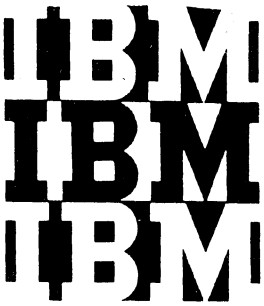
File Number S360-36 (DOS)
Order No. GY24-5086-4



Program Logic

DOS IPL and Job Control

Program Number 360N-CL-453

This reference publication describes the internal logic of the IBM Operating System, Initial Program Load (IPL) and Job Control Programs. It is intended for use by persons involved in program maintenance and by system programmers who are altering the program design. Program logic information is not needed for normal operation of the IPL and Job Control Programs. It is designed to be used as a supplement to the program listing.

Effective use of this manual requires an understanding of IBM System/360 or System/370 operation and of the IBM Disk Operating System control and service programs, macro instructions, and operating procedures. Reference publications for this information are listed in the Preface of this manual.

For titles and abstracts of other associated publications, see the IBM System/360 and System/370 Bibliography, GA22-6822.

DOS Release 25

Fifth Edition (June 1971)

This publication was formerly titled IBM System/360 Disk Operating System IPL and Job Control Programs. Although titles of some DOS publications (including this one) have been simplified, the change does not affect the contents of the publications.

This edition applies to Release 25 of the IBM Disk Operating System and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the specifications herein; before using this publication in connection with the operation of IBM systems, consult the latest System/360 and System/370 SRL Newsletter, GN20-0360, for the editions that are applicable and current.

This edition is a major revision of, and obsoletes, Y24-5086-3.

Summary of Amendments

This edition contains maintenance changes, and it reflects Job Accounting Interface, OLTEP, Data Set Security, PCIL (Private Core Image Library), MCAR/CCH support, and RDE (Reliability Data Extractor). The IBM 3211 Printer and the IBM 1255/1259 Magnetic Character Readers are also supported in this edition.

The flowchart symbols used in this manual conform with American National Standards Institute, Inc., flowcharting standards. See Appendix C for an explanation of the new symbols.

Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

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 Dept., P.O. Box 24, Uithoorn, Netherlands. Comments become the property of IBM.

This Program Logic Manual (PLM) is a detailed guide to the IBM Disk Operating System IPL and Job Control programs. It supplements the program listings by providing descriptive text and flowcharts.

Note: Although titles of some DOS publications have been simplified, the change does not affect the contents of the publications.

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

- Introduction to DOS Logic, GY24-5017.
- DOS Linkage Editor, GY24-5080.
- DOS Supervisor and Related Transients, GY24-5151.
- DOS Logical Transients, GY24-5152.
- DOS System Service Programs, GY24-5153.
- DOS Librarian, GY24-5079.

Prerequisite publications that will aid in the use of this manual are:

- IBM System/360 Principles of Operation, GA22-6821.
- DOS System Control and Service, GC24-5036.
- IBM System/360 Disk and Tape Operating Systems, Assembler Specifications, GC24-3414.

Publications related in subject matter to the seven system control PLMs are:

- DOS Supervisor and I/O Macros, GC24-5037.
- DOS System Generation, GC24-5033.
- DOS Operating Guide, GC24-5022.
- DOS Messages, GC24-5074.
- DOS Data Management Concepts, GC24-3427.

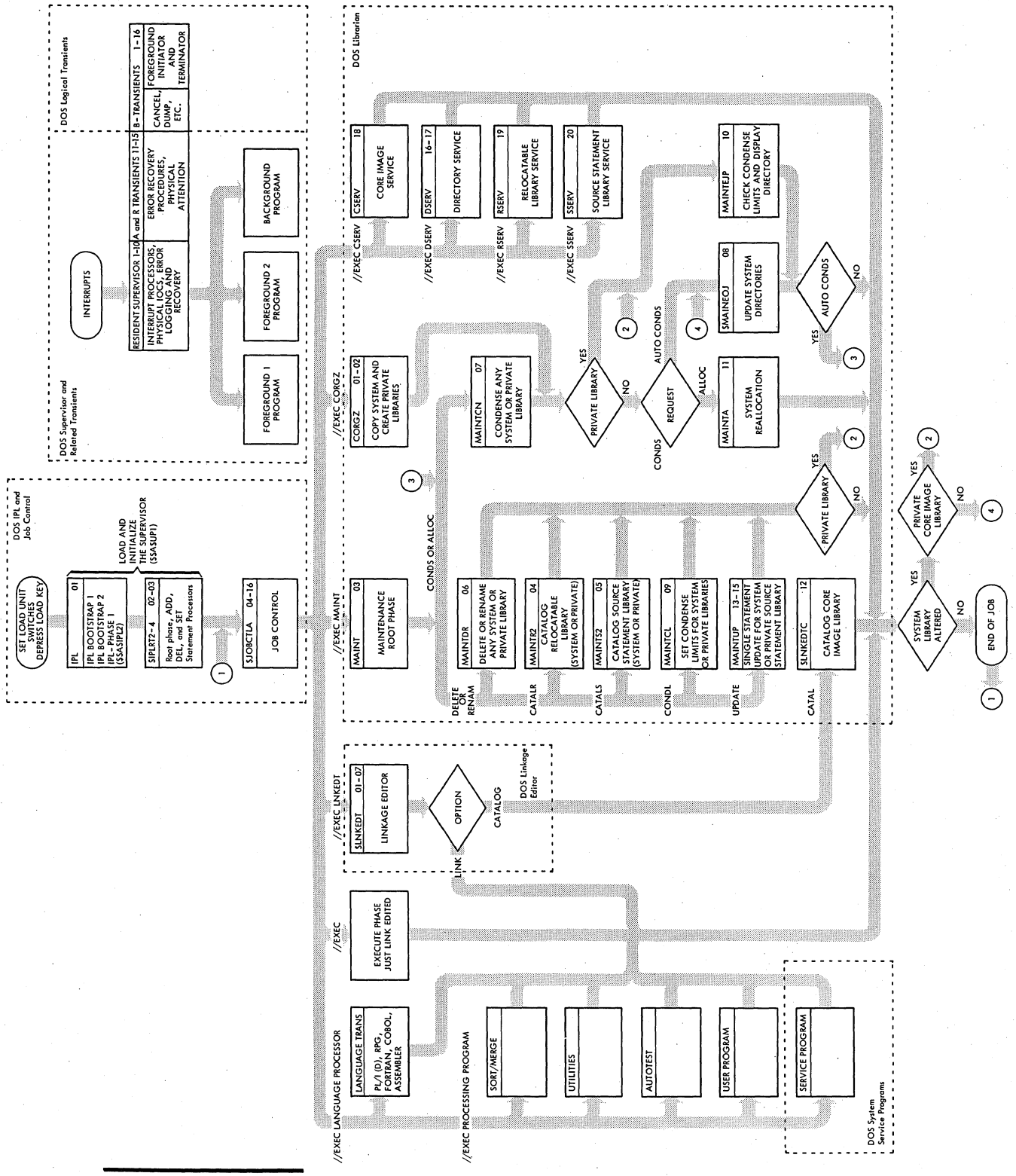
Titles and abstracts of other related publications are listed in the IBM System/360 and System/370 Bibliography, GA22-6822.

This manual consists of four major sections. The first section is an introduction to the IPL and Job Control programs. The next two sections are a discussion of contents of the IPL and Job Control phases. The last section of the manual, the appendixes, contains label lists, error messages, charts, and tables for use as references in analyzing program details.

The detailed flowcharts are identified by letters AA through ZZ. Numerals such as 00 for the program level flowcharts identify the more general flowcharts.

CHARTS

Chart 00. Disk Operating System Program Flow



INITIAL PROGRAM LOAD (IPL)

The IPL program must be executed each time it is necessary to load a new supervisor control program or to change the channel and unit assignment for SYSRES. The IPL program:

1. Operates in the supervisor mode.
2. Loads supervisor into core from SYSRES.
3. Performs any ADDs and DELETEs of devices to the supervisor PUB table.
4. Sets the date and time of day (if supported) into supervisor communications region.
5. Places the system in the problem mode.
6. Exits to EOJ when it is finished.

Additional information is supplied in the section, IPL Program.

JOB CONTROL PROGRAM ($JOBCTLA-$JOBCTLN)

The job control program provides job-to-job transition for:

- Background programs
- Foreground programs if BJF (Batch-Job-Foreground) option is specified. This program also prepares job steps for execution. (One or more programs can be executed within a single job. Each such execution is called a job step.)

On the basis of information provided in job control statements, Job Control performs the following functions:

- Prepares the system for execution of programs in a batched job environment.
- Assigns device address to symbolic units.
- Sets up fields in the communications region(s).
- Edits and stores volume and file label information.
- Prepares for restarting checkpointed programs.
- Clears the problem program area to binary zero between job steps.

Job Control is executed in the problem program area and is overlaid by the job step it is preparing to execute. For additional information, refer to the section, Job Control Program.

INITIAL PROGRAM LOAD PROGRAM (IPL), CHART 01

IPL is a 2-phase program consisting of:

- $$A$IPL1 (a 64-byte bootstrap routine), and
- $$A$IPL2 (less than 4096 bytes).

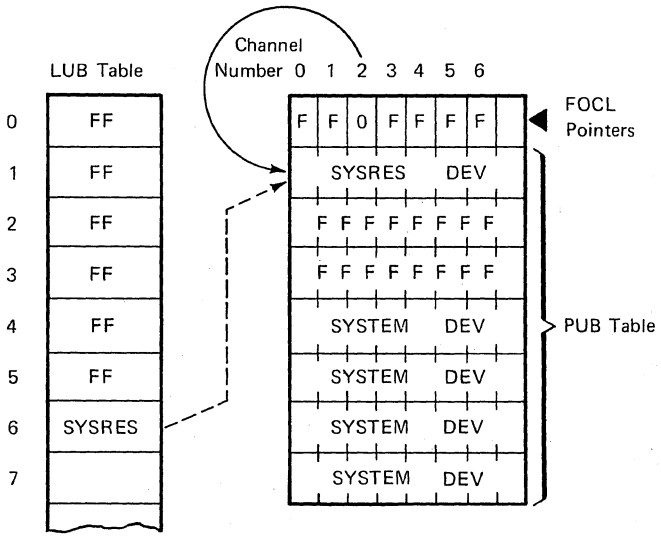
The $$A$IPL1 bootstrap program is located on SYSRES at 00 00 1 (CC HH R). The operator sets the channel and unit of SYSRES in the load unit switches and presses the load key. Microprogramming reads the first record (24 bytes) from SYSRES into main storage starting at location 00. This 24-byte record consists of a PSW starting at location 0 and two chained CCWs starting at location 8.

Microprogramming executes the first CCW at location 8, which reads in the next 40 bytes (3 more chained CCWs and a seek address) from SYSRES (cylinder 0, track 0, record 2). The second CCW is a seek for the $$A$IPL2 program on SYSRES (cylinder 0, track 01, record 5). The next three CCWs are a search, transfer in channel, and read for cylinder 0, track 01, record 5 to load the $$A$IPL2 program. Control is transferred to the $$A$IPL2 program by loading the PSW at location 0. This PSW was loaded as part of $$A$IPL1.

$$A$IPL2 examines the CPU type. If it is a System/360 CPU, a switch is set indicating that a hard wait will occur when a machine check is encountered. $$A$IPL2 clears storage from its own end to the end of main storage. A program check is forced and the program check new PSW returns control to the $$A$IPL2 program. The address at which the program check occurred is saved as the end-of-storage address. There is no provision in the $$A$IPL2 program to clear main storage below location 12,288.

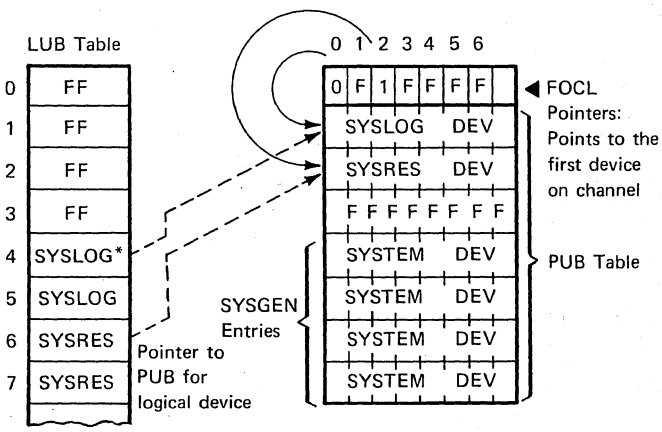
The transient directory is searched for the core image library disk address of the supervisor. The supervisor is read into main storage starting at location 00. The I/O tables that are located within the supervisor are moved to the end of the supervisor (see Figure 3, step 3). A 2-device system is then built in low storage for the IPL operation.

Figures 1 and 2 show examples of I/O tables built by $$A$IPL2. Figure 1 shows the I/O tables for a 1-device system, and Figure 2 shows the I/O tables for a 2-device system. Figure 3 is a map of main storage.



Note: It is assumed that SYSRES is on channel 2.

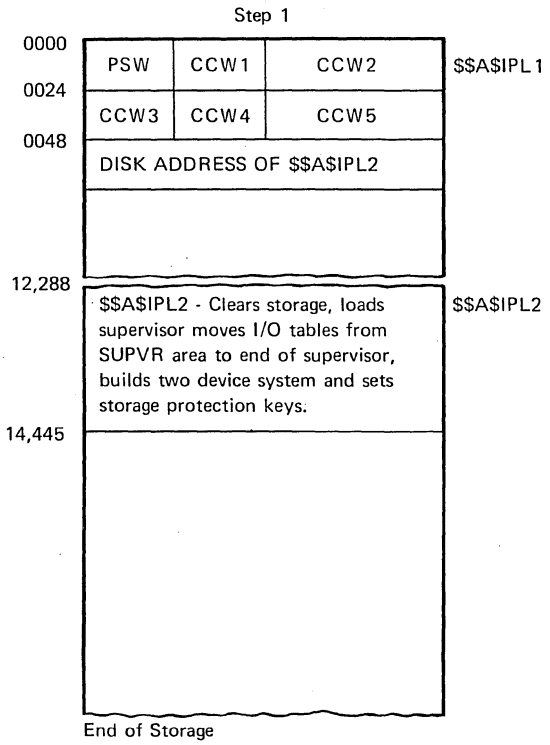
Figure 1. I/O Table for One-Device System



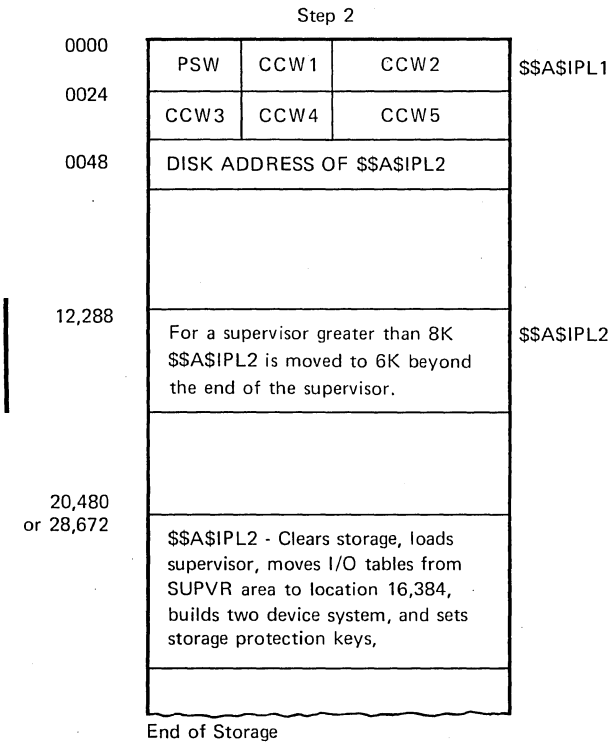
Note: It is assumed that SYSRES is on channel 2 and that the communication device SYSLOG is on channel 0.

* SYSUSE LUB (displacement 18 in the LUB table) is assigned for SYSRDR or SYSLOG.

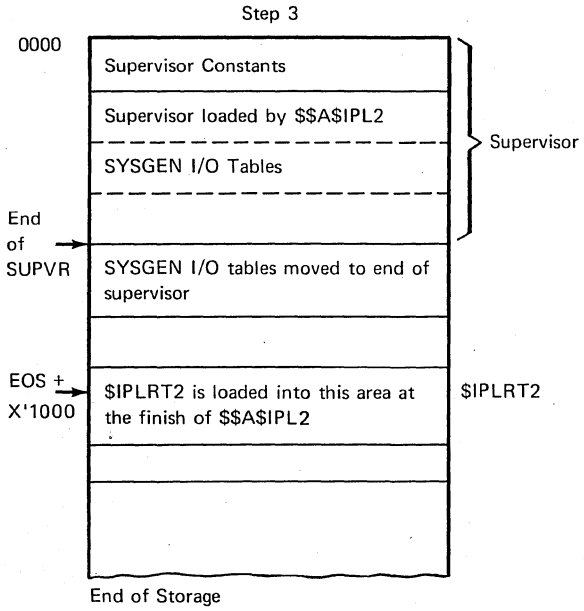
Figure 2. I/O Table for Two-Device System



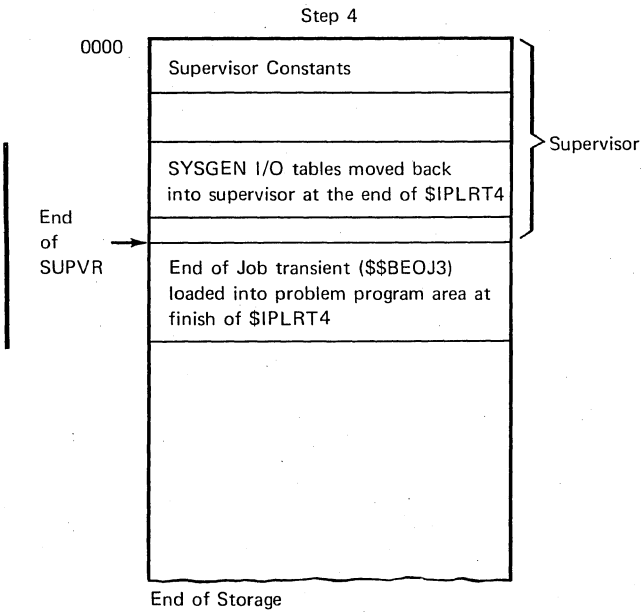
Step 1 - represents the main storage map after $$A$IPL2 is loaded.
NOTE: Storage addresses are in decimal notation.



Step 2 - represents the main storage map after $$A$IPL2 is loaded.



Step 3 - represents the main storage map after $$A$IPL2 loads the supervisor.



Step 4 - represents the main storage map after $IPLRT4 is executed.

Figure 3. IPL Main Storage Map

After the system I/O tables have been moved, a PUB is built in low storage for SYSRES. A LUB is assigned for this PUB and the FOCL is set to point to the PUB for the SYSRES device. The system is put into the wait state and the operator has the option of selecting the communication device desired for IPL. If the desired communication device is:

1. A card reader, and it is already assigned as SYSRDR, the operator presses the external interrupt key causing an external interrupt.

2. A card reader, and it is not assigned as SYSRDR, the operator presses the start key on the reader causing an I/O (device end) interrupt.

3. A console printer-keyboard, the operator presses the request key causing an I/O attention interrupt.

After the operator has taken the appropriate action for choosing a communication device, a PUB and LUB are added and the FOCL is updated to show the new device. This completes building of the 2-device system for IPL.

A check is made to determine if the storage protection feature is supported. If so, the storage protection keys are set. The supervisor area, in blocks of 2K, receives a storage protection key of 0. The upper part of the supervisor that is not an even multiple of 2K and the remainder of main storage are not protected. They receive a storage protection key of 1. The $$A$IPL2 issues a SVC of 4 to load the $IPLRT2 program, overlaying the $$A$IPL2 program.

$IPLRT2, CHART 02

The $IPLRT2 program is loaded and executed every time the operator chooses to IPL the system. It is loaded (by the $$A$IPL2 program) starting at location EOS+4096 (see Figure 3, step 3). Before loading $IPLRT2, the $$A$IPL2 program has moved the system I/O tables to high core. A 2-device system, SYSRES and SYSUSE (SYSUSE=SYSRDR or SYSLOG), has been built by the $$A$IPL2 program for IPL operations.

The $IPLRT2 program performs the following functions:

- Scans the LUB table entries in each partition and establishes partition ownership for each entry.
- Checks for RMS support. When the IPL program is executed on a System/360 CPU, System/370 functions are negated, RETAIN/370 support is turned off (if present), and coding to simulate System/360 support is moved into the supervisor. In the case of a System/370 CPU, the address of the extended logout area is found and $$BCCHHR is fetched to build the load list.
- Monitors core usage and indicates whether there is enough core storage available to load both $IPLRT3 and $IPLRT4 into core at the same time. When there is not enough core storage available, a switch is set, indicating that $IPLRT3 and $IPLRT4 will be separately loaded into the same area as needed.
- Contains the error subroutines to issue the error messages when necessary.
- Loads the appropriate phase(s) into core storage after the $IPLRT2 program. If the first control card is a SET card, $IPLRT4 is loaded. If the first control card read is a DEL or an ADD card and there is sufficient core storage available to allow both $IPLRT3 and $IPLRT4 to be loaded, both are loaded. If there is not sufficient core storage available to have $IPLRT4 loaded after $IPLRT3, each is loaded into the same area as it is needed.

The ADD, DEL, and SET statements are entered from the IPL communication device (SYSRDR or SYSLOG). The formats for these statements are described in Figure 4.

After a card is read, the operation code is evaluated by a translate and test instruction to determine the type of statement.

$IPLRT3, CHART 03

The $IPLRT3 program is loaded and executed only when ADD or DEL control cards are submitted to the IPL program. It is loaded by the $IPLRT2 program.

The $IPLRT3 program performs the following functions:

- Adds a device to the system.
- Deletes a device from the system.

Add Routine ($IPLRT3)

The add routine checks to ensure the device is not already assigned. It then determines where to add the PUB in the PUB table and moves all the PUB entries beyond this point down one PUB length to make room for the new PUB. The new PUB is then inserted in the area just vacated. The LUB table and FOCL pointers are updated to reflect the new entry and the routine returns to read another control statement.

Delete Routine ($IPLRT3)

The delete routine first checks to see if the device to be deleted is in the PUB table and then determines the location in the PUB table of the PUB to be deleted. All PUBs beyond this point are moved up one PUB length overlaying the PUB to be deleted. The LUB table and FOCL pointers are updated so they no longer point to a nonexisting PUB entry. The routine returns to read another control statement.

$IPLRT4, CHART 03

The $IPLRT4 program is loaded and executed every time the operator decides to IPL the system. It is loaded by the $IPLRT2 program.

The $IPLRT4 program performs the following functions:

- Sets the system date.
- Sets the system time of day, if the timer feature is supported.
- Checks the channels for file-protect support for each device when DASDFP option is specified.
- Scans the PUB table. If a 3211 is found, $IPLRT4 loads $$BUFLDR.

Set Routine ($IPLRT4)

The set time of day routine determines the operand format of the set statement.

- The DATERT subroutine converts the month, day, and year to decimal. This information is then stored in the system date field of the communication region (displacement 79).
- The TIMERT subroutine is used if the timer feature is supported. It converts the hours, minutes, and seconds to decimal and determines the time of day in total seconds. The total seconds, multiplied by 300, is stored at core location X'54'.

The SET card signals the end of the control statements. The system assignments for SYSRES and the communication device (SYSRDR or SYSLOG) are checked and permanently assigned. The system I/O tables are moved from their temporary location in high core to their permanent location in the supervisor area. This move overlays the two-device IPL I/O tables (that were built by $$A$IPL2) and finishes the IPL operation.

The end-of-job transient is loaded with a SVC 14 to initiate normal job processing. If the supervisor contains teleprocessing support, $$BEOJ3 is fetched; otherwise, $$BEOJ4 is fetched.

ADD -- Add a Device to the PUB Table

Operation	Operand
ADD	X'cuu'[(k)],devicetype[,X'ss']

X'cuu' = Channel and unit numbers in hexadecimal

k = S, if the device is switchable (is physically attached to two adjacent channels). The designated channel is the lower of the two channels.

k = 0-255 indicates the priority of the device, if the device cannot be switched. The highest priority is 0. If k is not given, a priority of 255 is assumed.

devicetype* = 2400T7 for 7-track, IBM 2400 Series Magnetic Tape Units.
2400T9 for 9-track, IBM 2400 Series Magnetic Tape Units.
1442N1 for 1442N1, IBM Card Read Punch.

X!ss' = Device specifications used for tape mode. If device specifications are not specified, X!ss' has the following set values:

X'C0' for 9-track tape
X'90' for 7-track tape
X'00' for nontapes

If you specify X'C8' for an 800 BPI single-density 9-track tape drive, you will save time during the tape OPEN. X'00', X'01', X'02', and X'03' are invalid as X'ss' for magnetic tape. These four values are used to specify SADxxx requirements for IBM 2702 lines*. X'ss' is required for the IBM 1255, 1259, 1270**, 1275**, 1412, 1419, and 1419P device types. It specifies the external interrupt bit (in the old PSW) used by this device to indicate READ COMPLETE. The specifications are:

X'01' PSW bit 31 X'08' PSW bit 28
X'02' PSW bit 30 X'10' PSW bit 27
X'04' PSW bit 29 X'20' PSW bit 26

*Device type codes and a complete list of density settings can be obtained from the Supervisor and Transients PLM.

**This device is not available in the United States of America.

The end-of-block character Ⓑ (alter code 5) must be given after each ADD statement if the communication device is a printer-keyboard.

DEL -- Delete a Device from PUB Table

Operation	Operand
DEL	X'cuu'

Where cuu is the channel and unit numbers, in hex, of the device to be deleted.

The end-of-block character Ⓑ (alter code 5) must be given after each DEL statement if the communication device is a printer-keyboard.

SET -- Set Date and Time of Day

Operation	Operand
SET	[DATE=n1][,CLOCK=n2]

The entries in the operand represent the following:

DATE = n1 Sets the system date to the specified value. n1 has one of the following formats:

mm/dd/yy
dd/mm/yy

Where mm specifies the month, dd specifies the day, and yy specifies the year. The format used is that selected when the system was generated.

CLOCK = n2 Must be given at IPL time if the timer feature is present. Sets the system clock to the specified value. n2 has the following format:

hh/mm/ss

Where hh specifies hours (00-23), mm specifies minutes (00-59), and ss specifies seconds (00-59).

Figure 4. ADD, DEL, and SET Statements

The job control program provides job-to-job transition for background programs and, optionally, for foreground programs. It also prepares program job steps for execution. (One or more programs can be executed within a single job. Each such execution is called a job step.) To obtain job-to-job transition for foreground programs, Batch-Job-Foreground (BJF) support must be specified at system generation time, and a BATCH command must be issued by the operator. Job Control then provides this feature in the partition specified by the operand of the BATCH command if the following minimum requirements are met:

- At least 10K of main storage in the partition.
- Separate system I/O files for the partition.

Job Control performs various functions on the basis of information provided in job control statements:

- Prepares programs for execution.
- Assigns device addresses to symbolic units.
- Sets up field in the communication region(s).
- Edits and stores volume and file label information.
- Prepares for restarting checkpointed programs.
- Clears the program area to binary zeros between job steps.
- Prepares input for the linkage editor program if the LINK option has been specified. The statements: ENTRY, ACTION, PHASE, and INCLUDE, when present in input stream, are copied to SYSLNK as card images. An INCLUDE statement with a blank operand causes the contents of SYSIPT to be copied to SYSLNK until a /* statement is read from SYSIPT. Blank cards from SYSIPT are ignored. This preparation for the linkage editor program is valid for the BG partition only, unless the supervisor includes the PCIL option. If the LINK option is not specified and the CATALR card precedes the PHASE card, these cards are copied to SYSPCH as card images preceding the compilation.

The job control program is executed in the program area and is overlaid by the job step it is preparing for execution. A JOB statement in the input stream marks the beginning of a job and a /& statement marks the end of a job. An EXEC statement calls for execution of a job step. A job step is normally ended with the EOJ macro.

I/O FLOW

The I/O flow for the job control program consists of:

- Input
  SYSRDR
  SYSIPT
- Output
  SYSLST
  SYSLNK
- I/O
  SYSLOG

PROGRAM FLOW

Functionally, Job Control consists of eight phases and one B-transient that are identified as $JOBCTLA, $JOBCTLD, $JOBCTLF, $JOBCTLG, $JOBCTLJ, $JOBCTLK, $JOBCTLM, $JOBCTLN, and $$BLSTIO.

$JOBCTLA (CHART 04)

This phase is the initial entry into Job Control. It is loaded every time Job Control is fetched and is considered the root phase. (It is resident in main storage at all times during job control execution and contains routines that are used by the other phases of Job Control.)

Job control input is read from SYSRDR or SYSLOG depending on the setting of the job control input switch (COMREG+56, bit 2). As each control statement is read, it is analyzed to determine which of the processing routines is to be used. The phase containing the correct processing

routine is loaded if it is not already in main storage as a result of the previous control statement.

Figure 5 represents the storage allocation for Job Control.

$JOBCTLD (CHARTS 05 AND 06)

Contains the processing routines for the following control statements:

1. ASSGN
2. CLOSE

$JOBCTLF (CHART 07)

Contains the processing routines for the following control statements:

1. DVCDN
2. DVCUP
3. LISTIO
4. RESET
5. UNA
6. UNBATCH
7. MAP

$JOBCTLG (CHARTS 08 AND 09)

Contains the processing routines for the following control statements:

1. CANCEL
2. /& (EOJ)
3. EXEC
4. JOB
5. OPTION

$JOBCTLJ (CHARTS 10, 11, AND 12)

Contains processing routines for the following control statements:

1. ACTION
2. ENTRY
3. PHASE
4. INCLUDE
5. DATE
6. SET
7. UPSI
8. MTC
9. UCS
10. HOLD
11. RELSE
12. CATALR card (not a control statement)
13. NOLOG
14. LOG
15. STOP
16. PAUSE
17. ALLOC

$JOBCTLK (CHARTS 13 AND 14)

Contains processing routines for the following control statements:

1. RSTRT
2. LBLTYP
3. VOL
4. TPLAB
5. DLAB
6. XTENT
7. TLBL
8. DLBL
9. EXTENT

$JOBCTLM (CHART 15)

Initializes the recorder file and contains processing routines for the ROD control statement.

$JOBCTLN (CHART 16)

Provides interface between the DOS system and $JOBACCT to allow the user to access Job Accounting information. This phase contains processing routines for the following conditions:

1. Normal end of job.
2. Simulated end of job.
3. End of job and pause.
4. EXEC card encountered.

Note: $JOBCTLN interfaces with $JOBACCT phase supplied by user.

$$BLSTIO

This B-transient contains subroutines used by DVCDN and LISTIO control statement processors of $JOBCTLF. When required by these processors, $$BLSTIO is fetched (SVC 2) into supervisor B-transient area.

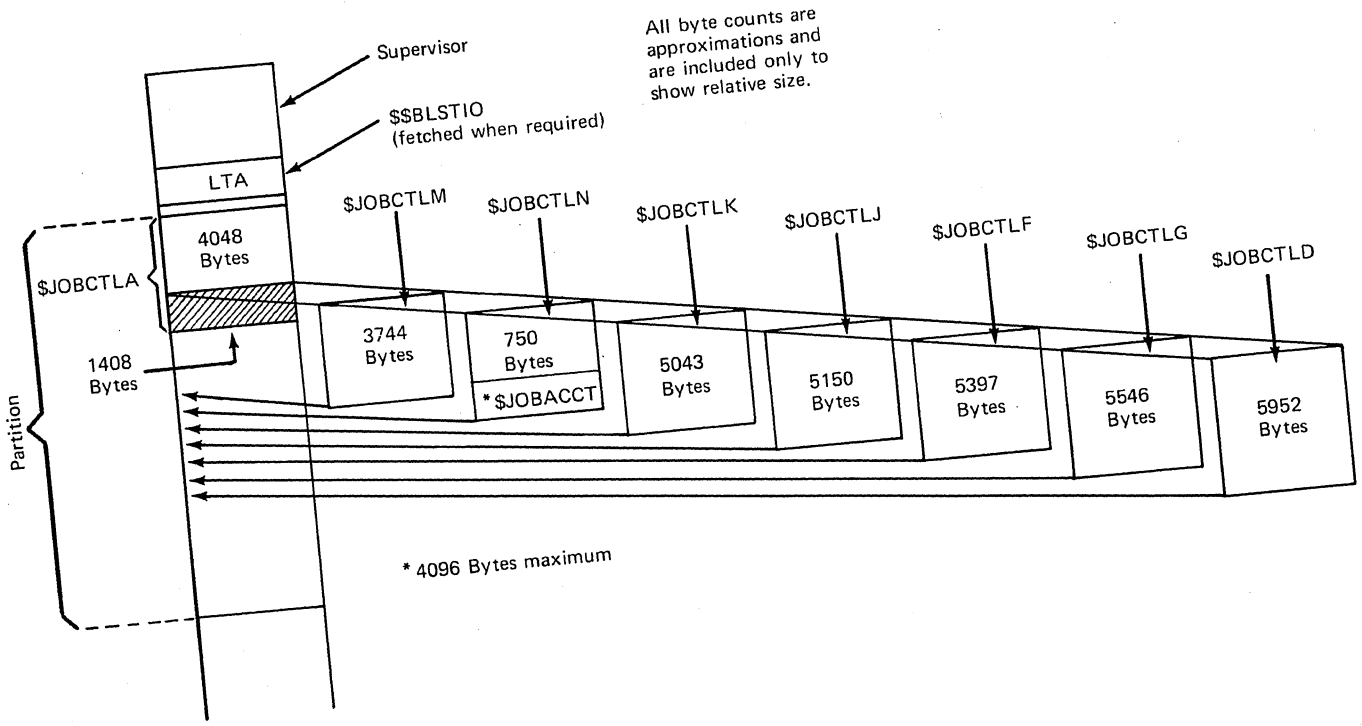


Figure 5. Job Control Storage Allocation

Chart 01. Initial Program Load ($$A$IPL1 and $$A$IPL2)

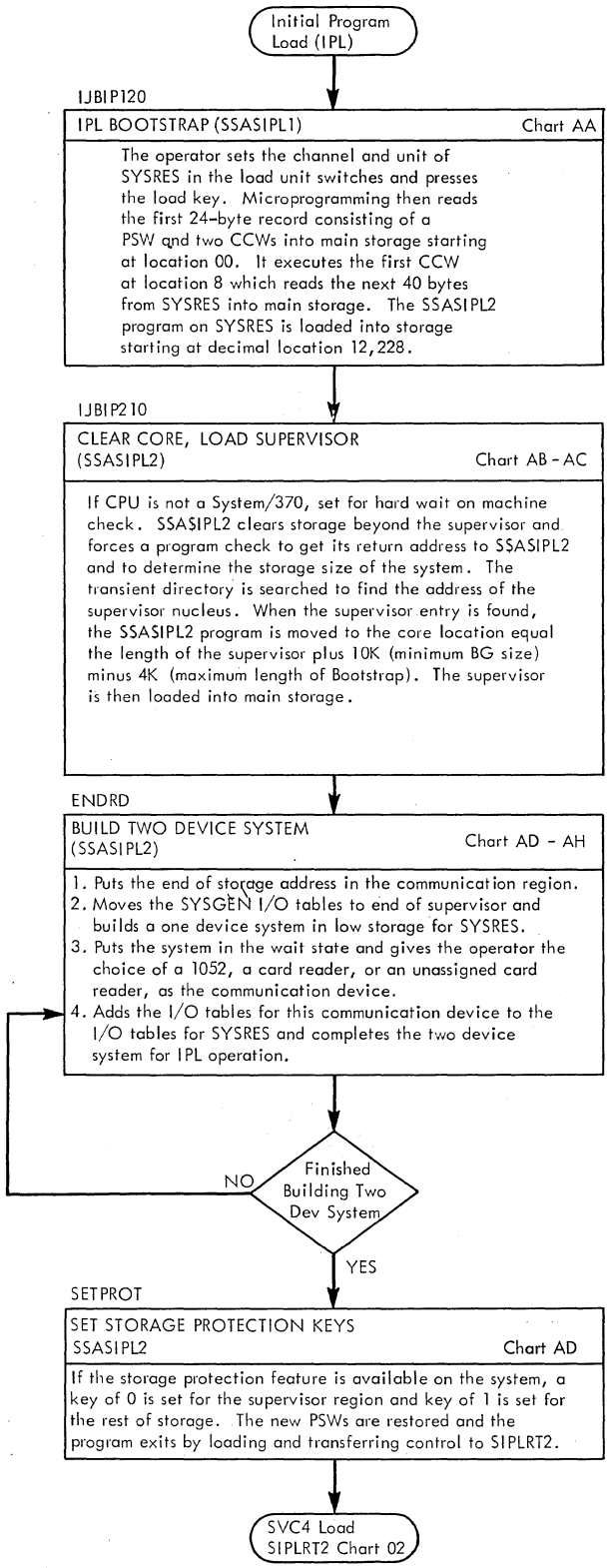


Chart 02. Initial Program Load ($IPLRT2)

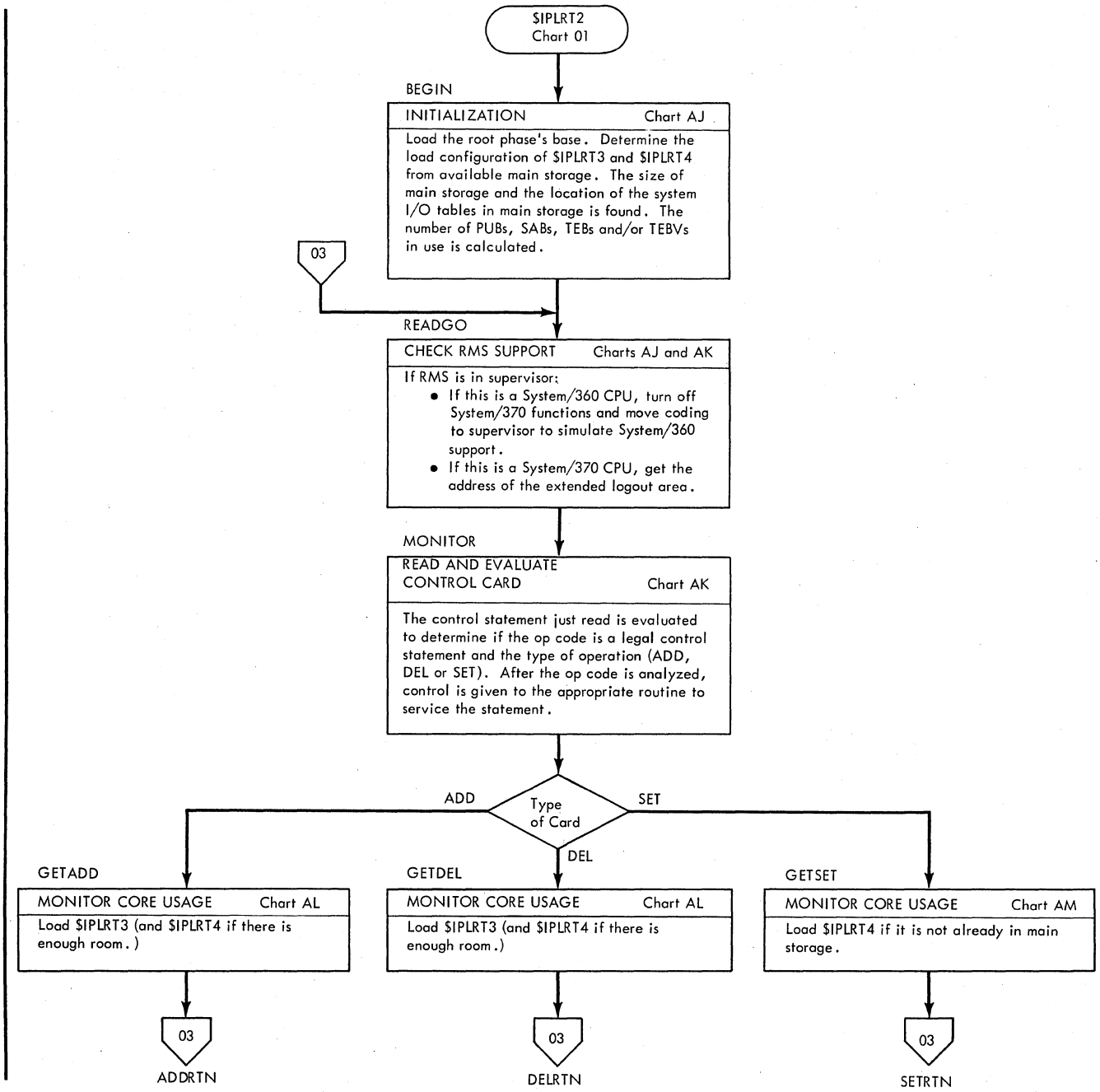


Chart 03. Initial Program Load ($IPLRT3 and $IPLRT4)

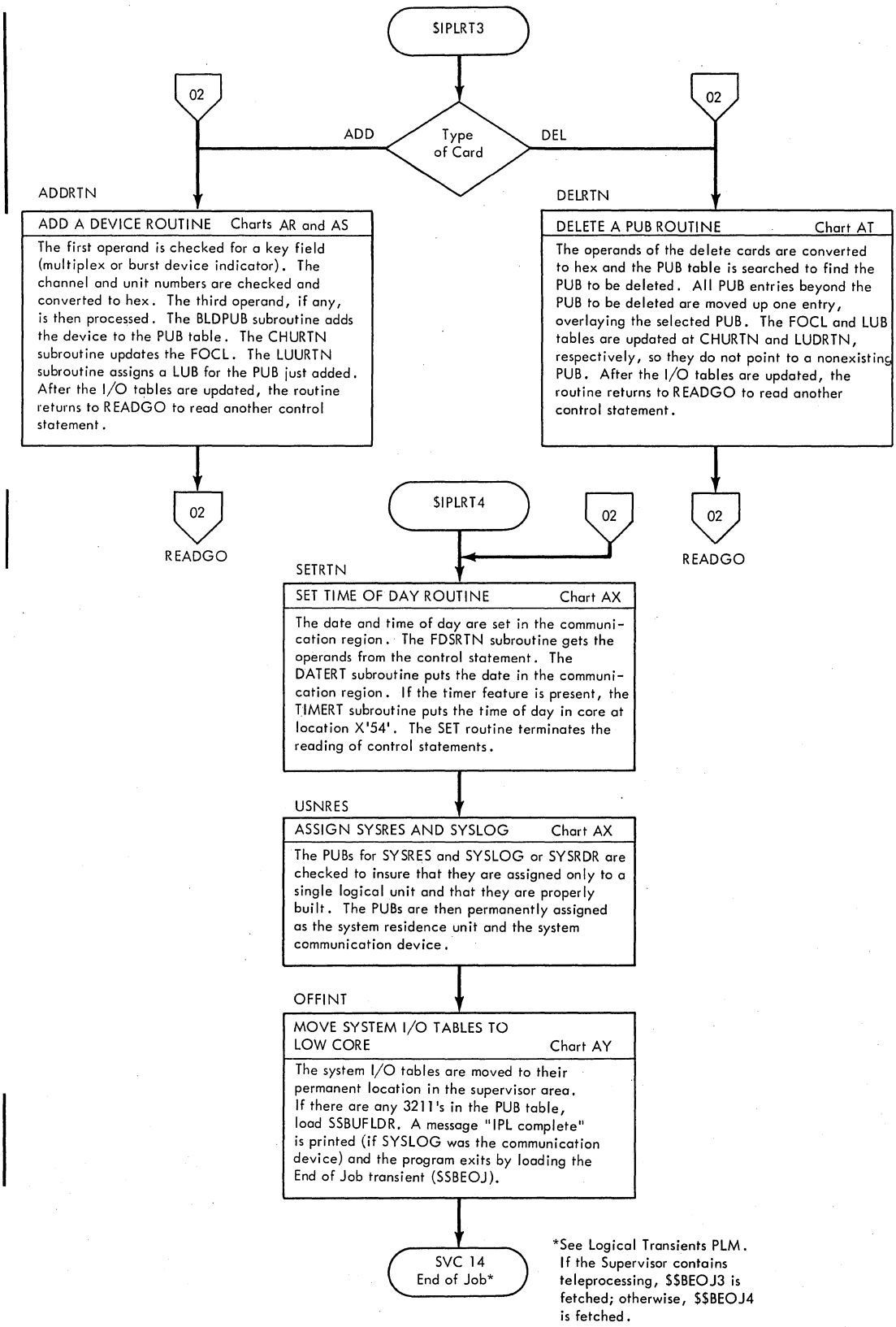


Chart 04. Job Control ($JOBCTLA) Root Phase

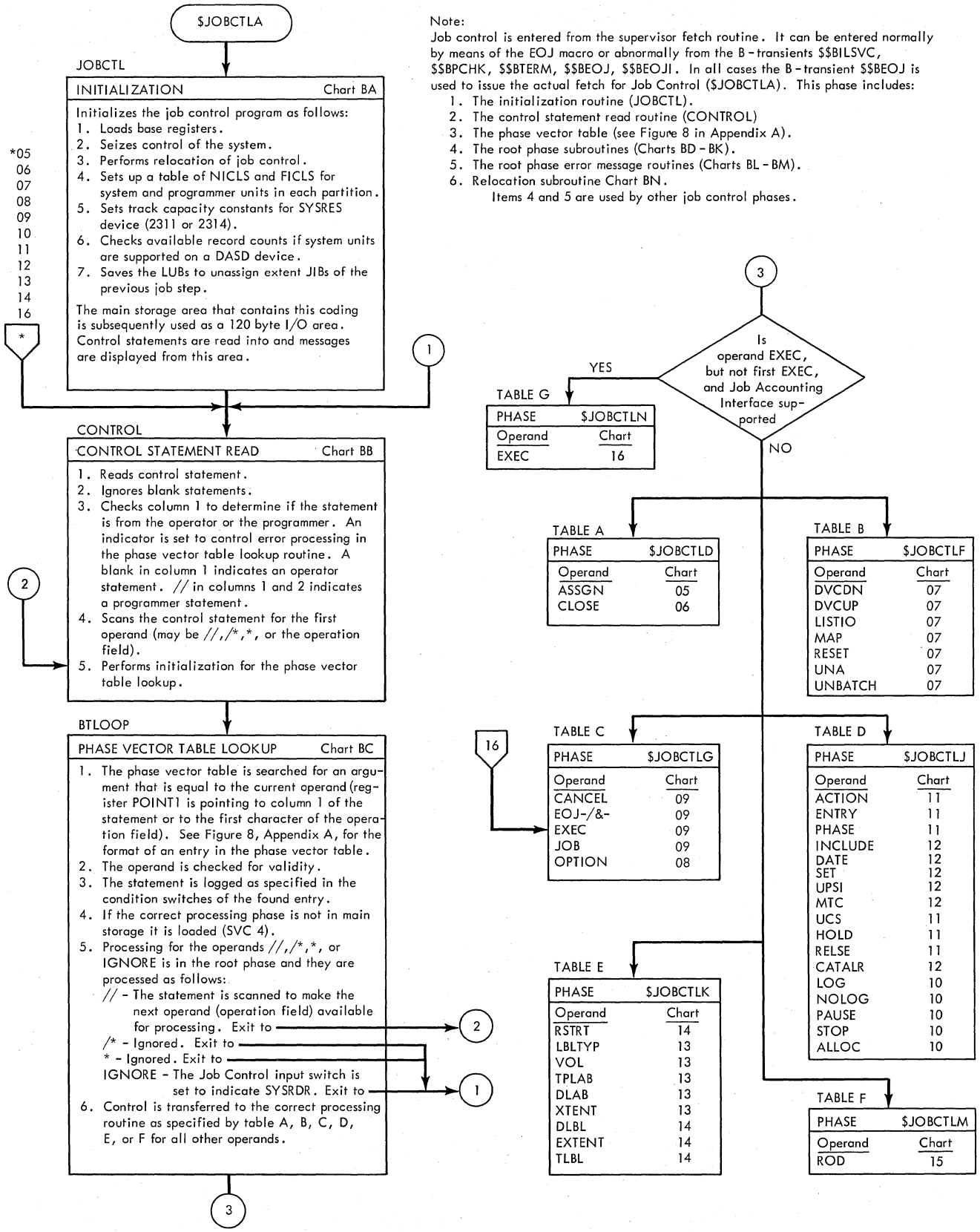


Chart 05. Job Control ($JOBCTLD) Statement Processor (Part 1 of 2)

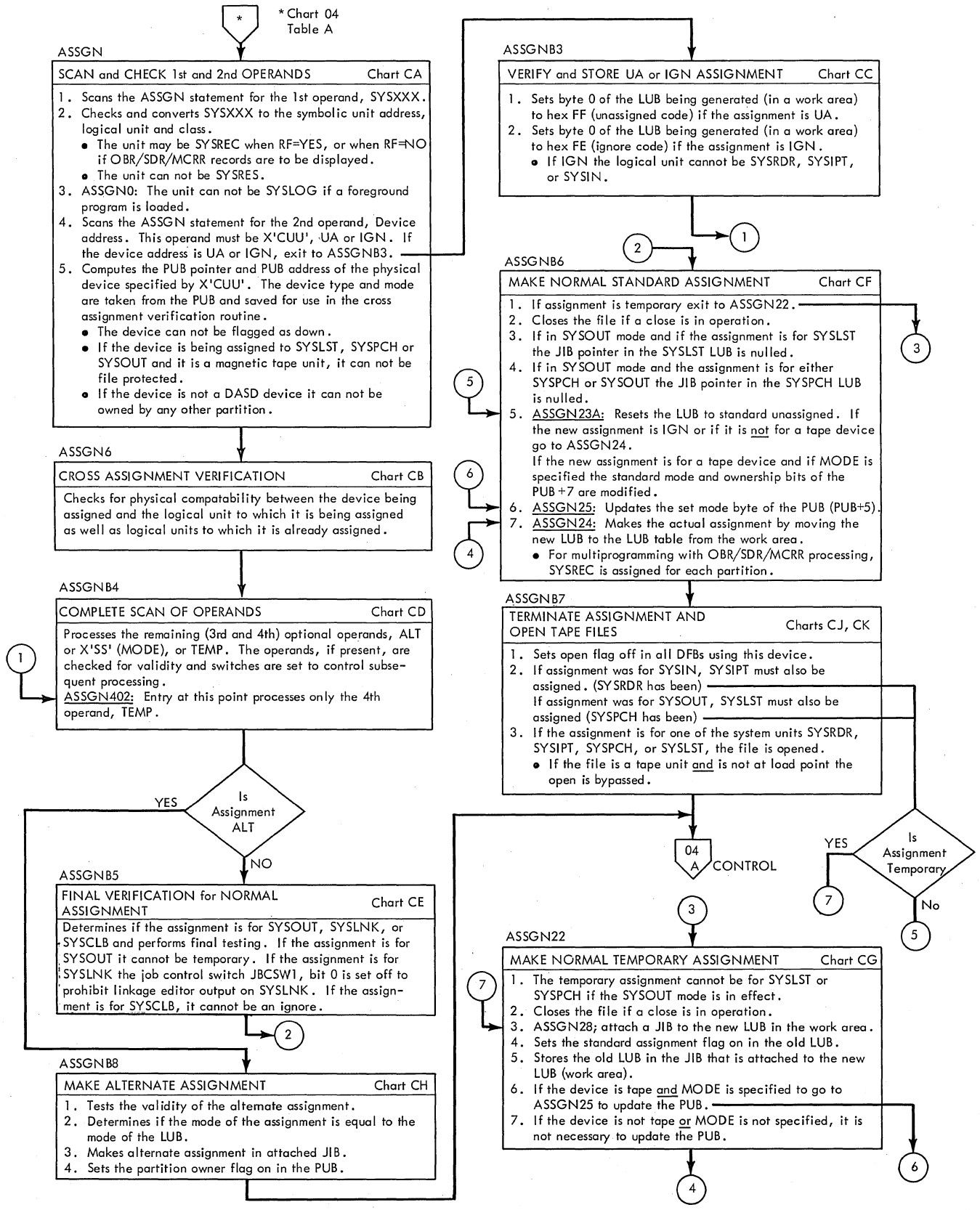


Chart 06. Job Control ($JOBCTLD) Statement Processor (Part 2 of 2)

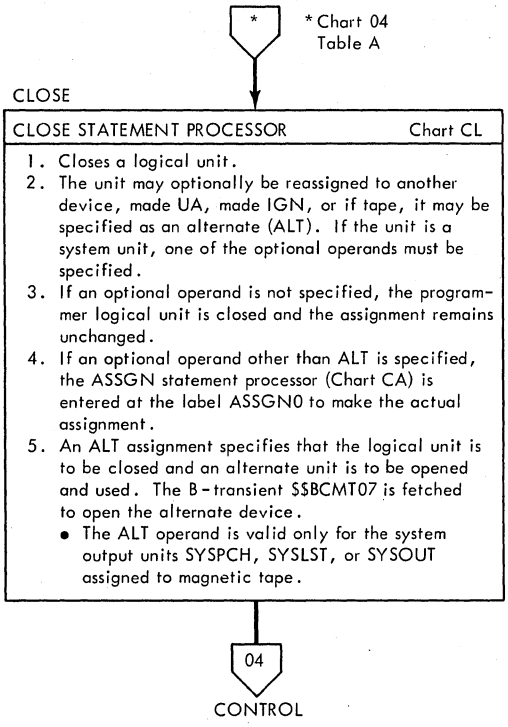


Chart 07. Job Control ($JOBCTLF) Statement Processor

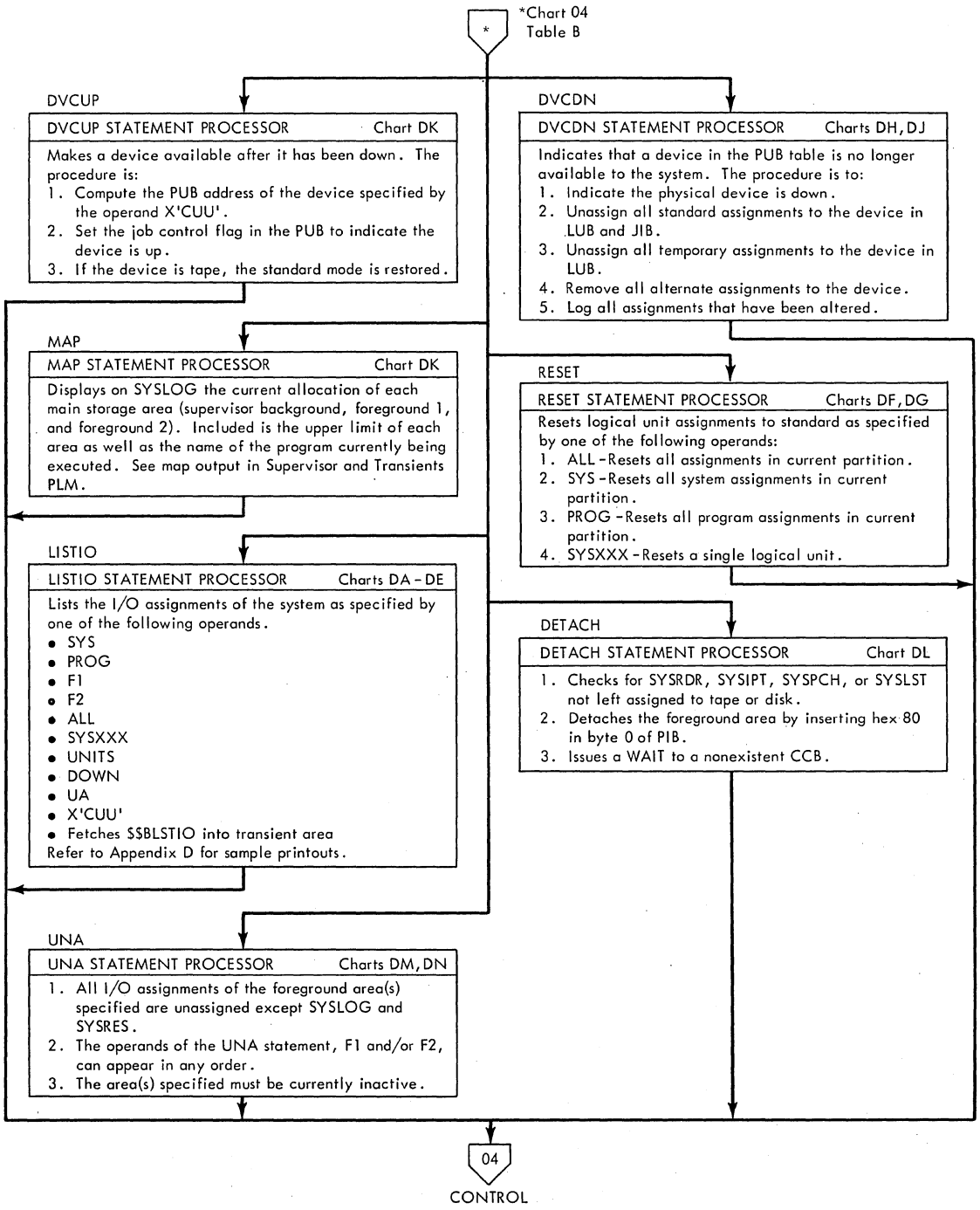


Chart 08. Job Control ($JOBCTLG) Statement Processor (Part 1 of 2)

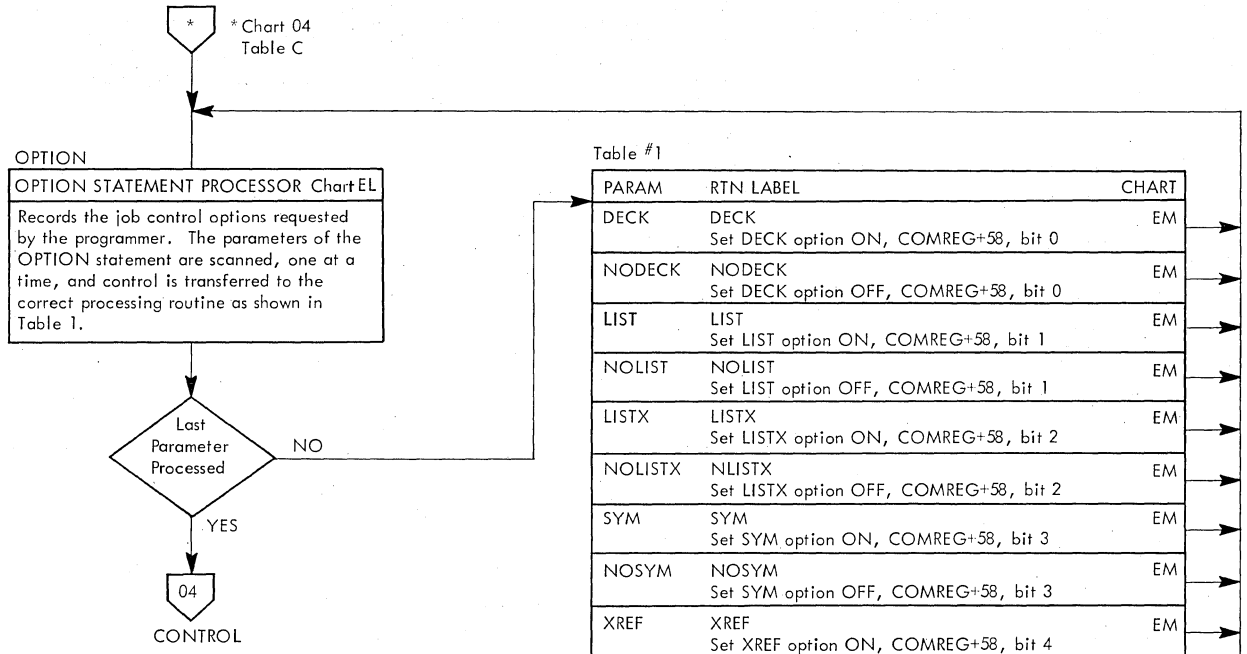


Table #1

PARAM	RTN LABEL	CHART
DECK	DECK Set DECK option ON, COMREG+58, bit 0	EM
NODECK	NODECK Set DECK option OFF, COMREG+58, bit 0	EM
LIST	LIST Set LIST option ON, COMREG+58, bit 1	EM
NOLIST	NOLIST Set LIST option OFF, COMREG+58, bit 1	EM
LISTX	LISTX Set LISTX option ON, COMREG+58, bit 2	EM
NOLISTX	NLISTX Set LISTX option OFF, COMREG+58, bit 2	EM
SYM	SYM Set SYM option ON, COMREG+58, bit 3	EM
NOSYM	NOSYM Set SYM option OFF, COMREG+58, bit 3	EM
XREF	XREF Set XREF option ON, COMREG+58, bit 4	EM
NOXREF	NOXREF Set XREF option OFF, COMREG+58, bit 4	EM
ERRS	ERRS Set ERRS option ON, COMREG+58, bit 5	EM
NOERRS	NOERRS Set ERRS option OFF, COMREG+58, bit 5	EM
48C	C48 Set 48 character set option ON, COMREG+58, bit 6	EM
60C	C60 Set 60 character set option ON, COMREG+58, bit 6	EM
NOLOG	OPTNLG Set the SYSLST-LOG option OFF, COMREG+59, bit 3	EM
USRLABEL	USRLBL Initialize Job Control Label Processing for USER LABEL Processing.	EN
STDLABEL	STDLBL Initialize Job Control Label Processing for STANDARD LABEL Processing. Labels Available to Programs in ANY Partition.	EN
PARSTD	PARSTD Initialize Job Control Label Processing for PARTITION STANDARD LABEL Processing. Labels Available to Programs in Only ONE Partition.	EN
DUMP	DUMP Set Job control dump option ON, COMREG+59, bit 1	EP
NODUMP	NODUMP Set job control dump option OFF, COMREG+59, bit 1	EP
LINK	LINK Set LINKEDT control bits, COMREG+57, bits 0,2 Valid for BG only	EN
NOLINK	NOLINK Set LINK option OFF, COMREG+57, bit 0	EP
LOG	OPTLOG Set the SYSLST-LOG option ON, COMREG+59, bit 3	EP
CATAL	CATAL Set LINKEDT control bits, COMREG+57, bits 0,2,3 Valid for BG only	EN
SYSPARM	SYSPRM Save character string	EP

Chart 09. Job Control ($JOBCTLG) Statement Processor (Part 2 of 2)

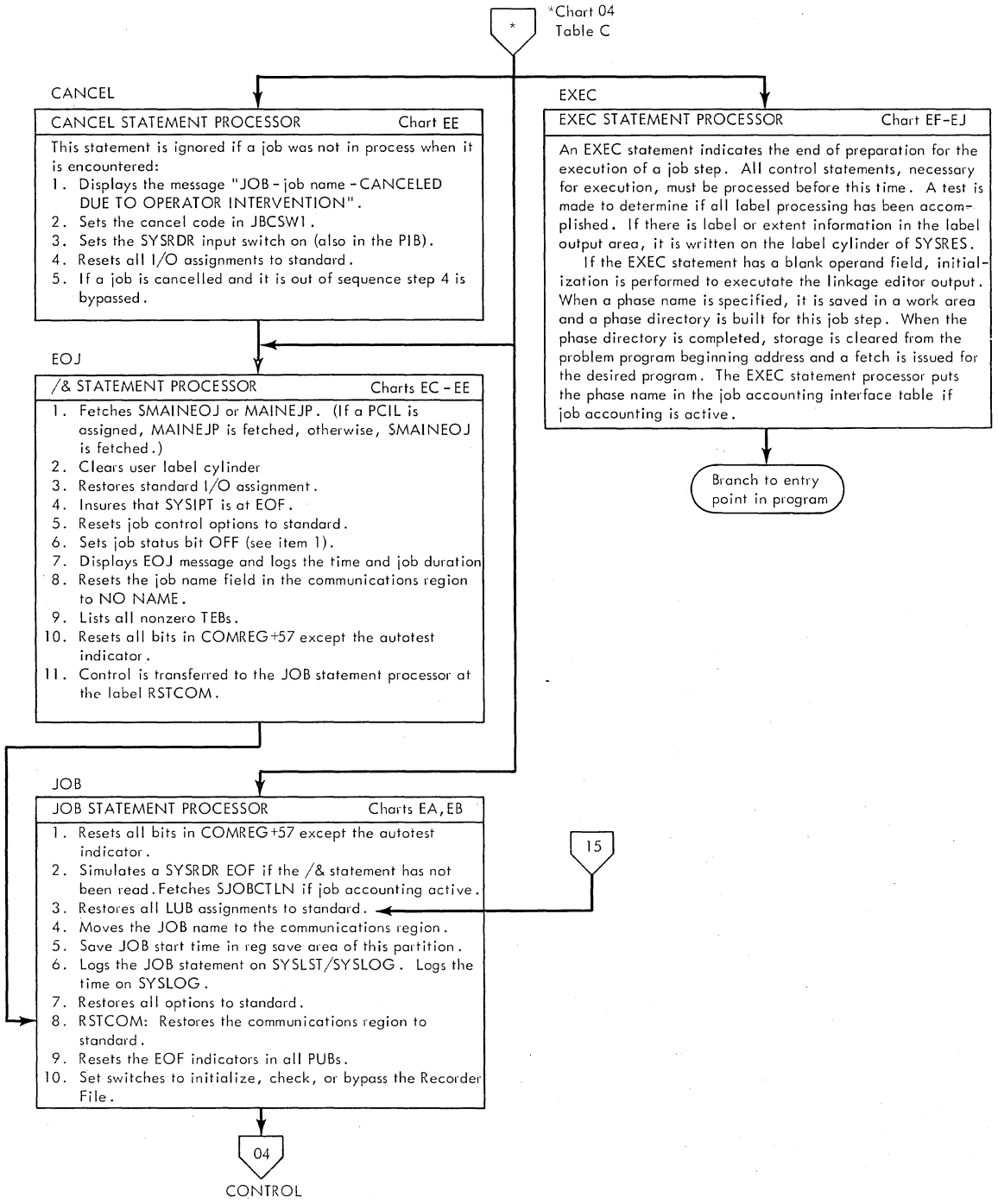


Chart 10. Job Control ($JOBCTLJ) Statement Processor (Part 1 of 3)

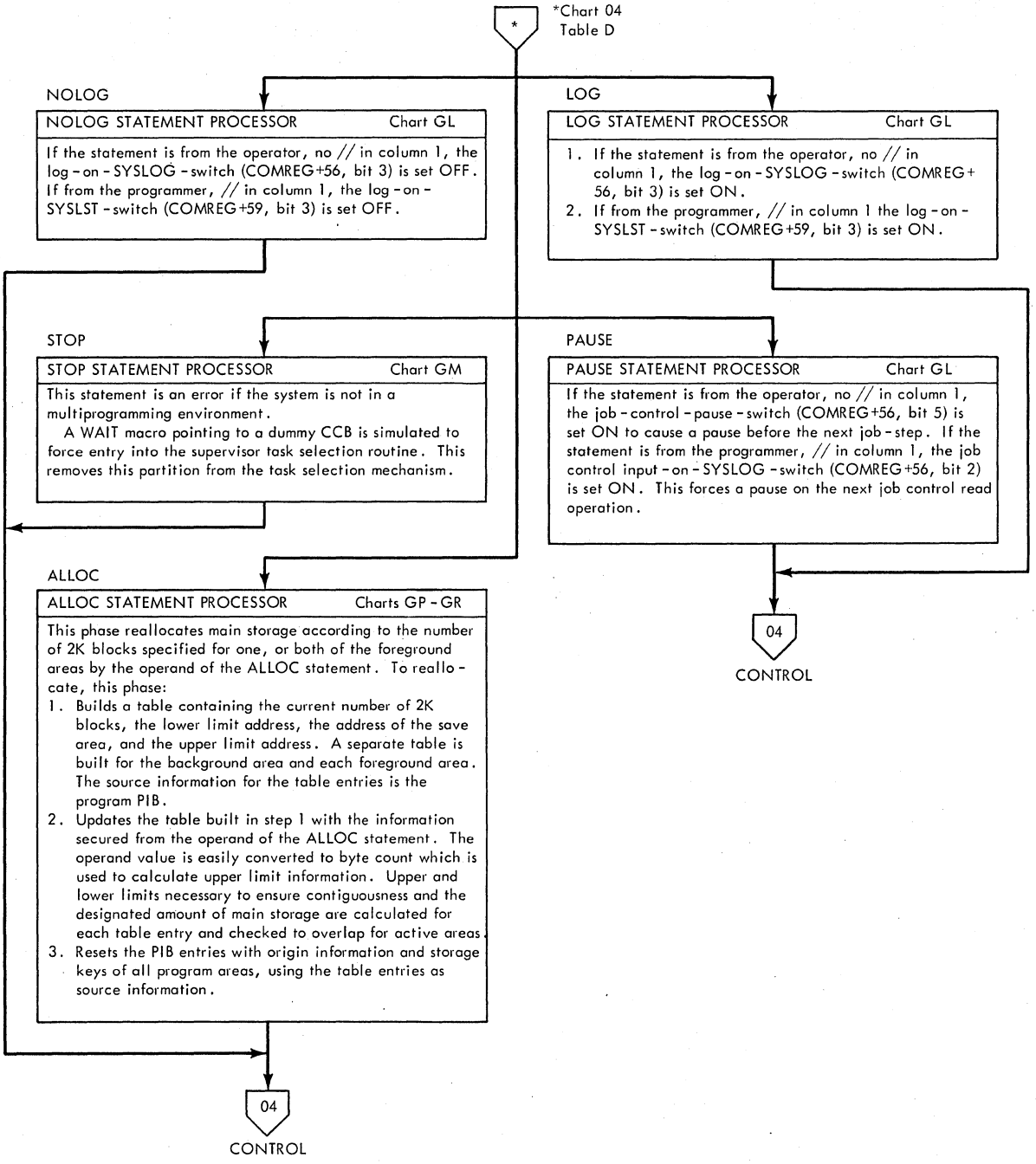


Chart 11. Job Control ($JOBCTLJ) Statement Processor (Part 2 of 3)

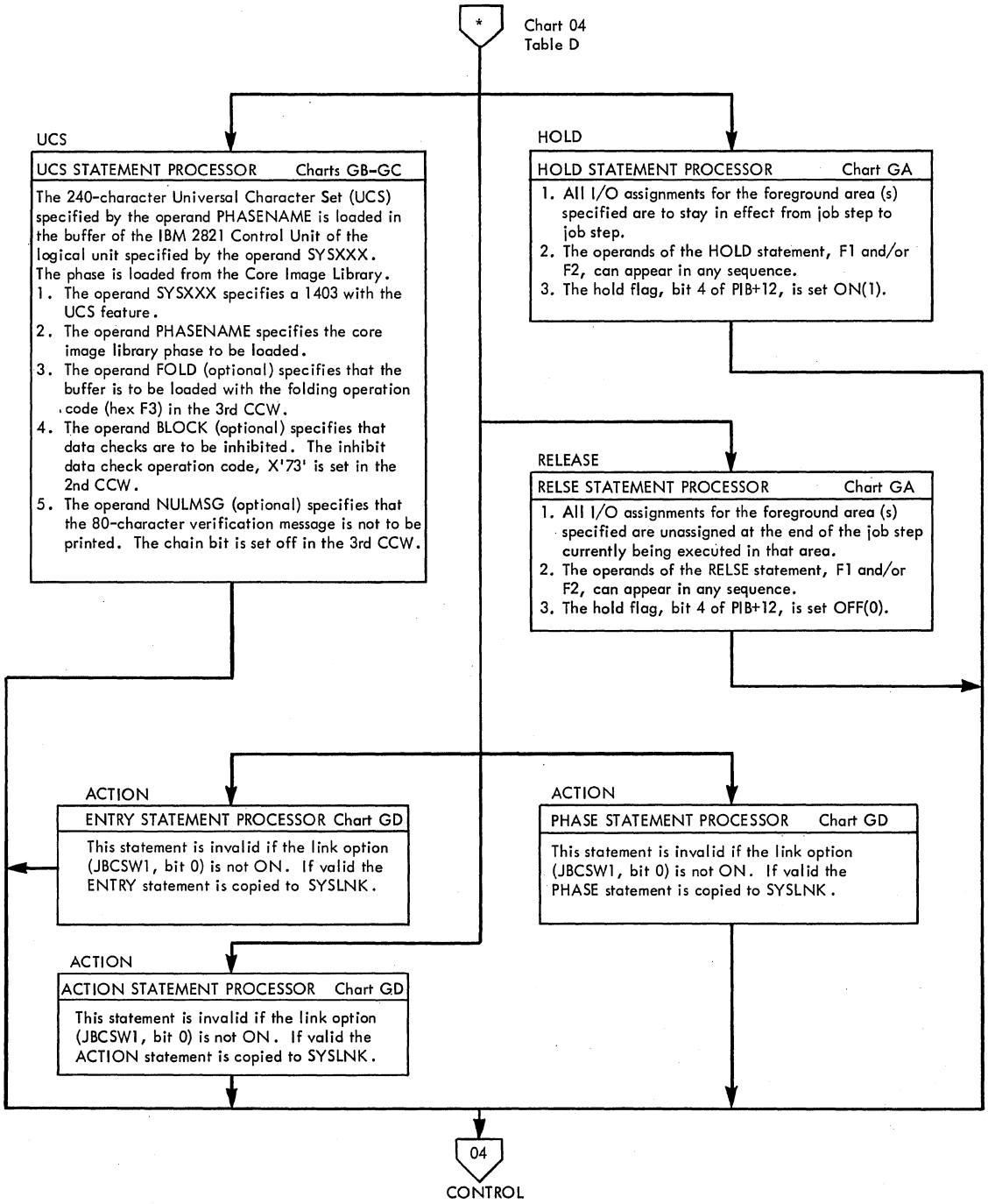


Chart 12. Job Control ($JOBCTLJ) Statement Processor (Part 3 of 3)

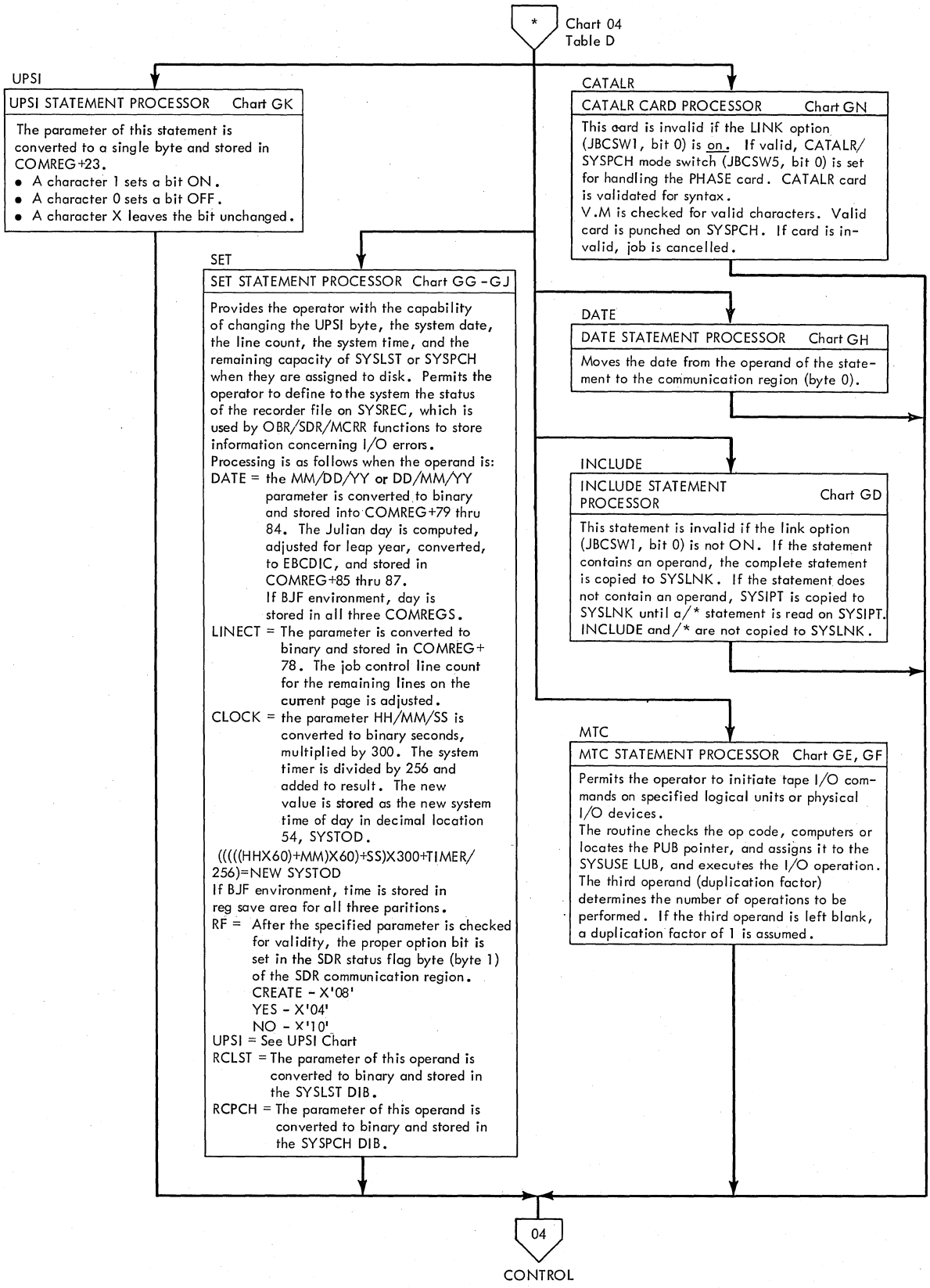


Chart 13. Job Control ($JOBCTLK) Statement Processor (Part 1 of 2)

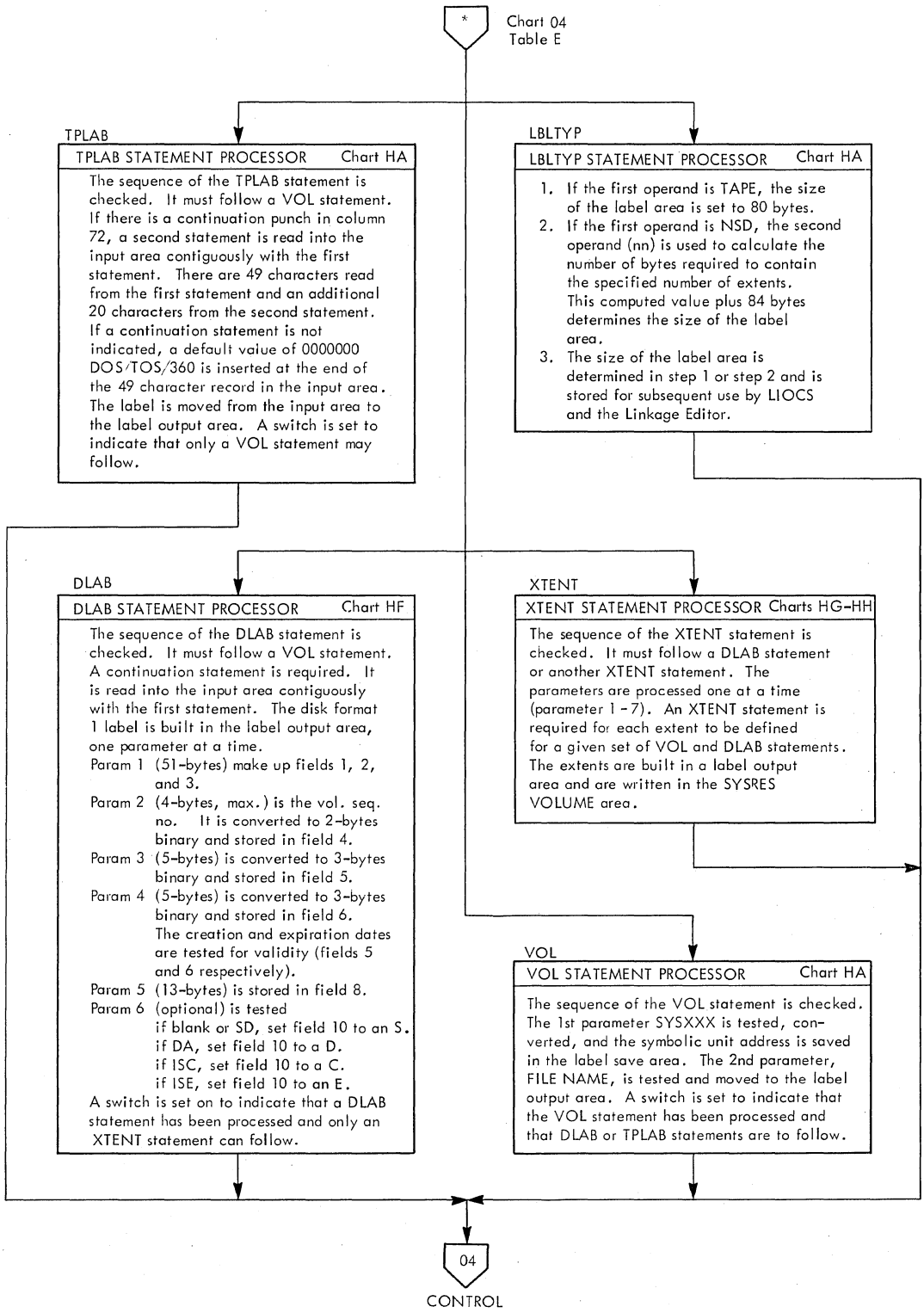


Chart 14. Job Control ($JOBCTLK) Statement Processor (Part 2 of 2)

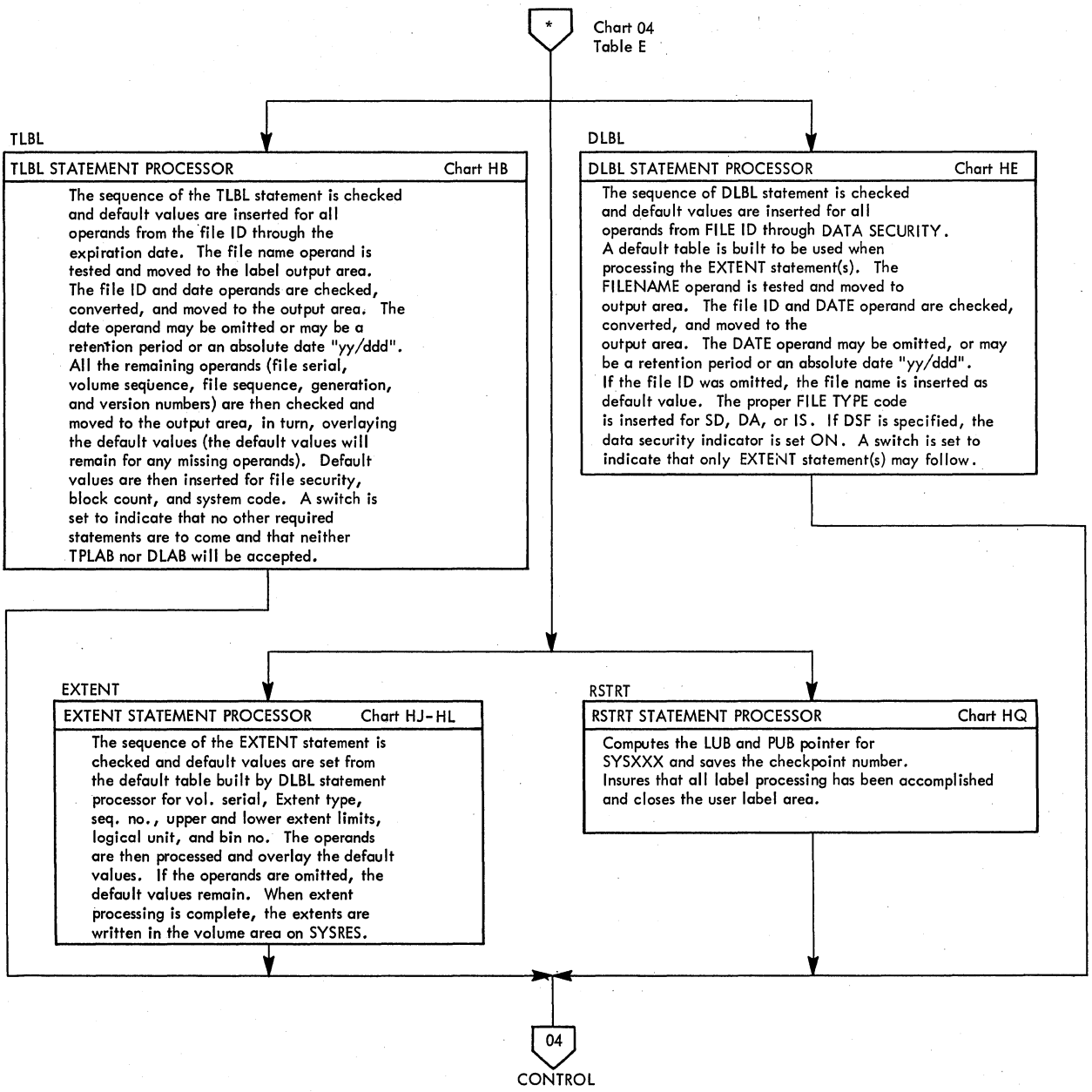


Chart 15. Job Control ($JOBCTLM) Statement Processor

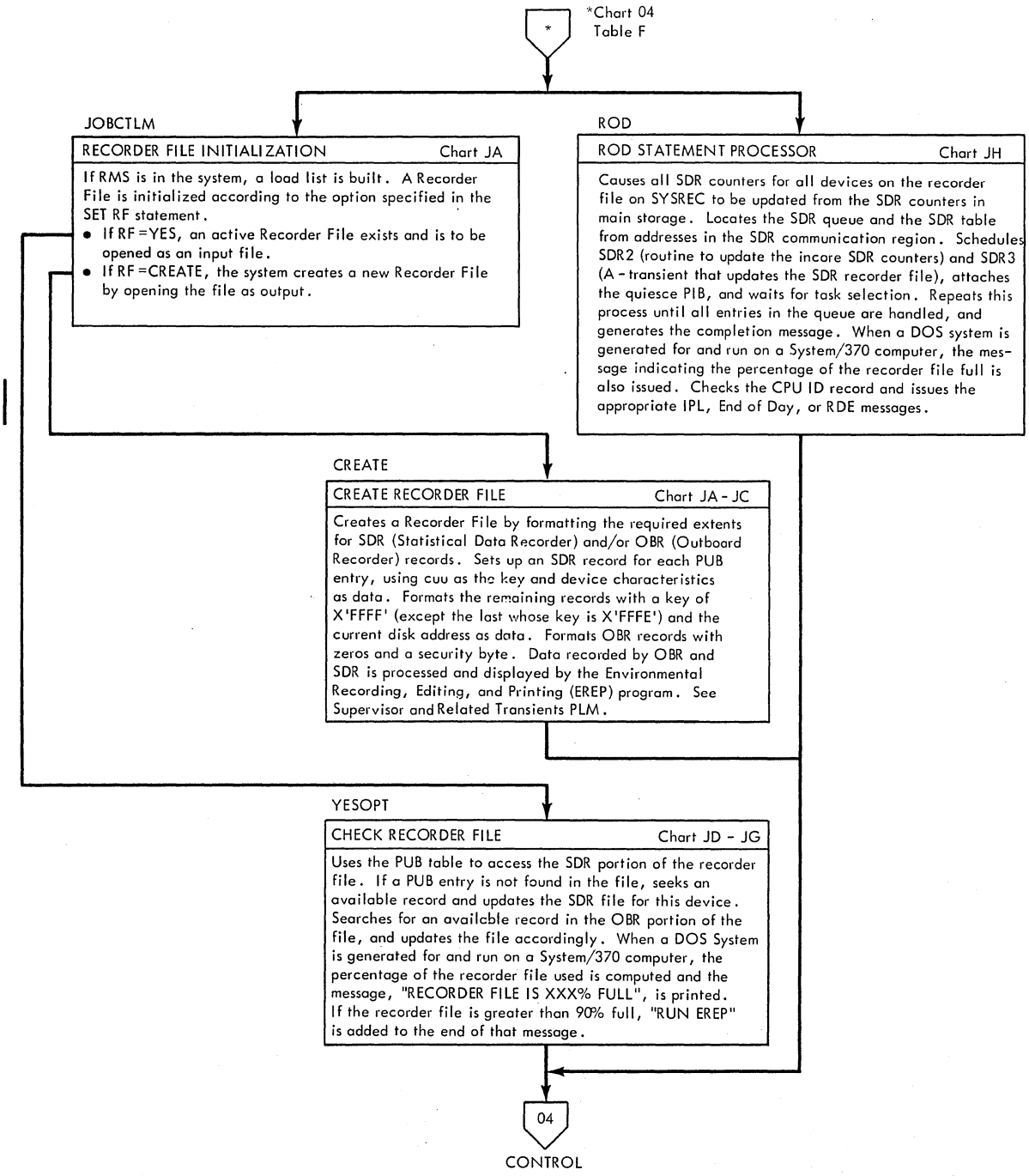


Chart 16. Job Control ($JOBCTLN) Statement Processor

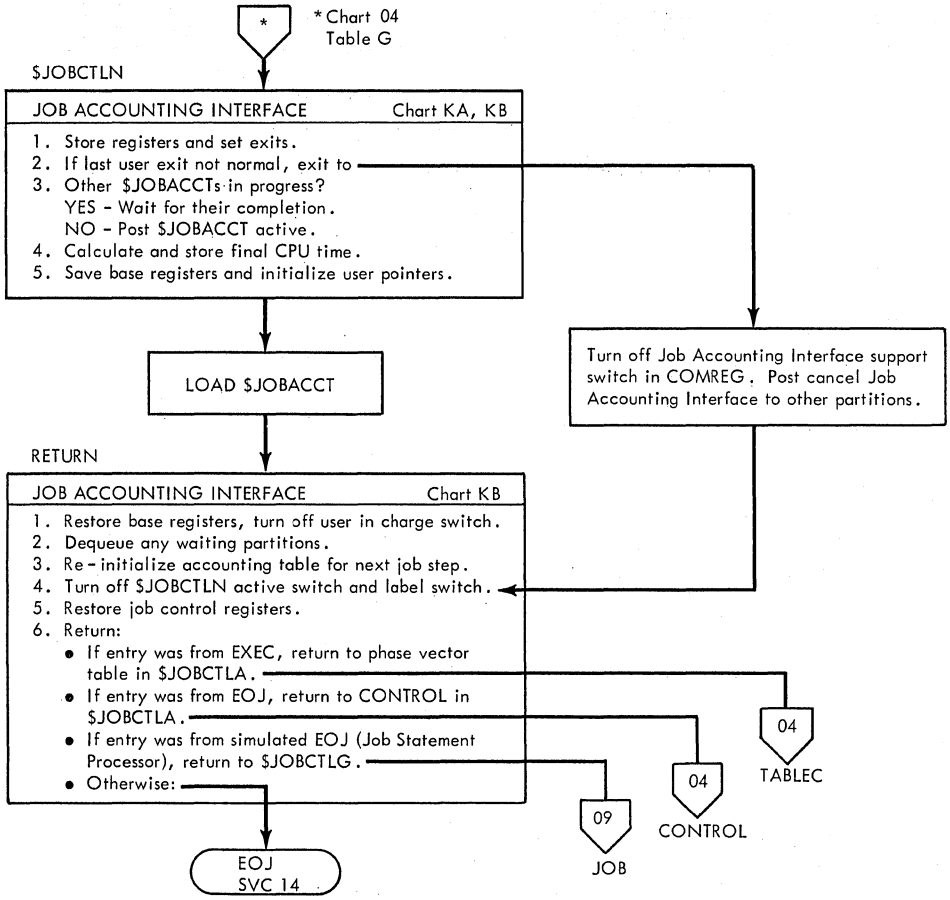
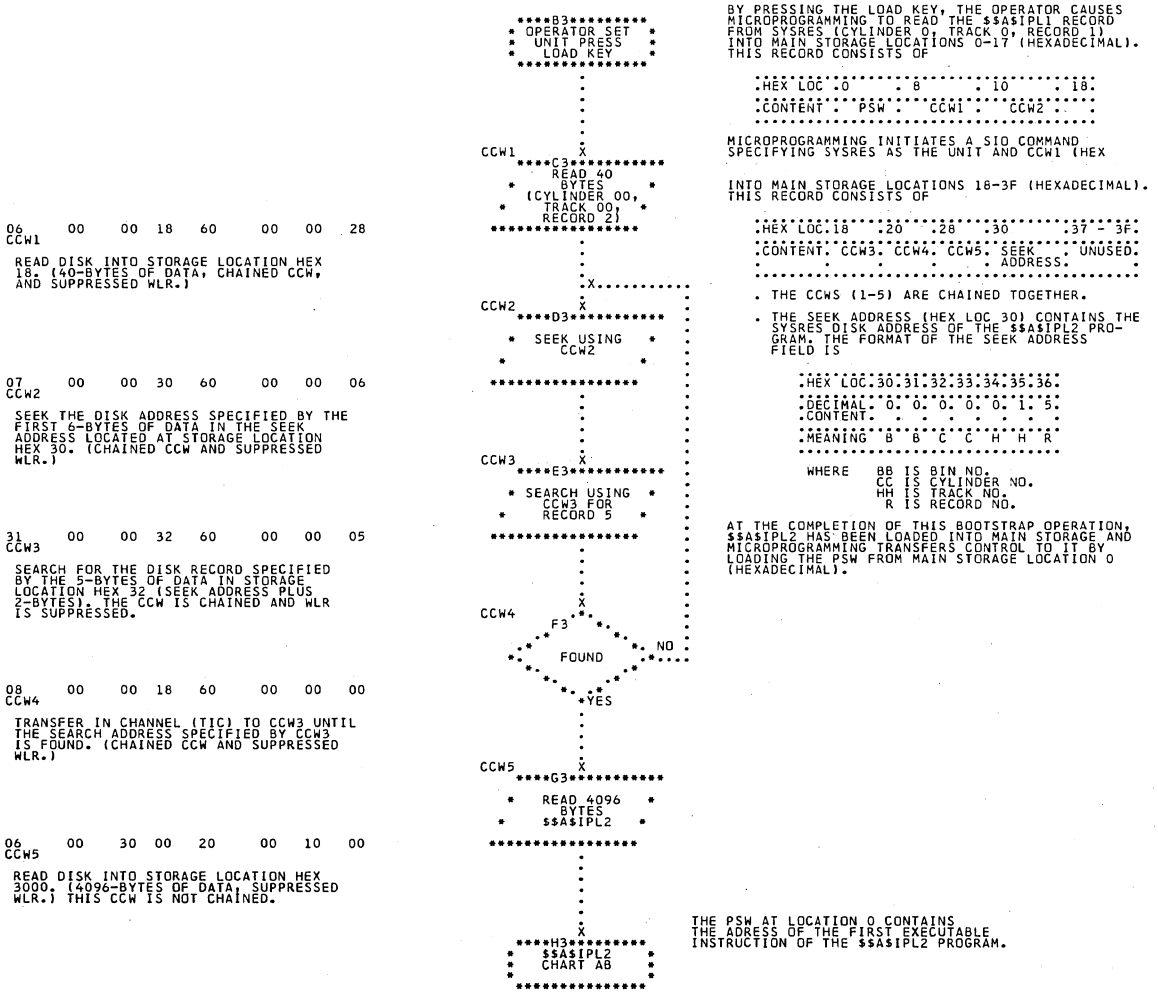


Chart AA. $$A$IPL1 - IPL Bootstrap

Refer to Chart 01.



BY PRESSING THE LOAD KEY, THE OPERATOR CAUSES MICROPROGRAMMING TO READ THE $$A$IPL1 RECORD FROM SYSRES (CYLINDER 0, TRACK 0, RECORD 1) INTO MAIN STORAGE LOCATIONS 0-17 (HEXADECIMAL). THIS RECORD CONSISTS OF

HEX LOC	0	8	10	18
CONTENT	PSW	CCW1	CCW2	

MICROPROGRAMMING INITIATES A SIO COMMAND SPECIFYING SYSRES AS THE UNIT AND CCW1 (HEX

INTO MAIN STORAGE LOCATIONS 18-3F (HEXADECIMAL). THIS RECORD CONSISTS OF

HEX LOC	18	20	28	30	37 - 3F
CONTENT	CCW3	CCW4	CCW5	SEEK ADDRESS	UNUSED

- THE CCWS (1-5) ARE CHAINED TOGETHER.
- THE SEEK ADDRESS (HEX LOC 30) CONTAINS THE SYSRES DISK ADDRESS OF THE $$A$IPL2 PROGRAM. THE FORMAT OF THE SEEK ADDRESS FIELD IS

HEX LOC	30	31	32	33	34	35	36
DECIMAL	0	0	0	0	0	1	5
CONTENT	.	.	.	.	.	.	.
MEANING	B	B	C	C	H	H	R

WHERE BB IS BIN NO.
CC IS CYLINDER NO.
HH IS TRACK NO.
R IS RECORD NO.

AT THE COMPLETION OF THIS BOOTSTRAP OPERATION, $$A$IPL2 HAS BEEN LOADED INTO MAIN STORAGE AND MICROPROGRAMMING TRANSFERS CONTROL TO IT BY LOADING THE PSW FROM MAIN STORAGE LOCATION 0 (HEXADECIMAL).

THE PSW AT LOCATION 0 CONTAINS THE ADRESS OF THE FIRST EXECUTABLE INSTRUCTION OF THE $$A$IPL2 PROGRAM.

06 00 00 18 60 00 00 28
CCW1

READ DISK INTO STORAGE LOCATION HEX 18. (40-BYTES OF DATA, CHAINED CCW, AND SUPPRESSED WLR.)

07 00 00 30 60 00 00 06
CCW2

SEEK THE DISK ADDRESS SPECIFIED BY THE FIRST 6-BYTES OF DATA IN THE SEEK ADDRESS LOCATED AT STORAGE LOCATION HEX 30. (CHAINED CCW AND SUPPRESSED WLR.)

31 00 00 32 60 00 00 05
CCW3

SEARCH FOR THE DISK RECORD SPECIFIED BY THE 5-BYTES OF DATA IN STORAGE LOCATION HEX 32 (SEEK ADDRESS PLUS 2-BYTES). THE CCW IS CHAINED AND WLR IS SUPPRESSED.

08 00 00 18 60 00 00 00
CCW4

TRANSFER IN CHANNEL (TIC) TO CCW3 UNTIL THE SEARCH ADDRESS SPECIFIED BY CCW3 IS FOUND. (CHAINED CCW AND SUPPRESSED WLR.)

06 00 30 00 20 00 10 00
CCW5

READ DISK INTO STORAGE LOCATION HEX 3000. (4096-BYTES OF DATA, SUPPRESSED WLR.) THIS CCW IS NOT CHAINED.

Chart AB. $$A$IPL2 - Clear Storage and Load Supervisor (Part 1 of 2)
Refer to Chart 01.

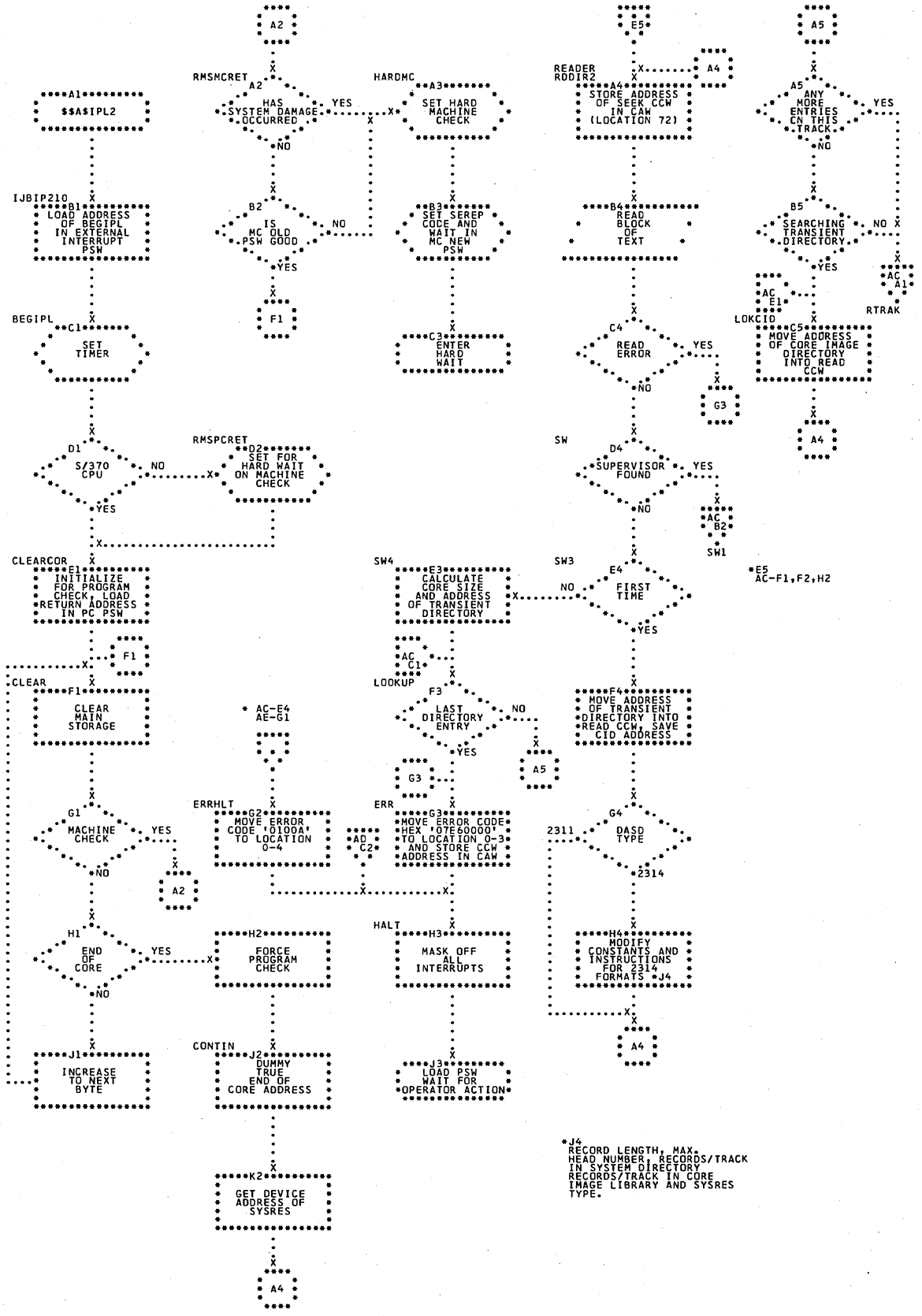


Chart AC. $$A$IPL2 - Load Supervisor (Part 2 of 2)
Refer to Chart 01.

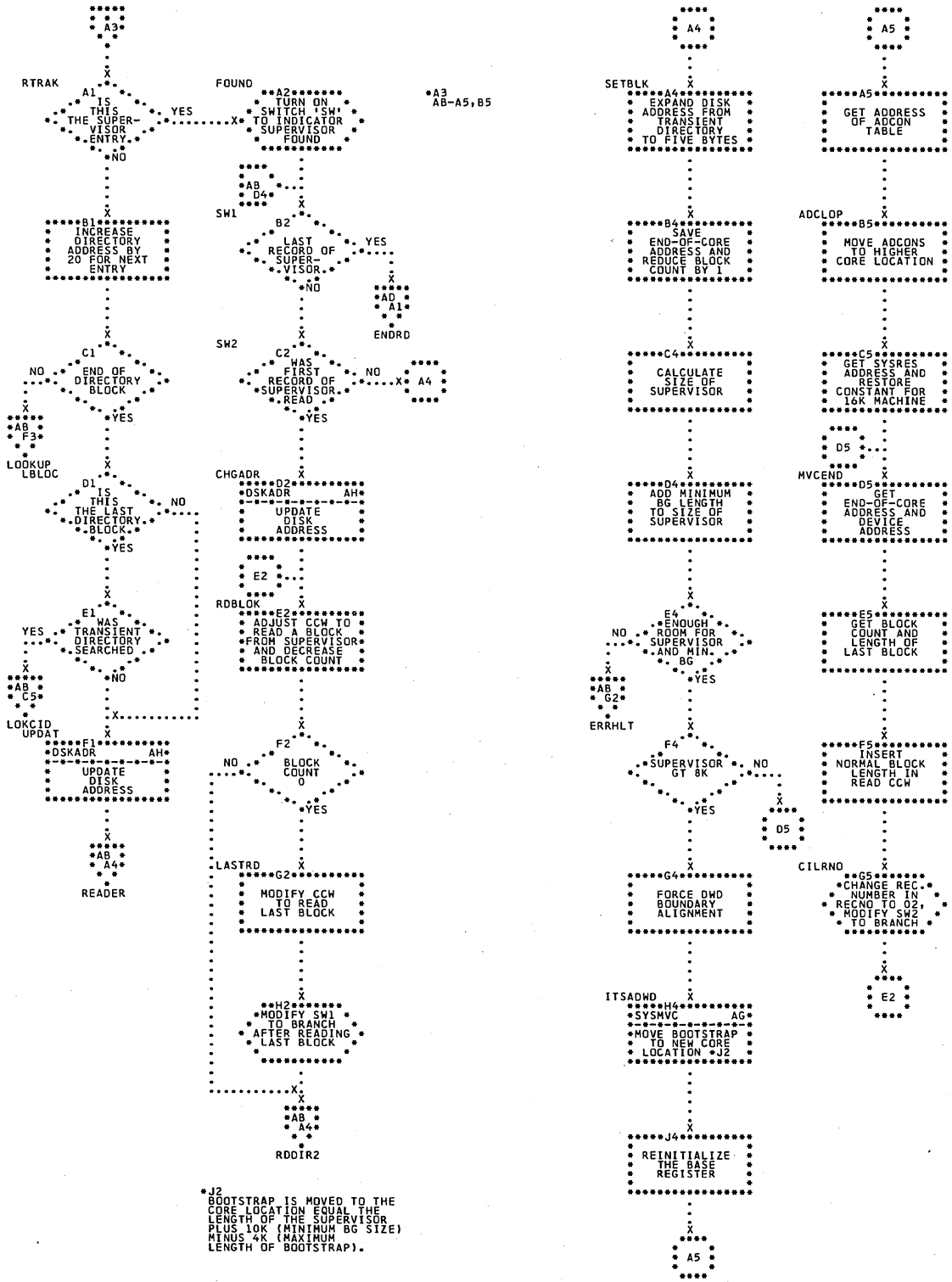


Chart AD. $$A$IPL2 - Build Two-Device System
Refer to Chart 01.

