE78  01F9FF  011 06 0B CSECT     IJBDS141  01EE78  003808
                                            *  ENTRY  STATTAB   01F9AE

DSERV2    01EE78  01EE78  01F7BF  011 07 03 CSECT     IJBDS250  01EE78  002C80

DSERV2F   01EE78  01EE78  01FDDF  011 07 06 CSECT     IJBDS25F  01EE78  002338

DSERV3    01EE78  01EE78  01F2FF  011 07 0A CSECT     IJBDS350  01EE78  0013D0

DSERV4    01EE78  01EE78  01F25F  011 08 01 CSECT     IJBDS450  01EE78  000F48

DSERV5    01EE78  01EE78  01F16F  011 08 02 CSECT     IJBDS550  01EE78  000B60

DSERV3F   01EE78  01EE78  01FC5F  011 08 03 CSECT     IJBDS35F  01EE78  000868

DSERV6    01EE78  01EE78  01F21F  011 08 07 CSECT     IJBDS650  01EE78  000580
```

Format A appears as the Linkage Editor Map if Core Image Library on CKD.
DSK-AD contains CCC HH RR in hexadecimal format.

Figure 19. Linkage Editor Map (Part 2 of 3)

Format B

```
    PHASE   XFR-AD  LOCORE  HICORE  DSK-AD    ESD TYPE  LABEL     LOADED  REL-FR

  DSERV     01EB84  01E878  01F49F  00006209  CSECT     IJJCPDV1  01E878  01E878
                                                 ENTRY  IJJCPDV2  01E878

                                              CSECT     IJBDS050  01EB40  002800
                                              *  ENTRY  R356A     01EB4A
                                                 ENTRY  IJJCPD3   01EF80

  DSERVC    01F4A0  01F4A0  01F7D9  00006217  CSECT     IJBDSC    01F4A0  002800

  DSERVF    01F7E0  01F7E0  01FA7B  00006219  CSECT     IJBDSF    01F7E0  002800

  DSERV1    01FA80  01FA80  02C58F  00006221  CSECT     IJBDS141  01FA80  002800
                                              *  ENTRY  STATTAB   02053E

  DSERV2    01FA80  01FA80  0203C7  00006227  CSECT     IJBDS250  01FA80  001CF0

  DSERV2F   01FA80  01FA80  020997  00006233  CSECT     IJBDS25F  01FA80  0013A8

  DSERV3    01FA80  01FA80  01FF07  00006241  CSECT     IJBDS350  01FA80  000490

  DSERV4    01FA80  01FA80  01FE7F  00006245  CSECT     IJBDS450  01FA80  000008

  DSERV5    01FA80  01FA80  01FD77  00006247  CSECT     IJBDS550  01FA80 -0003F8

  DSERV3F   01FA80  01FA80  02085F  00006249  CSECT     IJBDS35F  01FA80 -0006F0

  DSERV6    01FA80  01FA80  01FE47  00006257  CSECT     IJBDS650  01FA80 -0014D0
```

Format B appears as the Linkage Editor Map of Core Image Library on FBA.
DSK-AD contains block number in decimal format.

Figure 19. Linkage Editor Map (Part 3 of 3)

INDEX

SY33-8556-3

DOS/VSE Linkage Editor Logic Printed in U.S.A. SY33-8556-3

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

DOS/VSE Linkage Editor Logic

SY33-8556-3

READER'S COMMENT FORM

This sheet is for comments and suggestions about this manual. We would appreciate *your* views, favorable or unfavorable, in order to aid us in improving *this* publication. This form will be sent directly to the author's department. Please include your name and address if you wish a reply. Contact your IBM branch office for answers to technical questions about the system or when requesting additional publications. Thank you.

Name

Address

What is your occupation?

How did you use this manual?

As a reference source

As a classroom text

As a self-study text

Your comments* and suggestions:

*** We would especially appreciate your comments on any of the following topics:**

Clarity of the text
Organization of the text

Accuracy
Cross-references

Index
Tables

Illustrations
Examples

Appearance
Printing

Paper
Binding

SY33-8556-3

YOUR COMMENTS, PLEASE . . .

This manual is part of a library that serves as a reference source for systems analysts, programmers and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

CUT ALONG THIS LINE

Fold

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

IBM Corporation
1133 Westchester Avenue
White Plains, N.Y. 10604

Attention: Department 813 BP

Fold

Fold

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601