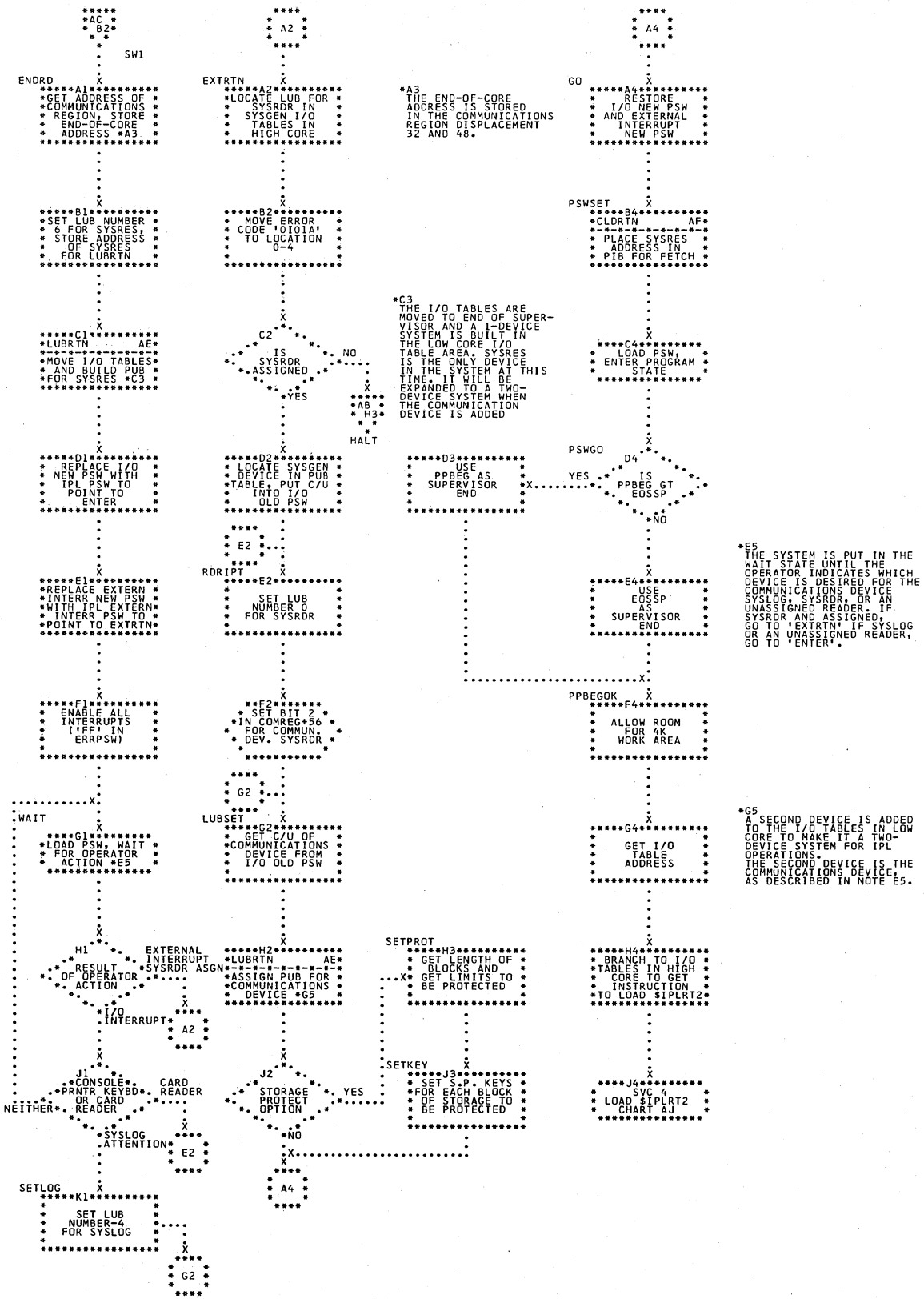


Chart AE. $$A$IPL2 - Move I/O Tables
Refer to Chart 01.

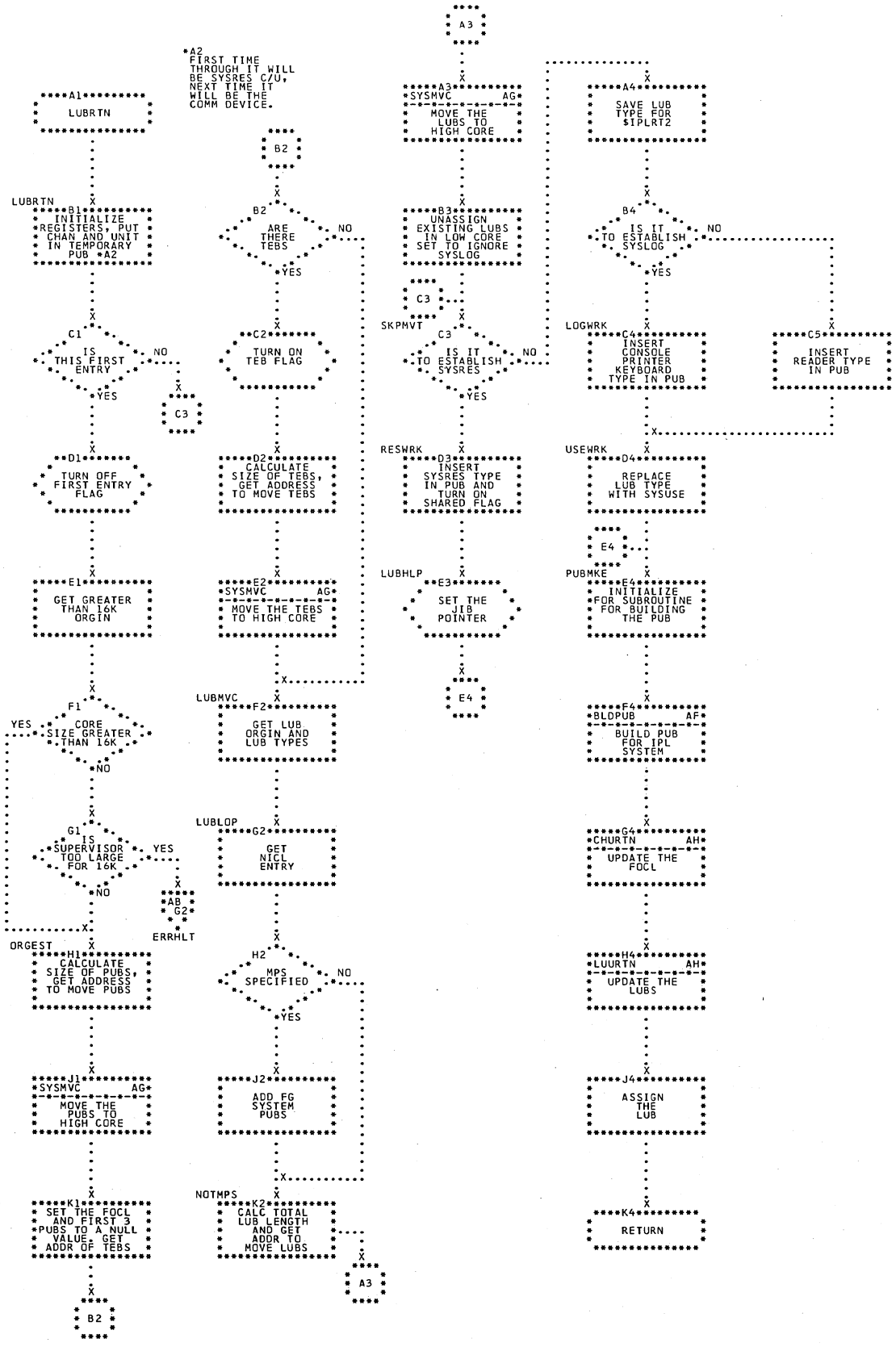


Chart AF. $$A$IPL2 - Build PUB Table
Refer to Chart 01.

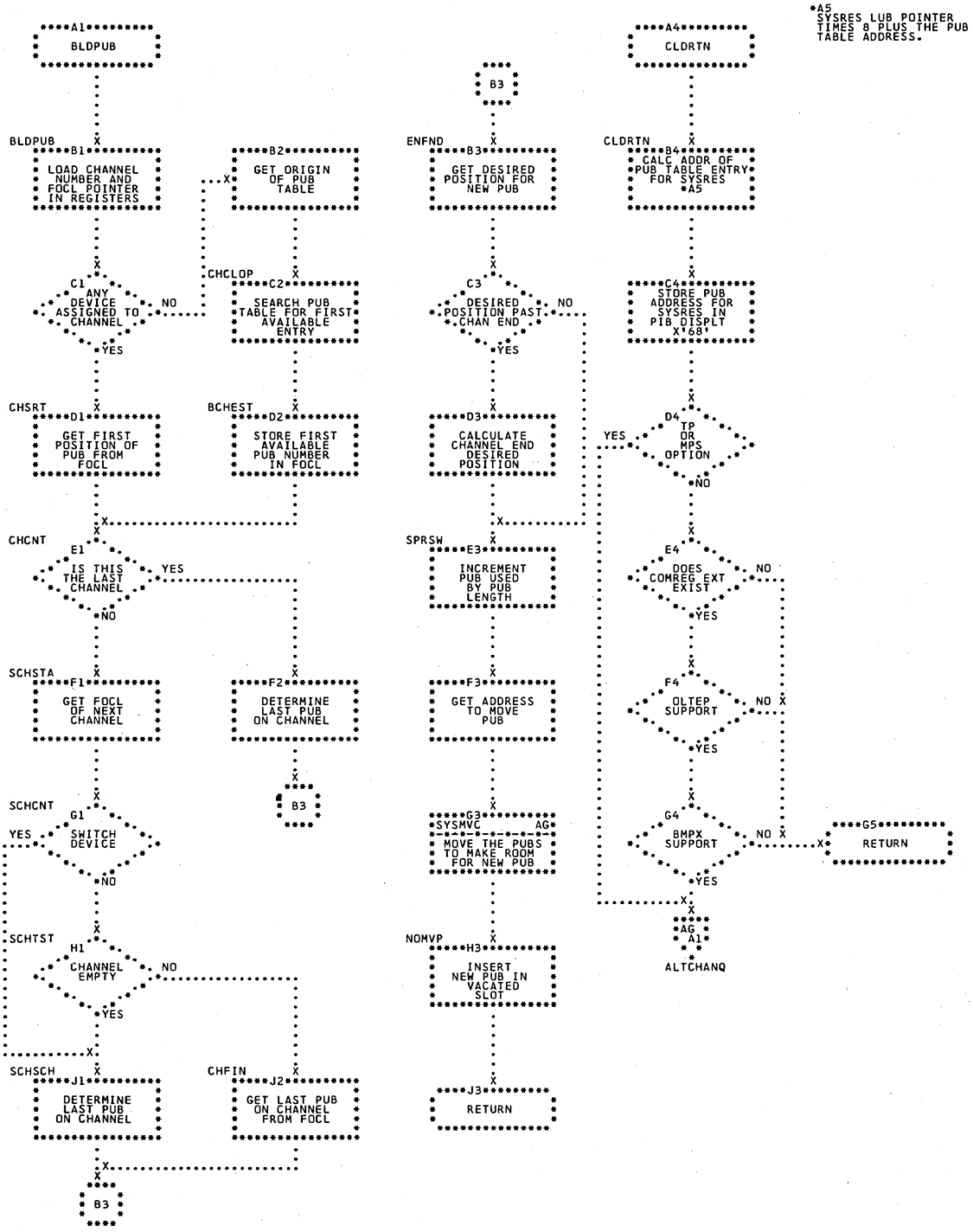


Chart AG. $$A$IPL2 - Common Move Subroutine
Refer to Chart 01.

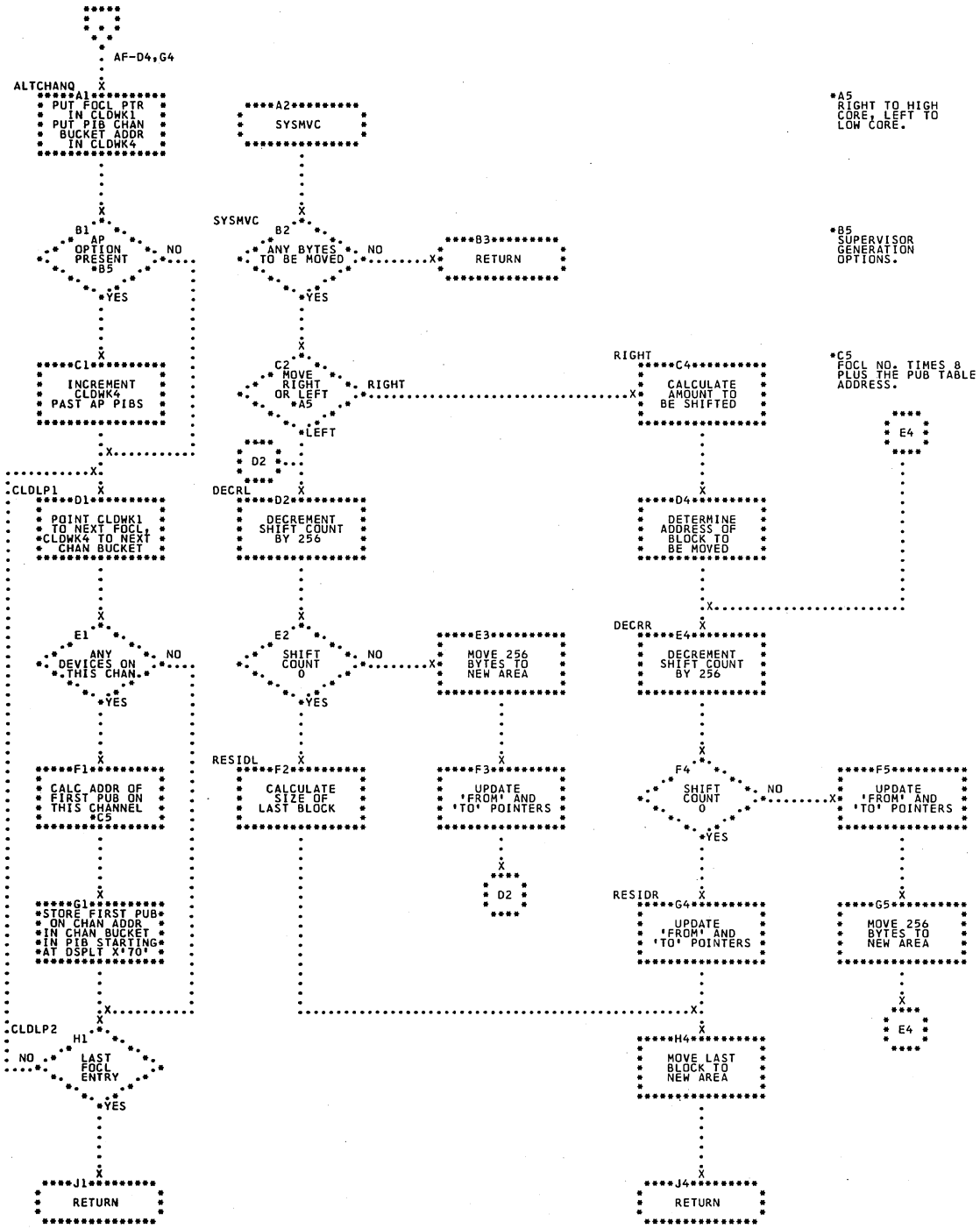


Chart AH. $$A$IPL2 - Update Subroutines
Refer to Chart 01.

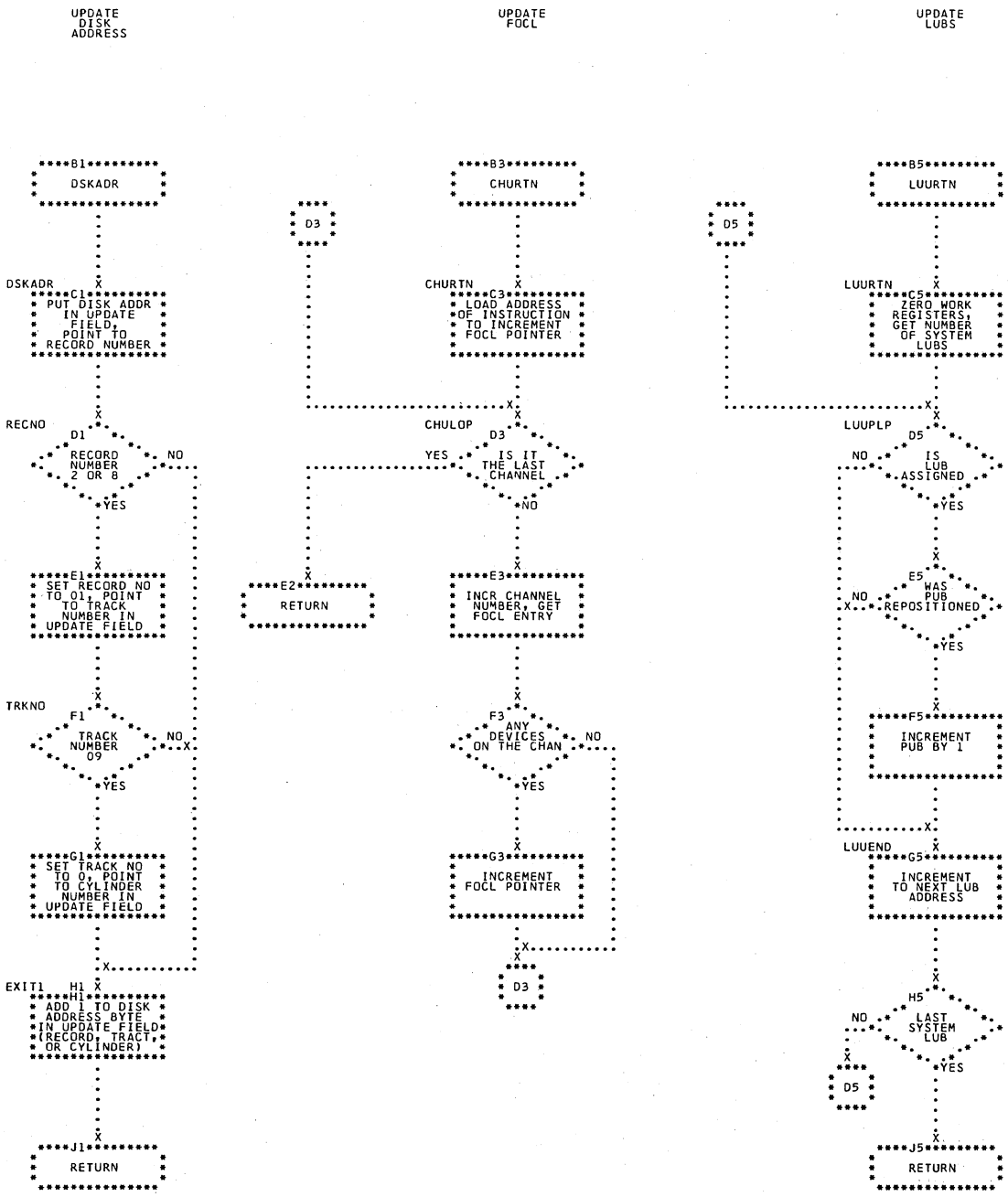


Chart AJ. $IPLRT2 - Initialization Routine
Refer to Chart 02.

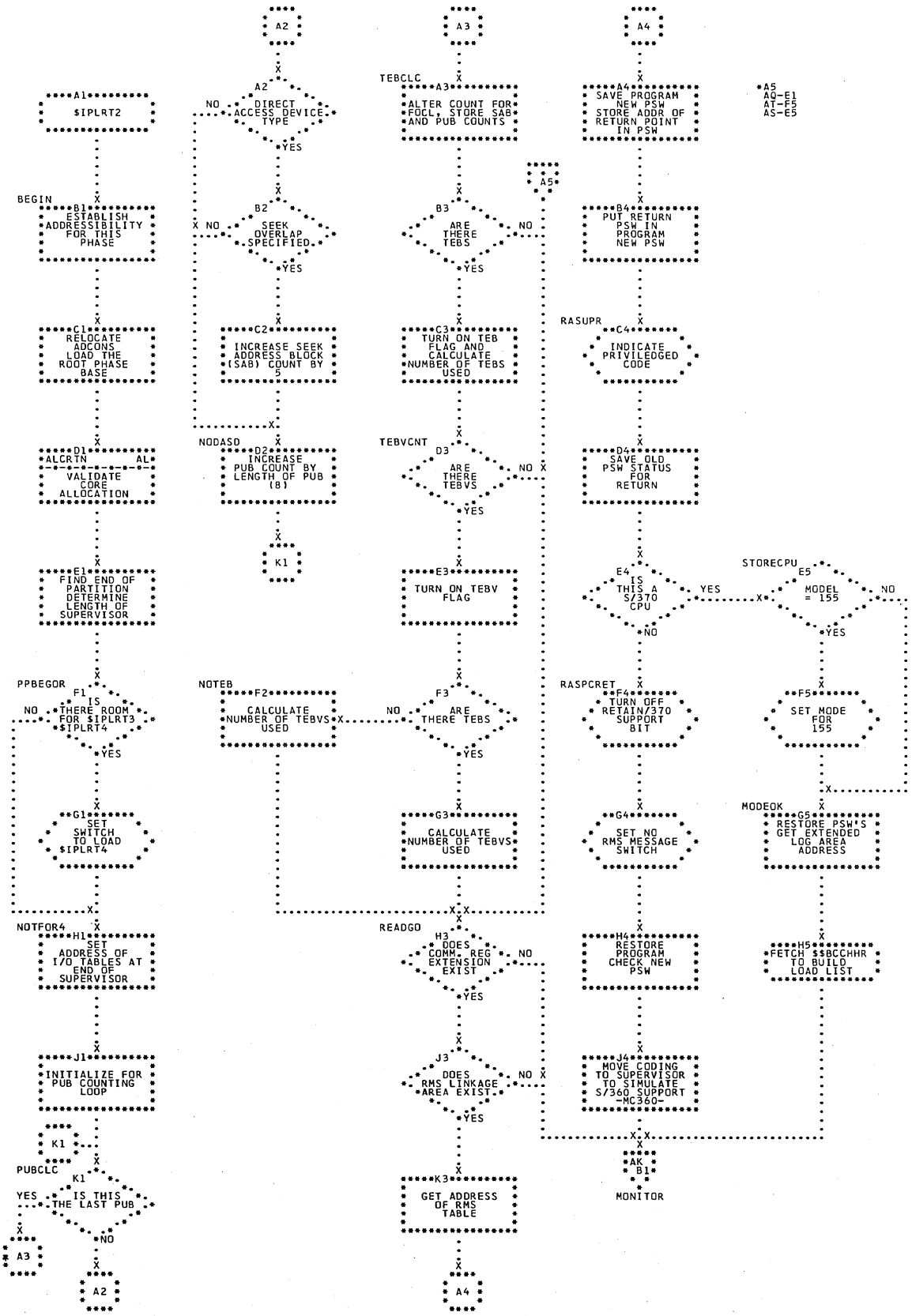


Chart AK. $IPLRT2 - Monitor, Read Control Card, and Operation Scan Routines
Refer to Chart 02.

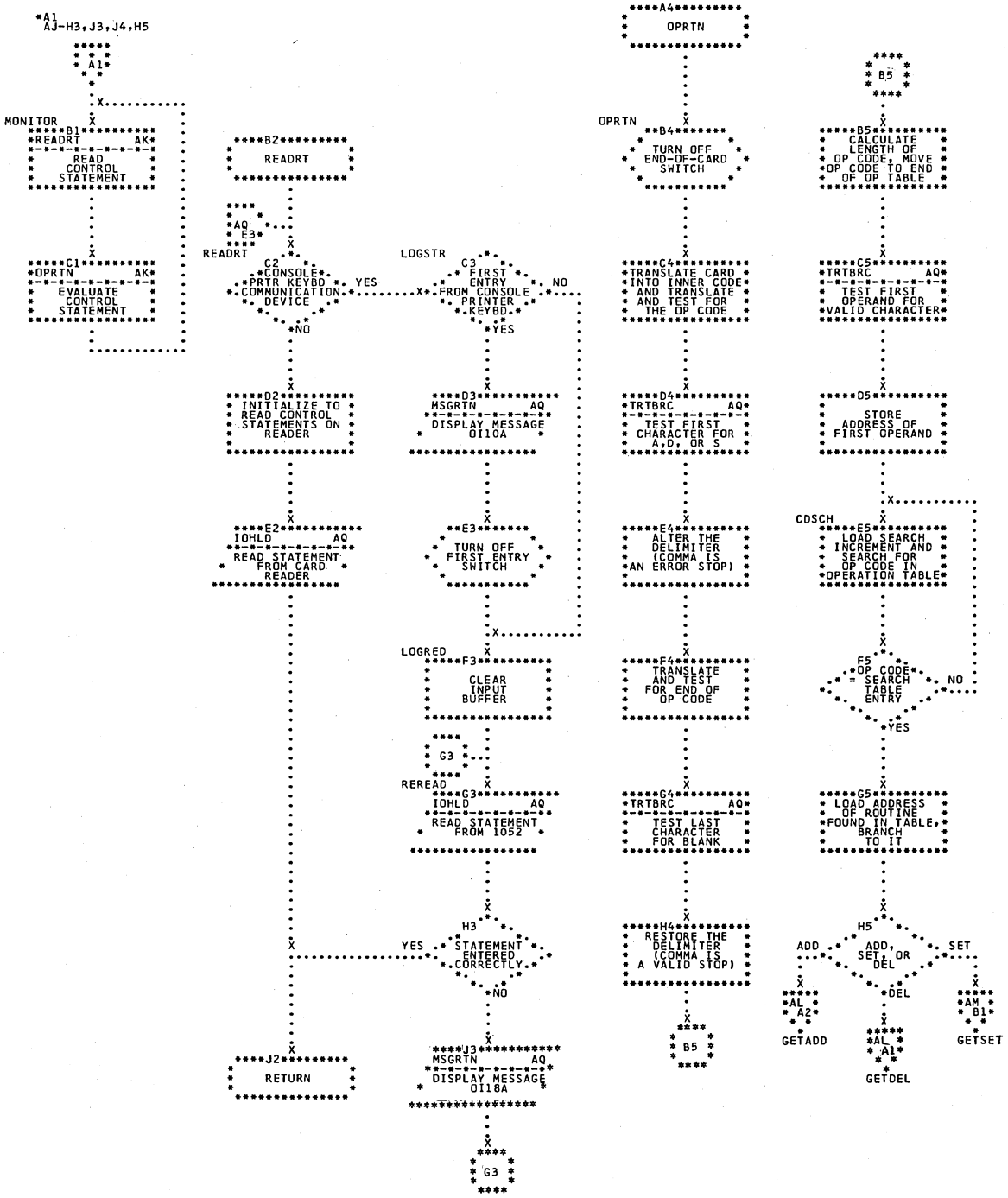


Chart AL. $IPLRT2 - Monitor Core Usage for ADD and DEL Cards and Allocation Subroutine
Refer to Chart 02.

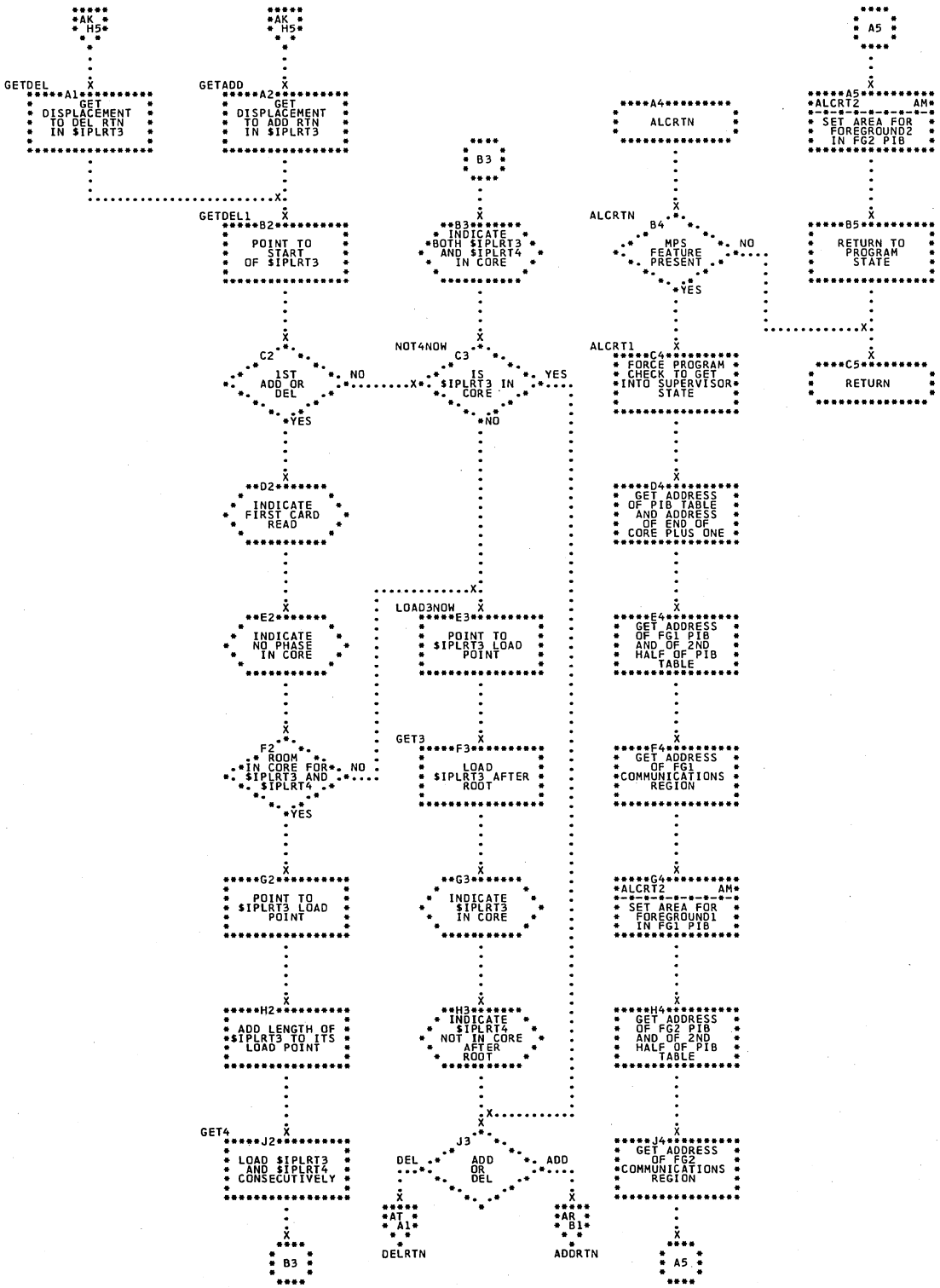


Chart AM. $IPLRT2 - Monitor Core Usage for SET Card and Allocation Subroutine
Refer to Chart 02.

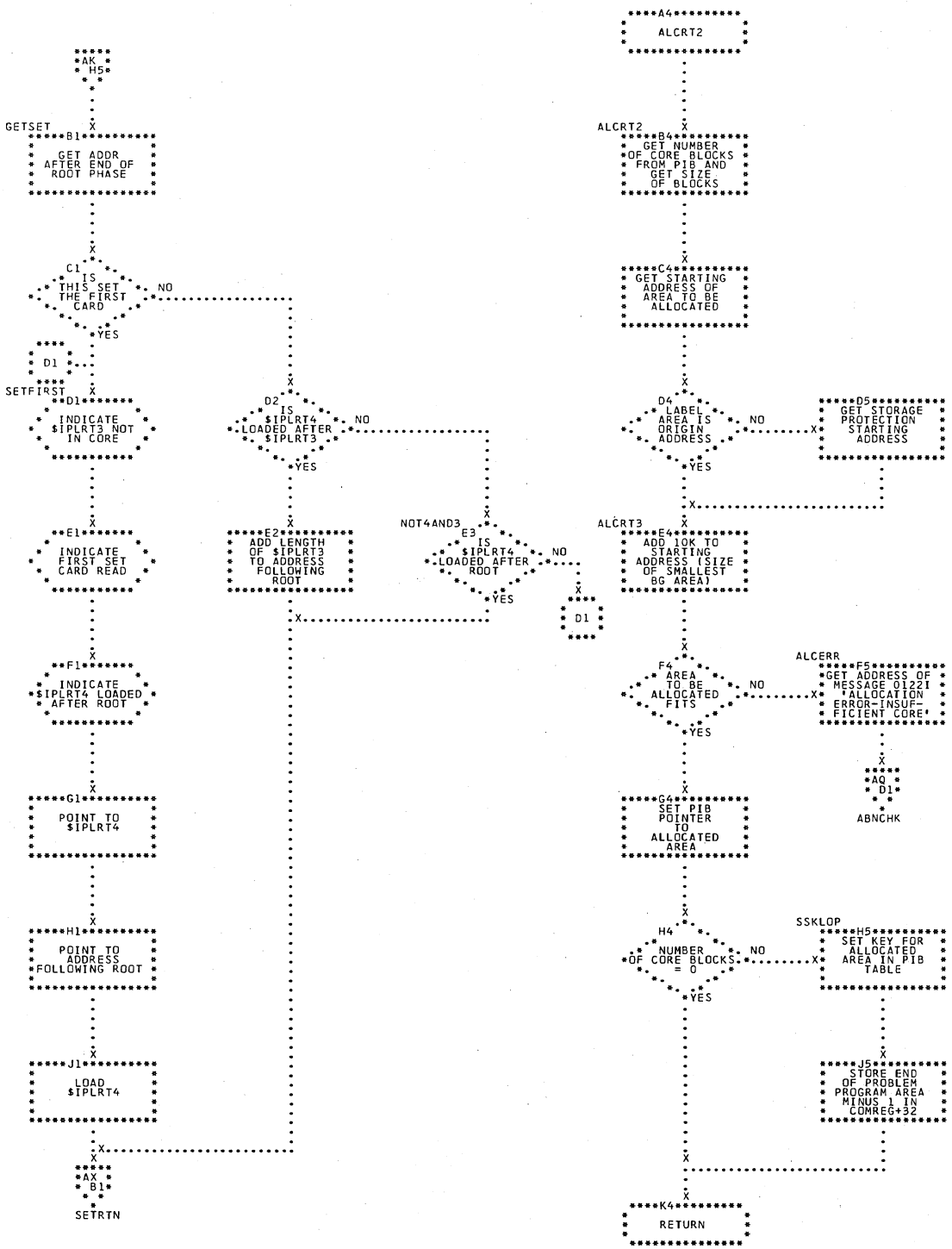


Chart AN. $IPLRT2 - Move Routine
Refer to Chart 02.

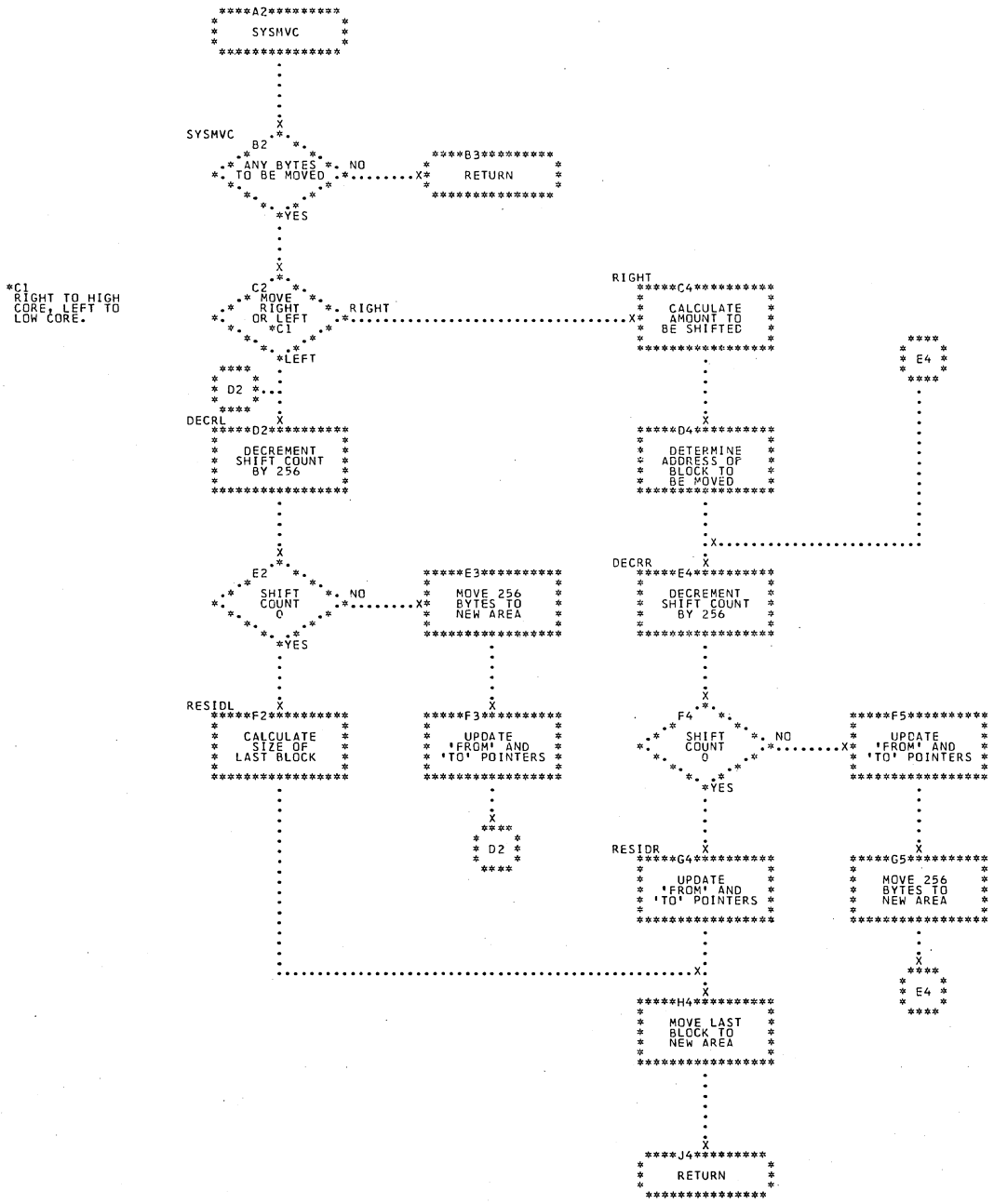


Chart AP. $IPLRT2 - Update LUB, Get Operand, and Conversion Subroutines
Refer to Chart 02.

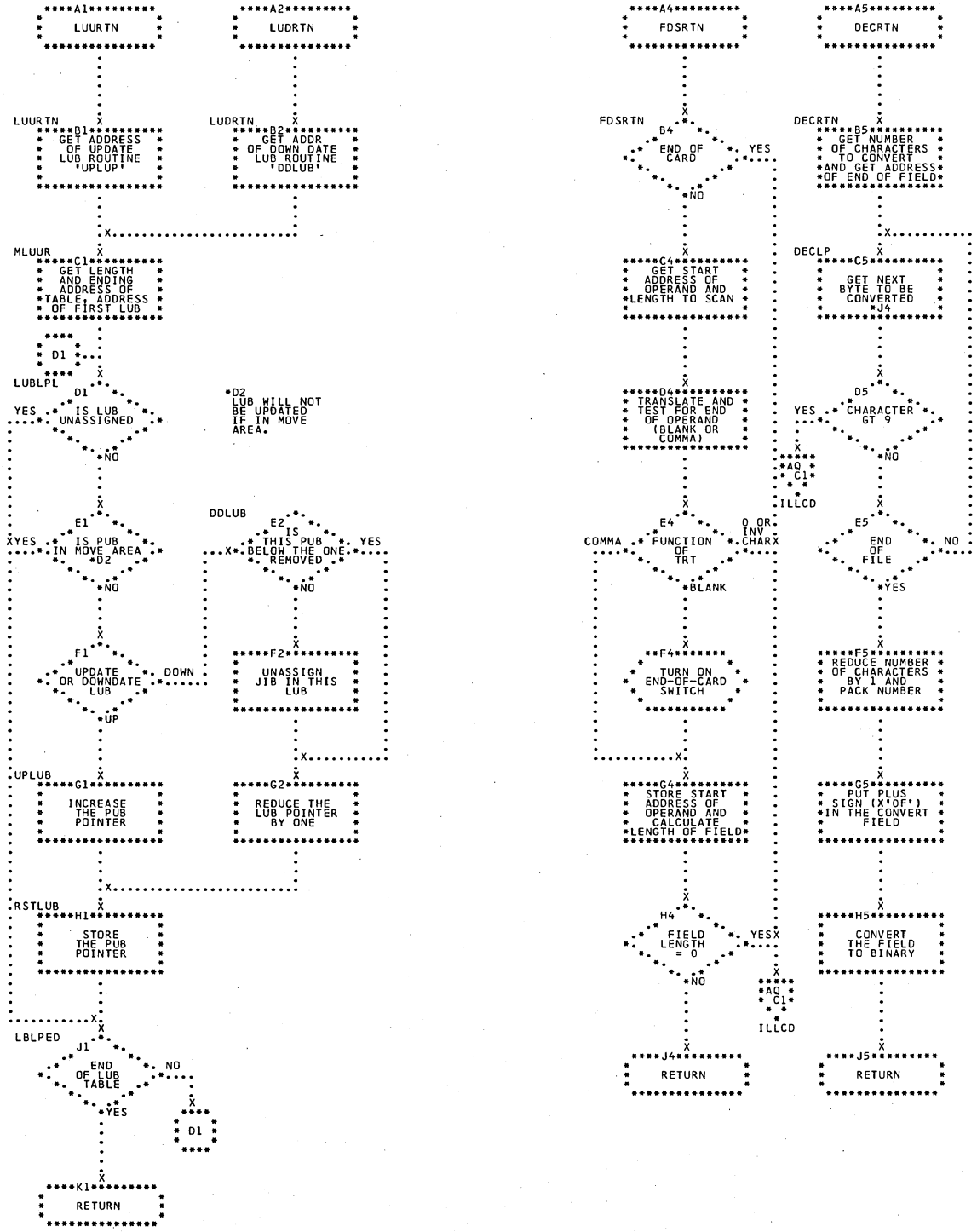


Chart AQ. $IPLRT2 - I/O Subroutines
Refer to Chart 02.

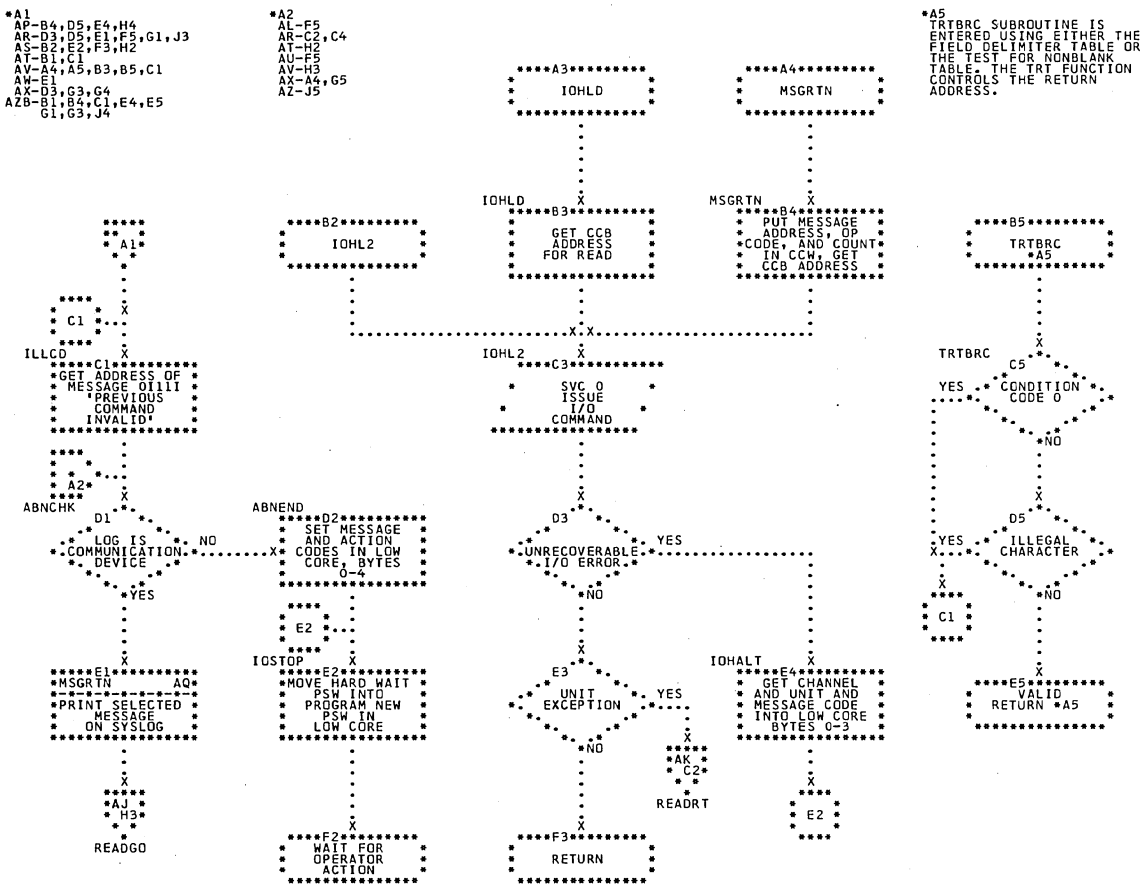


Chart AR. $IPLRT3 - ADD a Device (Part 1 of 2)
Refer to Chart 03.

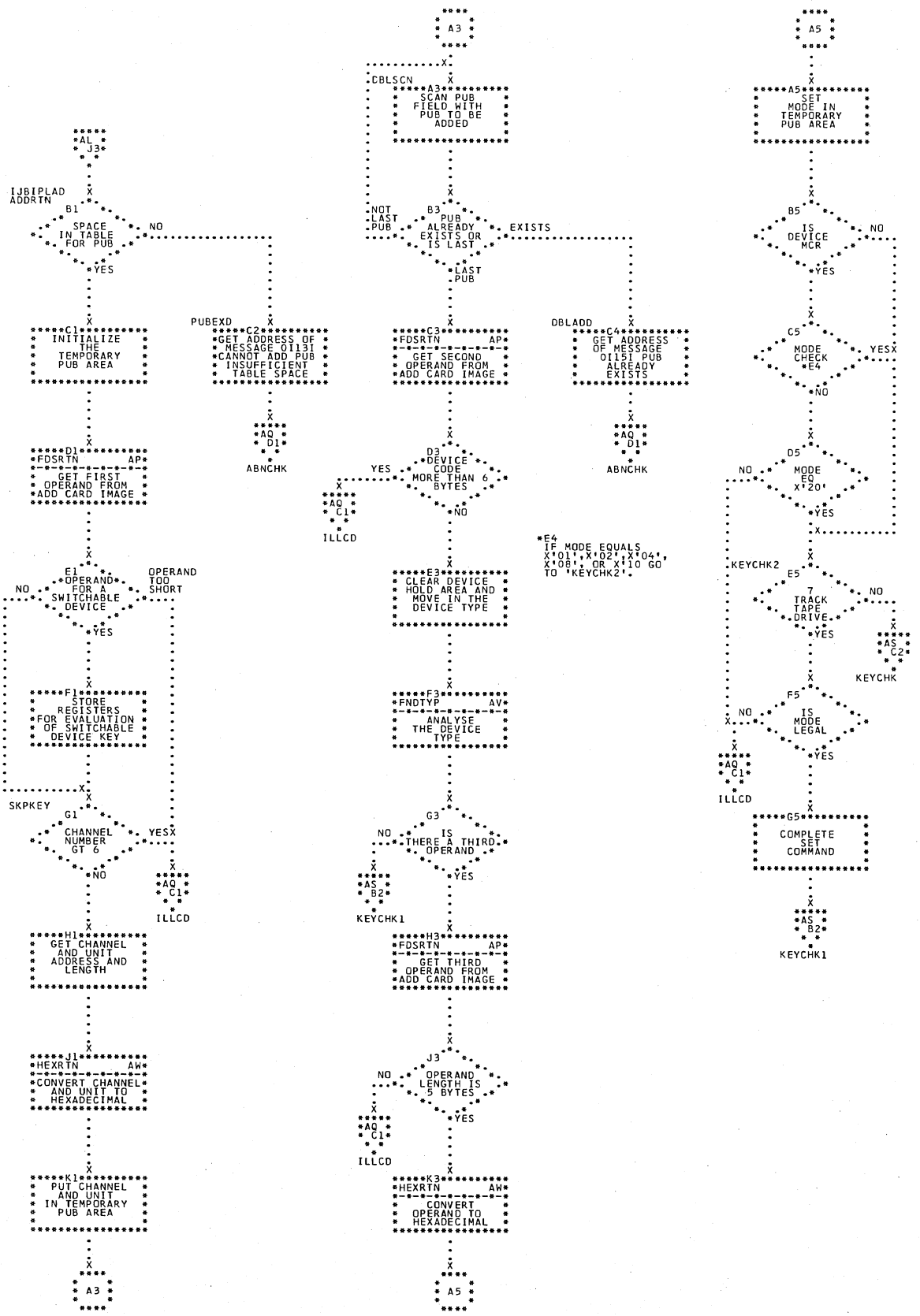


Chart AS. $IPLRT3 - ADD a Device (Part 2 of 2)
Refer to Chart 03.

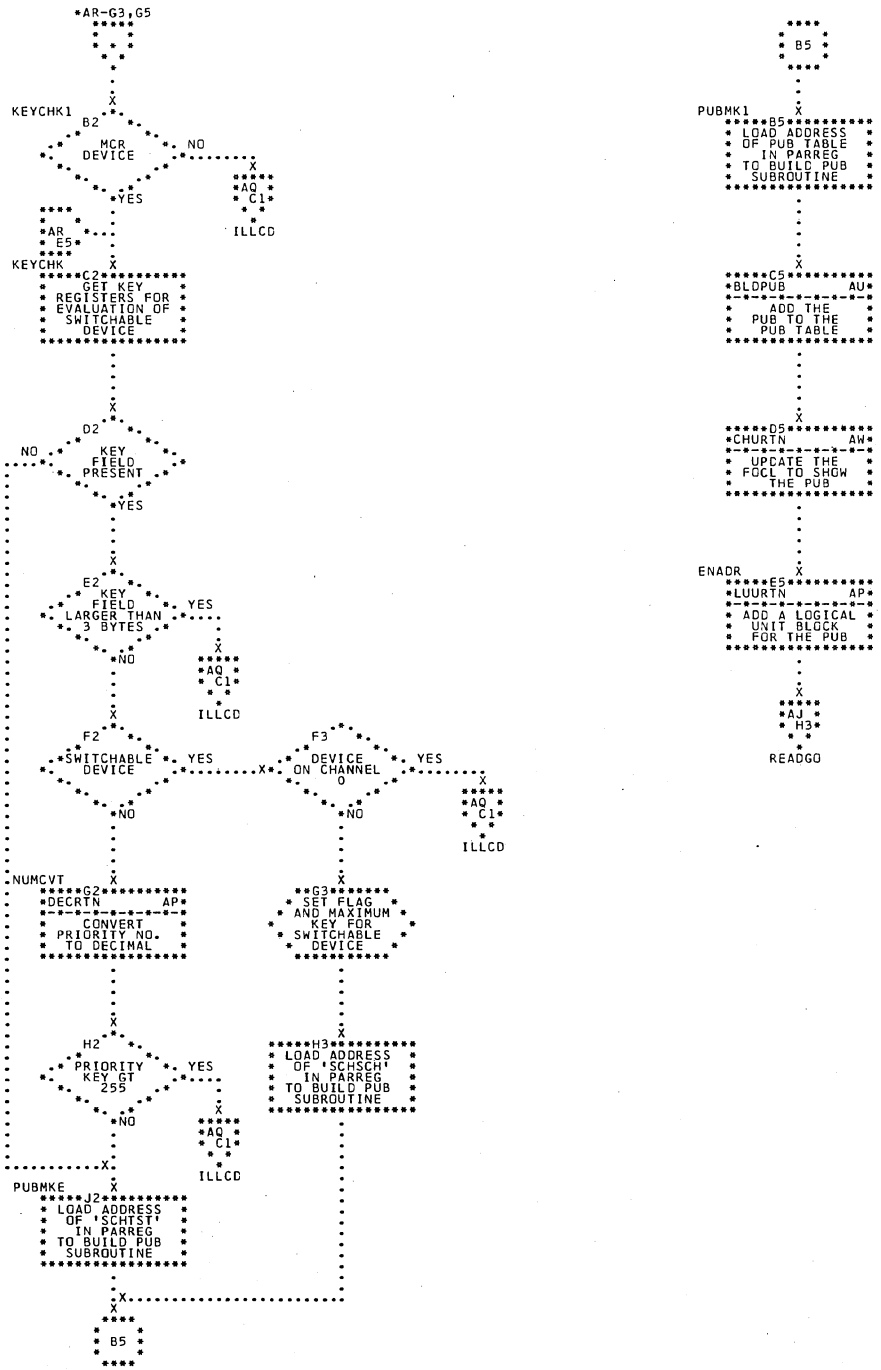


Chart AT. $IPLRT3 - Delete a PUB
Refer to Chart 03.

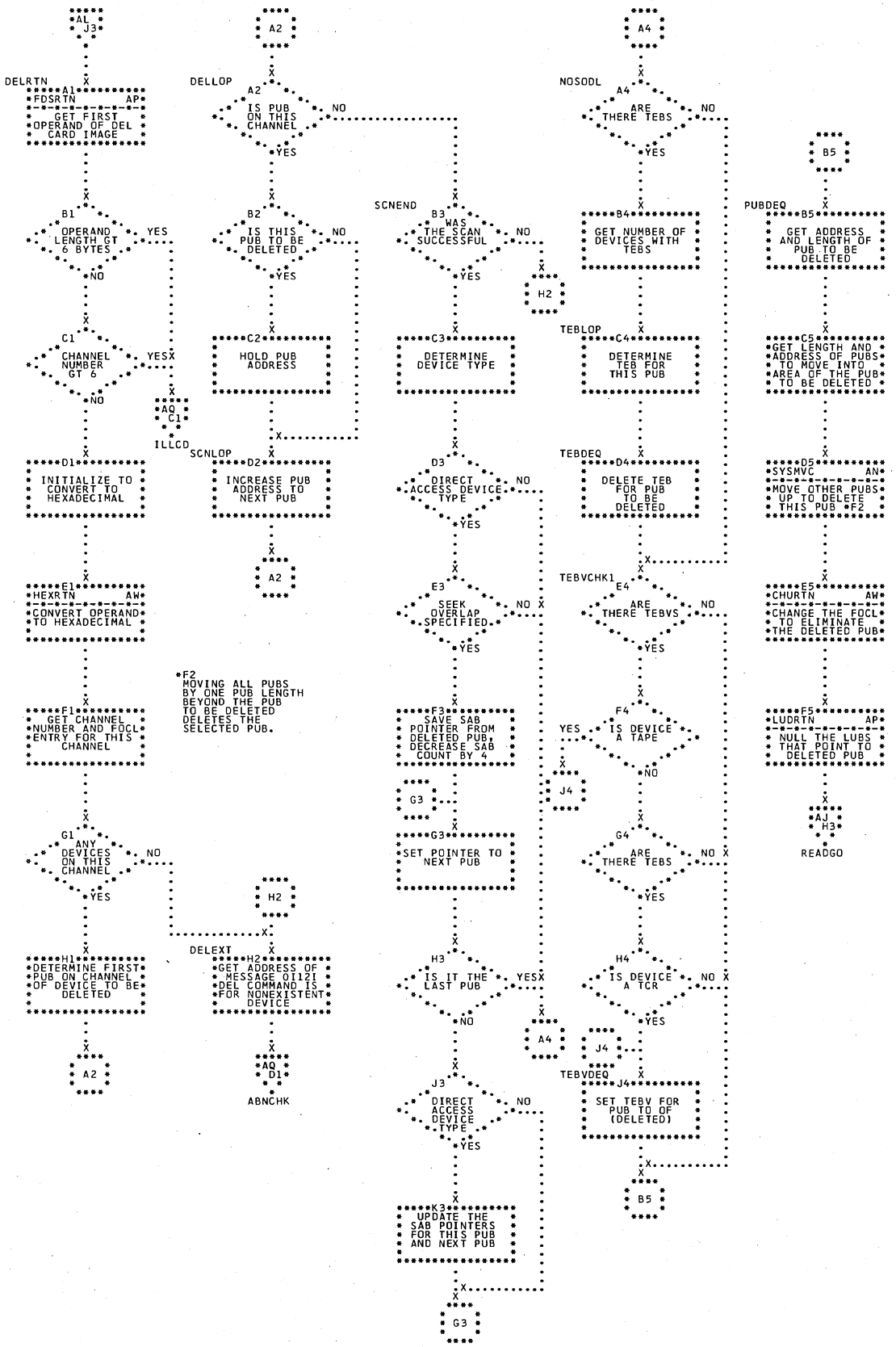


Chart AU. $IPLRT3 - Build PUB Table Subroutine
Refer to Chart 03.

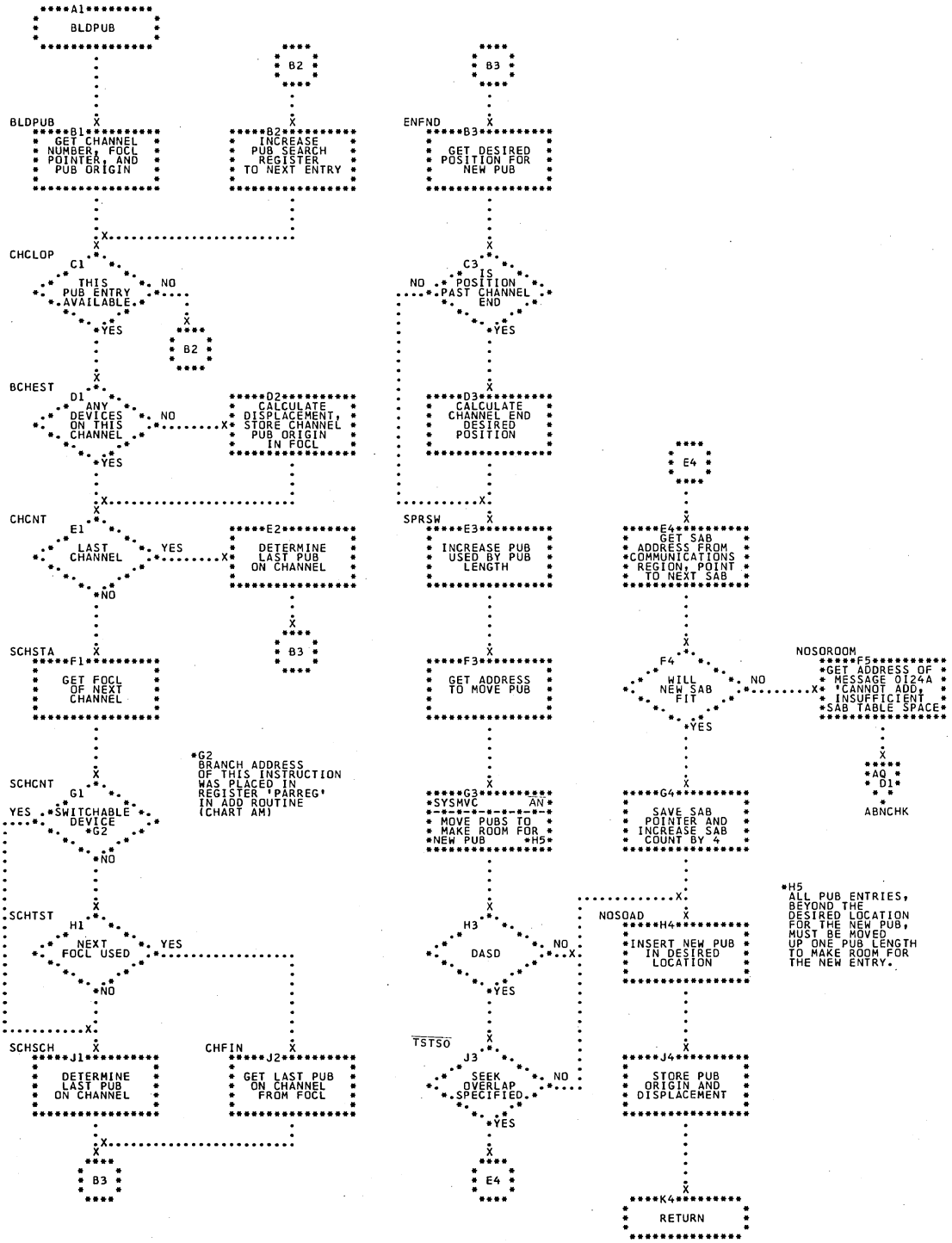


Chart AV. $IPLRT3 - Device Type Conversion Subroutine
Refer to Chart 03.

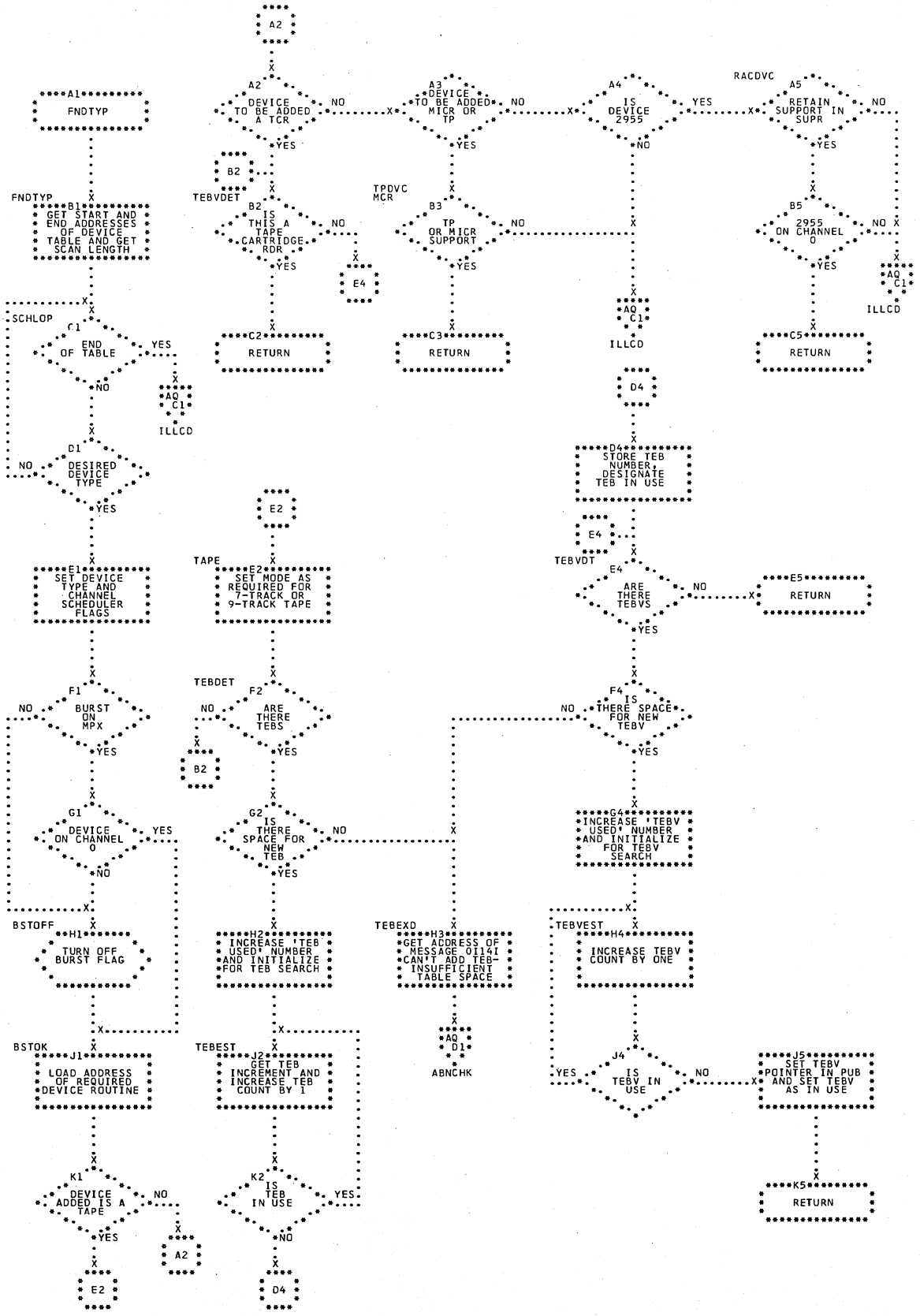


Chart AW. $IPLRT3 - Conversion and Update FOCL Subroutines
Refer to Chart 03.

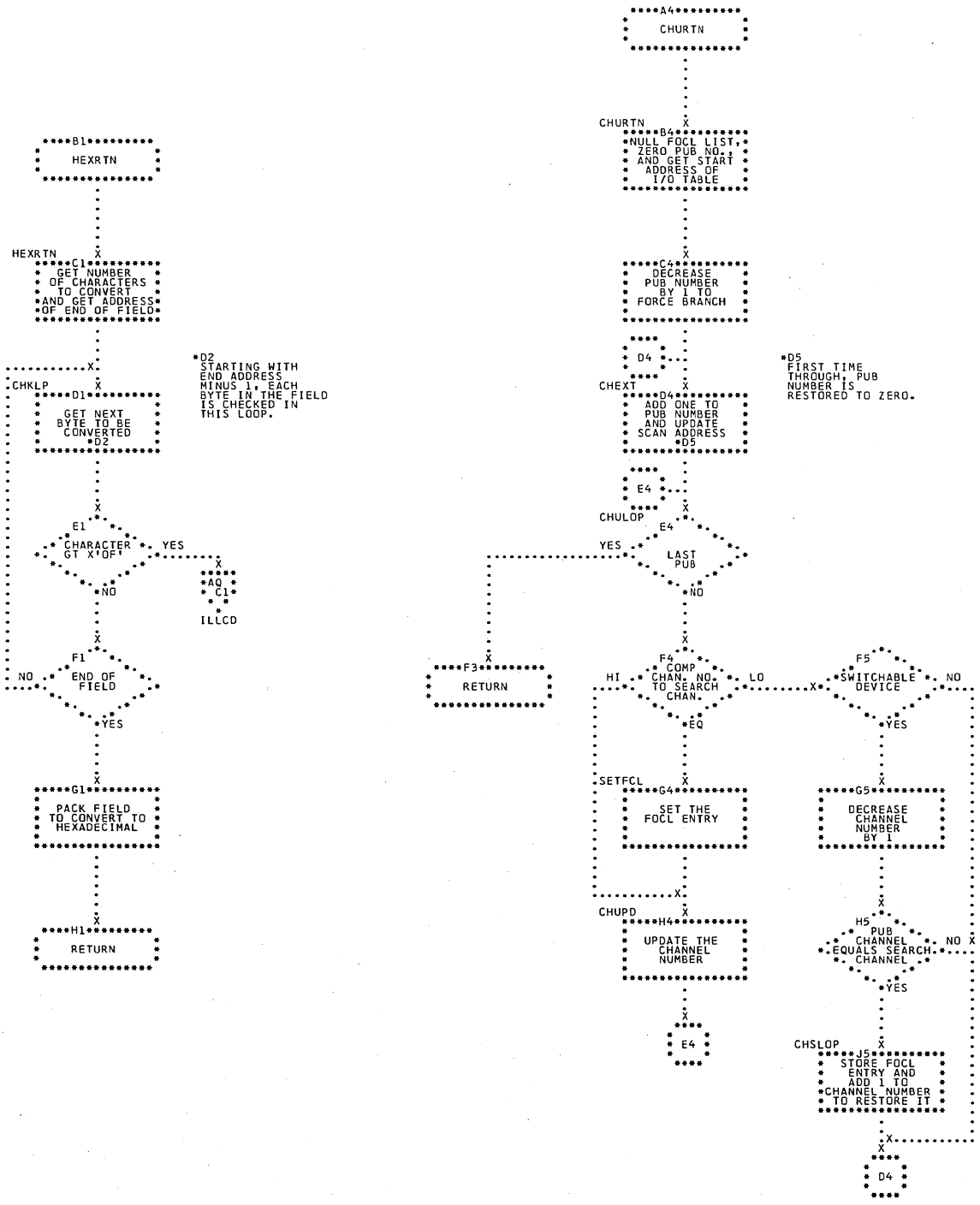


Chart AX. $IPLRT4 - SET Statement Processor and Assign SYSLOG
Refer to Chart 03.

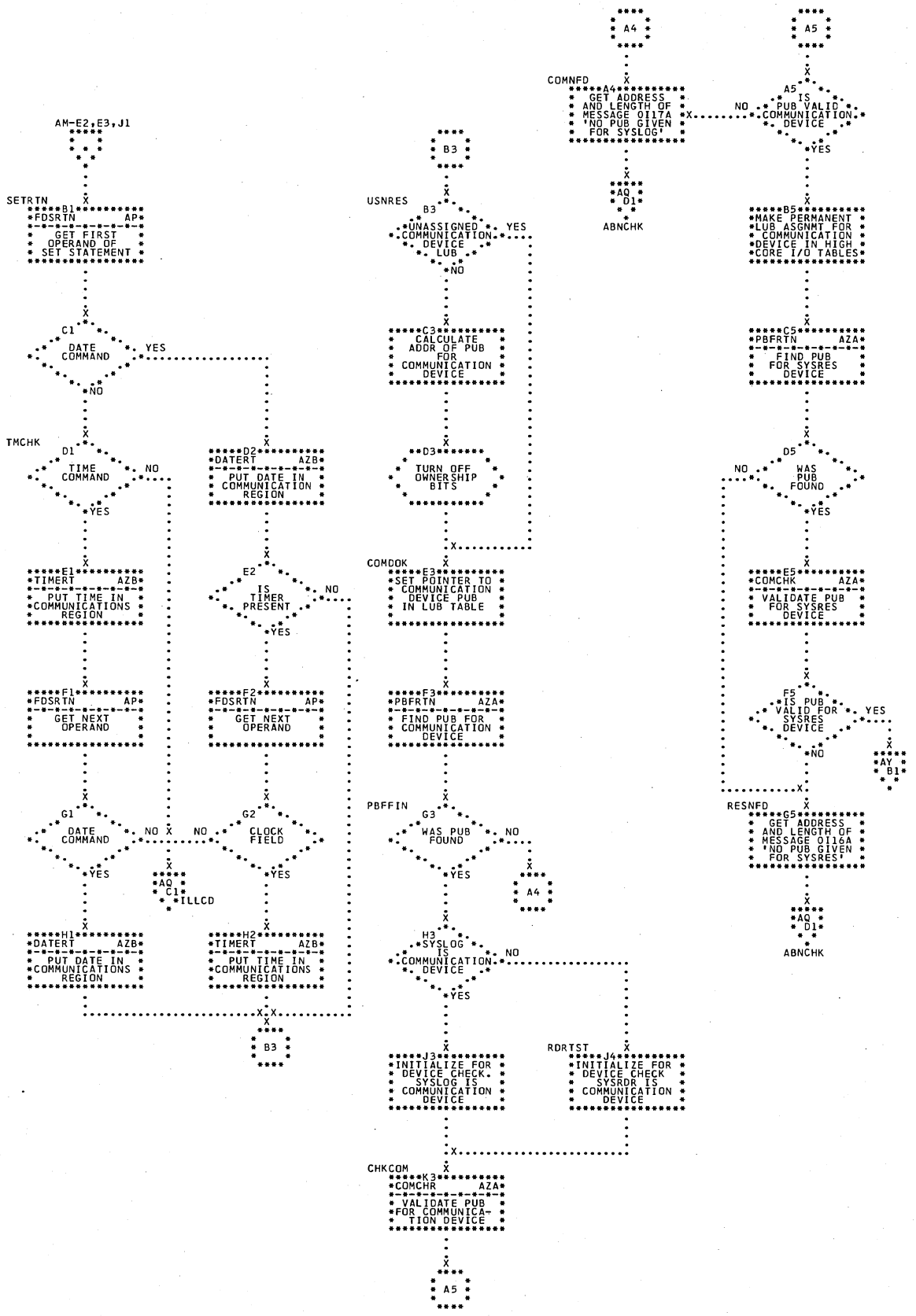


Chart AY. $IPLRT4 - Assign SYSRES and Move I/O Tables
Refer to Chart 03.

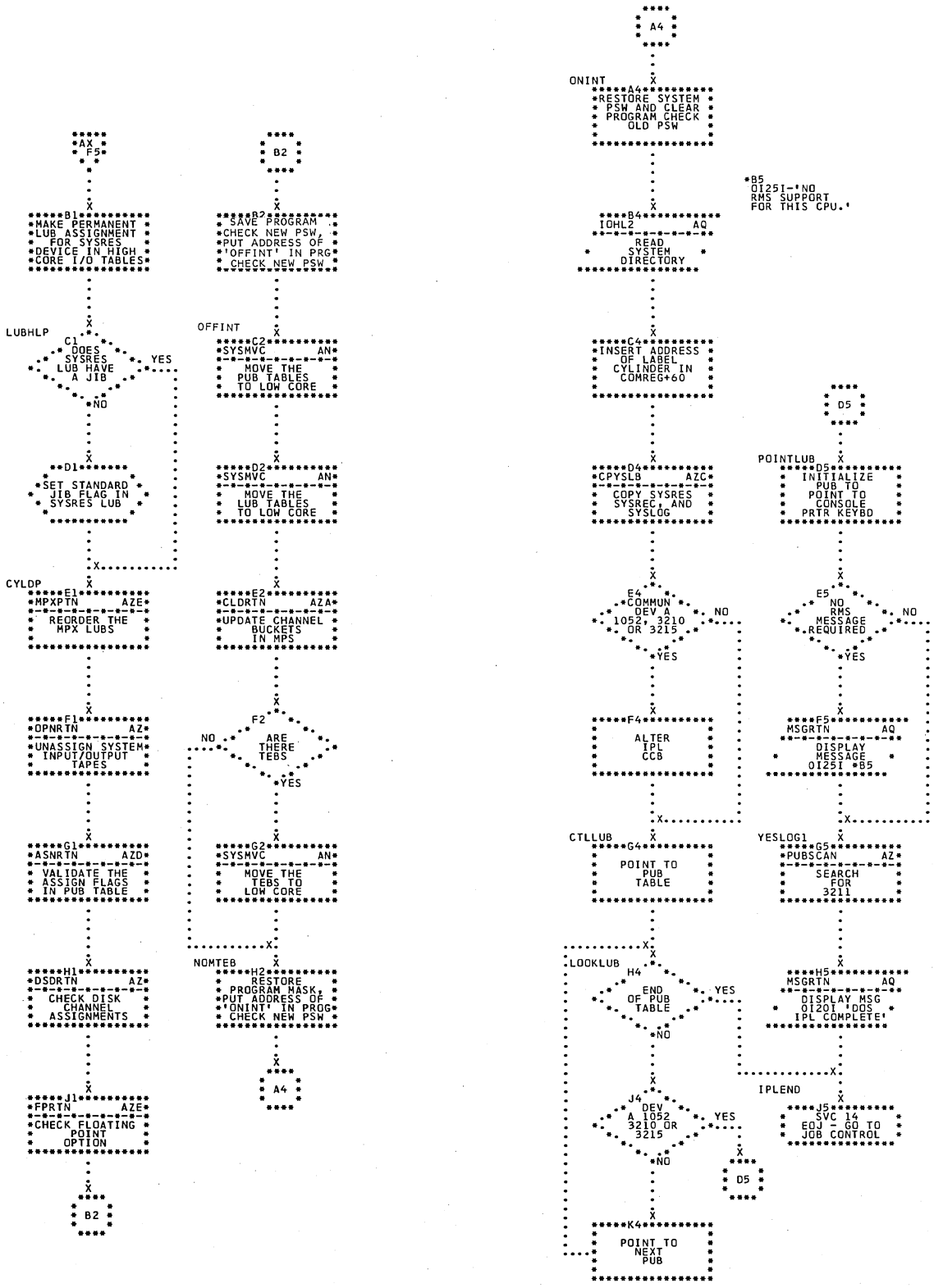


Chart AZ. $IPLRT4 - I/O and Check Device Type Subroutines
Refer to Chart 03.

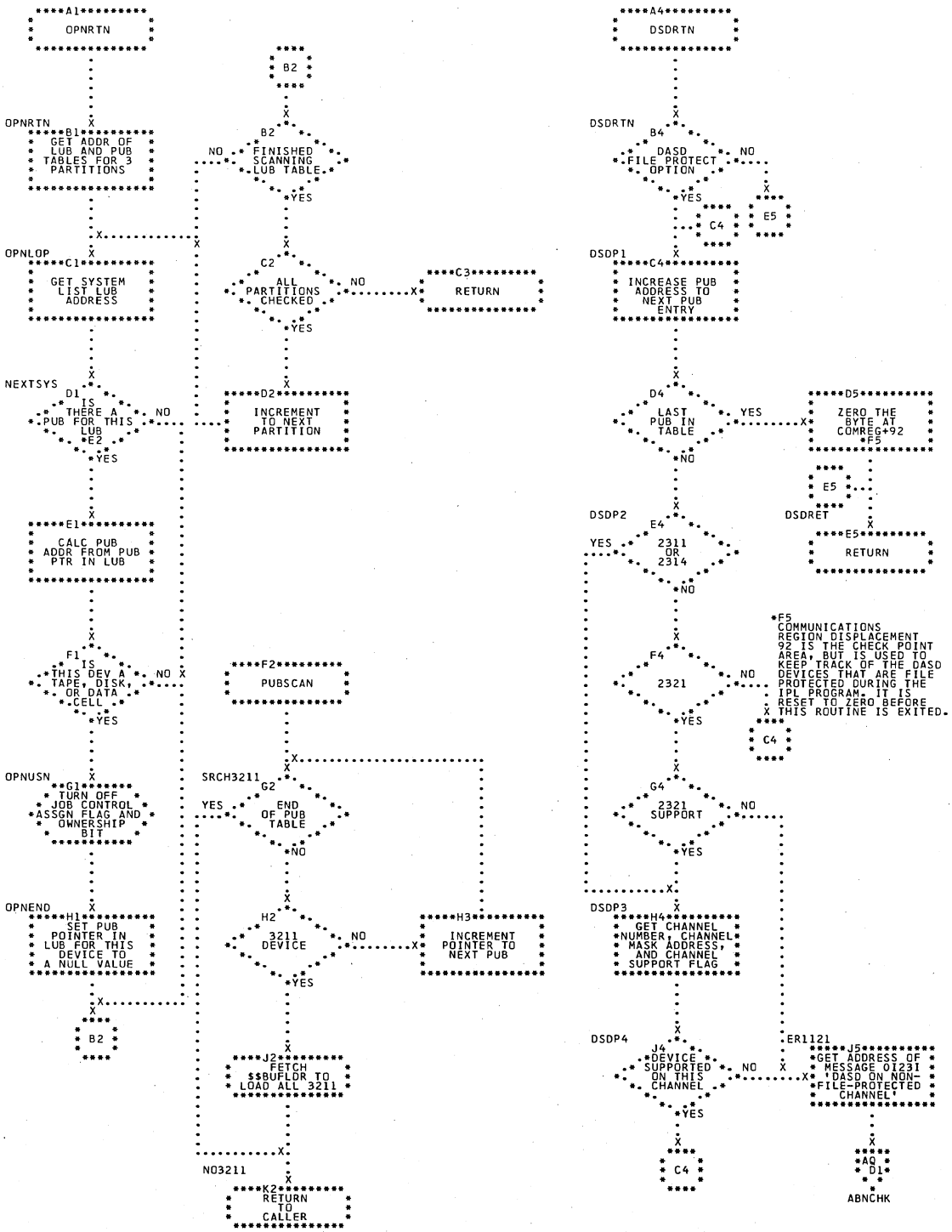


Chart AZA. $IPLRT4 - Find PUB and I/O Subroutines
Refer to Chart 03.

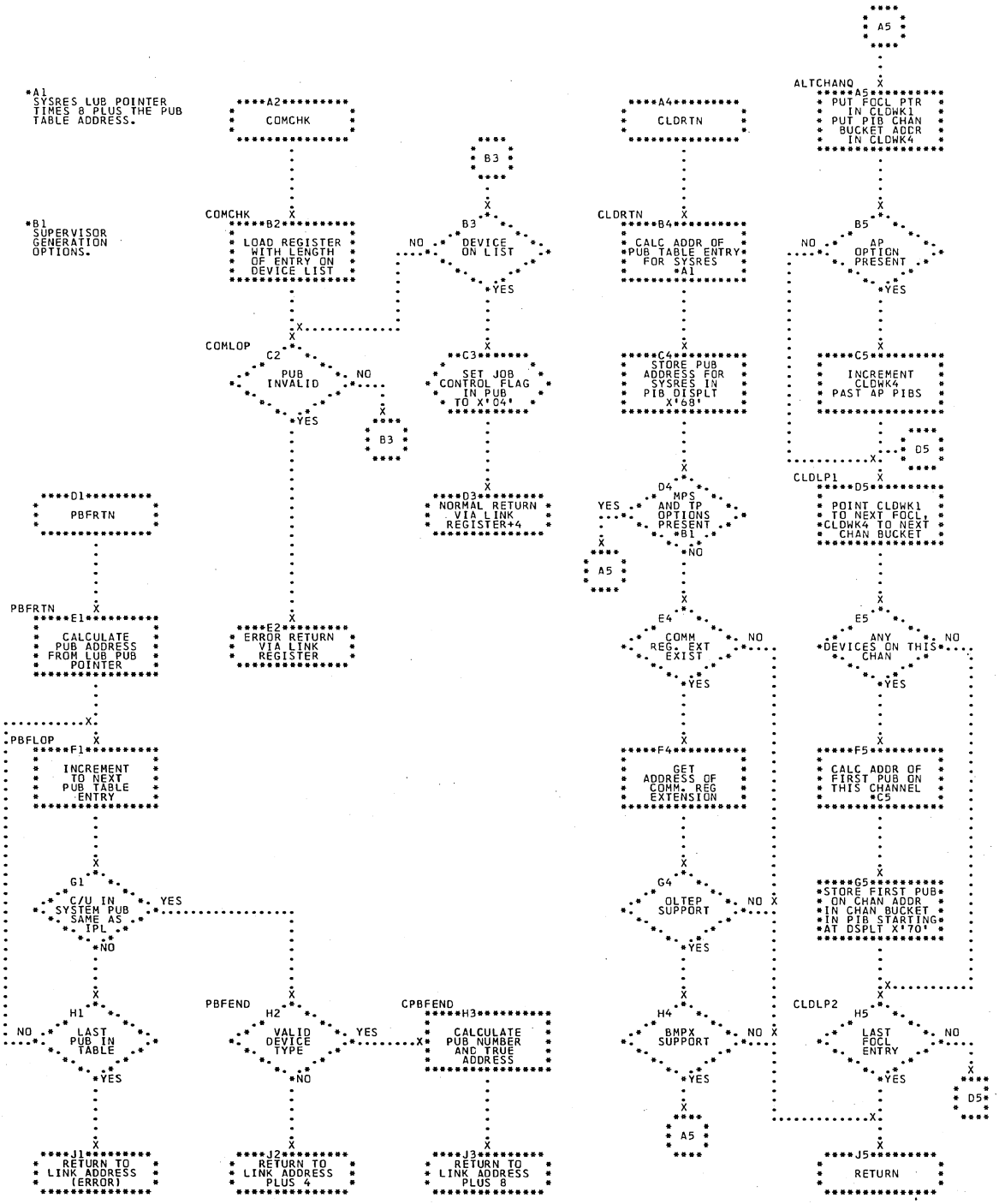
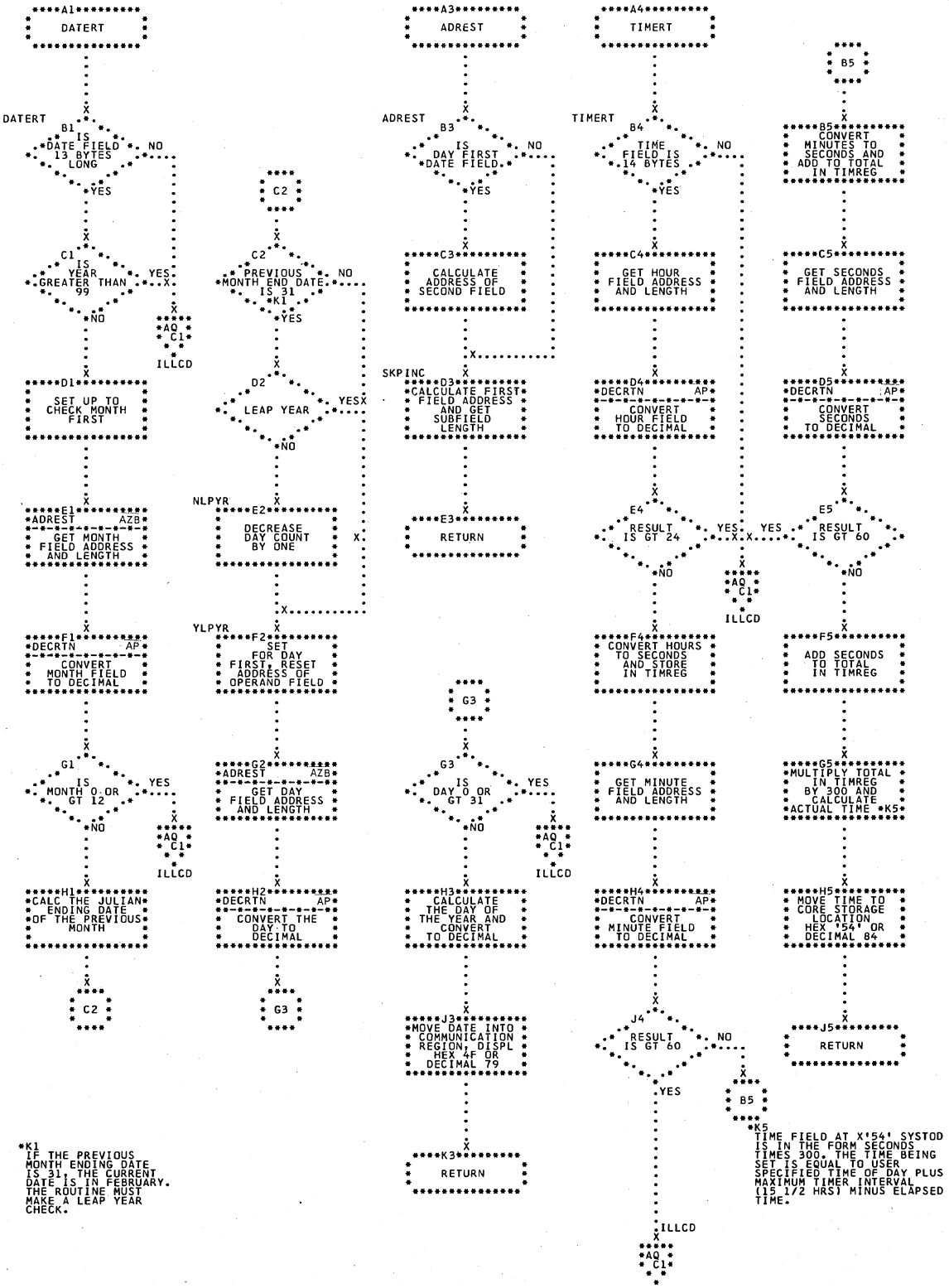


Chart AZB. $IPLRT4 - Date and Time Subroutines
Refer to Chart 03.



*K1
IF THE PREVIOUS MONTH ENDING DATE IS 31, THE CURRENT DATE IS IN FEBRUARY. THE ROUTINE MUST MAKE A LEAP YEAR CHECK.

*K5
TIME FIELD AT X'54' SYSTOD IS IN THE FORM SECONDS TIMES 300. THE TIME BEING SET IS EQUAL TO USER SPECIFIED TIME OF DAY PLUS MAXIMUM TIMER INTERVAL (15 1/2 HRS) MINUS ELAPSED TIME.

Chart AZC. $IPLRT4 - Copy Subroutine
Refer to Chart 03.

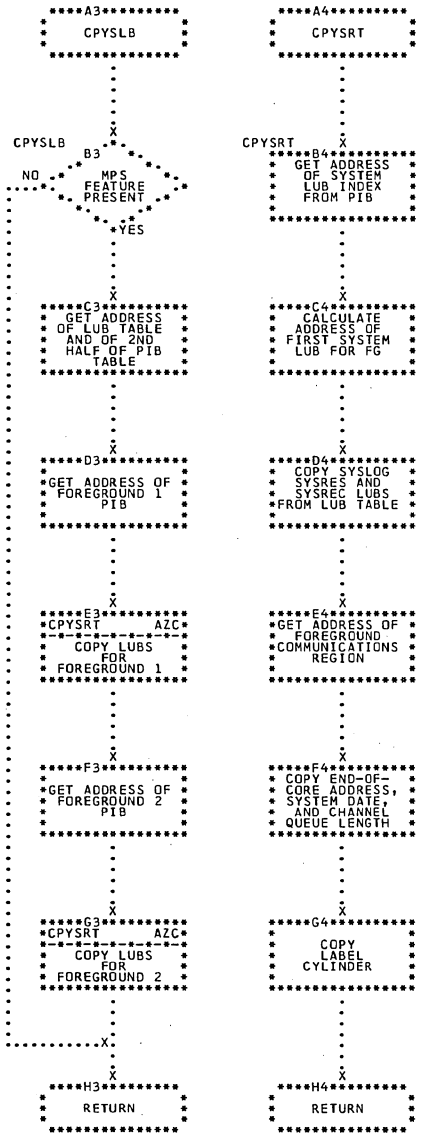


Chart AZD. $IPLRT4 - Set Job Control Flags Subroutine
Refer to Chart 03.

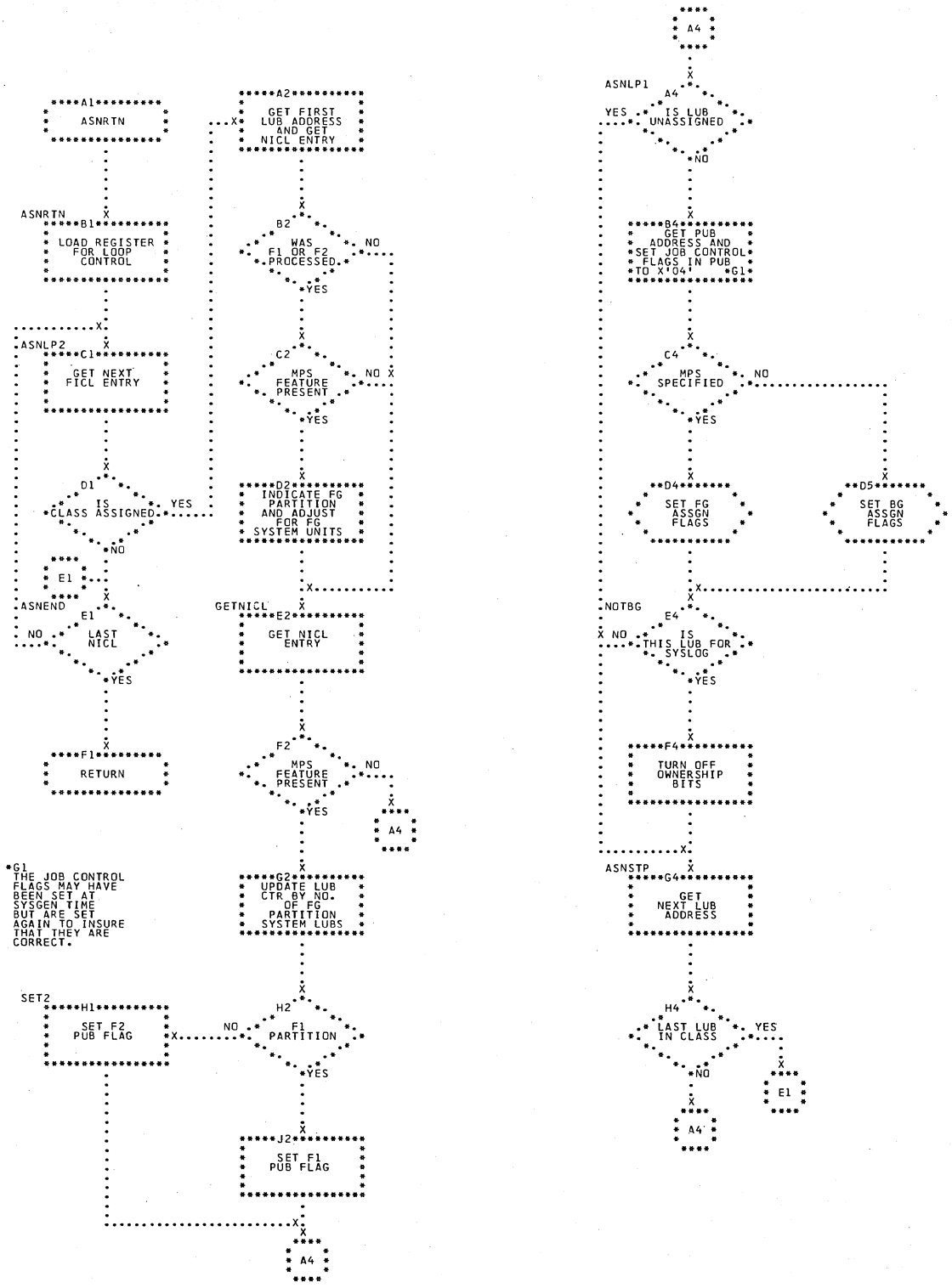


Chart AZE. $IPLRT4 - Reorder MPX Channel LUBs and PUBs
Refer to Chart 03.

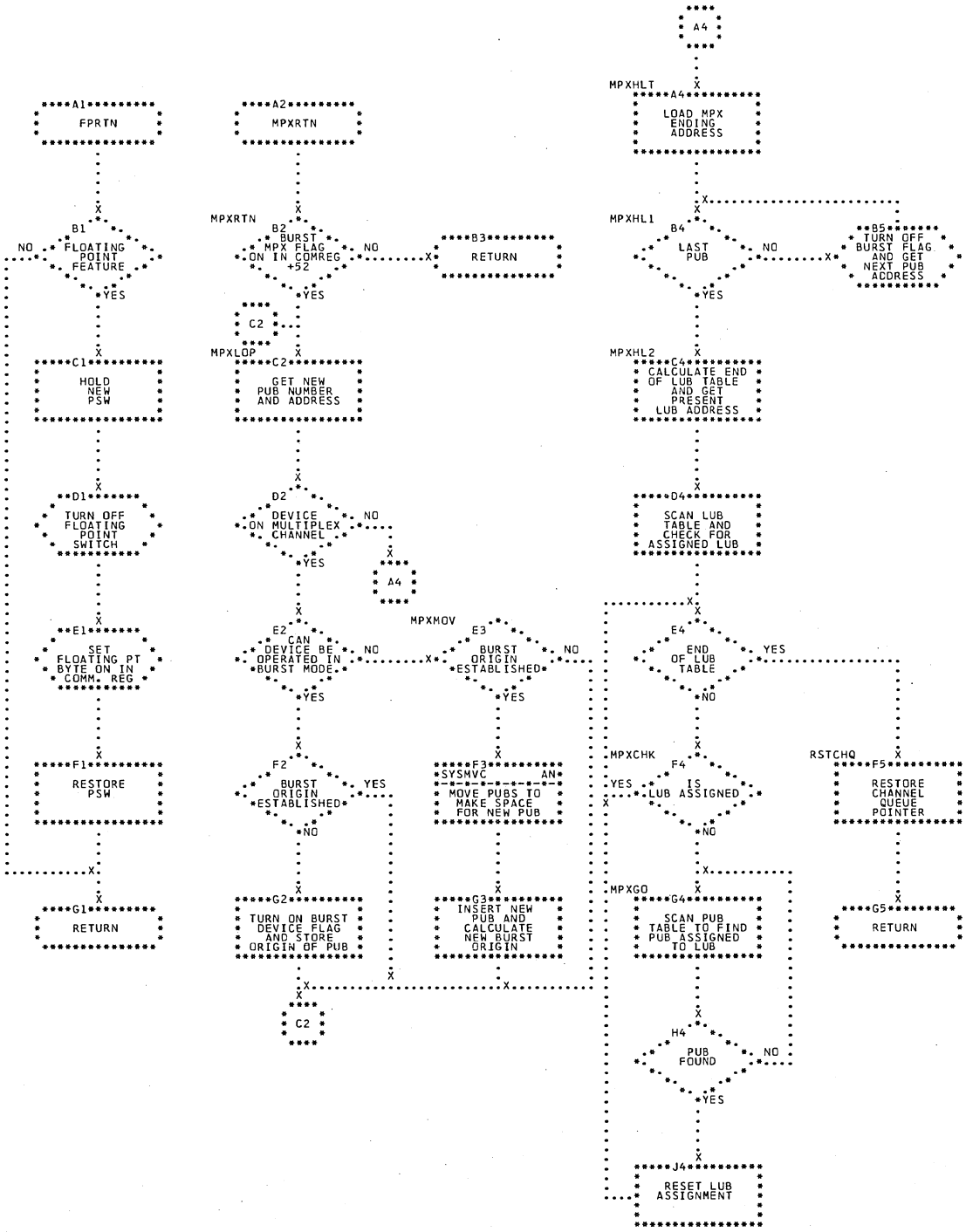


Chart BA. $JOBCTLA - Initialization
Refer to Chart 04.

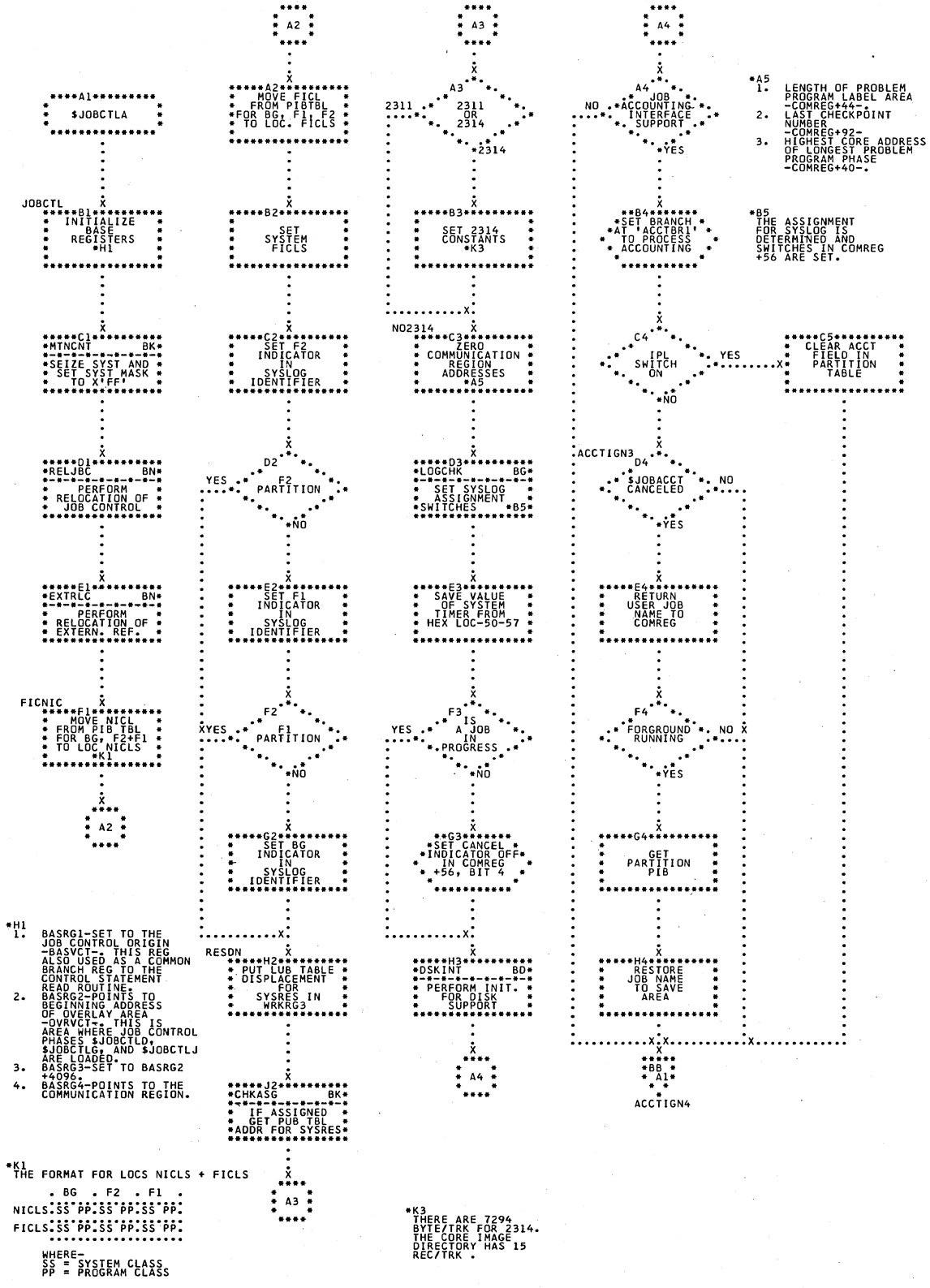


Chart BB. $JOBCTLA - Initialization and Control Statement Read
Refer to Chart 04.

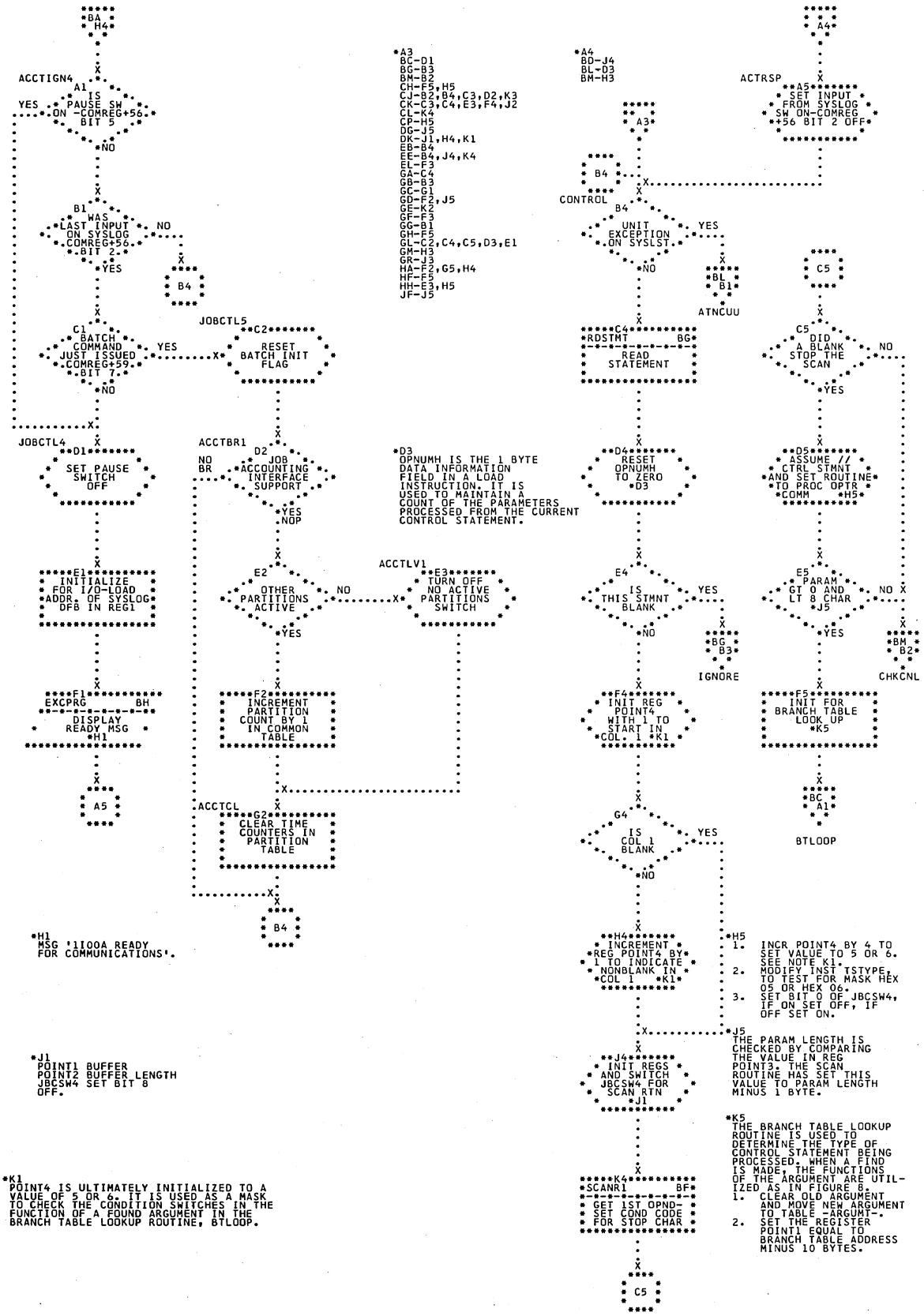


Chart BC. $JOBCTLA - Phase Vector Table Lookup
Refer to Chart 04.

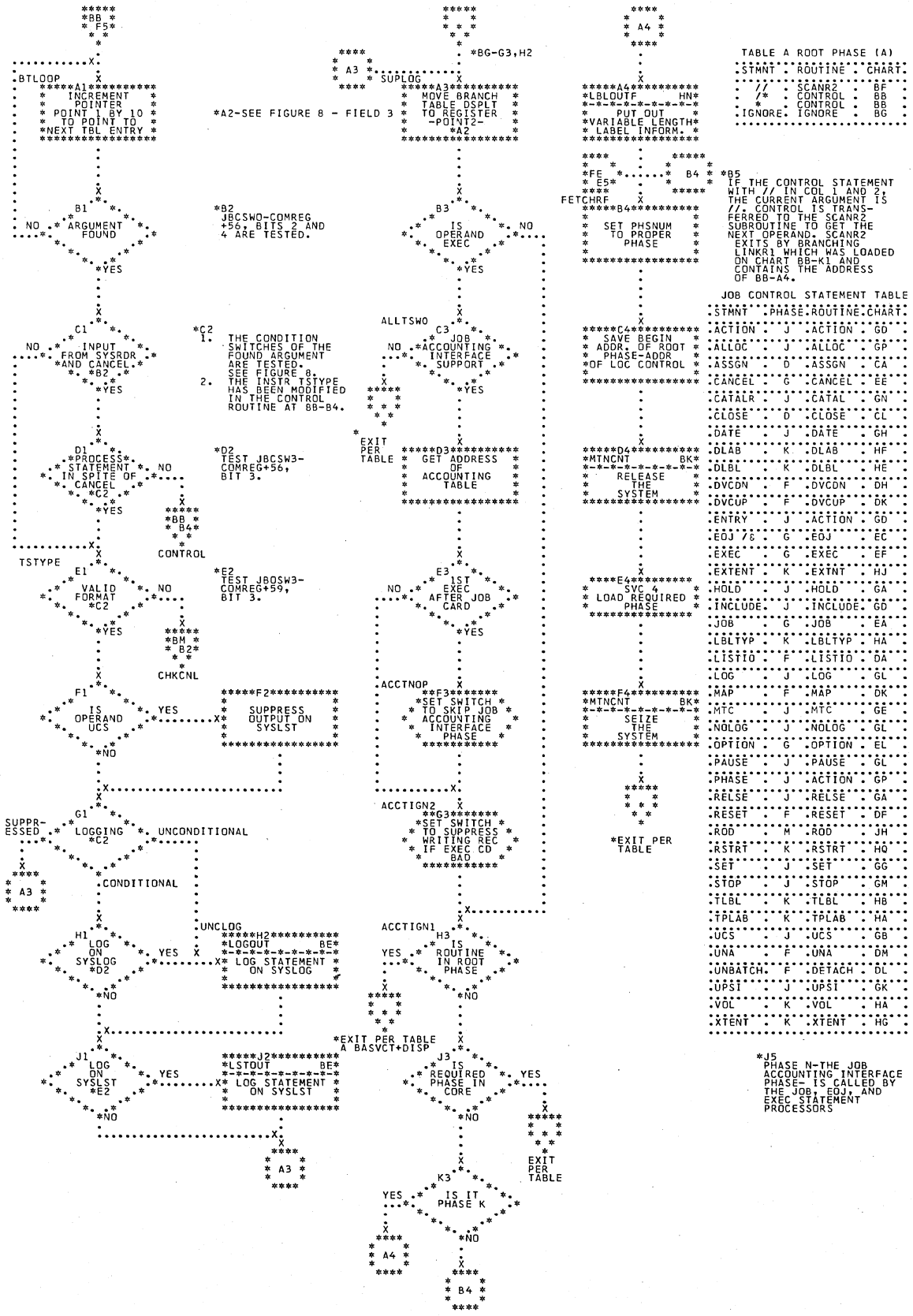


TABLE A ROOT PHASE (A)

STMNT	ROUTINE	CHART
//	SCANR2	BF
/*	CONTROL	BB
*	CONTROL	BB
IGNORE	IGNORE	BG

JOB CONTROL STATEMENT TABLE

STMNT	PHASE	ROUTINE	CHART
ACTION	J	ACTION	GD
ALLOC	J	ALLOC	GP
ASSGN	D	ASSGN	CA
CANCEL	G	CANCEL	EE
CATALR	J	CATAL	GN
CLOSE	D	CLOSE	CL
DATE	J	DATE	GH
DLAB	K	DLAB	HF
DLBL	K	DLBL	HE
DVCDN	F	DVCDN	DH
DVCUP	F	DVCUP	DK
ENTRY	J	ACTION	GD
EOJ /&	G	EOJ	EC
EXEC	G	EXEC	EF
EXTENT	K	EXTNT	HJ
HOLD	J	HOLD	GA
INCLUDE	J	INCLUDE	GD
JOB	G	JOB	EA
LBLTYP	K	LBLTYP	HA
LISTIO	F	LISTIO	DA
LOG	J	LOG	GL
MAP	F	MAP	DK
MTC	J	MTC	GE
NOLOG	J	NOLOG	GL
OPTION	G	OPTION	EL
PAUSE	J	PAUSE	GL
PHASE	J	ACTION	GP
RELSE	J	RELSE	GA
RESET	F	RESET	DF
ROD	M	ROD	JH
RSTRT	K	RSTRT	HQ
SET	J	SET	GG
STOP	J	STOP	GM
TLBL	K	TLBL	HB
TPLAB	K	TPLAB	HA
UCS	J	UCS	GB
UNA	F	UNA	DM
UNBATCH	F	DETACH	DL
UPSI	J	UPSI	GK
VOL	K	VOL	HA
XTENT	K	XTENT	HG

Chart BD. $JOBCTLA - DSKINT Subroutine
Refer to Chart 04.

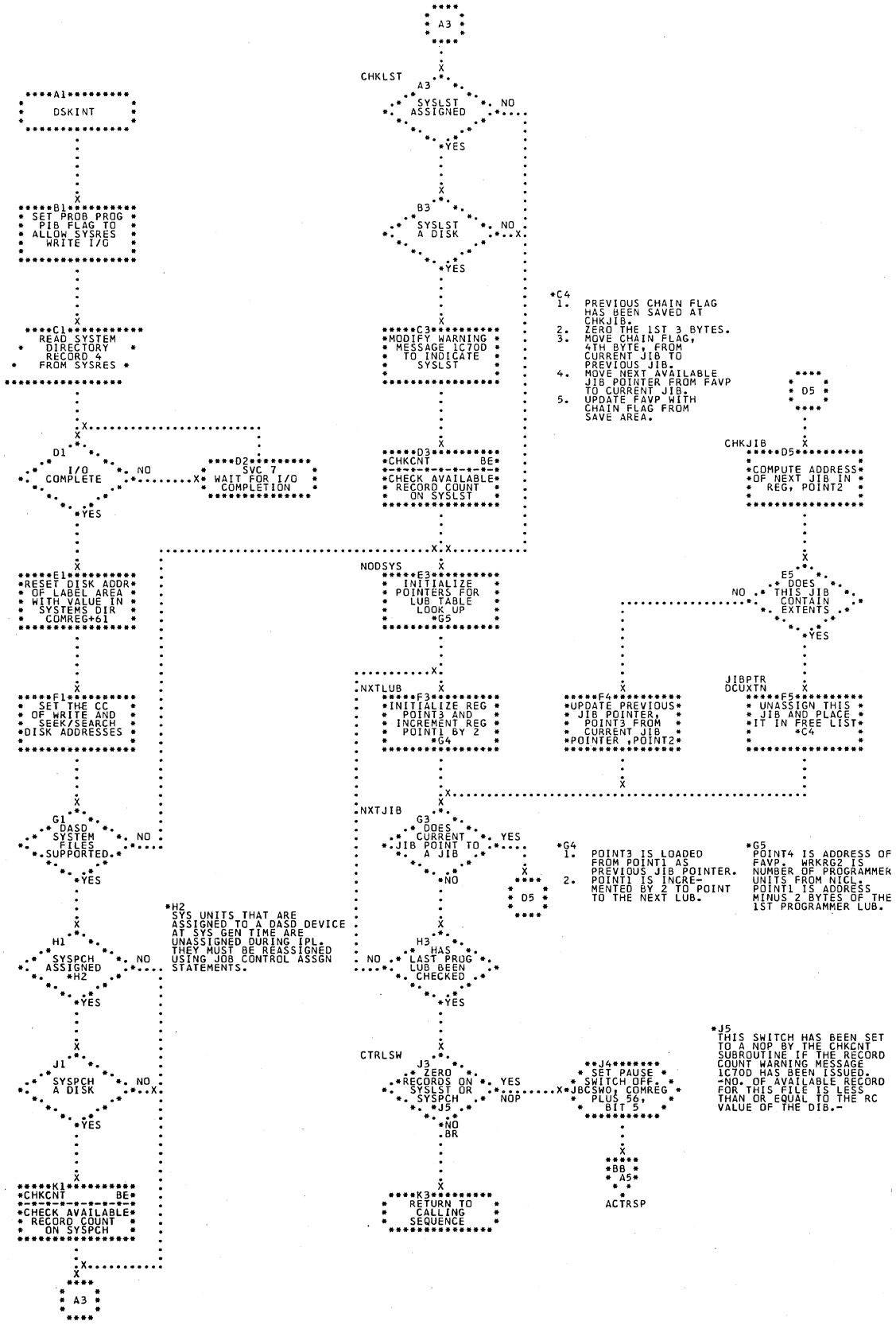


Chart BE. $JOBCTLA - Message Subroutines
Refer to Chart 04.

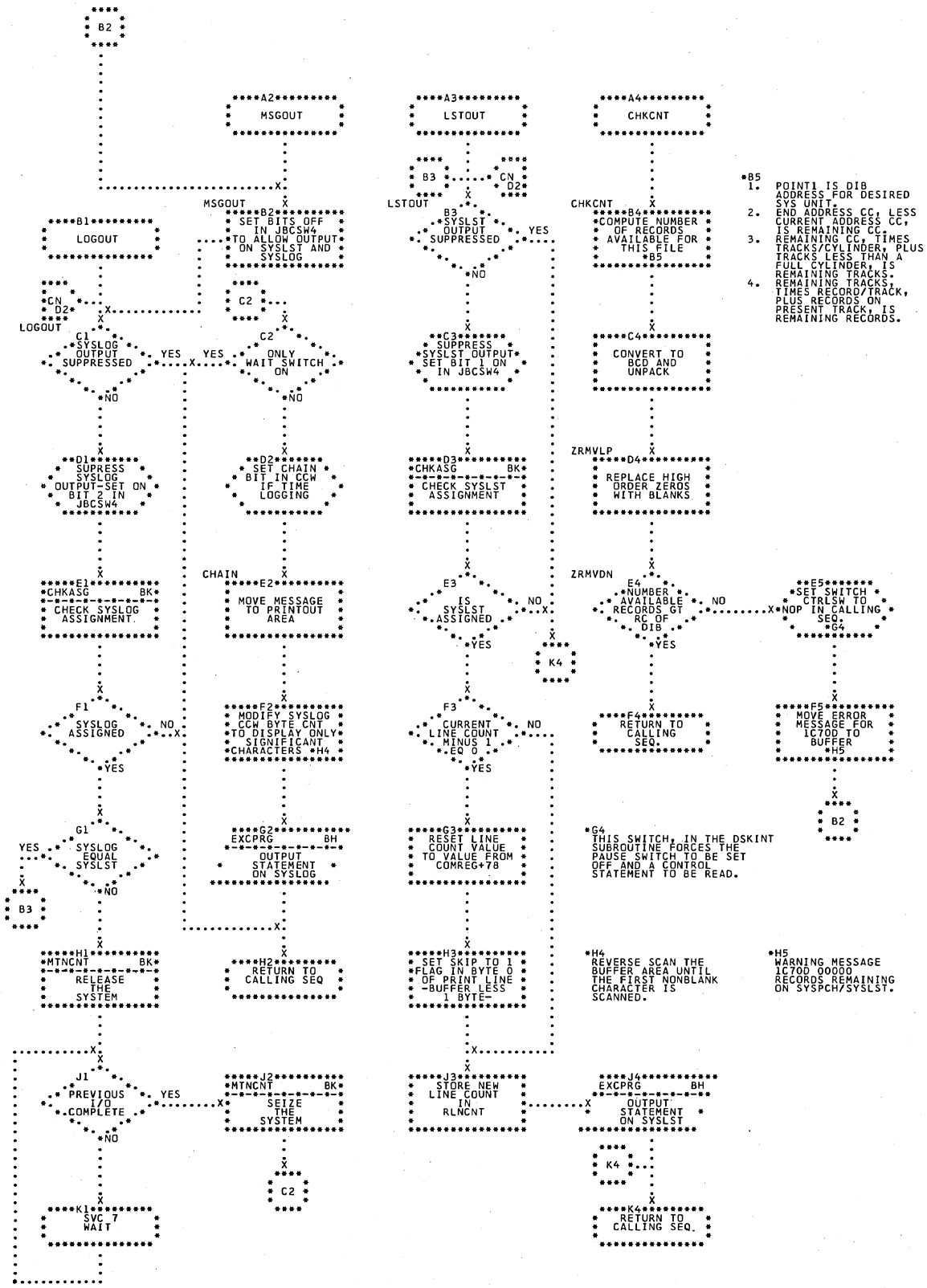


Chart BF. $JOBCTLA - Operand Scan Subroutines
Refer to Chart 04.

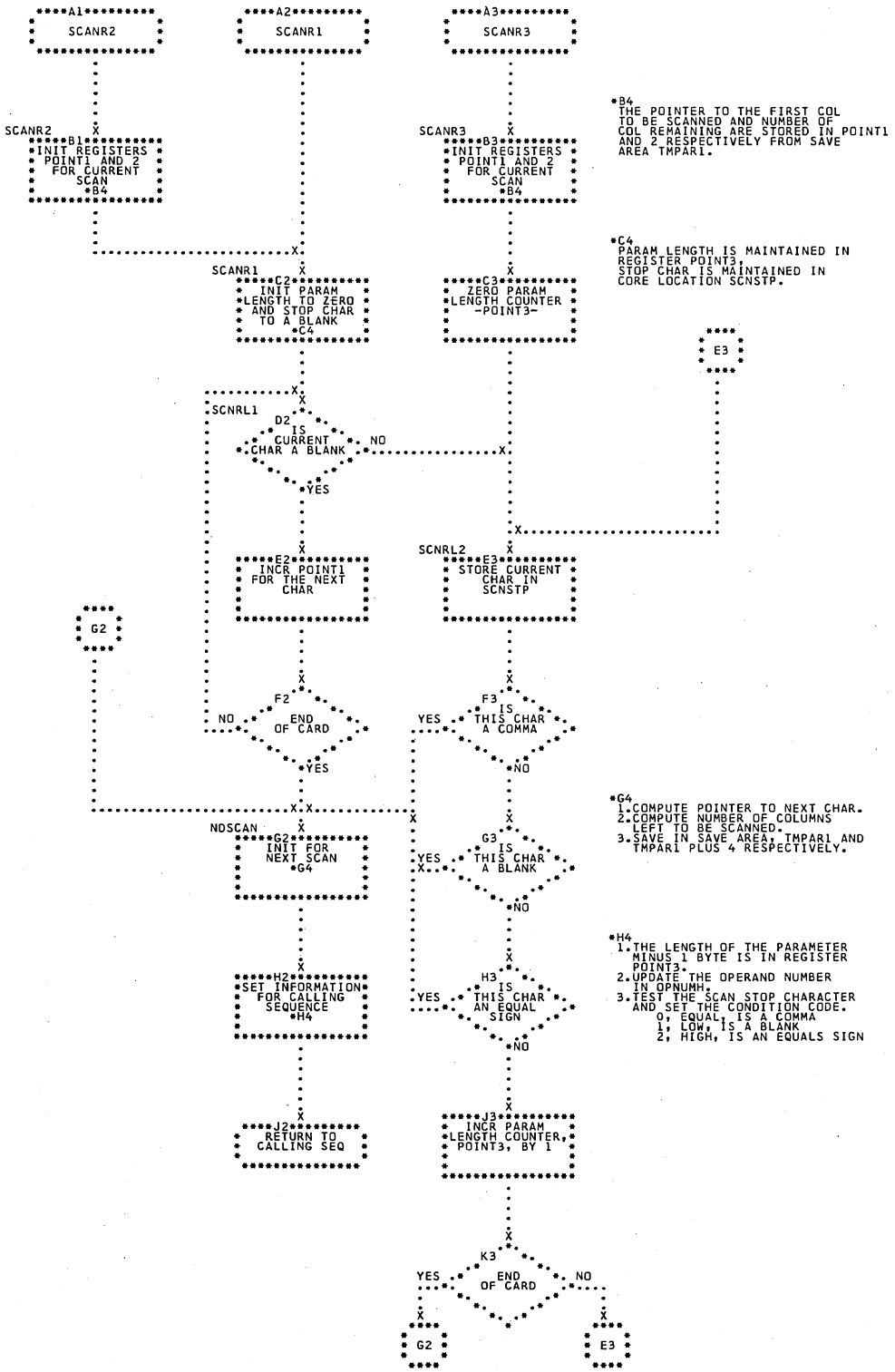


Chart BG. $JOBCTLA - Miscellaneous Subroutines
Refer to Chart 04.

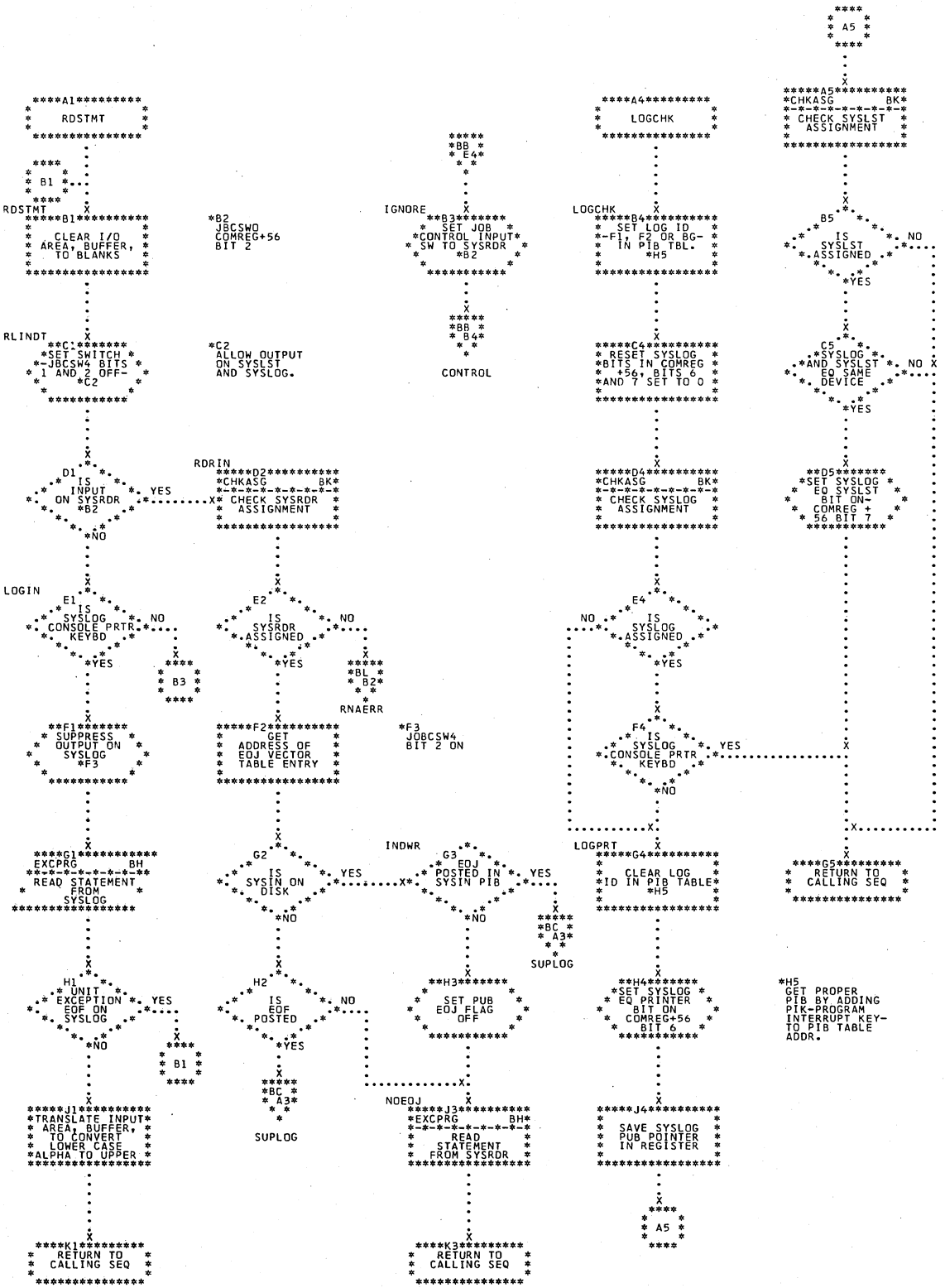


Chart BH. $JOBCTLA - EXCP Subroutines (Part 1 of 2)
Refer to Chart 04.

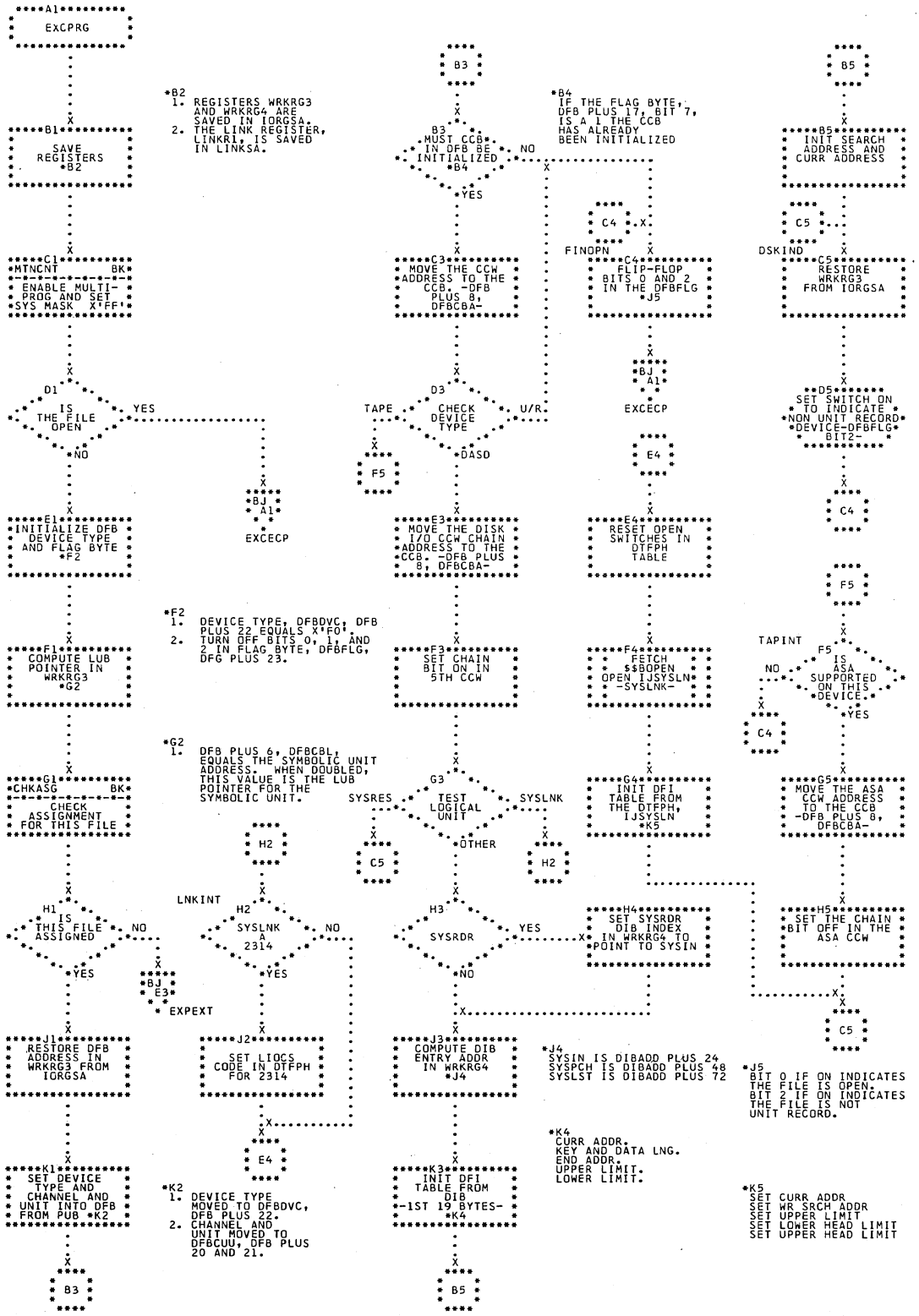


Chart BJ. $JOBCTLA - EXCP Subroutines (Part 2 of 2)
Refer to Chart 04.

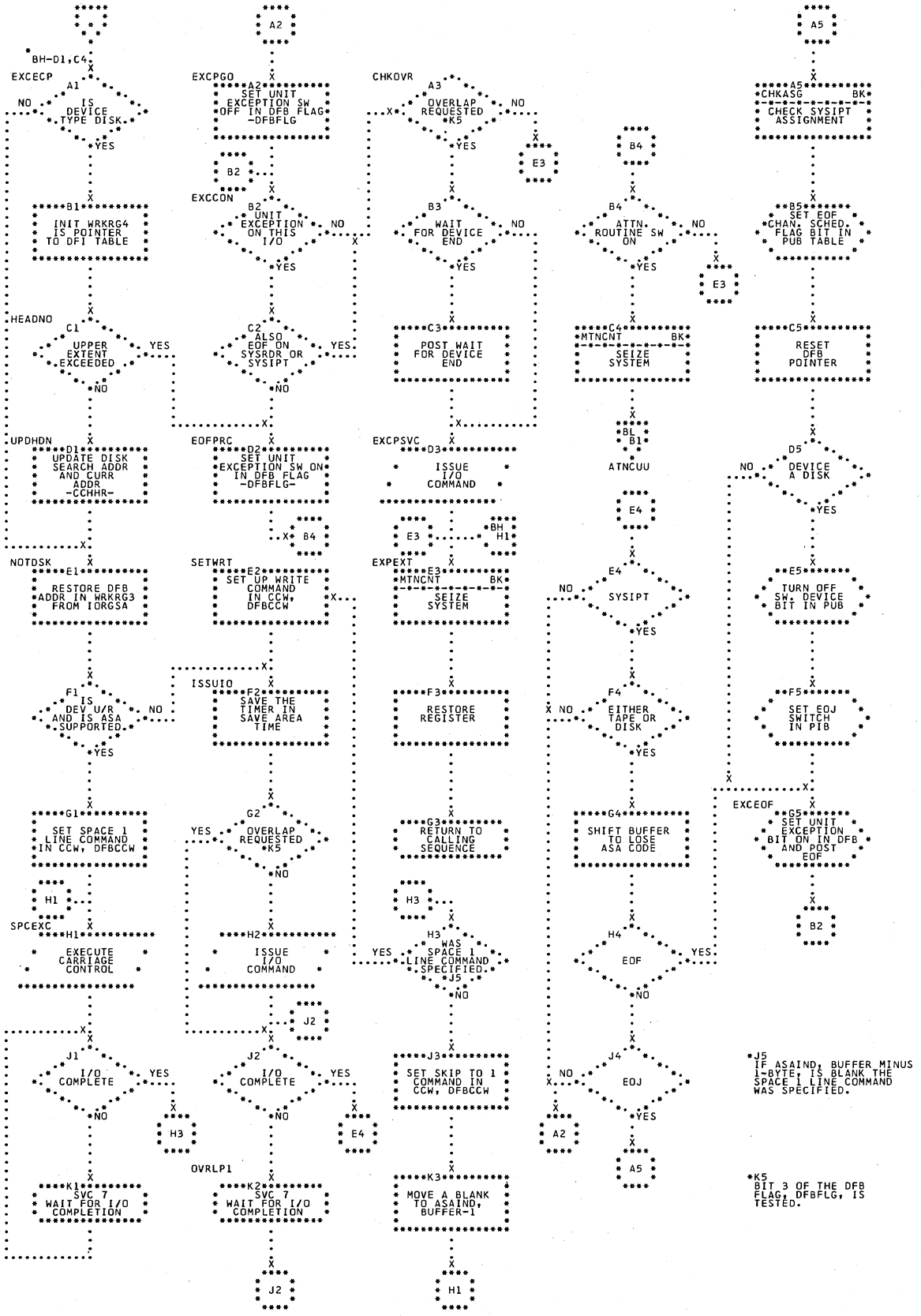


Chart BK. $JOBCTLA - Miscellaneous Subroutines
Refer to Chart 04.

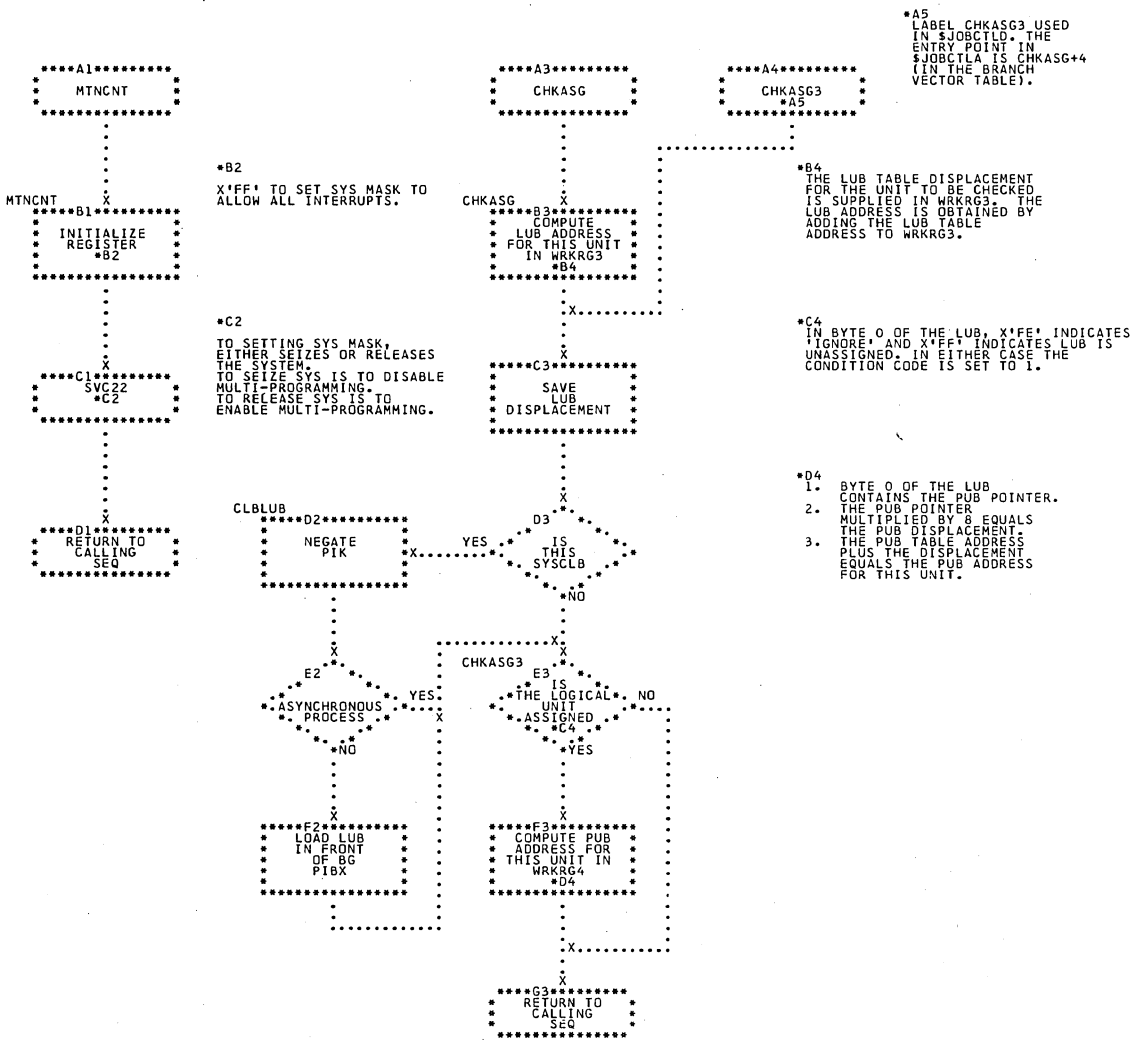


Chart BL. $JOBCTLA - Error Subroutines (Part 1 of 2)
Refer to Chart 04.

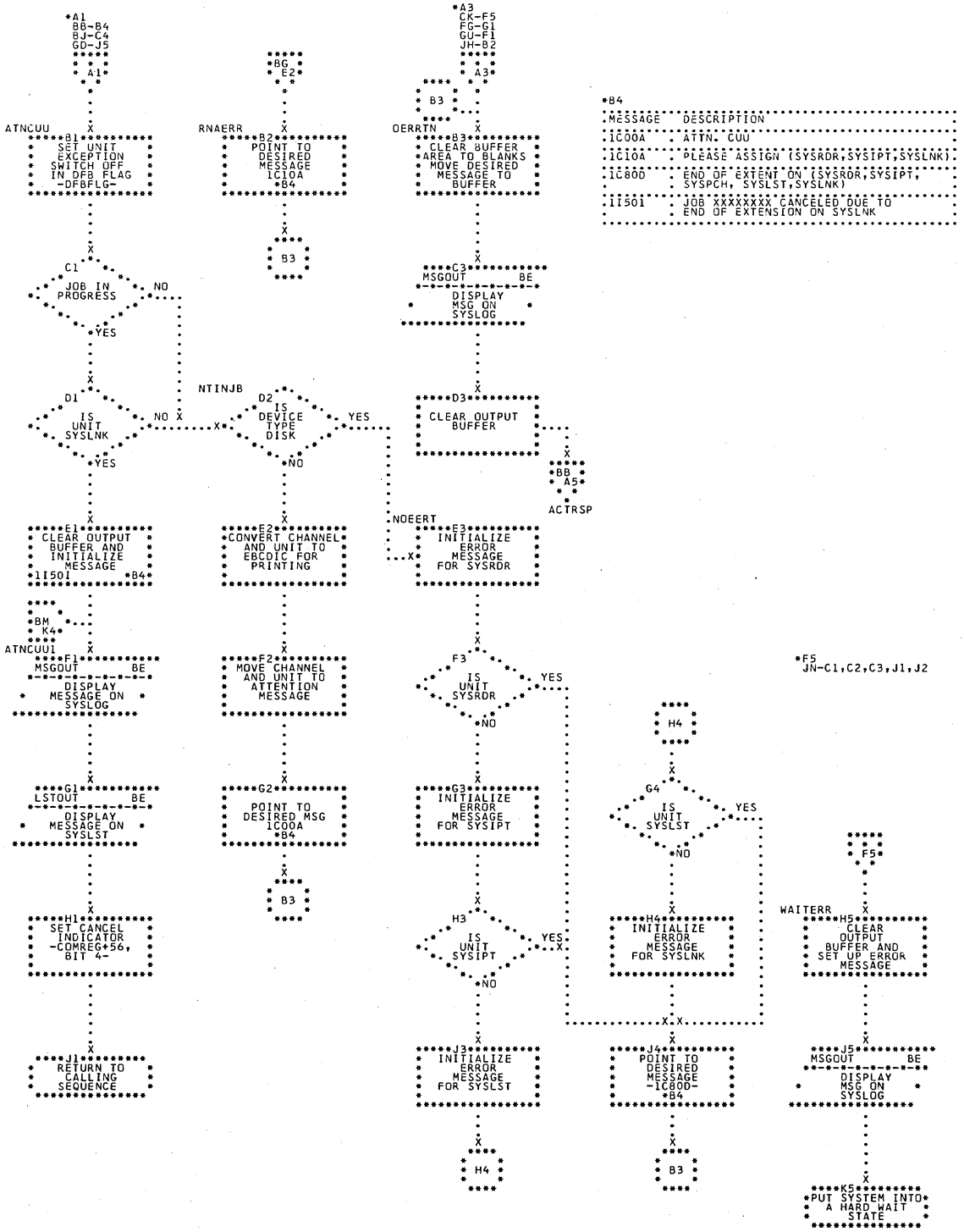


Chart BM. $JOBCTLA - Error Subroutines (Part 2 of 2)
Refer to Chart 04.

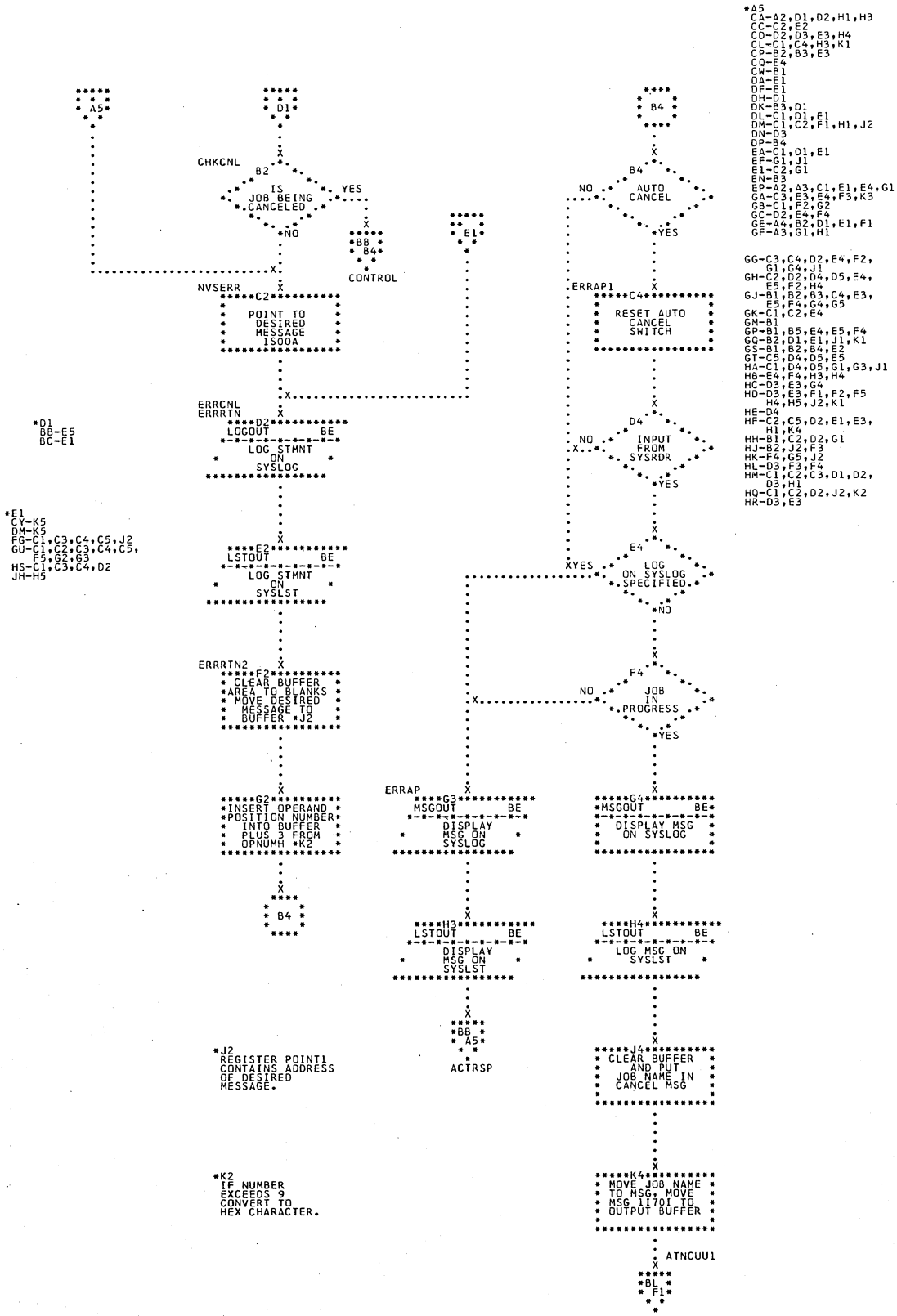


Chart BN. $JOBCTLA - Relocation Subroutines
Refer to Chart 04.

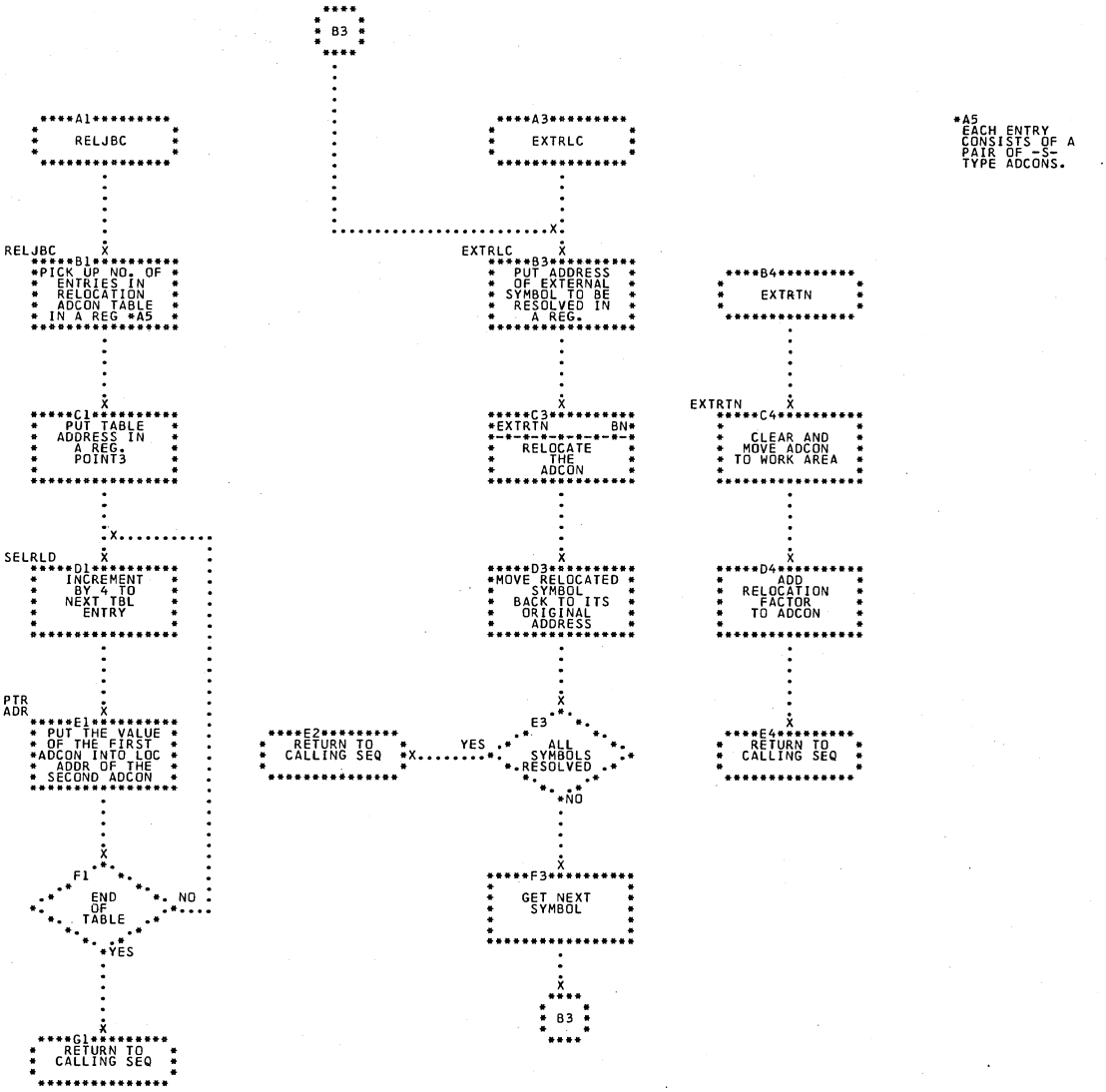


Chart CA. $JOBCTLD - ASSGN Statement Processor (Part 1 of 10)
Refer to Chart 05.

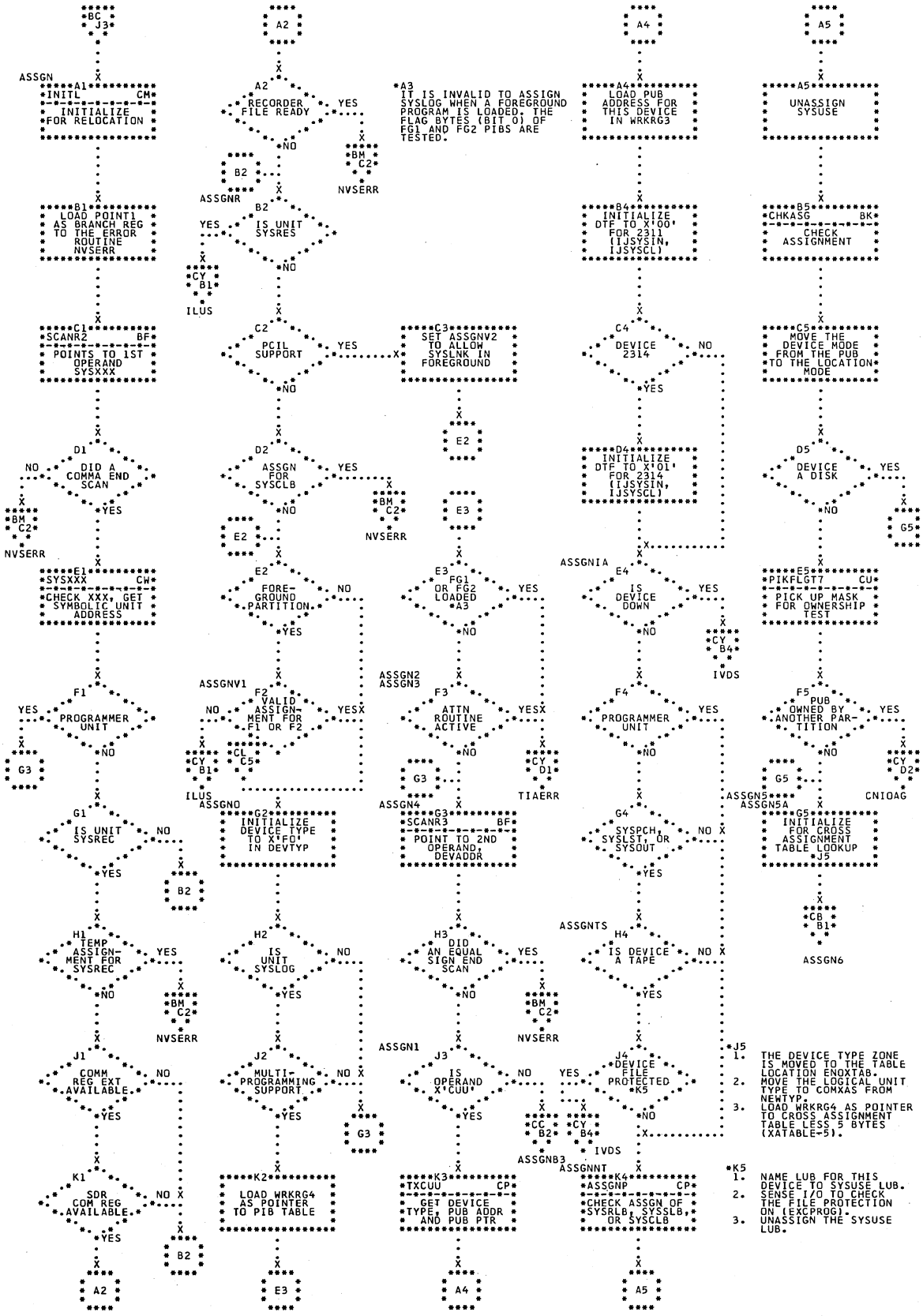
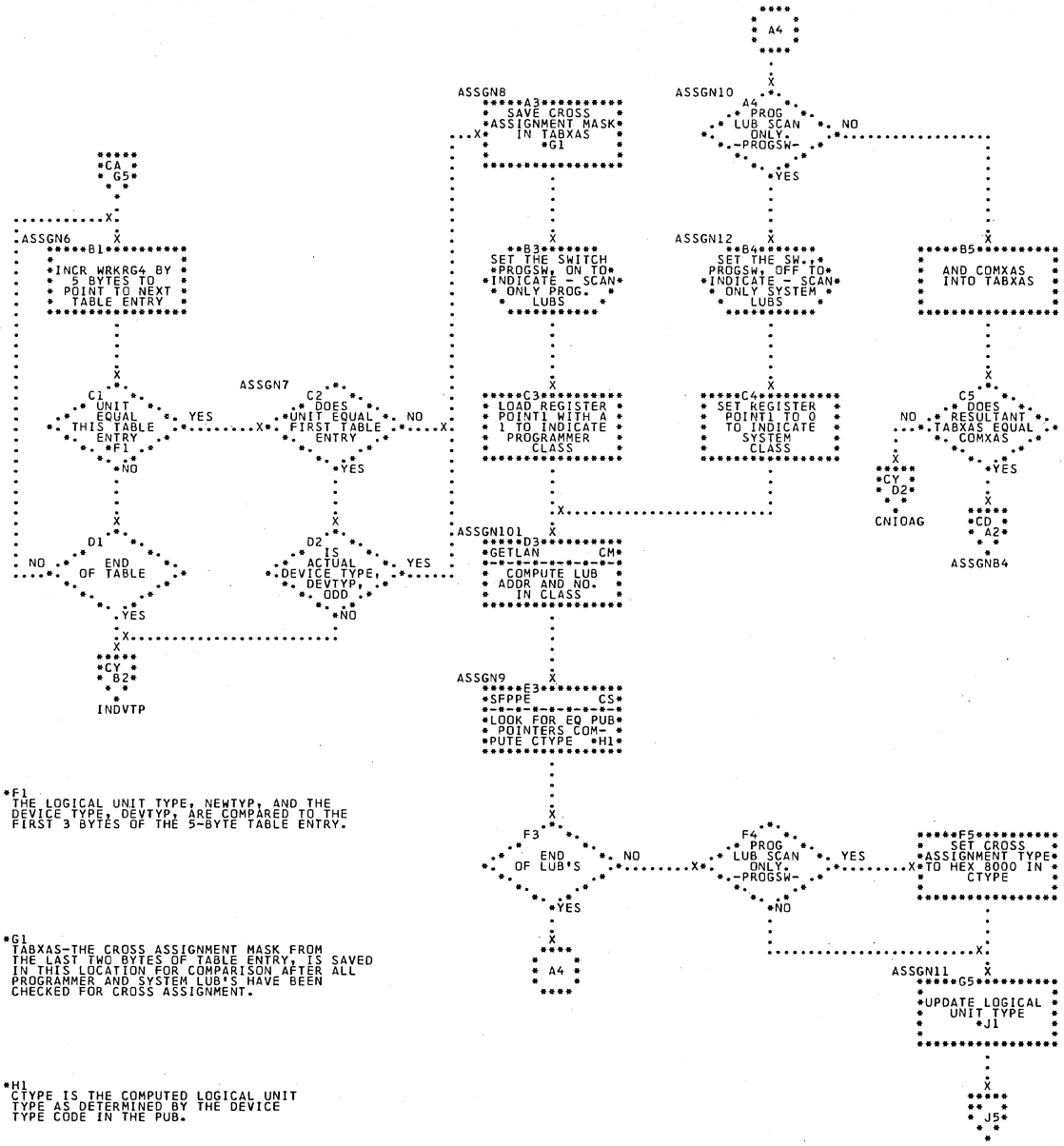


Chart CB. $JOBCTLD - ASSGN Statement Processor (Part 2 of 10)
Refer to Chart 05.



*F1
THE LOGICAL UNIT TYPE, NEWTYP, AND THE DEVICE TYPE, DEVTYP, ARE COMPARED TO THE FIRST 3 BYTES OF THE 5-BYTE TABLE ENTRY.

*G1
TABXAS-THE CROSS ASSIGNMENT MASK FROM THE LAST TWO BYTES OF TABLE ENTRY, IS SAVED IN THIS LOCATION FOR COMPARISON AFTER ALL PROGRAMMER AND SYSTEM LUB'S HAVE BEEN CHECKED FOR CROSS ASSIGNMENT.

*H1
CTYPE IS THE COMPUTED LOGICAL UNIT TYPE AS DETERMINED BY THE DEVICE TYPE CODE IN THE PUB.

*J1
EACH TIME AN EQUAL PUB POINTER IS FOUND, THE VALUE IN CTYPE IS ORED INTO COMXAS TO BUILD A LOGICAL ACCUMULATION OF BITS TO BE COMPARED TO THE VALUE IN TABXAS. COMXAS CONTAINED THE LOGICAL UNIT TYPE AS IT WAS SET IN NEWTYP BY THE SUBROUTINE SYSXXX. CTYP CONTAINS THE LOGICAL UNIT TYPE HEX 80 IF PROGRAMMER LUBS ARE BEING SCANNED, OR THE COMPUTED VALUE SET IN THE SUBROUTINE SFPPE IF SYSTEM LUBS ARE BEING SCANNED.

*J5
RETURN TO THE SFPPE SUBROUTINE BY BRANCHING TO THE ADDRESS IN LINKR4.

Chart CC. $JOBCTLD - ASSGN Statement Processor (Part 3 of 10)
Refer to Chart 05.

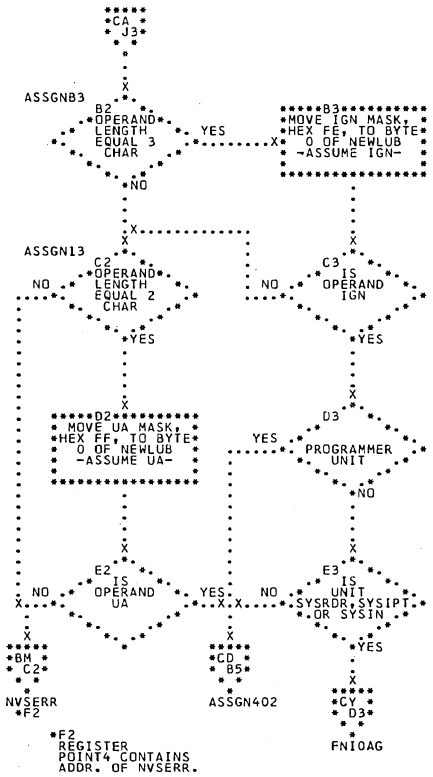


Chart CD. $JOBCTLD - ASSGN Statement Processor (Part 4 of 10)
Refer to Chart 05.

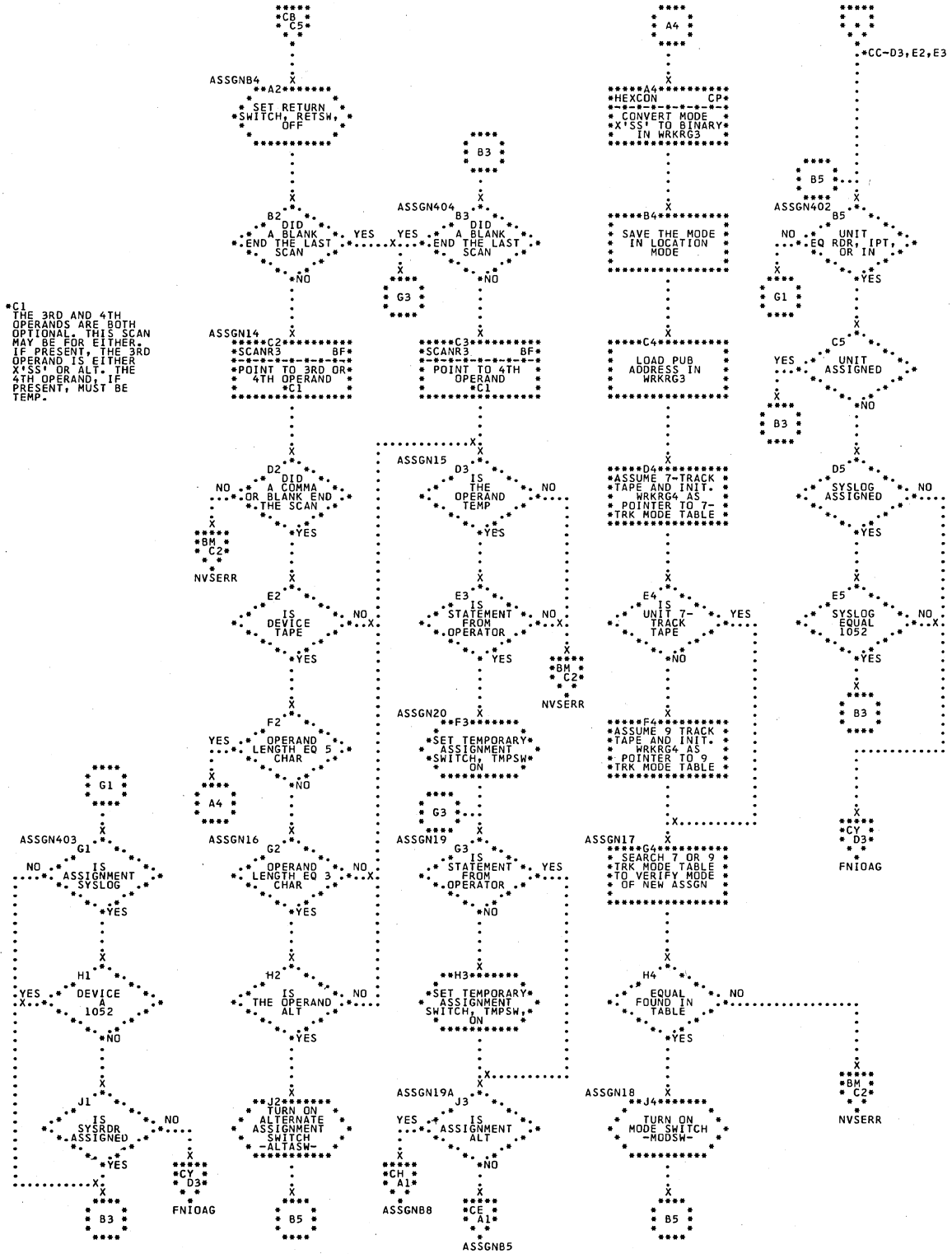


Chart CE. $JOBCTLD - ASSGN Statement Processor (Part 5 of 10)
Refer to Chart 05.

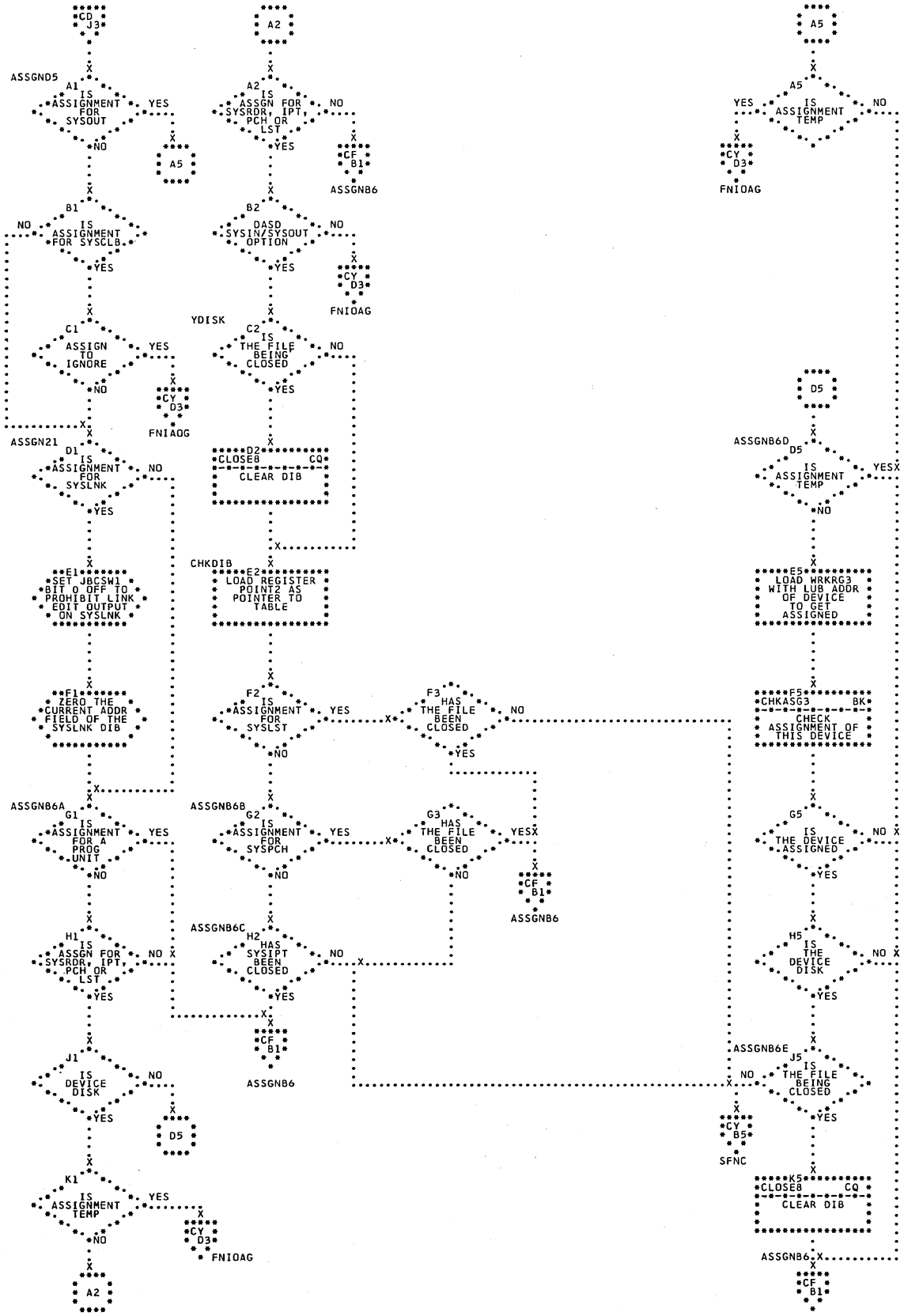


Chart CF. $JOBCTLD - ASSGN Statement Processor (Part 6 of 10)
Refer to Chart 05.

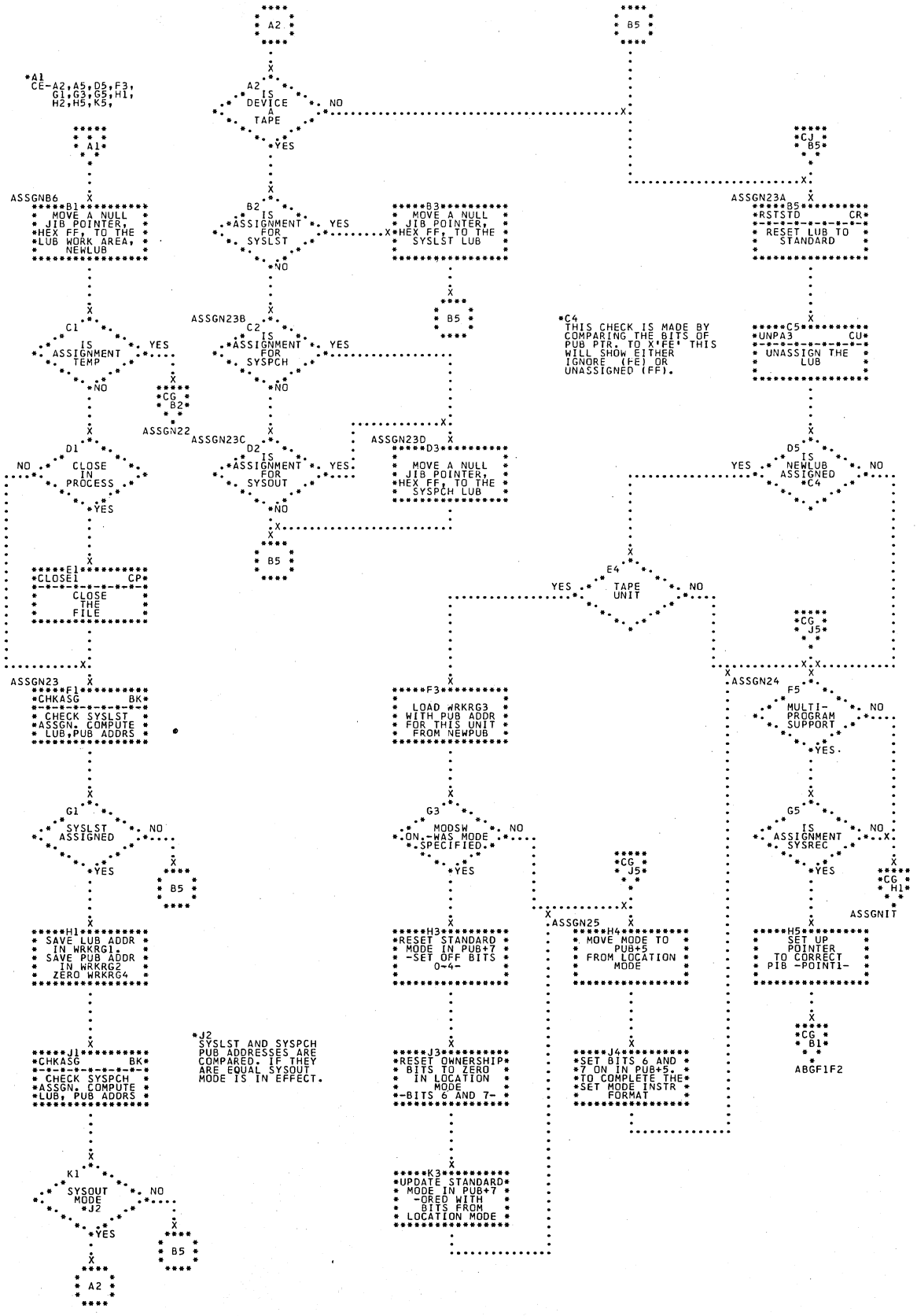


Chart CG. $JOBCTLD - ASSGN Statement Processor (Part 7 of 10)
Refer to Chart 05.

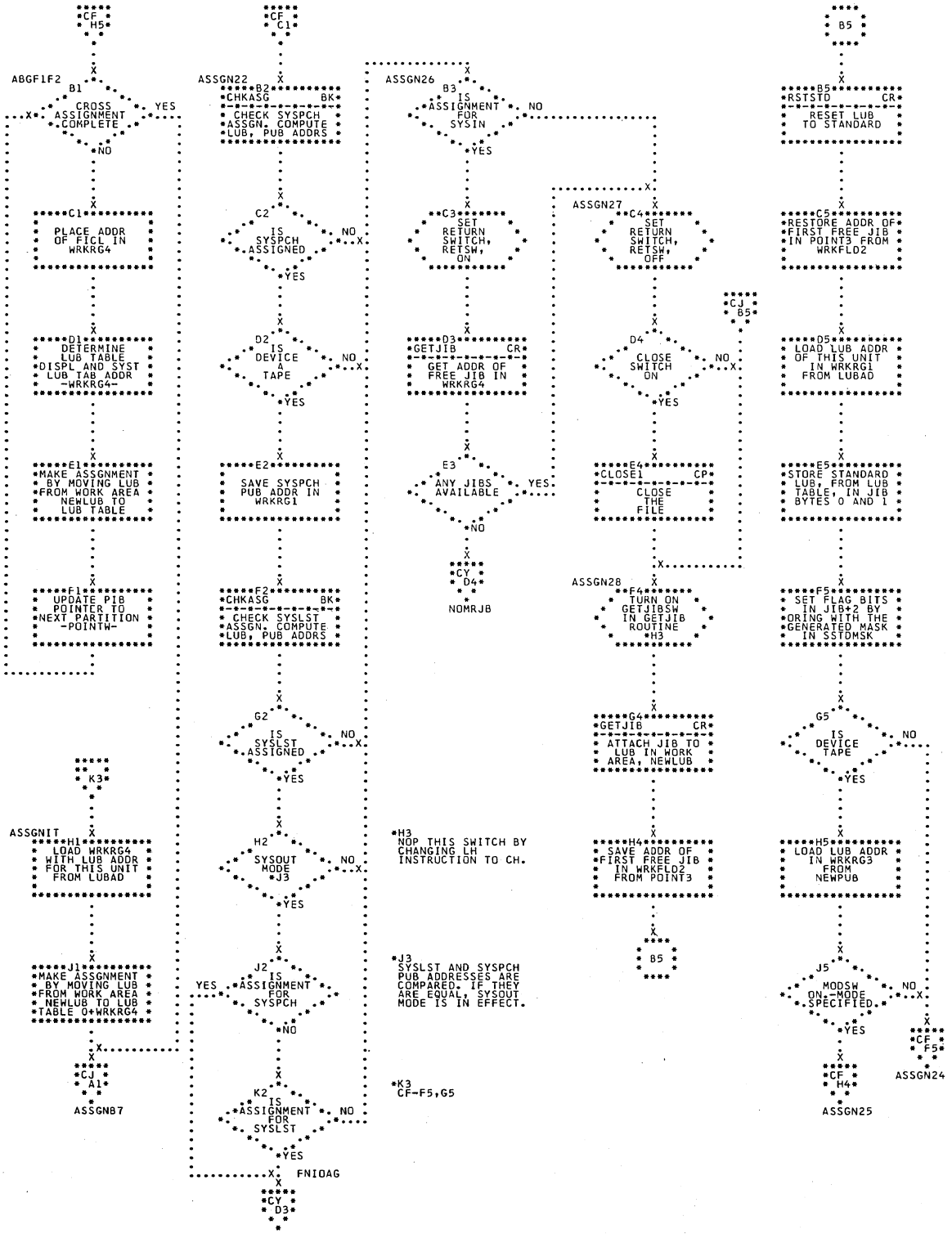


Chart CH. $JOBCTLD - ASSGN Statement Processor (Part 8 of 10)
Refer to Chart 05.

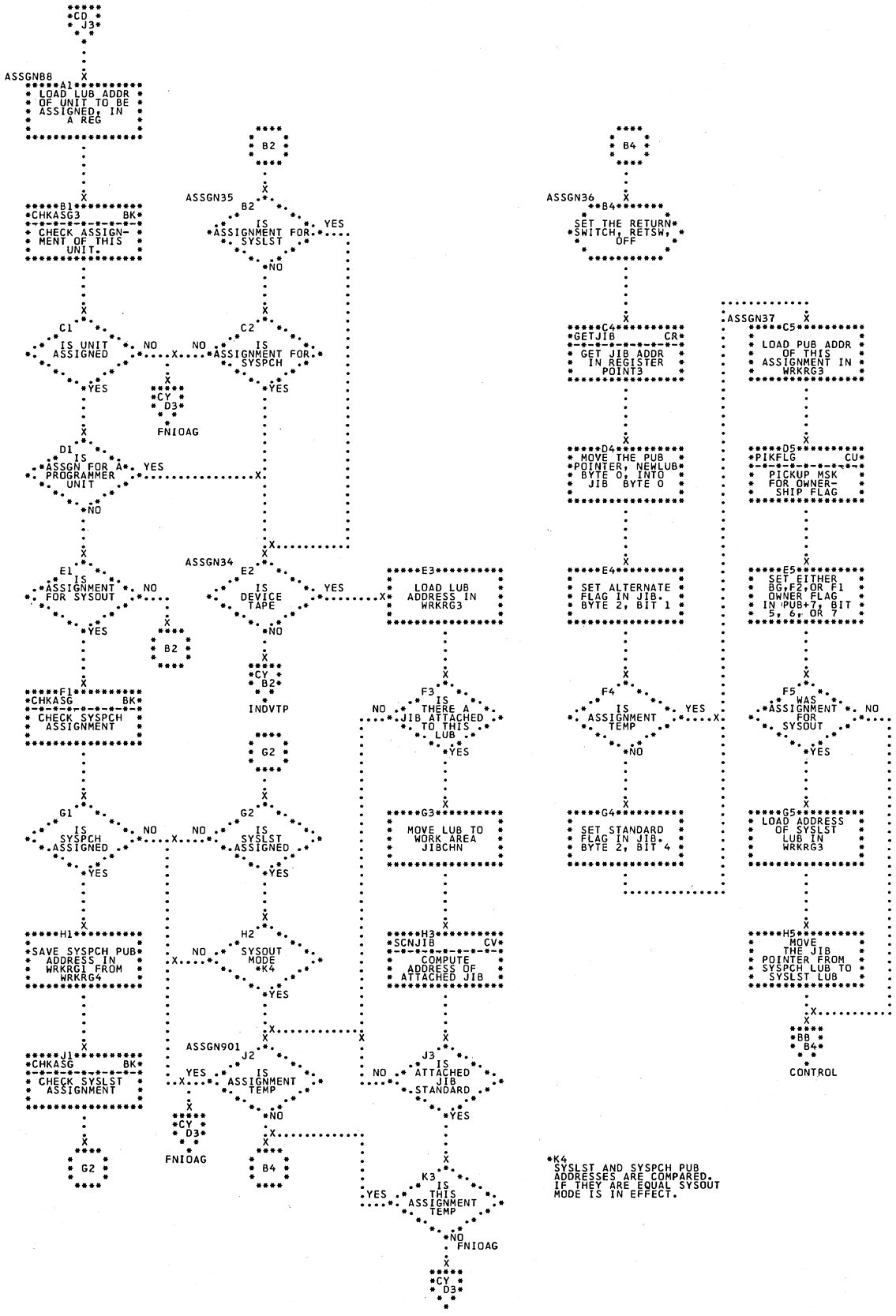


Chart CJ. $JOBCTLD - ASSGN Statement Processor (Part 9 of 10)
Refer to Chart 05.

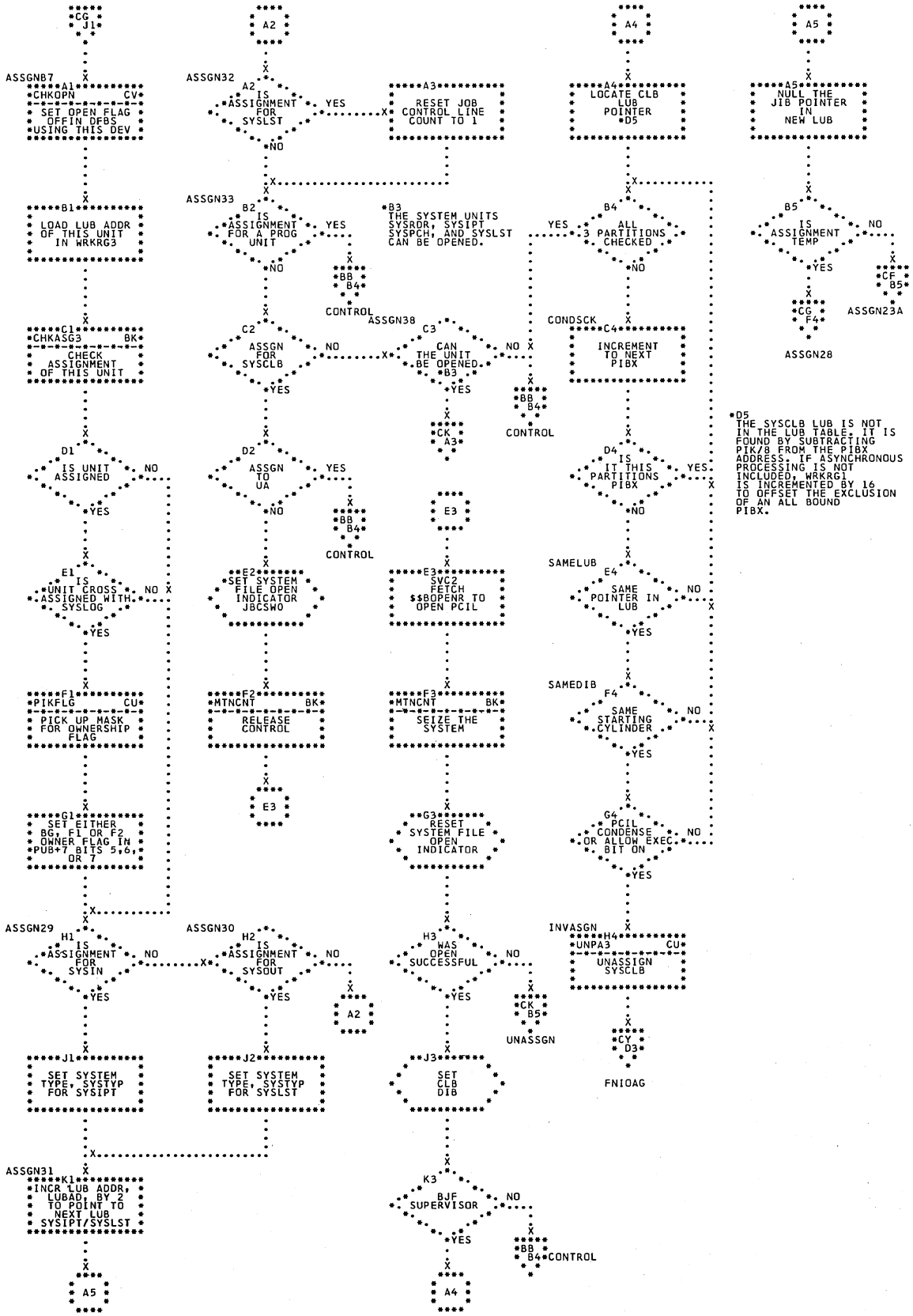


Chart CK. $JOBCTLD - ASSGN Statement Processor (Part 10 of 10)
Refer to Chart 05.

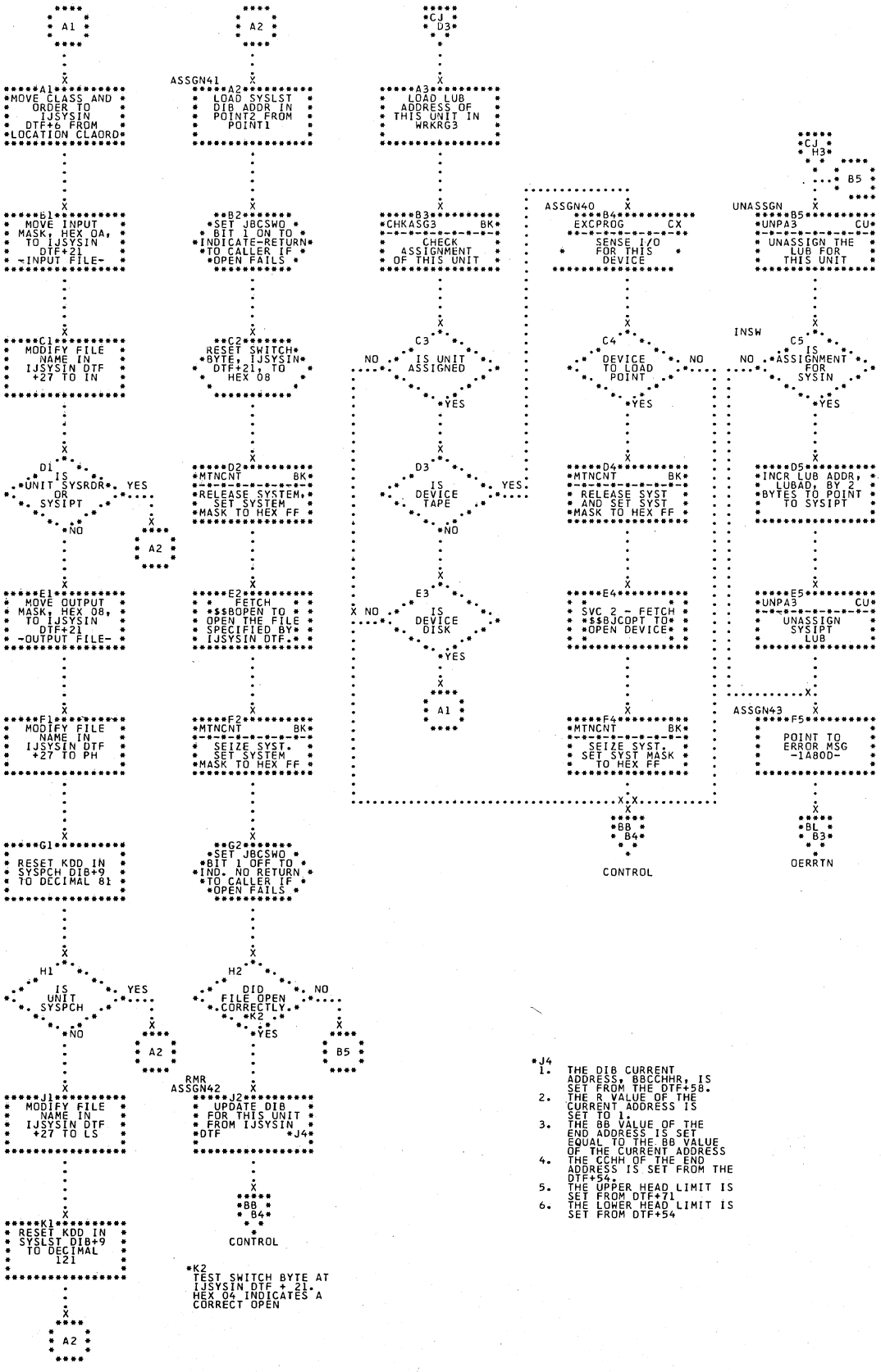


Chart CL. $JOBCTLD - CLOSE Statement Processor
Refer to Chart 06.

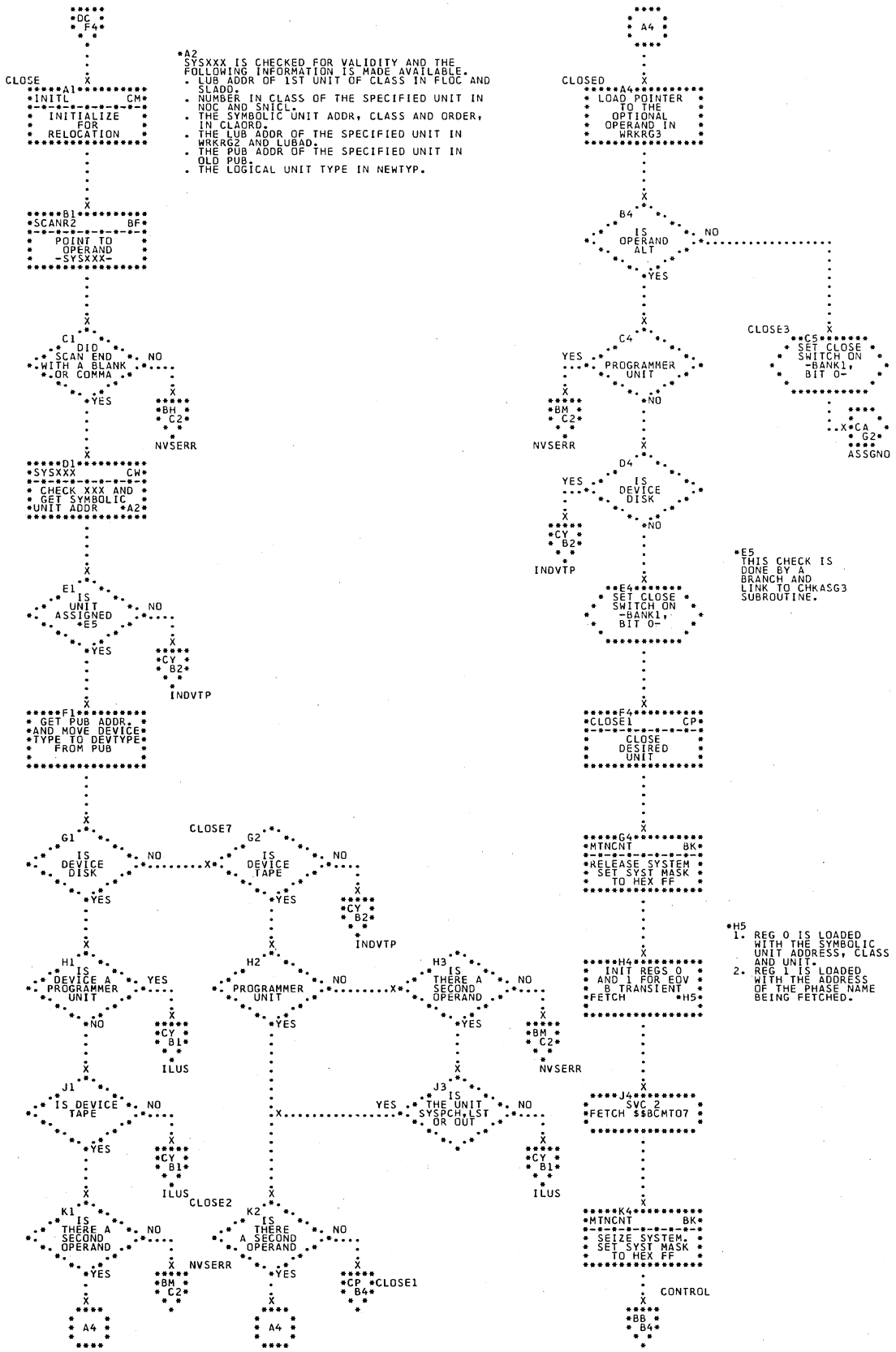


Chart CM. $JOBCTLD - Miscellaneous Subroutines (Part 1 of 3)
Refer to Chart 06.

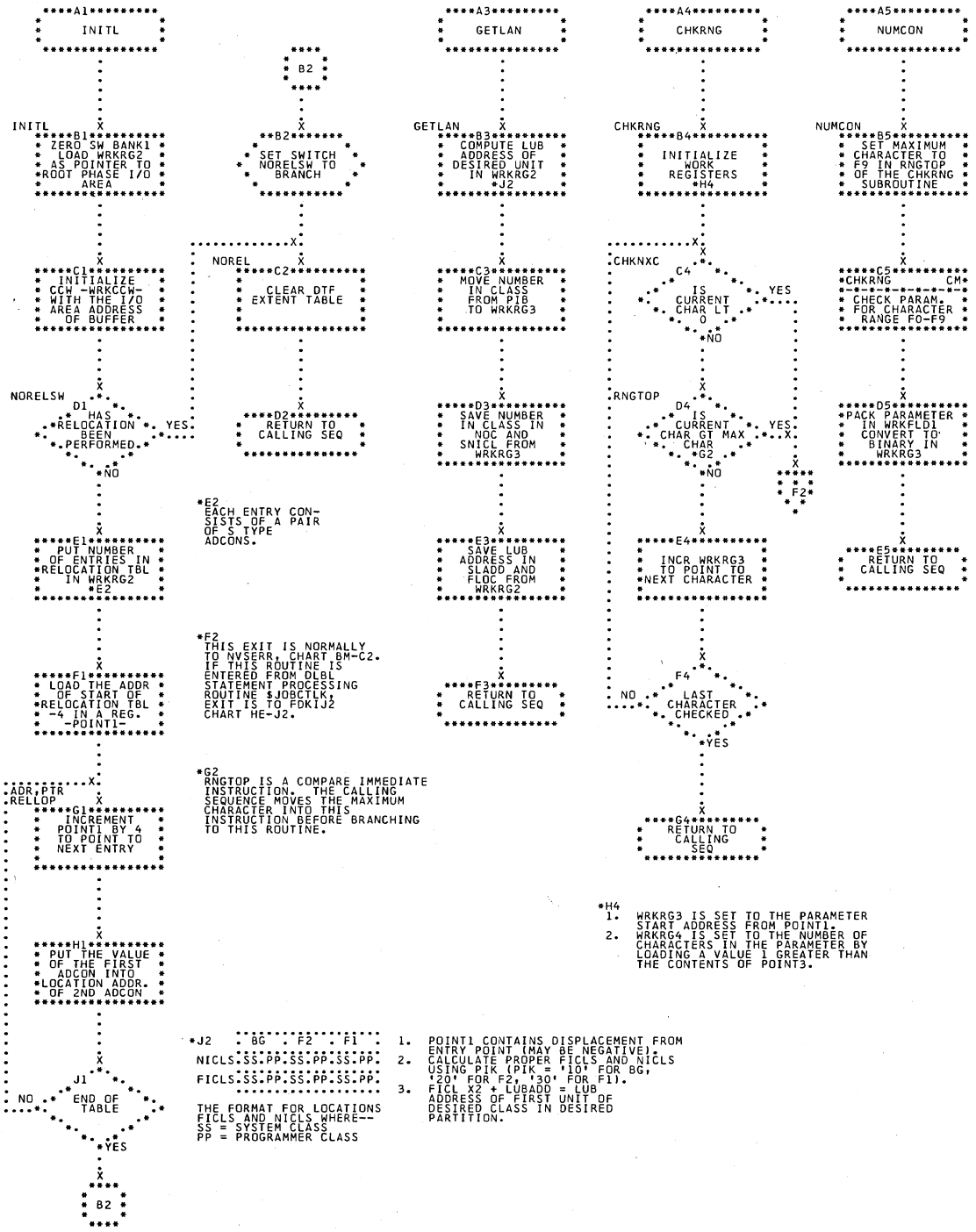


Chart CN. $JOBCTLD - Miscellaneous Subroutines (Part 2 of 3)
Refer to Chart 06.

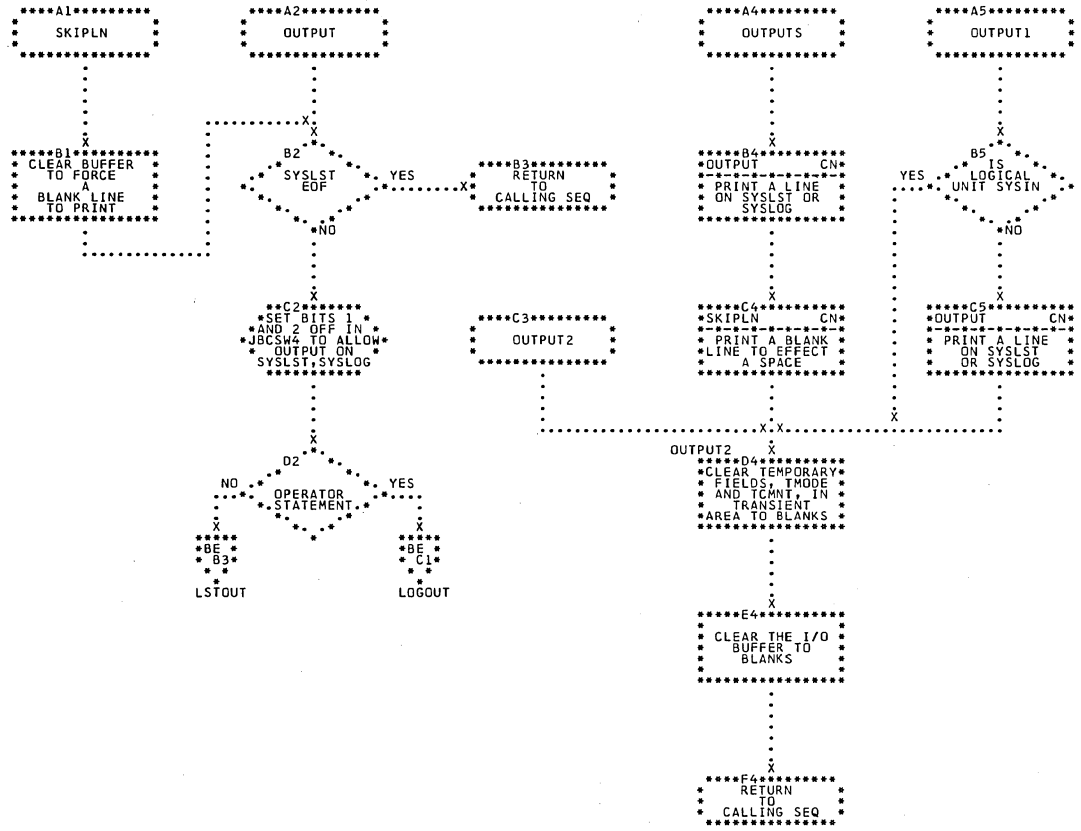


Chart CP. $JOBCTLD - Miscellaneous Subroutines (Part 3 of 3)
Refer to Chart 06.

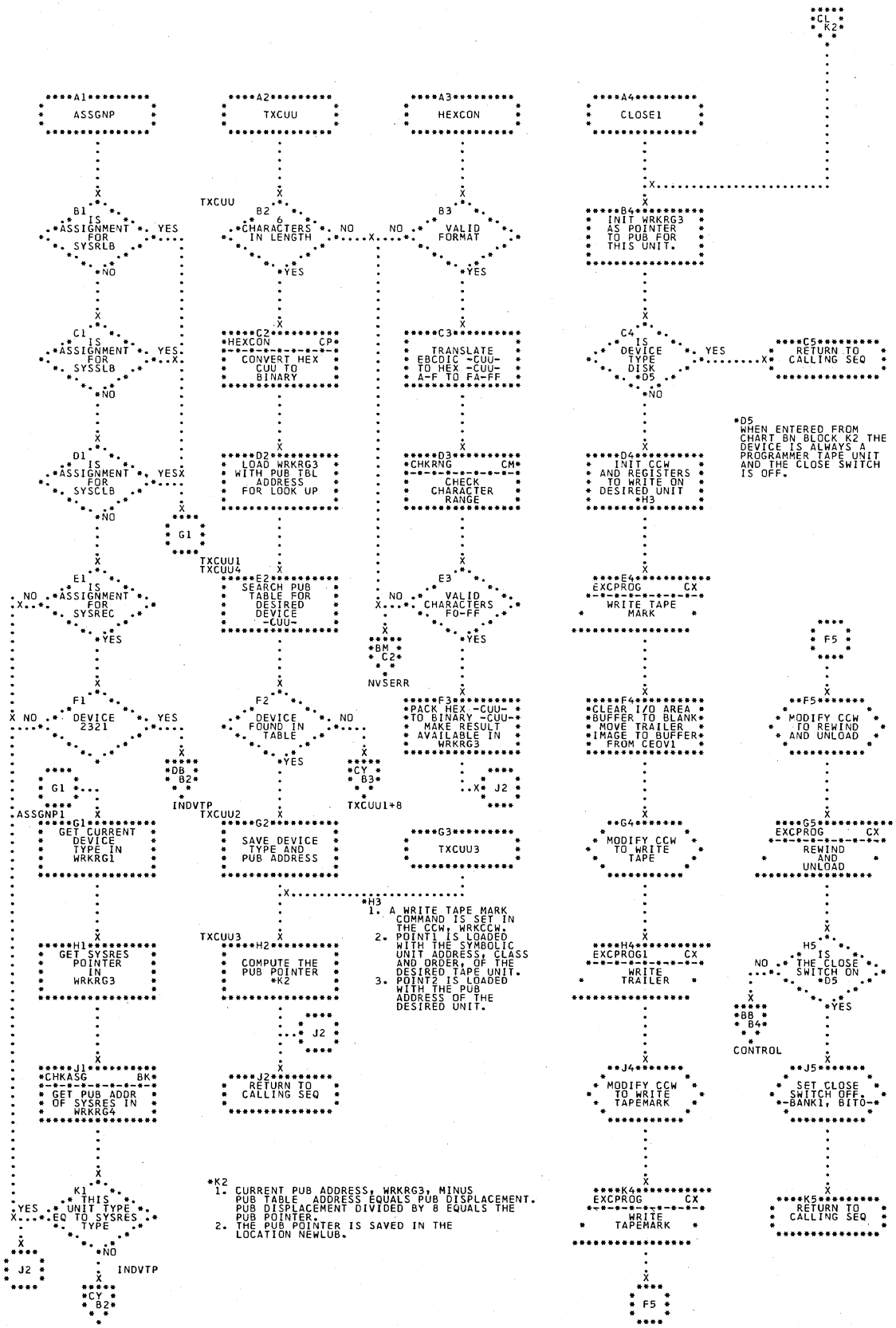


Chart CQ. $JOBCTLD - Close Subroutine
Refer to Chart 06.

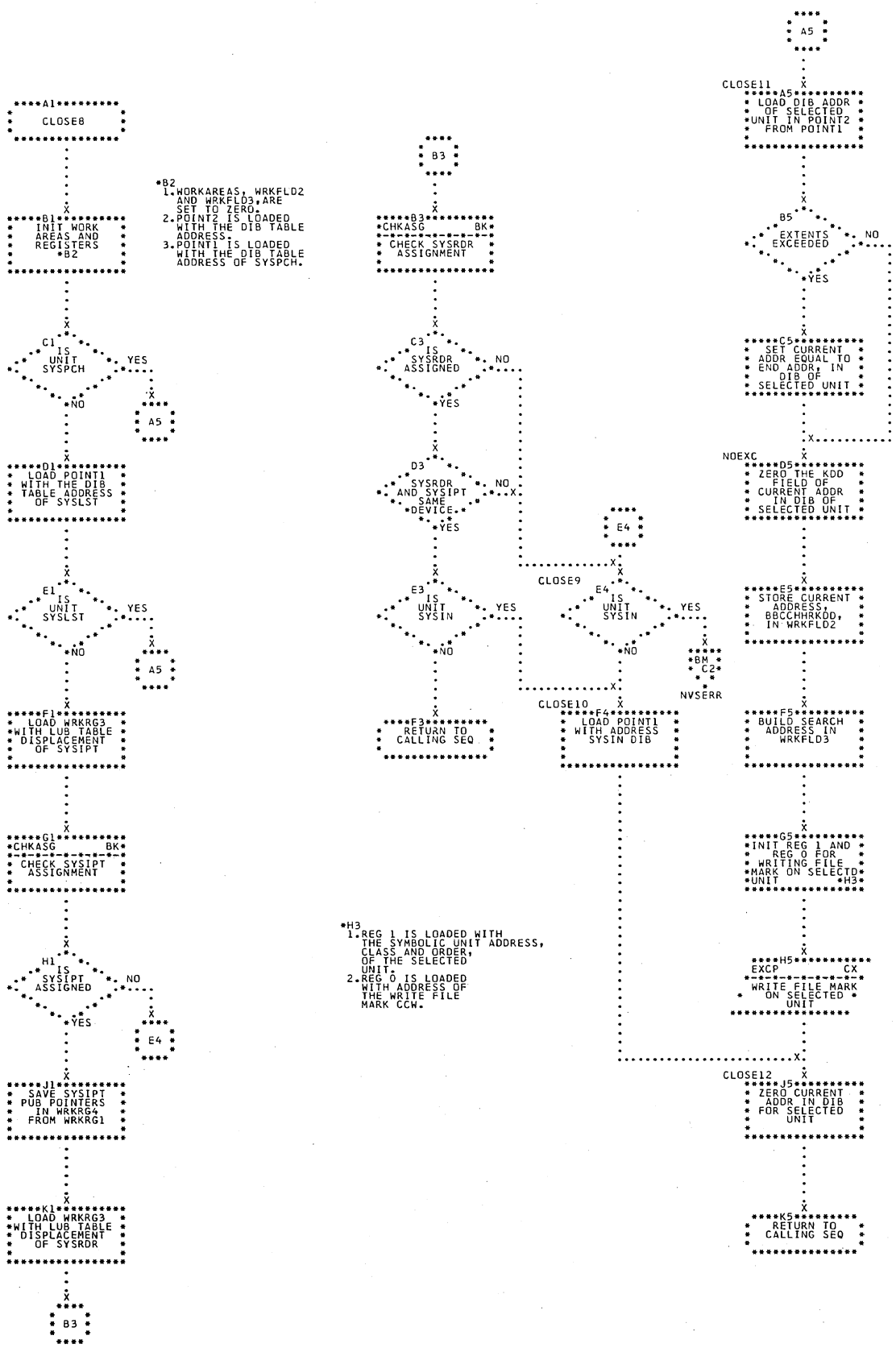


Chart CR. $JOBCTLD - Miscellaneous Subroutines (Part 1 of 5)
Refer to Chart 06.

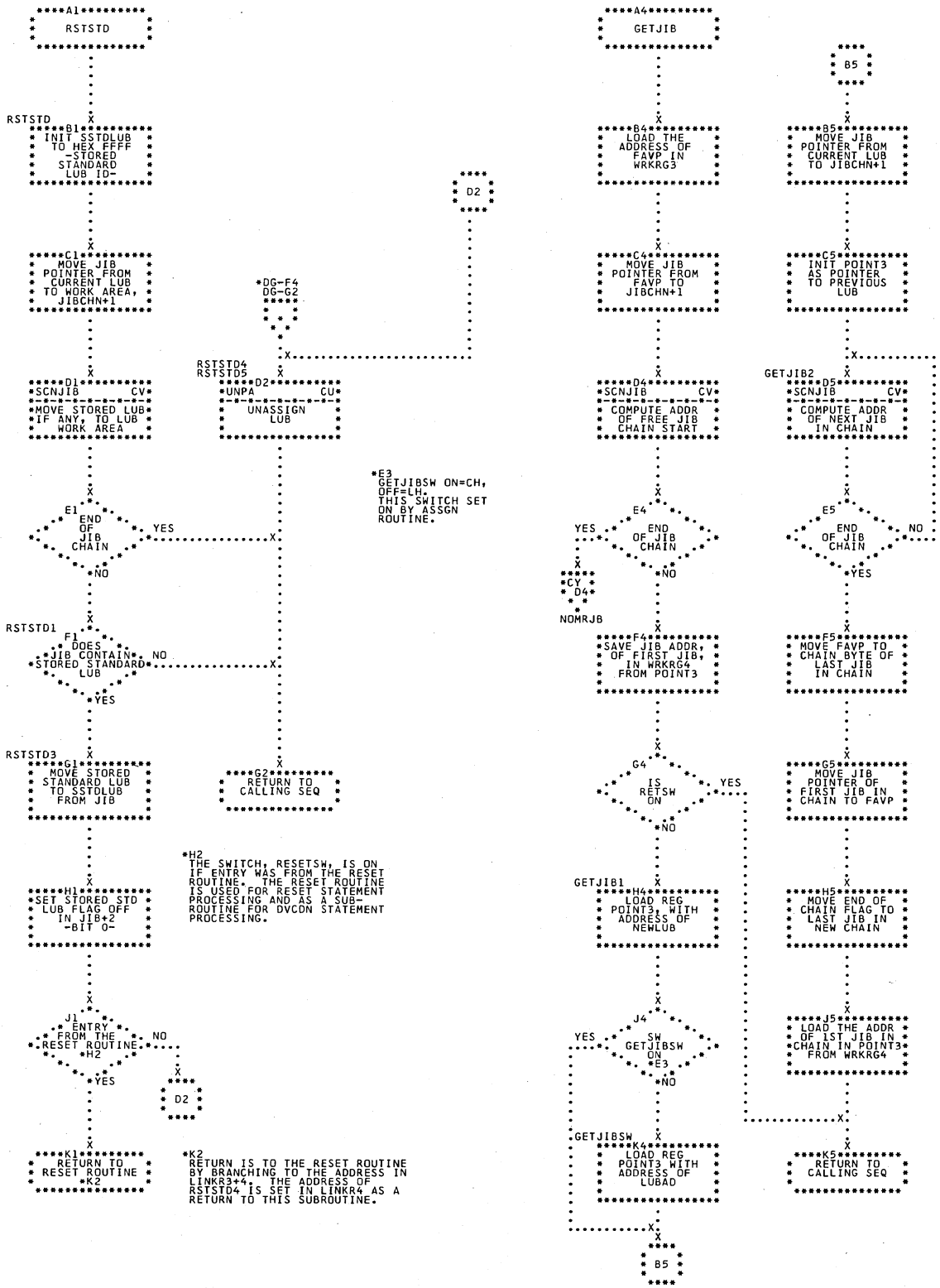


Chart CS. $JOBCTLD - Miscellaneous Subroutines (Part 2 of 5)
Refer to Chart 06.

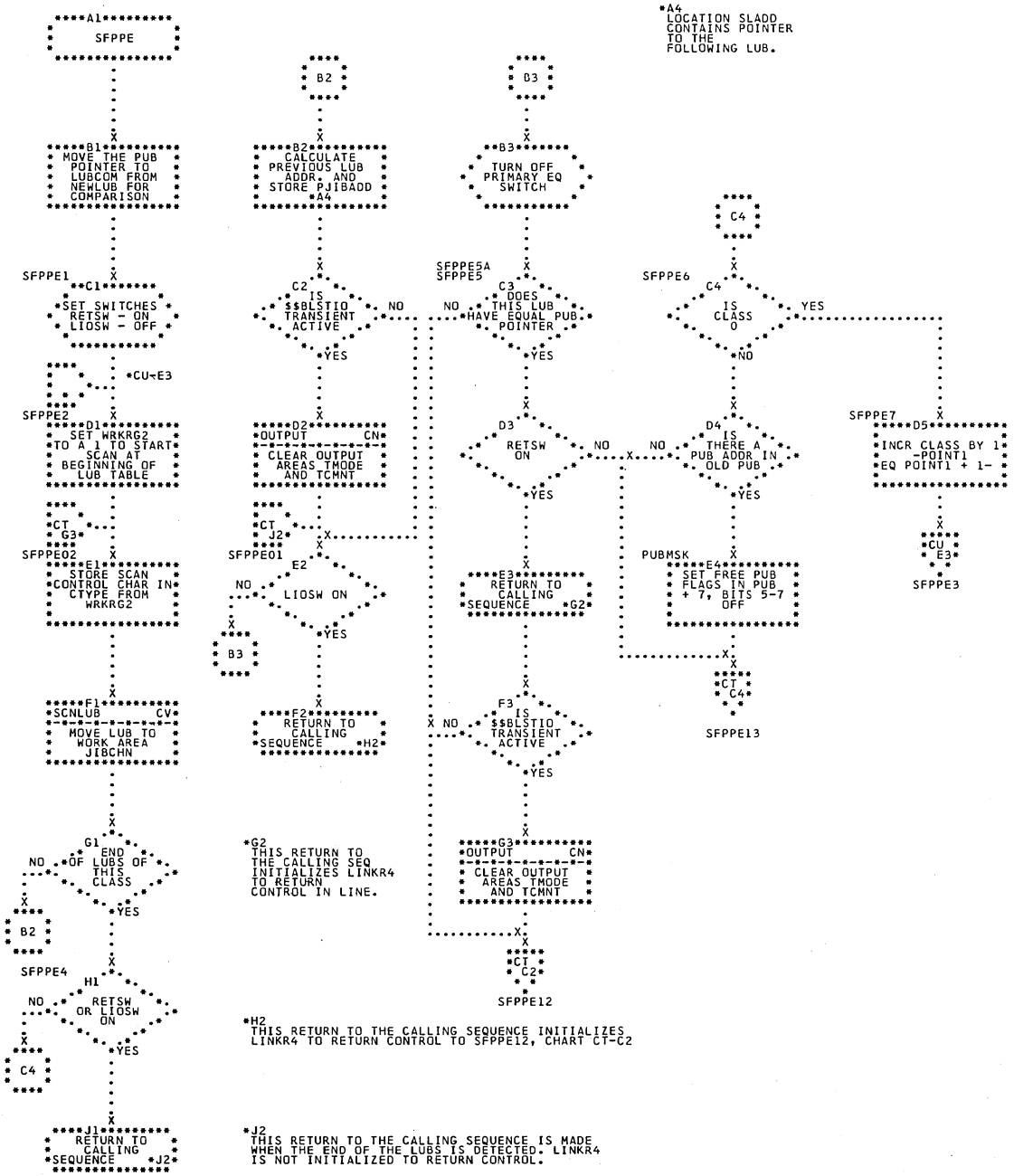


Chart CT. $JOBCTLD - Miscellaneous Subroutines (Part 3 of 5)
Refer to Chart 06.

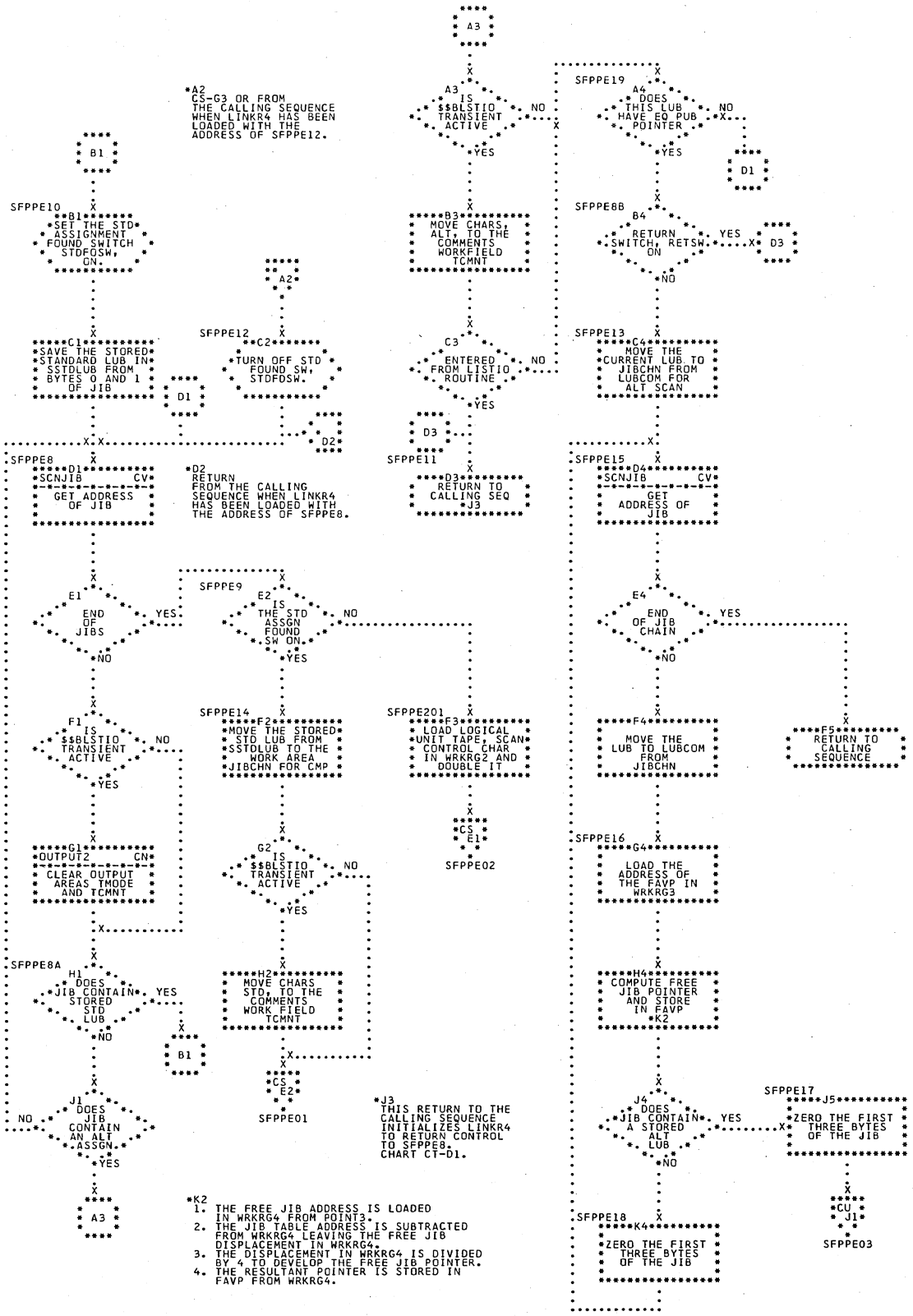


Chart CU. $JOBCTLD - Miscellaneous Subroutines (Part 4 of 5)
Refer to Chart 06.

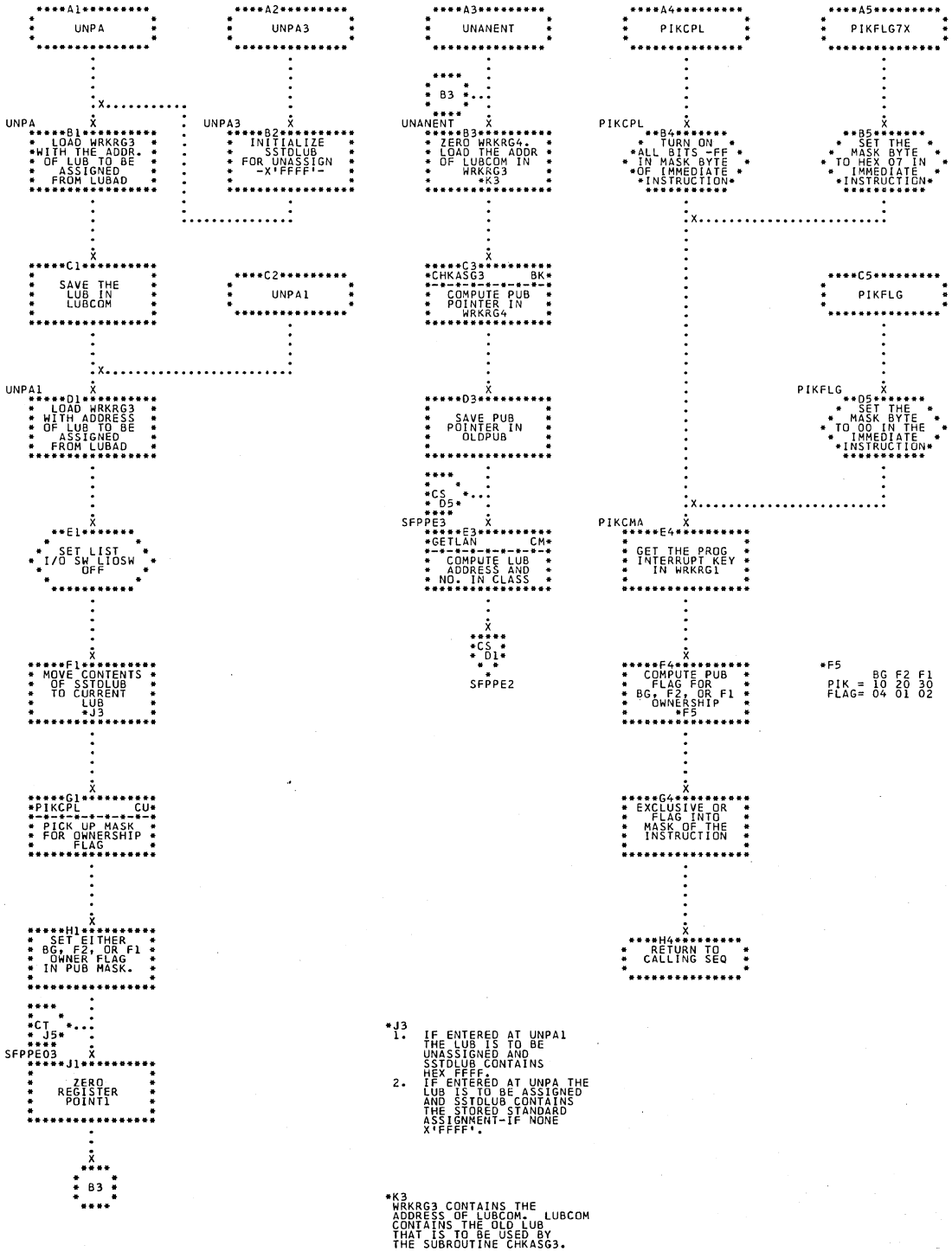


Chart CV. $JOBCTLD - Miscellaneous Subroutines (Part 5 of 5)
Refer to Chart 06.

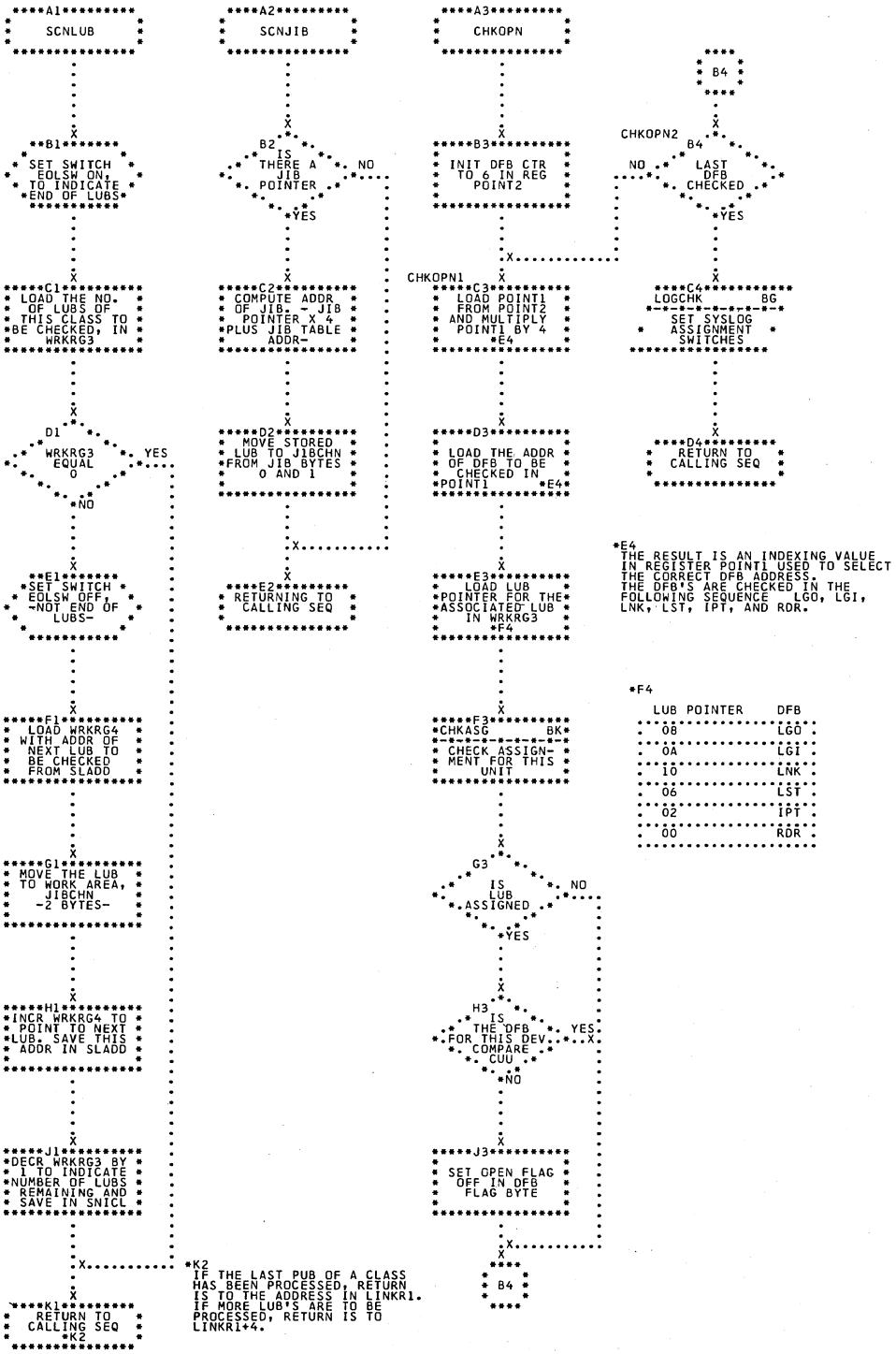


Chart CW. $JOBCTLD - SYSXXX Operand Processor
Refer to Chart 06.

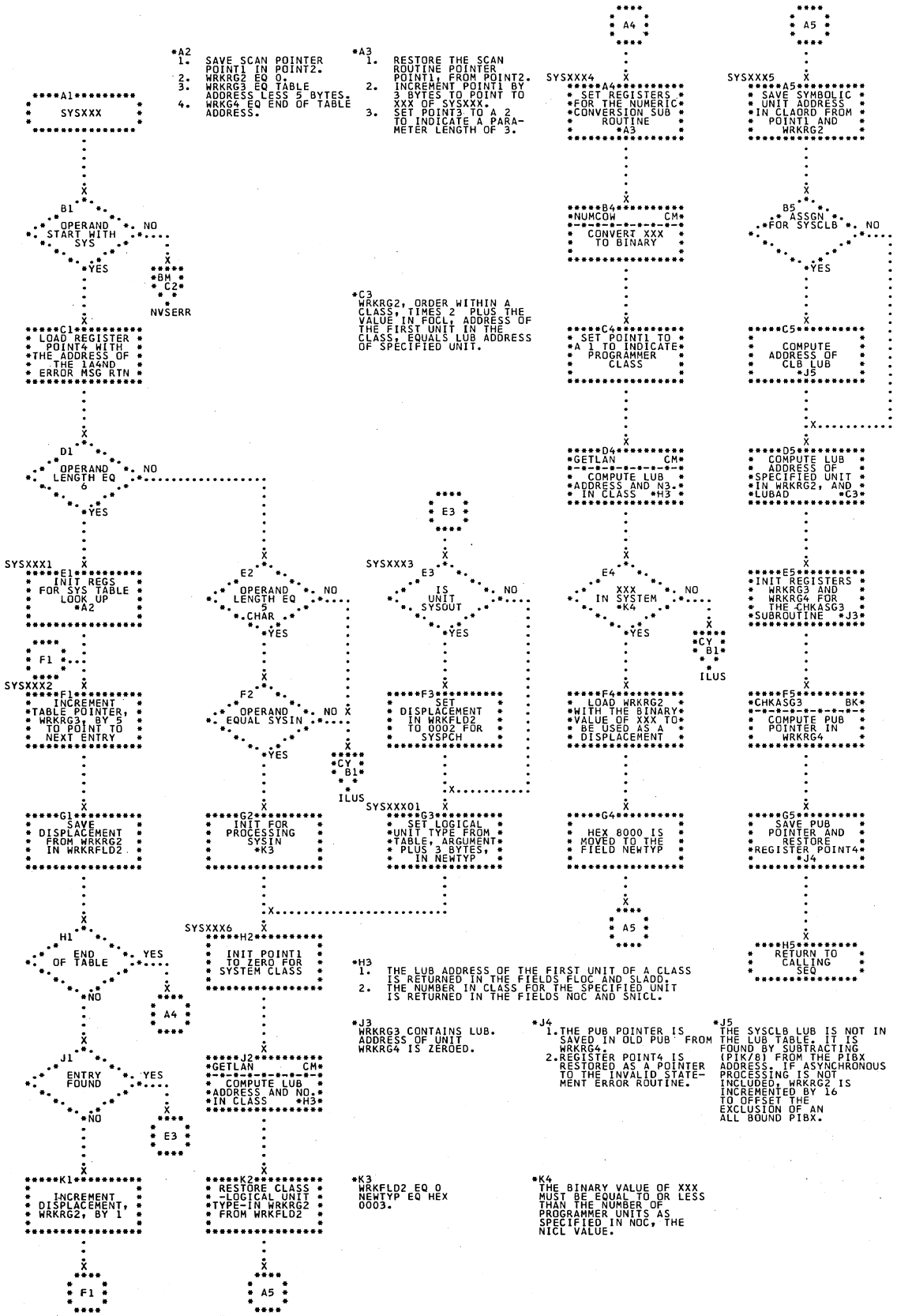


Chart CX. $JOBCTLD - EXCP Subroutines
Refer to Chart 06.

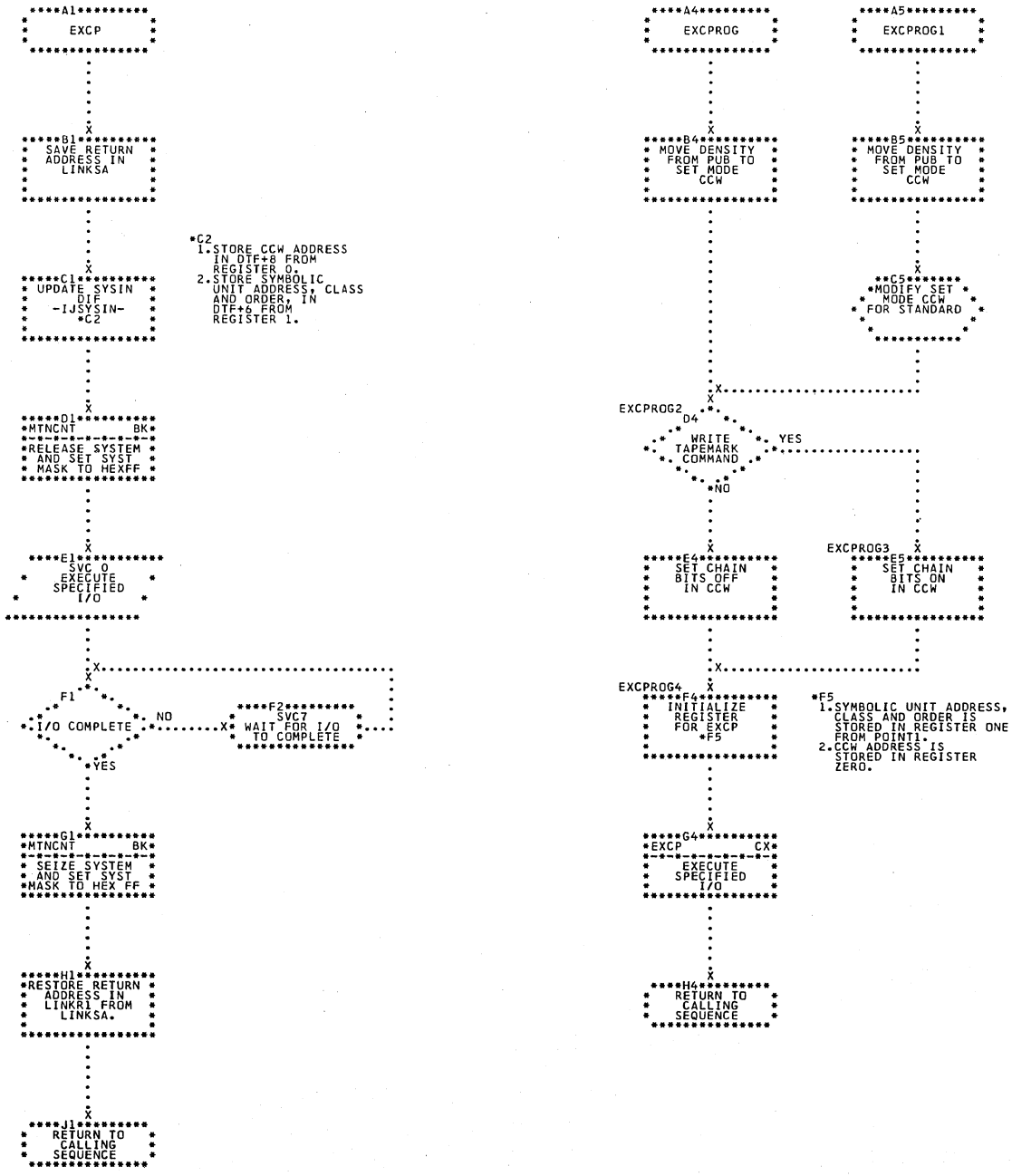
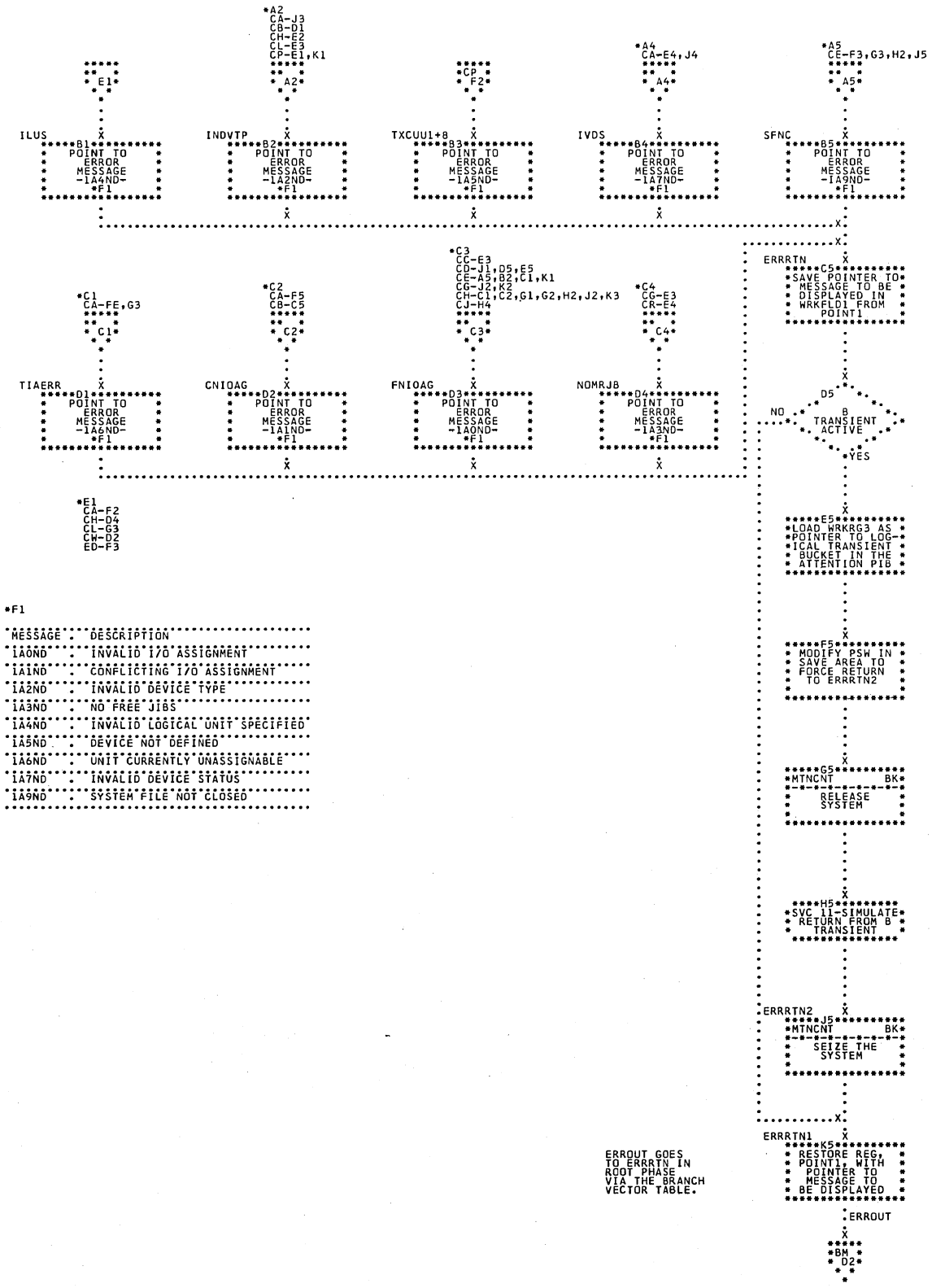


Chart CY. $JOBCTLD - Error Subroutines
Refer to Chart 06.



*F1

MESSAGE	DESCRIPTION
1AOND	INVALID I/O ASSIGNMENT
1A1ND	CONFLICTING I/O ASSIGNMENT
1A2ND	INVALID DEVICE TYPE
1A3ND	NO FREE JIBS
1A4ND	INVALID LOGICAL UNIT SPECIFIED
1A5ND	DEVICE NOT DEFINED
1A6ND	UNIT CURRENTLY UNASSIGNABLE
1A7ND	INVALID DEVICE STATUS
1A9ND	SYSTEM FILE NOT CLOSED

ERROUT GOES TO ERRRTN IN ROOT PHASE VIA THE BRANCH VECTOR TABLE.

Chart DA. $JOBCTLF - LISTIO Statement Processor (Part 1 of 5)
Refer to Chart 07.

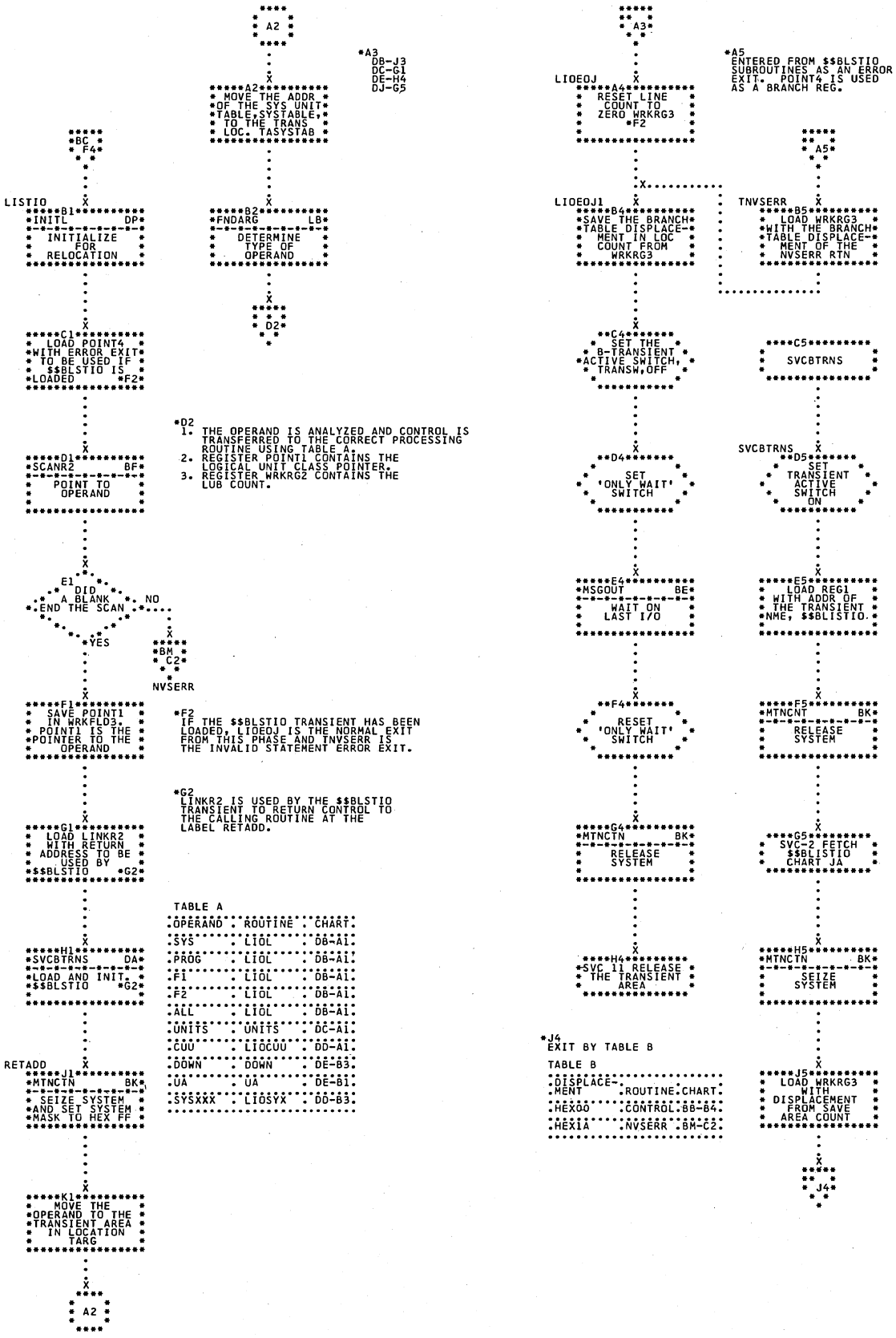


Chart DB. $JOBCTLF - LISTIO Statement Processor (Part 2 of 5)
Refer to Chart 07.

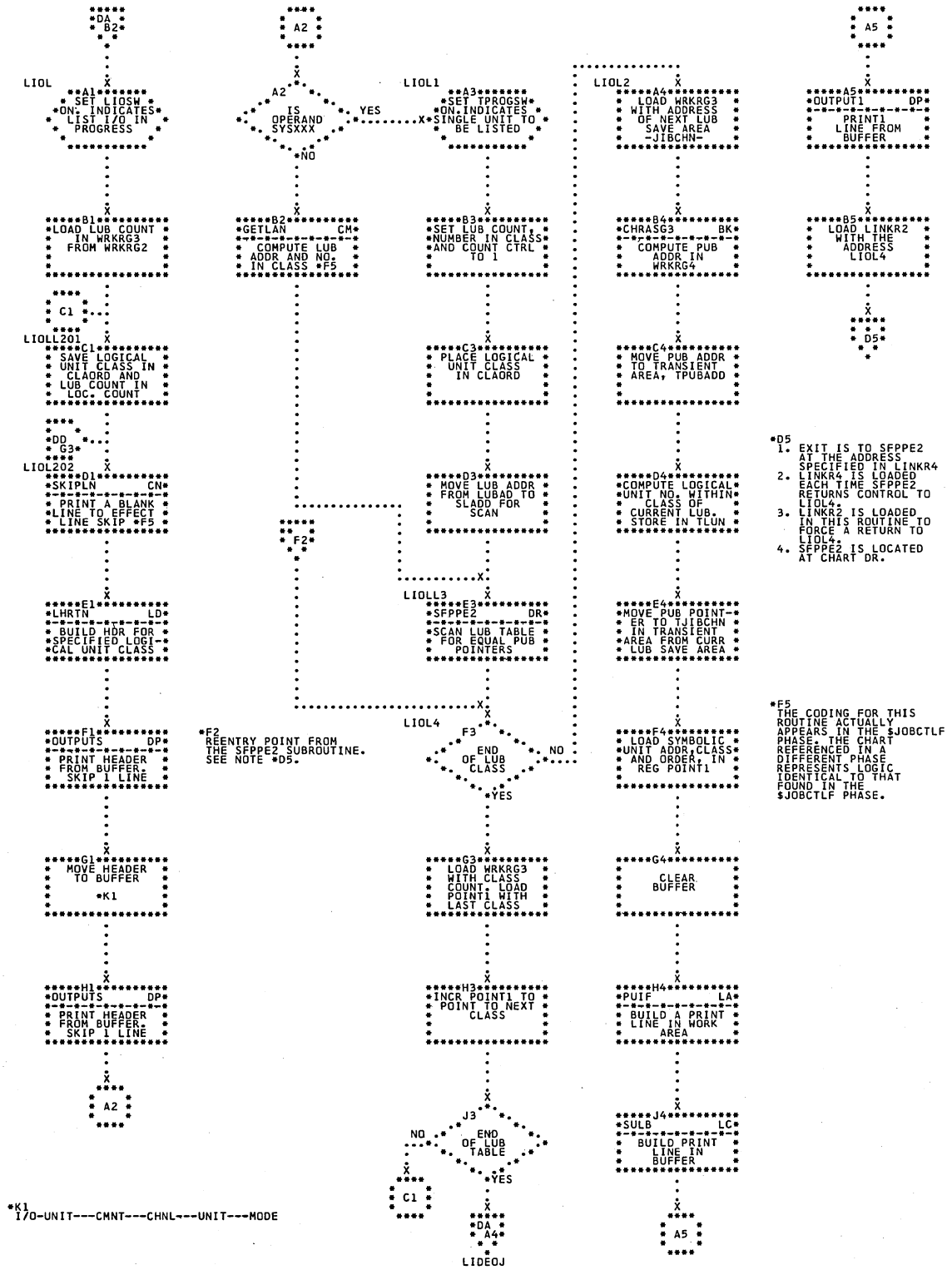


Chart DC. $JOBCTLF - LISTIO Statement Processor (Part 3 of 5)
Refer to Chart 07.

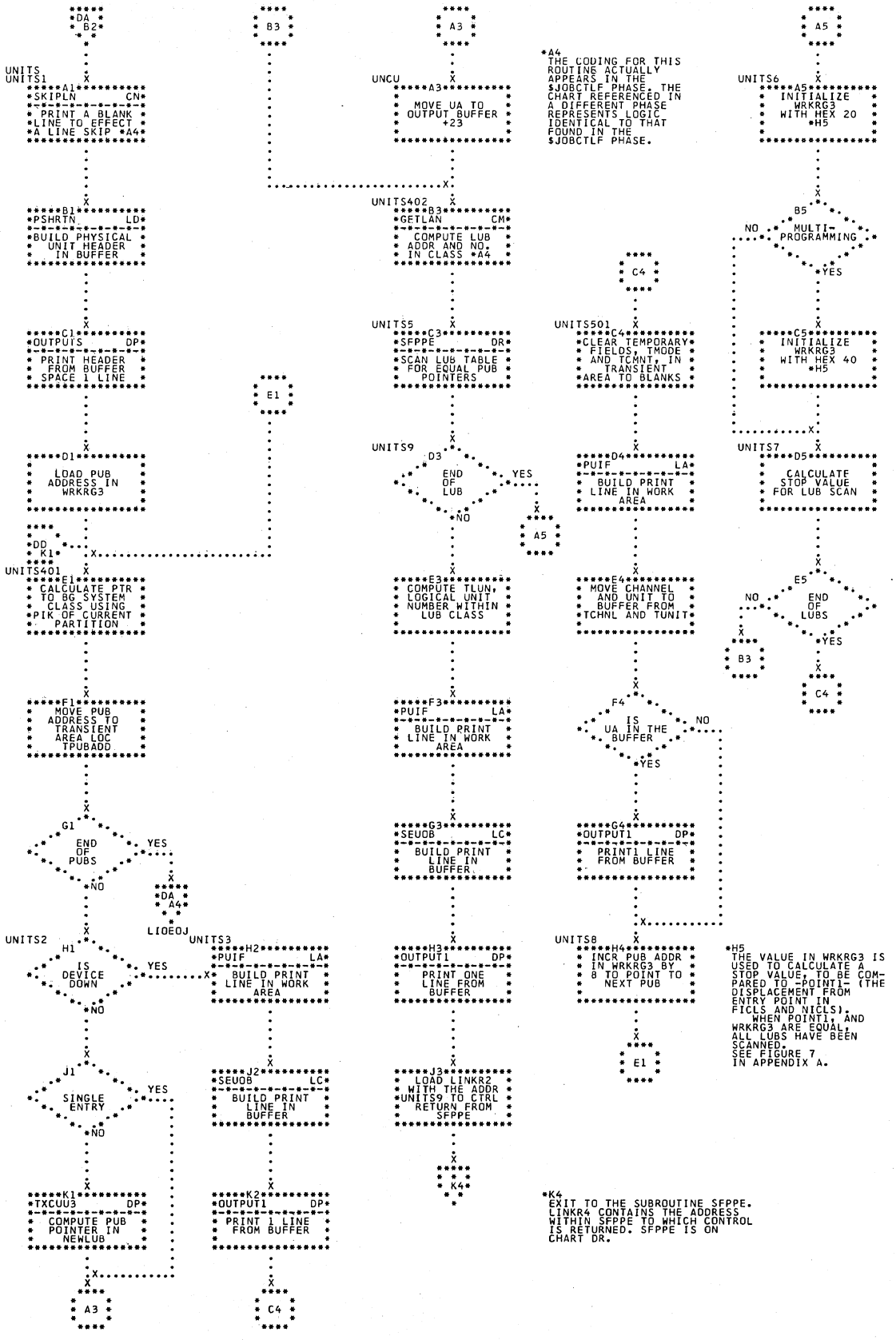


Chart DD. $JOBCTLF - LISTIO Statement Processor (Part 4 of 5)
Refer to Chart 07.

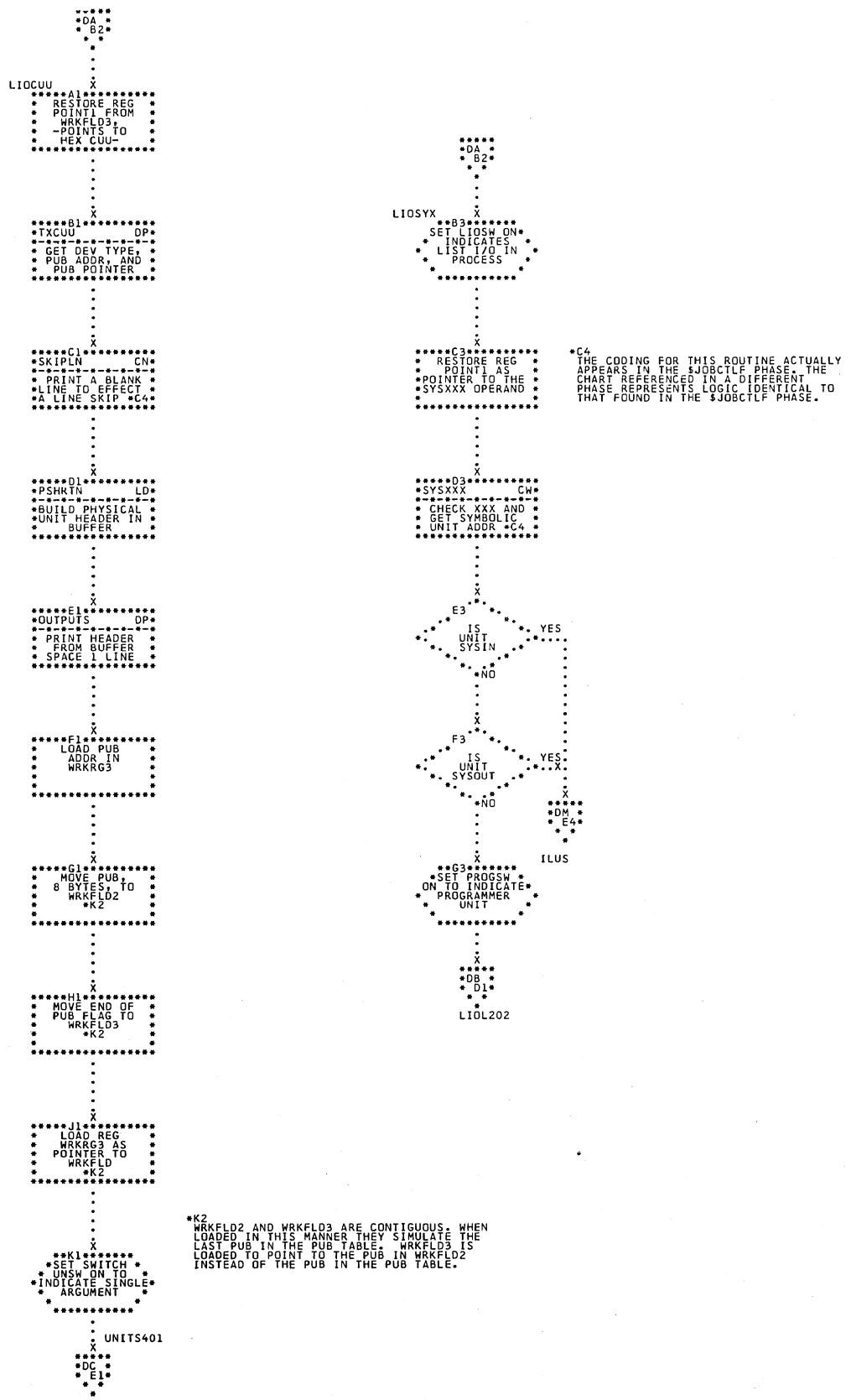


Chart DE. $JOBCTLF - LISTIO Statement Processor (Part 5 of 5)
Refer to Chart 07.

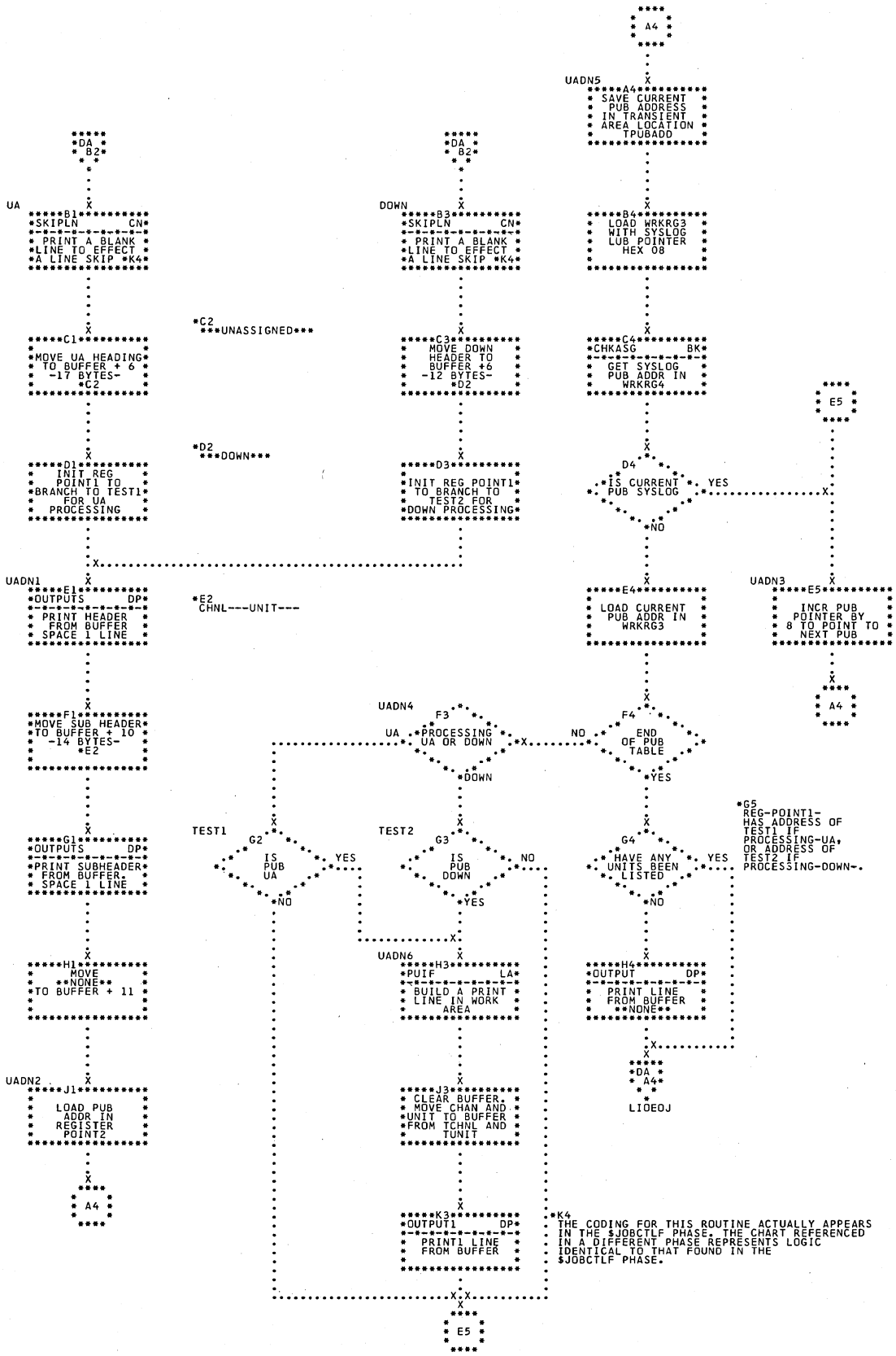


Chart DF. $JOBCTLF - RESET Statement Processor (Part 1 of 2)
Refer to Chart 07.

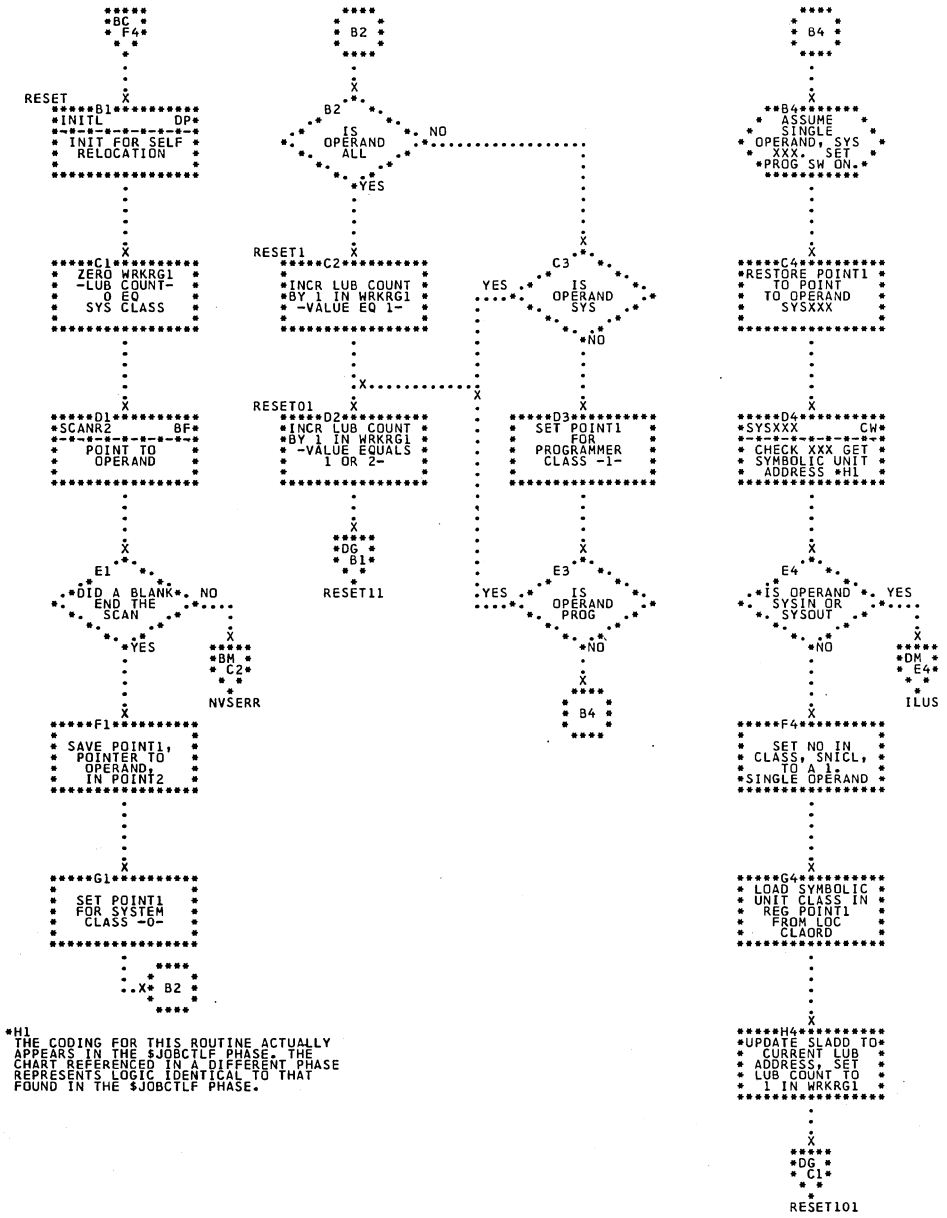


Chart DG. $JOBCTLF - RESET Statement Processor (Part 2 of 2)
Refer to Chart 07.

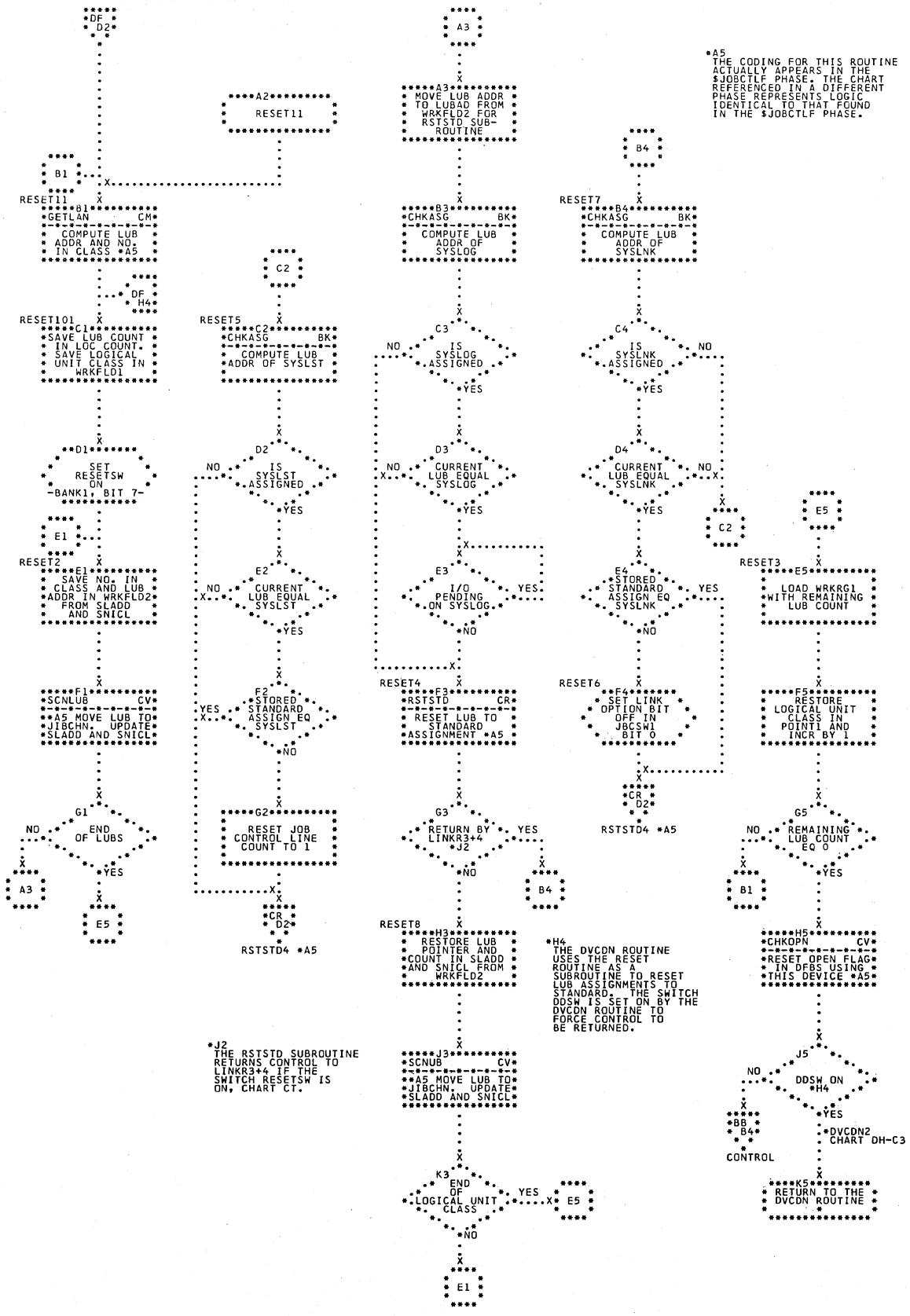


Chart DH. $JOBCTLF - DVCDN Statement Processor (Part 1 of 2)
Refer to Chart 07.

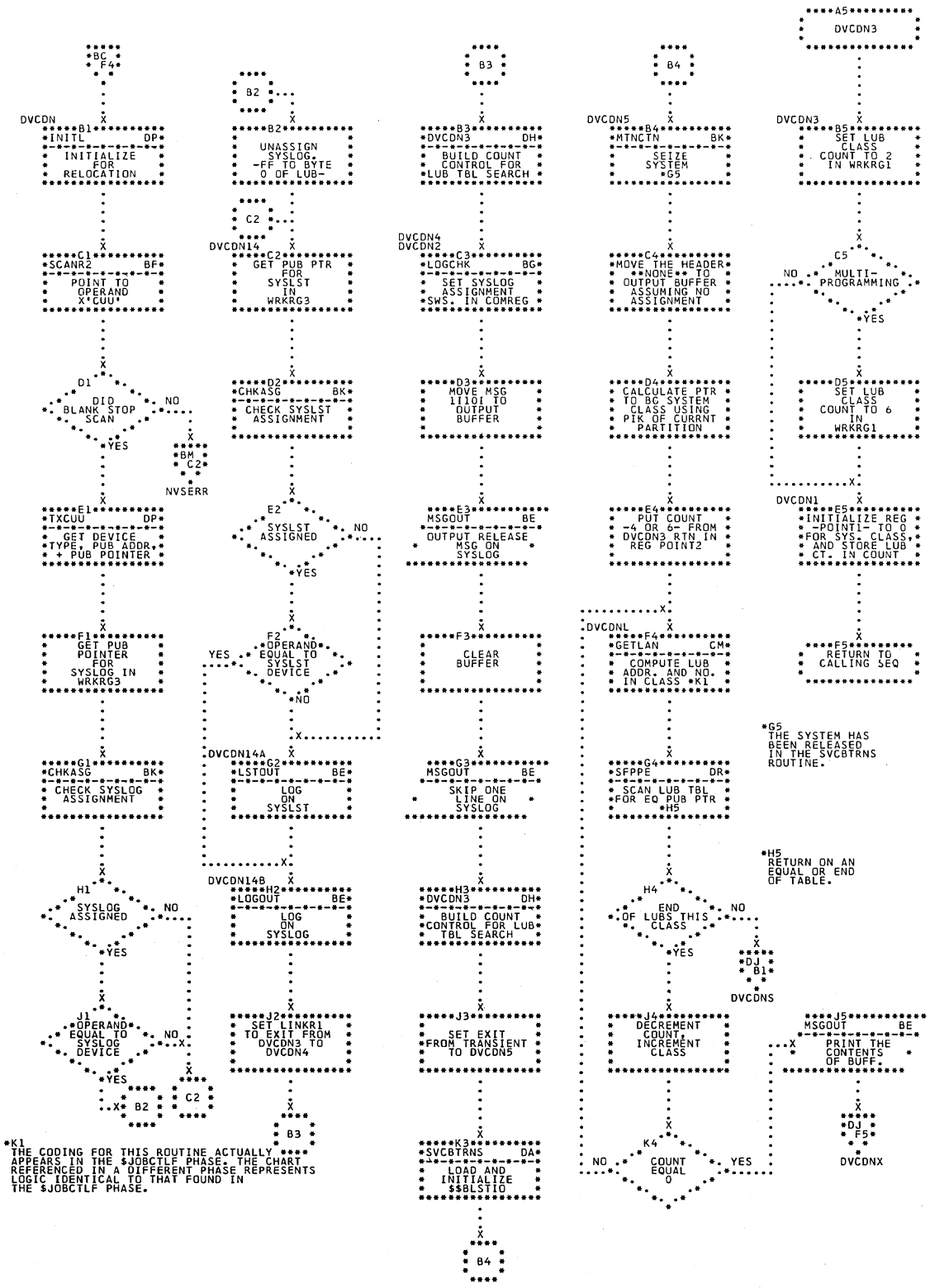


Chart DJ. $JOBCTLF - DVCDN Statement Processor (Part 2 of 2).
Refer to Chart 07.

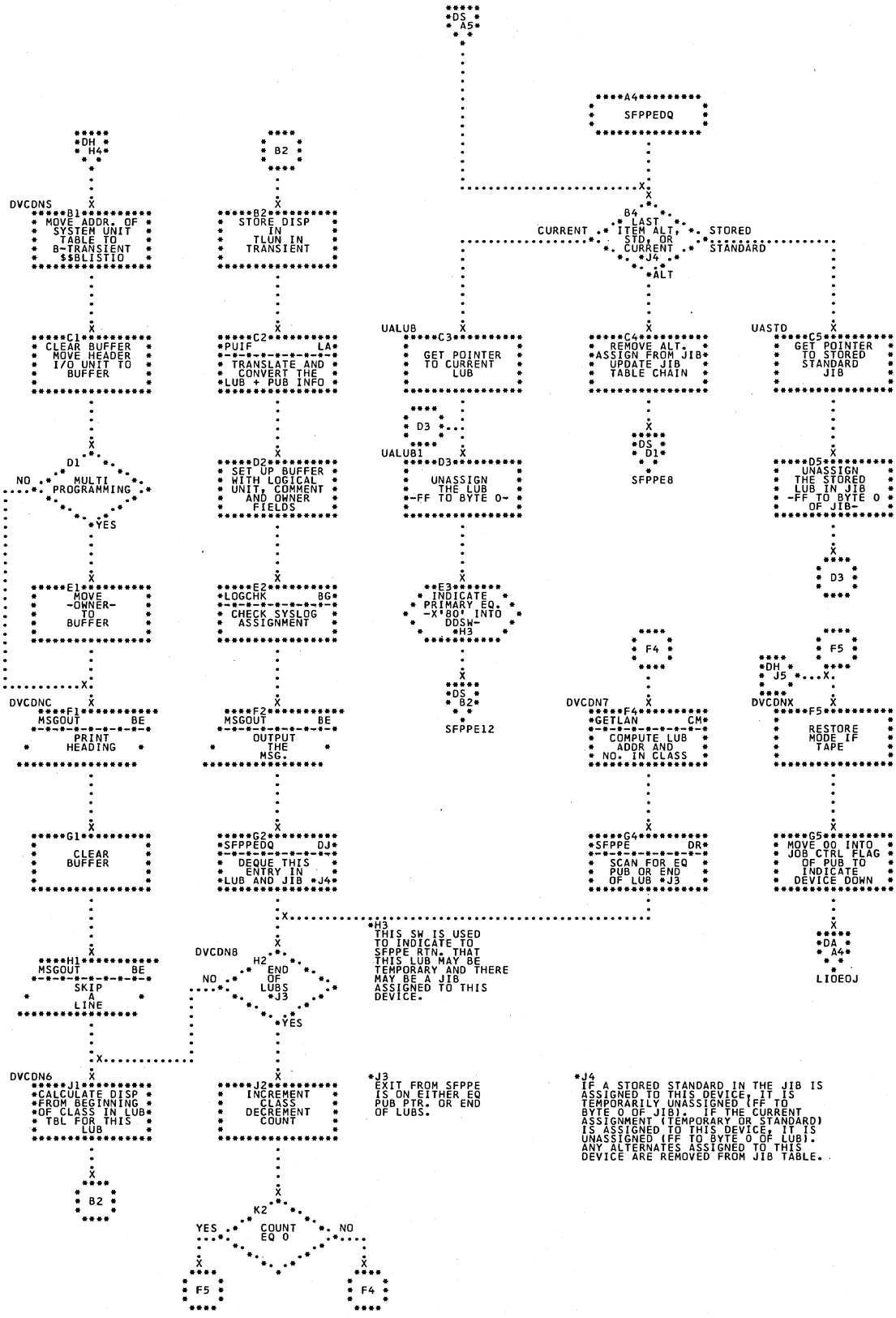


Chart DK. $JOBCTLF - DVCUP and MAP Statement Processors
Refer to Chart 07.

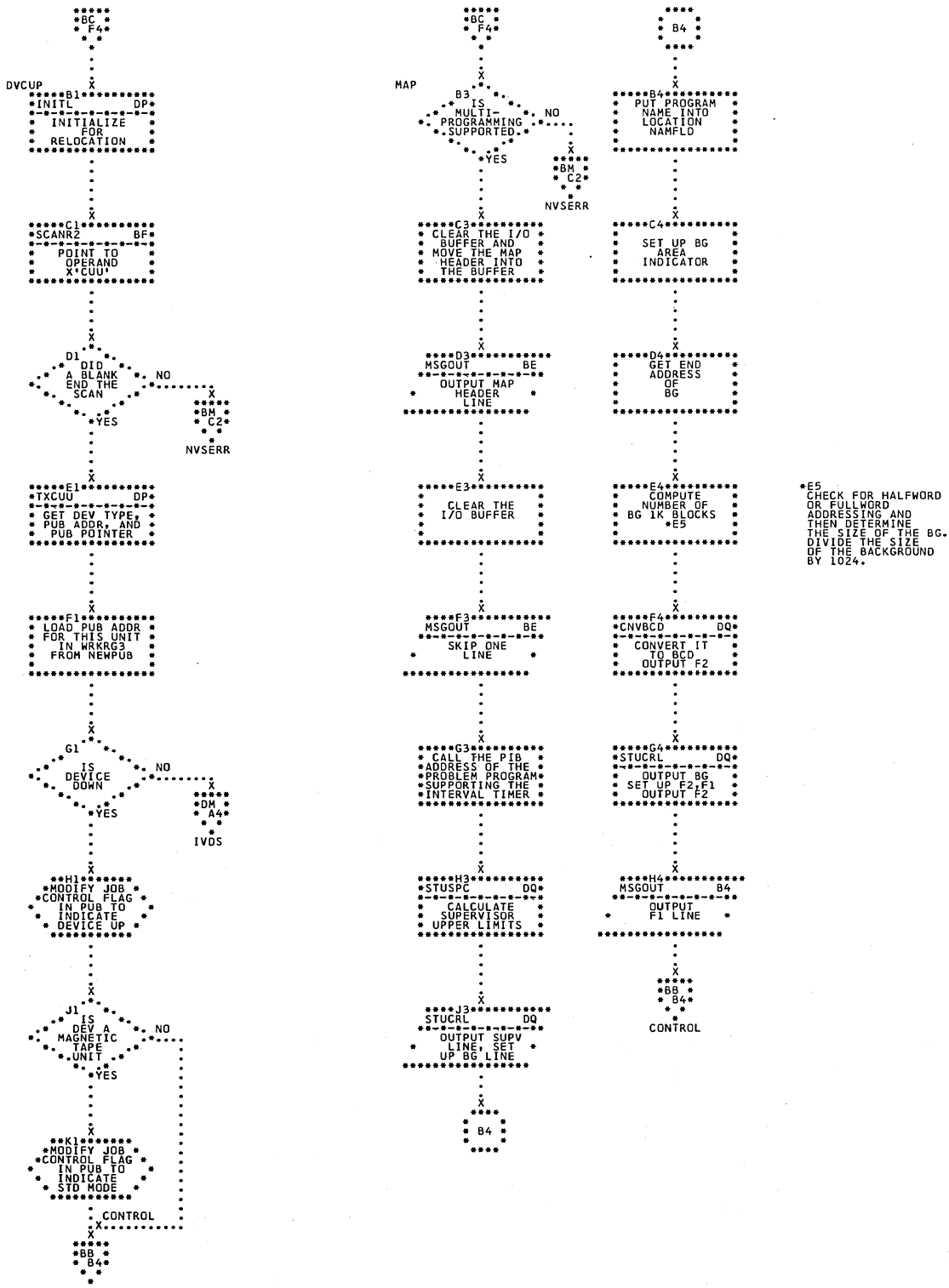


Chart DL. $JOBCTLF - UNBATCH Statement Processor
Refer to Chart 07.

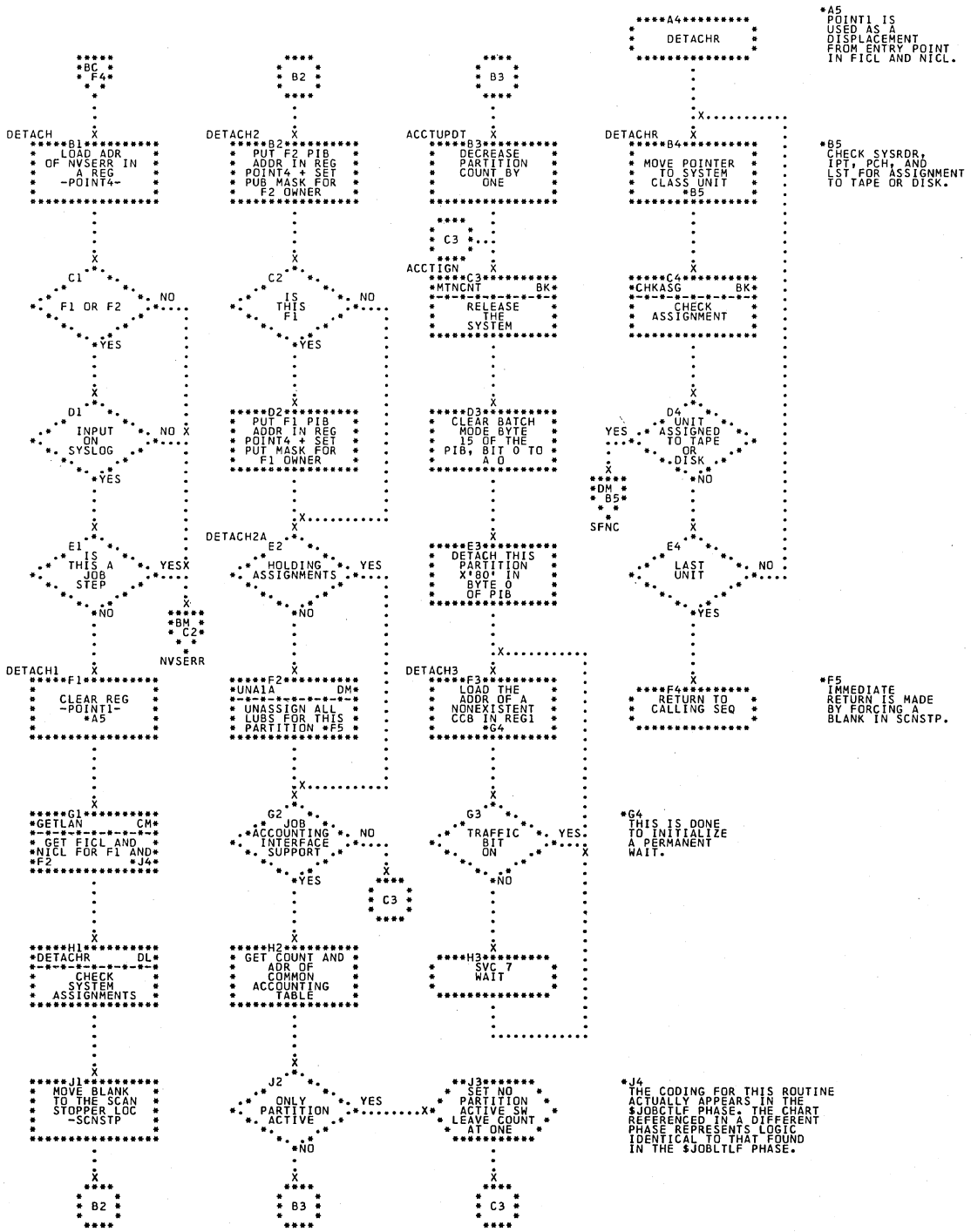


Chart DM. $JOBCTLF - Error Subroutines and UNA Statement Processor (Part 1 of 2)
Refer to Chart 07.

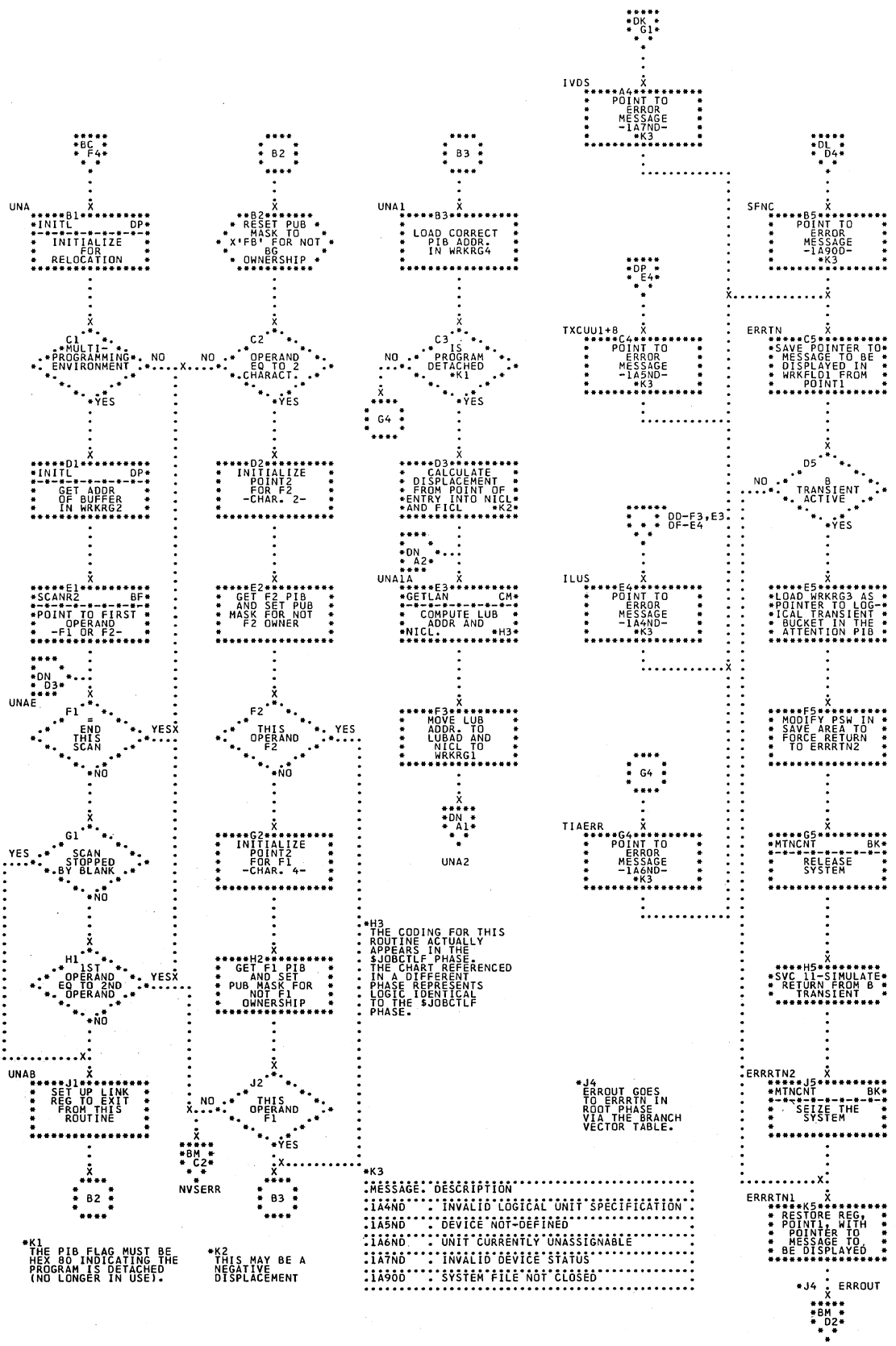


Chart DN. $JOBCTLF - UNA Statement Processor (Part 2 of 2)
Refer to Chart 07.

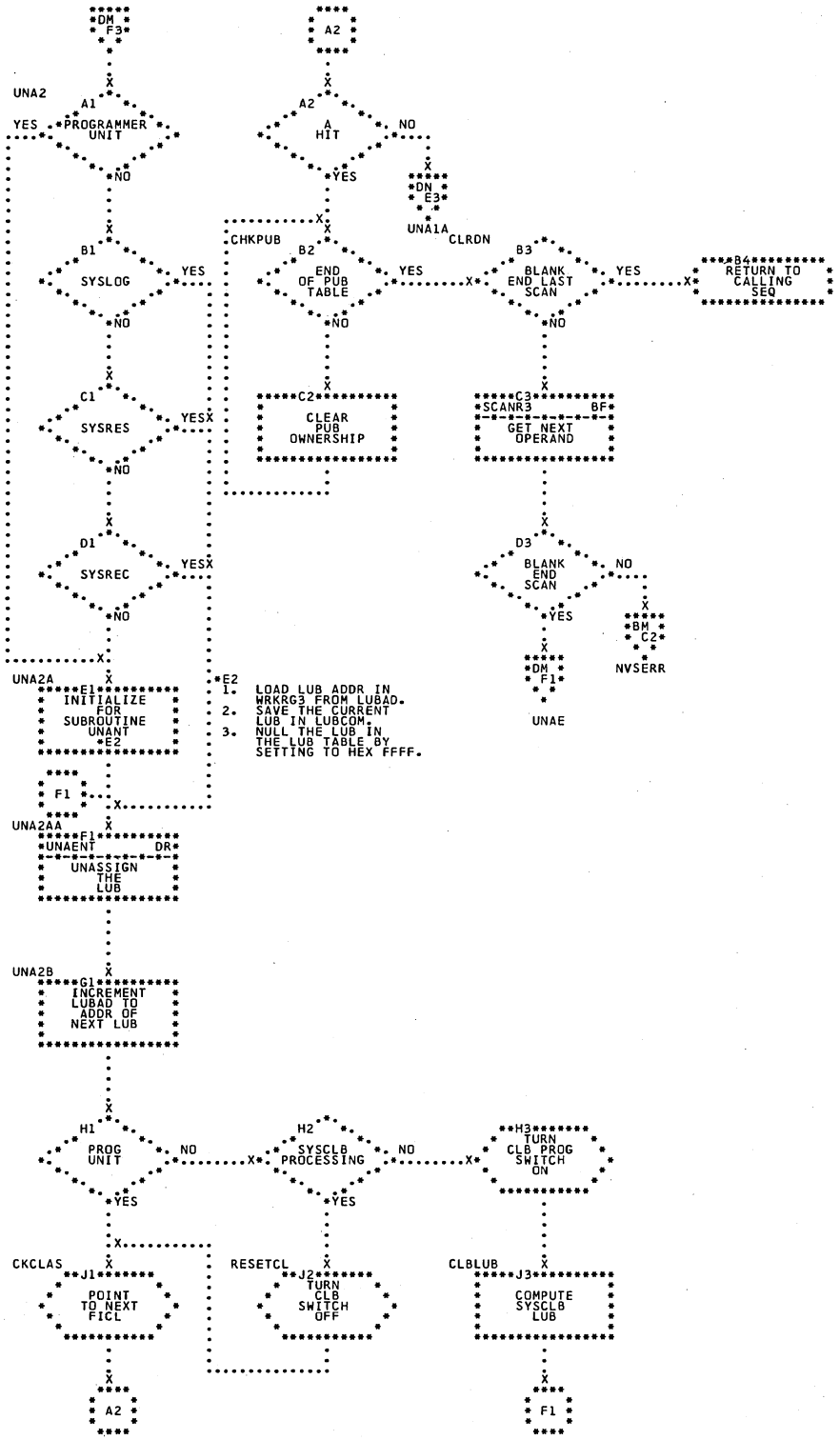


Chart DP. $JOBCTLF - Miscellaneous Subroutines (Part 1 of 4)
Refer to Chart 07.

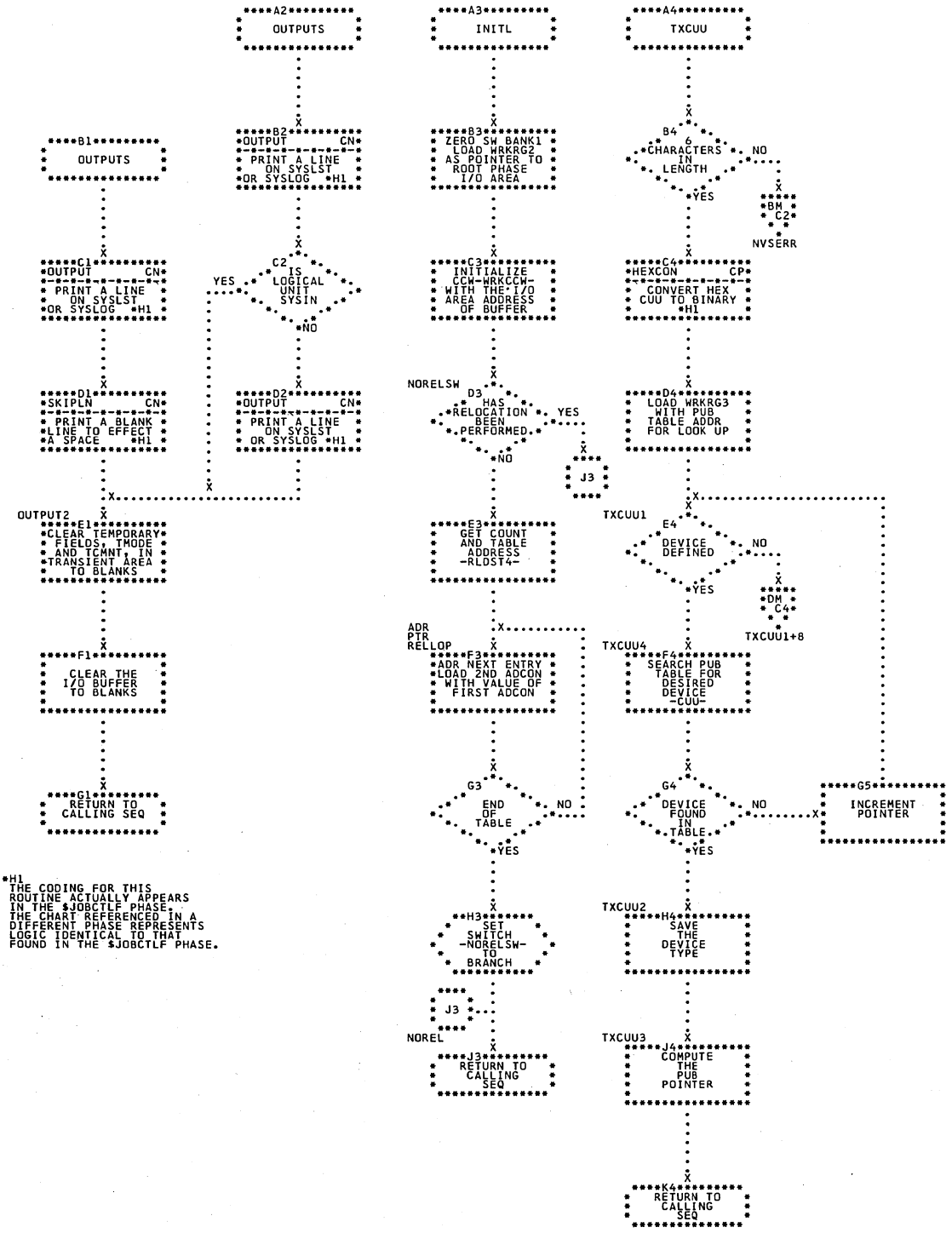
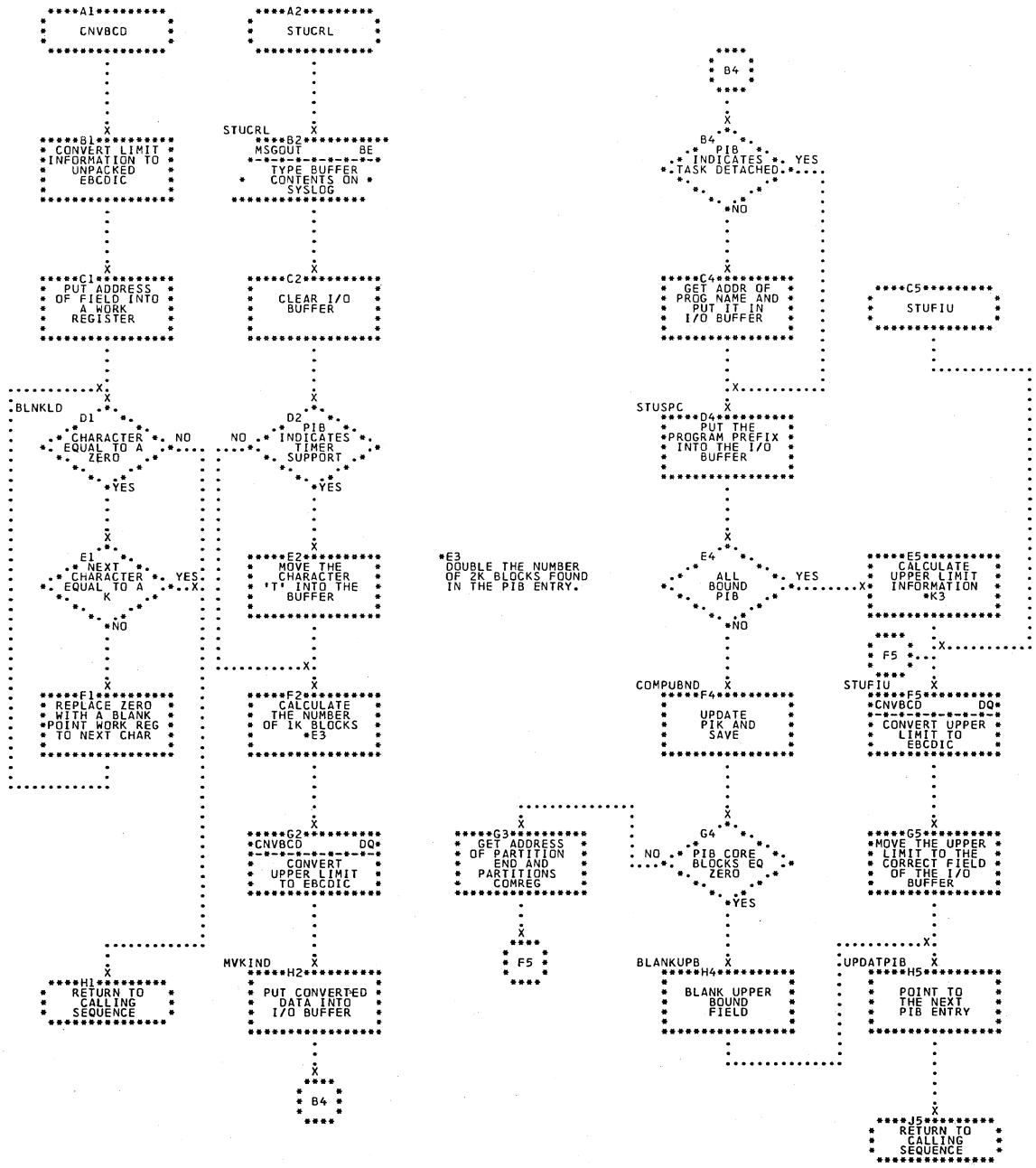


Chart DQ. $JOBCTLF - Miscellaneous Subroutines (Part 2 of 4)
Refer to Chart 07.



*K3
THE SUPERVISOR AND BG PROGRAM UPPER LIMITS ARE CALCULATED BY GETTING THE BG AND F2 ORIGIN ADDRESSES RESPECTIVELY AND DECREASING THEIR VALUE BY ONE.

Chart DR. $JOBCTLF - Miscellaneous Subroutines (Part 3 of 4)
Refer to Chart 07.

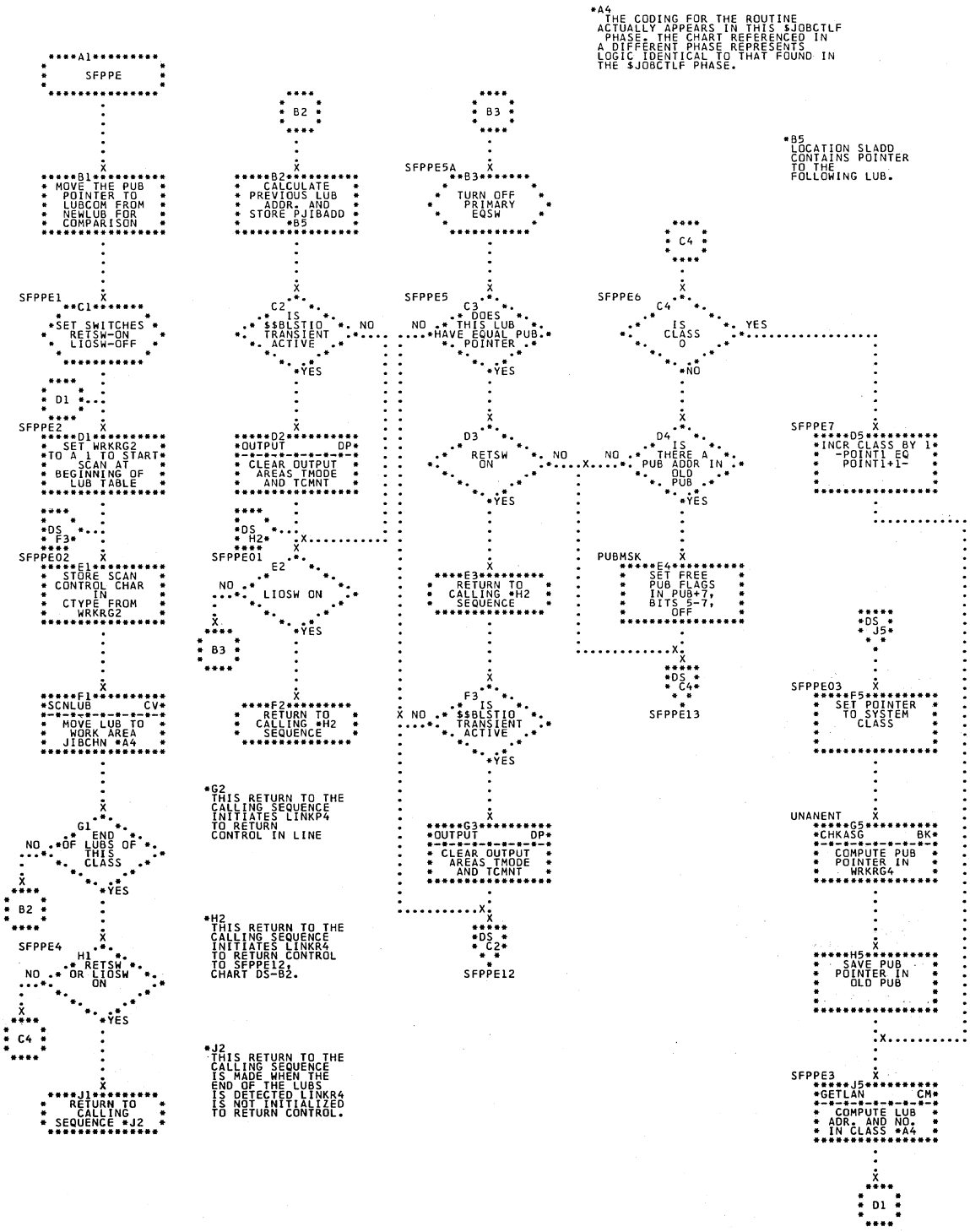


Chart DS. $JOBCTLF - Miscellaneous Subroutines (Part 4 of 4)
Refer to Chart 07.

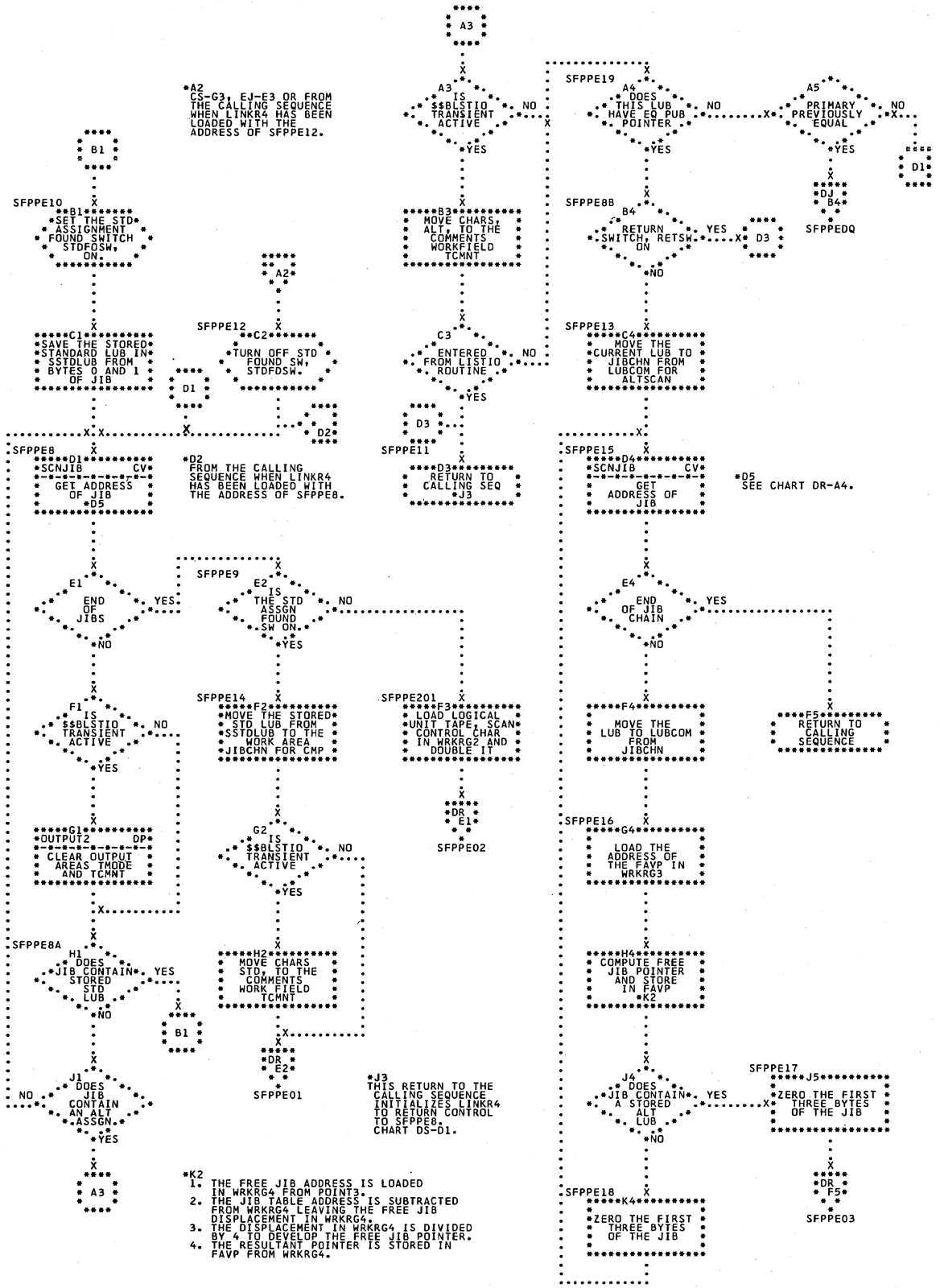


Chart EA. $JOBCTLG - JOB Statement Processor (Part 1 of 2)
Refer to Chart 09.

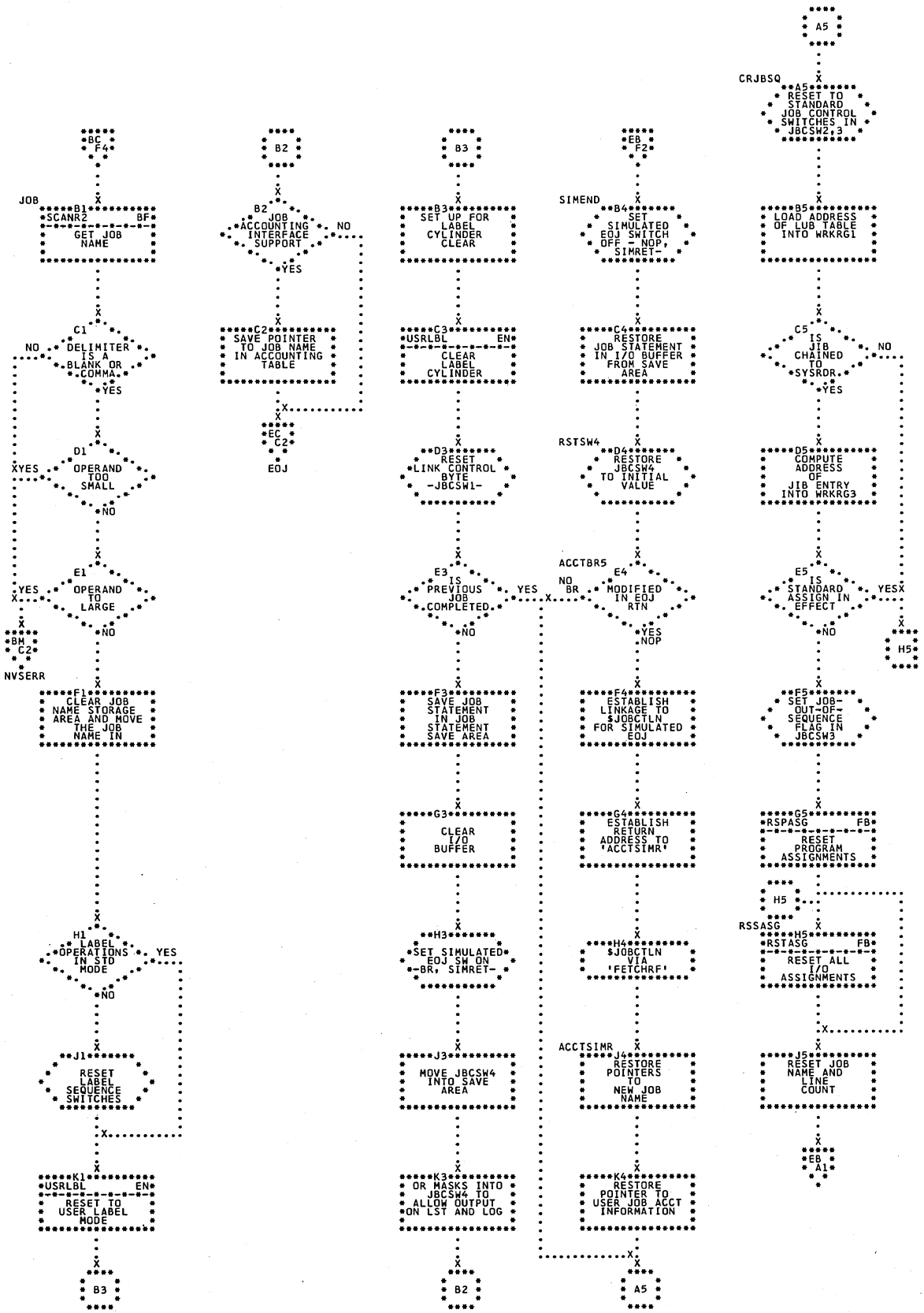


Chart EB. $JOBCTLG - JOB Statement Processor (Part 2 of 2)
Refer to Chart 09.

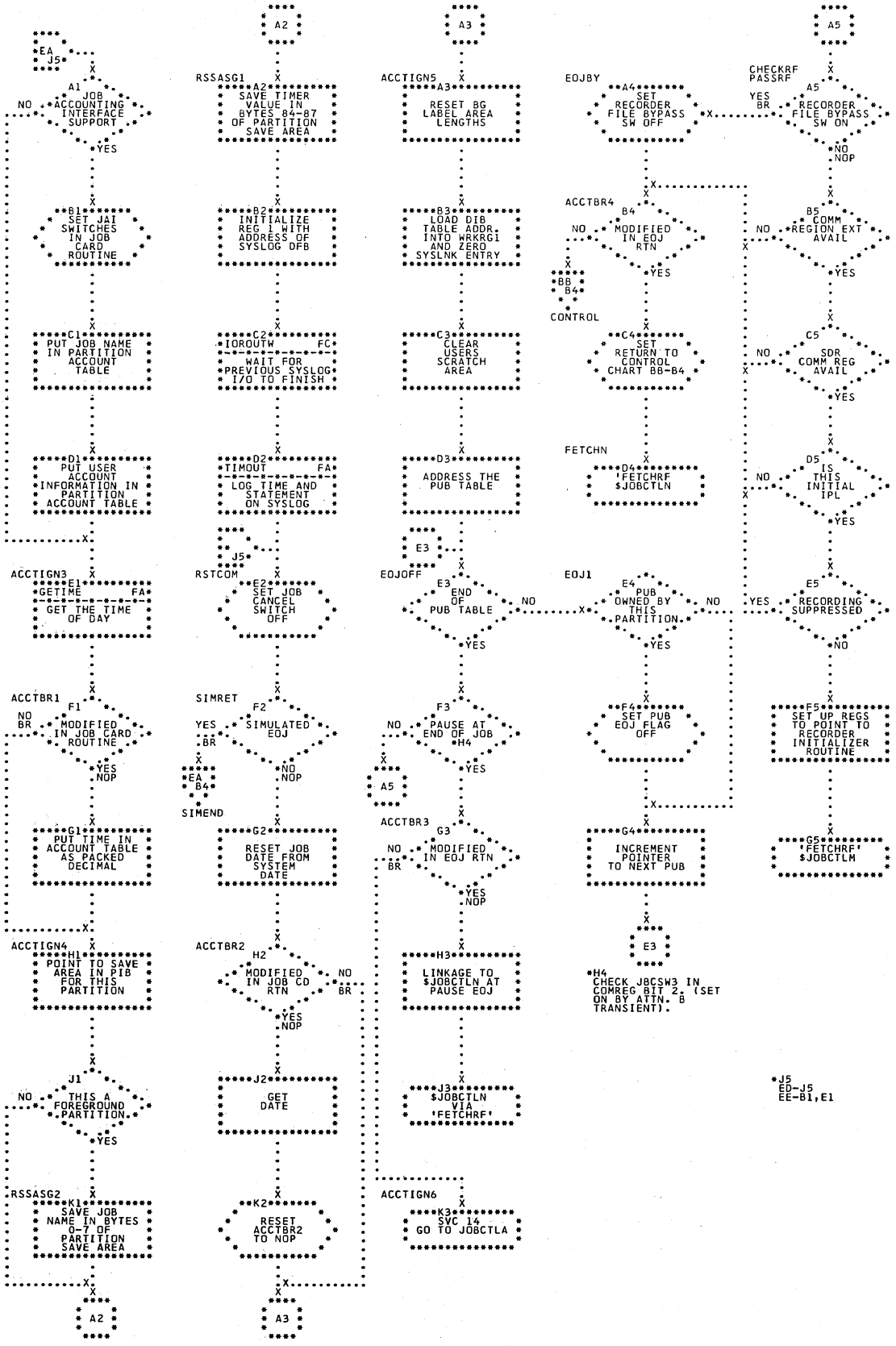


Chart EC. $JOBCTLG - /& Statement Processor (Part 1 of 3)
Refer to Chart 09.

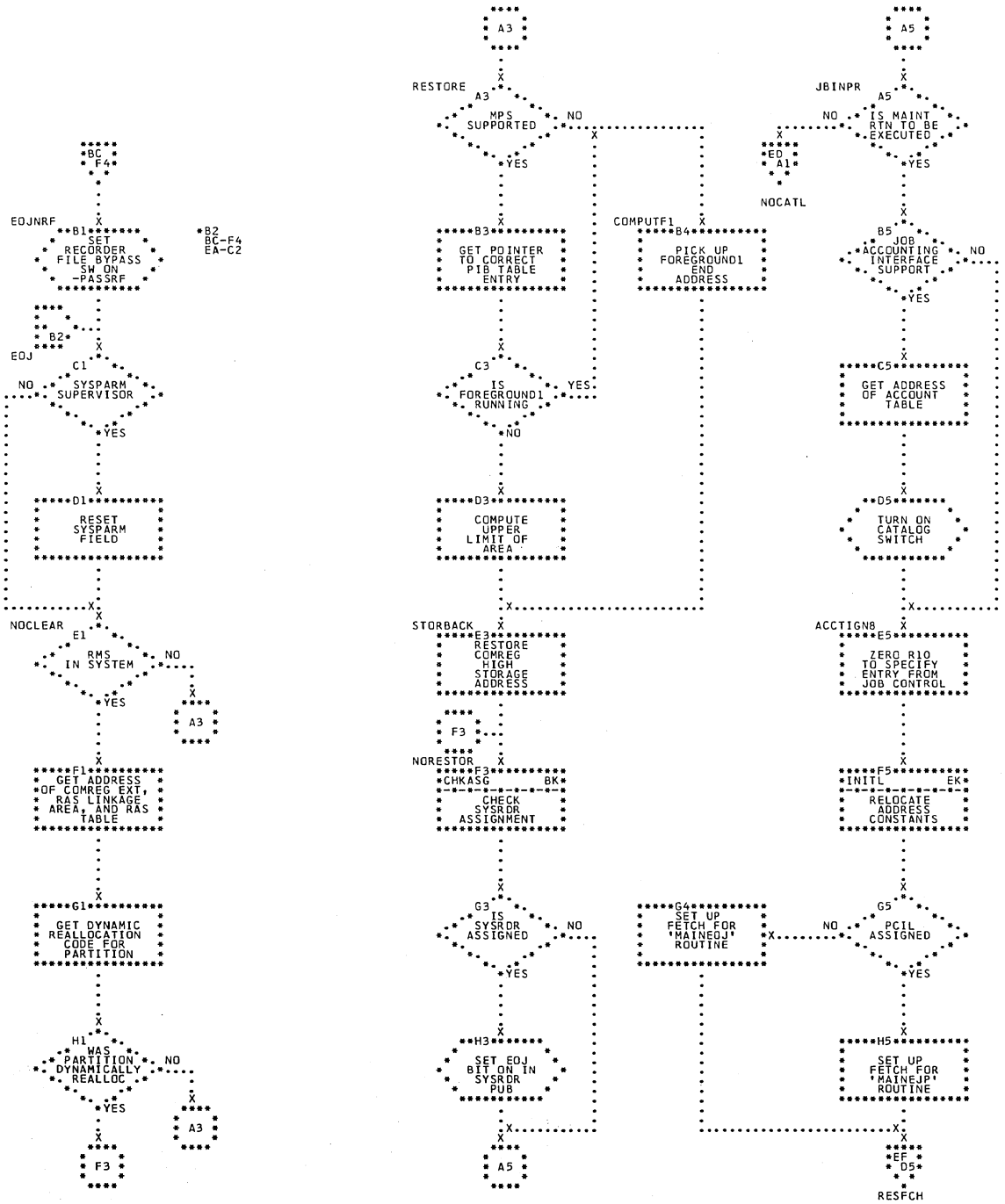


Chart ED. $JOBCTLG - /& Statement Processor (Part 2 of 3)
Refer to Chart 09.

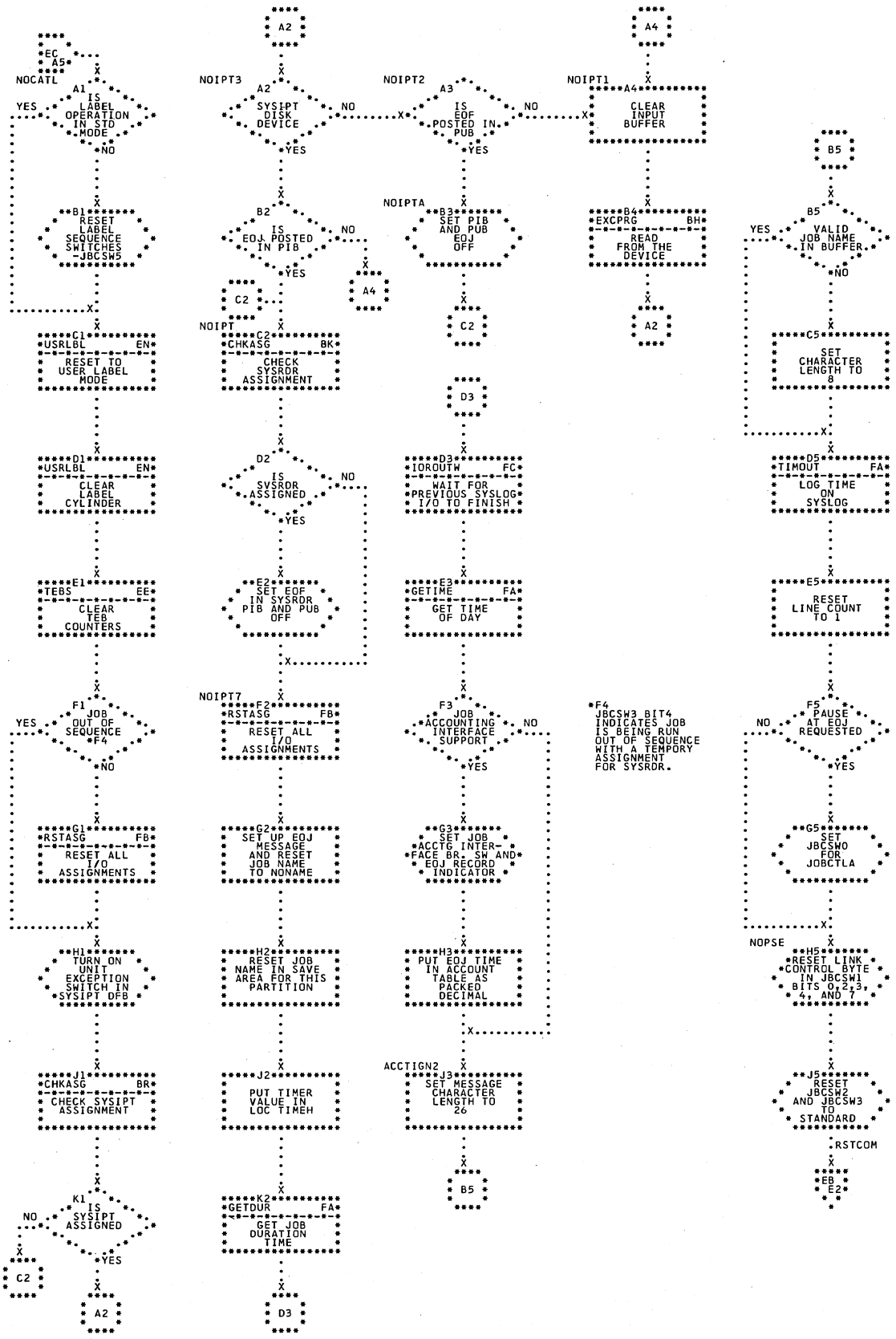


Chart EE. $JOBCTLG - /& Statement Processor (Part 3 of 3) and CANCEL Statement Processor
Refer to Chart 09.

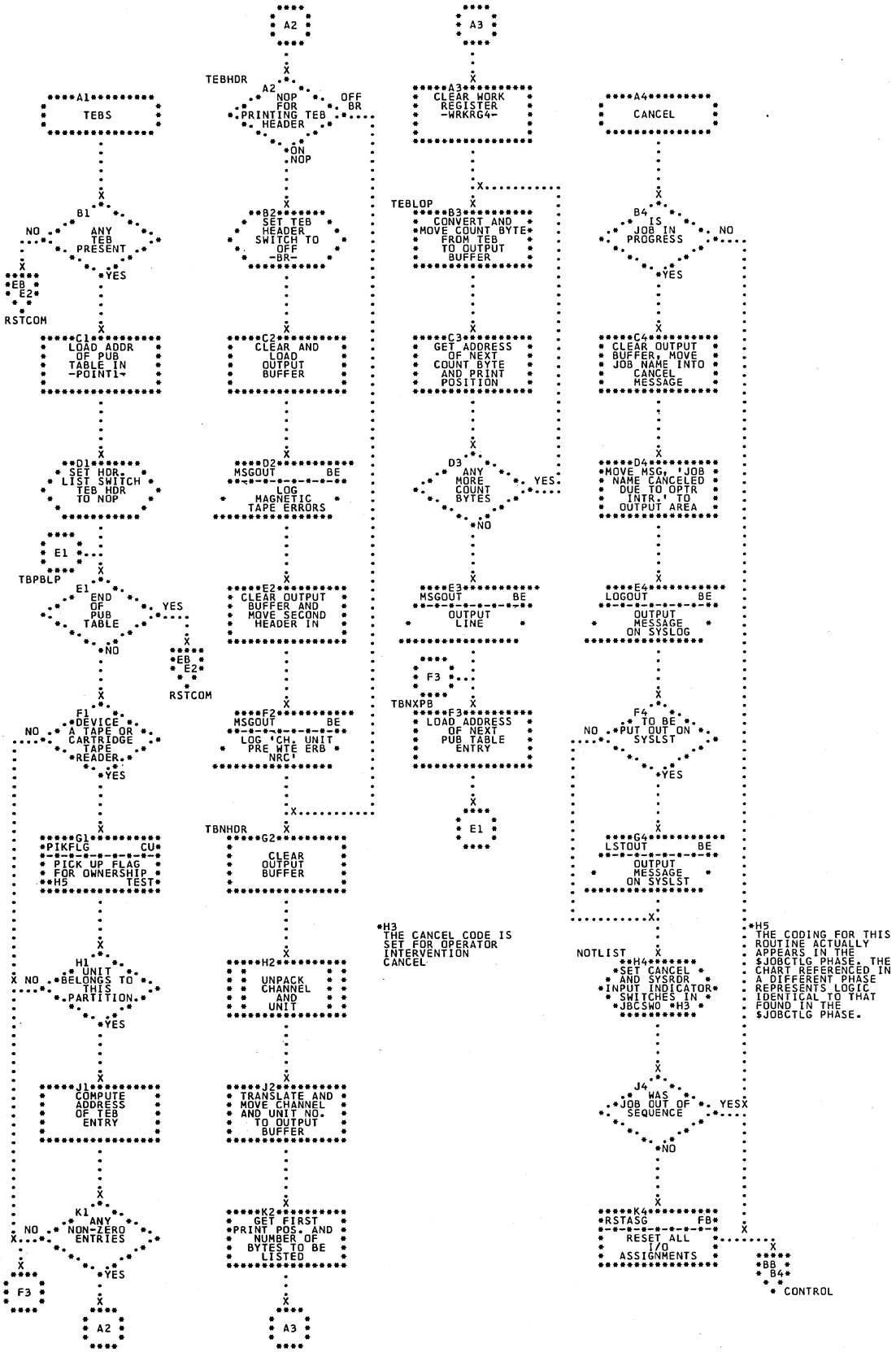


Chart EF. $JOBCTLG - EXEC Statement Processor (Part 1 of 5)
Refer to Chart 09.

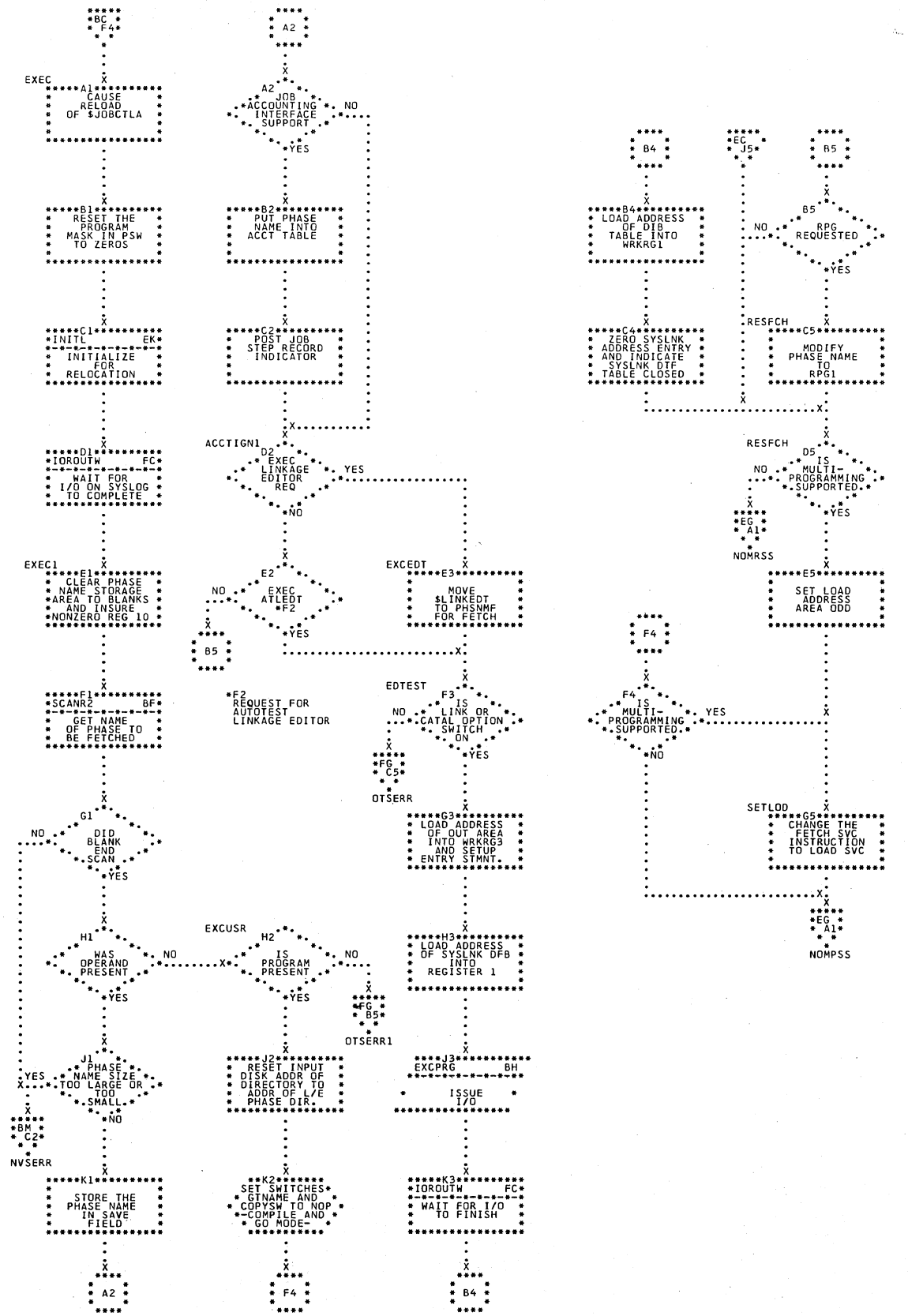


Chart EG. $JOBCTLG - EXEC Statement Processor (Part 2 of 5)
Refer to Chart 09.

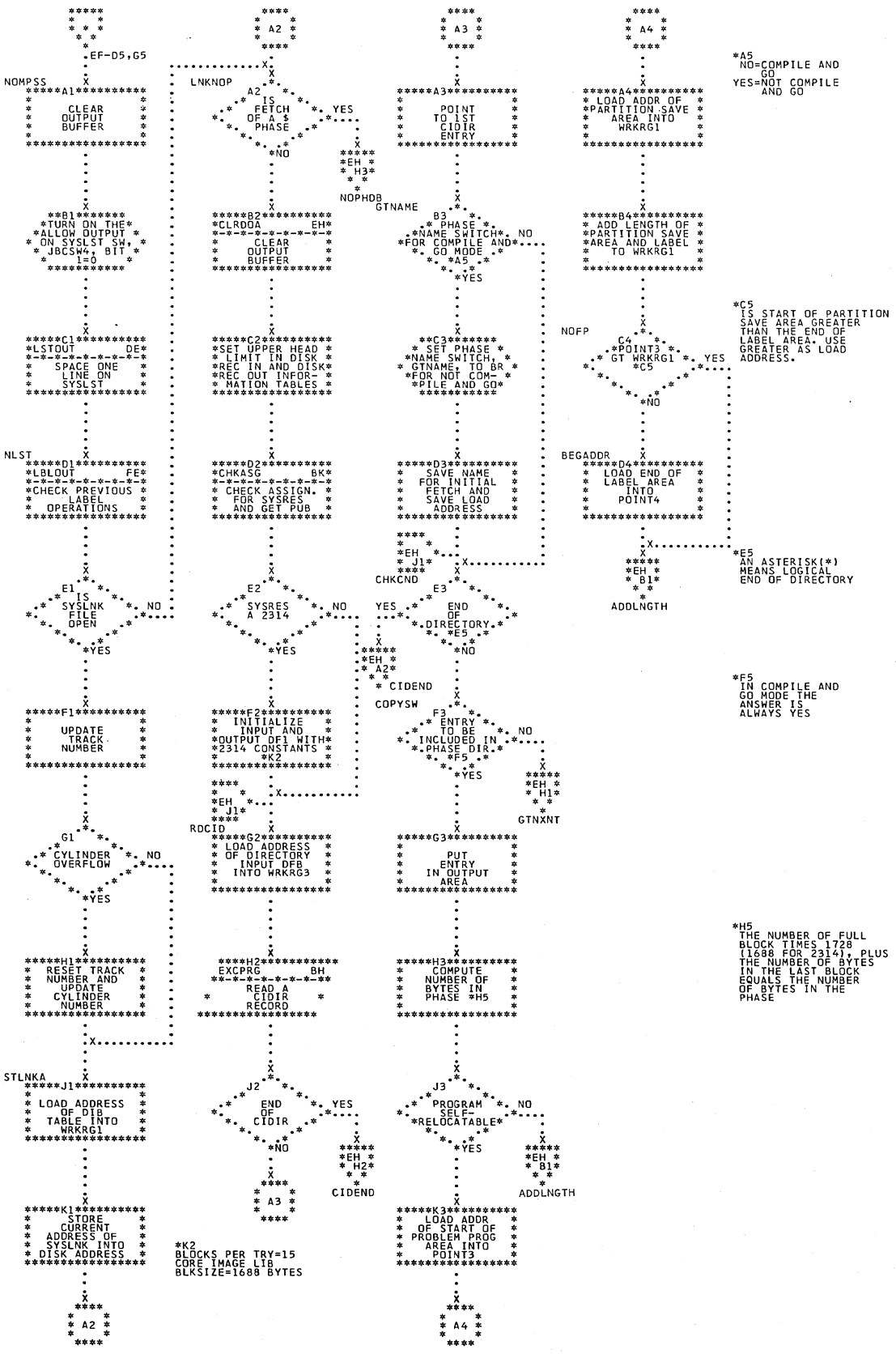


Chart EH. $JOBCTLG - EXEC Statement Processor (Part 3 of 5)
Refer to Chart 09.

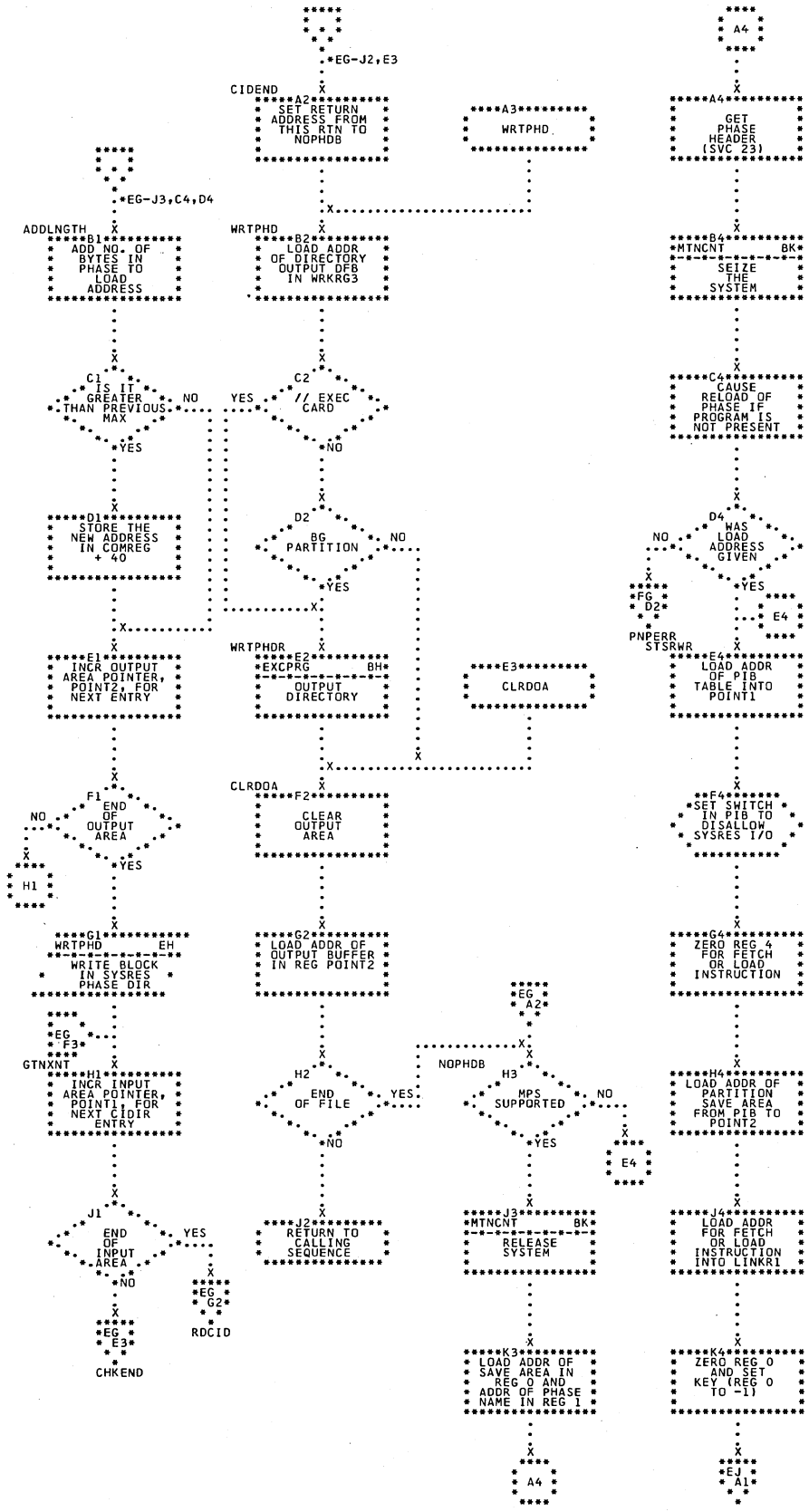


Chart EJ. $JOBCTLG - EXEC Statement Processor (Part 4 of 5)
Refer to Chart 09.

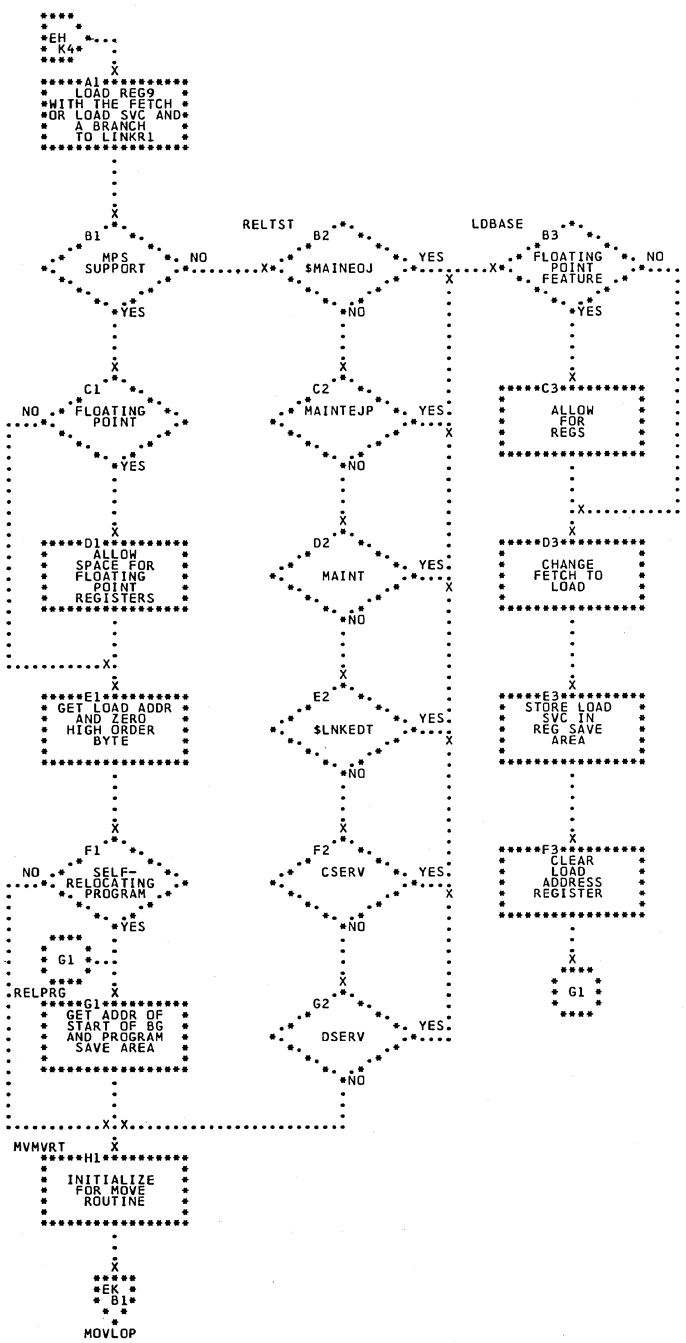


Chart EK. $JOBCTLG - EXEC Statement Processor (Part 5 of 5)
Refer to Chart 09.

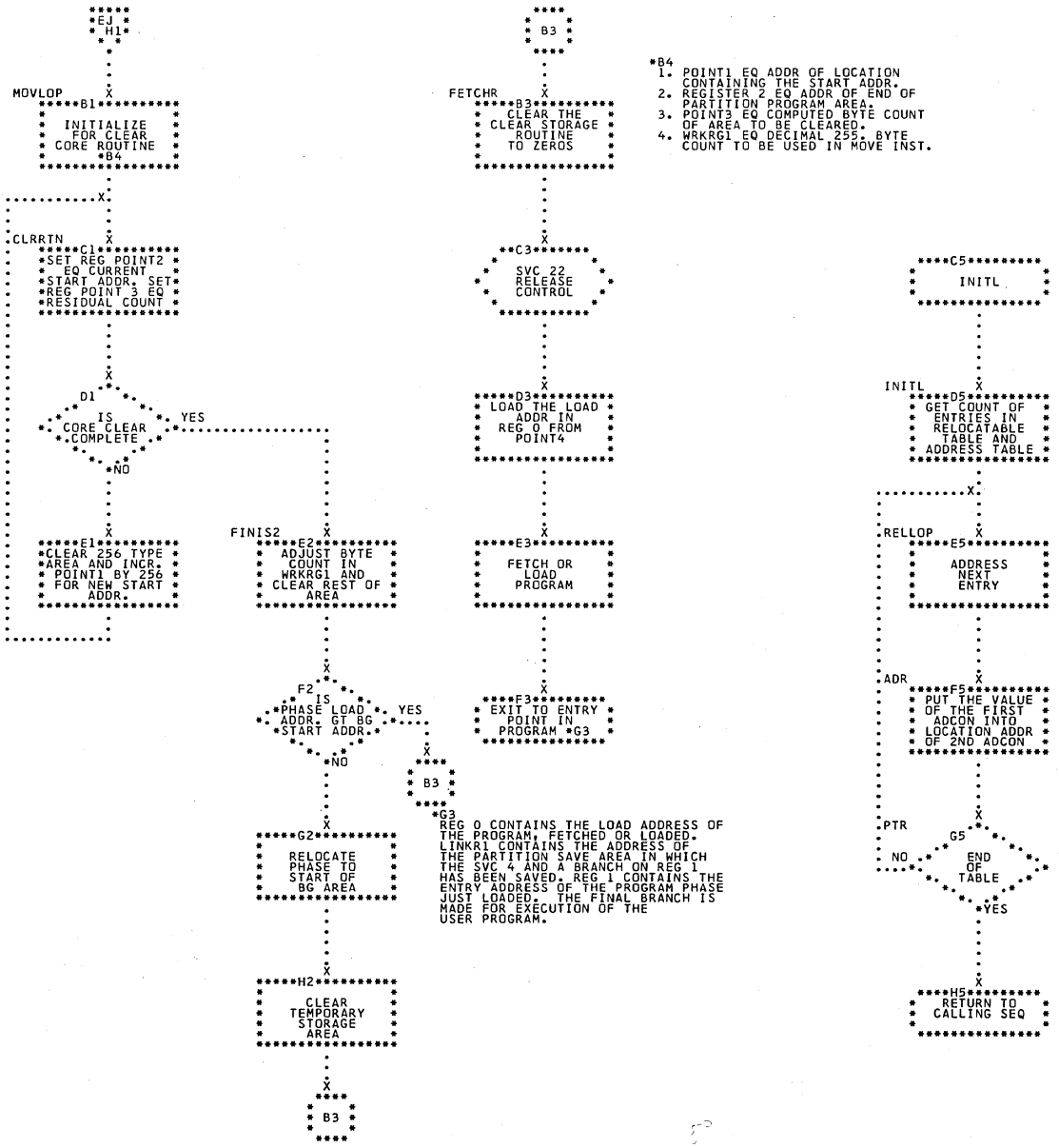
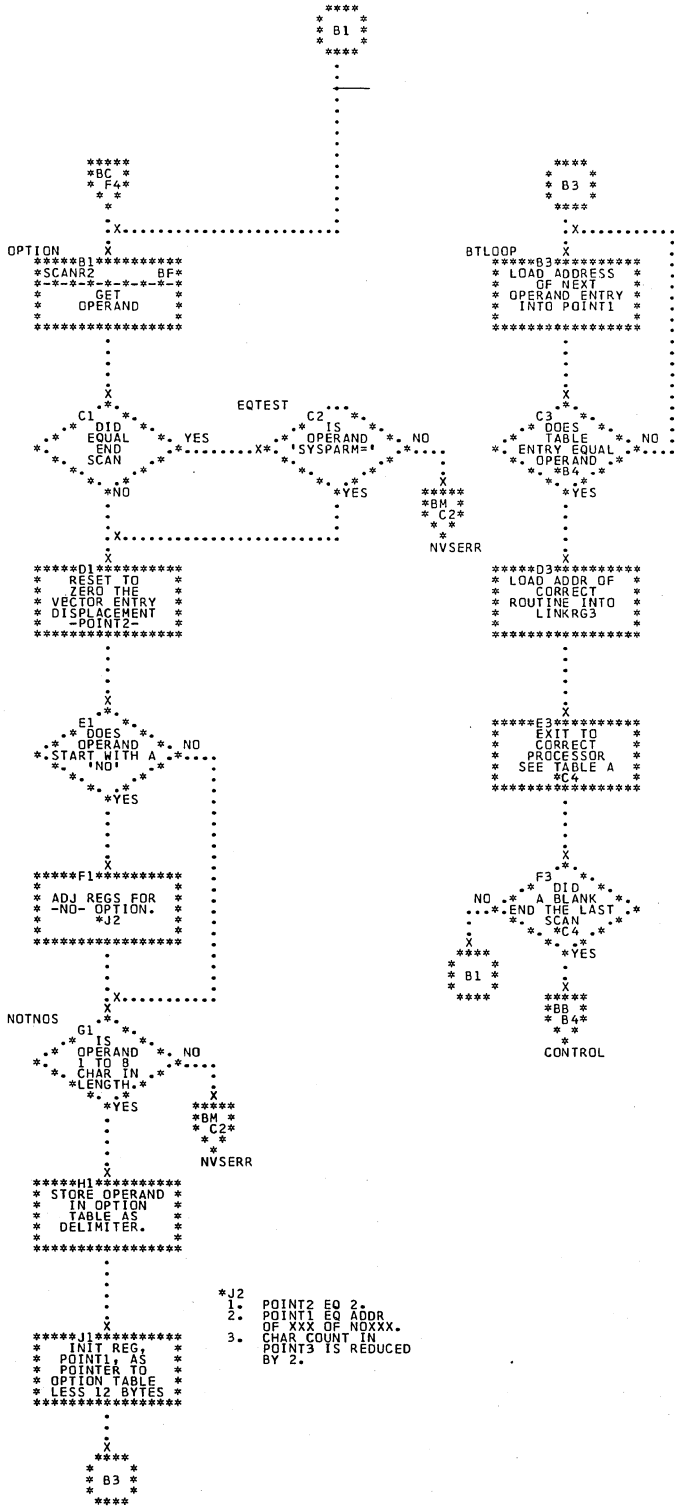


Chart EL. $JOBCTLG - OPTION Statement Processor (Part 1 of 4)
Refer to Chart 08.



*J2
1. POINT2 EQ 2.
2. POINT1 EQ ADDR OF XXX OF NOXXX.
3. CHAR COUNT IN POINT3 IS REDUCED BY 2.

*B4
BECAUSE THE OPERAND HAS BEEN STORED IN THE TABLE AS A DELIMITER, IT IS NOT POSSIBLE TO SEARCH THE ENTIRE TABLE WITHOUT FINDING AN EQUAL.

*C4
EACH PROCESSOR RETURNS CONTROL TO BLOCK F3 UNLESS AN INVALID OPERAND IS SPECIFIED. CONTROL IS TRANSFERRED TO NVOPTN AT BLOCK C2 IF AN INVALID OPERAND IS SPECIFIED.

TABLE A

OPERAND	CHART	FLOWCHART LABEL
DECK	EM	DECK
NODECK	EM	NODECK
LIST	EM	LIST
NOLIST	EM	NOLIST
LISTX	EM	LISTX
NLISTX	EM	NLISTX
SYM	EM	SYM
NOSYM	EM	NOSYM
XREF	EM	XREF
NOXREF	EM	NOXREF
ERRS	EM	ERRS
NOERRS	EM	NOERRS
48C	EM	C48
60C	EM	C60
NOLOG	EM	OPTNLG
LOG	EP	OPTLOG
USRLABEL	EN	USRLBL
STDLABEL	EN	STDLBL
DUMP	EP	DUMP
NODUMP	EP	NODUMP
LINK	EN	LINK
NOLINK	EP	NOLINK
CATAL	EN	CATAL
PARSTD	EN	PARSTD
SYSPARM=	EP	SYSPRM

Chart EM. $JOBCTLG - OPTION Statement Processor (Part 2 of 4)
Refer to Chart 08.

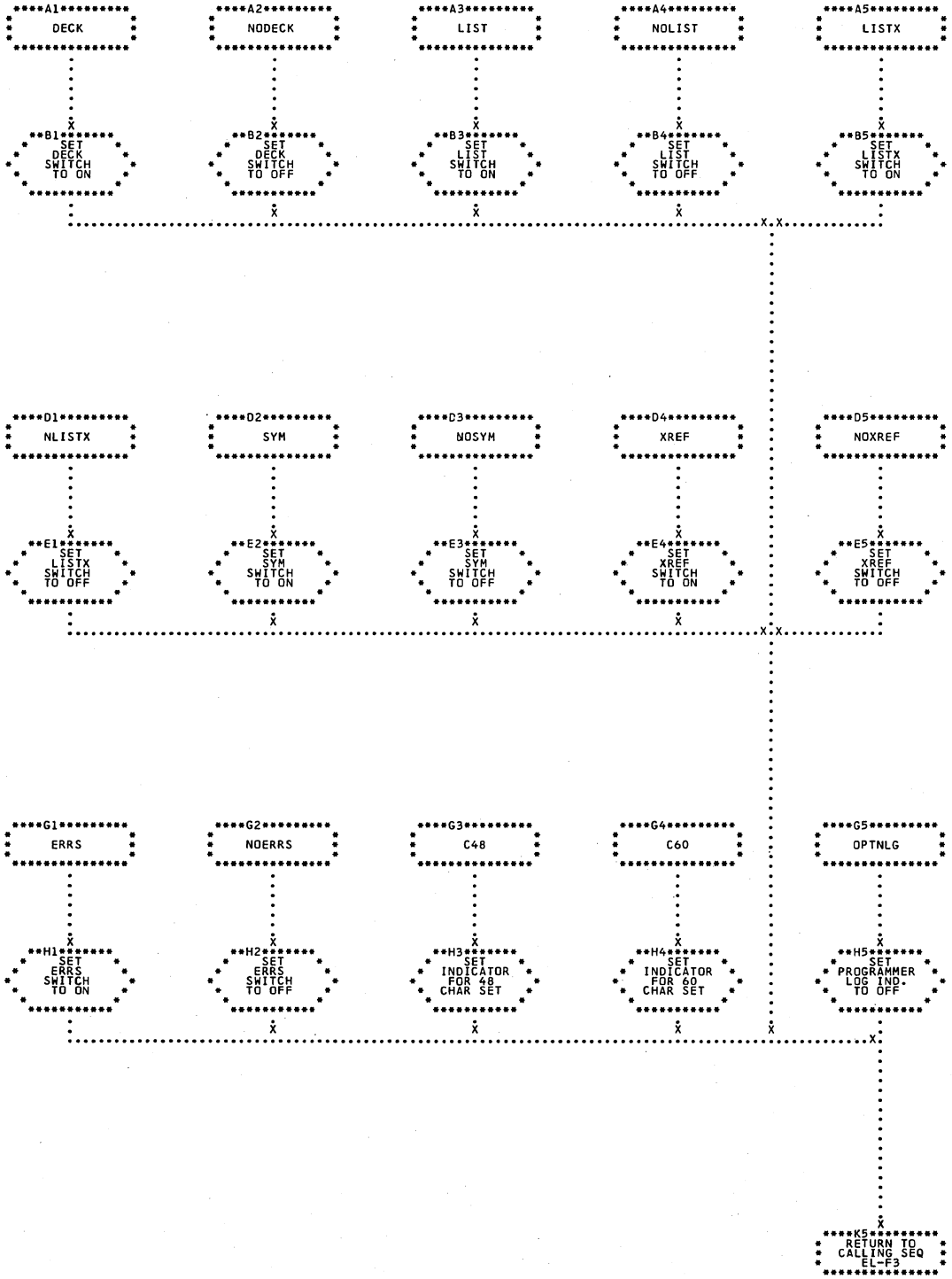
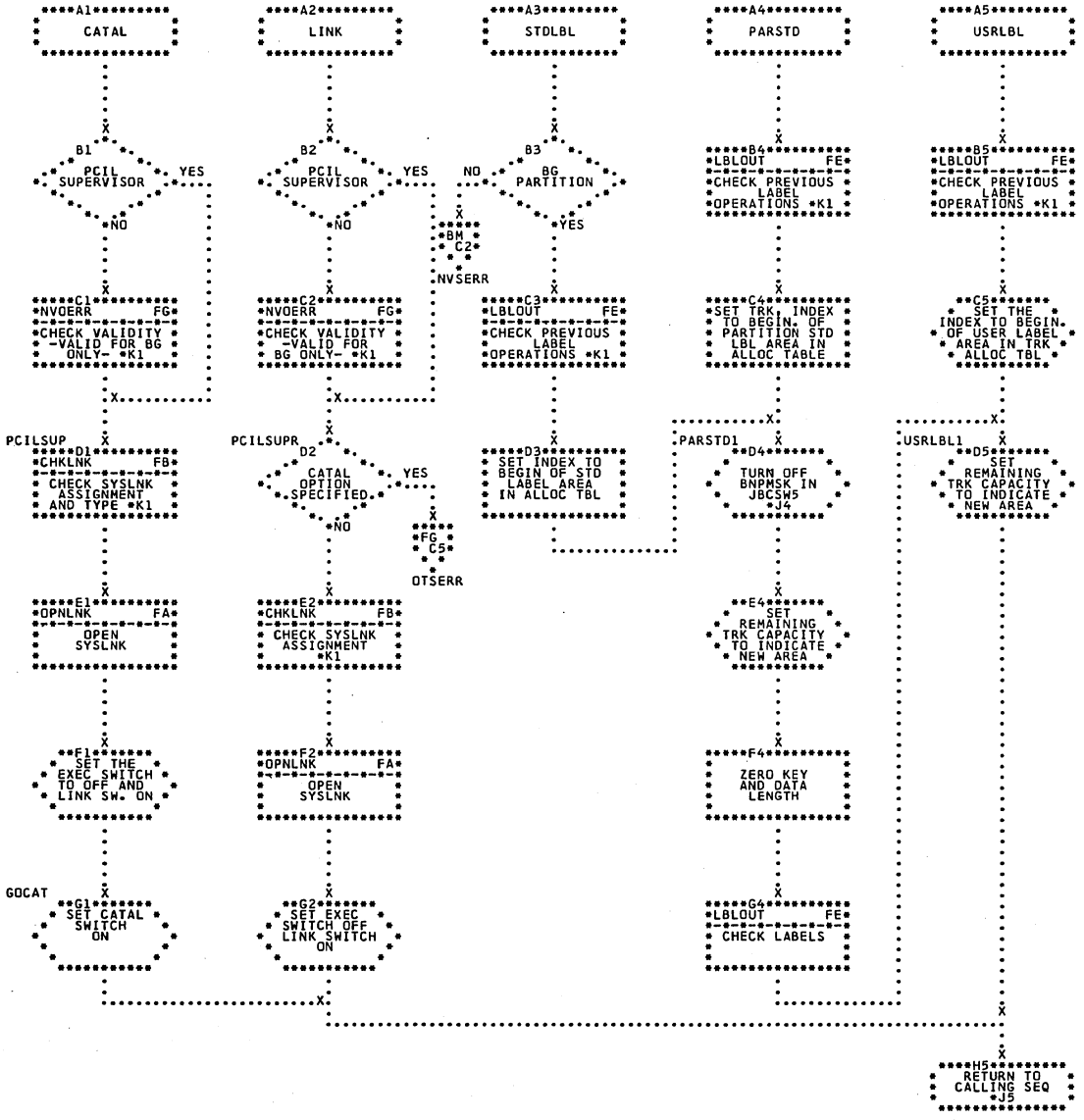


Chart EN. $JOBCTLG - OPTION Statement Processor (Part 3 of 4)
Refer to Chart 08.



*J4
BNPMSK IS BLOCK NOT PRESENT INDICATOR

*J5
RETURN IS TO EL-F3.

*K1
IF AN ERROR IS FOUND, RETURN IS NOT TO THIS ROUTINE.

Chart EP. $JOBCTLG - OPTION Statement Processor (Part 4 of 4)
Refer to Chart 08.

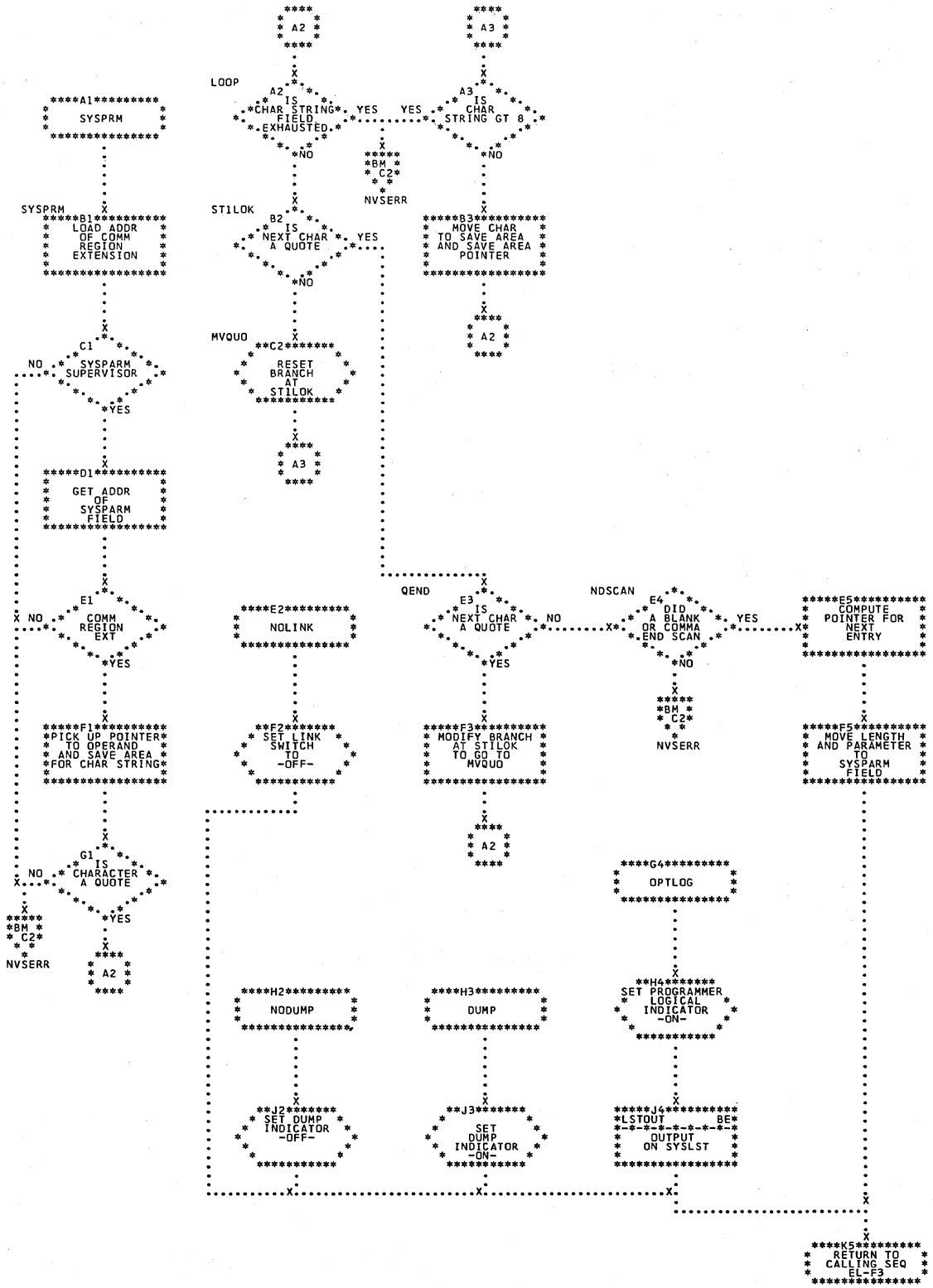


Chart FA. $JOBCTLG - Time Stamping Subroutines
Refer to Chart 09.

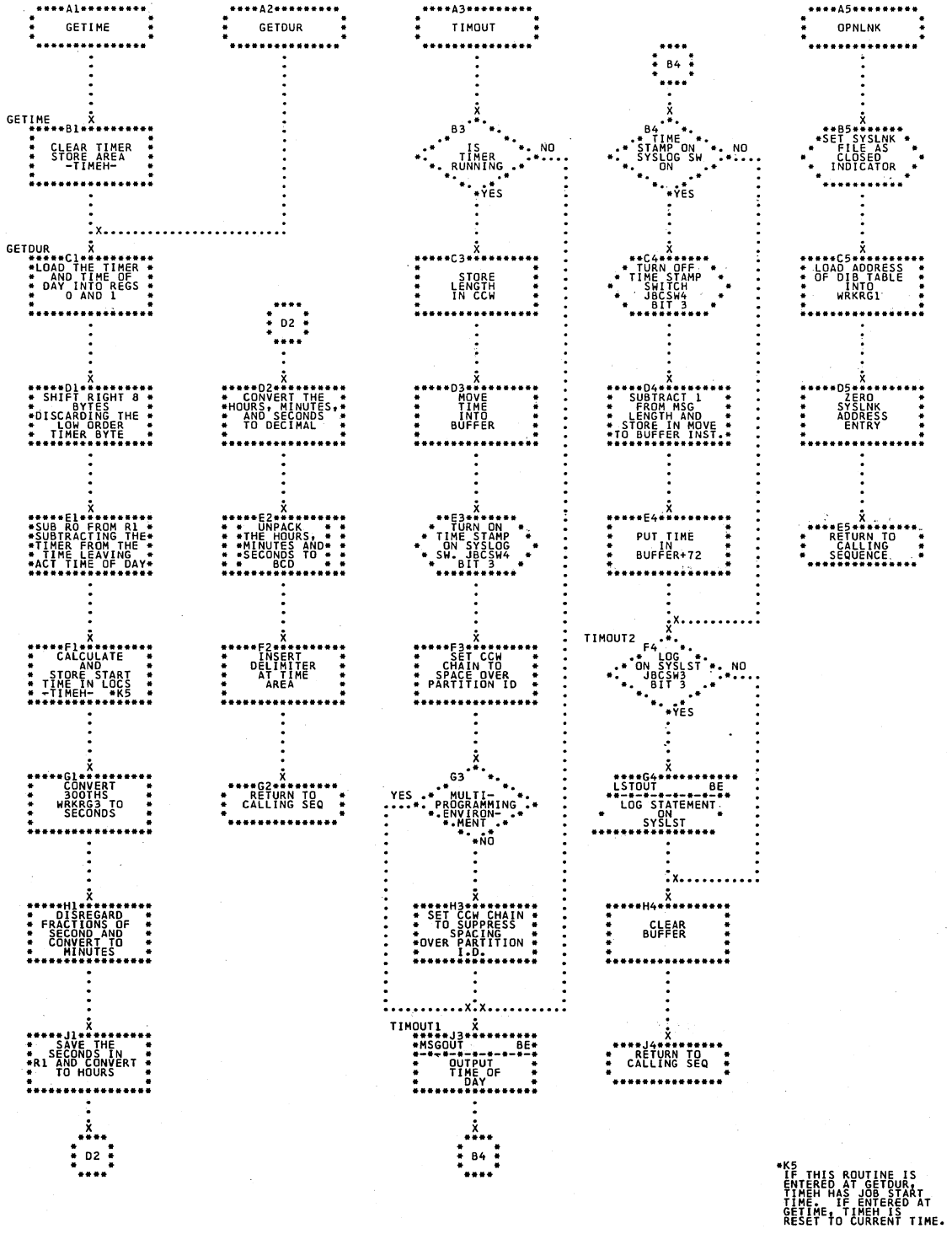


Chart FB. $JOBCTLG - Miscellaneous Subroutines (Part 1 of 3)
Refer to Chart 09.

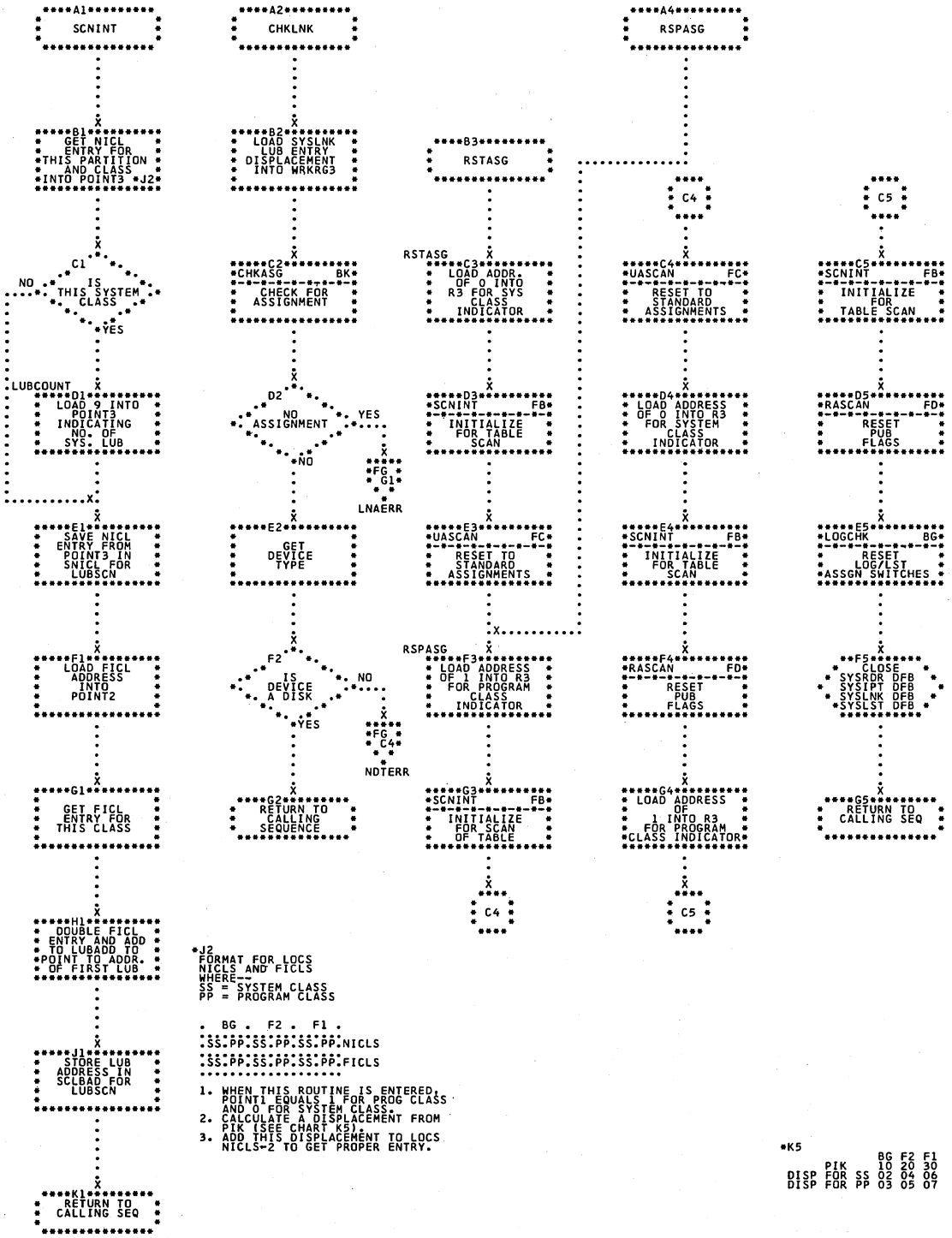


Chart FC. $JOBCTLG - Miscellaneous Subroutines (Part 2 of 3)
Refer to Chart 09.

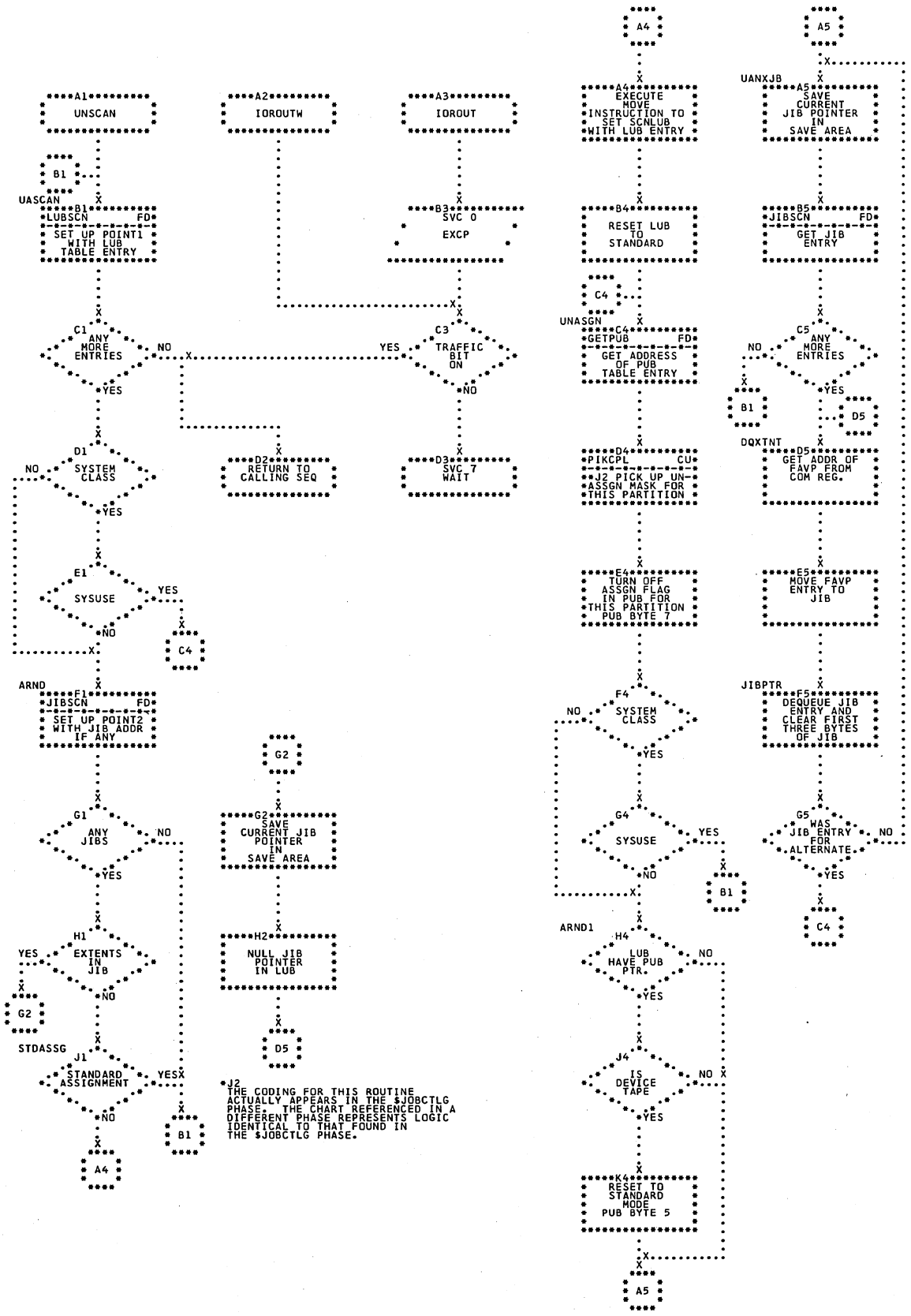


Chart FD. $JOBCTLG - Miscellaneous Subroutines (Part 3 of 3)
Refer to Chart 09.

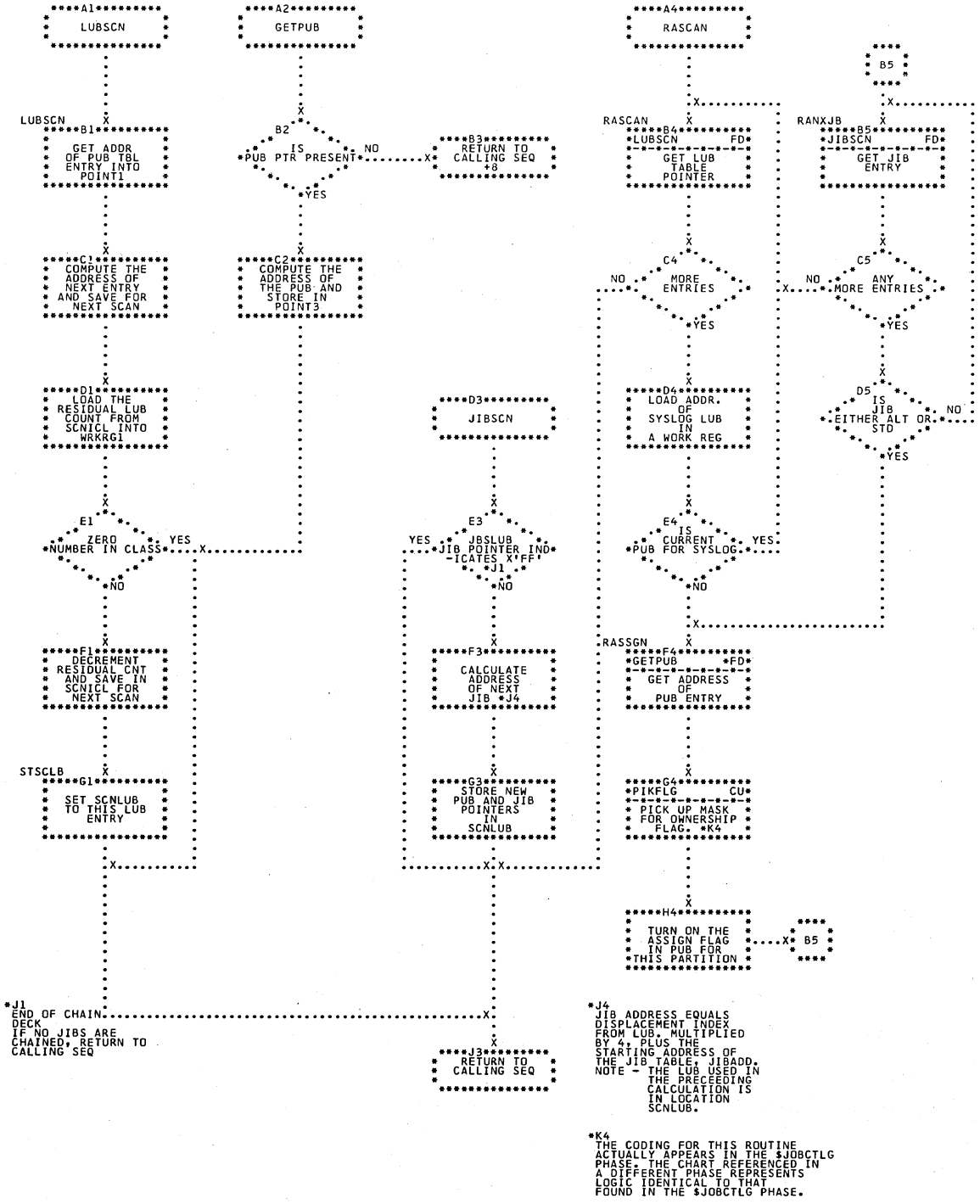


Chart FE. $JOBCTLG - Label Processing Subroutines (Part 1 of 2)
Refer to Charts 08 and 09.

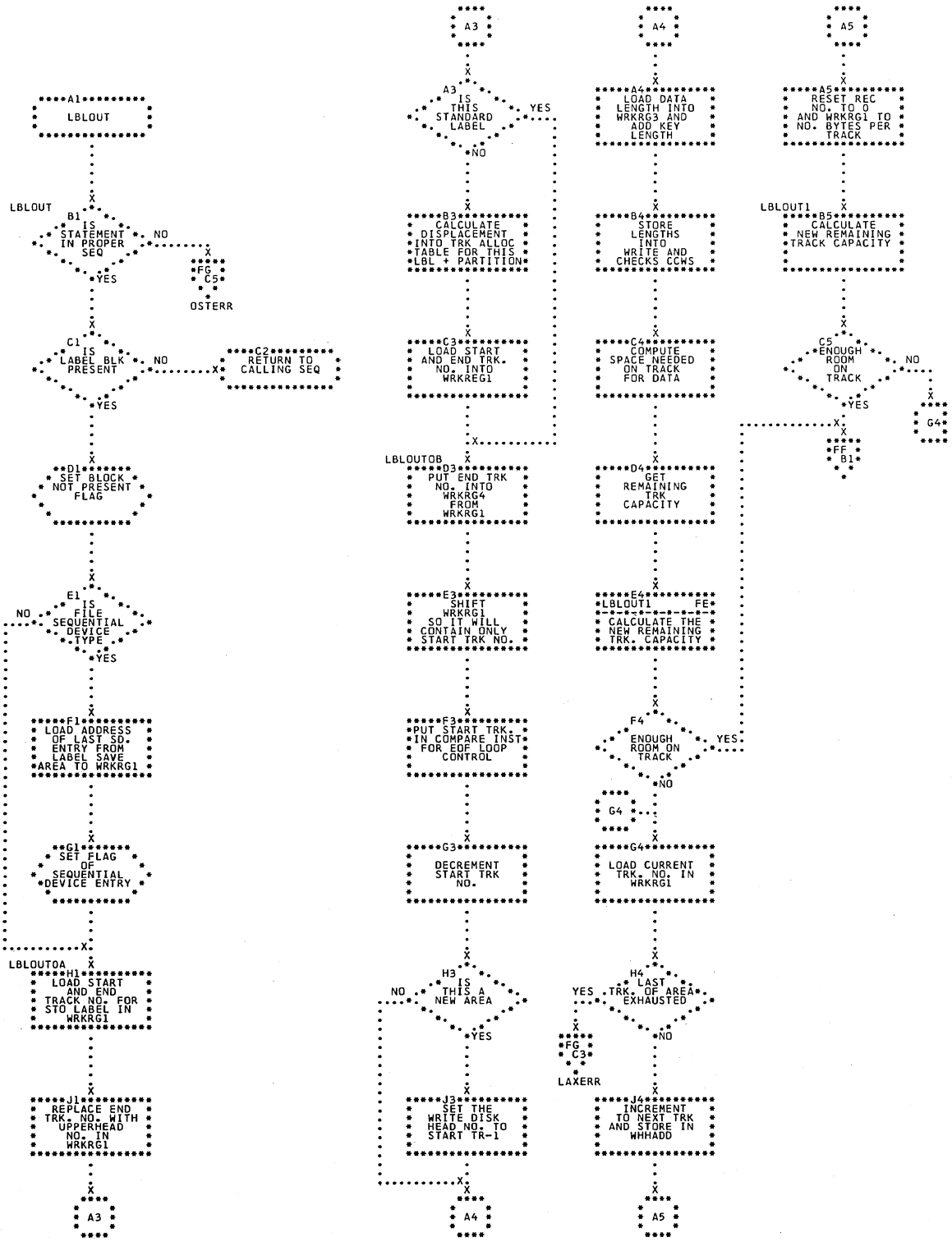


Chart FF. $JOBCTLG - Label Processing Subroutines (Part 2 of 2)
Refer to Charts 08 and 09.

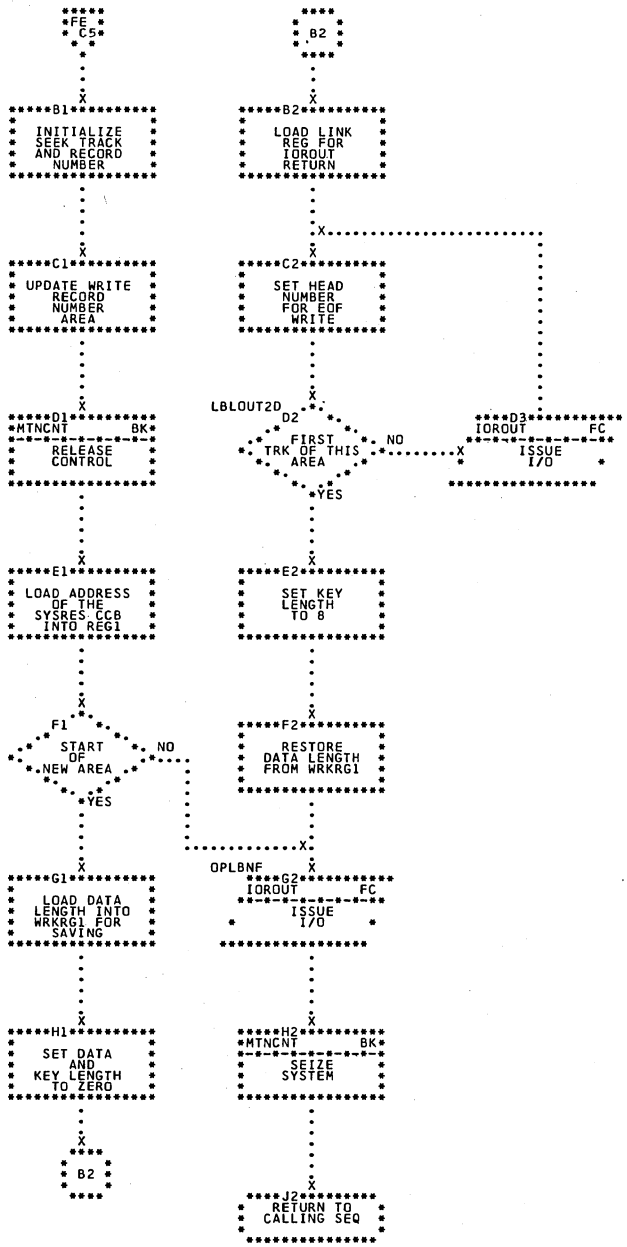


Chart FG. $JOBCTLG - Error Subroutines
Refer to Charts 08 and 09.

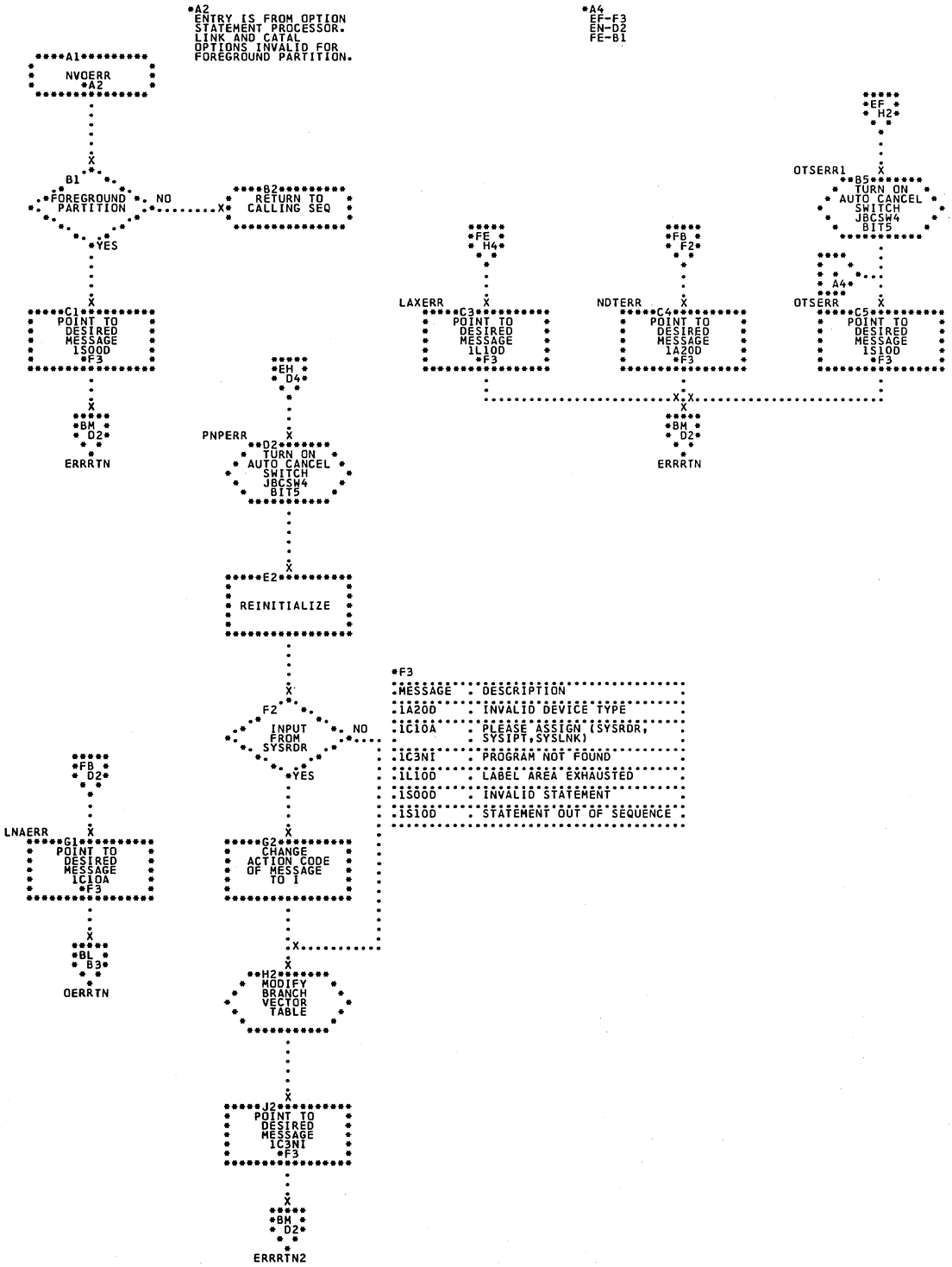


Chart GA. $JOBCTLJ - RELSE and HOLD Statement Processors
Refer to Chart 11.

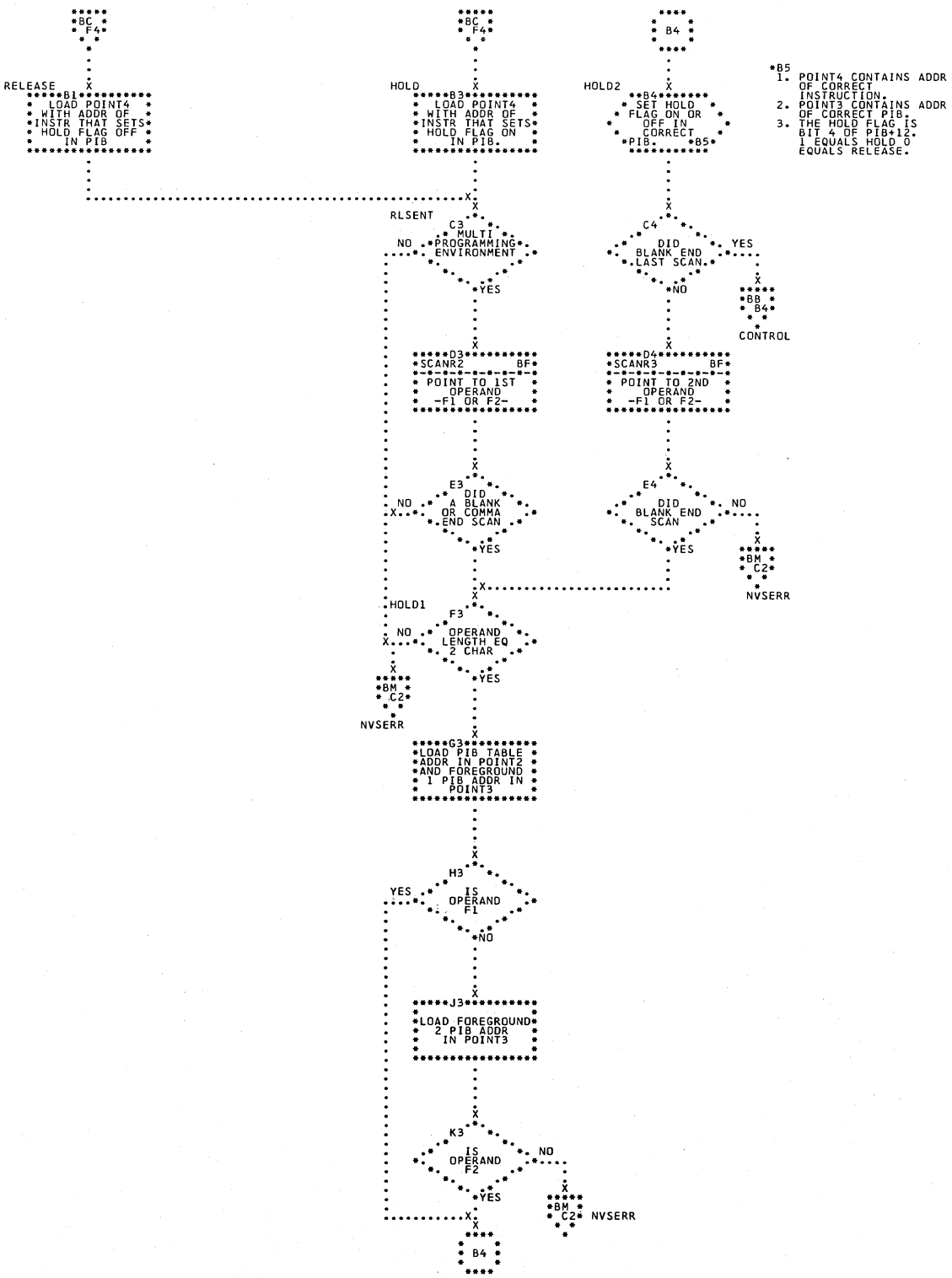


Chart GB. $JOBCTLJ - UCS Statement Processor (Part 1 of 2)
Refer to Chart 11.

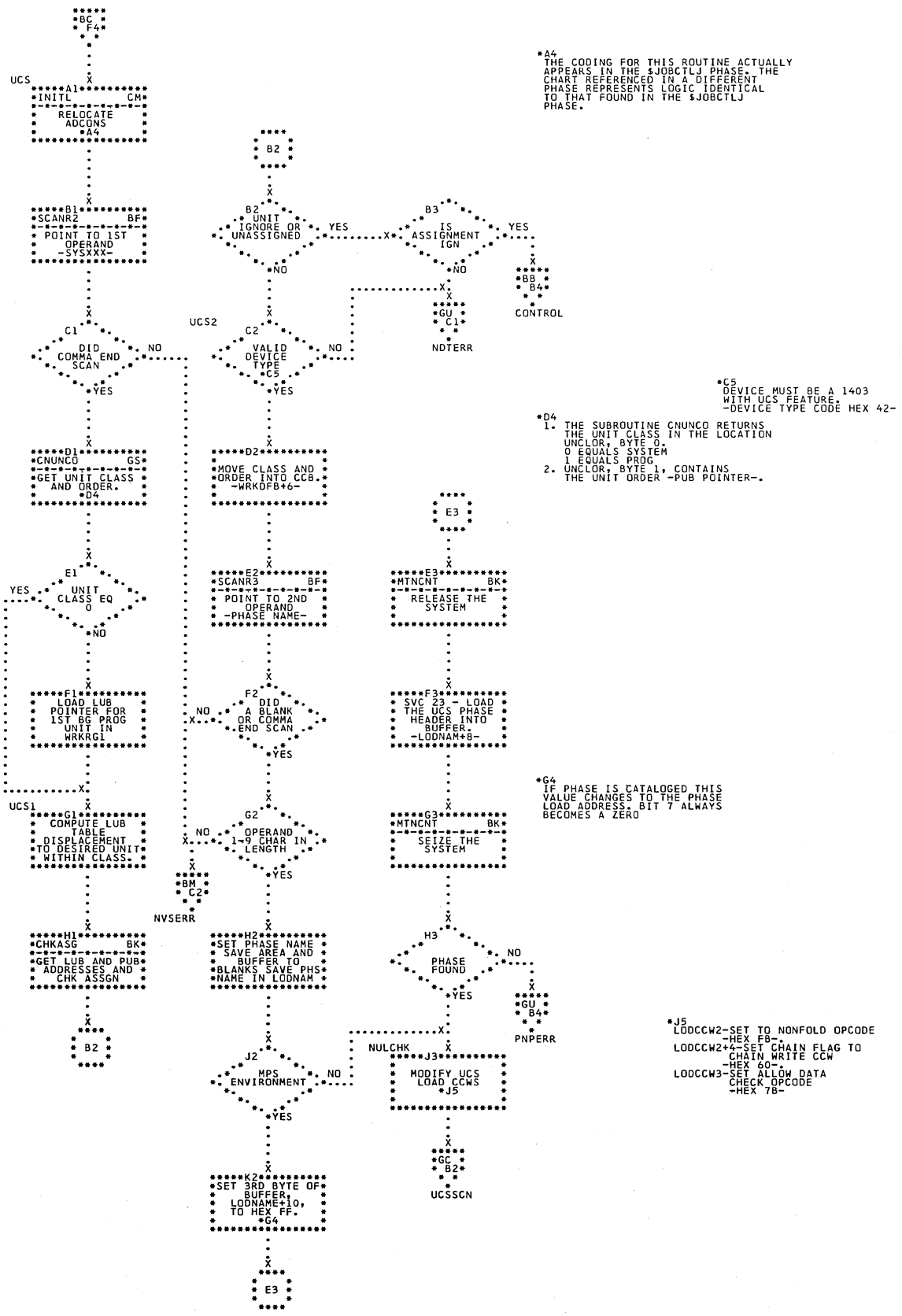


Chart GC. $JOBCTLJ - UCS Statement Processor (Part 2 of 2)
Refer to Chart 11.

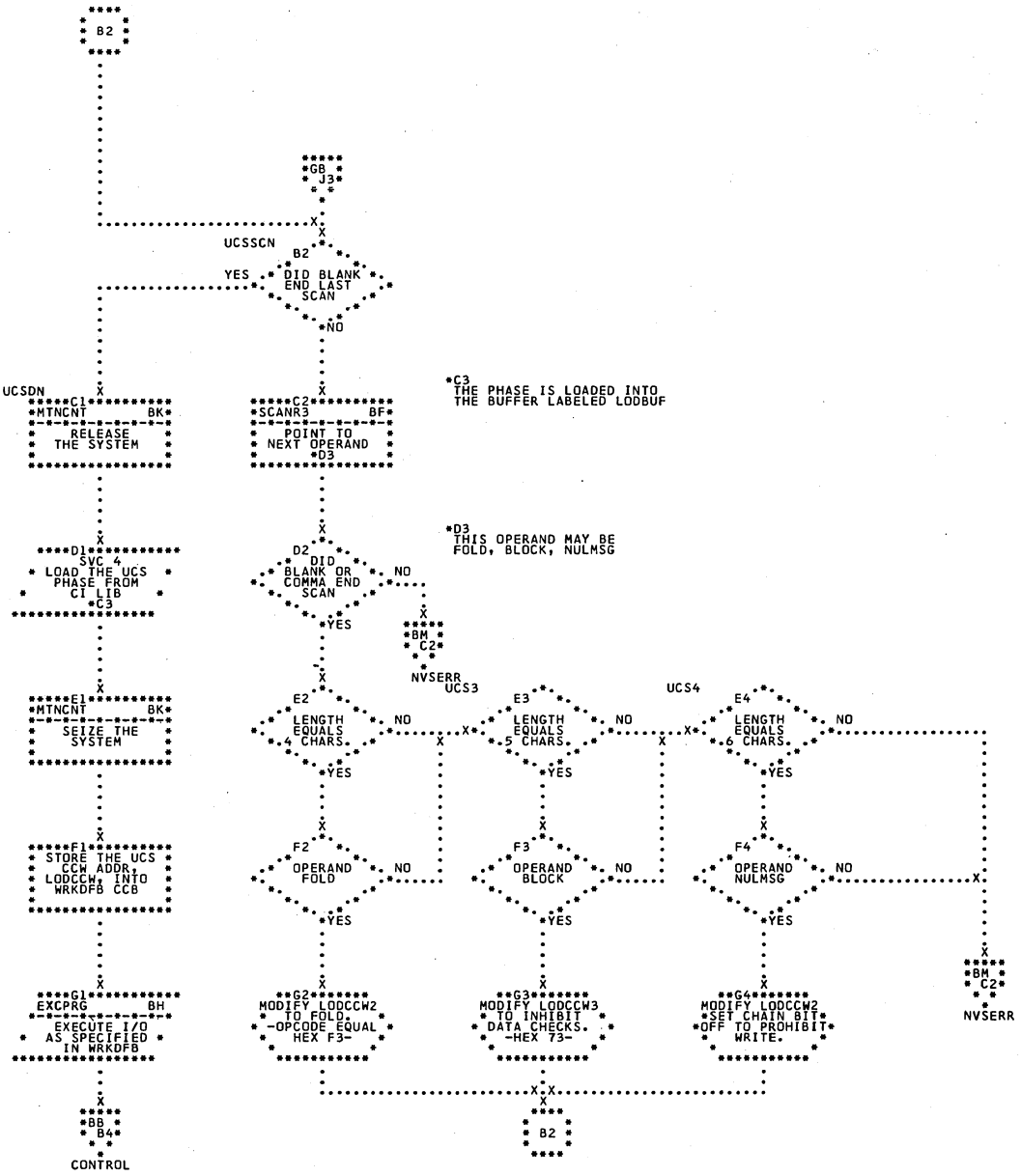


Chart GD. $JOBCTLJ - ACTION and INCLUDE Statement Processors
Refer to Charts 11 and 12.

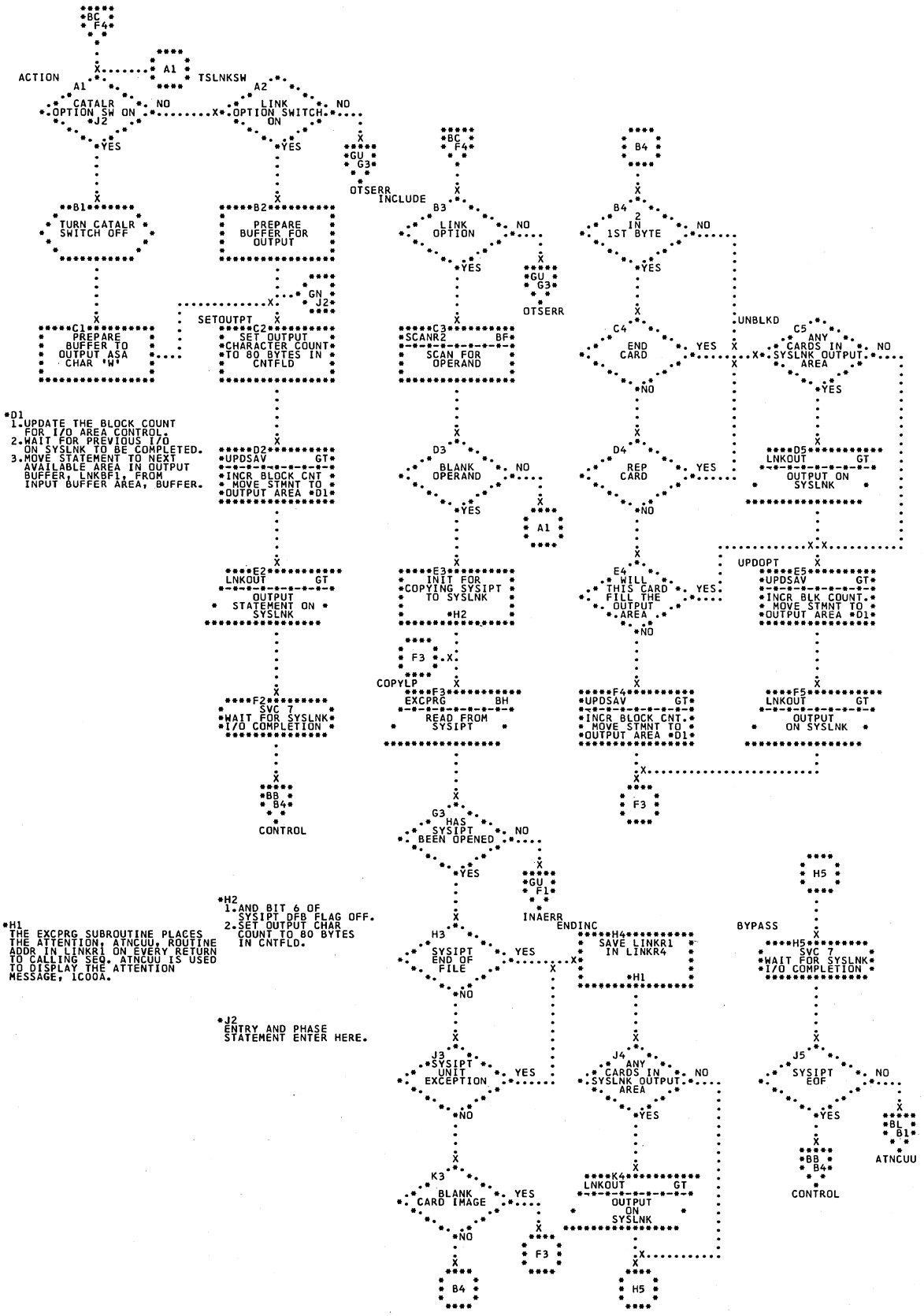


Chart GE. $JOBCTLJ - MTC Statement Processor (Part 1 of 2)
Refer to Chart 12.

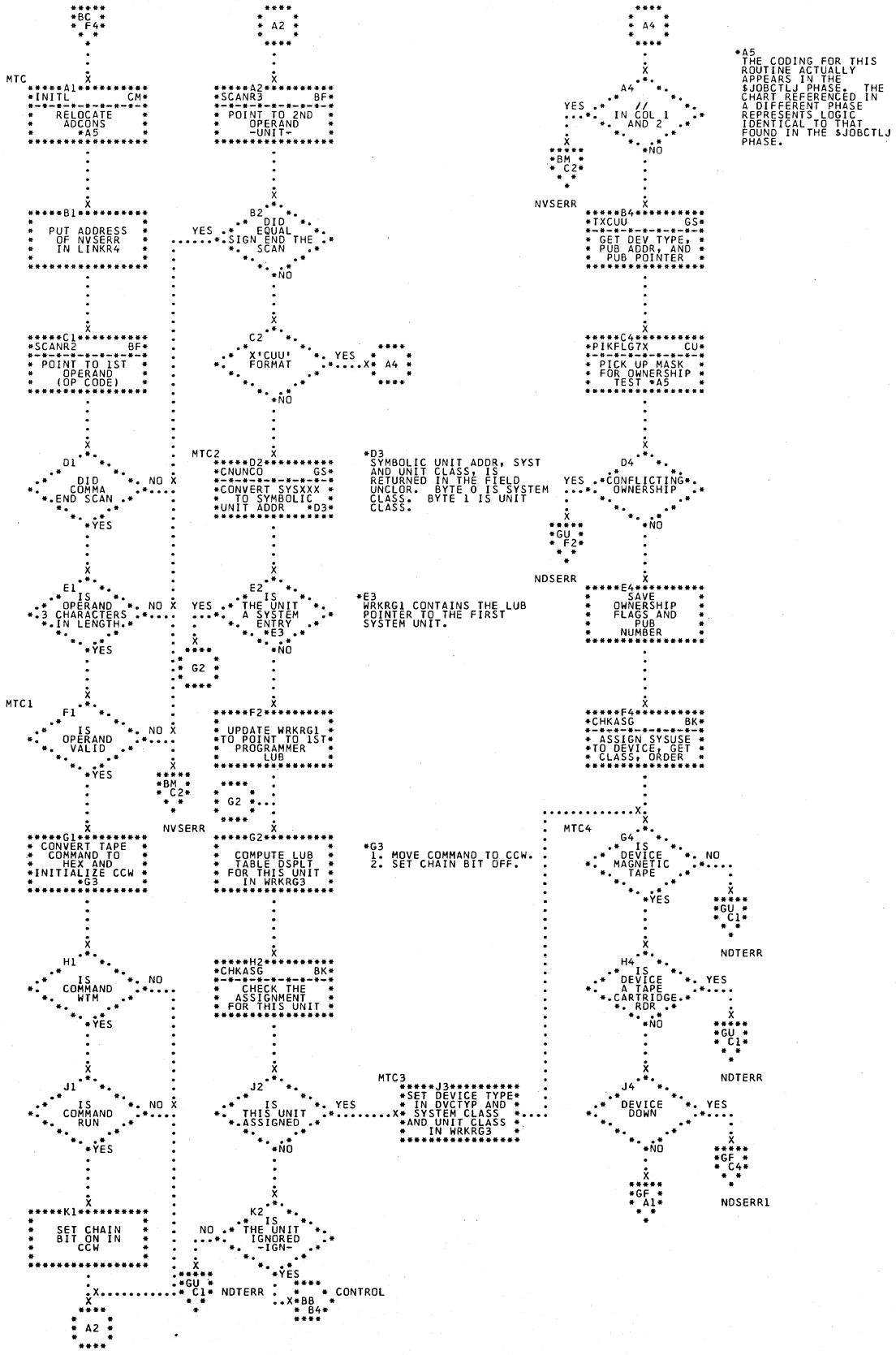


Chart GF. $JOBCTLJ - MTC Statement Processor (Part 2 of 2)
Refer to Chart 12.

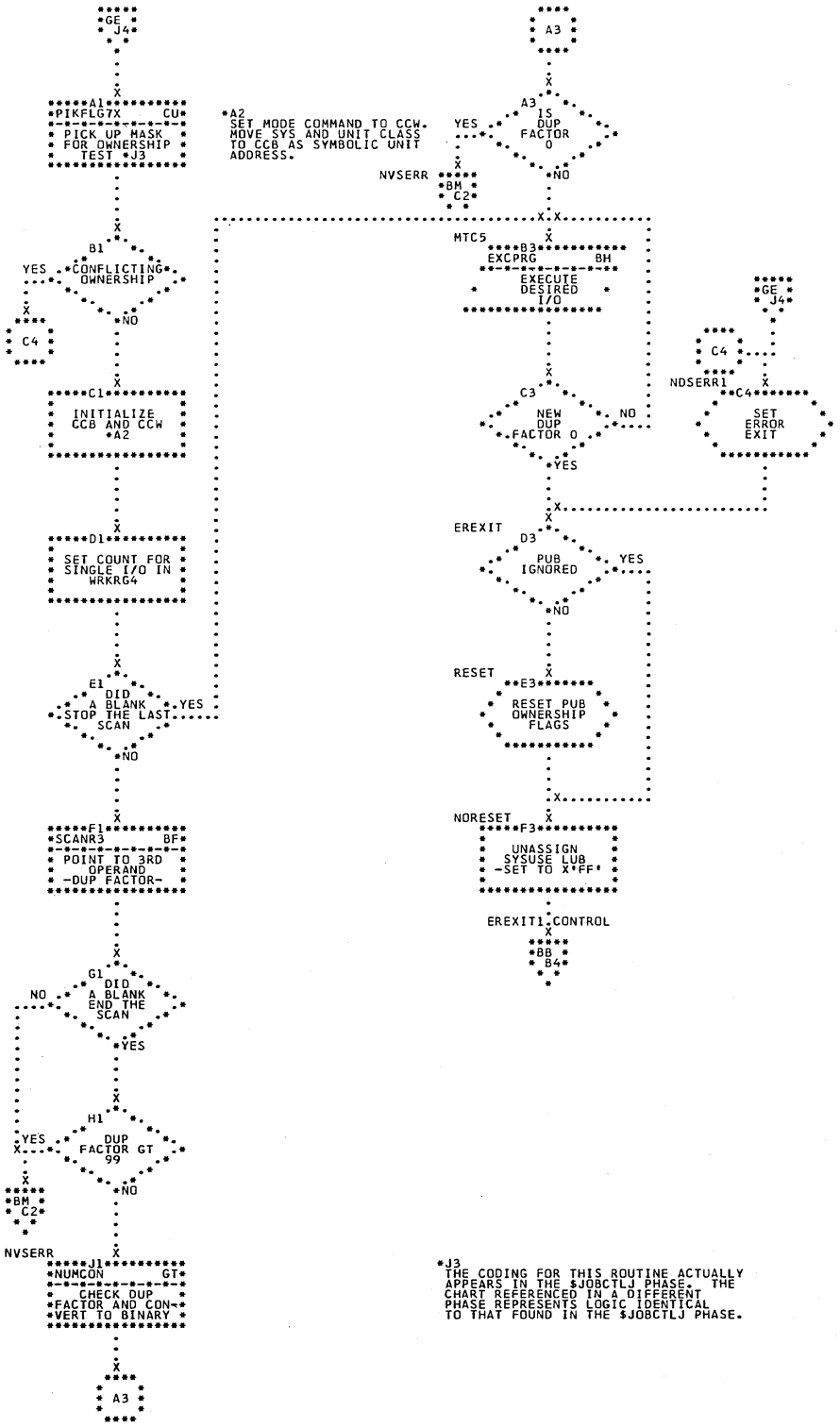


Chart GG. $JOBCTLJ - SET Statement Processor (Part 1 of 3)
Refer to Chart 12.

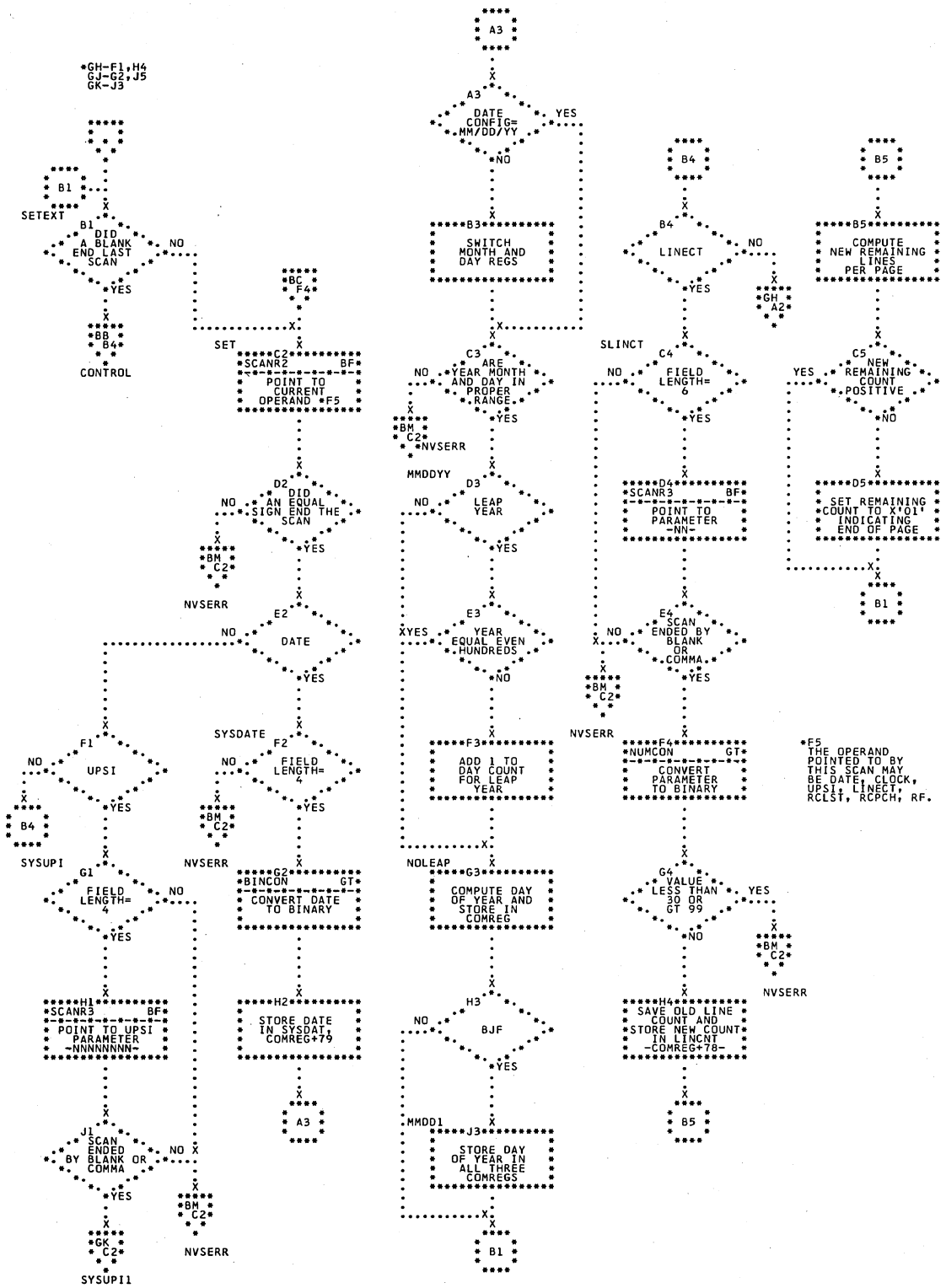


Chart GH. $JOBCTLJ - SET Statement Processor (Part 2 of 3)
Refer to Chart 12.

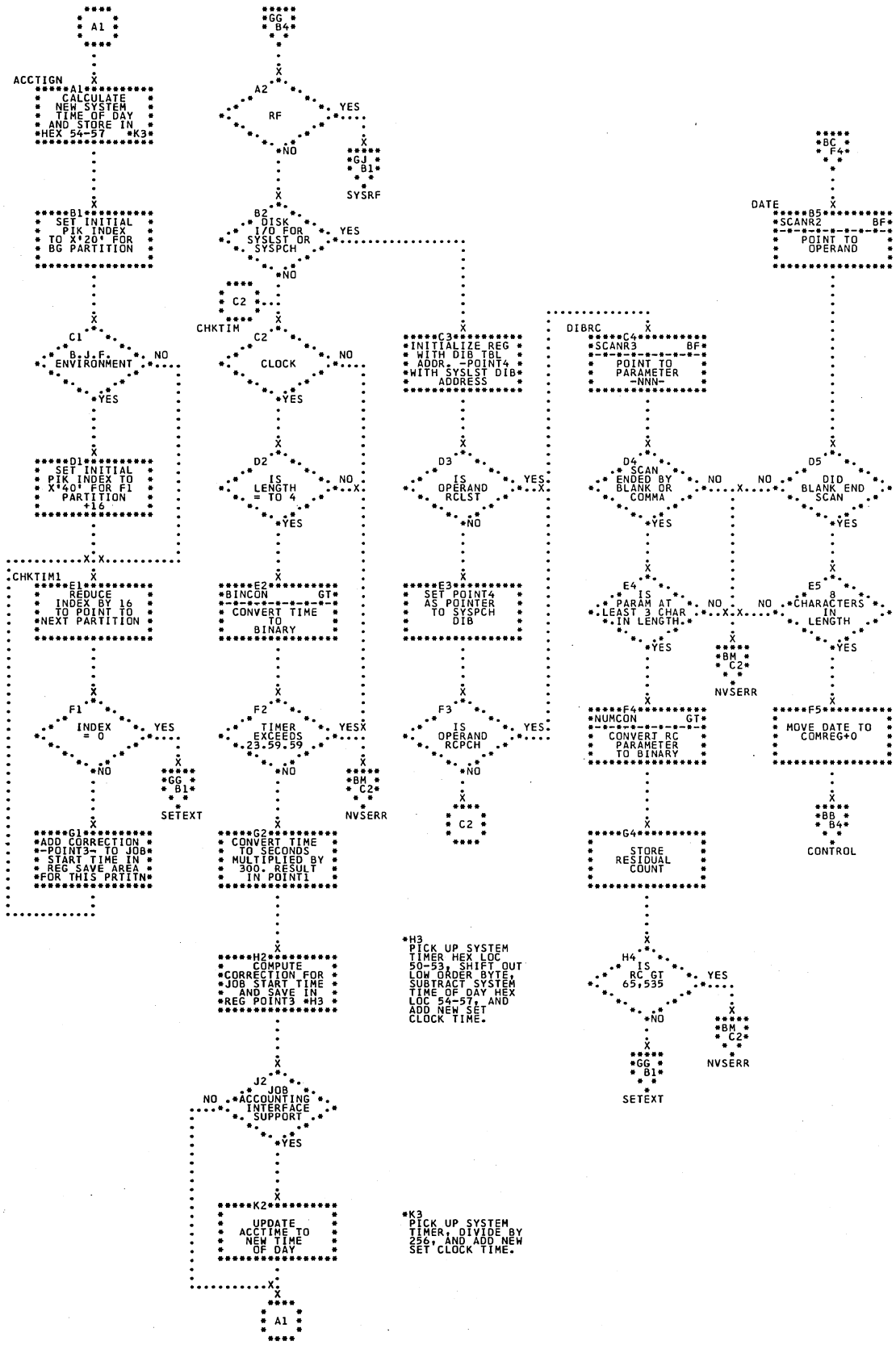


Chart GJ. $JOBCTLJ - SET Statement Processor (Part 3 of 3)
Refer to Chart 12.

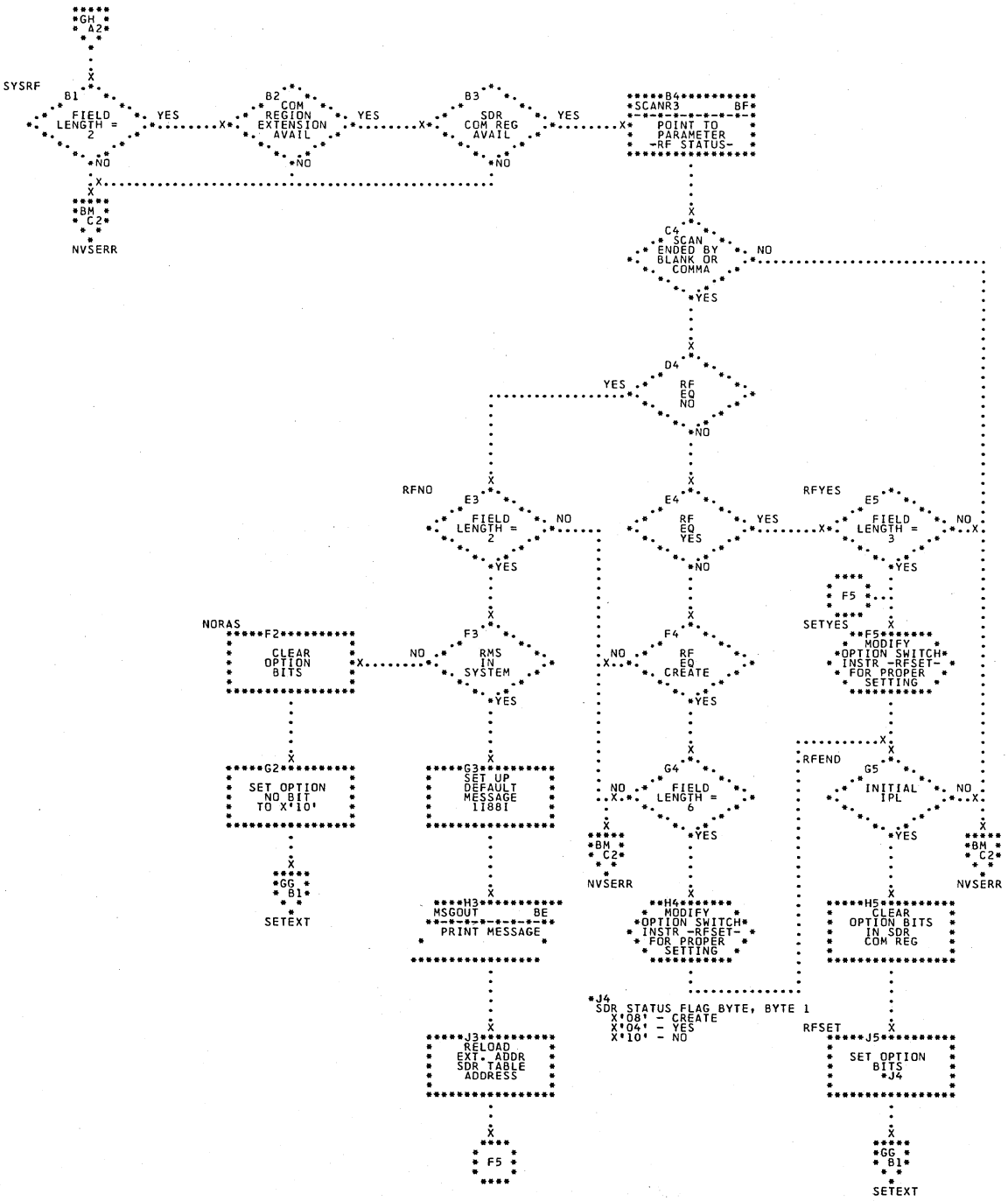
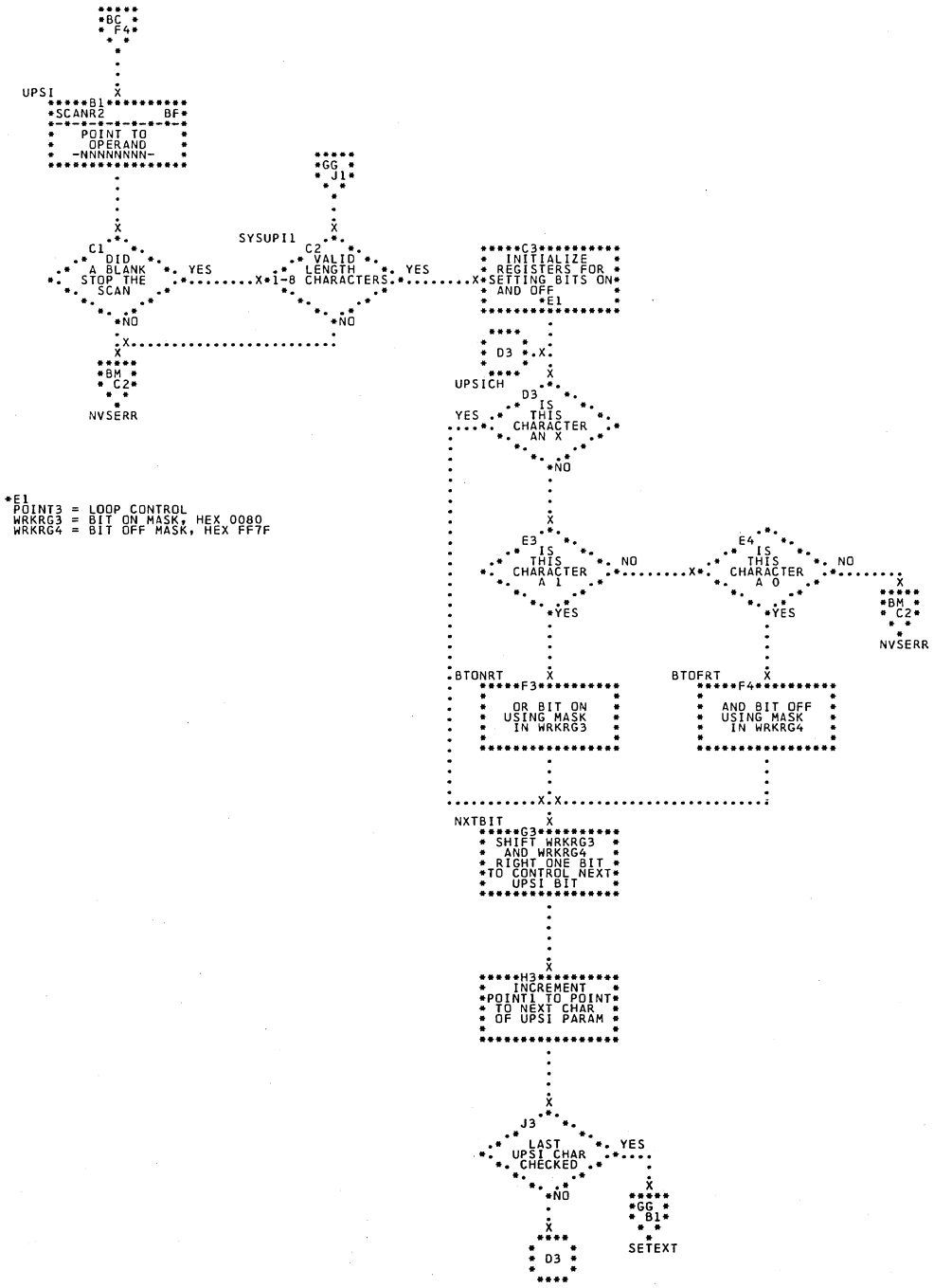


Chart GK. $JOBCTLJ - UPSI Statement Processor
Refer to Chart 12.



*E1
POINT3 = LOOP CONTROL
WRKRG3 = BIT ON MASK, HEX 0080
WRKRG4 = BIT OFF MASK, HEX FF7F

Chart GL. $JOBCTLJ - PAUSE, LOG, and NOLOG Statement Processors
Refer to Chart 10.

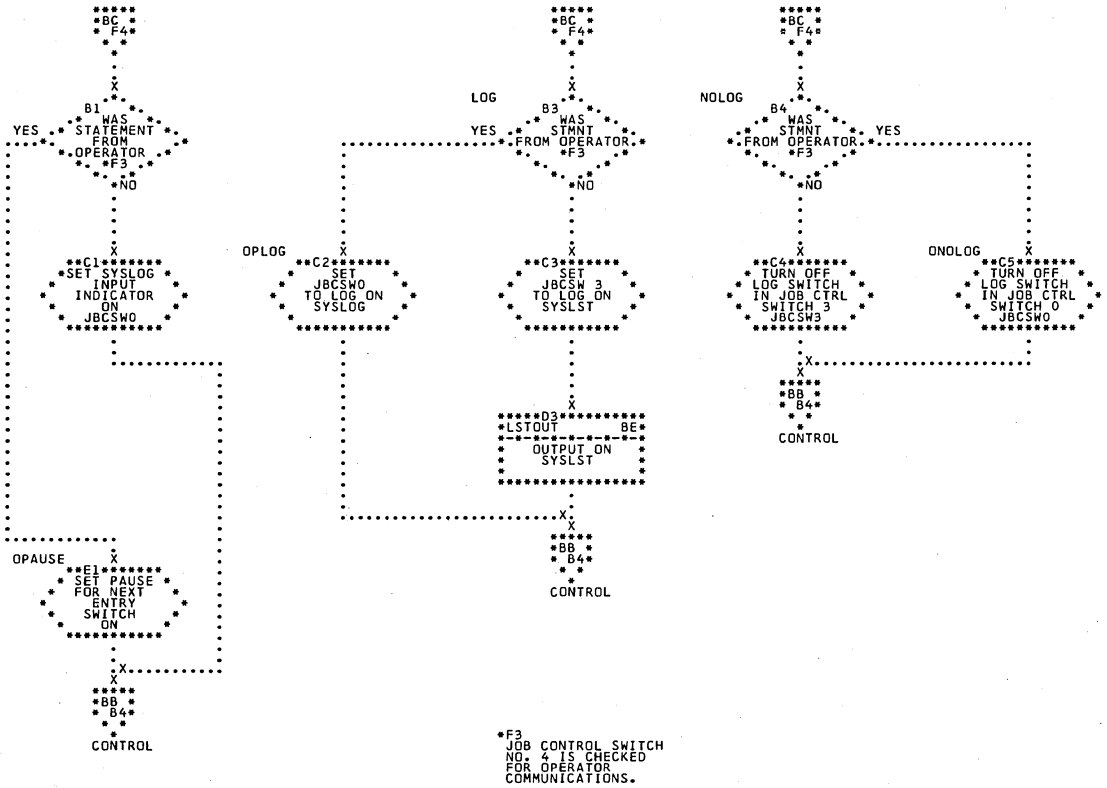


Chart GM. $JOBCTLJ - STOP Statement Processor
Refer to Chart 10.

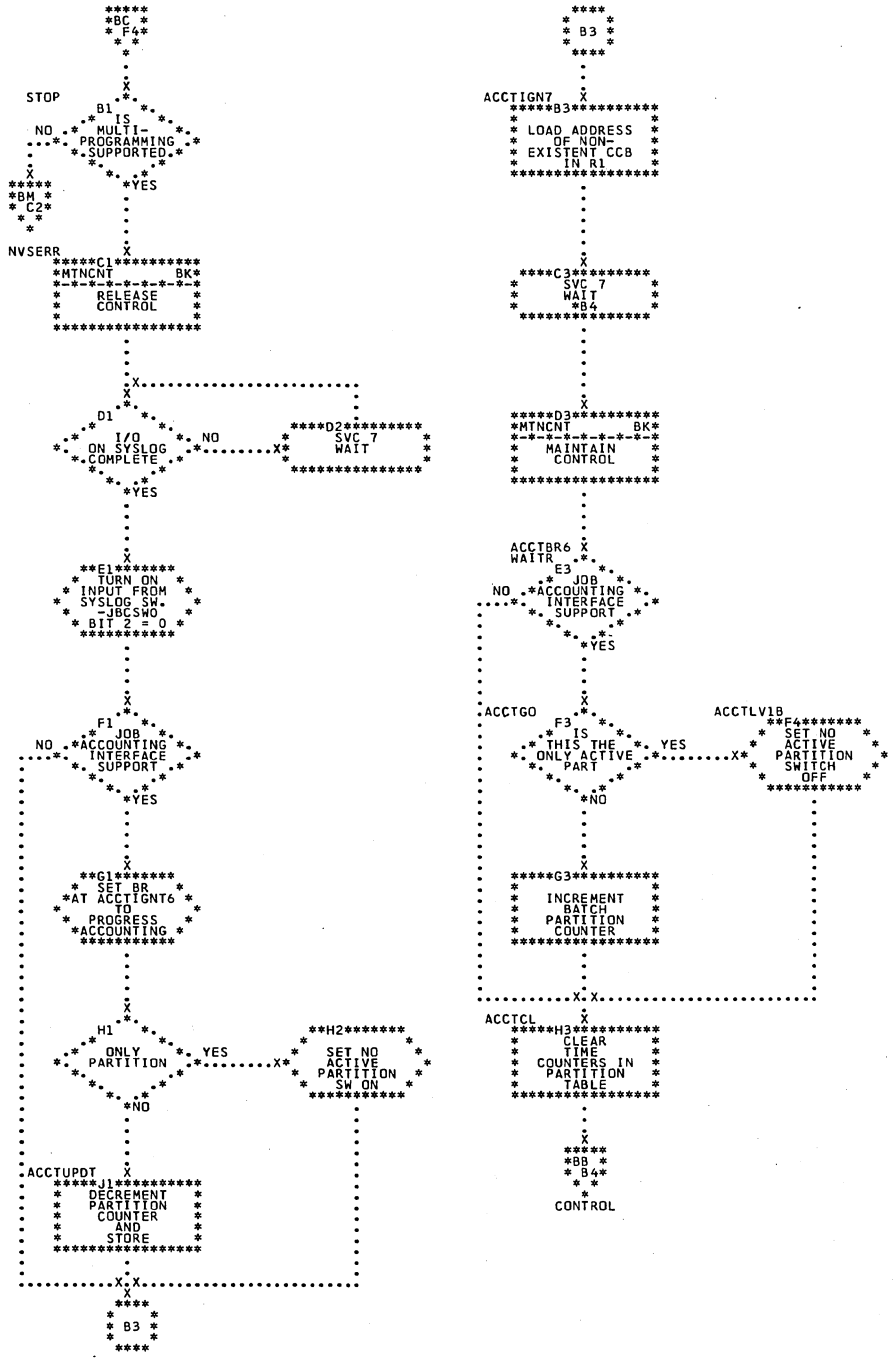


Chart GN. $JOBCTLJ - CATALR Card Processor
Refer to Chart 12.

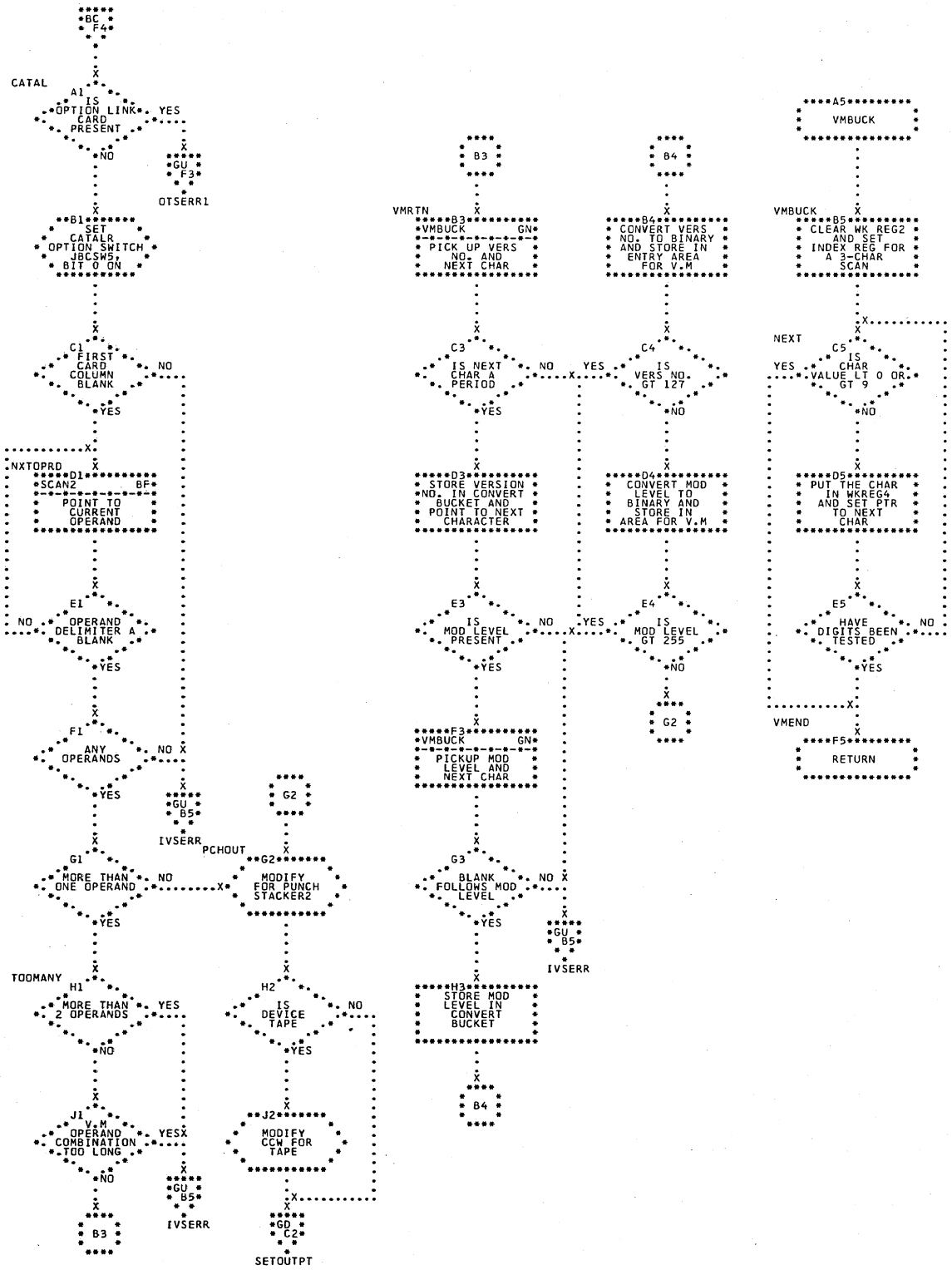
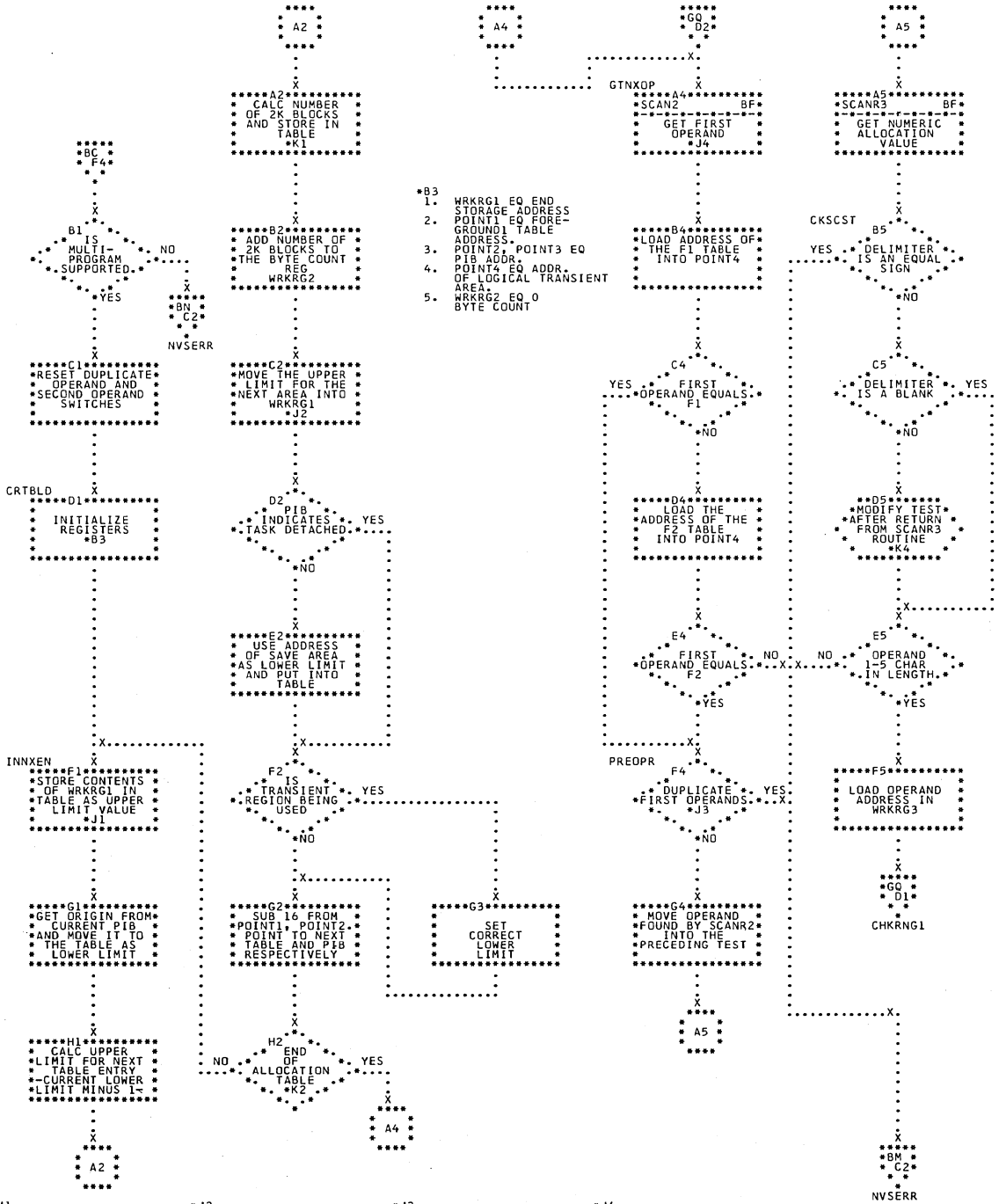


Chart GP. $JOBCTLJ - ALLOC Statement Processor (Part 1 of 3)
Refer to Chart 10.



*J1
THIS VALUE IS ALSO STORED IN THE ACTIVE LOWER LIMIT FIELD INDICATING THIS FIELD IS INACTIVE. IT CONTAINS A HEX FF IN THE LAST BYTE

*J2
WRGRG1 CONTAINS CHANGING (DYNAMIC) UPPER LIMIT INFORMATION.

*J3
EXIT ON THE NO LEG FIRST TIME THROUGH THIS ROUTINE.

*J4
THE ALLOC STATEMENT FORMAT - ALLOC F1=2K, F2=2K. F1= OR F2= IS CONSIDERED THE FIRST OPERAND. IF BOTH F1 AND F2 ARE SPECIFIED SCANR2 IS USED TWICE.

*K1
CURRENT UPPER LIMIT MINUS NEXT UPPER LIMIT DIVIDED BY 2048.

*K2
THIS LOOP IS EXECUTED UNTIL 3 TABLES, ONE FOR EACH TYPE OF PROBLEM PROGRAM, ARE BUILT. THE TABLES ARE BUILT IN THE SEQUENCE F1, F2, AND BG. THE SEQUENCE IS A REVERSAL OF THE PHYSICAL ORDER. SEE LABEL LIST FOR A TABLE, F2TBEN AND F1TBEN EXPANSIONS.

*K4
THE INSTRUCTION AT LOCATION CKSCST IS MODIFIED TO BRANCH TO THE ERROR HANDLING ROUTINE IF THE SECOND OPERAND IS DELIMITED BY AN EQUAL SIGN OR A COMMA. THE COMMA IS ALLOWED TO DELIMIT THE FIRST OPERAND.

Chart GQ. $JOBCTLJ - ALLOC Statement Processor (Part 2 of 3)
Refer to Chart 10.

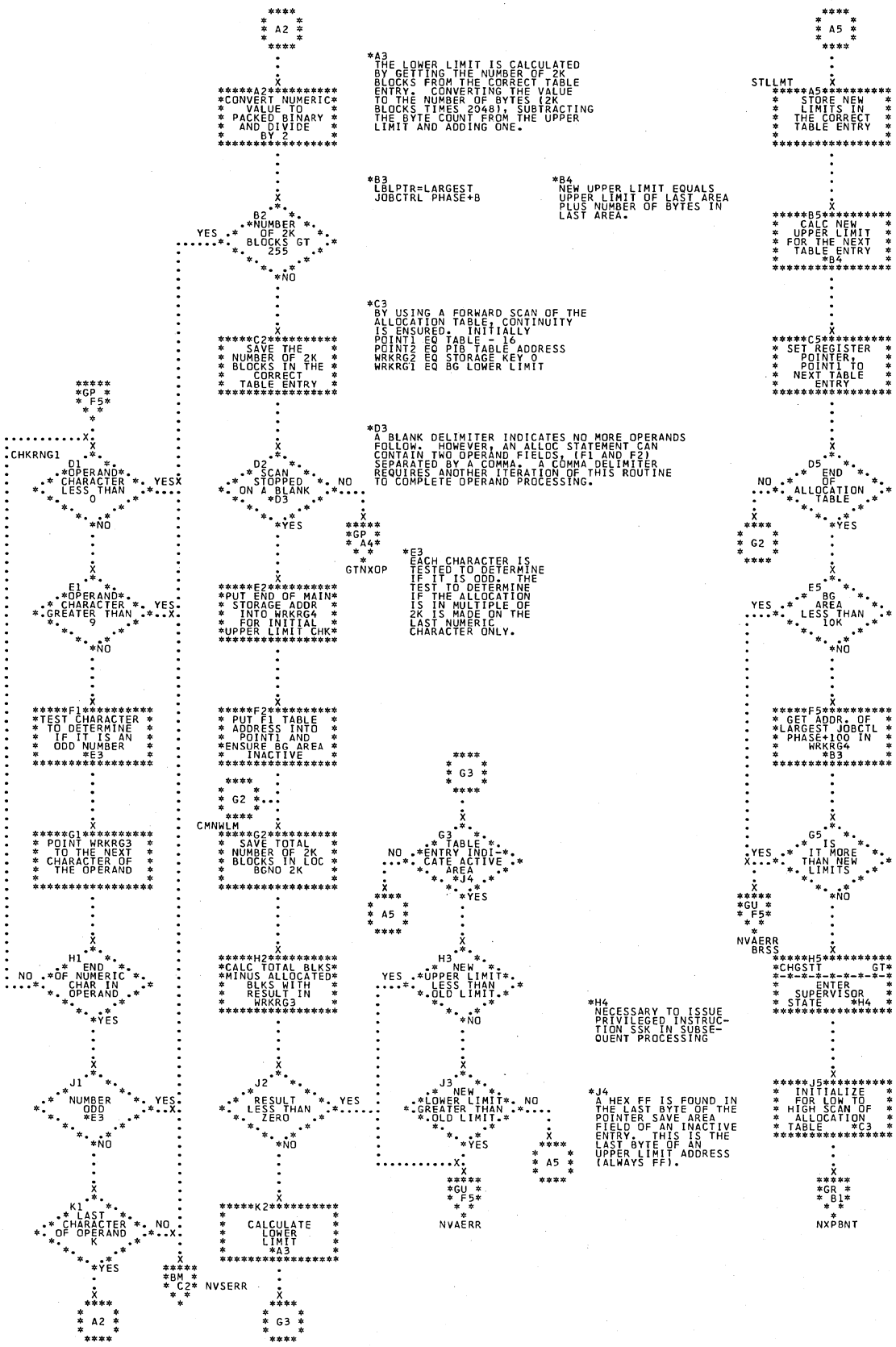


Chart GR. $JOBCTLJ - ALLOC Statement Processor (Part 3 of 3)
Refer to Chart 10.

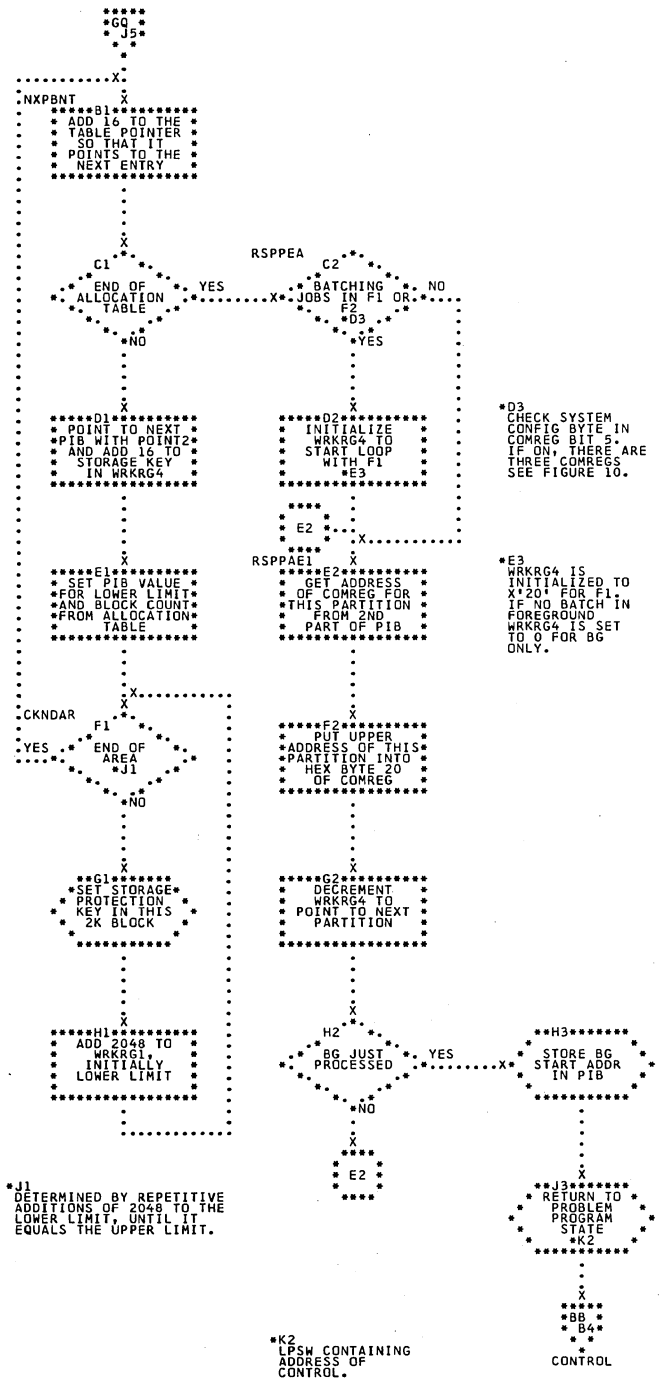
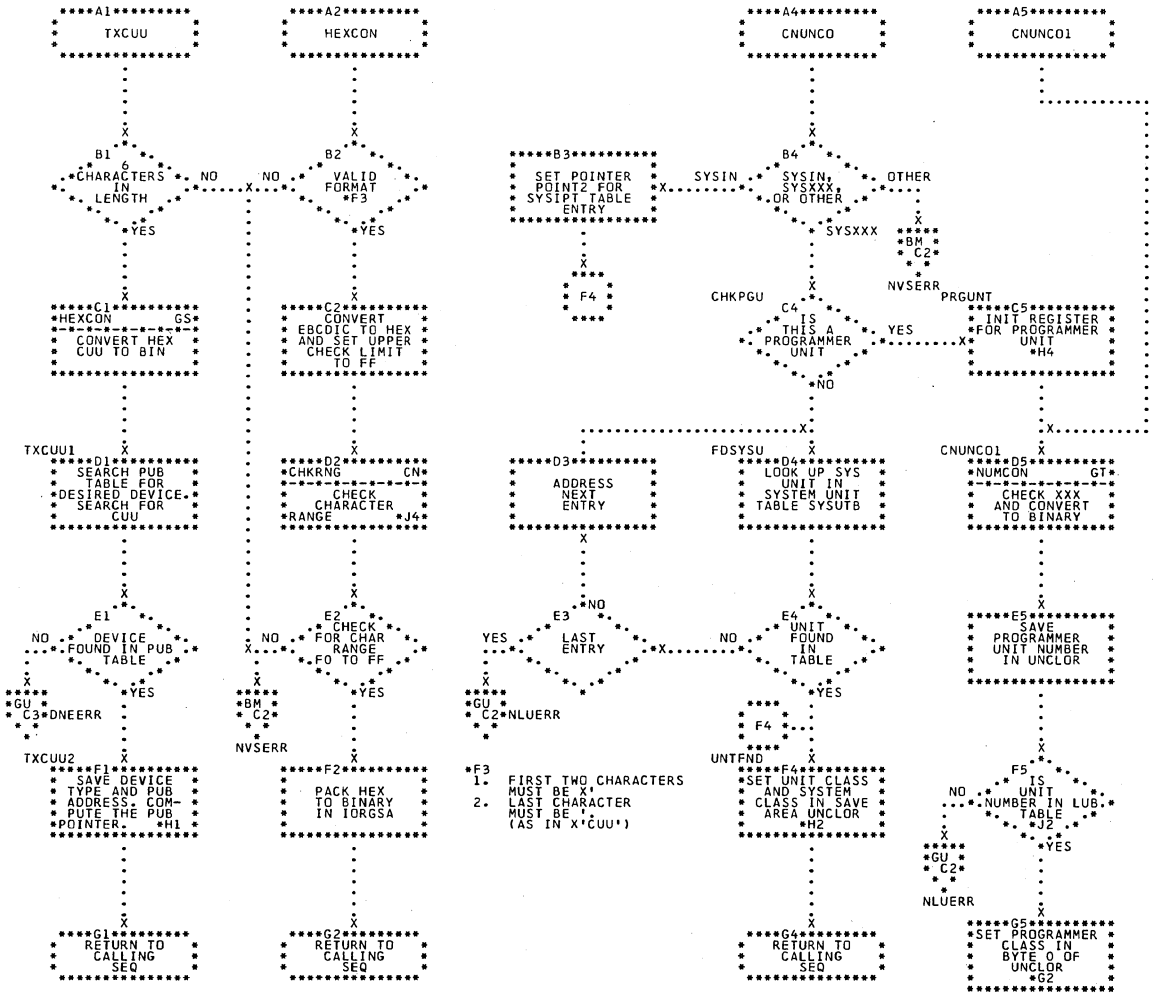


Chart GS. $JOBCTLJ - Miscellaneous Subroutines (Part 1 of 2)
Refer to Charts 11 and 12.



*H1
1. THE DEVICE TYPE IS SAVED IN DVCTYP FROM THE PUB.
2. THE PUB ADDRESS IS SAVED IN REGISTER POINT4.
3. THE PUB POINTER IS COMPUTED IN WRKRG3.

*H2
THE SYSTEM CLASS, NICL AND FICL POINTER, IS SET TO A 1 IN UNCLOR BYTE 0.

*H4
1. POINT1 IS INCREMENTED BY 3 TO POINT TO THE FIRST CHARACTER OF XXX.
2. POINT3 IS INITIALIZED TO A 2 AS A CHARACTER COUNT REGISTER.

*F3
1. FIRST TWO CHARACTERS MUST BE X'
2. LAST CHARACTER MUST BE ' (AS IN X'CUU')

*J2
1. UNCLOR, BYTE 1, THE UNIT CLASS, LUB POINTER, IS THE FUNCTION OF THE SYS ARGUMENT IN THE TABLE SYSUTB,
2. UNCLOR, BYTE 0, THE SYSTEM CLASS, NICL AND FICL POINTER, IS SET TO ZERO BECAUSE THE ROUTINE IS WORKING WITH A SYSTEM UNIT.

*J4
THE CODING FOR THIS ROUTINE ACTUALLY APPEARS IN THE $JOBCTLJ PHASE. THE CHART REFERENCED IN A DIFFERENT PHASE REPRESENTS LOGIC IDENTICAL TO THAT FOUND IN THE $JOBCTLJ PHASE.

*K2
CHECK ONLY LUBS FOR THIS PARTITION

Chart GT. $JOBCTLJ - Miscellaneous Subroutines (Part 2 of 2)
Refer to Charts 11 and 12.

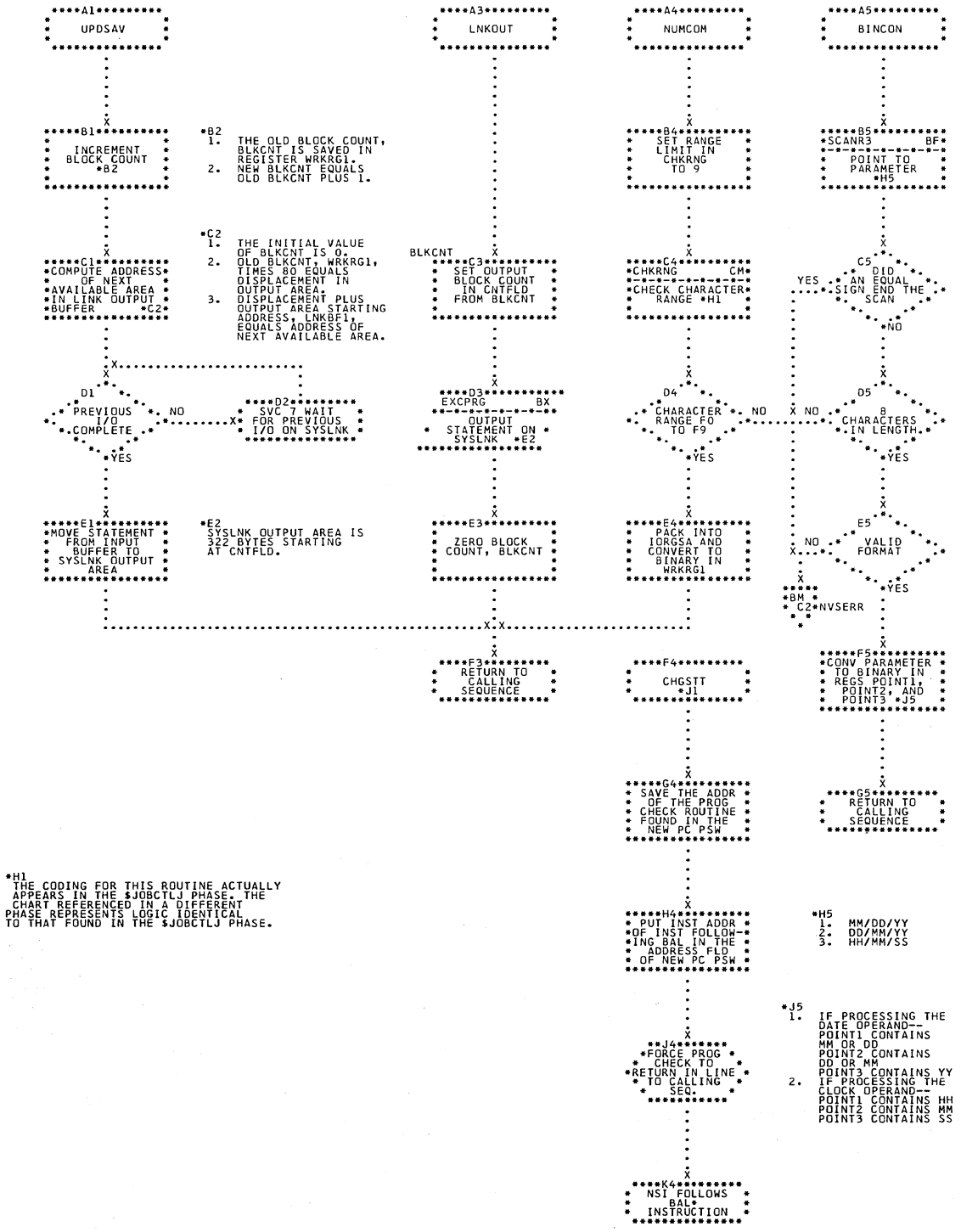
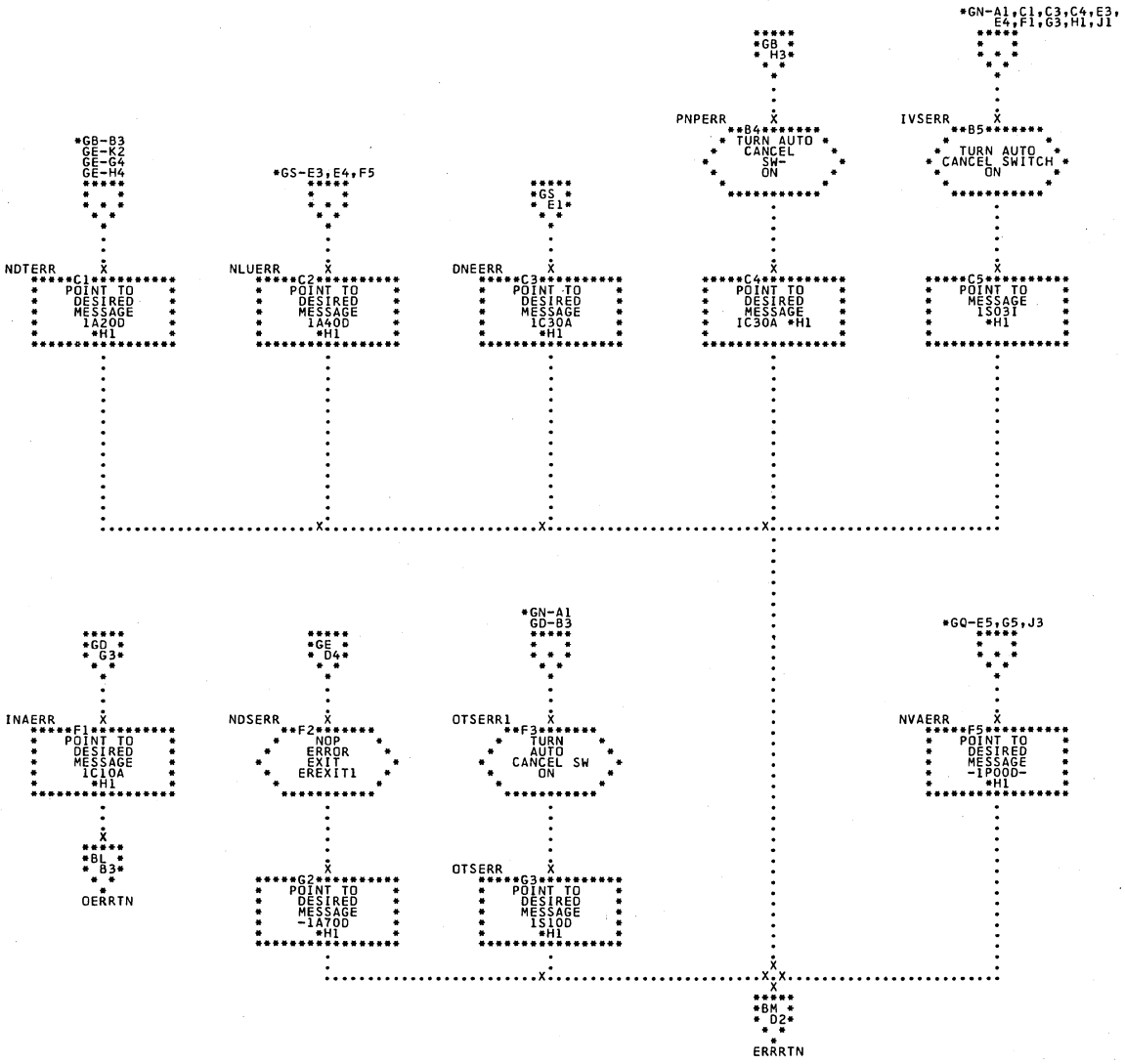


Chart GU. $JOBCTLJ - Error Subroutines
Refer to Charts 11 and 12.



*H1

MESSAGE	DESCRIPTION
1A20D	INVALID DEVICE TYPE
1A40D	INVALID LOGICAL UNIT SPECIFICATION
1A50D	DEVICE NOT DEFINED
1A70D	INVALID DEVICE STATUS
1C10A	PLEASE ASSIGN SYSRDR,SYSIPT,SYSLNK
1C30A	PROGRAM NOT FOUND
1P00D	INVALID ALLOCATION
1S03I	INVALID STATEMENT
1S10D	STATEMENT OUT OF SEQUENCE

Chart HA. $JOBCTLK - LBLTYP, VOL, & TPLAB Statement Processors
Refer to Chart 13.

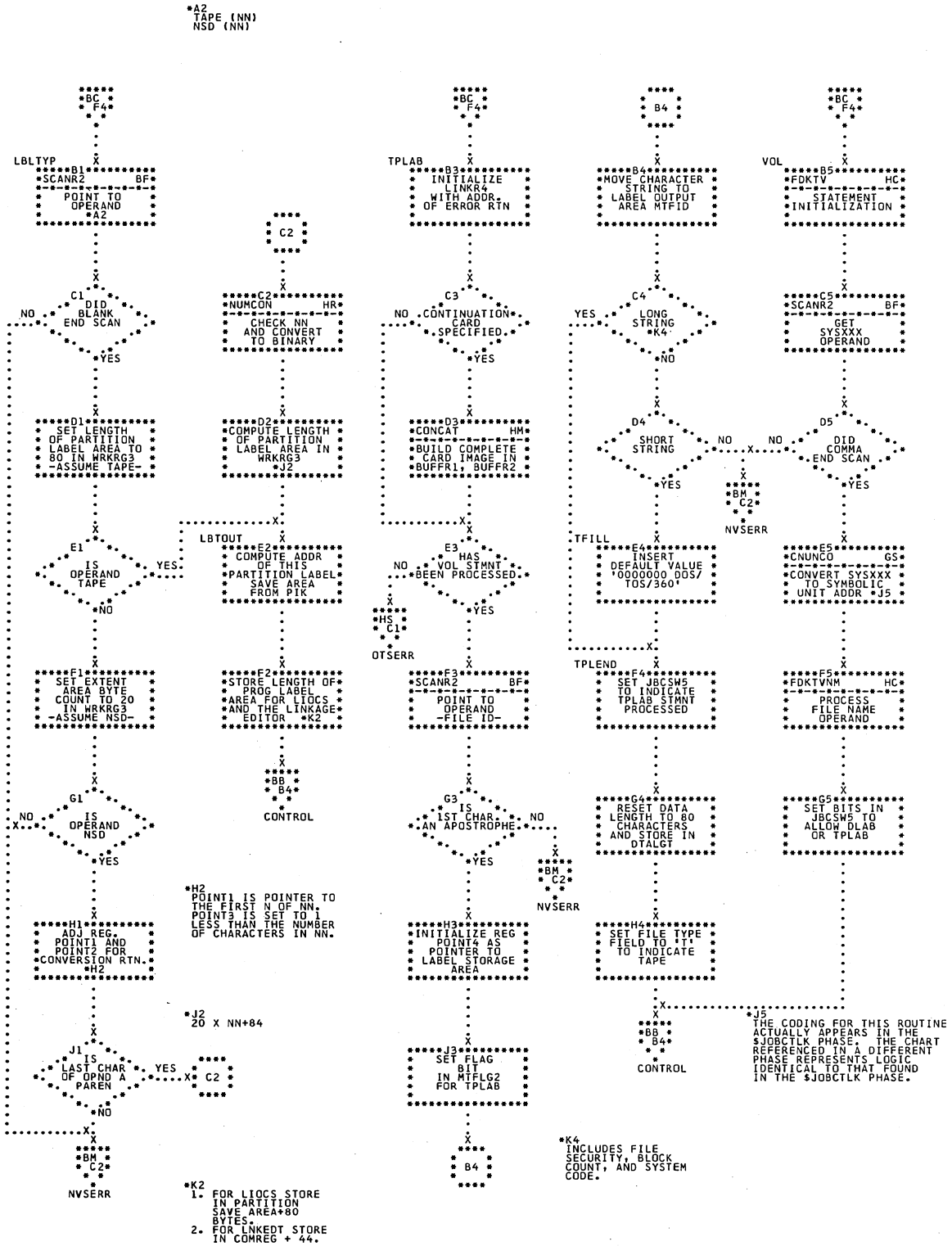


Chart HB. $JOBCTLK - TLBL Statement Processor
Refer to Chart 14.

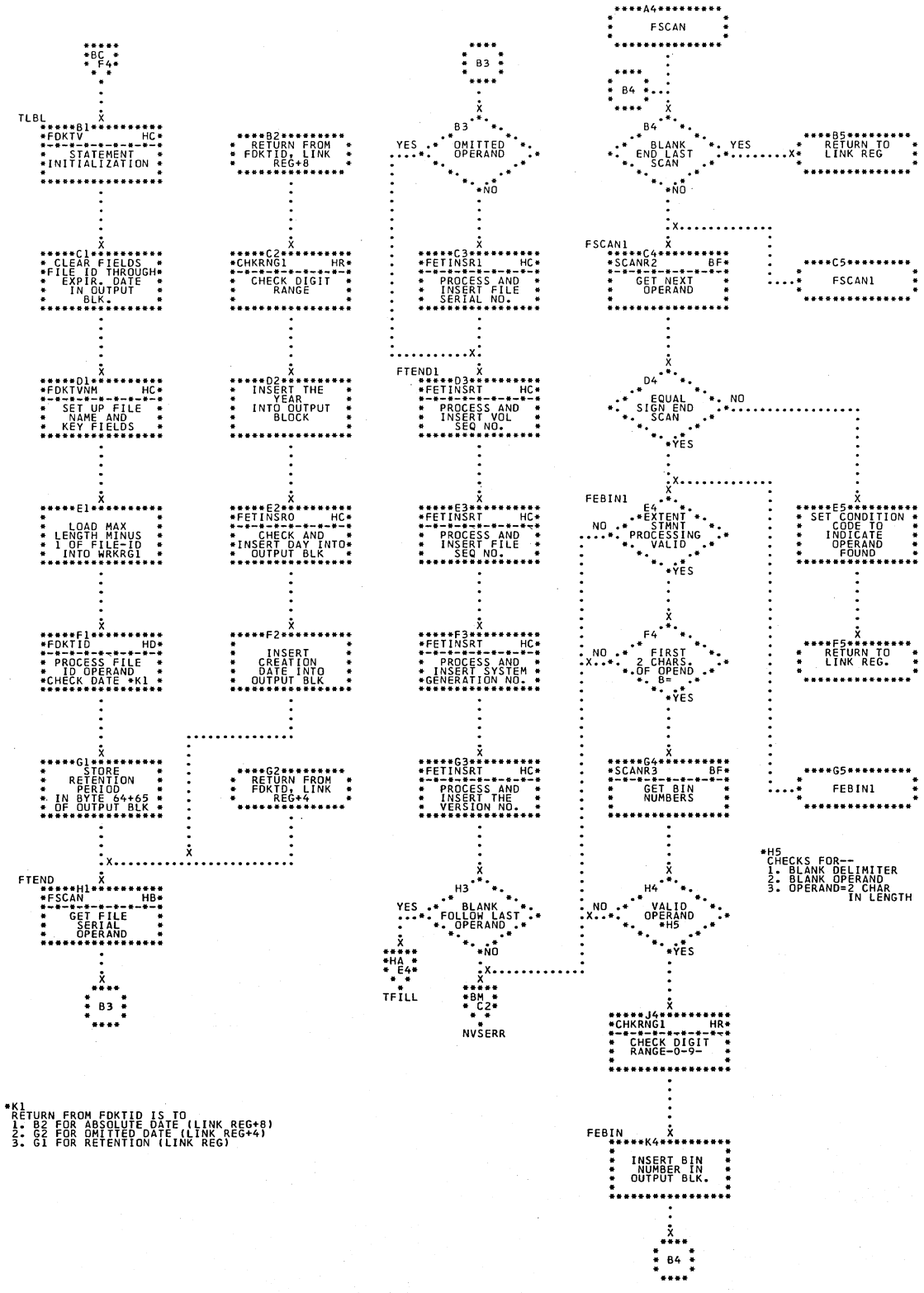


Chart HC. $JOBCTLK - Label Processing Subroutines (Part 1 of 2)
Refer to Charts 13 and 14.

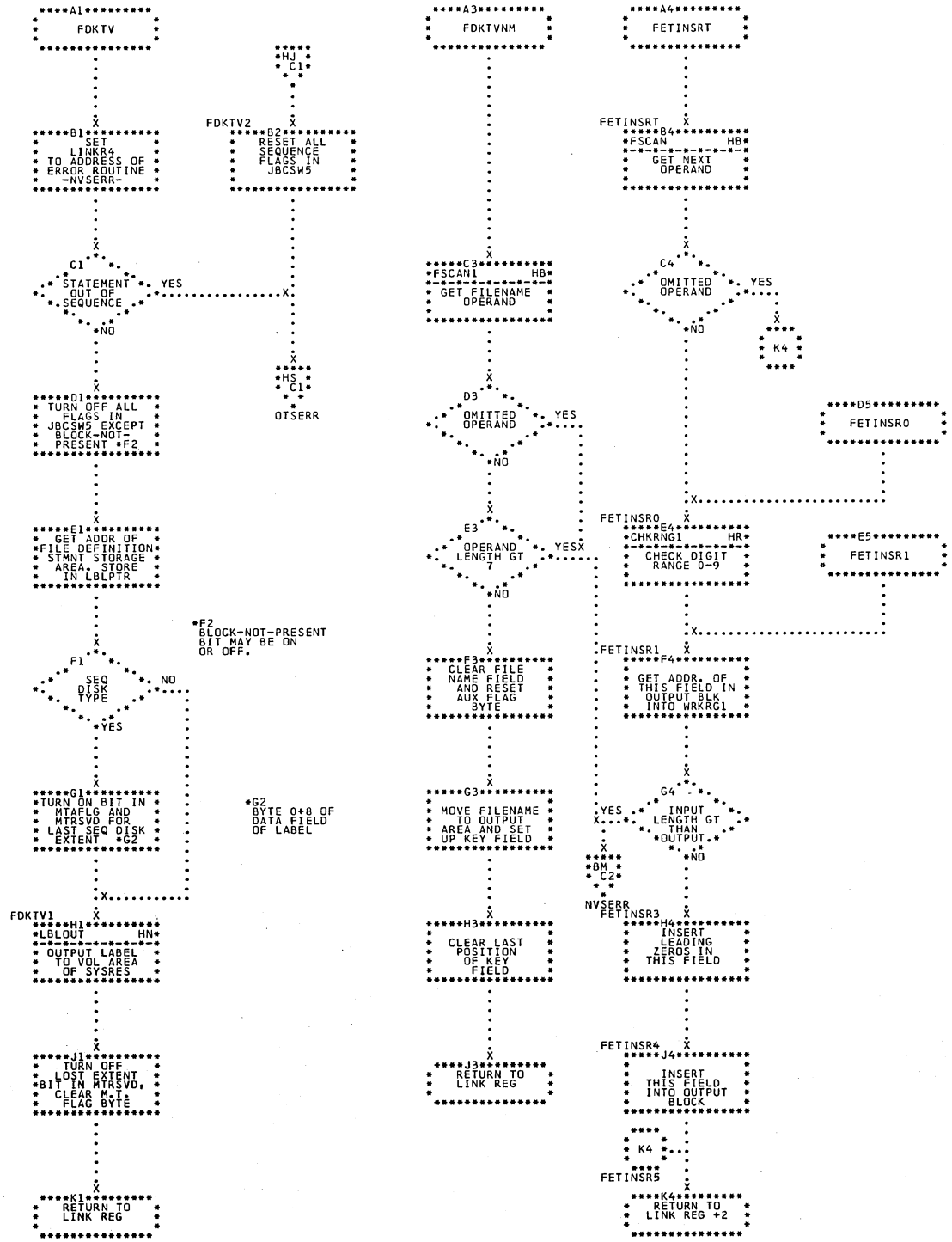


Chart HD. $JOBCTLK - Label Processing Subroutines (Part 2 of 2)
Refer to Charts 13 and 14.

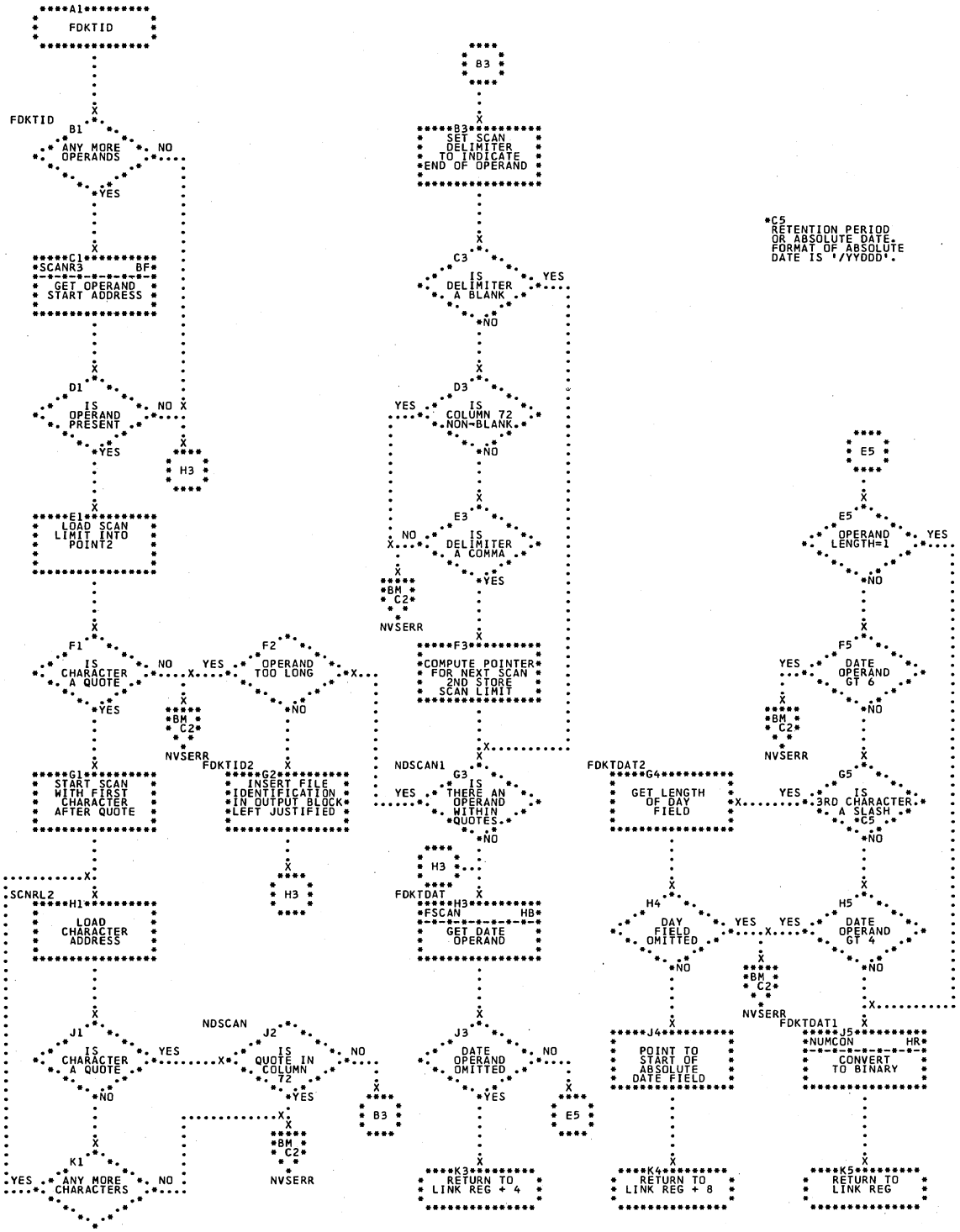


Chart HE. $JOBCTLK - DLBL Statement Processor
Refer to Chart 14.

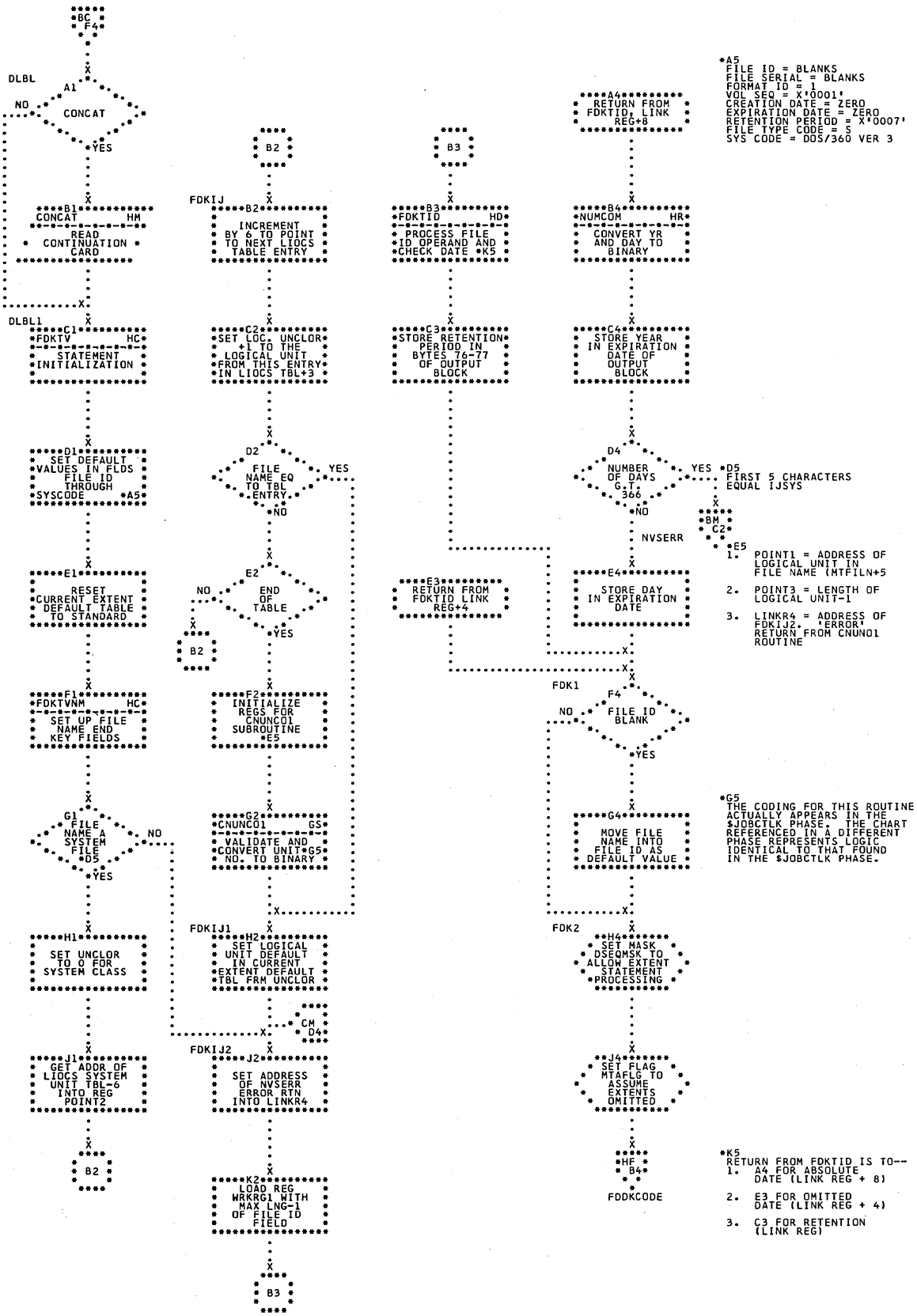


Chart HF. $JOBCTLK - DLAB Statement Processor
Refer to Chart 13.

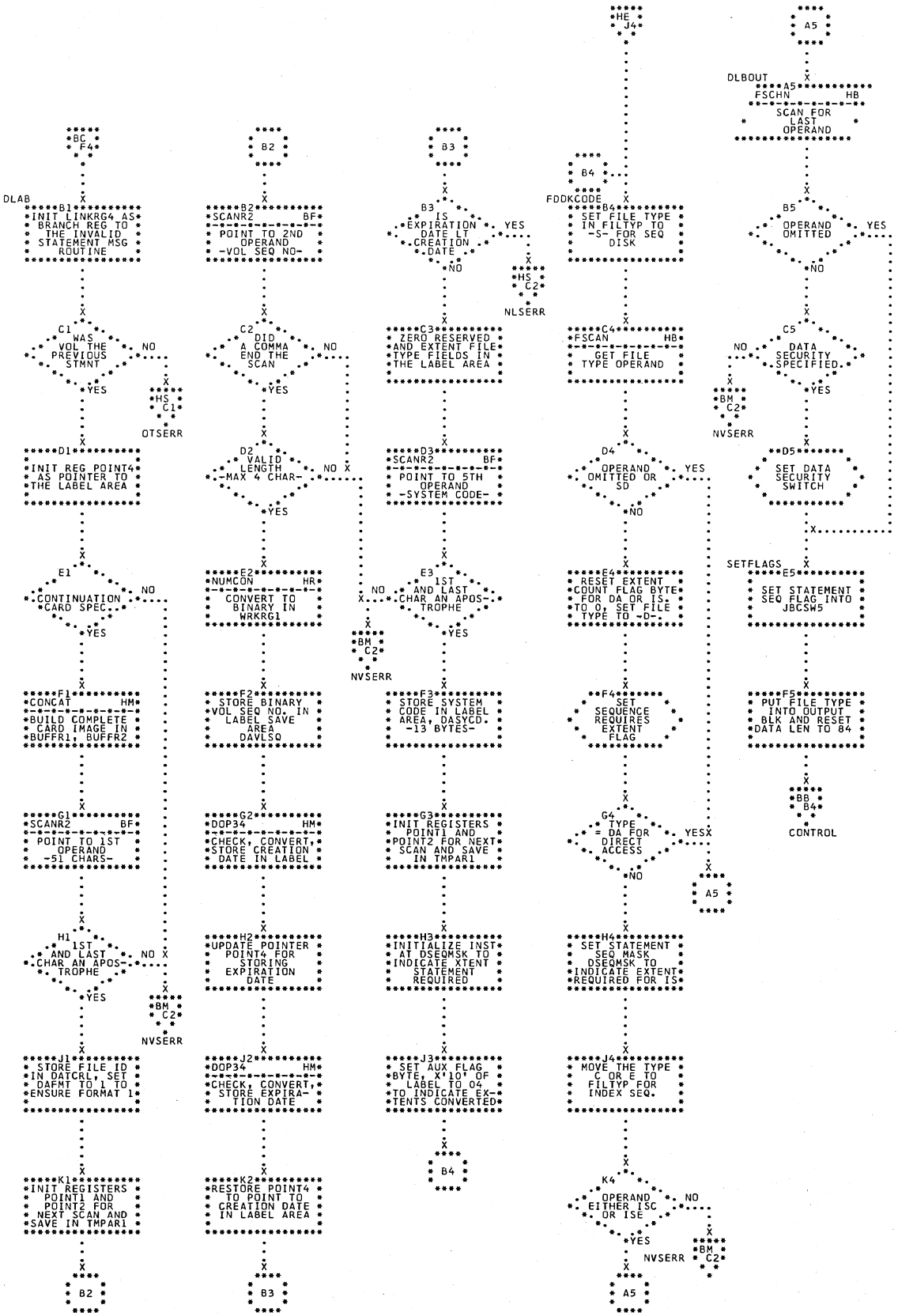


Chart HG. $JOBCTLK - XTENT Statement Processor (Part 1 of 2)
Refer to Chart 13.

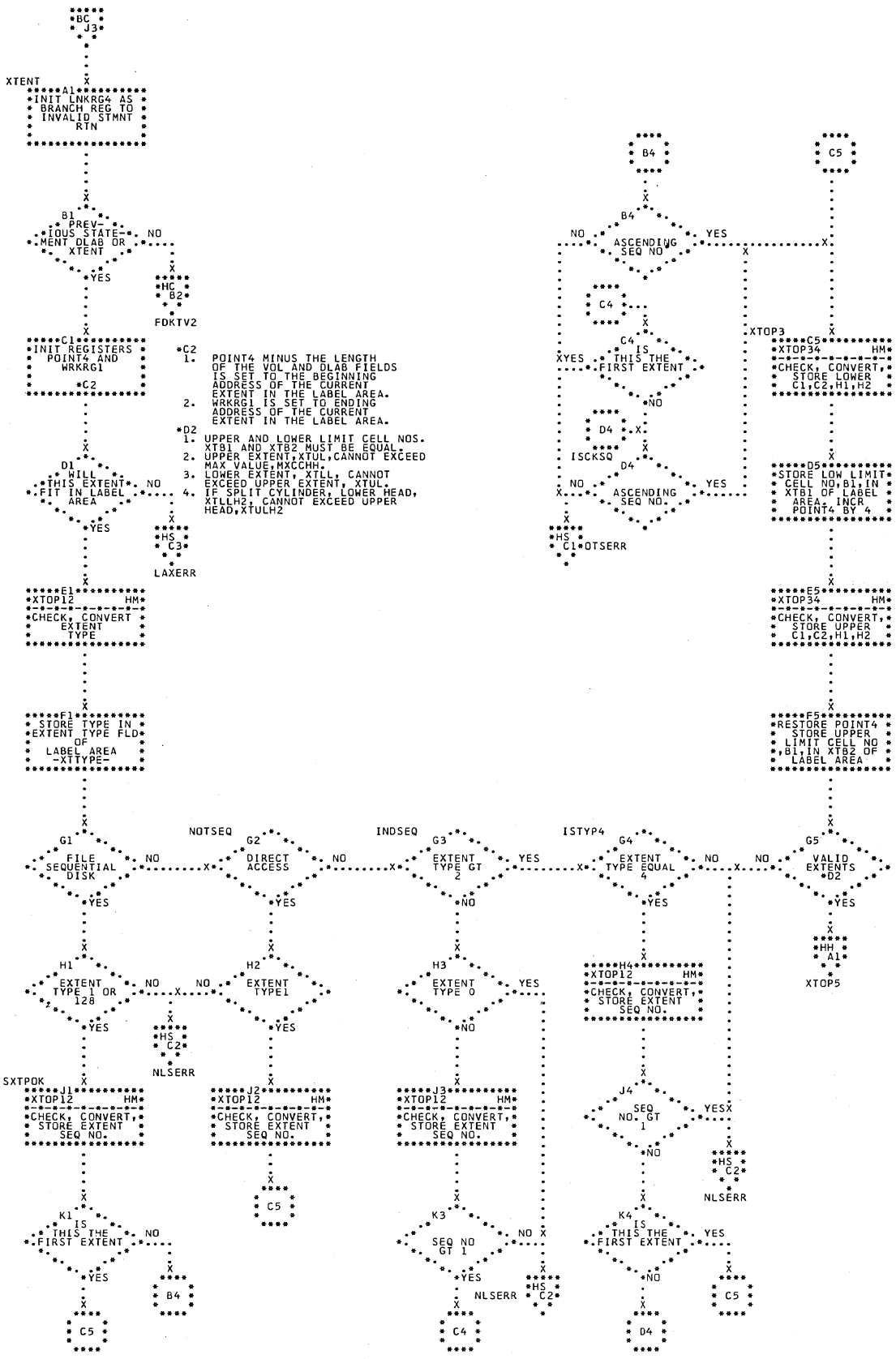


Chart HH. $JOBCTLK - XTENT Statement Processor (Part 2 of 2)
Refer to Chart 13.

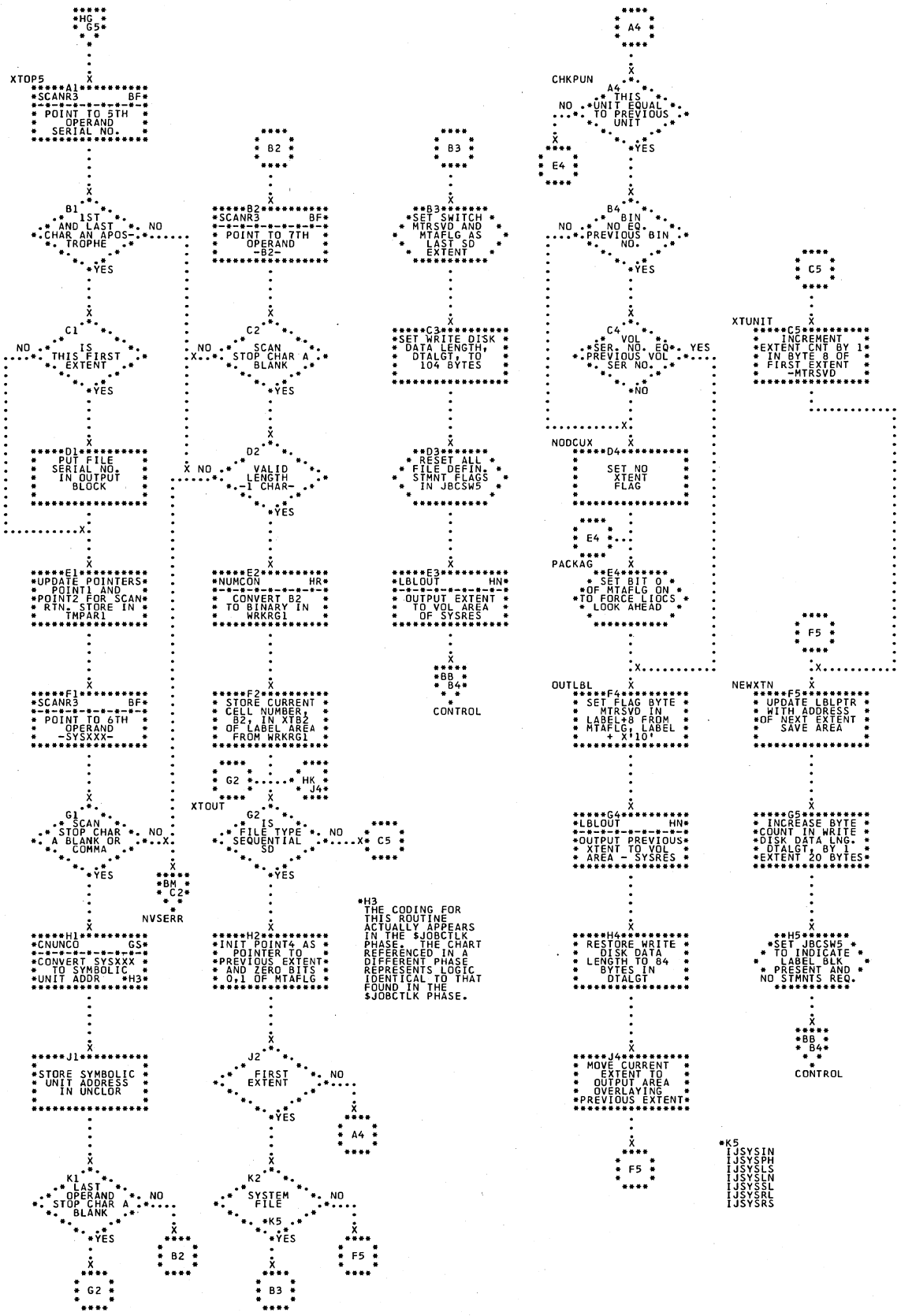


Chart HJ. $JOBCTLK - EXTENT Statement Processor (Part 1 of 3)
Refer to Chart 14.

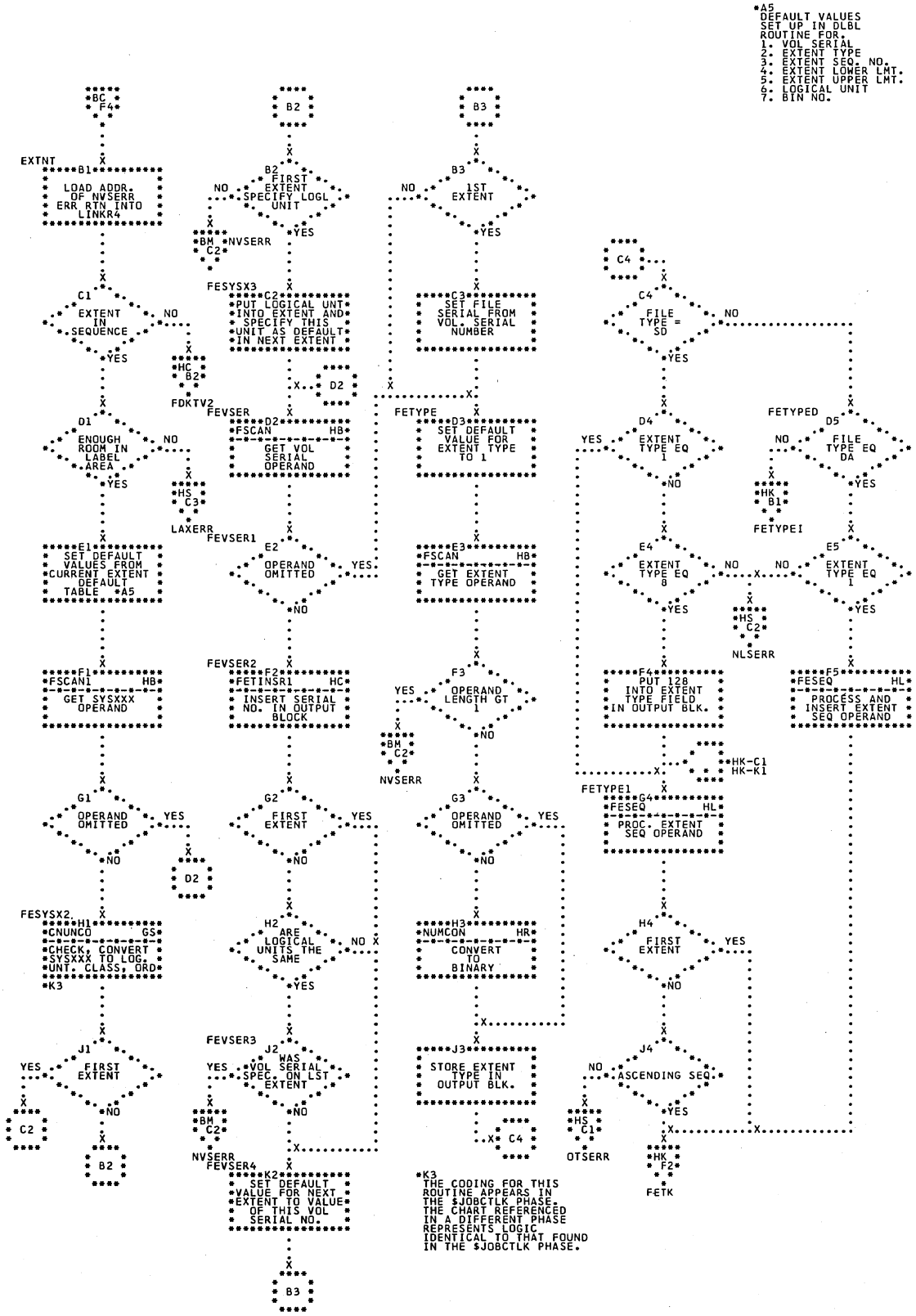


Chart HK. $JOBCTLK - EXTENT Statement Processor (Part 2 of 3)
Refer to Chart 14.

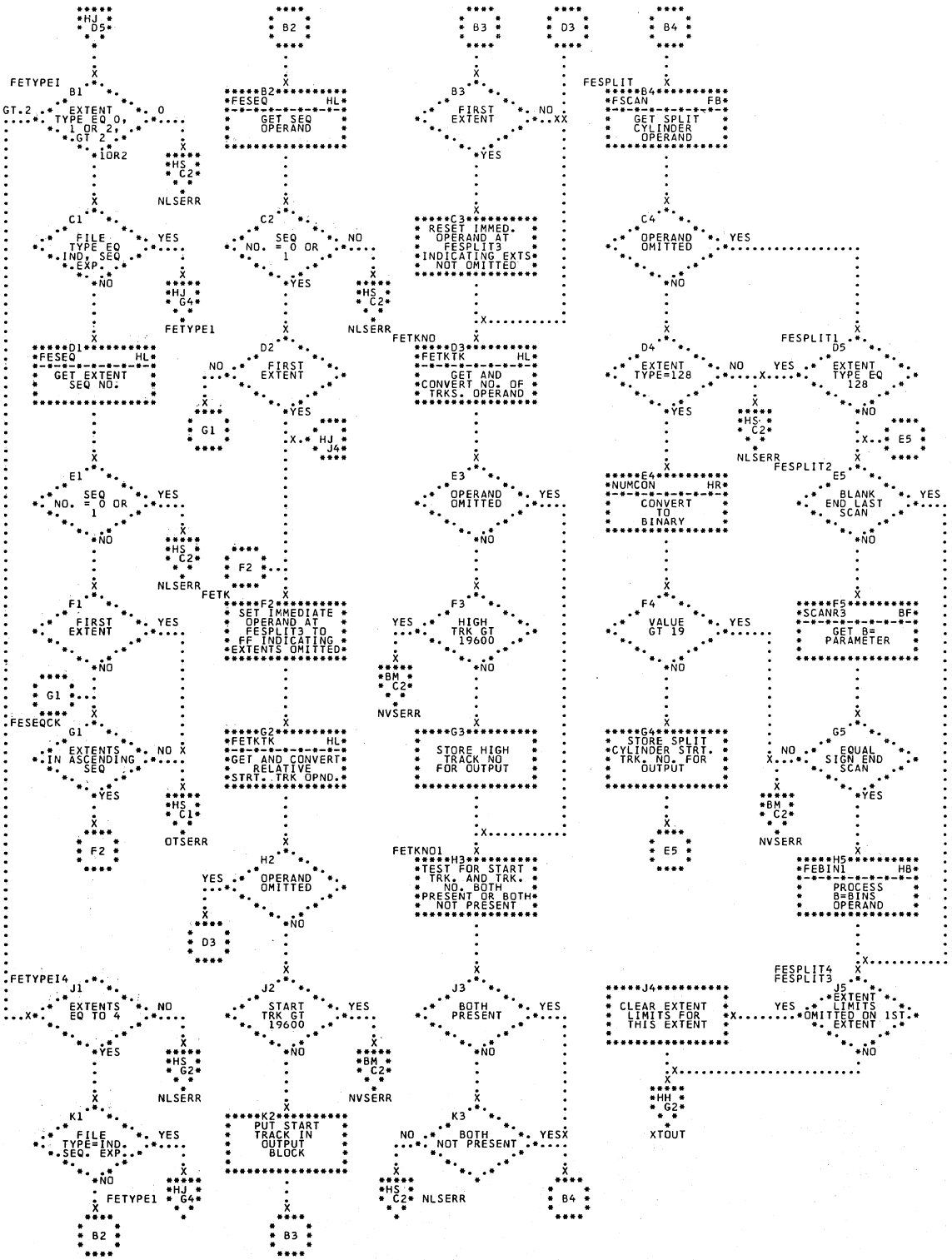


Chart HL. $JOBCTLK - EXTENT Statement Processor (Part 3 of 3)
Refer to Chart 14.

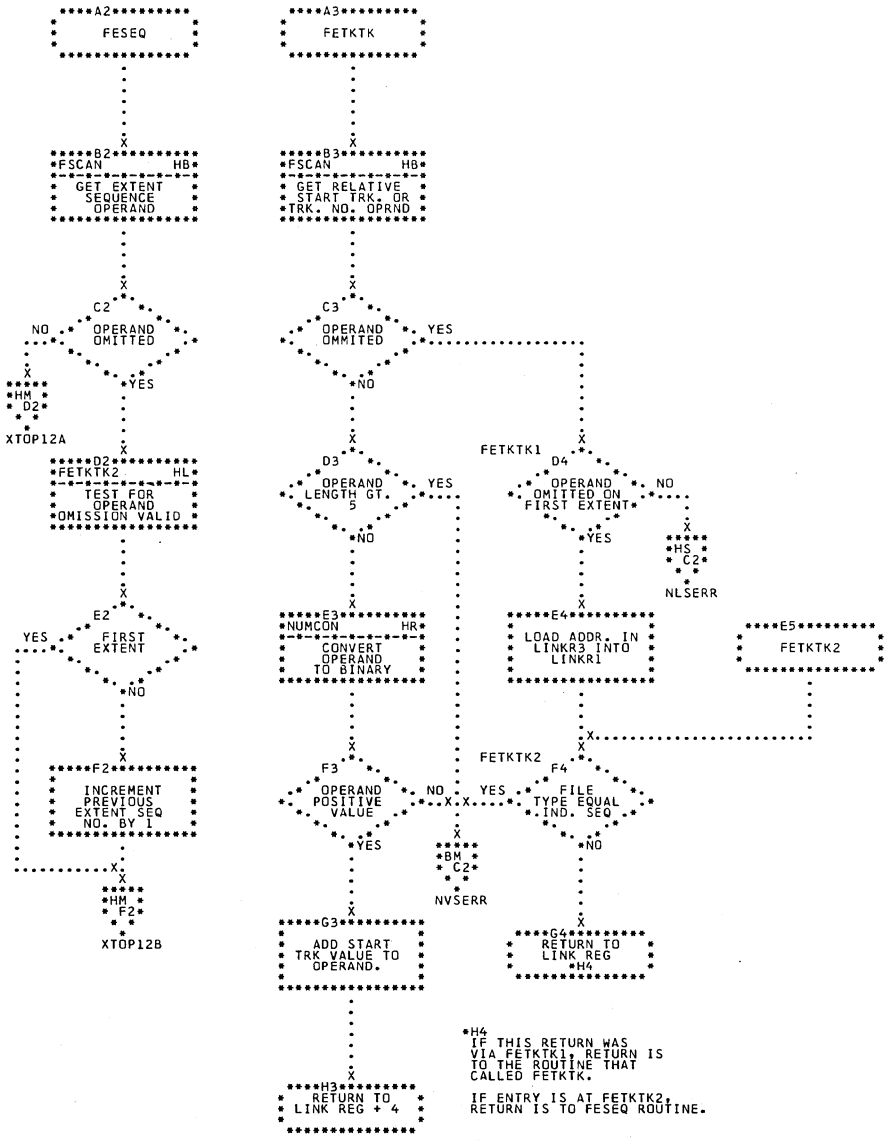


Chart HM. $JOBCTLK - Label Processing Subroutines (Part 1 of 3)
Refer to Charts 13 and 14.

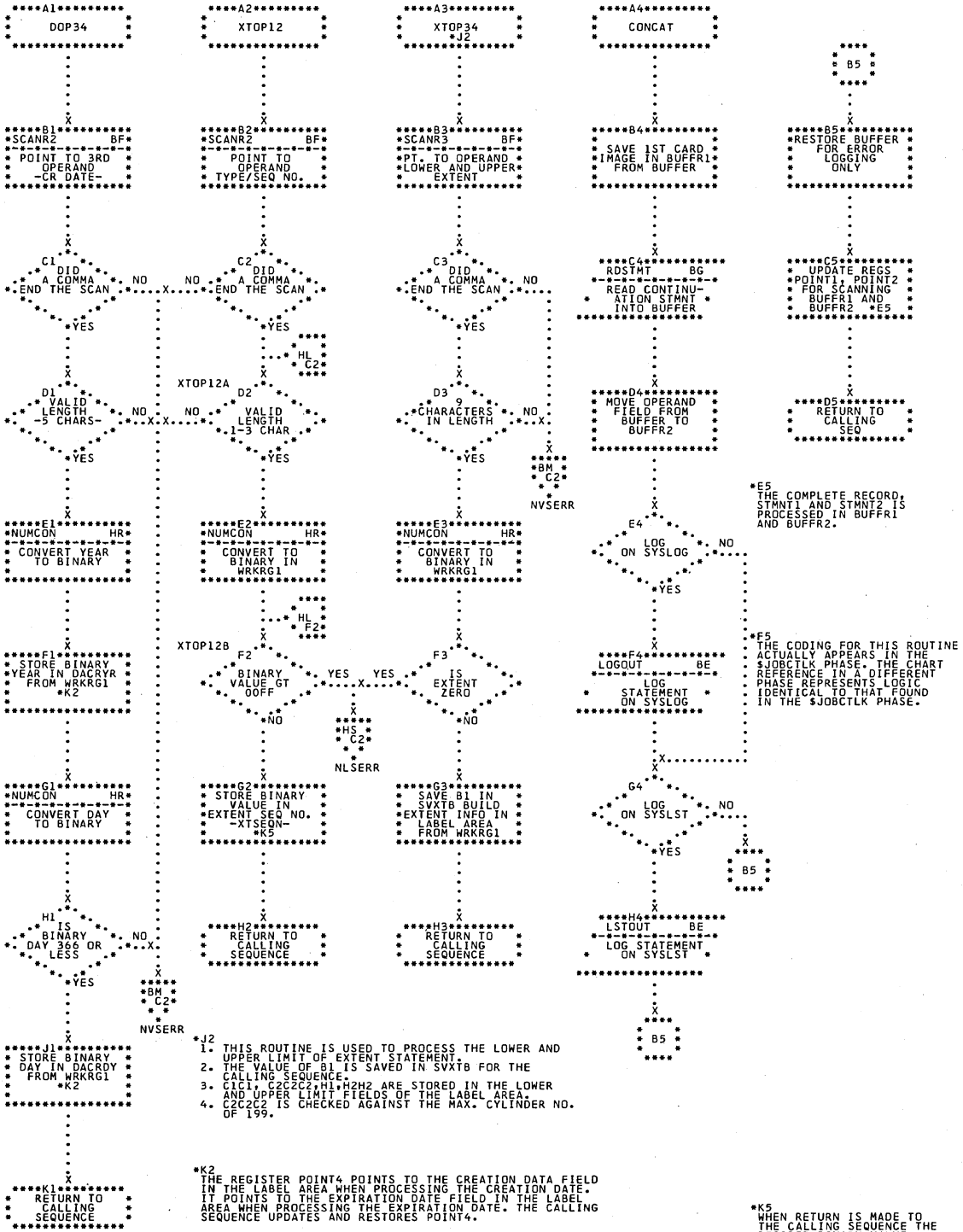


Chart HN. $JOBCTLK - Label Processing Subroutines (Part 2 of 3)
Refer to Charts 13 and 14.

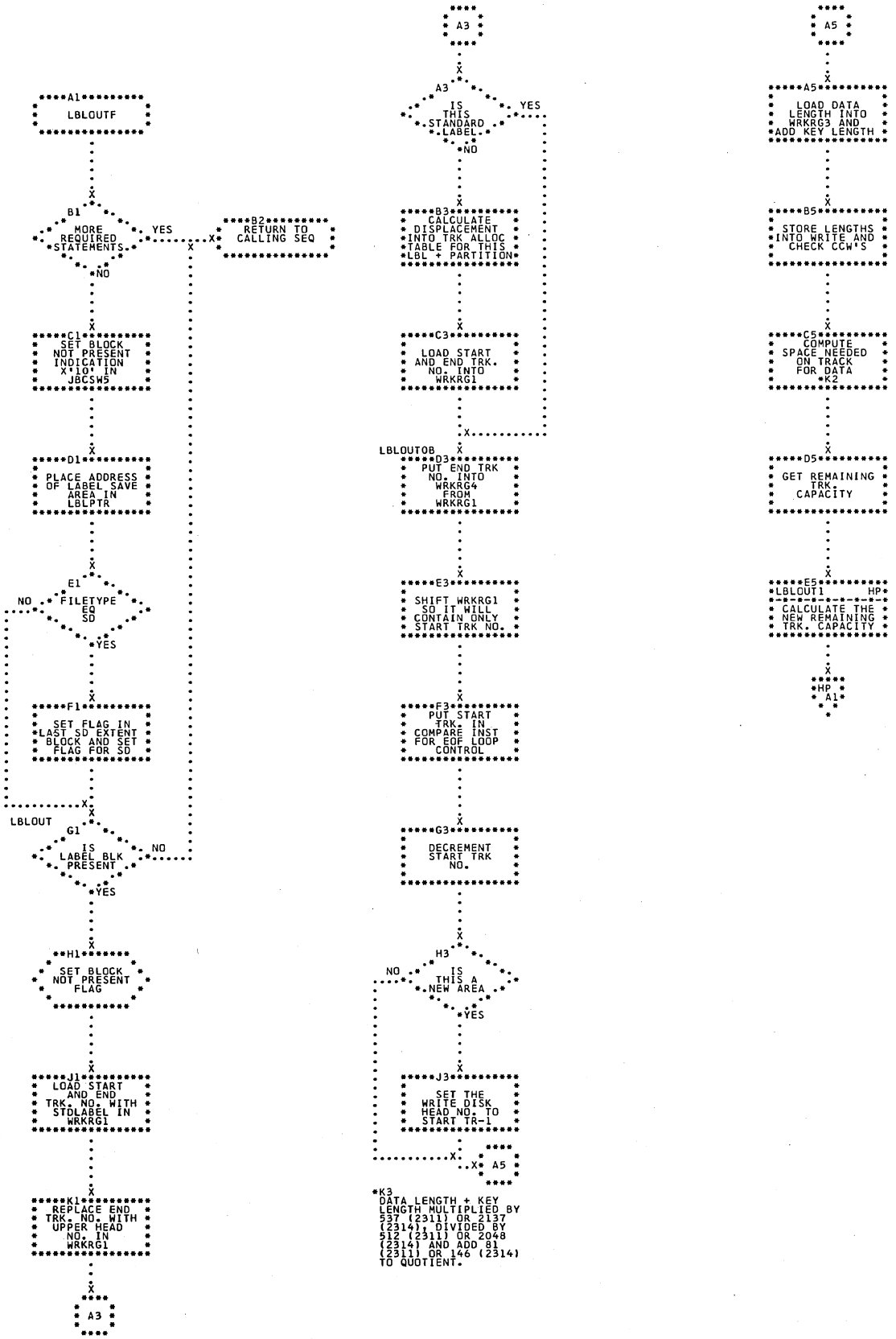


Chart HP. $JOBCTLK - Label Processing Subroutines (Part 3 of 3)
Refer to Charts 13 and 14.

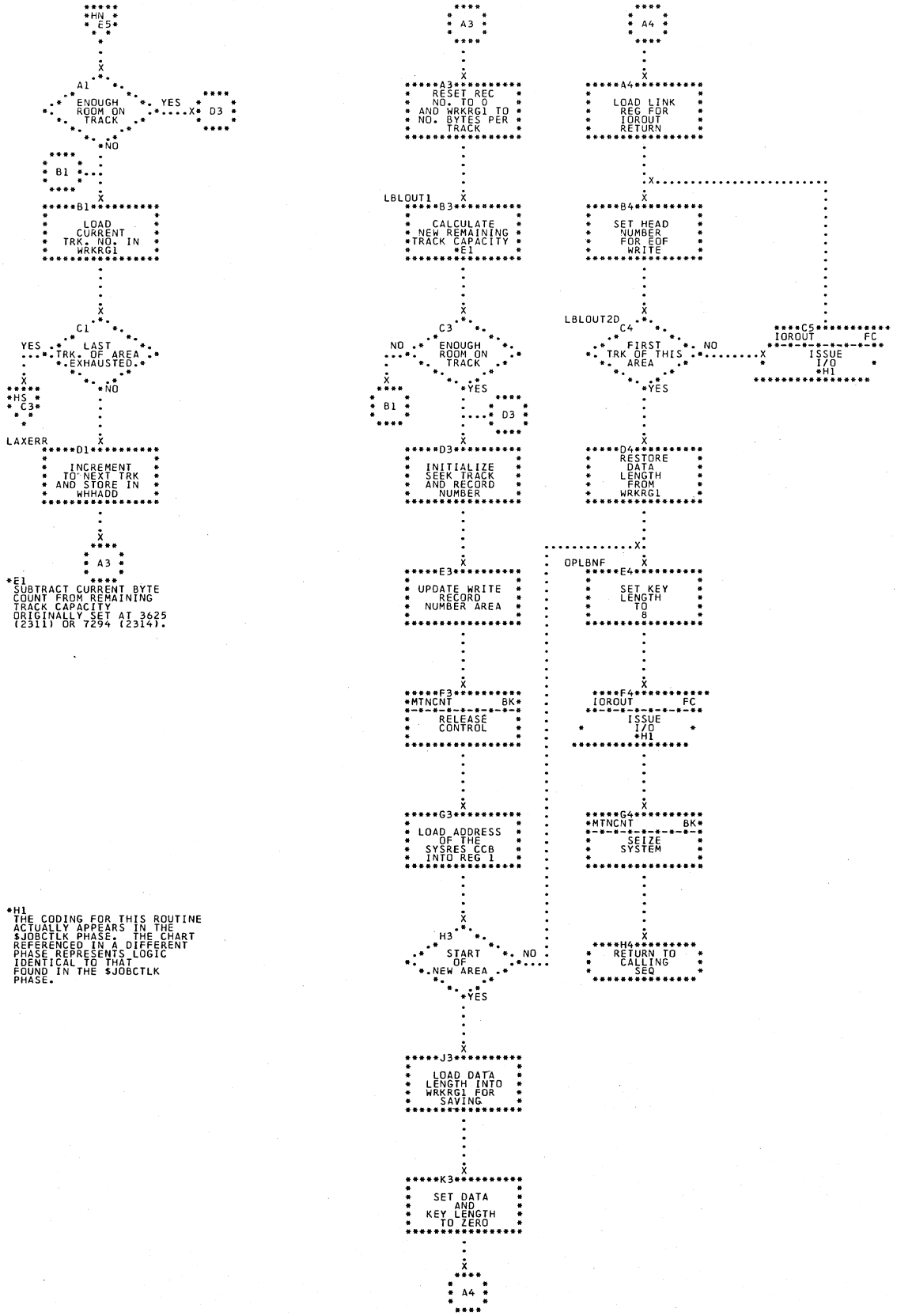


Chart HQ. $JOBCTLK - RSTRT Statement Processor
Refer to Chart 14.

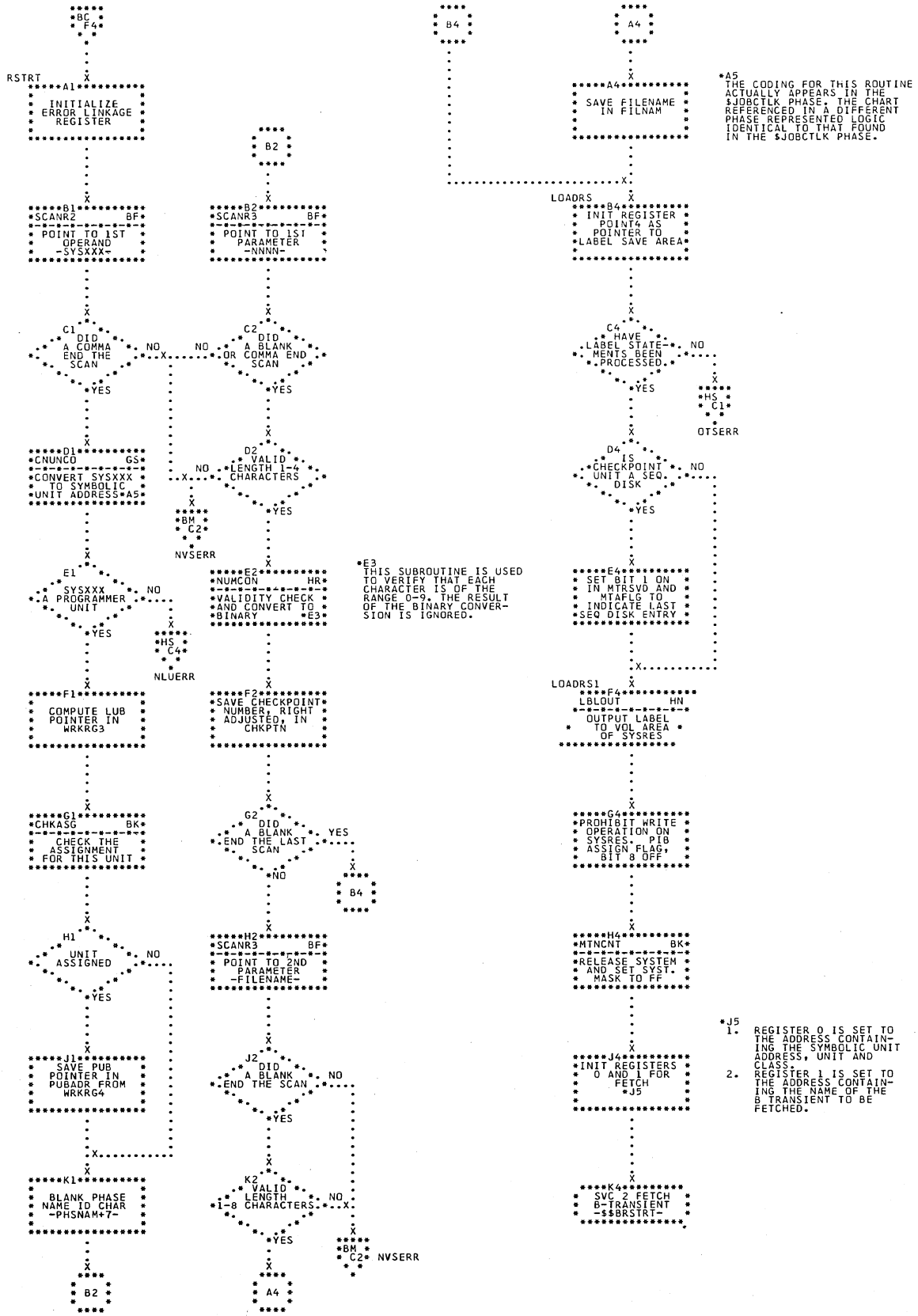


Chart HR. $JOBCTLK - Miscellaneous Subroutines
Refer to Charts 13 and 14.

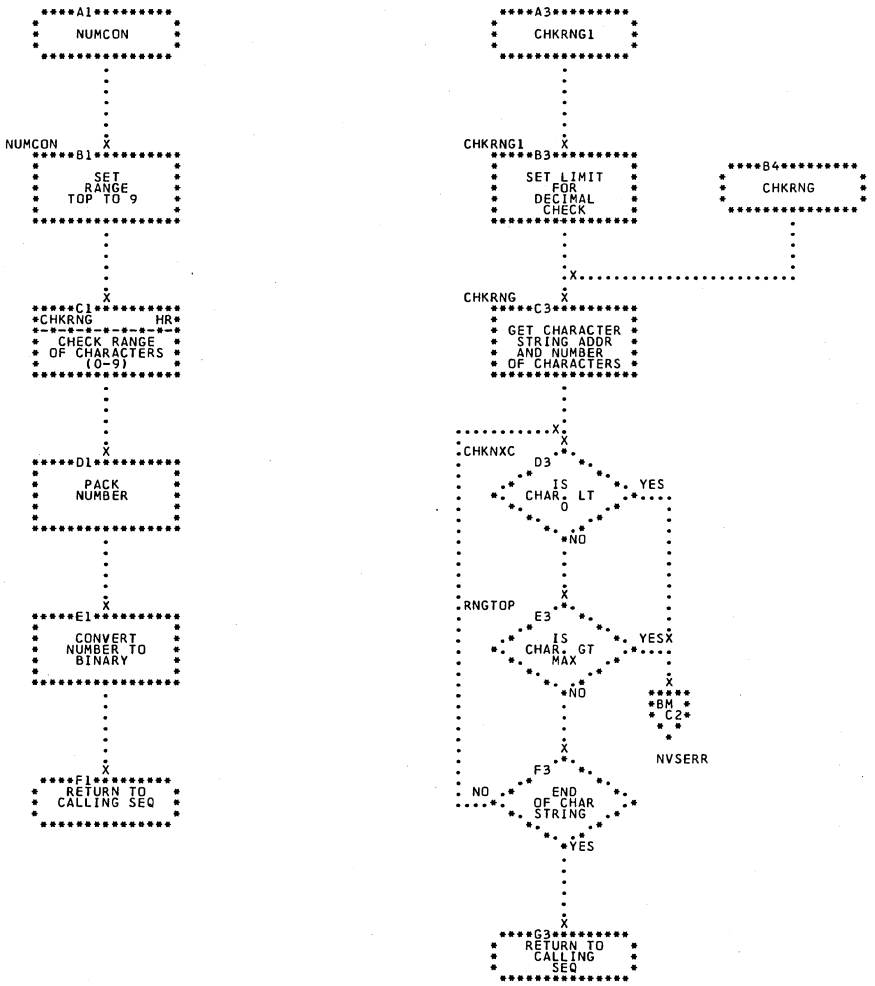
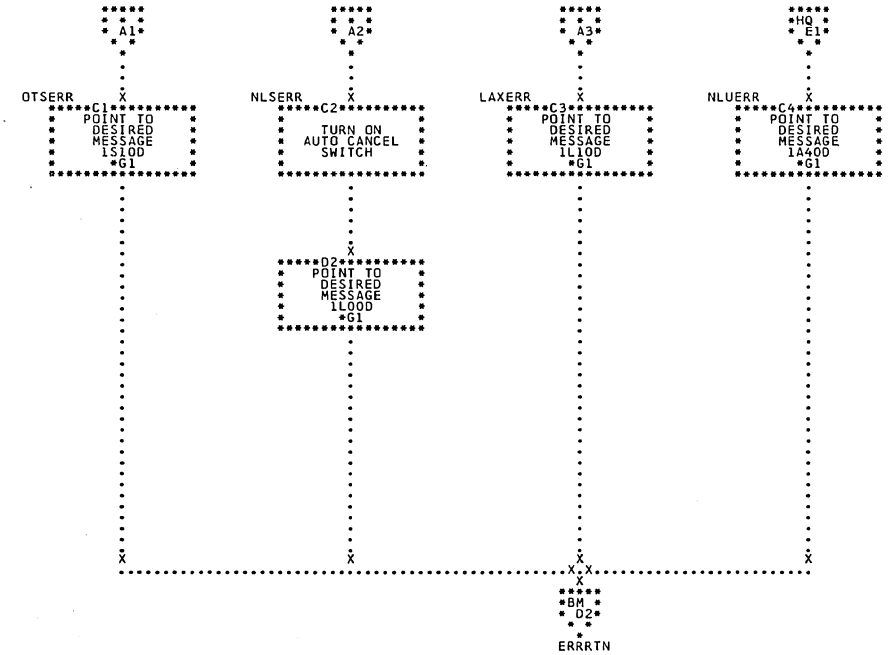


Chart HS. $JOBCTLK - Error Subroutines
Refer to Charts 13 and 14.

*A1
HA-E3
HC-D2
HF-C1
HG-B4,C4,D4
HJ-J4
HK-F1,G1
HP-C4

*A2
HF-B3
HG-G4,G5,H1,H2,H3,J4,K3
HJ-E4,E5
HK-B1,C2,D4,D5,E1,J1,K3
HL-D4
HM-F2,F3

*A3
HG-D1
HJ-D1
HP-C1



*G1

MESSAGE	DESCRIPTION
1A40D	INVALID LOGICAL UNIT SPECIFICATION
1L00D	INVALID LABEL SYNTAX
1L10D	LABEL AREA EXHAUSTED
1S10D	STATEMENT OUT OF SEQUENCE

Chart JA. $JOBCTLM - Recorder File Initialization
Refer to Chart 15.

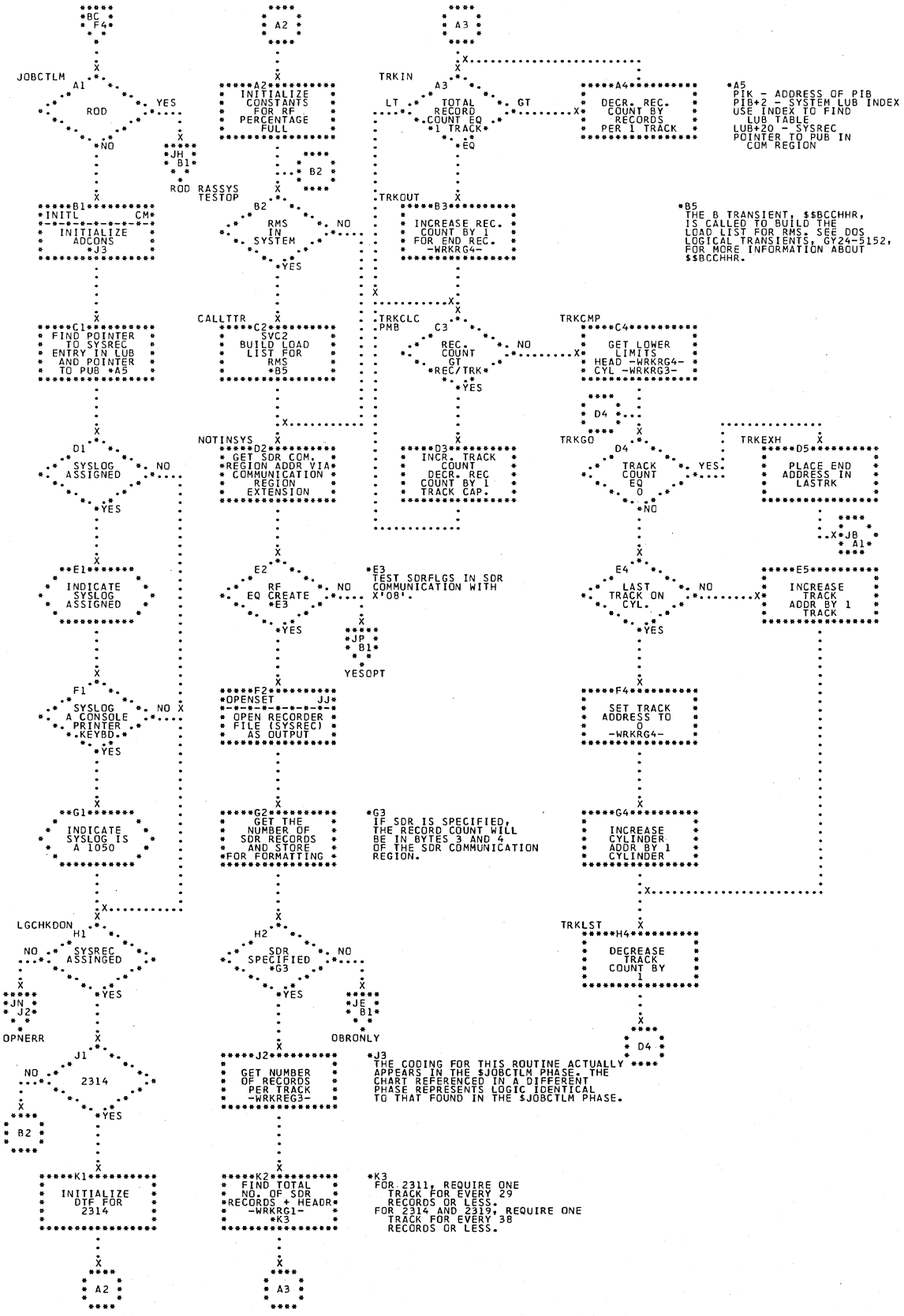


Chart JB. $JOBCTLM - Create Recorder File (Part 1 of 2)
Refer to Chart 15.

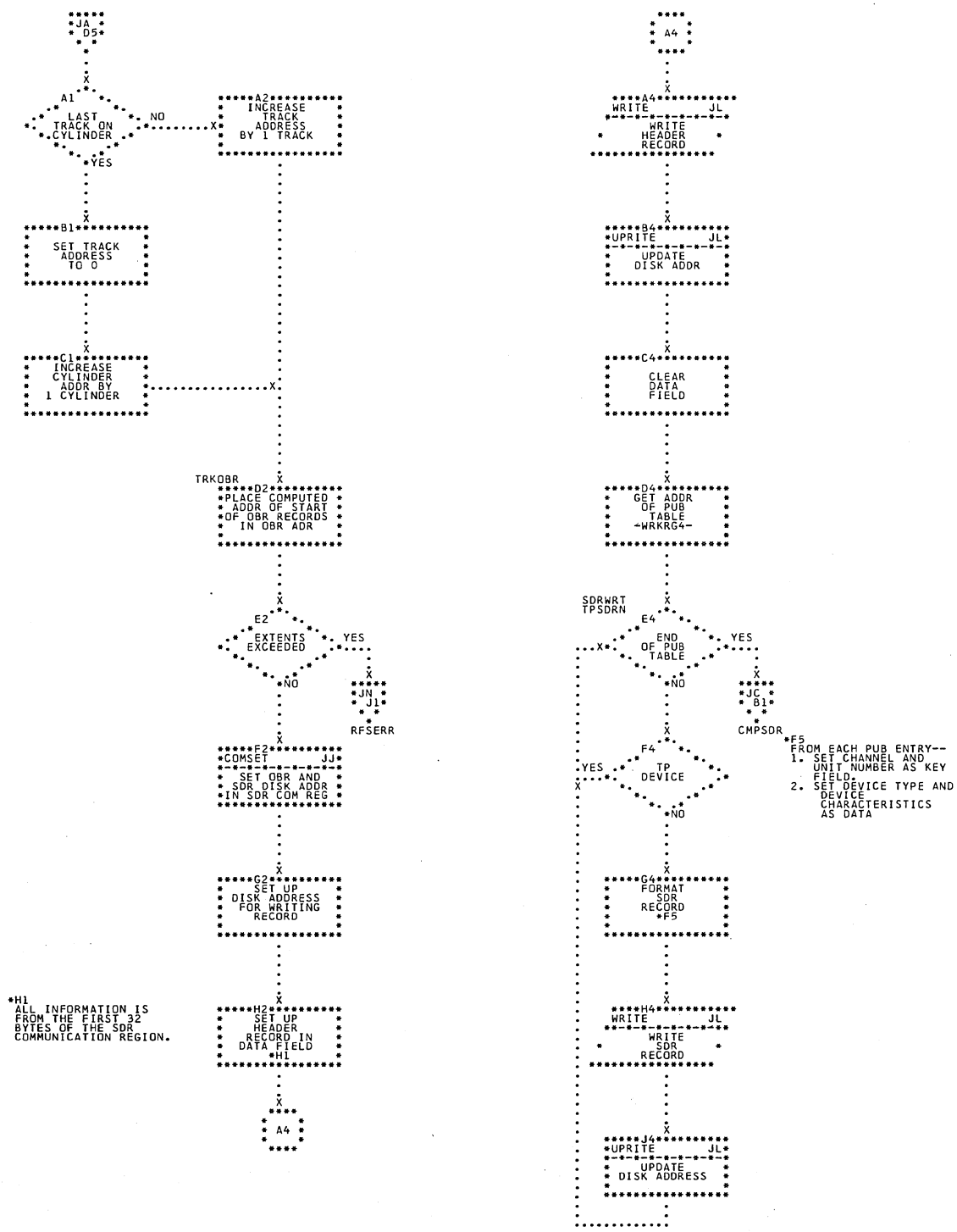


Chart JC. $JOBCTLM - Create Recorder File (Part 2 of 2)
Refer to Chart 15.

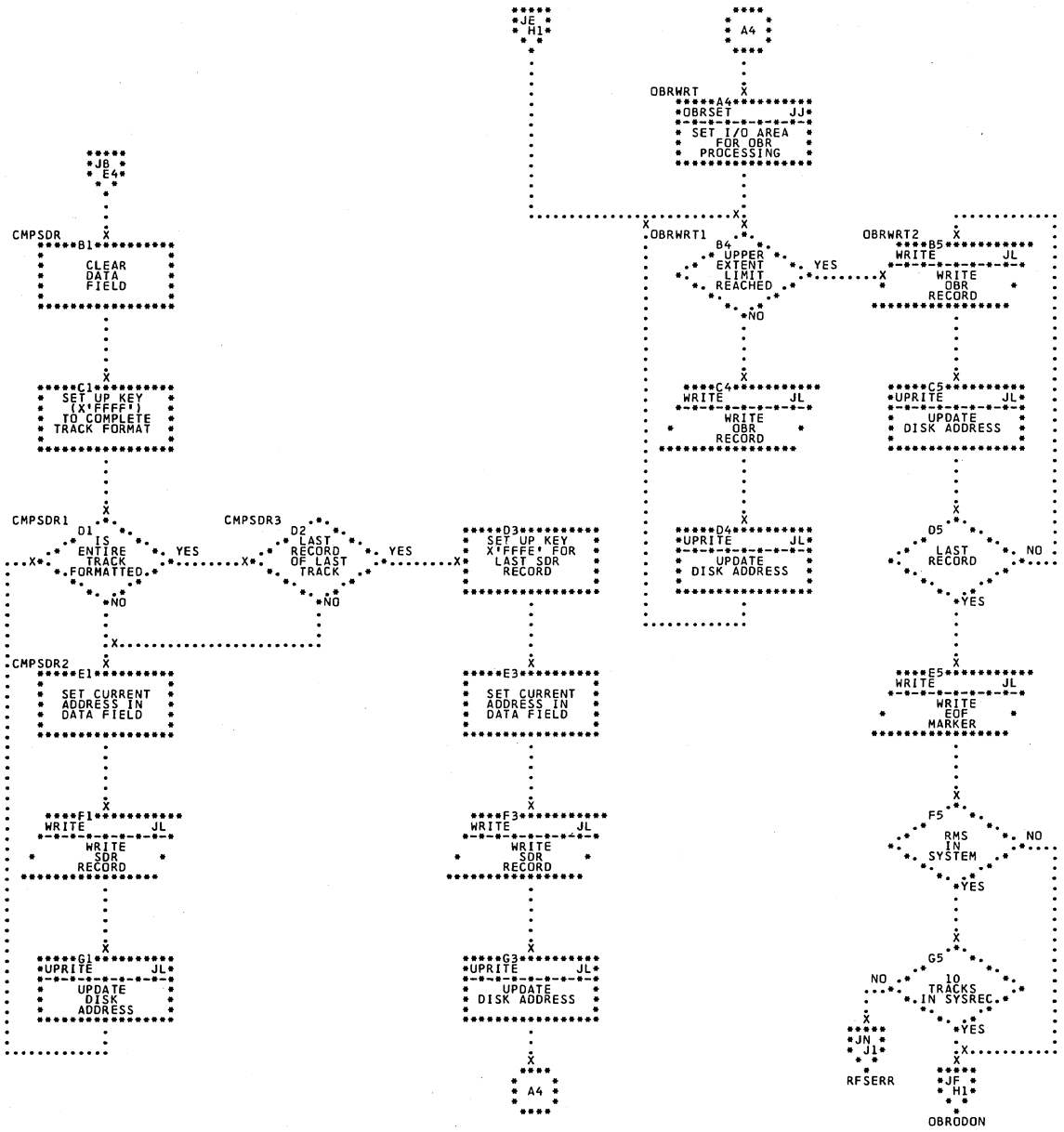


Chart JD. $JOBCTLM - Check Recorder File (Part 1 of 4)
Refer to Chart 15.

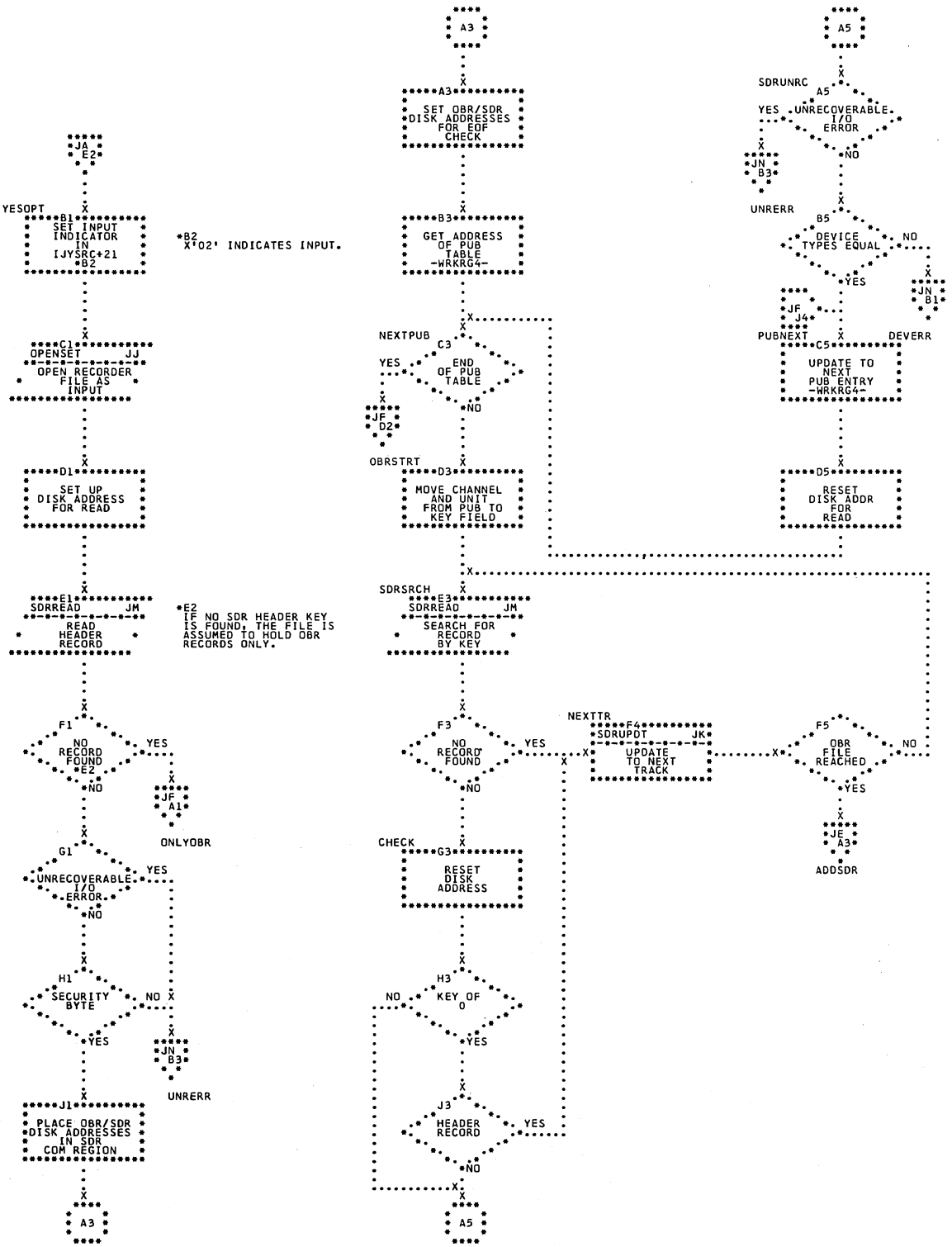


Chart JE. $JOBCTLM - Check Recorder File (Part 2 of 4)
Refer to Chart 15.

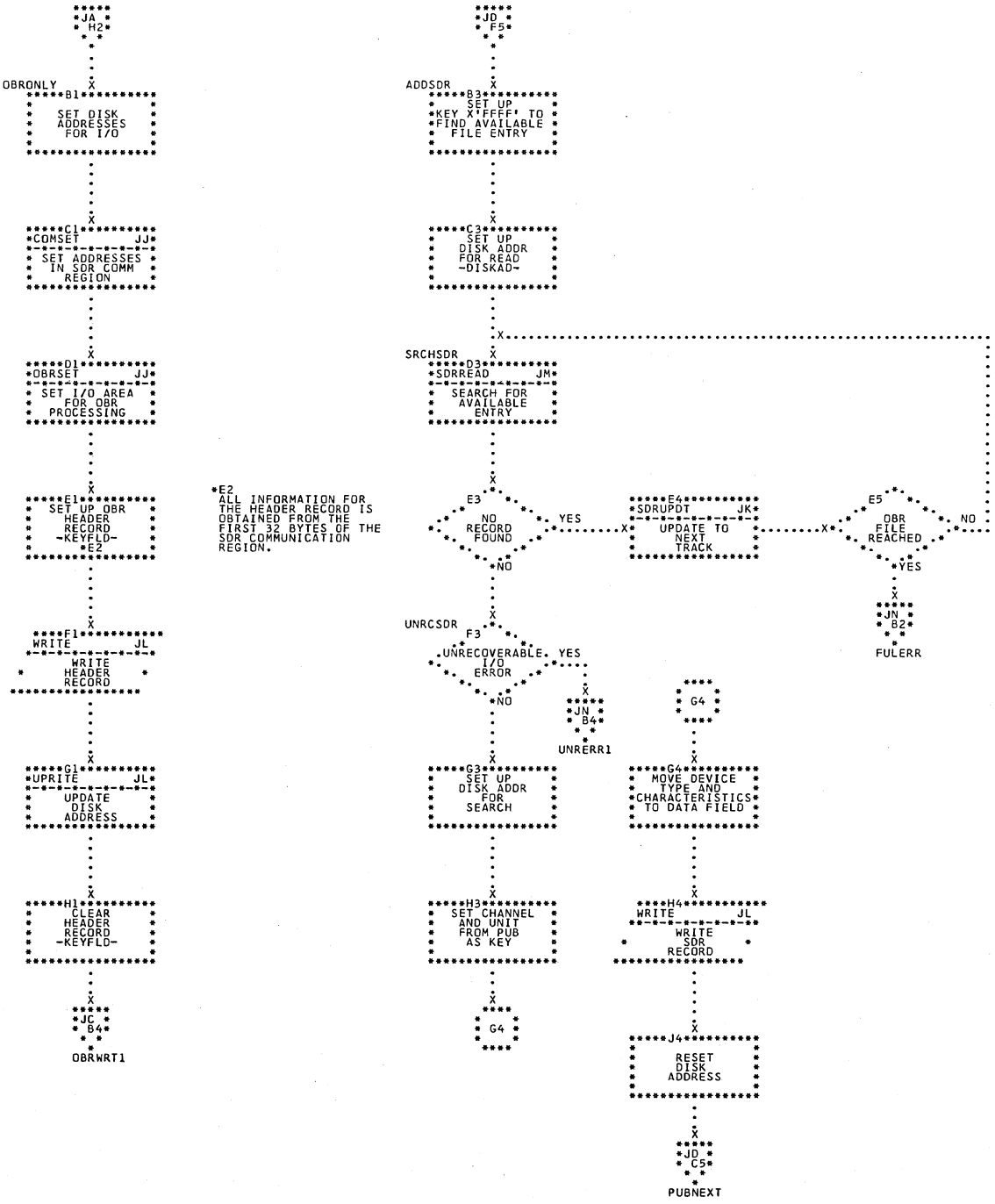


Chart JF. $JOBCTLM - Check Recorder File (Part 3 of 4)
Refer to Chart 15.

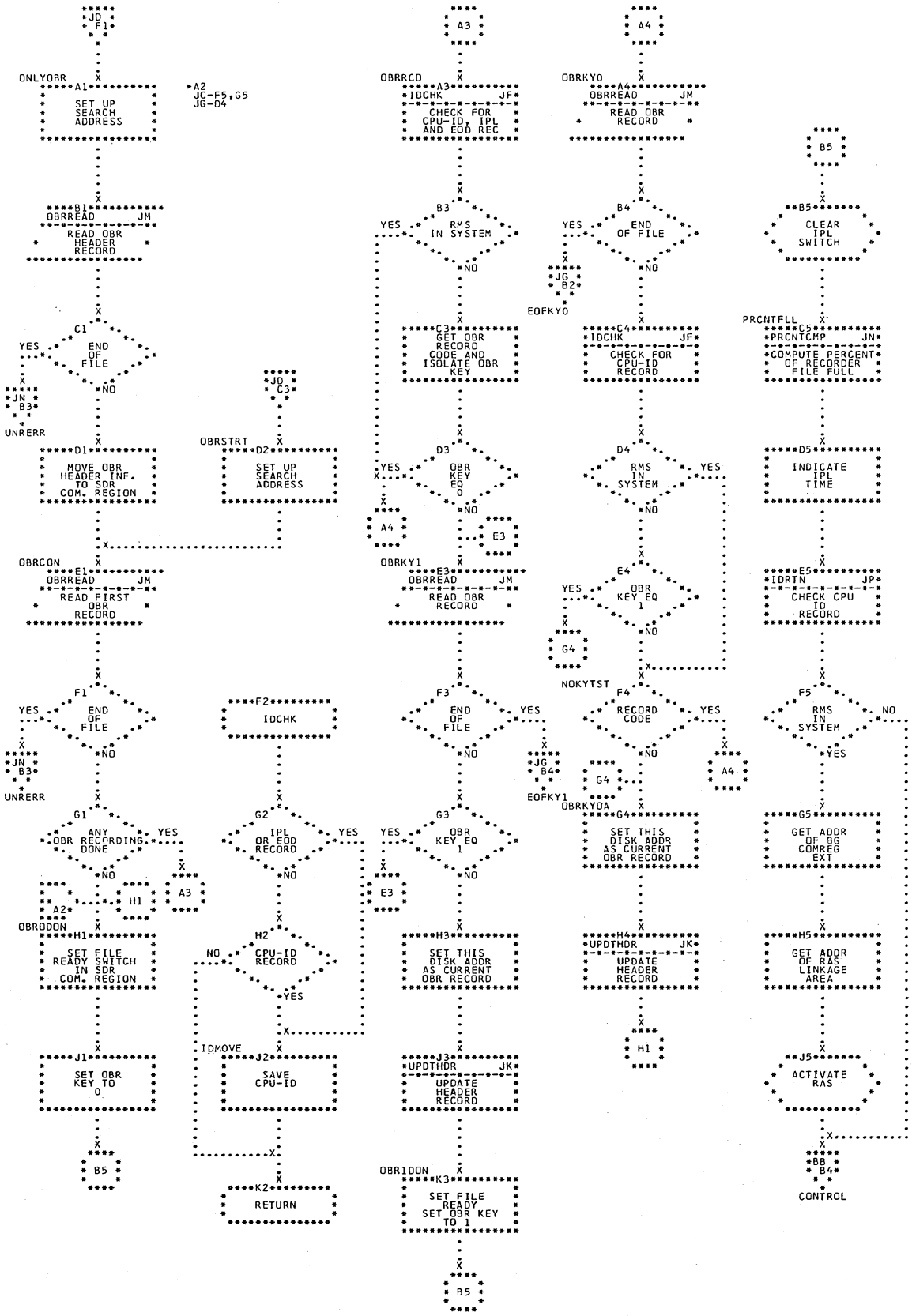


Chart JG. $JOBCTLM - Check Recorder File (Part 4 of 4)
Refer to Chart 15.

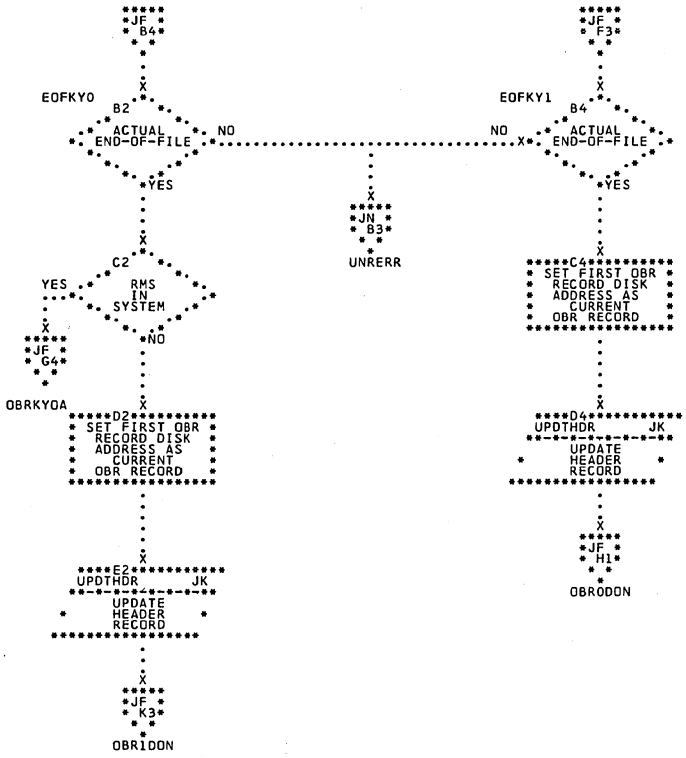


Chart JH. $JOBCTLM - ROD Statement Processor
Refer to Chart 15.

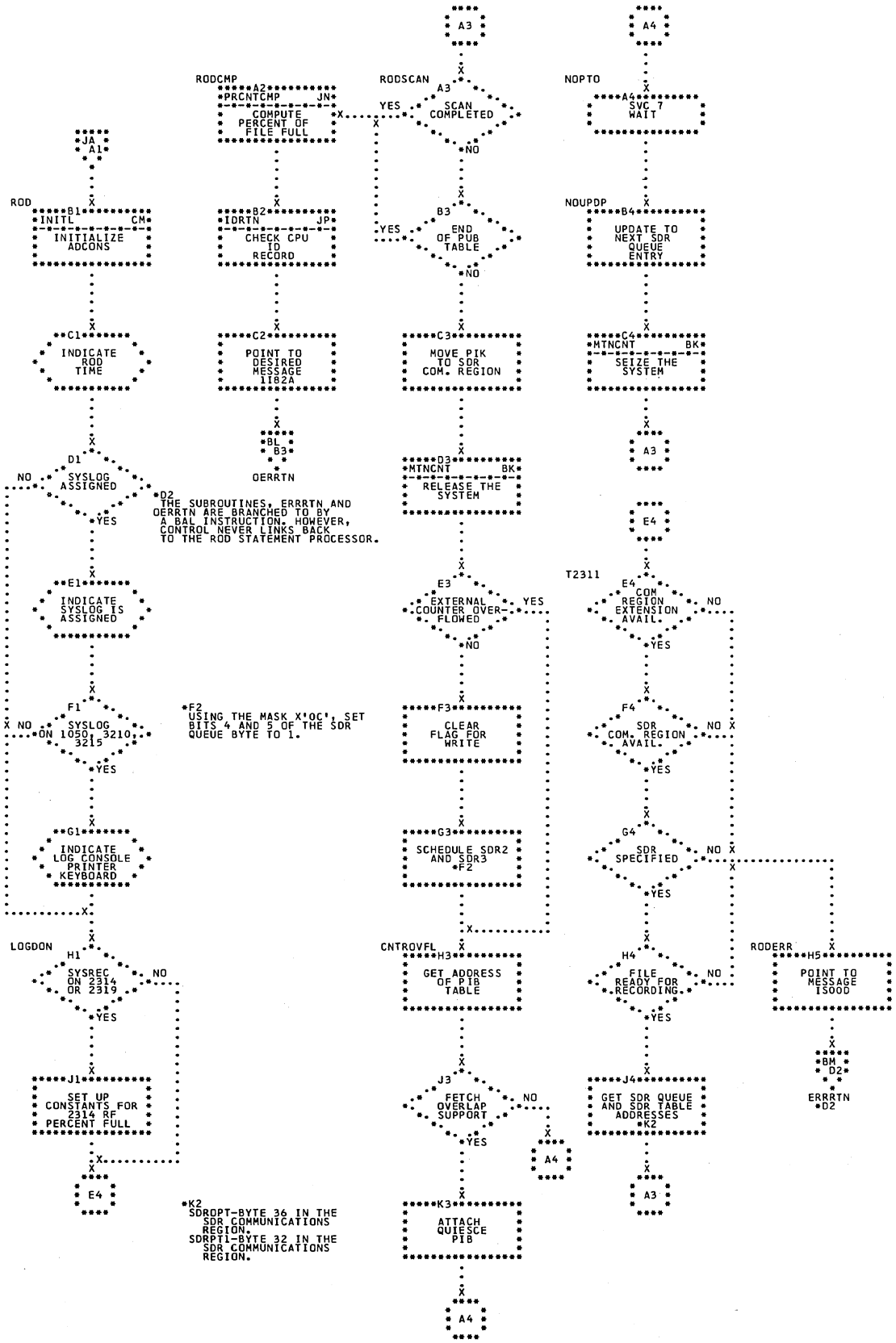


Chart JJ. $JOBCTLM - Miscellaneous Subroutines (Part 1 of 2)
Refer to Chart 15.

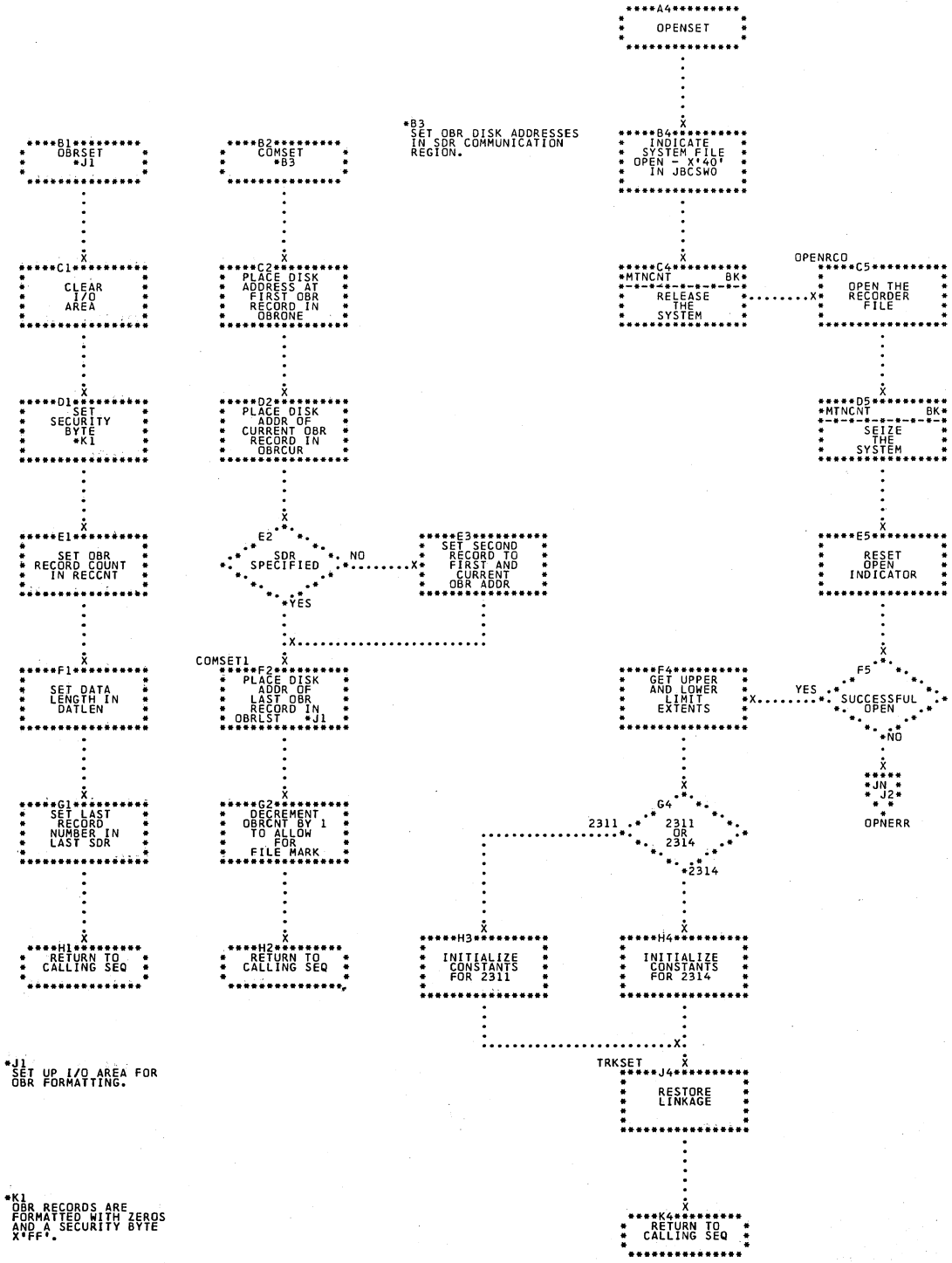


Chart JK. $JOBCTLM - Miscellaneous Subroutines (Part 2 of 2)
Refer to Chart 15.

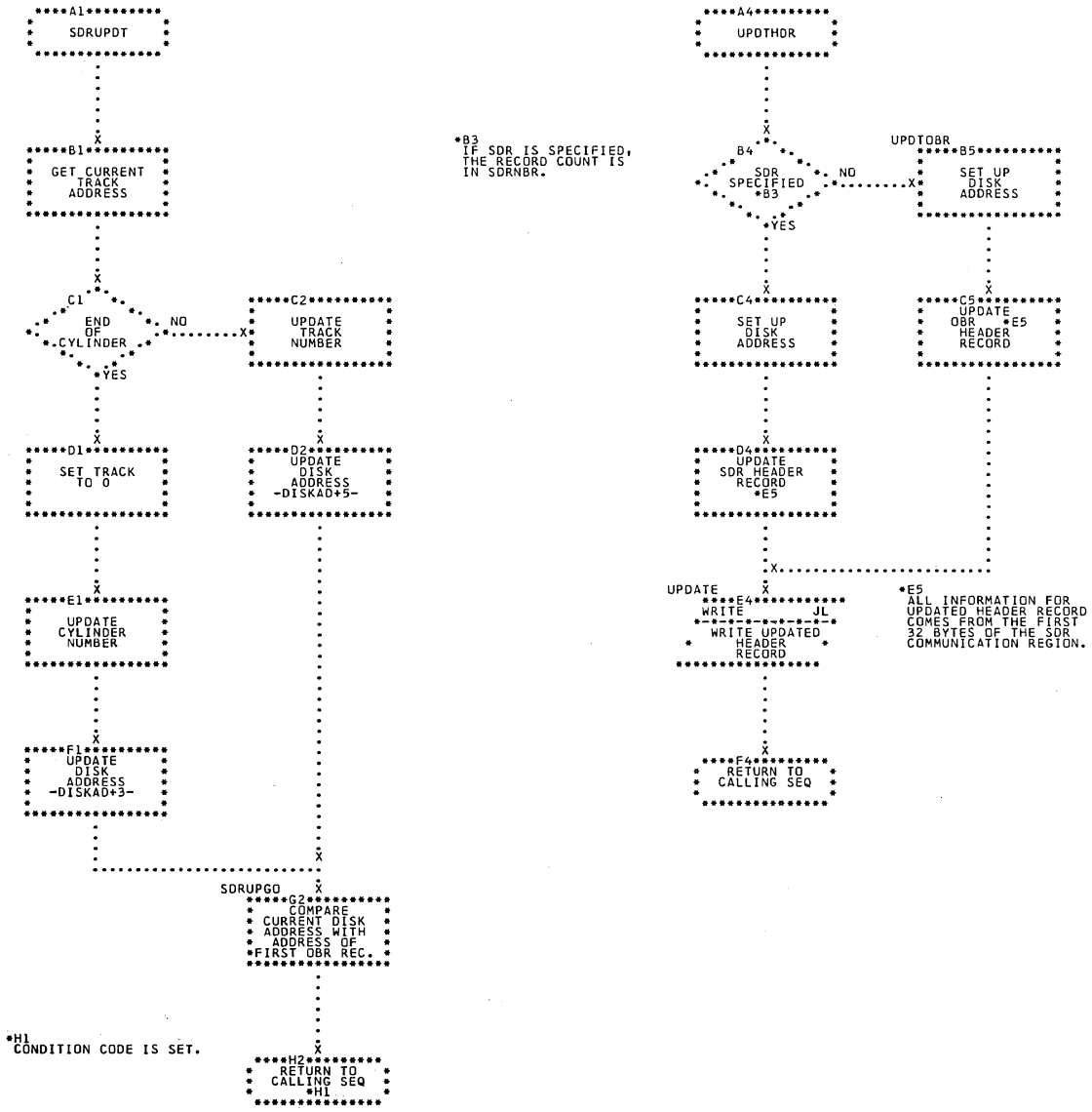


Chart JL. $JOBCTLM - I/O Subroutines (Part 1 of 2)
Refer to Chart 15.

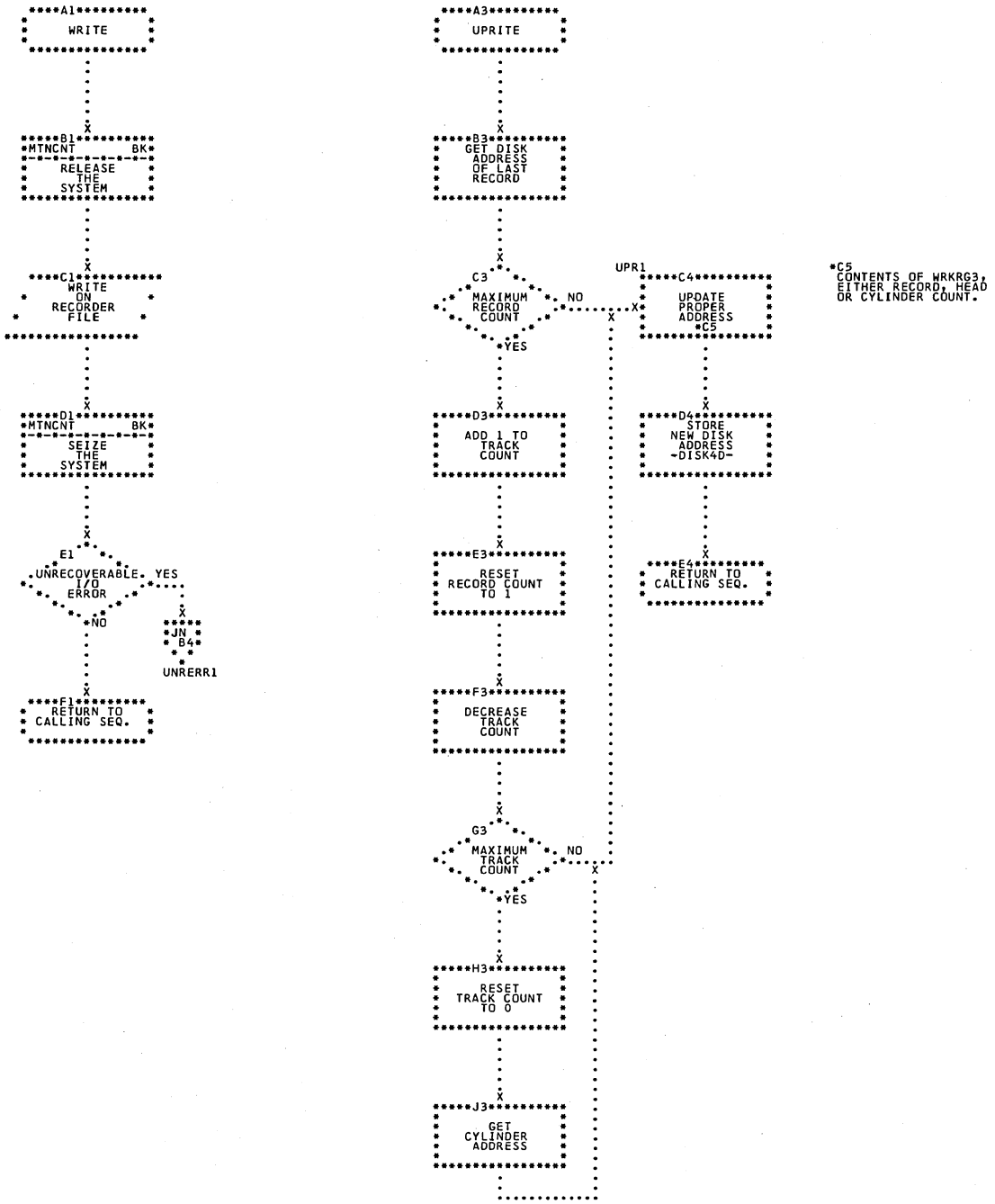


Chart JM. $JOBCTLM - I/O Subroutines (Part 2 of 2)
Refer to Chart 15.

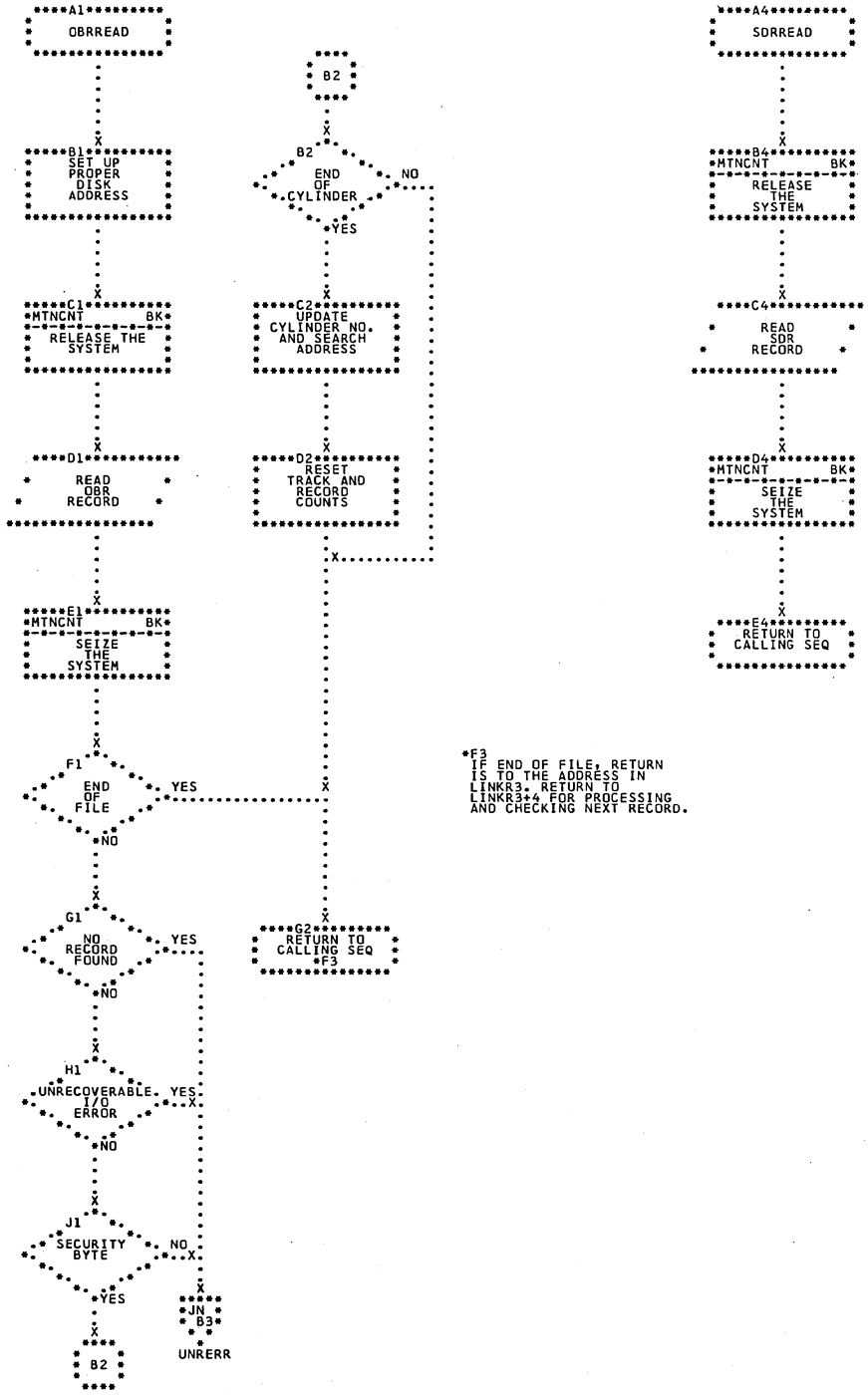


Chart JN. $JOBCTLM - RMS and Error Subroutines
Refer to Chart 15.

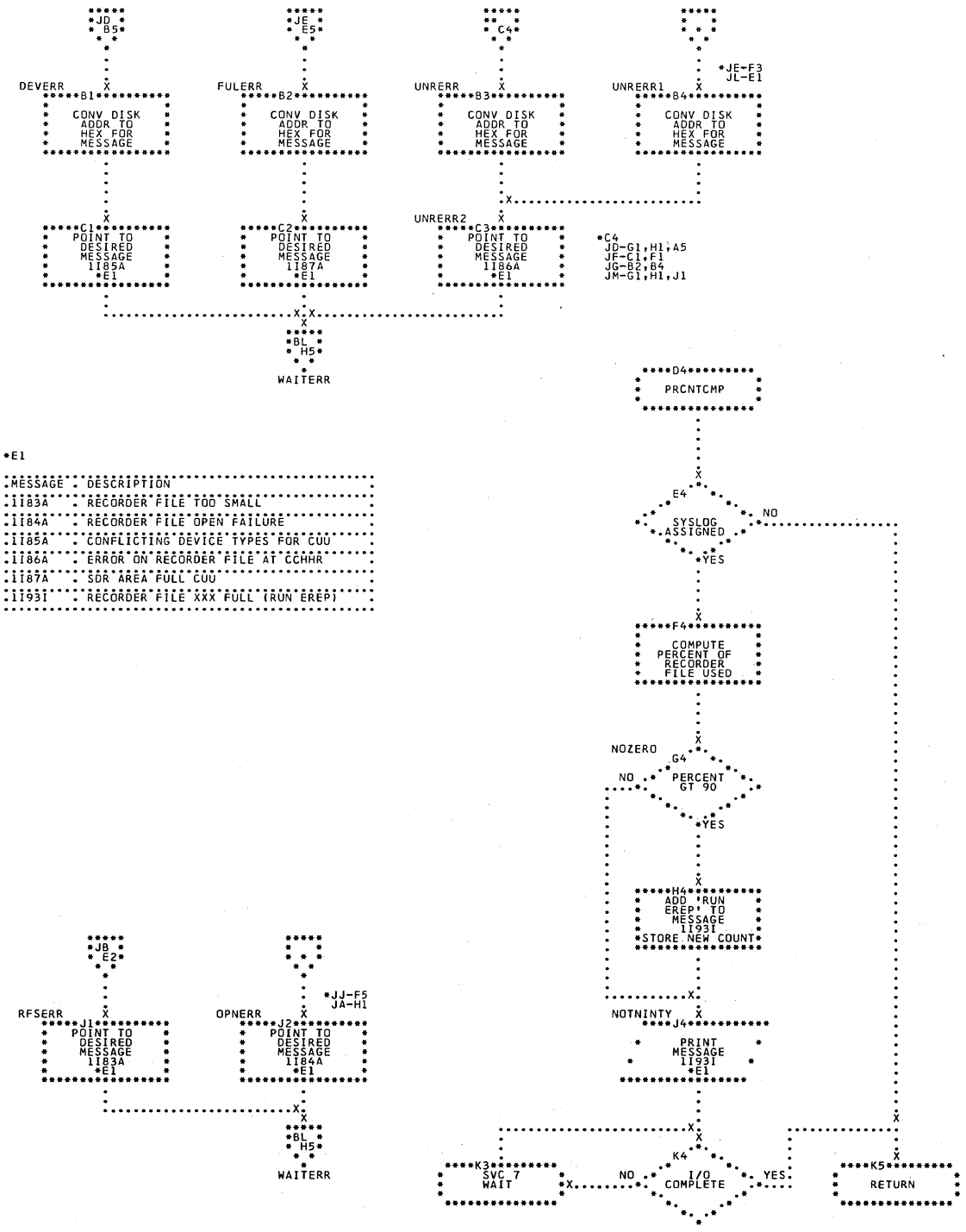


Chart JP. $JOBCTLM - Miscellaneous Subroutines (Part 1 of 2)
Refer to Chart 15.

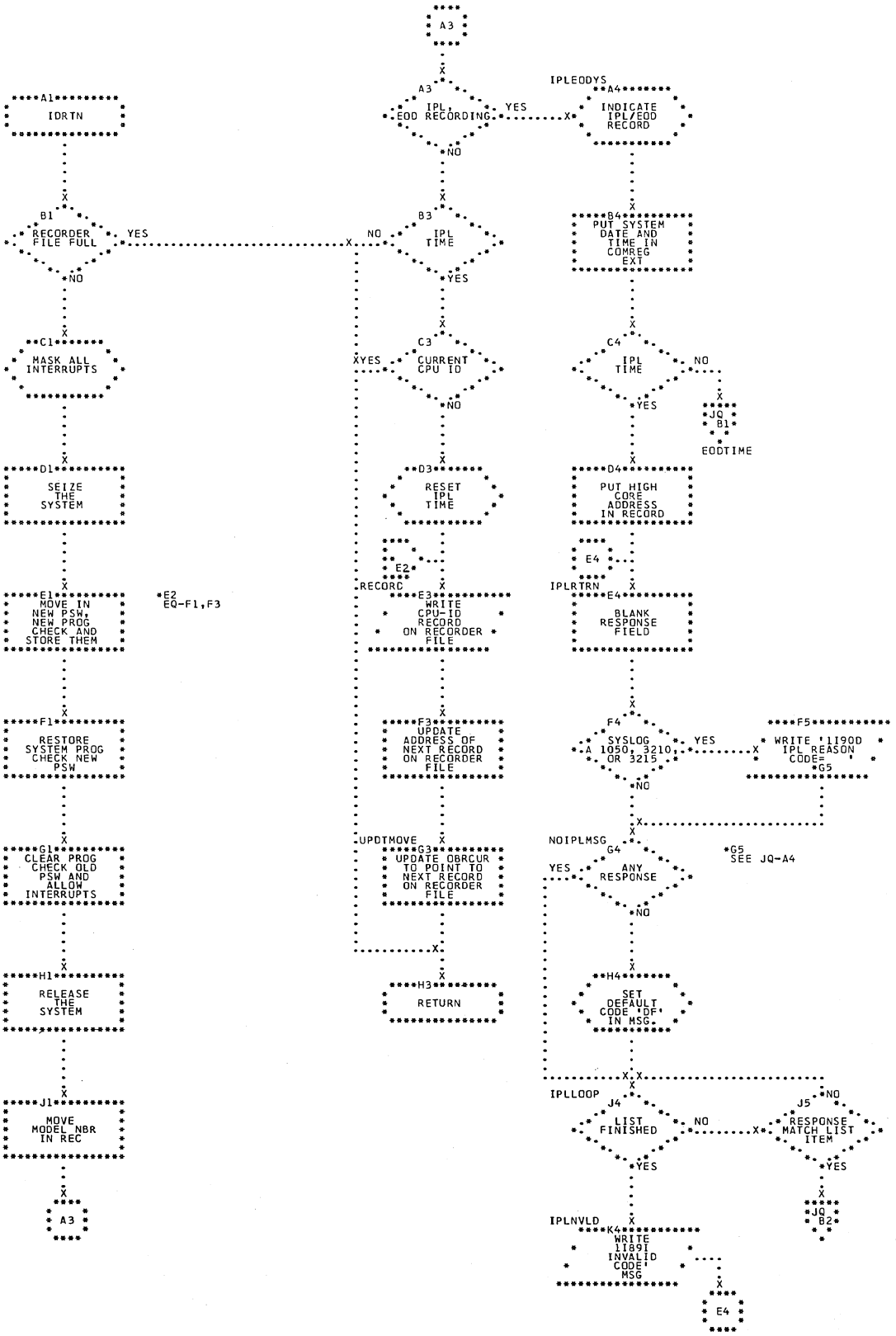
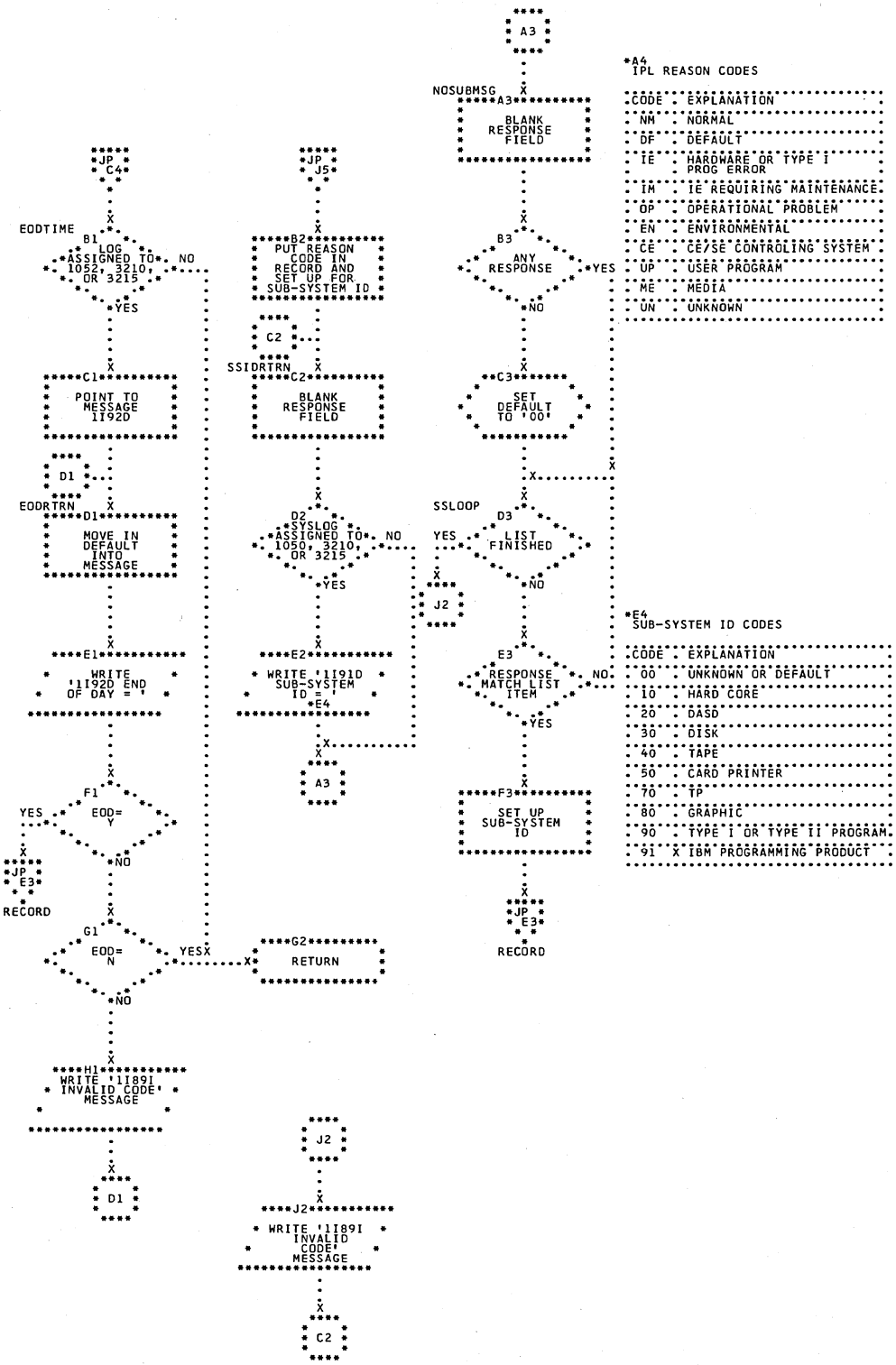


Chart JQ. $JOBCTLM - Miscellaneous Subroutines (Part 2 of 2)
Refer to Chart 15.



*A4
IPL REASON CODES

CODE	EXPLANATION
NM	NORMAL
DF	DEFAULT
IE	HARDWARE OR TYPE I PROG ERROR
IM	IE REQUIRING MAINTENANCE
OP	OPERATIONAL PROBLEM
EN	ENVIRONMENTAL
CE	CE/SE CONTROLING SYSTEM
UP	USER PROGRAM
ME	MEDIA
UN	UNKNOWN

*E4
SUB-SYSTEM ID CODES

CODE	EXPLANATION
00	UNKNOWN OR DEFAULT
10	HARD CORE
20	DASD
30	DISK
40	TAPE
50	CARD PRINTER
70	TP
80	GRAPHIC
90	TYPE I OR TYPE II PROGRAM
91 X	IBM PROGRAMMING PRODUCT

Chart KA. $JOBCTLN - Job Accounting Interface (Part 1 of 2)
Refer to Chart 16.

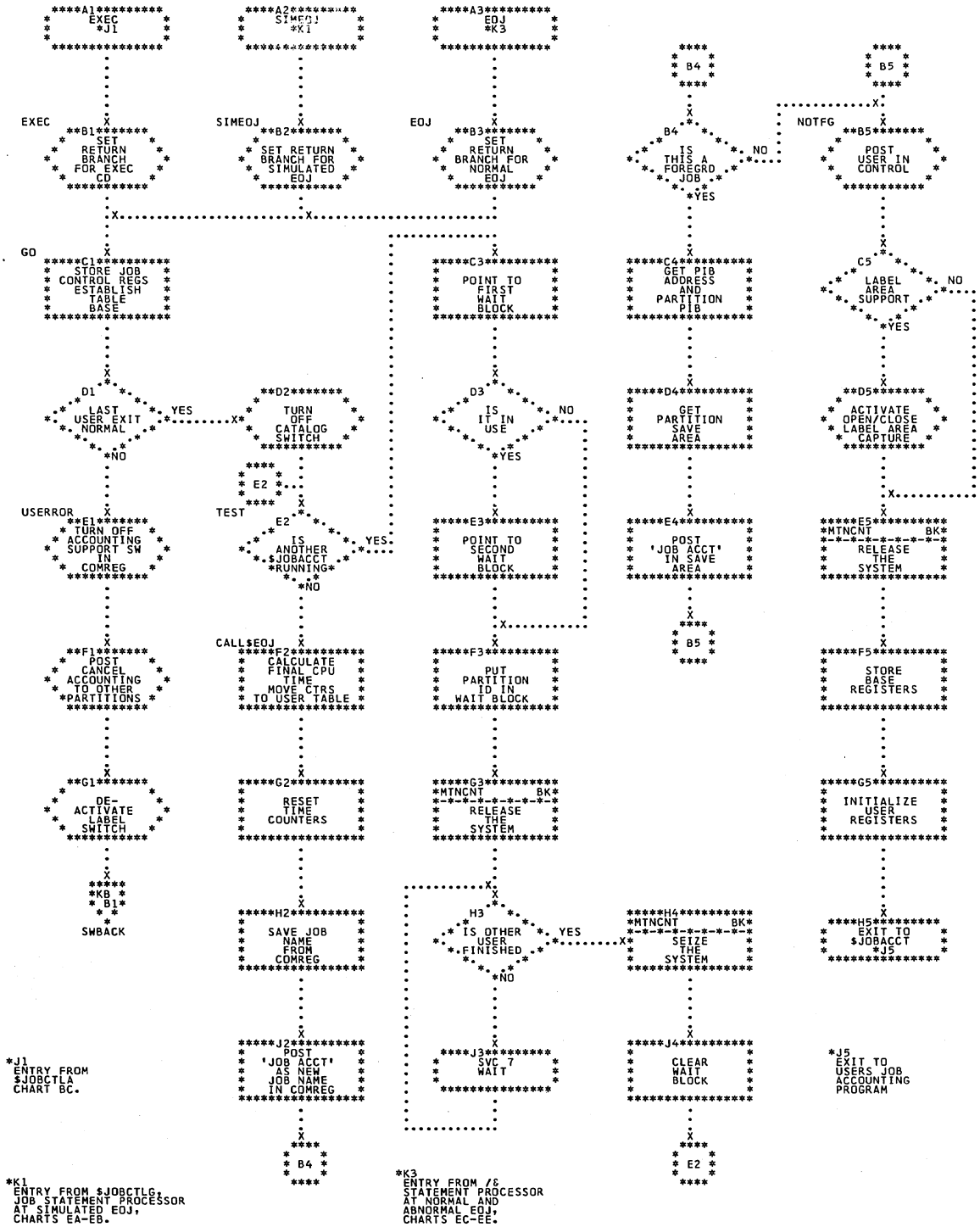


Chart KB. $JOBCTLN - Job Accounting Interface (Part 2 of 2)
Refer to Chart 16.

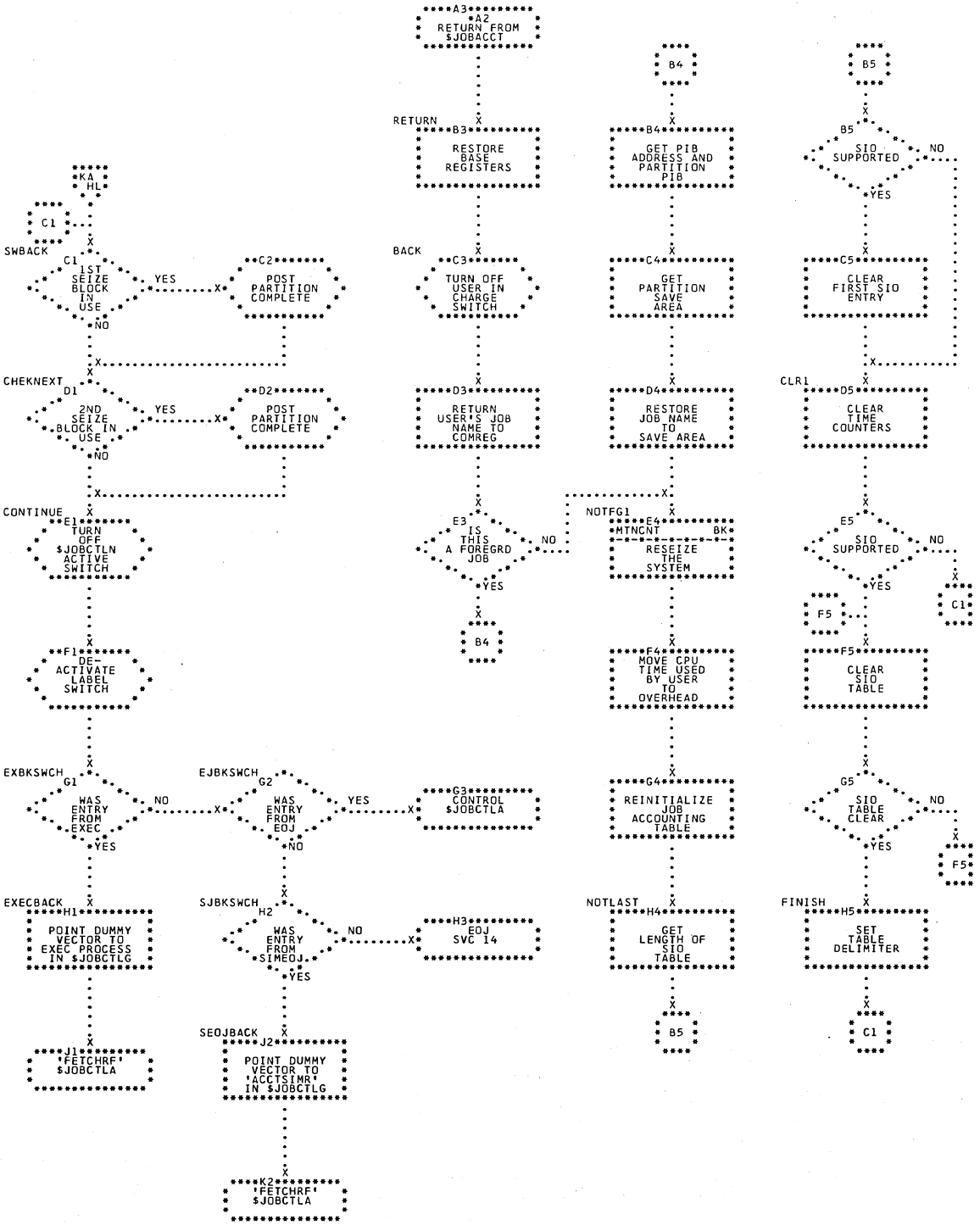


Chart LA. $$BLSTIO - Initialization
Refer to Chart 07.

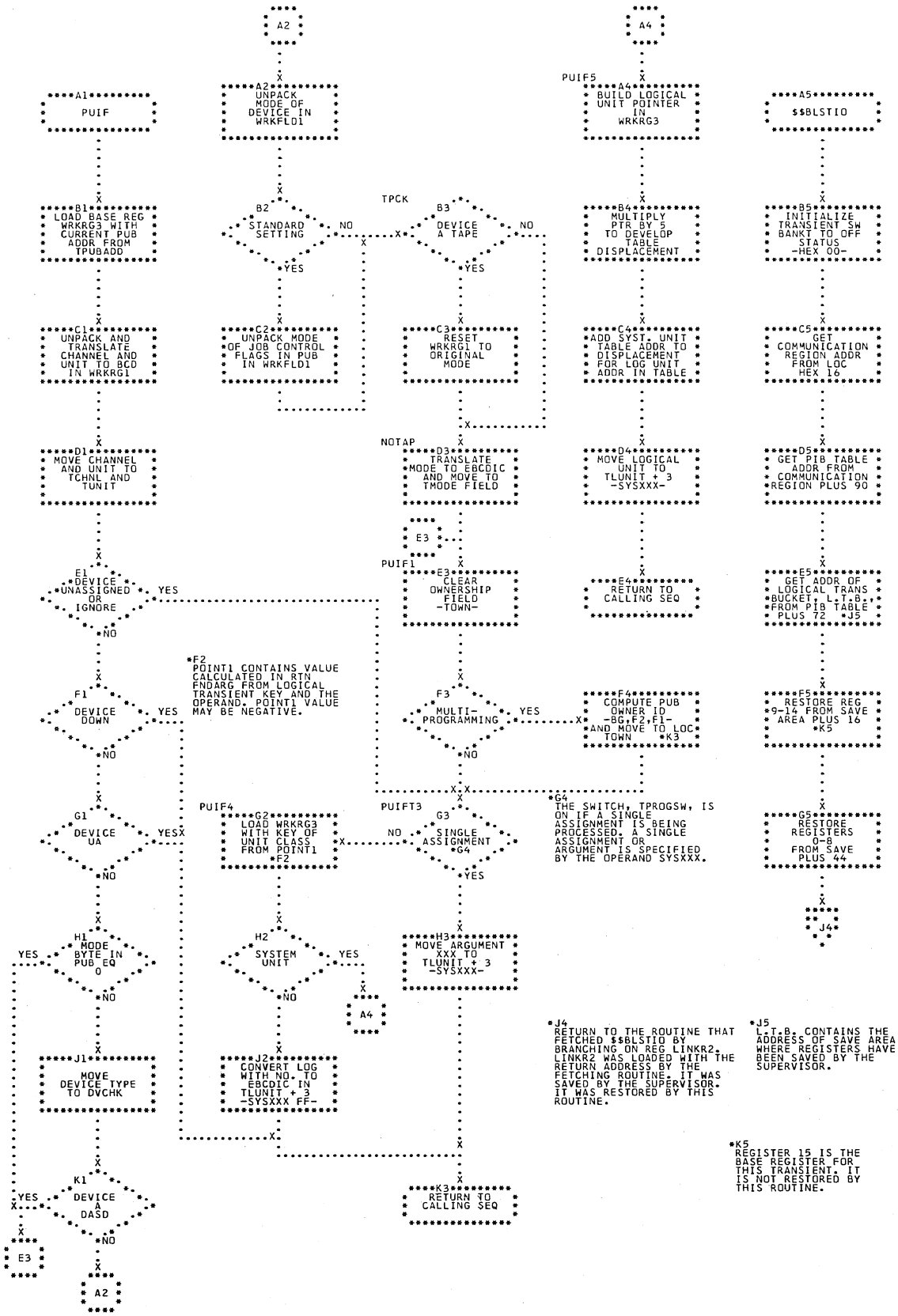


Chart LB. $$BLSTIO - Operand Identification Subroutine
Refer to Chart 07.

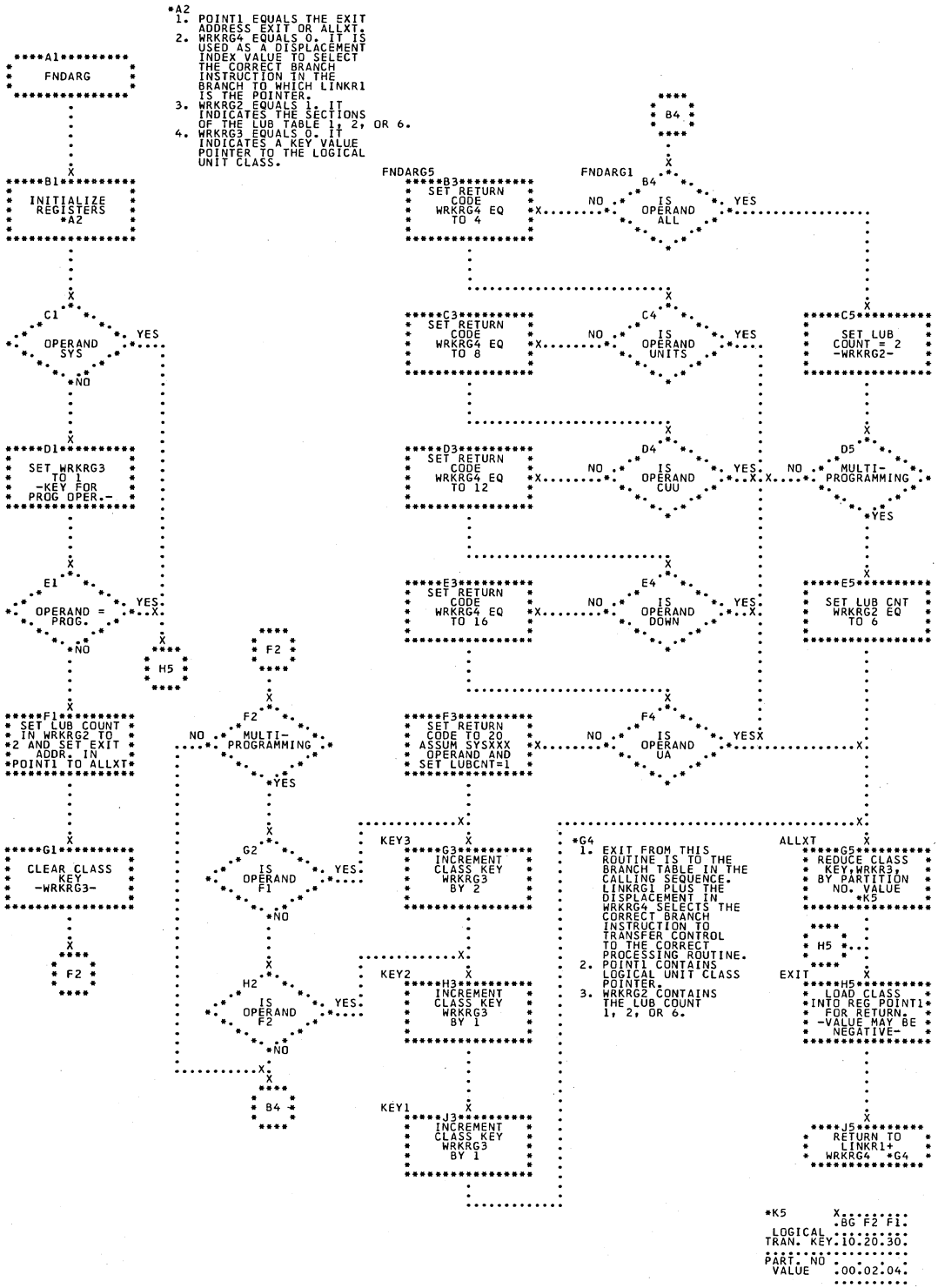
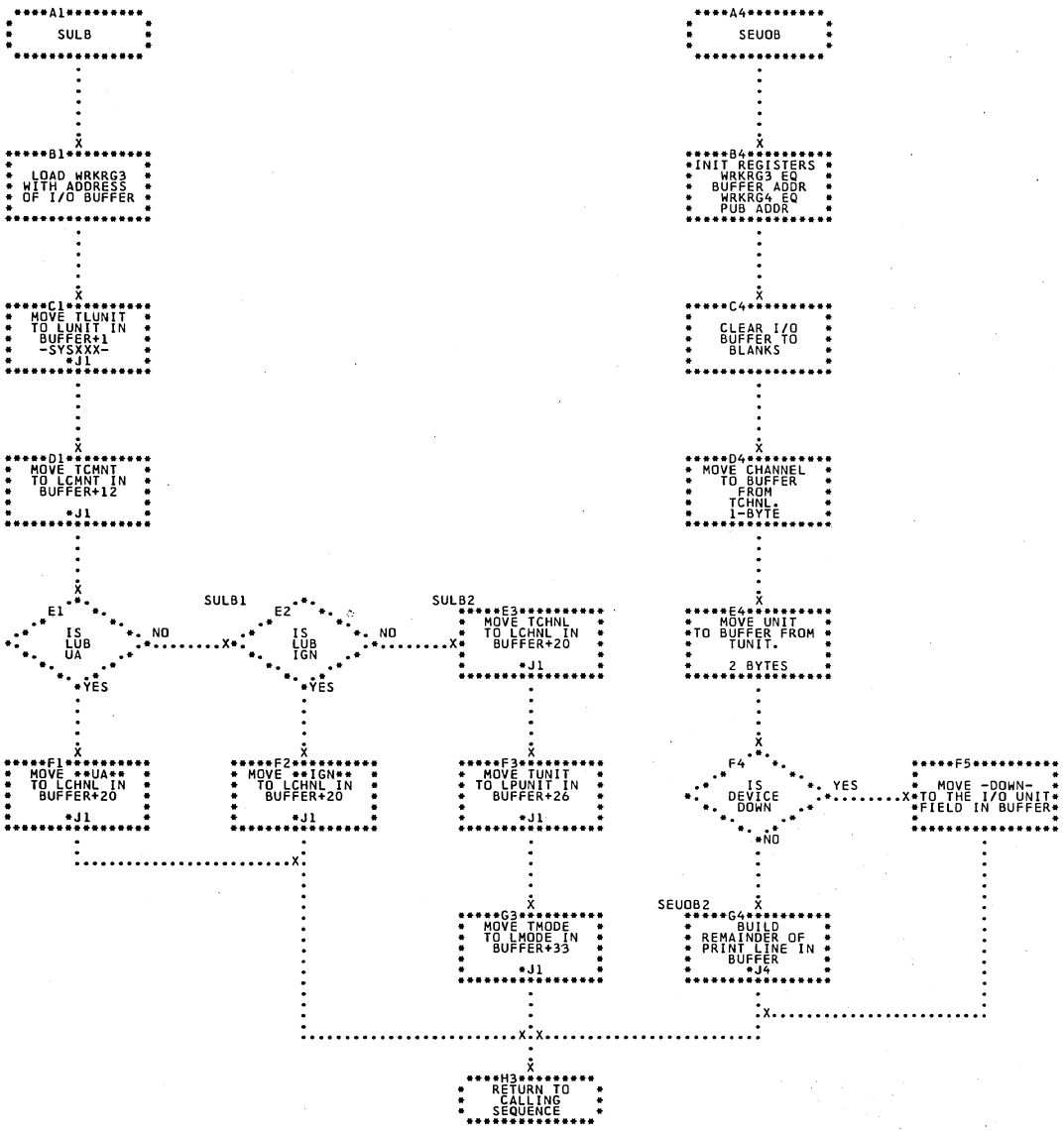


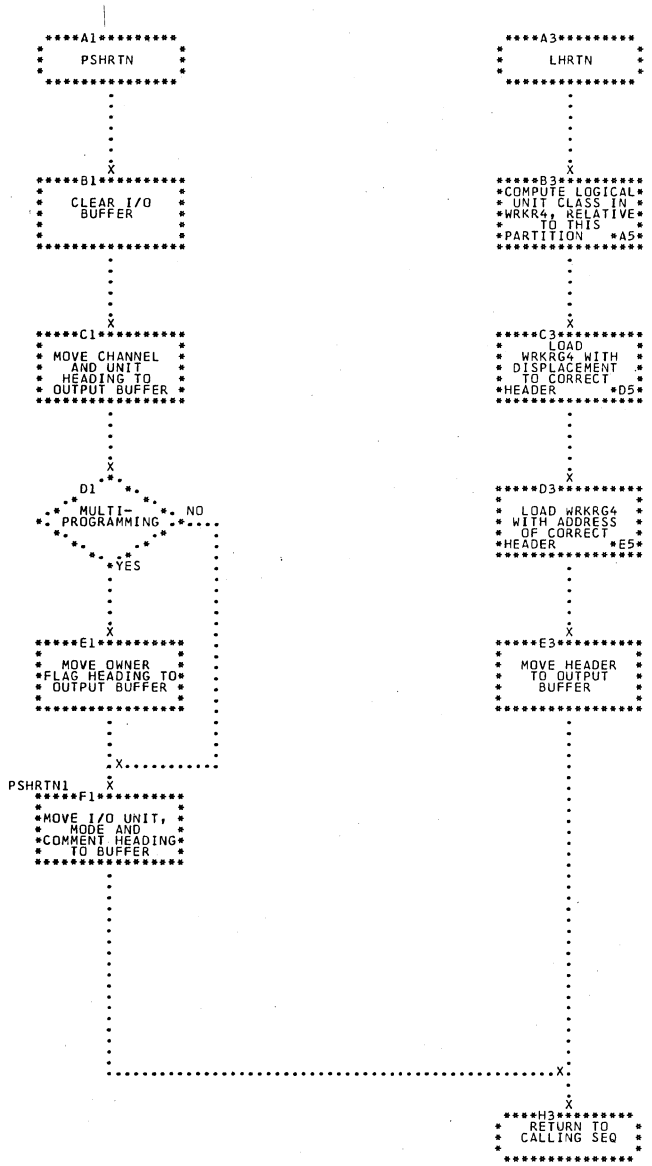
Chart LC. $$BLSTIO - Build Print Line Subroutine
Refer to Chart 07.



*J1
LUNIT -SYSXXX----LCMNT -XXX----LCHNL **-UA-** **-IGN-** LPUNIT XX LMODE XX

*J4
1. MOVE OWNER TO OWNER FIELD
2. MOVE MODE FROM TMODE TO MODE FIELD
3. MOVE LOGICAL UNIT FROM TLUNIT TO THE I/O UNIT FIELD
4. MOVE THE COMMENT FROM TCMNT TO THE COMMENT FIELD-MODE LESS 6 BYTES

Chart LD. $$BLSTIO - Build Header Subroutine
Refer to Chart 07.



*A5

	BG	F2	F1
LOGICAL TRAN. KEY		10	20 30
PARTITION VALUE NUMBER	00	02	04

PARTITION VALUE NUMBER + CLASS KEY, POINTI (FROM FNDARG ROUTINE) X2 = DISPLACEMENT INTO HEADING INDEX TABLE.

*D5
BG = HEX 74
F2 = HEX 9C
F1 = HEX AD

*E5
HEADER ADDRESS EQUALS INDEX DISPLACEMENT (WRKRG4) + CONTENTS REG 15. REG 15 = BASE REG OF $$BLST10 TRANSIENT.

INITIAL PROGRAM LOAD ($$A$IPL1, $$A$IPL2, $IPLRT2, $IPLRT3, AND $IPLRT4) CHARTS AA-AZE

Label	Phase	Chart
ABNCHK	$IPLRT2	AQ
ABNEND	$IPLRT2	AQ
ADCLOP	$$A$IPL2	AC
ADREST	$IPLRT4	AZB
ADDRTN	$IPLRT3	AR
ALCERR	$IPLRT2	AM
ALCRTN	$IPLRT2	AL
ALCRT1	$IPLRT2	AL
ALCRT2	$IPLRT2	AM
ALCRT3	$IPLRT2	AM
ALTCHANQ	$$A$IPL2	AG
	$IPLRT4	AZA
ASNEND	$IPLRT4	AZD
ASNLP1	$IPLRT4	AZD
ASNLP2	$IPLRT4	AZD
ASNRTN	$IPLRT4	AZD
ASNSTP	$IPLRT4	AZD
BCHEST	$$A$IPL2	AF
	$IPLRT3	AU
BEGIN	$IPLRT2	AJ
BEGIPL	$$A$IPL2	AB
BLDPUB	$$A$IPL2	AF
	$IPLRT3	AU
BSTOFF	$IPLRT3	AV
BSTOK	$IPLRT3	AV
CALLBCHR	$IPLRT2	AY
CCW1	$$A$IPL1	AA
CCW2	$$A$IPL1	AA
CCW3	$$A$IPL1	AA
CCW4	$$A$IPL1	AA
CCW5	$$A$IPL1	AA
CDSCH	$IPLRT2	AK
CHCLOP	$$A$IPL2	AF
	$IPLRT3	AU
CHCNT	$$A$IPL2	AF
	$IPLRT3	AU
CHEXT	$IPLRT3	AW
CHFIN	$$A$IPL2	AF
	$IPLRT3	AU
CHGADR	$$A$IPL2	AC
CHKCOM	$IPLRT4	AX
CHKLP	$IPLRT3	AW
CHSLOP	$IPLRT3	AW
CHSRT	$$A$IPL2	AF
CHULOP	$$A$IPL2	AH
	$IPLRT3	AW
CHUPD	$IPLRT3	AW
CHURTN	$$A$IPL2	AH
	$IPLRT3	AW
CILRNO	$$A$IPL2	AC
CLDLP1	$$A$IPL2	AG
	$IPLRT4	AZA
CLDLP2	$$A$IPL2	AG
	$IPLRT4	AZA
CLDRTN	$IPLRT4	AZA
CLEAR	$$A$IPL2	AB
CLEARCOR	$$A$IPL2	AB
COMMA	$IPLRT2	*
COMCHK	$IPLRT4	AZA
COMDOK	$IPLRT4	AX
COMLOP	$IPLRT4	AZA
COMNFD	$IPLRT4	AX
COMRGN	$IPLRT2	*
	Communications region DSECT origin.	
CONTIN	$$A$IPL2	AB
CPBFEND	$IPLRT4	AZA
CPYSLB	$IPLRT4	AZC
CPYSRT	$IPLRT4	AZC
CTTLUB	$IPLRT4	AY
CYLDP	$IPLRT4	AY
DATERT	$IPLRT4	AZB
DBLADD	$IPLRT3	AR
DBLSCN	$IPLRT3	AR
DDLUB	$IPLRT2	AP
DECLP	$IPLRT2	AP
DECRL	$$A$IPL2	AG
	$IPLRT2	AN
DECRR	$$A$IPL2	AG
	$IPLRT2	AN
DECRTN	$IPLRT2	AP
DELEXT	$IPLRT3	AT
DELIM	$IPLRT2	*
	Translate table for field delimiter.	
DELLOP	$IPLRT3	AT
DELRTN	$IPLRT3	AT
DSDP1	$IPLRT4	AZ
DSDP2	$IPLRT4	AZ
DSDP3	$IPLRT4	AZ
DSDP4	$IPLRT4	AZ
DSDRET	$IPLRT4	AZ
DSDRTN	$IPLRT4	AZ
DSKADR	$$A$IPL2	AH
DVCLST	$IPLRT2	*
	Device type table used to add PUB for a specified device.	
ENADR	$IPLRT3	AS
ENDRD	$$A$IPL2	AD
ENFND	$$A$IPL2	AF
	$IPLRT3	AU
ER1121	$IPLRT4	AZ
ERR	$$A$IPL2	AB
ERRHLT	$$A$IPL2	AB
EXIT1	$$A$IPL2	AH
EXTRTN	$$A$IPL2	AD
	Label of the address that the program returns to if SYSRDR is assigned and the operator indicates he has add and/or delete cards in SYSRDR. The return is via an external interrupt.	

---

*Listing Only.

FDSRTN	$IPLRT2	AP
FNDTYP	$IPLRT3	AV
FOUND	$$A$IPL2	AC
FPRTN	$IPLRT4	AZE
GETADD	$IPLRT2	AL
GETDEL	$IPLRT2	AL
GETDEL1	$IPLRT2	AL
GETNICL	$IPLRT4	AZD
GETSET	$IPLRT2	AM
GET3	$IPLRT2	AL
GET4	$IPLRT2	AL
GO	$$A$IPL2	AD
HALT	$$A$IPL2	AB
HARDMC	$$A$IPL2	AB
HEXRTN	$IPLRT3	AW
IJBIPLAD	$IPLRT3	AR
IJBIP210	$$A$IPL2	AB
ILLCD	$IPLRT2	AQ
IOFLD	$$A$IPL2	*
	$IPLRT2	*

Origin of high core I/O tables.

IOHALT	$IPLRT2	AQ
IOHLD	$IPLRT2	AQ
IOHL2	$IPLRT2	AQ
IOSTOP	$IPLRT2	AQ
IPLEND	$IPLRT4	AY
ITSADWD	$$A$IPL2	AC
KEYCHK	$IPLRT3	AS
KEYCHK1	$IPLRT3	AS
KEYCHK2	$IPLRT3	AR
LASTRD	$$A$IPL2	AC
LBLOC	$$A$IPL2	AC
LBLPED	$IPLRT2	AP
LOAD3NOW	$IPLRT2	AL
LOGRED	$IPLRT2	AK
LOGSTR	$IPLRT2	AK
LOGWRK	$$A$IPL2	AE
LOKCLD	$$A$IPL2	AB
LOOKLUB	$IPLRT4	AY
LOOKUP	$$A$IPL2	AB
LOWCRE	$IPLRT2	*

Origin of low core map DSECT.

LUBHLP	$IPLRT2	AE
	$IPLRT4	AY
LUBLOP	$$A$IPL2	AE
LUBLPL	$IPLRT2	AP
LUBMVC	$$A$IPL2	AE
LUBRTN	$$A$IPL2	AE
LUBSET	$$A$IPL2	AD
LUDRTN	$IPLRT2	AP
LUUEND	$$A$IPL2	AH
LUUPLP	$$A$IPL2	AH
LUURTN	$$A$IPL2	AH
	$IPLRT2	AP
MAXKEY	$$A$IPL2	*
	$IPLRT3	*
MCR	$IPLRT3	AV
MLUUR	$IPLRT2	AP
MODEOK	$IPLRT2	AJ
MONITOR	$IPLRT2	AJ
MPXCHK	$IPLRT4	AZE
MPXGO	$IPLRT4	AZE
MPXHLT	$IPLRT4	AZE
MPXHL1	$IPLRT4	AZE
MPXHL2	$IPLRT4	AZE
MPXLOP	$IPLRT4	AZE
MPXMOV	$IPLRT4	AZE
MPXRTN	$IPLRT4	AZE
MSGRTN	$IPLRT2	AQ
MVCEND	$$A$IPL2	AC
NEXTSYS	$IPLRT4	AZ
NLPYR	$IPLRT4	AZB
NODASD	$IPLRT2	AJ
NOMTEB	$IPLRT4	AY
NOMVP	$$A$IPL2	AF
NONBLK	$IPLRT2	*

Translate table for nonblank scan.

NOSOAD	$IPLRT3	AU
NOSODL	$IPLRT3	AT
NOSOROOM	$IPLRT3	AU
NOTBG	$IPLRT4	AZD
NOTEB	$IPLRT2	AJ
NOTFOR4	$IPLRT2	AJ
NOTMPS	$$A$IPL2	AE
NOT4AND3	$IPLRT2	AM
NOT4NOW	$IPLRT2	AL
NO3211	$IPLRT4	AZ
NUMCVT	$IPLRT3	AS
OFFINT	$IPLRT4	AY
ONINT	$IPLRT4	AY
OPNEND	$IPLRT4	AZ
OPNLOP	$IPLRT4	AZ
OPNRTN	$IPLRT4	AZ
OPNUSN	$IPLRT4	AZ
OPRTN	$IPLRT2	AK
ORGEST	$$A$IPL2	AE
PBFEND	$IPLRT4	AZA
PBFFIN	$IPLRT4	AX
PBFLOP	$IPLRT4	AZA
PBFRTN	$IPLRT4	AZA
PIBTAB	$IPLRT2	*

Origin of PIB table DSECT.

POINTLUB	$IPLRT4	AY
PPBEGOK	$$A$IPL2	AD
PPBEGOR	$IPLRT2	AJ
PSWGO	$$A$IPL2	AD
PSWSET	$$A$IPL2	AD
PUBCLC	$IPLRT2	AJ
PUBDEQ	$IPLRT3	AT
PUBEXD	$IPLRT3	AR
PUBMKE	$$A$IPL2	AE
	$IPLRT3	AS
PUBMK1	$IPLRT3	AS
RACDVC	$IPLRT3	AV
RASPCRET	$IPLRT2	AJ
RASUPR	$IPLRT2	AJ
RDBLOK	$$A$IPL2	AC
RDDIR2	$$A$IPL2	AB
RDRIPT	$$A$IPL2	AD
RDRTST	$IPLRT4	AX
READER	$$A$IPL2	AB
READGO	$IPLRT2	AJ
READRT	$IPLRT2	AK
RECNO	$$A$IPL2	AH
REREAD	$IPLRT2	AK
RESIDL	$$A$IPL2	AG
	$IPLRT2	AN
RESIDR	$$A$IPL2	AG

---

*Listing Only.

	$IPLRT2	AN
RESNFD	$IPLRT4	AX
RESWRK	$$A$IPL2	AE
RIGHT	$$A$IPL2	AG
	$IPLRT2	AN
RMSMCRET	$$A$IPL2	AB
RMSPCRET	$$A$IPL2	AB
RSTCHQ	$IPLRT4	AZE
RSTLUB	$IPLRT2	AP
RTRAK	$$A$IPL2	AC
SCHCNT	$$A$IPL2	AF
	$IPLRT3	AU
SCHLOP	$IPLRT3	AV
SCHSCH	$$A$IPL2	AF
	$IPLRT3	AU
SCHSTA	$$A$IPL2	AF
	$IPLRT3	AU
SCHTST	$$A$IPL2	AF
	$IPLRT3	AU
SCNEND	$IPLRT3	AT
SCNLOP	$IPLRT3	AT
SETBLK	$$A$IPL2	AC
SETFIRST	$IPLRT2	AM
SETFCL	$IPLRT3	AW
SETKEY	$$A$IPL2	AD
SETLOG	$$A$IPL2	AD
SETPROT	$$A$IPL2	AD
SETRTN	$IPLRT4	AX
SET2	$IPLRT4	AZD
SKPINC	$IPLRT4	AZB
SKPKEY	$IPLRT3	AR
SKPMVT	$$A$IPL2	AE
SPRSW	$$A$IPL2	AF
	$IPLRT3	AU
SRCH32H	$IPLRT4	AZ
SSKLOP	$IPLRT2	AM
STORECPU	$IPLRT2	AJ
SW	$$A$IPL2	AB

This switch is initially off (NOP). It is used after the transient directory has been searched and the supervisor nucleus has been found to branch around the instructions used in searching for the supervisor nucleus.

SW1	$$A$IPL2	AC

NOP/BR switch that is turned on (BR) for reading the last block of the supervisor into main storage.

SW2	$$A$IPL2	AC

NOP/BR switch that is turned on (BR) when first block of the supervisor is read.

SW3	$$A$IPL2	AB

This switch is initially off (NOP). It is used after the transient directory address and SYSRES device type has been determined to branch around these instructions.

SW4	$$A$IPL2	AB
SYSMVC	$$A$IPL2	AG
	$IPLRT2	AN
TAPE	$IPLRT3	AV
TEBCLC	$IPLRT2	AJ
TEBDEQ	$IPLRT3	AT
TEBDET	$IPLRT3	AV
TEBEST	$IPLRT3	AV
TEBEXD	$IPLRT3	AV
TEBLOP	$IPLRT3	AT
TEBVCHK1	$IPLRT3	AT
TEBVCNT	$IPLRT2	AJ
TEBVDEQ	$IPLRT3	AT
TEBVDET	$IPLRT3	AV
TEBVDT	$IPLRT3	AV
TEBVEST	$IPLRT3	AV
TIMERT	$IPLRT4	AZB
TMCHK	$IPLRT4	AX
TPDVC	$IPLRT3	AV
TRKNO	$$A$IPL2	AH
TRTBRC	$IPLRT2	AQ
TRTTAB	$IPLRT2	*

Control card translate table.

TSTSO	$IPLRT3	AU
UPDAT	$$A$IPL2	AC
UPLUB	$IPLRT2	AP
USEWRK	$$A$IPL2	AE
USNRES	$IPLRT4	AX
WAIT	$$A$IPL2	AD
YESLOG1	$IPLRT4	AY
YLPYR	$IPLRT4	AZB

JOB CONTROL ($JOBCTLA, $JOBCTLD, $JOBCTLF, $JOBCTLG, $JOBCTLJ, $JOBCTLK, $JOBCTLM, $JOBCTLN, AND $$BLSTIO) CHARTS BA-LD

Label	Phase	Chart
ABGF1F2	$JOBCTLD	CG
ACCTBR1	$JOBCTLA	BB
	$JOBCTLG	EB
ACCTBR2	$JOBCTLG	EB
ACCTBR3	$JOBCTLG	EB
ACCTBR4	$JOBCTLG	EB
ACCTBR5	$JOBCTLG	EA
ACCTBR6	$JOBCTLJ	GM
ACCTCL	$JOBCTLA	BB
	$JOBCTLJ	GM
ACCTIGN	$JOBCTLF	DL
	$JOBCTLJ	GH
ACCTIGN1	$JOBCTLA	BC
	$JOBCTLG	EF
ACCTIGN2	$JOBCTLA	BC
	$JOBCTLG	ED
ACCTIGN3	$JOBCTLA	BA
	$JOBCTLG	EB
ACCTIGN4	$JOBCTLA	BB
	$JOBCTLG	EB
ACCTIGN5	$JOBCTLG	EB
ACCTIGN6	$JOBCTLG	EB
ACCTIGN7	$JOBCTLJ	GM
ACCTIGN8	$JOBCTLG	EC
ACCTLV1	$JOBCTLA	BB
ACCTLV1B	$JOBCTLJ	GM
ACCTNOTP	$JOBCTLA	BC
ACCTSIMR	$JOBCTLG	EA

---

*Listing Only.

ACCTSWO	$JOBCTLA	BC
ACCTUPDT	$JOBCTLF	DL
	$JOBCTLJ	GM
ACTION	$JOBCTLJ	GD
ACTRSP	$JOBCTLA	BB
ADDLNGTH	$JOBCTLG	EH
ADDSDR	$JOBCTLM	JE
ADR	$JOBCTLA	BN
	$JOBCTLD	CM
	$JOBCTLF	DP
	$JOBCTLG	EK
ALLXT	$$BLSTIO	KB
ALLOC	$JOBCTLJ	GP
ALTASW	$JOBCTLD	*

Bank 1, bit 1: if on, specifies ALT assignment.

- Set ON: ASSGN statement processor.
- Set OFF: INITL subroutine.
- Controls processing in the ASSGN processor.

ASSGN	$JOBCTLD	CA
ASSGNB3	$JOBCTLD	CC
ASSGNB4	$JOBCTLD	CD
ASSGNB5	$JOBCTLD	CE
ASSGNB6	$JOBCTLD	CF
ASSGNB6A	$JOBCTLD	CE
ASSGNB6B	$JOBCTLD	CE
ASSGNB6C	$JOBCTLD	CE
ASSGNB6D	$JOBCTLD	CE
ASSGNB6E	$JOBCTLD	CE
ASSGNB7	$JOBCTLD	CJ
ASSGNB8	$JOBCTLD	CH
ASSGNIT	$JOBCTLD	CG
ASSGNNT	$JOBCTLD	CA
ASSGNP	$JOBCTLD	CP

Subroutine: Checks assignment of SYSRLB and SYSSLB.

ASSGNP1	$JOBCTLD	CP
ASSGNR	$JOBCTLD	CA
ASSGNTS	$JOBCTLD	CA
ASSGNV1	$JOBCTLD	CA
ASSGN0	$JOBCTLD	CA
ASSGN1	$JOBCTLD	CA
ASSGN1A	$JOBCTLD	CA
ASSGN10	$JOBCTLD	CB
ASSGN101	$JOBCTLD	CB
ASSGN11	$JOBCTLD	CB
ASSGN12	$JOBCTLD	CB
ASSGN13	$JOBCTLD	CC
ASSGN14	$JOBCTLD	CD
ASSGN15	$JOBCTLD	CD
ASSGN16	$JOBCTLD	CD
ASSGN17	$JOBCTLD	CD
ASSGN18	$JOBCTLD	CD
ASSGN19	$JOBCTLD	CD
ASSGN19A	$JOBCTLD	CD
ASSGN2	$JOBCTLD	CA
ASSGN20	$JOBCTLD	CD
ASSGN21	$JOBCTLD	CE
ASSGN22	$JOBCTLD	CG
ASSGN23	$JOBCTLD	CF
ASSGN23A	$JOBCTLD	CF
ASSGN23B	$JOBCTLD	CF
ASSGN23C	$JOBCTLD	CF
ASSGN23D	$JOBCTLD	CF
ASSGN24	$JOBCTLD	CF
ASSGN25	$JOBCTLD	CF
ASSGN26	$JOBCTLD	CG
ASSGN27	$JOBCTLD	CG
ASSGN28	$JOBCTLD	CG
ASSGN29	$JOBCTLD	CJ
ASSGN3	$JOBCTLD	CA
ASSGN30	$JOBCTLD	CJ
ASSGN31	$JOBCTLD	CJ
ASSGN32	$JOBCTLD	CJ
ASSGN33	$JOBCTLD	CJ
ASSGN34	$JOBCTLD	CH
ASSGN35	$JOBCTLD	CH
ASSGN36	$JOBCTLD	CH
ASSGN37	$JOBCTLD	CH
ASSGN38	$JOBCTLD	CJ
ASSGN4	$JOBCTLD	CA
ASSGN40	$JOBCTLD	CK
ASSGN402	$JOBCTLD	CD
ASSGN403	$JOBCTLD	CD
ASSGN404	$JOBCTLD	CD
ASSGN41	$JOBCTLD	CK
ASSGN42	$JOBCTLD	CK
ASSGN43	$JOBCTLD	CK
ASSGN5	$JOBCTLD	CA
ASSGN5A	$JOBCTLD	CA
ASSGN6	$JOBCTLD	CB
ASSGN7	$JOBCTLD	CB
ASSGN8	$JOBCTLD	CB
ASSGN9	$JOBCTLD	CB
ASSGN901	$JOBCTLD	CH
ATNCUU	$JOBCTLA	BL
ATNCUU1	$JOBCTLA	BL
BACK	$JOBCTLN	KB
BANK1	$JOBCTLD	*

2-byte switch bank. Reset to zeros in the INITL subroutine.

Byte 0, bit 0 see CLOSESW
1 see ALTASW
2 see STDFDSW
3 see RETSW
4 see EOLSW
5 see PROGSW
6 see LIOSW
7 see RESETSW

Byte 1, bit 0 see DDSW
1 see TRANSW
2 not used
3 not used
4 not used
5 see UNSW
6 see TMPSW
7 see MODSW

BEGADDR	$JOBCTLG	EG
BINCON	$JOBCTLJ	GT

Subroutine: Converts DATE and CLOCK parameters of the SET statement to

---

*Listing Only

binary in registers POINT1, POINT2, and POINT3.

BLKCNT	$JOBCTLJ	GT
BLNKLD	$JOBCTLF	DQ
BRSS	$JOBCTLJ	GQ
BTLOOP	$JOBCTLA	BC
	$JOBCTLG	EL
BTOFRT	$JOBCTLJ	GK
BTONRT	$JOBCTLJ	GK
BUFFER	$JOBCTLA	*
	through	
	$JOBCTLM	

120-byte I/O area that allows control statements to be read into the main storage area previously occupied by the job control initialization routine, JOBCTL.

BYPASS	$JOBCTLJ	GD
CALL$EOJ	$JOBCTLN	KA
CALLTTR	$JOBCTLM	JA
CANCEL	$JOBCTLG	EE
CATAL	$JOBCTLG	EN
	$JOBCTLJ	GN
CHAIN	$JOBCTLA	BE
CHECK	$JOBCTLM	JD
CHECKNEXT	$JOBCTLN	KB
CHECKRF	$JOBCTLG	EB
CHGSTT	$JOBCTLJ	GT

Subroutine: Changes system status from problem program state to supervisor state and from supervisor state to problem program state. Used by the ALLOC statement processor.

CHKASG	$JOBCTLA	BK

Root phase subroutine to check the assignment of a logical unit. If the specified unit is unassigned, the condition code 1 is set and control is returned to the calling sequence. If the unit is assigned, the PUB pointer is supplied to the calling sequence in WRKRG3.

CHKASG3	$JOBCTLA	BK

Equated to CHKASG+4. This is the entry into the CHKASG subroutine that is used when the LUB table address of the unit to be checked has already been loaded in WRKRG3.

CHKCND	$JOBCTLG	EG
CHKCNL	$JOBCTLA	BM
CHKCNT	$JOBCTLA	BE

Subroutine: Checks the area available for output records in the disk area allocated for SYSPCH or SYSLST. This subroutine is a part of $JOBCTLA and is overlaid when any other phase of Job Control is loaded.

CHKDIB	$JOBCTLA	CE
CHKJIB	$JOBCTLA	BD
CHKLNK	$JOBCTLG	FB

Subroutine: Checks SYSLNK assignment and device type.

CHKLST	$JOBCTLA	BD
CHKNXC	$JOBCTLD	CM
	$JOBCTLK	HR
CHKOPN	$JOBCTLD	CV

Subroutine: Sets the open indicator off in the DFB if the CUU of the DFB does not equal the CUU of the assignment.

CHKOPN1	$JOBCTLD	CV
CHKOPN2	$JOBCTLD	CV
CHKOVR	$JOBCTLA	BJ
CHKPGU	$JOBCTLJ	GS
CHKPRN	$JOBCTLG	GP
CHKPUB	$JOBCTLF	DN
CHKPUN	$JOBCTLK	HH
CHKRNG	$JOBCTLD	CM
	$JOBCTLG	HR

Subroutine: Checks the range of each character in a parameter.

Upon entry:

- Register POINT1 contains the address of the first byte of the parameter.
- Register POINT3 contains the number of characters to be checked minus 1.

The compare immediate instruction, RNGTOP, is modified by the calling sequence to compare for the maximum character (numeric = 9, hex = F).

CHKRNG1	$JOBCTLJ	GQ
	$JOBCTLK	HR
CHKTIM	$JOBCTLJ	GH
CHKTIM1	$JOBCTLJ	GH
CIDEND	$JOBCTLG	EH
CKCLAS	$JOBCTLF	DN
CKNDAR	$JOBCTLJ	GR
CKSCST	$JOBCTLJ	GP
CLBLUB	$JOBCTLA	BK
	$JOBCTLF	DN
CLOSE	$JOBCTLD	CL
CLOSED	$JOBCTLD	CL
CLOSESW	$JOBCTLD	*

BANK1, bit 0: If on specifies that a close is in process.

- Set ON: CLOSE statement processor.
- Set OFF: INITL subroutine and CLOSE1 subroutine.
- Controls processing in:

  1. CLOSE1 subroutine.
  2. ASSGN statement processor.

---

*Listing Only

CLOSE1	$JOBCTLD	CP
CLOSE10	$JOBCTLD	CQ
CLOSE11	$JOBCTLD	CQ
CLOSE12	$JOBCTLD	CQ
CLOSE2	$JOBCTLD	CL
CLOSE3	$JOBCTLD	CL
CLOSE7	$JOBCTLD	CL
CLOSE8	$JOBCTLD	CQ
CLOSE9	$JOBCTLD	CQ
CLRDN	$JOBCTLF	DN
CLRDOA	$JOBCTLG	EH

Portion of the EXEC statement processor used as a subroutine to clear the output area to blanks. Register POINT1 is reset to the beginning address of the area.

CLRRTN	$JOBCTLG	EK
CMNWLM	$JOBCTLJ	GP
CMPSDR	$JOBCTLM	JC
CMPSDR1	$JOBCTLM	JC
CMPSDR2	$JOBCTLM	JC
CMPSDR3	$JOBCTLM	JC
CNIOAG	$JOBCTLD	CY
CNTROVFL	$JOBCTLM	JH
CNUNCO	$JOBCTLJ	GS

Subroutine: Converts the operand SYSXXX to system and unit class in the location UNCLOR.

CNUNCO1	$JOBCTLJ	GS
CNVBCD	$JOBCTLF	DQ

Subroutine: Converts data to EBCDIC for output.

COMPUBND	$JOBCTLF	DQ
COMPUTF1	$JOBCTLG	EC
COMSET	$JOBCTLM	JJ

Subroutine: Calculates the first, current, and last OBR disk addresses and moves them to the SDR communication region.

COMSET1	$JOBCTLM	JJ
COMTBL	$JOBCTLJ	*

Beginning of table of magnetic tape commands.

CONCAT	$JOBCTLK	HM

Subroutine: Combines the operand fields of two control statements, the first control statement contains a continuation punch.

CONDSCK	$JOBCTLD	CJ
CONTINUE	$JOBCTLN	KB
CONT1	$JOBCTLA	BB
CONTROL	$JOBCTLA	BB
COPYLP	$JOBCTLJ	GD
COPYSW	$JOBCTLG	EG
CRJBSQ	$JOBCTLG	EA
CRTBLD	$JOBCTLJ	GP
CTIMR	$JOBCTLG	*

A 4-byte area in partition register save area used to save job start time.

CTRLSW	$JOBCTLA	BD
C48	$JOBCTLG	EM
C60	$JOBCTLG	EM
DATE	$JOBCTLJ	GH
DCUXTN	$JOBCTLA	BD
DDSW	$JOBCTLD	*

BANK1+1, bit 0: If on specifies DVCDN in progress.

- Set ON: DVCDN statement processor.
- Set OFF: INITL subroutine.
- Controls exit from the RESET statement processor.

DECK	$JOBCTLG	EM
DETACH	$JOBCTLF	DL

Entry into the UNBATCH statement processing routine.

DETACH1	$JOBCTLF	DL
DETACH2	$JOBCTLF	DL
DETACH2A	$JOBCTLF	DL
DETACH3	$JOBCTLF	DL
DETCHR	$JOBCTLF	DL
DEVERR	$JOBCTLM	JN
DFB	$JOBCTLA through $JOBCTLM	*

Data file block (Figure 6).

DIBRC	$JOBCTLJ	GH
DLAB	$JOBCTLK	HF
DLBL	$JOBCTLK	HE
DLBL1	$JOBCTLK	HE
DLBOUT	$JOBCTLK	HF
DNEERR	$JOBCTLJ	GU
DOP34	$JOBCTLK	HM

Subroutine: Converts creation date and expiration date to binary and stores in label output area.

DOWN	$JOBCTLF	DE
DQXTNT	$JOBCTLG	FC
DSKIND	$JOBCTLA	BH
DSKINT	$JOBCTLA	BD

Subroutine to perform Job Control initialization for DASD. If SYSLST/SYSPCH are assigned to disk, the available record count is checked. All extent JIBs currently attached to programmer units are unassigned and placed on the free list. This subroutine is a part of $JOBCTLA and is overlaid when any other phase of Job Control is loaded.

DUMP	$JOBCTLG	EP
DVCDN	$JOBCTLF	DH
DVCDNC	$JOBCTLF	DJ
DVCDNL	$JOBCTLF	DH
DVCDNS	$JOBCTLF	DJ
DVCDNX	$JOBCTLF	DJ
DVCDN1	$JOBCTLF	DH
DVCDN14	$JOBCTLF	DH
DVCDN14A	$JOBCTLF	DH

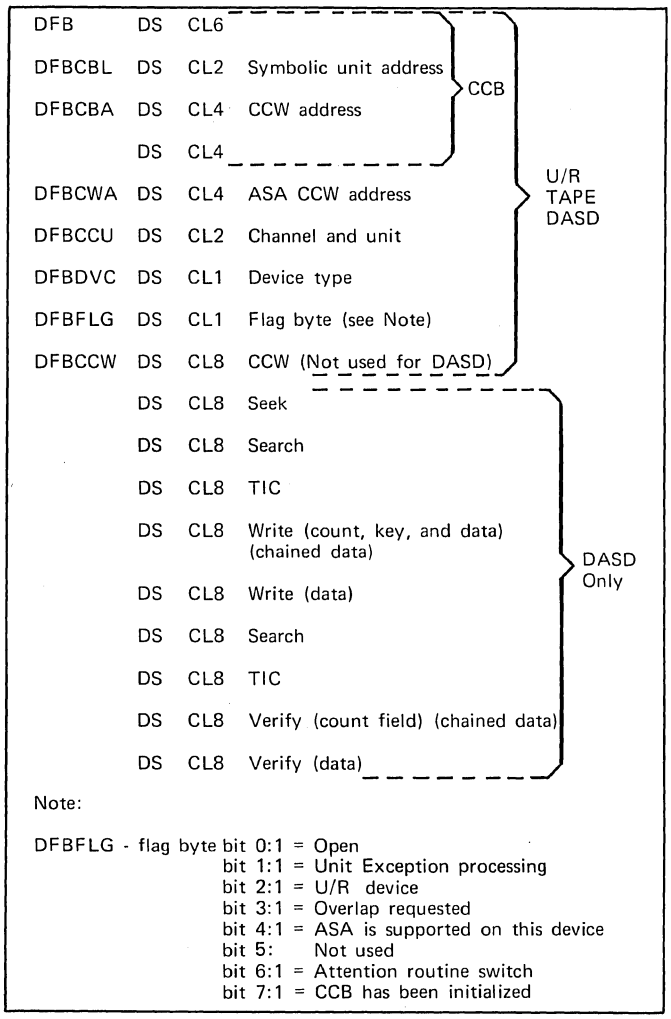


Figure 6. DFB Format

DVCDN14B	$JOBCTLF	DH
DVCDN2	$JOBCTLF	DH
DVCDN3	$JOBCTLF	DH
DVCDN4	$JOBCTLF	DH
DVCDN5	$JOBCTLF	DH
DVCDN6	$JOBCTLF	DJ
DVCDN7	$JOBCTLF	DJ
DVCDN8	$JOBCTLF	DJ
DVCUP	$JOBCTLF	DK
EDTEST	$JOBCTLG	EK
EJBKSWCH	$JOBCTLN	KB
ENDINC	$JOBCTLJ	GD
EODRTRN	$JOBCTLM	JQ
EODTIME	$JOBCTLM	JQ
EOFKY0	$JOBCTLM	JG
EOFKY1	$JOBCTLM	JG
EOFPRC	$JOBCTLA	BJ
EOJ	$JOBCTLG	EC
	$JOBCTLN	KA
EOJBY	$JOBCTLG	EB
EOJNRF	$JOBCTLG	EC
EOJOFF	$JOBCTLG	EB
EOJ1	$JOBCTLG	EB

EOLSW $JOBCTLD *

BANK1, bit 4: If on, specifies that the end of a class in the LUB table has been reached.

- Set ON: SCANLUB subroutine.
- Set OFF: INITL subroutine and SCANLUB subroutine.
- Controls processing in:

  1. ASSGN statement processor.
  2. DVCDN statement processor.
  3. LISTIO statement processor.

EQTEST	$JOBCTLG	EL
EREXIT	$JOBCTLJ	GF
EREXIT1	$JOBCTLJ	GF
ERRAP	$JOBCTLA	BM
ERRAP1	$JOBCTLA	BM
ERRCNL	$JOBCTLA	BM
ERRRTN	$JOBCTLA	BM
	$JOBCTLD	CY
ERRRTN1	$JOBCTLD	CY
ERRRTN2	$JOBCTLA	BM
	$JOBCTLD	CY
ERRS	$JOBCTLG	FQ
EXBKSWCH	$JOBCTLN	KB
EXCCON	$JOBCTLA	BJ
EXCECP	$JOBCTLA	BJ
EXCEDT	$JOBCTLG	EF
EXCEOF	$JOBCTLA	BJ
EXCP	$JOBCTLD	CX
EXCPGO	$JOBCTLA	BJ
EXCPRG	$JOBCTLA	BH

Root phase subroutine to perform I/O on a specified logical unit. If the file has not been opened, this routine initializes its DFB including the CCB before executing the I/O.

EXCPROG $JOBCTLD CX

Subroutine: Performs I/O on tape when user tape density is to be used.

- The symbolic unit address (class and order) is supplied in register POINT1.
- The PUB address is supplied in register POINT2.

EXCPROG1 $JOBCTLD CX

Entry point to the EXCPROG subroutine when IBM standard tape density is to be used.

EXCPROG2	$JOBCTLD	CX
EXCPROG3	$JOBCTLD	CX
EXCPROG4	$JOBCTLD	CX
EXCPSVC	$JOBCTLA	BJ

---

*Listing Only.

```
EXCUSR      $JOBCTLG        EF
EXEC        $JOBCTLG        EF
            $JOBCTLN        KA
EXEC1       $JOBCTLG        EF
EXECBACK    $JOBCTLN        KB
EXIT        $$BLSTIO        LB
EXPEXT      $JOBCTLA        BJ
EXTNT       $JOBCTLK        HJ
EXTRLC      $JOBCTLA        BN
```

Root phase subroutine adds relocation factor to external symbols.

```
EXTRTN      $JOBCTLA        BN
FDDKCODE    $JOBCTLK        HF
FDKDFAUL    $JOBCTLJ        *
```

Default values for disk label vol. sequence through system code.

```
FDKIJ       $JOBCTLK        HE
FDKIJ1      $JOBCTLK        HE
FDKIJ2      $JOBCTLK        HE
FDKTDAT     $JOBCTLK        HD
FDKTDAT1    $JOBCTLK        HD
FDKTDAT2    $JOBCTLK        HD
FDKTID      $JOBCTLK        HD
```

Subroutine: Processes FILE ID and DATE operands in DLBL and TLBL statements.

```
FDKTV       $JOBCTLK        HC
```

Subroutine: Initialization for DLBL, TLBL, and VOL statement processing routines.

```
FDKTVNM     $JOBCTLK        HC
```

Subroutine: Processes FILENAME operand.

```
FDKTV1      $JOBCTLK        HC
FDKTV2      $JOBCTLK        HC
FDK1        $JOBCTLK        HE
FDK2        $JOBCTLK        HE
FDSYSU      $JOBCTLJ        GS
FEBIN       $JOBCTLK        HB
FEBIN1      $JOBCTLK        HB
```

Subroutine: Processes bin numbers operand in EXTENT statement.

```
FESEQ       $JOBCTLK        HL
```

Subroutine: Processes EXTENT SEQUENCE operand.

```
FESEQCK     $JOBCTLK        HK
FESPLIT     $JOBCTLK        HK
FESPLIT1    $JOBCTLK        HK
FESPLIT2    $JOBCTLK        HK
FESPLIT3    $JOBCTLK        HK
FESPLIT4    $JOBCTLK        HK
FESYSX3     $JOBCTLK        HJ
FESYX2      $JOBCTLK        HJ
FETCHN      $JOBCTLG        EB
FETCHR      $JOBCTLG        EK
FETCHRF     $JOBCTLA        BC
```

---

*Listing Only.

```
FETINSRT    $JOBCTLK        HC
```

Subroutine: Inserts numeric operands in output block for output to label cylinder on SYSRES.

```
FETINSR0    $JOBCTLK        HC
FETINSR1    $JOBCTLK        HC
FETINSR3    $JOBCTLK        HC
FETINSR4    $JOBCTLK        HC
FETINSR5    $JOBCTLK        HC
FETK        $JOBCTLK        HK
FETKNO      $JOBCTLK        HK
FETKNO1     $JOBCTLK        HK
FETKTK      $JOBCTLK        HL
```

Subroutine: Processes START TRACK and NUMBER of TRACKS operand in EXTENT statement.

```
FETKTK1     $JOBCTLK        HL
FETKTK2     $JOBCTLK        HL
FETYPE      $JOBCTLK        HJ
FETYPED     $JOBCTLK        HJ
FETYPEI     $JOBCTLK        HK
FETYPEI4    $JOBCTLK        HK
FETYPE1     $JOBCTLK        HJ
FEVSER      $JOBCTLK        HJ
FEVSER1     $JOBCTLK        HJ
FEVSER2     $JOBCTLK        HJ
FEVSER3     $JOBCTLK        HJ
FEVSER4     $JOBCTLK        HJ
FICLS       $JOBCTLA        *
            through
            $JOBCTLM
```

A 6-byte field that contains first in class lists for background, foreground 2, and foreground 1 system and programmer units, translated from 4-byte FICL in supervisor. (Figure 7)

	BG		F2		F1	
FICLS	SS	PP	SS	PP	SS	PP
NICLS	SS	PP	SS	PP	SS	PP

Where: SS = System Class
PP = Programmer Class

Figure 7. Format for NICLS and FICLS

```
FICNIC      $JOBCTLA        BA
FINISH      $JOBCTLN        KB
FINIS2      $JOBCTLG        EK
FINOPN      $JOBCTLA        BH
FLOC        $JOBCTLD        *
```

Label of a 2-byte location that holds the address of the first LUB of a class. Loaded by the GETLAN subroutine.

```
FNDARG      $$BLSTIO        LB
```

Subroutine: Used by the LISTIO statement processor to determine the operand of the LISTIO statement.

```
FNDARG1     $$BLSTIO        LB
FNDARG5     $$BLSTIO        LB
```

```
FNIOAG      $JOBCTLD        CY
FSCAN       $JOBCTLK        HB
     Subroutine:  Scans operands of label
     statements and sets condition code for
     operand omitted, operand length equal
     to 1, or operand length equal to
     greater than 1.

FSCAN1      $JOBCTLK        HB
FTEND       $JOBCTLK        HB
FTEND1      $JOBCTLK        HB
FTKTBL      $JOBCTLG        *
     Beginning of label cylinder track
     allocation table.

FULERR      $JOBCTLM        JN
FXDFAUL     $JOBCTLJ        *
     Beginning of standard default table
     for volume serial through logical
     unit.

FXDFAUL1    $JOBCTLJ        *
     Beginning of current default table
     built by DLBL statement processing
     routine to be used when processing
     EXTENT statement.

GETDUR      $JOBCTLG        FA
     Subroutines:  Computes job duration
     and converts to EBCDIC.

GETIME      $JOBCTLG        FA
     Subroutine:  Computes time of day and
     converts it to EBCDIC.

GETJIB      $JOBCTLD        CR
GETJIBSW    $JOBCTLD        CR
GETJIB1     $JOBCTLD        CR
GETJIB2     $JOBCTLD        CR
GETLAN      $JOBCTLD        CM
     Subroutine:  Determines number of
     logical units in a class and address
     of first LUB of the class in any one
     partition.

GETPUB      $JOBCTLG        FD
     Subroutine:  Computes the address of a
     given PUB entry and stores it in
     register POINT3.

GO          $JOBCTLN        KA
GOCAT       $JOBCTLG        EN
GTNAME      $JOBCTLG        EG
GTNXNT      $JOBCTLG        EH
GTNXOP      $JOBCTLG        GP
HEADNO      $JOBCTLA        BJ
HEXCON      $JOBCTLD        CP
            $JOBCTLJ        GS
     Subroutine:  Converts the operand
     X'CUU' to binary and saves it in the
     location IORGSA.

HOLD        $JOBCTLJ        GA
HOLD1       $JOBCTLJ        GA
HOLD2       $JOBCTLJ        GA
IDCHK       $JOBCTLM        JF
IDMOVE      $JOBCTLM        JF
IDRTN       $JOBCTLM        JP

IGNORE      $JOBCTLA        BG
ILUS        $JOBCTLD        CY
INAERR      $JOBCTLJ        GU
INCLUDE     $JOBCTLJ        GD
INDSEQ      $JOBCTLK        HG
INDVTP      $JOBCTLD        CY
INDWR       $JOBCTLA        BG
INITL       $JOBCTLD        CM
            $JOBCTLF        DP
            $JOBCTLG        EK
INNXEN      $JOBCTLJ        GP
INSW        $JOBCTLD        CK
INVASGN     $JOBCTLD        CJ
IOROUT      $JOBCTLG        FC
     Subroutine:  Performs I/O and waits
     for traffic bit to be posted.

IOROUTW     $JOBCTLG        FC
     Subroutine:  Waits for previous I/O
     operation to complete.

IPLEODYS    $JOBCTLM        JP
IPLLOOP     $JOBCTLM        JP
IPLNVLD     $JOBCTLM        JP
IPLRTRN     $JOBCTLM        JP
ISCKSQ      $JOBCTLK        HG
ISSUIO      $JOBCTLA        BJ
ISTYP4      $JOBCTLK        HG
IVDS        $JOBCTLD        CY
IVSERR      $JOBCTLJ        GU
JBCSW0      $JOBCTLA        *
     Displacement 56 in the communication
     region (COMREG + 56).  See Figure 10.

JBCSW1      $JOBCTLA        *
     Displacement 57 in the communication
     region (COMREG + 57).  See Figure 10.

JBCSW2      $JOBCTLA        *
     Displacement 58 in the communication
     region (COMREG + 58).  See Figure 10.

JBCSW3      $JOBCTLA        *
     Displacement 59 in the communication
     region (COMREG + 59).  See Figure 10.

JBCSW4      $JOBCTL         *
     Bit 0:  0 = Job control statement
                 started with a //.

             1 = Job control statement
                 started without a //.

     Bit 1:  0 = Statement is to be logged
                 on SYSLST in case of
                 error.

             1 = Statement is not to be
                 logged on SYSLST in case
                 of error.

     Bit 2:  0 = Statement is to be logged
                 on SYSLOG in case of error

--------------------

     *Listing Only.
```

1 = Statement is not to be logged on SYSLOG in case of error.

Bit 3: 0 = No time stamping on SYSLOG or SYSLST.

1 = Time stamping will occur. Turned on by TIMOUT routine $JOBCTLG if timer is running.

Bit 4: 0 = Reserved.

Bit 5: 0 = No auto cancel.

1 = Automatic cancel on error.

Bit 6-7: Reserved.

JBCSW5 $JOBCTLA *
Bits 0-2: Always 0.

Bit 3: 0 = A label block is in output area ready to be written on the SYSRES label cylinder.

Bit 4: 1 = Required label statement to follow.

Bit 5: 1 = Only an XTENT statement may follow.

Bit 6: 1 = Only an EXTENT statement may follow.

Bit 7: 1 = Only a DLAB or TPLAB statement may follow.

JBINPR $JOBCTLG EC
JIBCHN $JOBCTLD *
Label of a 2-byte location used as a 2-byte work area for LUBs and JIBs.

JIBPTR $JOBCTLA BD
$JOBCTLG FC
JIBSCN $JOBCTLG FD
Subroutine: Scans the JIB table and makes the next JIB in a chain available for processing.

JOB $JOBCTLG EA
JOBCTL $JOBCTLA BA
JOBCTLM $JOBCTLM JA
JOBCTL4 $JOBCTLA BB
JOBCTL5 $JOBCTLA BB
KEY1 $$BLSTIO LB
KEY2 $$BLSTIO LB
KEY3 $$BLSTIO LB
LAXERR $JOBCTLG FG
$JOBCTLK HS
LBLDST $JOBCTLJ *
Beginning of DSECT used in building label information for output to label cylinder on SYSRES.

LBLOUT $JOBCTLG FE
$JOBCTLK HN
Subroutine: Writes label information in the VOL area of SYSRES from the job control label area.

LBLOUTF $JOBCTLK HN
LBLOUT0A $JOBCTLG FE
LBLOUT0B $JOBCTLG FE
$JOBCTLK HN
LBLOUT1 $JOBCTLG FE
$JOBCTLK HP
LBLOUT2D $JOBCTLG FF
$JOBCTLK HP
LBLTYP $JOBCTLK HA
LBTOUT $JOBCTLK HA
LDBASE $JOBCTLG EJ
LGCHKDON $JOBCTLM JA
LHRTN $$BLSTIO LD
LINCNT $JOBCTLA *
Maximum line count for SYSLST. Maintained in COMREG + 78.

LINK $JOBCTLG EN
LIOCUU $JOBCTLF DD
LIOEOJ $JOBCTLF DA
LIOEOJ1 $JOBCTLF DA
LIOL $JOBCTLF DB
LIOLL201 $JOBCTLF DB
LIOLL3 $JOBCTLF DB
LIOL1 $JOBCTLF DB
LIOL2 $JOBCTLF DB
LIOL202 $JOBCTLF DB
LIOL4 $JOBCTLF DB
LIOSW $JOBCTLD *
BANK1, bit 6: If ON specifies LISTIO in progress.

- Set ON: LISTIO statement processor.
- Set OFF: UNPA subroutines
  SFPPE subroutine
  INITL subroutine.
- Controls processing in the SFPPE subroutine.

LIOSYX $JOBCTLF DD
LIST $JOBCTLG EM
LISTIO $JOBCTLF DA
LISTX $JOBCTLG EM
LNAERR $JOBCTLG FG
LNKINT $JOBCTLA BH
LNKNOP $JOBCTLG EG
LNKOUT $JOBCTLJ GT
Subroutine: Controls block count and byte count for writing on SYSLNK.

LOADRS $JOBCTLK HQ
LOADRS1 $JOBCTLK HQ
LOG $JOBCTLJ GL

---

*Listing Only

LOGCHK $JOBCTLA BG
Root phase subroutine sets switches in JBCSW0 (COMREG + 56) bits 6 and 7 to indicate SYSLOG device type and assignment.

LOGDON $JOBCTLM JH
LOGIN $JOBCTLA BG
LOGOUT $JOBCTLA BE
Root phase subroutine outputs a control statement or a message on SYSLOG.

LOGPRT $JOBCTLA BG
LOOP $JOBCTLG EP
LSTOUT $JOBCTLA BE
Root phase subroutine outputs a control statement or a message on SYSLST.

LUBADD $JOBCTLD *
Label of a 2-byte location that holds the address of the LUB being assigned. Loaded by the subroutine, SYSXXX.

LUBCOM $JOBCTLD *
Label of a 2-byte location used for temporary storage of a LUB for comparison. Used by the SCNLUB subroutine.

LUBCOUNT $JOBCTLG FB
LUBSCN $JOBCTLG FD
Subroutine: Makes the address of the next LUB entry of a class available in register POINT1.

MAP $JOBCTLF DK
MMDDYY $JOBCTLJ GG
MMDD1 $JOBCTLJ GG
MODSW $JOBCTLD *
BANK1+1, bit 7: If on, specifies that the mode must be set for this assignment.

- Set ON: ASSGN statement processor.
- Set OFF: INITL subroutine.
- Controls processing in the ASSGN statement processor.

MOVLOP $JOBCTLG EK
MSGOUT $JOBCTLA BE
Root phase entry into the LOGOUT subroutine. Sets switches to allow output on both SYSLST and SYSLOG.

MTC $JOBCTLJ GE
MTC1 $JOBCTLJ GE
MTC2 $JOBCTLJ GE
MTC3 $JOBCTLJ GE
MTC4 $JOBCTLJ GE
MTC5 $JOBCTLJ GF
MTFILL $JOBCTLJ *
Default value for system code in tape label '0000000DOS/TOS/360'.

MTNCNT $JOBCTLA BK
MVKIND $JOBCTLF DQ
MVMVRT $JOBCTLG EJ
MVQUO $JOBCTLG EP
NDSCAN $JOBCTLA BF
$JOBCTLG EP
$JOBCTLK HD
NDSCAN1 $JOBCTLK HD
NDSERR $JOBCTLJ GU
NDSERR1 $JOBCTLJ GF
NDTERR $JOBCTLG FG
$JOBCTLJ GU
NEWLUB $JOBCTLD *
Label of a 2-byte work area in which a LUB is built for a new assignment. The subroutine TXCUU computes a PUB pointer and stores it in byte 0.

NEWPUB $JOBCTLD *
Label of a 2-byte location that holds the address of the PUB to be assigned to the LUB that is being assigned. Loaded by the subroutine TXCUU.

NEWXTN $JOBCTLK HH
NEXT $JOBCTLJ GN
NEXTPUB $JOBCTLM JD
NEXTTR $JOBCTLM JD
NICLS $JOBCTLA through $JOBCTLK *
A 6-byte field that contains number in class lists for background, foreground 2, foreground 1 system and programmer units, translated from 4-byte NICL in supervisor. (Figure 7)

NLISTX $JOBCTLG EM
NLSERR $JOBCTLK HS
NLST $JOBCTLG EG
NLUERR $JOBCTLJ GU
$JOBCTLK HS
NOC $JOBCTLD *
Label of a 2-byte location that holds the number of logical units of a class. Loaded by a the GETLAN subroutine.

NOCATL $JOBCTLG ED
NOCLEAR $JOBCTLG EC
NODCUX $JOBCTLK HH
NODECK $JOBCTLG EM
NODSYS $JOBCTLA BD
NODUMP $JOBCTLG EP
NOEERR $JOBCTLA BL
NOEERT $JOBCTLA BL
NOEOJ $JOBCTLA BG
NOERRS $JOBCTLG EM
NOEXC $JOBCTLD CQ
NOFP $JOBCTLG EG
NOIPLMSG $JOBCTLM JP
NOIPT $JOBCTLG ED
NOIPTA $JOBCTLG ED
NOIPT1 $JOBCTLG ED

---

*Listing Only.

NOIPT2	$JOBCTLG	ED
NOIPT3	$JOBCTLG	ED
NOIPT7	$JOBCTLG	ED
NOKYTST	$JOBCTLM	JF
NOLEAP	$JOBCTLJ	GG
NOLINK	$JOBCTLG	EP
NOLIST	$JOBCTLG	EM
NOLOG	$JOBCTLJ	GL
NOMPSS	$JOBCTLG	EG
NOMRJB	$JOBCTLD	CY
NOPHDB	$JOBCTLG	EH
NOPSE	$JOBCTLG	ED
NOPTO	$JOBCTLM	JH
NORAS	$JOBCTLJ	GJ
NOREL	$JOBCTLD	CM
	$JOBCTLF	DP
NORELSW	$JOBCTLD	CM
	$JOBCTLF	DP
NORESET	$JOBCTLJ	GF
NORESTOR	$JOBCTLG	EC
NOSUBMSG	$JOBCTLM	JQ
NOSYM	$JOBCTLG	EM
NOTAP	$$BLSTIO	LA
NOTDSK	$JOBCTLA	BJ
NOTFG	$JOBCTLN	KA
NOTFG1	$JOBCTLN	KB
NOTINSYS	$JOBCTLM	JA
NOTLAST	$JOBCTLN	KB
NOTLIST	$JOBCTLG	EE
NOTNINTY	$JOBCTLM	JN
NOTNOS	$JOBCTLG	EL
NOTSEQ	$JOBCTLK	HG
NOUPDP	$JOBCTLM	JH
NOXREF	$JOBCTLG	EM
NOZERO	$JOBCTLM	JN
NO2314	$JOBCTLA	BA
NTINJB	$JOBCTLA	BL
NULCHK	$JOBCTLJ	GB
NUMCON	$JOBCTLD	CM
	$JOBCTLJ	GT
	$JOBCTLK	HR

Subroutine: Converts EBCDIC character 0-9 to binary in WRKRG1.

NVAERR	$JOBCTLJ	GU
NVOERR	$JOBCTLG	FG
NVOPTN	$JOBCTLG	EL
NVSERR	$JOBCTLA	BM
NXPBNT	$JOBCTLJ	GR
NXTBIT	$JOBCTLJ	GK
NXTJIB	$JOBCTLA	BD
NXTLUB	$JOBCTLA	BD
NXTOPRD	$JOBCTLJ	GN
OBRCON	$JOBCTLM	JF
OBRKY0	$JOBCTLM	JF
OBRKY0A	$JOBCTLM	JF
OBRKY1	$JOBCTLM	JF
OBRONLY	$JOBCTLM	JE
OBRRCD	$JOBCTLM	JF
OBRREAD	$JOBCTLM	JM

Subroutine: Issues read commands for the OBR portion of the recorder file.

OBRSET	$JOBCTLM	JM

Subroutine: Sets up the I/O area for OBR record formatting.

OBRSTRT	$JOBCTLM	JF
OBRWRT	$JOBCTLM	JC
OBRWRT1	$JOBCTLM	JC
OBRWRT2	$JOBCTLM	JC
OBR0DON	$JOBCTLM	JF
OBR1DON	$JOBCTLM	JF
OERRTN	$JOBCTLA	BL
OLDPUB	$JOBCTLD	*

Label of a 2-byte location used to hold the address of the PUB currently assigned to the LUB that is being reassigned. Loaded by the subroutine SYSXXX.

ONLYOBR	$JOBCTLM	JF
ONOLOG	$JOBCTLJ	GL
OPAUSE	$JOBCTLJ	GL
OPENRCO	$JOBCTLM	JJ
OPENSET	$JOBCTLM	JJ

Subroutine: Opens the recorder file (SYSREC) according to the initialization of the DTF table. If RF=CREATE, the file is opened as an output file. If RF=YES, the file is an input file.

OPLBNF	$JOBCTLG	FF
	$JOBCTLK	HP
OPLOG	$JOBCTLJ	GL
OPNERR	$JOBCTLM	JN
OPNLNK	$JOBCTLG	FA

Subroutine: Opens SYSLNK DFB.

OPNUMH	$JOBCTLA	*

A 1-byte field which is the data byte in a LOAD instruction. This byte maintains a parameter count by the control statement processing routines. It is reset to zero following each successful control statement read operation.

OPTION	$JOBCTLG	EL
OPTLOG	$JOBCTLG	EP
OPTNLG	$JOBCTLG	EM
OTSERR	$JOBCTLG	FG
	$JOBCTLJ	GU
	$JOBCTLK	HS
OTSERR1	$JOBCTLG	FG
	$JOBCTLJ	GU
OUTLBL	$JOBCTLK	HH
OUTPUT	$JOBCTLD	CN
OUTPUTS	$JOBCTLD	CN
	$JOBCTLF	DP
OUTPUT1	$JOBCTLD	CN
	$JOBCTLF	DP
OUTPUT2	$JOBCTLD	CN
	$JOBCTLF	DP
OVRLP1	$JOBCTLA	BJ
OVRVCT	$JOBCTLA	*

Beginning address of the overlay area where $JOBCTLD, $JOBCTLG, $JOBCTLF, $JOBCTLK and $JOBCTLJ are loaded. This address is within the $JOBCTLA

---

*Listing Only.

root phase. $JOBCTLA initialized the base register, BASRG2, with this address each time it is loaded.

PACKAG	$JOBCTLK	HH
PARSTD	$JOBCTLG	EN
PARSTD1	$JOBCTLG	EN
PASSRF	$JOBCTLG	EB
PAUSE	$JOBCTLJ	GL
PCHOUT	$JOBCTLJ	GN
PCILSUP	$JOBCTLG	EN
PCILSUPR	$JOBCTLG	EN
PIKCMA	$JOBCTLD	CU
PIKCPL	$JOBCTLD	CU
PIKFLG	$JOBCTLD	CU
PIKFLG7X	$JOBCTLD	CU
PMB	$JOBCTLM	JA
PNPERR	$JOBCTLG	FG
	$JOBCTLJ	GU
PRCNTCMP	$JOBCTLM	JN
PRCNTFLL	$JOBCTLM	JF
PREOPR	$JOBCTLJ	GP
PRGUNT	$JOBCTLJ	GS
PROGSW	$JOBCTLD	*

BANK1, bit 5: If ON, specifies that the programmer LUBs are to be scanned.

- Set ON: LISTIO statement processor, ASSGN statement processor, RESET statement processor.
- Set OFF: INITL subroutine ASSGN statement processor.
- Controls processing in:
  1. LISTIO statement processor.
  2. ASSGN statement processor.

PSHRTN	$$BLSTIO	LD

Subroutine: Used by the LISTIO statement processor to build a header print line.

PSHRTN1	$$BLSTIO	LD
PTR	$JOBCTLA	BN
	$JOBCTLD	CM
	$JOBCTLF	DP
	$JOBCTLG	EK
PUBMSK	$JOBCTLD	CS
	$JOBCTLF	DR
PUBNEXT	$JOBCTLM	JD
PUIF	$$BLSTIO	LA

Subroutine: Used by the LISTIO and DVCDN statement processors. Extracts information from the PUB preparatory to building a print line.

PUIFT3	$$BLSTIO	LA
PUIF1	$$BLSTIO	LA
PUIF4	$$BLSTIO	LA
PUIF5	$$BLSTIO	LA
QEND	$JOBCTLG	EP
RANXJB	$JOBCTLG	FD
RASCAN	$JOBCTLG	FD

Subroutine: Scans program LUBs and sets assignment flags in assigned PUBs.

RASSGN	$JOBCTLG	FD
RASSYS	$JOBCTLM	JA
RDCID	$JOBCTLG	EG
RDRIN	$JOBCTLA	BG
RDSTMT	$JOBCTLA	BG

Root phase subroutine reads a statement from SYSRDR or SYSLOG. The job control switch, JBCSW0 (COMREG + 56) bit 0, is tested to determine if SYSRDR or SYSLOG is to be used. If SYSLOG is specified but it is not a 1052, the switch is changed to indicate SYSRDR and the subroutine is reentered via the control routine.

RECORD	$JOBCTLM	JP
RELEASE	$JOBCTLJ	GA
RELJBC	$JOBCTLA	BN

Root phase subroutine modifies address constants making Job Control relocatable.

RELLOP	$JOBCTLD	CM
	$JOBCTLF	DP
	$JOBCTLG	EK
RELPRG	$JOBCTLG	EJ
RELTST	$JOBCTLG	EJ
RESDN	$JOBCTLA	BA
RESET	$JOBCTLF	DF
	$JOBCTLJ	GF
RESETCL	$JOBCTLF	DF
RESETSW	$JOBCTLD	*

BANK1, bit 7: If ON, specifies RESET in progress.

- Set ON: RESET statement processor
- Set OFF: INITL subroutine.
- Controls exit from the RSTSTD subroutine.

RESET01	$JOBCTLF	DF
RESET1	$JOBCTLF	DF
RESET101	$JOBCTLF	DG
RESET11	$JOBCTLF	DG
RESET2	$JOBCTLF	DG
RESET3	$JOBCTLF	DG
RESET4	$JOBCTLF	DG
RESET5	$JOBCTLF	DG
RESET8	$JOBCTLF	DG
RESFCH	$JOBCTLG	EF
RESTORE	$JOBCTLG	EC
RETADD	$JOBCTLF	DA
RETSW	$JOBCTLD	*

BANK1, bit 3:

- Set ON: SFPPE subroutine. ASSGN statement processor.

---

*Listing Only.

- Set OFF: ASSGN statement processor.

DVCDN statement processor INITL subroutine.

- Controls processing in the SFPPE subroutine.
- Controls processing in the GETJIB subroutine.

```
RETURN      $JOBCTLN        KB
RFEND       $JOBCTLJ        GJ
RFNO        $JOBCTLJ        GJ
RFSERR      $JOBCTLM        JN
RFSET       $JOBCTLJ        GJ
RFYES       $JOBCTLJ        GJ
RLINDT      $JOBCTLA        BG
RLNCNT      $JOBCTLA        *
```

Save area for SYSLST current line count.

```
RLSENT      $JOBCTLJ        GA
RMR         $JOBCTLD        CK
RNAERR      $JOBCTLA        BL
RNAME       $JOBCTLG        *
```

An 8-byte area in partition register save area used to save job name.

```
RNGTOP      $JOBCTLD        CM
            $JOBCTLK        HR
ROD         $JOBCTLM        JH
RODCMP      $JOBCTLM        JH
RODERR      $JOBCTLM        JH
RODSCAN     $JOBCTLM        JH
RSPASG      $JOBCTLG        FB
RSPPAE1     $JOBCTLJ        GR
RSPPEA      $JOBCTLJ        GR
RSSASG      $JOBCTLG        EA
RSSASG1     $JOBCTLG        EA
RSSASG2     $JOBCTLG        EB
RSTASG      $JOBCTLG        FB
```

Subroutine: Restores system and programmer assignments to standard.

```
RSTCOM      $JOBCTLG        EB
RSTRT       $JOBCTLK        HQ
RSTSTD      $JOBCTLD        CR
```

Subroutine: Restores a LUB to its standard I/O assignment or unassigns a nonstandard assignment.

```
RSTSTD1     $JOBCTLD        CR
RSTSTD3     $JOBCTLD        CR
RSTSTD4     $JOBCTLD        CR
RSTSTD5     $JOBCTLD        CR
RSTSW4      $JOBCTLG        EA
SAMEDIB     $JOBCTLD        CJ
SAMELUB     $JOBCTLD        CJ
SCANR1      $JOBCTLA        BF
```

Root phase subroutine scans a control statement and makes a parameter available for processing.

- A comma, blank, or equal sign ends the scan.
- Register POINT1 contains the address of the 1st character of the parameter.
- Register POINT2 contains the number of characters remaining to be scanned.
- Register POINT3 contains a count of the number of characters in the parameter. This character count is 1 less than the actual character count and is used to control the character count in move, and compare instructions.
- Between entries into the subroutine, registers POINT1 and POINT2 are saved in an area labeled TMPAR1.
- The entry SCANR1, when it is desired, scans for the operation field.
- The entry SCANR2 scans for prime operands.
- The entry SCANR3 scans for the parameter of a prime operand.

Example:

```
SET RCLST = 2000, LINCNT = 99,...
```

```
SCANR2      $JOBCTLA        BF
SCANR3      $JOBCTLA        BF
SCNINT      $JOBCTLG        FB
```

Subroutine: Initializes for LUB and PUB table scan.

```
SCNJIB      $JOBCTLD        CV
```

Subroutine: Computes address of a JIB in register POINT3. The stored LUB of the JIB is moved to the location JIBCHN.

```
SCNLUB      $JOBCTLD        CV
```

Subroutine: Makes all LUBs of a class available one at a time. The current LUB is made available in the location JIBCHN. The address of the next LUB is saved in the location SLADD. The residual number of LUBs (in a class) is saved in the location SNICL.

```
SCNRL1      $JOBCTLA        BF
SCNRL2      $JOBCTLA        BF
            $JOBCTLK        HD
SDRREAD     $JOBCTLM        JM
```

Subroutine: Issues read commands for the SDR portion of the recorder file.

```
SDRSRCH     $JOBCTLM        JD
SDRUNRC     $JOBCTLM        JD
SDRUPDT     $JOBCTLM        JK
```

Subroutine: Updates the seek address when searching for an available entry

in the SDR portion of the recorder file.

SDRUPGO	$JOBCTLM	JK
SEOJBACK	$JOBCTLN	KB
SDRWRT	$JOBCTLM	JB
SELRLD	$JOBCTLA	BN
SET	$JOBCTLJ	GG
SETEXT	$JOBCTLJ	GG
SETFLAGS	$JOBCTLK	HF
SETLOD	$JOBCTLG	EF
SETOUTPT	$JOBCTLJ	GD
SETWRT	$JOBCTLA	BJ
SETYES	$JOBCTLJ	GJ
SEUOB	$$BLSTIO	LC

Subroutine: Used by the LISTIO statement processor to build a print line in the output area labeled BUFFER.

SEUOB2	$$BLSTIO	LC
SFNC	$JOBCTLD	CY
SFPPE	$JOBCTLD	CS
	$JOBCTLF	DR
SFPPEDQ	$JOBCTLF	DJ

Subroutine: Used by DVCDN Statement processor to unassign LUBs and JIBs attached to DOWNed device.

SFPPE01	$JOBCTLD	CS
	$JOBCTLF	DR
SFPPE02	$JOBCTLD	CS
	$JOBCTLF	DR
SFPPE03	$JOBCTLD	CU
	$JOBCTLF	DR
SFPPE1	$JOBCTLD	CS
	$JOBCTLF	DR
SFPPE10	$JOBCTLF	DS
SFPPE11	$JOBCTLF	DS
SFPPE12	$JOBCTLF	DS
SFPPE13	$JOBCTLF	DS
SFPPE14	$JOBCTLF	DS
SFPPE15	$JOBCTLF	DS
SFPPE16	$JOBCTLF	DS
SFPPE17	$JOBCTLF	DS
SFPPE18	$JOBCTLF	DS
SFPPE19	$JOBCTLF	DS
SFPPE2	$JOBCTLD	CS
	$JOBCTLF	DR
SFPPE201	$JOBCTLF	DS
SFPPE3	$JOBCTLD	CU
	$JOBCTLF	DR
SFPPE4	$JOBCTLD	CS
	$JOBCTLF	DR
SFPPE5	$JOBCTLD	CS
	$JOBCTLF	DR
SFPPE5A	$JOBCTLD	CS
	$JOBCTLF	DR
SFPPE6	$JOBCTLD	CS
	$JOBCTLF	DR
SFPPE7	$JOBCTLD	CS
	$JOBCTLF	DR
SFPPE8	$JOBCTLD	CT
	$JOBCTLF	DS
SFPPE8A	$JOBCTLD	CT
	$JOBCTLF	DS
SFPPE8B	$JOBCTLF	DS
SFPPE9	$JOBCTLD	CT
	$JOBCTLF	DS
SIMEND	$JOBCTLG	EA
SIMEOJ	$JOBCTLN	KA
SIMRET	$JOBCTLG	EB
SJBKSWCH	$JOBCTLN	KB
SKIPLN	$JOBCTLD	CQ
SLADD	$JOBCTLD	*

Label of a 2-byte location used to hold the address of the next LUB of a class to be scanned. Loaded by the GETLAN subroutine.

SLINCT	$JOBCTLJ	GG
SPCEXC	$JOBCTLA	BJ
SRCHSDR	$JOBCTLM	JE
SSIDRTRN	$JOBCTLM	JQ
SSLOOP	$JOBCTLM	JQ
STDASGN	$JOBCTLG	FC
STDFDSW	$JOBCTLD	*

BANK1, bit 2: If ON, specifies that a stored standard assignment has been found.

- Set ON: SFPPE subroutine.
- Set OFF: SFPPE subroutine INITL subroutine.
- Controls processing in the SFPPE subroutine.

STDLBL	$JOBCTLG	EN
STILOK	$JOBCTLG	EP
STLLMT	$JOBCTLJ	GQ
STLNKA	$JOBCTLG	EG
STOP	$JOBCTLJ	GM
STORBACK	$JOBCTLG	EC
STSCLB	$JOBCTLG	FD
STSRWR	$JOBCTLG	EH
STUCRL	$JOBCTLF	DQ

Subroutine: Writes a line from the I/O area labeled BUFFER and initializes for the next line.

STUF1U	$JOBCTLF	DQ

Subroutine: Converts the upper limit to EBCDIC and stores it in the correct field of a print line for the MAP processor.

STUSPC	$JOBCTLF	DQ

Subroutine: Builds the portion of a MAP print line indicating program prefix and upper limit.

SULB	$$BLSTIO	LC

Subroutine: Used by the LISTIO statement processor to build the LUNIT, LCMNT, LCHNL, LPUNIT, and LMODE fields of a print line.

SULB1	$$BLSTIO	LC
SULB2	$$BLSTIO	LC
SUPLOG	$JOBCTLA	BC

---

*Listing Only

SVCBTRNS $JOBCTLF DA
Subroutine: Fetches the B-transient $$BLSTIO. The register LINKR2 returns control to the calling sequence from the B-transient.

SWBACK $JOBCTLN KB
SXTPOK $JOBCTLK HG
SYM $JOBCTLG EM
SYSDATE $JOBCTLJ GG
SYSPRM $JOBCTLG EP
SYSRF $JOBCTLJ GJ
SYSTABLE $JOBCTLF *
Beginning of logical unit system table used to determine logical unit type and LUB table displacement.

SYSUPI $JOBCTLJ GG
SYSUPI1 $JOBCTLJ GK
SYSUTB $JOBCTLJ *
Beginning of table of LIOCS system units.

SYSXXX $JOBCTLD CW
Subroutine: Converts a logical unit designated as SYSXXX, into:

1. Symbolic unit address (class and order) in the location CLOARD.

2. LUB address in the location LUBAD.

3. PUB pointer of the PUB currently assigned to this logical unit in the location OLDPUB.

4. The internal representation of device type in location NEWTYP.

SYSXXX01 $JOBCTLD CW
SYSXXX1 $JOBCTLD CW
SYSXXX2 $JOBCTLD CW
SYSXXX3 $JOBCTLD CW
SYSXXX4 $JOBCTLD CW
SYSXXX5 $JOBCTLD CW
SYSXXX6 $JOBCTLD CW
TAPINT $JOBCTLA BH
TBLADR $JOBCTLA *
Label of the Phase-Vector Table contained in the root phase ($JOBCTLA). This table is used to determine the correct phase and processing routine required to process a given control statement.

The operation field of the control statement is compared to each entry in the table until an equal is found. The equal entry identifies the correct phase and the displacement within the phase of the branch instruction that directs the program to the correct processing routine. The entry also contains a 1-byte condition switch bank used to control processing for format verification, logging conventions, and cancel procedures for the statement. Figure 8 shows the format of an entry in the Phase-Vector Table.

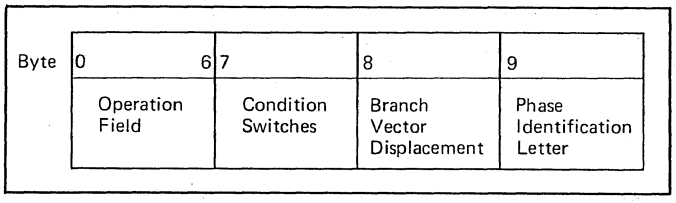


Figure 8. Phase-Vector Table Entry Format

Operation Field: EBCDIC representation of the operation field.

Condition Switches:
Bit 0 - reserved.
1 - statement is to be processed even though a cancel condition exists.

2,3 - Both on; suppress logging. 2 off, 3 on; unconditional SYSLOG logging and conditional SYSLST logging. Both off; conditional logging on SYSLOG and SYSLST.

4 - statement may start with //.

5 - statement may start without //.

6 - statement may start in column 1.

7 - statement may start in other than column 1.

Branch Vector Displacement:
Displacement within the phase that is added to the phase origin address to develop the address of a branch instruction which transfers control to the correct processing routine.

Phase Identification Letter: Contains the EBCDIC character A, D, F, G, J, K, M or N and identifies the job control phase containing the processing routine.

Example of the JOB control statement entry:

```
DC CL7'JOB'

DC X'7A'

DC AL1(12)
```

---

*Listing Only.

DC C'G'

1. The JOB statement is to be processed even if a cancel is being executed.

2. Logging on both SYSLOG and SYSLST is suppressed.

3. The statement may not start without // and may not start in other than column one.

4. The branch-vector table entry is located at a displacement of 12 bytes from the beginning of the phase with suffix 'G' ($JOBCTLG).

TBNHDR	$JOBCTLG	EE
TBNXPB	$JOBCTLG	EE
TBPBLP	$JOBCTLG	EE
TEBHDR	$JOBCTLG	EE
TEBLOP	$JOBCTLG	EE
TEBS	$JOBCTLG	EE
TEST	$JOBCTLN	KA
TESTOP	$JOBCTLM	JA
TEST1	$JOBCTLF	DE
TEST2	$JOBCTLF	DE
TFILL	$JOBCTLK	HA
TIAERR	$JOBCTLD	CY
TIME	$JOBCTLA	*

An 8-byte area used to contain system timer and system time of day, loaded by $JOBCTLA initialization from Hex locations 50-57.

TIMOUT	$JOBCTLG	FA

Subroutine: Used by the JOB and EOJ statement processors to log the time of day.

TIMOUT1	$JOBCTLG	FA
TIMOUT2	$JOBCTLG	FA
TLBL	$JOBCTLK	HB
TMPSW	$JOBCTLD	*

BANK1+1, bit 6: If ON, specifies that the assignment is temporary.

- Set ON: ASSGN statement processor.
- Set OFF: INITL subroutine.
- Controls processing in the ASSGN statement processor.

TNVSERR	$JOBCTLF	DA
TOOMANY	$JOBCTLJ	GN
TPCK	$$BLISTIO	LA
TPLAB	$JOBCTLK	HA
TPLEND	$JOBCTLK	HA
TRANSW	$JOBCTLD	*

BANK1+1, bit 1: If ON, specifies that the Job Control transient, $$BLISTIO, has been loaded in the B-transient area.

- Set ON: SVCBTRNS subroutine.
- Set OFF: LISTIO statement processor, INITL subroutine.
- Controls processing in:
  1. SFPPE subroutine
  2. ERRRTN common error routine.

TRKCLC	$JOBCTLM	JA
TRKCMP	$JOBCTLM	JA
TRKEXH	$JOBCTLM	JA
TRKGO	$JOBCTLM	JA
TRKIN	$JOBCTLM	JA
TRKLST	$JOBCTLM	JA
TRKOBR	$JOBCTLM	JB
TRKOUT	$JOBCTLM	JA
TRKSET	$JOBCTLM	JJ
TSLNKSW	$JOBCTLJ	GD
TSTYPE	$JOBCTLA	BC
TXCUU	$JOBCTLD	CP
	$JOBCTLF	DP
	$JOBCTLJ	GS

Subroutine: Converts the operand X'CUU' from hex to binary.

- PUB address is saved in register POINT4.
- PUB pointer is computed in register WRKRG3.
- Device type (from the PUB) is saved in the location DVCTYP.

TXCUU1	$JOBCTLD	CP
	$JOBCTLF	DP
	$JOBCTLJ	CY
TXCUU1+8	$JOBCTLD	CP
TXCUU2	$JOBCTLD	CP
	$JOBCTLF	DP
	$JOBCTLJ	GS
TXCUU3	$JOBCTLD	CP
	$JOBCTLF	DP
TXCUU4	$JOBCTLD	CP
	$JOBCTLF	DP
T2311	$JOBCTLM	JH
UA	$JOBCTLF	DE
UADN1	$JOBCTLF	DE
UADN2	$JOBCTLF	DE
UADN3	$JOBCTLF	DE
UADN4	$JOBCTLF	DE
UADN5	$JOBCTLF	DE
UADN6	$JOBCTLF	DE
UALUB	$JOBCTLF	DJ
UALUB1	$JOBCTLF	DJ
UANXJB	$JOBCTLG	FC
UASCAN	$JOBCTLG	FC

Subroutine: Restores program LUBs to standard assignments. Detaches program JIBs and places them in free list. Resets program ownership flags in PUBs.

---

*Listing Only.

UASTD	$JOBCTLF	DJ
UCS	$JOBCTLJ	GB
UCSDN	$JOBCTLJ	GC
UCSSCN	$JOBCTLJ	GC
UCS1	$JOBCTLJ	GB
UCS2	$JOBCTLJ	GB
UCS3	$JOBCTLJ	GC
UCS4	$JOBCTLJ	GC
UNA	$JOBCTLF	DM
UNAB	$JOBCTLF	DM
UNAE	$JOBCTLF	DM
UNANENT	$JOBCTLD	CU
UNASGN	$JOBCTLG	FC
UNASSGN	$JOBCTLD	CK
UNA1	$JOBCTLF	DM
UNA1A	$JOBCTLF	DM
UNA2	$JOBCTLF	DN
UNA2A	$JOBCTLF	DN
UNA2AA	$JOBCTLF	DN
UNA2B	$JOBCTLF	DN
UNBLKD	$JOBCTLJ	GD
UNCLOG	$JOBCTLA	BC
UNCU	$JOBCTLF	DC
UNITS	$JOBCTLF	DC
UNITS1	$JOBCTLF	DC
UNITS2	$JOBCTLF	DC
UNITS3	$JOBCTLF	DC
UNITS401	$JOBCTLF	DC
UNITS402	$JOBCTLF	DC
UNITS5	$JOBCTLF	DC
UNITS501	$JOBCTLF	DC
UNITS6	$JOBCTLF	DC
UNITS7	$JOBCTLF	DC
UNITS8	$JOBCTLF	DC
UNITS9	$JOBCTLF	DC
UNPA	$JOBCTLD	CU

Entry into the SFPPE subroutine to unassign a standard assignment. This entry is used by the ASSGN routine and the RSTSTD subroutine.

UNPA1	$JOBCTLD	CU
UNPA3	$JOBCTLD	CU
UNRCSDR	$JOBCTLM	JE
UNRERR	$JOBCTLM	JN
UNRERR1	$JOBCTLM	JN
UNRERR2	$JOBCTLM	JN
UNSW	$JOBCTLD	*

BANK1+1, bit 5: If ON, specifies a single unit is to be listed.

- Set ON: LISTIO statement processor when performing a LIOCUU function.
- Set OFF: INITL subroutine.
- Controls processing in the LISTIO statement processor.

UNTFND	$JOBCTLJ	GS
UPDATE	$JOBCTLM	JK

---

*Listing Only.

UPDHDN	$JOBCTLA	BJ
UPDOPT	$JOBCTLJ	GD
UPDSAV	$JOBCTLJ	GT

Subroutine: Moves data from input buffer to SYSLNK output area. Maintains a block count that is used for blocking records.

UPDTHDR	$JOBCTLM	JK

Subroutine: At completion of SDR and/or OBR processing, outputs an updated header record onto the recorder file.

UPDTMOVE	$JOBCTLM	JP
UPDTOBR	$JOBCTLM	JK
UPRITE	$JOBCTLM	JL

Subroutine: After each write, updates the disk address for the next write.

UPR1	$JOBCTLM	JL
UPSI	$JOBCTLJ	GK
UPSICH	$JOBCTLJ	GK
USERROR	$JOBCTLN	KA
USRLBL	$JOBCTLG	EN
USRLBL1	$JOBCTLG	EN
VMBUCK	$JOBCTLJ	GN
VMEND	$JOBCTLJ	GN
VMRTN	$JOBCTLJ	GN
VOL	$JOBCTLK	HA
WAITERR	$JOBCTLA	BL
WAITR	$JOBCTLJ	GM
WRITE	$JOBCTLM	JL

Subroutine: Issues all write commands to the recorder file.

WRTPHD	$JOBCTLG	EH

Subroutine: Used by the EXEC statement processor to write the phase directory.

WRTPHDR	$JOBCTLG	EH
XREF	$JOBCTLG	EM
XTENT	$JOBCTLK	HG
XTOP12	$JOBCTLK	HM

Subroutine: Checks, converts, and stores extent type and sequence number in label output area.

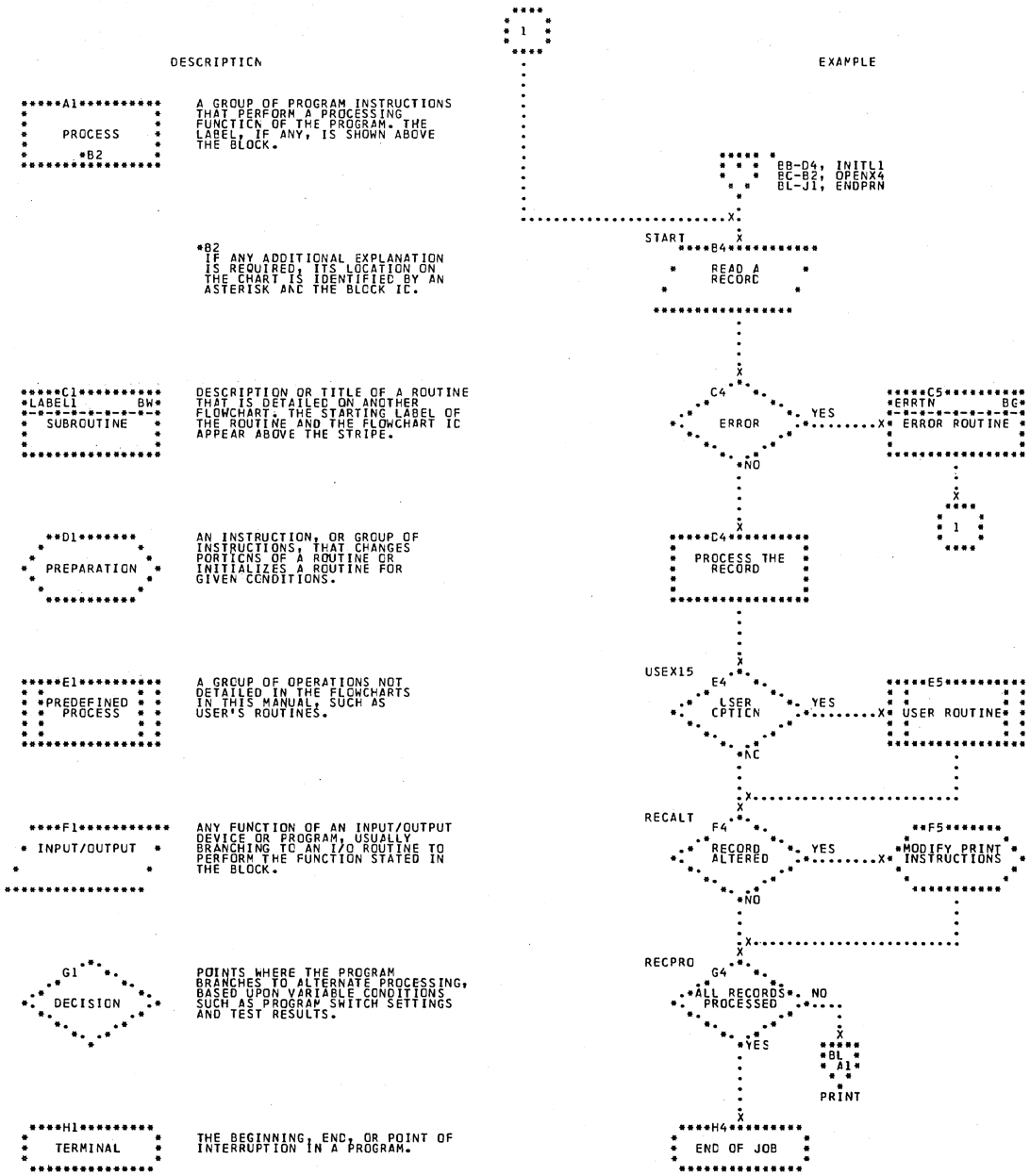
XTOP12A	$JOBCTLK	HM
XTOP12B	$JOBCTLK	HM
XTOP3	$JOBCTLK	HG
XTOP34	$JOBCTLK	HM

Subroutine: Checks, converts, and stores lower and upper extent limits in the label output area.

XTOP5	$JOBCTLK	HH
XTOUT	$JOBCTLK	HH
XTUNIT	$JOBCTLK	HH
YDISK	$JOBCTLD	CE
YESOPT	$JOBCTLM	JD
ZRMVDN	$JOBCTLA	BE
ZRMVLP	$JOBCTLA	BE

Message	Phase	Chart ID
0I00A	$$A$IPL2	AB
0I01A	$$A$IPL2	AD
0I10A	$IPLRT2	AK
0I11I	$IPLRT2	AQ
0I12I	$IPLRT3	AT
0I13I	$IPLRT3	AR
0I14I	$IPLRT3	AV
0I15I	$IPLRT3	AR
0I16A	$IPLRT4	AX
0I17A	$IPLRT4	AX
0I18A	$IPLRT2	AK
0I20I	$IPLRT4	AY
0I22I	$IPLRT4	AZC
0I23I	$IPLRT4	AZ
0I24A	$IPLRT3	AU
0I25I	$IPLRT4	AY
1A0ND	$JOBCTLD	CC,CD,CE, CG,CH,CJ
1A1ND	$JOBCTLD	CA,CB
1A2ND	$JOBCTLD	CA,CB,CH, CL,CP
1A20D	$JOBCTLG	FB
	$JOBCTLJ	GB,GE
1A3ND	$JOBCTLD	CG,CR
1A4ND	$JOBCTLD	CA,CL,CH,CW
	$JOBCTLF	DD,DF
1A40D	$JOBCTLJ	GS
	$JOBCTLK	HQ
1A5ND	$JOBCTLD	CP
	$JOBCTLF	DP
1A50D	$JOBCTLJ	GS
1A6ND	$JOBCTLD	CA
	$JOBCTLF	DM
1A7ND	$JOBCTLD	CA
	$JOBCTLF	DK
1A70D	$JOBCTLJ	GE
1A90D	$JOBCTLF	DL
1A9ND	$JOBCTLD	CE
1C00A	$JOBCTLA	BL
1C10A	$JOBCTLA	BL
	$JOBCTLG	FG
	$JOBCTLJ	GD
1C30A	$JOBCTLJ	GB,GM
1C3NI	$JOBCTLG	FG
1C70D	$JOBCTLA	BE
1C80D	$JOBCTLA	BL
1I00A	$JOBCTLA	BB
1I50I	$JOBCTLA	BL
1I70I	$JOBCTLA	BM
1I83A	$JOBCTLM	JB
1I84A	$JOBCTLM	JA,JJ
1I85A	$JOBCTLM	JD
1I86A	$JOBCTLM	JD,JF,JG,JM
1I87A	$JOBCTLM	JE
1I88I	$JOBCTLJ	GJ
1I89I	$JOBCTLM	JP,JQ
1I90D	$JOBCTLM	JP
1I91D	$JOBCTLM	JQ
1I92D	$JOBCTLM	JQ
1I93I	$JOBCTLM	JN
1L00D	$JOBCTLK	HF,HG,HJ, HK,HL,HM
1L10D	$JOBCTLG	FE
	$JOBCTLK	HG,HJ,HP
1P00D	$JOBCTLG	GQ
1S00A	$JOBCTLA through $JOBCTLM	See BM-B1
1S00D	$JOBCTLG	FG
	$JOBCTLM	JH
1IS03I	$JOBCTLJ	GN
1S10D	$JOBCTLG	EF,EN,FE
	$JOBCTLJ	GD,GN
	$JOBCTLK	HA,HC,HF,HG, HJ,HK,HP

APPENDIX C: EXPLANATION OF FLOWCHART SYMBOLS



1

```
// LISTIO SYS

              *** BACKGROUND ***

I/O UNIT   CMNT    CHNL    UNIT    MODE

 SYSRDR             0       0C
 SYSIPT             0       0C
 SYSPCH             0       0D
 SYSLST             1       0A
 SYSLOG             0       1F
 SYSLNK             1       91
 SYSRES             1       92
 SYSSLB            ** UA **
 SYSRLB            ** UA **
 SYSREC             1       91
```

2

```
// LISTIO PROG

              *** BACKGROUND ***

I/O UNIT   CMNT    CHNL    UNIT    MODE

 SYS000             0       91
 SYS001             0       91
 SYS002             0       91
 SYS003             0       91
 SYS004            ** UA **
 SYS005            ** UA **
```

3

```
// LISTIO F2

              *** FOREGROUND 2 ***

I/O UNIT   CMNT    CHNL    UNIT    MODE

 SYSRDR            ** UA **
 SYSIPT            ** UA **
 SYSPCH            ** UA **
 SYSLST            ** UA **
 SYSLOG            ** UA **
 SYSLNK            ** UA **
 SYSRES             1       92
 SYSSLB            ** UA **
 SYSRLB            ** UA **
 SYSREC             1       91

              *** FOREGROUND 2 ***

I/O UNIT   CMNT    CHNL    UNIT    MODE

 SYS000            ** UA **
 SYS001            ** UA **
 SYS002            ** UA **
 SYS003            ** UA **
 SYS004            ** UA **
 SYS005            ** UA **
 SYS006            ** UA **
 SYS007            ** UA **
 SYS008            ** UA **
 SYS009            ** UA **
 SYS010            ** UA **
 SYS011            ** UA **
 SYS012            ** UA **
 SYS013            ** UA **
 SYS014            ** UA **
 SYS015            ** UA **
```

Note: The 1st line of each sample shows the control statement as it was logged by job control.

1. List all system units.
2. List all background programmer units.
3. List all foreground 2 units.
4. List all units.
5. List a specific unit (SYSXXX).
6. List the logical units assigned to all physical devices.
7. List all unassigned units.
8. List all down units.
9. List all logical units assigned to a specified physical unit.

4

```
// LISTIO ALL

              *** BACKGROUND ***

I/O UNIT   CMNT    CHNL    UNIT    MODE

 SYSRDR             0       0C
 SYSIPT             0       0C
 SYSPCH             0       0D
 SYSLST             1       0A
 SYSLOG             0       1F
 SYSLNK             1       91
 SYSRES             1       92
 SYSSLB            ** UA **
 SYSRLB            ** UA **
 SYSREC             1       91

              *** BACKGROUND ***

I/O UNIT   CMNT    CHNL    UNIT    MODE

 SYS000            ** UA **
 SYS001             1       91
 SYS002             1       91
 SYS003             1       91
 SYS004            ** UA **
 SYS005            ** UA **
 SYS006            ** UA **
 SYS007            ** UA **
 SYS008            ** UA **
 SYS009            ** UA **
 SYS010            ** UA **
 SYS011            ** UA **
 SYS012            ** UA **
 SYS013            ** UA **
 SYS014            ** UA **
 SYS015            ** UA **

              *** FOREGROUND 2 ***

I/O UNIT   CMNT    CHNL    UNIT    MODE

 SYSRDR            ** UA **
 SYSIPT            ** UA **
 SYSPCH            ** UA **
 SYSLST            ** UA **
 SYSLOG            ** UA **
 SYSLNK            ** UA **
 SYSRES             1       92
 SYSSLB            ** UA **
 SYSRLB            ** UA **
 SYSREC             1       91

              *** FOREGROUND 2 ***

I/O UNIT   CMNT    CHNL    UNIT    MODE

 SYS000            ** UA **
 SYS001            ** UA **
 SYS002            ** UA **
 SYS003            ** UA **
 SYS004            ** UA **
 SYS005            ** UA **
 SYS006            ** UA **
 SYS007            ** UA **
 SYS008            ** UA **
 SYS009            ** UA **
 SYS010            ** UA **
 SYS011            ** UA **
 SYS012            ** UA **
 SYS013            ** UA **
 SYS014            ** UA **
 SYS015            ** UA **

              *** FOREGROUND 1 ***

I/O UNIT   CMNT    CHNL    UNIT    MODE

 SYSRDR            ** UA **
 SYSIPT            ** UA **
 SYSPCH            ** UA **
 SYSLST            ** UA **
 SYSLOG            ** UA **
 SYSLNK            ** UA **
 SYSRES             1       92
 SYSSLB            ** UA **
 SYSRLB            ** UA **
 SYSREC             1       91
```

```
              *** FOREGROUND 1 ***

I/O UNIT   CMNT    CHNL    UNIT    MODE

 SYS000            ** UA **
 SYS001            ** UA **
 SYS002            ** UA **
 SYS003            ** UA **
 SYS004            ** UA **
 SYS005            ** UA **
 SYS006            ** UA **
 SYS007            ** UA **
 SYS008            ** UA **
 SYS009            ** UA **
 SYS010            ** UA **
 SYS011            ** UA **
 SYS012            ** UA **
 SYS013            ** UA **
 SYS014            ** UA **
 SYS015            ** UA **
```

5

```
// LISTIO SYSRDR

              *** BACKGROUND ***

I/O UNIT   CMNT    CHNL    UNIT    MODE

 SYSRDR             0       0C
```

6

```
// LISTIO UNITS

CHNL    UNIT     OWNER    I/O UNIT    CMNT    MOD

  0      0C       BG       SYSRDR
  0      0C       BG       SYSIPT
  0      0D       BG       SYSPCH
  0      0E                * UA *
  0      1F       BG       SYSLOG
  0      1F       BG       SYSIN
  1      0A       BG       SYSLST
  1      90                * UA *
  1      91       BG       SYSLNK
  1      91       BG       SYSREC
  1      91       BG       SYS001
  1      91       BG       SYS002
  1      91       BG       SYS003
  1      91       F2       SYSREC
  1      91       F1       SYSREC
  1      92       BG       SYSRES
  1      92       F2       SYSRES
  1      92       F1       SYSRES
  1      80                * UA *
  1      81                * UA *
  1      82                * UA *
  1      83                * UA *
  1      84                * UA *
```

7

```
// LISTIO UA

        *** UNASSIGNED ***

          CHNL    UNIT

            1      80
            1      81
            1      82
            1      83
```

8

```
// LISTIO DOWN

            *** DOWN ***

            CHNL    UNIT

             ** NONE **
```

9

```
// LISTIO X@01F@

CHNL    UNIT     OWNER    I/O UNIT    CMNT    MODE

 0      1F       BG       SYSLOG
```

Figure 9. Sample LISTIO Printouts

APPENDIX E: COMREG AND CCB

COMREG*

Displacement hexadecimal	0	8	0A	0C	17	18	20	24	28	2C
Displacement decimal	0	8	10	12	23	24	32	36	40	44
	Date	Address of PPBEG	Address of EOSSP	Problem Program Use	UPSI Byte	Job Name	Highest Storage Address of the Partition	End Address of Last Phase Fetched or Loaded	Address of Uppermost Byte of Phase with Highest Ending Address	Label Area Length
	XXXXXXXX	XX	XX	XXXXXXXXXXX	X	XXXXXXXX	XXXX	XXXX	XXXX	XX

Displacement hexadecimal	2E	30	34	35	36	37	38	39	3A	3B	3C	3E
Displacement decimal	46	48	52	53	54	55	56	57	58	59	60	62
	PIK (PID)	End of Storage Address	Machine Confg. Byte	System Confg. Byte	Standard Language Translator I/O Options	Dump, Log and ASCII Options	Job Control Byte	Linkage Control Byte	Language Translator Control Byte	Job Duration Indicator Byte	Disk Address of Label Cylinder	Address of FOCL
	XX	XXXX	X	X	X	X	X	X	X	X	XX	XX

Job Control Switches

Displacement hexadecimal	40	42	44	46	48	4A	4C	4E	4F	58	5A	5C
Displacement decimal	64	66	68	70	72	74	76	78	79	88	90	92
	Address of PUB	Address of FAVP	Address of JIB	Address of TEB	Address of FICL	Address of NICL	Address of LUB	Line Count for SYSLST	System Date	LIOCS Comm. Bytes	Address of 1st Part of PIB Table	ID Number of Last Checkpoint
	XX	XX	XX	XX	XX	XX	XX	X	XXXXXXXXX	XX	XX	XX

Displacement hexadecimal	5E	60	62	64	66	68	6A	6C	6E
Displacement decimal	94	96	98	100	102	104	106	108	110
	Length of LUB ID Queue = No. of Channel Queue Entries	Address of Disk Information Block (DIB)	Address of Error Recovery Block	Address of PC Option Table less 8 bytes	Address of IT Option Table less 8 bytes	Address of OC Option Table less 8 bytes	Key of Program with Timer Support	Address of the LUBID Queue	Logical Transient Key
	XX	XX	XX	XX	XX	XX	XX	XX	XX

Displacement hexadecimal	70	7C	7E	80	84	86	87	88
Displacement decimal	112	124	126	128	132	134	135	136
	Supervisor Constants	Address of 2nd Part of PIB Table	Address of MICR DTF Table (PDTABB)	Address of QTAM Vector Table	Address of BG Comm. Region	Option Indicator	System Configuration Byte 2	Pointer to Comm. Region Extension
	XXXXXXXXXXX	XX	XX	XXXX	XX	X	X	XXXX

* The address of the communications region is in fixed location X'14' - X'17'.

Displacement values illustrated can be used to access the listing and/or the key that follows the figure. The key offers more detailed information about each area when necessary.

Figure 10. Supervisor Communications Region (Part 1 of 5)

Key to Communications Region Displacements:

0 MM/DD/YY or DD/MM/YY obtained from the job control date statement. Format controlled by COMREG + 53 (System Configuration Byte, date convention bit 0).

8 Address of the problem program area.

10 Address of the beginning of the problem program area. Y (EOSSP)=Y (PPBEG) if the storage protection option has not been selected. Y (EOSSP) equals the first main storage location with a storage protection key of 1, if storage protection is supported.

12 User area. If seek separation option is specified, bytes 12 and 13 are used at IPL time for the address of the seek address block.

23 User program switch indicator.

24 Job name set by the job control program from information found in the job statement.

32 Address of the uppermost byte of the problem program area as determined by the IPL program (Clear storage routine determines the address, ENDRD routine of $$A$IPL2 stores it.), or the address of the uppermost byte of the partition as determined during processing of the ALLOC statement.

36 Address of the uppermost byte of the last phase of the problem program fetched or loaded. The initial value (as shown) is overlaid by the first fetch or load to the problem program area.

40 Highest ending main-storage address of the phase among all the phases having the same first four characters as the operand on the EXEC statement. For the background partition only, job control builds a phase directory of these phases. The address value may be incorrect if the program loads any of these phases above its link-edited origin address. If the EXEC statement has no operand, job control places in this location the ending address of the program just link-edited.

44 Length of the problem program label area.

46 Program Interrupt Key - PIK (if asynchronous processing is not supported): Value is equal to the displacement from the start of the PIB table to the PIB for the task.

OR

Partition Identifier - PID (if asynchronous processing is supported): Value is hex 10, 20, or 30 to identify the partition in which a maintask or a subtask is running. (See the communications region extension, displacement 18, for the PIK in an asynchronous processing supervisor.)

First byte - always zero.
Second byte - contains the key of the program that was last enabled for interrupts, or the partition identifier in an AP supervisor.

Task	PIK (PID) Value
*All Bound	X'00'
BG	X'10'
*F2	X'20'
*F1	X'30'
Attn Rtn	X'40'
Quiesce I/O	X'50'
Supervisor	X'60'

*These tasks do not exist in a non-MPS supervisor.

48 Logical end of main storage address.

Figure 10. Supervisor Communications Region (Part 2 of 5)

Key to Communications Region Displacements:

52 Machine Configuration Byte (Values set at supervisor generation time.)

Bit 0: 1 = Storage protect feature
0 = No storage protect feature
1: 1 = Decimal feature
0 = No decimal feature
2: 1 = Floating-point feature
0 = No floating-point feature
3: 1 = Physical transient overlap option
0 = No physical transient overlap option
4: 1 = Timer feature
0 = No timer feature
5: 1 = Channel switching device
0 = No channel switching device
6: 1 = Burst mode on multiplex channel support
0 = No burst mode on multiplex channel support
7: Reserved

53 System Configuration Byte

Bit 0: 1 = DDMMYYJJ
0 = MMDDYYJJ (Date convention bit set at generation time by STDJC)
1: 1 = Multiprogramming environment
0 = Batch job environment
2: 1 = DASD file-protect supported
0 = No file-protect support for DASD
3: 1 = DASD SYSIN - SYSOUT
0 = No DASD SYSIN - SYSOUT
4: 1 = Teleprocessing
0 = No teleprocessing
5: 1 = Batch job in foreground
0 = No BJF
6: 1 = Asynchronous processing
0 = No AP
7: 1 = Track Hold
0 = No Track Hold

54 This byte contains the standard language translator I/O options (set by the STDJC macro).

Bit 0:	DECK option	1 = yes, output object modules on SYSPCH
1:	LIST option	1 = yes, output source module listings and diagnostics on SYSLST
2:	LISTX option	1 = yes, output hexadecimal object module listings on SYSLST (compilers only)
3:	SYM option	1 = yes, output symbol tables on SYSLST/SYSPCH
4:	XREF option	1 = yes, output symbolic cross reference list on SYSLST
5:	ERRS option	1 = yes, output diagnostics on SYSLST (compilers only)
6:	CHARSET option	1 = 48, input on SYSIPT is 48 or 60 character set
7:	Reserved	

55 This byte contains the standard supervisor options for abnormal EOJ and control statement display, and the indicator for the presence of the ASCII-EBCDIC and EBCDIC-ASCII translation tables.

Bit 0:	Always on	
1:	DUMP option	1 = yes, dump registers and storage on SYSLST
2:	Reserved	
3:	LOG option	1 = yes, list all control statements on SYSLST
4-6:	Reserved	
7:	ASCII option	1 = yes, ASCII supported

Figure 10. Supervisor Communications Region (Part 3 of 5)

Key to Communications Region Displacement:

56 Job control byte

Bit 0: 1 = Job Accounting Interface (JA) not supported
0 = Job Accounting Interface (JA) is supported
1: 1 = Return to caller on LIOCS disk open failure
0 = Do not return to caller on LIOCS disk open failure
2: 1 = Job control input from SYSRDR
0 = Job control input from SYSLOG
3: 1 = Job control output on SYSLOG
0 = Job control output not on SYSLOG
4: 1 = Cancel job
0 = Do not cancel job
5: 1 = Pause at end-of-job step
0 = No pause at end-of-job step
6: 1 = SYSLOG is not a 1052
0 = SYSLOG is a 1052
7: 1 = SYSLOG is assigned to the same device as SYSLST
0 = SYSLOG is not assigned to the same device as SYSLST

57 Linkage control byte

Bit 0: 1 = SYSLNK open for output
0 = SYSLNK not open for output
1: 1 = $ or FG program phase deleted, renamed, or cataloged (flag bit for $MAINEOJ)
2: 1 = Allow EXEC
0 = Suppress EXEC
3: 1 = Catalog linkage editor output
0 = Do not catalog linkage editor output
4: 1 = Supervisor has been updated
0 = Supervisor has not been updated
5: 1 = Executing in AUTOTEST mode
0 = Not executing in AUTOTEST mode
6: 1 = Reallocate or condense in progress
7: 1 = Fetch $MAINEOJ at end of job to update system directory
0 = Do not fetch $MAINEOJ at end of job for update

58 Language processor control byte. This is a set of switches used to specify nonstandard language translator options. The switches within the byte are controlled by job control OPTION statements and when set to 1, override standard options. The format of this byte is identical to the stnadard option byte (displacement 54) with one exception: Bit 7 in this byte is used to indicate to LIOCS that the rewind and unload option has been specified.

59 Job duration indicator byte

Bit 0: 1 = Within a job condition
0 = Outside a job condition
1: 1 = Dump on an abnormal end-of-job condition
0 = No dump on abnormal EOJ
2: 1 = Pause at EOJ step
0 = No pause at EOJ
} Set by Attention Routine for Job Control
3: 1 = Job control output on SYSLST
0 = Output not on SYSLST
4: 1 = Job is being run out of sequence with a temporary assignment for SYSRDR
0 = Conditions for 1 setting not met
5: 1 = PCIL is being condensed
0 = PCIL is not being condensed
6: Reserved
7: 1 = Batch command just issued
0 = Condition for 1 setting did not occur

Figure 10. Supervisor Communications Region (Part 4 of 5)

Key to Communications Region Displacements:

60 Binary disk address of the volume label area (label cylinder).

62 → 76 As illustrated (Figures for information blocks, I/O tables, and pointers begin at Figure 21 which refers to more detailed Figures).

78 Set to the value nn specified in the LINES = nn parameter of the STDJC macro.

79 The format of the system date contained within this field is determined by the IPL program from information supplied in the date convention byte (displacement 53). Bytes 85-87 contain the day count.

88 Bytes reserved for use by LIOCS. Transient dump programs insert a key to indicate to the LIOCS end-of-volume routine, SSBCMT07, that it was called by a B-transient.

90 Address of the first part of the program information block (PIB) table. (See Figures 18 and 19).

92 ID number of the last checkpoint. Temporary indicator of file protected DASD. Used at IPL time, when DASDFP is specified.

94 Length of the LUBID queue (in bytes). This equals the number of channel queue entries. It can also be used to access the REQID, LUBDSP, and TKREQID queues: (See Figure 17 - GY24-5151).

96 Address of disk I/O position data. This is the starting address of the disk information block (DIB) table (See Figure 17).

98 Address of the beginning of the error recovery block. The error recovery block contains addresses of error recovery exits, error recovery queue information that can be used by physical transients routines, and defines storage for the error queue entries (See Figure 43 - GY24-5151).

100 → 104 Option Tables. (See Figure 13 - GY24-5151).

106 Key of the program (BG, F2, or F1) that has timer support.

108 Address of LUBID queue. (See Figure 17 - GY24-5151).

110 Logical Transient Key (LTK) contains the same value as the PIK (PID) (Displacement 46) when the logical transient is requested. When the transient area is not in use, LTK is equal to zero. The SVC 2 routine sets the LTK. The SVC 11 routine resets the LTK.

112 Supervisor constants:

DOLLARBO	(4 bytes) = C'SSBO'
SSKADR	(5 bytes) = XL5'0'
LTAREA	(3 bytes) = Adcon of LTSVPT, logical transient save pointer

124 Address of second part of program information block (PIB) table (See Figure 20).

126 Address of PDTABB, table of DTF addresses for MICR support.

128 Address of QTAM vector table (IJLQTTAD).

132 Address of background communications region.

134 Option Indicator Byte

Bit 0: 1 = MCRR indicated for OBR writer
0 = No MCRR indicated for OBR writer
1: 1 = EU interface active
0 = EU interface not active
2: 1 = TP request
0 = No TP request
3: 1 = Supervisor support for only 9-track tape
0 = Supervisor does not support 9-track tape exclusively
4: Reserved
5: 1 = RETAIN/370 support generated
0 = RETAIN/370 support not generated
6-7: Reserved

135 System Configuration Byte 2

Bit 0: 1 = PCIL supported
0 = PCIL not supported
1-7: Reserved

136 Pointer to communications region extension (See Figure 11).

Figure 10. Supervisor Communications Region (Part 5 of 5)

BGXTNSN (See Note)

0 (Hexadecimal Displacement)	4	8	0C	10	12	14	18	1C	20
0 (Decimal Displacement)	4	8	12	16	18	20	24	28	32
CE Table Address	Track Hold Table Address (THTABAD)	Difference Between 1st and 2nd Part of PIB Table (PIBDIFF)	AB Termination Table Address -8 (ABPTR)	ID of Task Owning LTA (LID)	ID of Task Running (PIK)	Task Requester ID Table Address (TKIDPTR)	Address Used by QTAM (MVCFLD)	SDR Table Address (SDRTABLE)	TEBV Table Address (TEBVTAB)
XXXX	XXXX	XXXX	XXXX	XX	XX	XXXX	XXXX	XXXX	XXXX

24 (Hexadecimal Displacement)	28	2C	30	34	38	3C
36 (Decimal Displacement)	40	44	48	52	56	60
OLTEP Linkage Address	RMS Linkage Address (RASLINK)	ASCII-EBCDIC Translation Table Address	(Reserved)	JAI Common Table Address (ACCTCOMN)	JAI Partition Table Address (ACCTxx)	&SYSPARM Field Address
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX

Key to displacements:

0 CE Table Address.

4 Track Hold Table Address (THTABAD).

8 Difference between addresses of first part of PIB table and second part of PIB table (PIBDIFF).

12 Abnormal Termination Table Address (minus 8) (ABPTR).

16 Identification (LID) of the task owning the Logical Transient Area. Contains same value as PIK (displacement 18) when LTA is in use. Contains zero when LTA is not in use.

18 Program Interrupt Key (PIK) if asynchronous processing is supported. Value is equal to the displacement of the start of the PIB table to the PIB of the main task or subtask being selected (running).
First byte - zero
Second byte - contains the displacement into the PIB table for a maintask or a subtask.
Maintask - PIK value is hex 10, 20, or 30.
Subtask - PIK valve is hex 70, 80, 90, . . . F0.

20 Task Requester ID Table Address (TKIDPTR).

24 MVCFLD address used by QTAM.

28 Statistical Data Recorder Table Address (SDRTABLE).

32 Tape Error Blocks by Volume Table Address (TEBVTAB).

36 Pointer to OLTEP Linkage Addresses

40 RMS Linkage Area Address (RASLINK)

44 ASCII-EBCDIC Translation Table Address.

48 (Reserved)

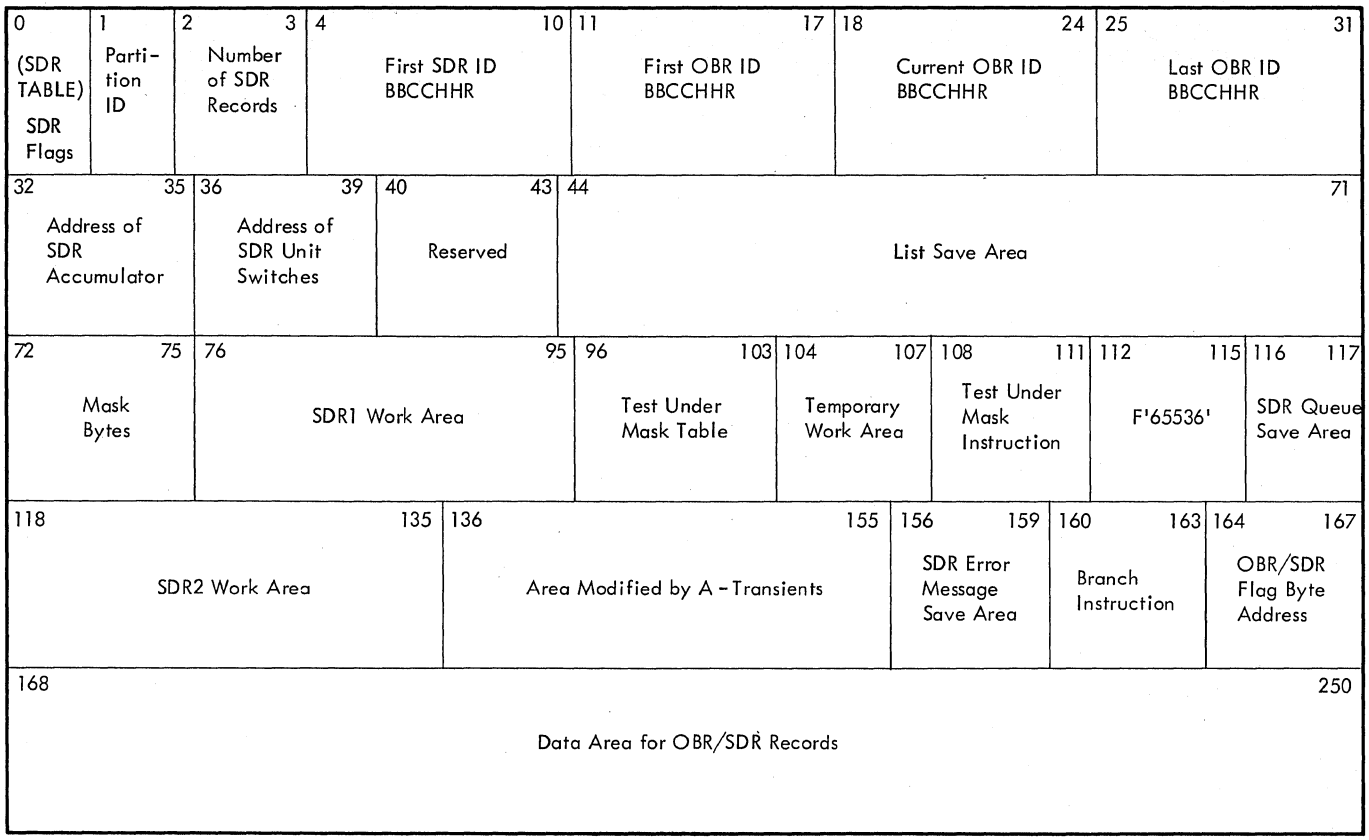
52 JAI Common Table Address (ACCTCOMN)

56 JAI Partition Table Address (ACCTxx; where xx = BG, F2, or F1).

60 Address of &SYSPARM Field.

Note: If communications regions are generated for the foreground partitions, the labels in those extensions will be F2XTNSN and F1XTNSN. The extensions, wherever used, are generated by the COMMNEX macro. Following the background extension (and immediately preceding the MCRR Linkage Table) is a six-byte area. The first four bytes are the address of the background save area (BGSAV), and the last two bytes are the value 4,096, used to restore base registers.

Figure 11. Background Communications Region Extension



Key to SDR Communications Region Displacements:

0 SDR Flags:

Bit 0: Key of OBR
1: RDE option
2: Initial IPL time
3: RF option = NO, recording is suppressed
4: RF option = CREATE
5: RF option = YES
6: Error while recording
7: Recorder file ready

Set and tested by Job Control.

1 Set by EREP transient $$BSDRUP to identify the partition making the call for EREP recording.

Settings: X'10' if EREP is running in BG.
X'20' if EREP is running in F2.
X'30' if EREP is running in F1.
X'01' with one of the above if recorder file is ready.
X'00' with one of the above if recorder file is not ready.

2 Initial number of SDR records specified. If SDR record count is not specified, the file is formatted for OBR records only ($JOBCTLM, see Chart 14)

4 Disk address of first SDR record.

11 Disk address of first OBR record.

18 Disk address of current OBR record.

25 Disk address of last OBR record.

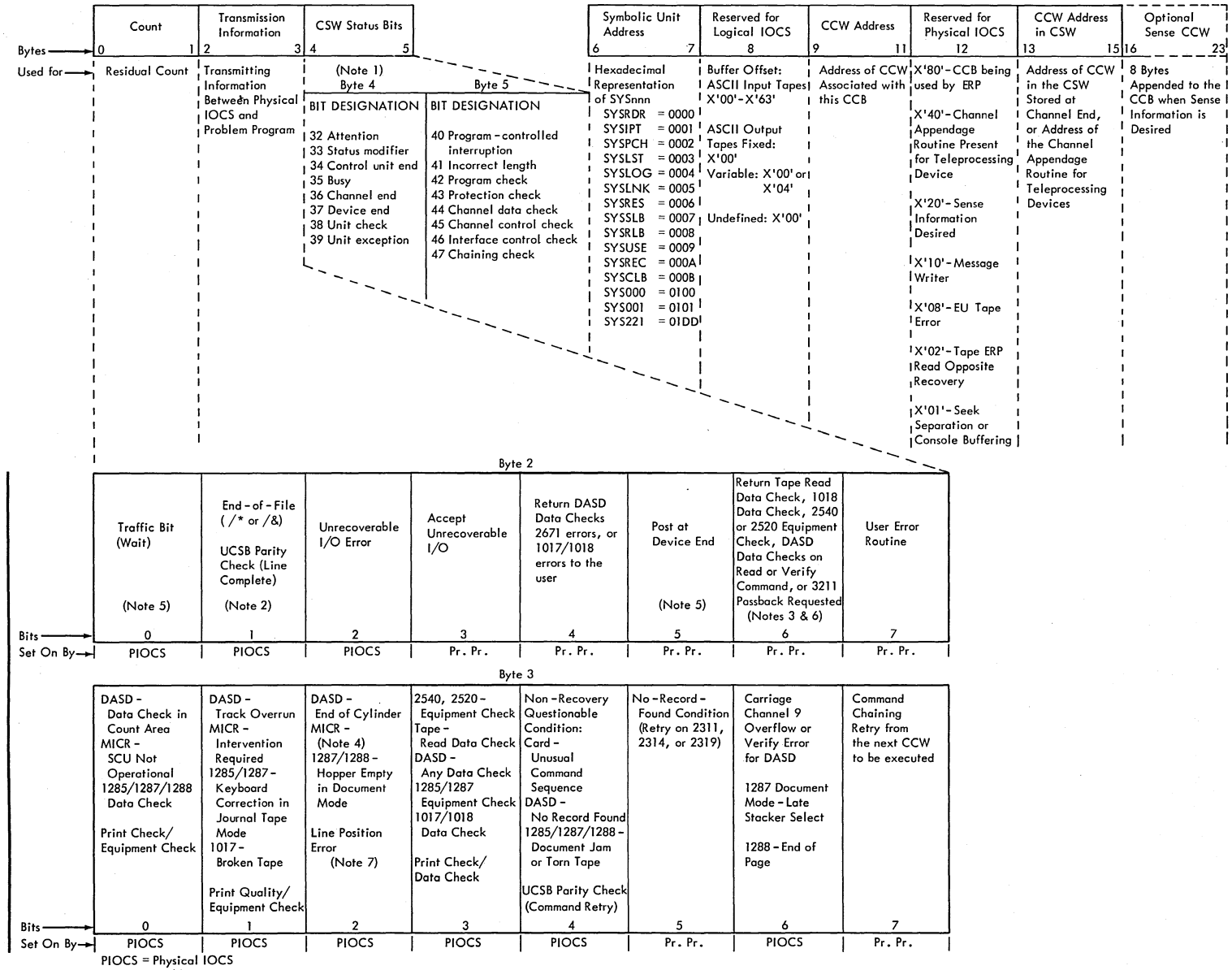
Figure 12. SDR Communications Region - SDRTABLE (Part 1 of 2)

Key to SDR Communications Region Displacements:

32	Address of SDR accumulator area which contains half-byte counters and accumulated error conditions.
36	Address of SDR unit switches. SDR switch byte (1 for each PUB): X'80' - Update operations complete X'40' - Counters on external file overflowed X'20' - I/O error during write X'08' - SDR update half-byte counters routine required X'04' - Update SDR record routine required Other - Reserved When entry contains X'01000000', indicates MCRR, no SDR supported.
40	Reserved.
44	SDR1 register save area.
72	Mask formats for interpretive error accumulator, SDR1: X'FF' - End of update X'FE' - Bypass counter X'FD' - Set up 'OR' condition to previous counter X'FC' - Ignore list item Other - Test bit in error queue
76	Used by the interpretive error accumulator routine to process list passed by OBR/SDR A-transient.
96	Used by the interpretive error accumulator routine.
104	Used by the interpretive error accumulator routine for address alignment.
108	Executed by the interpretive error accumulator routine.
112	Loop counter for the SDR counter update.
116	Save area for pointers to entries in the SDR error queue.
118	Work area where half byte error counters are unpacked and updated.
136	List of devices passed to the SDR processor and $$ANERAD.
156	Used by SDR/OBR recorder phases to pass error message displacements and disk error addresses in event of an error.
160	Entry point from OBR/SDR A-transients. Branches to label SDRMM.
164	Pointer into the OBR/SDR unit switches. Status posted by recorder phases. (See byte 36).
168	OBR and SDR records formatted by the recorder phases.

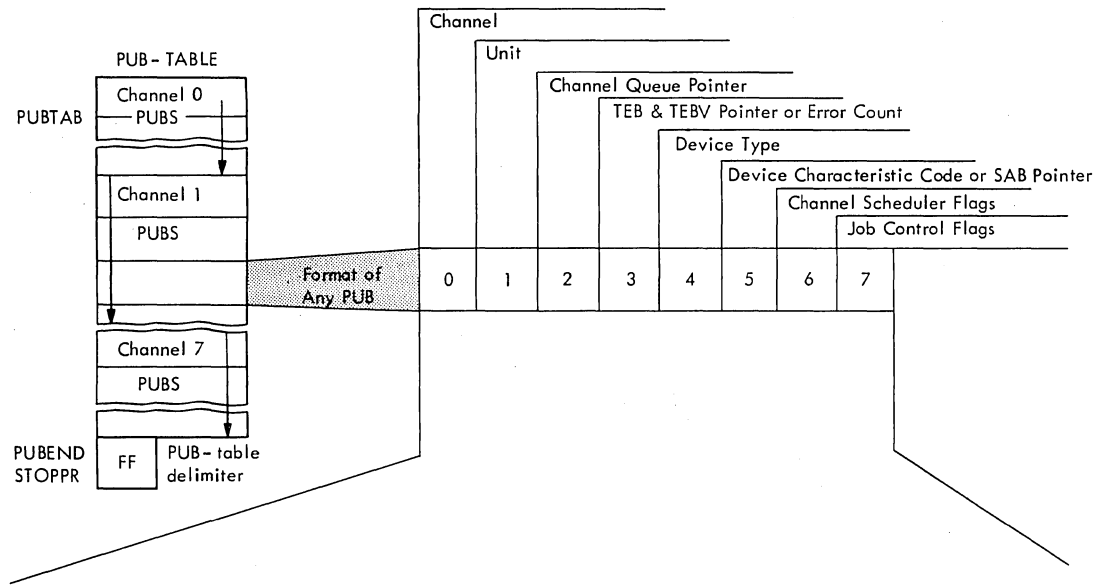
Figure 12. SDR Communications Region - SDRTABLE (Part 2 of 2)

Figure 13. Command Control Block (CCB)



PIOCS = Physical IOCS
Pr. Pr. = Problem Program

Note 1. Bytes 4 and 5 contain the status bytes of the Channel Status Word (Bits 32-47). If byte 2, bit 5 is on and device end results as a separate interrupt, device end will be ORed into CCB byte 4.
Note 2. Indicates /* or /& statement encountered on SYSRDR or SYSIPT. Byte 4, bit 7 (unit exception) is also on.
Note 3. DASD data checks on count not returned.
Note 4. For 1255/1259/1270/1275/1412/1419, disengage. For 1275/1419D, I/O Error in external interrupt routine (channel data check or busout check).
Note 5. The traffic bit (Byte 2, bit 0) is normally set on at channel end to signify that the I/O was completed. If byte 2, bit 5 has been set on, the traffic bit and bits 2 and 6 in byte 3 will be set on at device end. Also see Note 1.
Note 6. 1018 ERP does not support the Error Correction Function.
Note 7. This error occurs as an equipment check, data check, or FCB parity check.



BYTE 0 - Channel number. (Hex 0-7, FF=NULL)

BYTE 1 - I/O device unit number. (HEX 1F=1052, HEX 80=magnetic tape unit 0 . . .).

BYTE 2 - HEX 0, 1, 2, . . . points to the first channel queue entry for this device.

BYTE 3 - If device is a magnetic tape unit* and TEBs and/or TEBVs are specified, this byte is a TEB/TEBV pointer (HEX 1, 2, 3 . . .).

If device is a magnetic tape unit* but neither TEBs nor TEBVs are specified, this byte is an error counter.

If device is not a magnetic tape unit*, this byte is an error counter.

BYTE 4 - See Supervisor and Transients PLM.

BYTE 5 - SS of the MODE=parameter in the DVCGEN macro for a tape unit. See Supervisor and Transients PLM for magnetic tape density data.

For DASD with seek separation, this byte is used as the SAB pointer. With track hold but not seek separation supported, this byte contains a pointer to the Track Hold Table entry of X'FF' (with both SKSEP and TRKHLD specified, the track hold pointer is found in the SAB entry).
For MICR devices, this byte indicates the external interrupt line is in use.

BYTE 6 -
Bit 0: 1 = Device busy
1: 1 = Switchable device
2: 1 = EOJ for SYSRDR or SYSIPT
3: 1 = I/O error queued for recovery
4: 1 = Operator intervention required
5: 1 = Device end posting required
6: 1 = Burst device on MPX
7: 1 = 7-track tape unit

BYTE 7 -
Bit 0-4: standard MODE assignment for 7-track tape (all ones if not tape, all zeros if device is down).
5: device is assigned to a background job
6: device is assigned to a foreground 1 job
7: device is assigned to a foreground 2 job

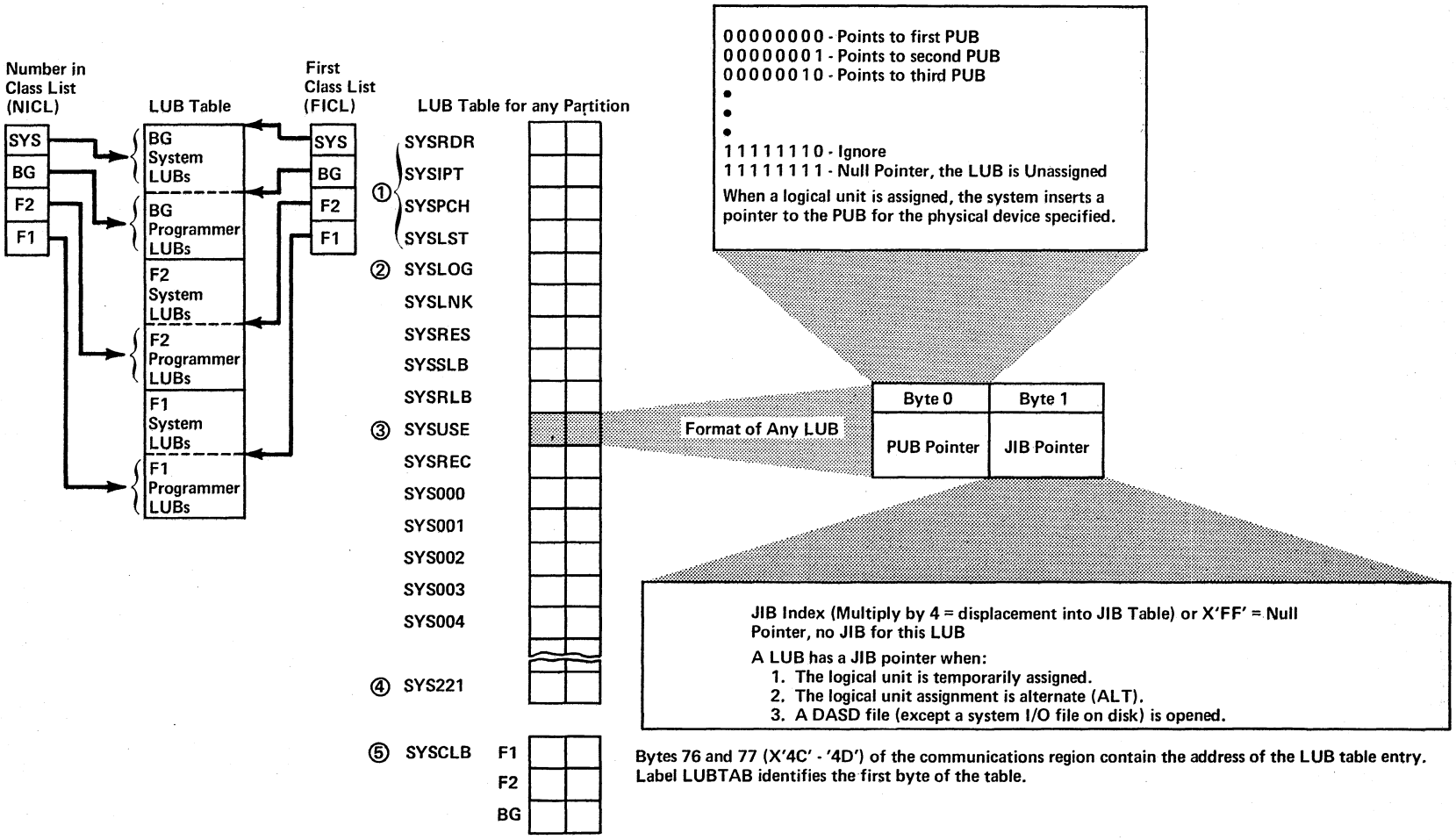
NOTE: A null PUB is generated for each device to be supported by the supervisor. Standard physical unit assignments are made to the PUB table at supervisor generation time. PUBs are ordered by channel and priority within a channel.

Bytes 64 and 65 (X'40'-'41') of the communications region contain the address of the PUB table entry. Label PUBTAB identifies the first byte of the table.

*2400 series or 3420 Magnetic Tape Units or 2495 Tape Cartridge Reader (TEBs); 2400 series or 3420 Magnetic Tape Units only (TEBVs)

Figure 14. PUB Table

Figure 15. NICL, FICL, and LUB Tables



Bytes 76 and 77 (X'4C' - '4D') of the communications region contain the address of the LUB table entry. Label LUBTAB identifies the first byte of the table.

① When in Single Program Initiation mode (Foreground 1 or 2): Must be unit record device and can be referenced by the program.

② When in Single Program Initiation mode (Foreground 1 or 2): Can be referenced by the program.

③ SYSUSE may be called SYSCTL in error recovery messages.

④ The maximum number of programmer logical units in the system is 222 if MPS = BJF, or 244 if MPS = YES or NO.

⑤ The SYSCLB (Private Core Image Library) LUB entry functions the same as other LUB entries, but is not part of the LUB Table. To locate the SYSCLB LUB in supervisor, perform the following steps:

1. Divide the PIK by 8.
2. Subtract the result in step 1 from the address of the PIB extension block.
3. If option AP = YES, the result of step 2 is the location of SYSCLB LUB. If option AP = NO, add 16 (for the all-bound PIBX) to the result of step 2.

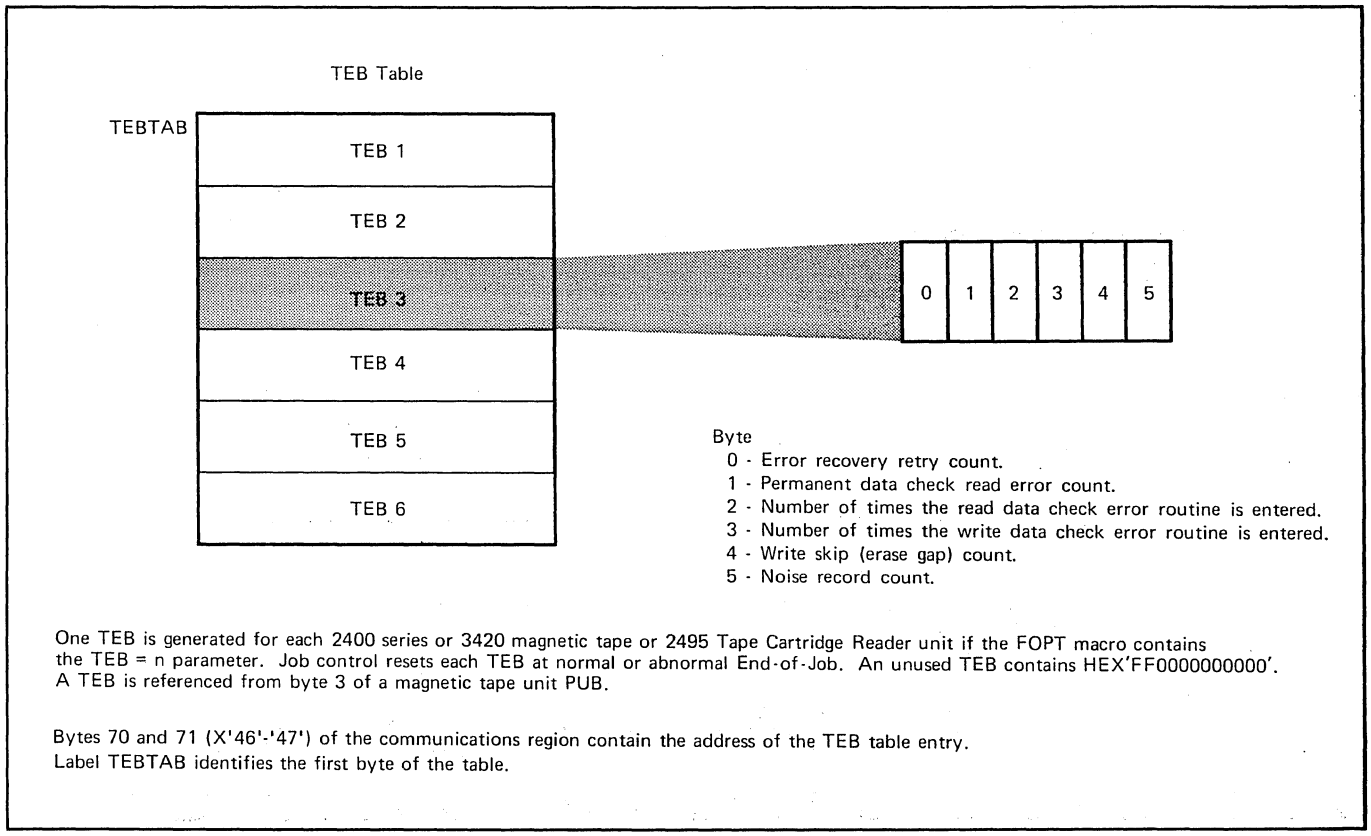
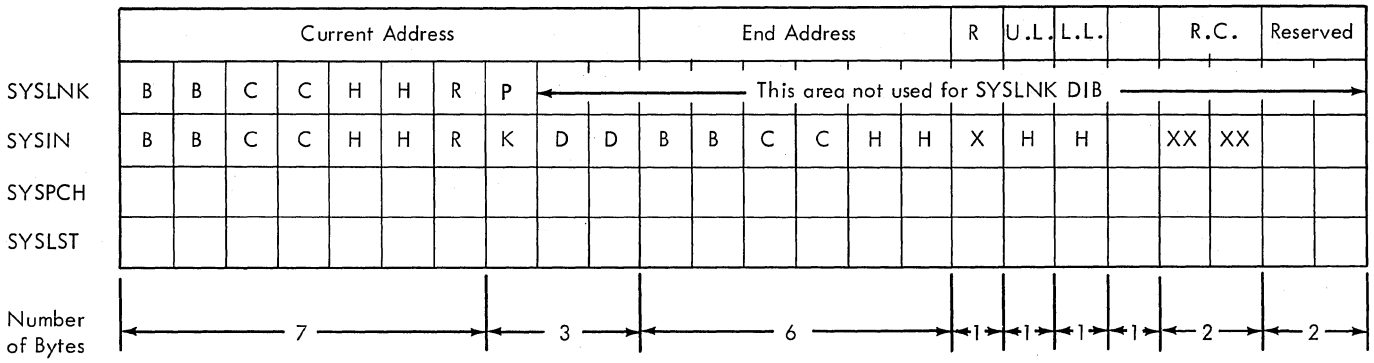


Figure 16. Tape Error Block (TEB)



KEY: Current Address: The next address to be used (for both input and output).
End Address : The last address within the limits of the extent.
R : Maximum number of records per track.
U.L. : Upper head limit.
L.L. : Lower head limit.
R.C. : Record Count - residual capacity for beginning of operator notification. This is set at system generation time with the SYSFIL parameter, or after IPL with the SET statement (RCLST and/or RCPCH operands). A warning message is issued by job control after end-of-job step when the minimum number of remaining records has been reached or exceeded during the previous job.
P : Starting cylinder of Private Core Image Library, if PCIL is assigned.
KDD : Key and data length for the symbolic device.

KDD for SYSIN = X'000050'
KDD for SYSPCH = X'000051'
KDD for SYSLST = X'000079'

Bytes 96 and 97 (X'60' - '61') of the communications region contain the address of the SYSLNK entry.
Label DSKPOS identifies the first byte of the table.

Figure 17. Disk Information Block (DIB) Table

PIB TABLE

Byte Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	= 16 Byte Length
All Bound PIB	Flag Byte See A *	Reserved	SP Prefix		Branch Instruction to the All Bound Routine							Reserved					
Problem Program PIB (Note 1)	Flag Byte See B *	Cancel Code (Fig.31)	SYSLOG ID (BG, F2, or F1)		NOP Instruct-ion (CR)	Address of the Partition Save Area			Number of Core Blocks (Note 2)	Address of the Origin of the Partition			PIB Assign Flag See D *	User LUB Index	Number of Program LUBs	Flag Byte See C *	
Attention PIB	Flag Byte See E *	Cancel Code (Fig.31)	SYSLOG ID (AR)		Branch Code (BC)	Active=Address of Save Area Inactive=Remainder of BC Instruction			Switch Byte See F *	Logical Transient Bucket (contains save area address)			X'07' See D *	Reserved	Address of the Logical Transient		
Quiesce PIB	Flag Byte See A *	Cancel Code (Fig.31)	C'/&'		Branch Instruction to Quiesce I/O Routine				Scratch Byte X'00'	X'00'	Channel PUB Table Index Values X'04'	X'08'	X'0C'	X'10'	X'14'	X'18'	
Supervisor PIB	Flag Byte See A *	Cancel Code (Fig.31)	SP Prefix		Branch Instruction to General Exit Routine				Address of SYSRES PUB		Length of Error Queue Entry		Constants to Clear Bytes 2-5 of CCB X'1F'	X'05'	X'00'	X'00'	
Subtask PIB for AP (Note 3)	Flag Byte See B *	Cancel Code (Fig.31)	SYSLOG ID (BG, F2, or F1)		NOP Instruc-tion	Address of the Save Area			Number of Core Blocks (Note 2)	Address of the Origin of the Main Task			PIB Assign Flag See D *	User LUB Index	Number of LUBs	Flag Byte See C *	

Note 1: Three problem program PIBs are built in this sequence when the MPS or BJF feature is selected as a generation option: { Background PIB, Foreground 2 PIB, Foreground 1 PIB }

When a batch-only environment is established at generation time, the All Bound and Foreground PIBs are excluded from the table, and only one (BG) problem program PIB is built. However, the X'20' bytes that F2 and F1 PIBs normally occupy (between PIBBG and PIBAR) are filled with 32 bytes of DIBs data.

Note 2: Number is in multiples of 2K for F2 and F1. BG is always 10K (X'0A').

Note 3: Total of nine subtask PIBs are generated, and only when AP is specified at generation time.

* See Figure 14 for flag byte expansions A, B, C, D, E and F.

Bytes 90 and 91 (X'5A'-'5B') of the communications region contain the address of the first part of the PIB Table. Label PIBTAB identifies the first byte of the table.

Figure 18. First Part of PIB Table

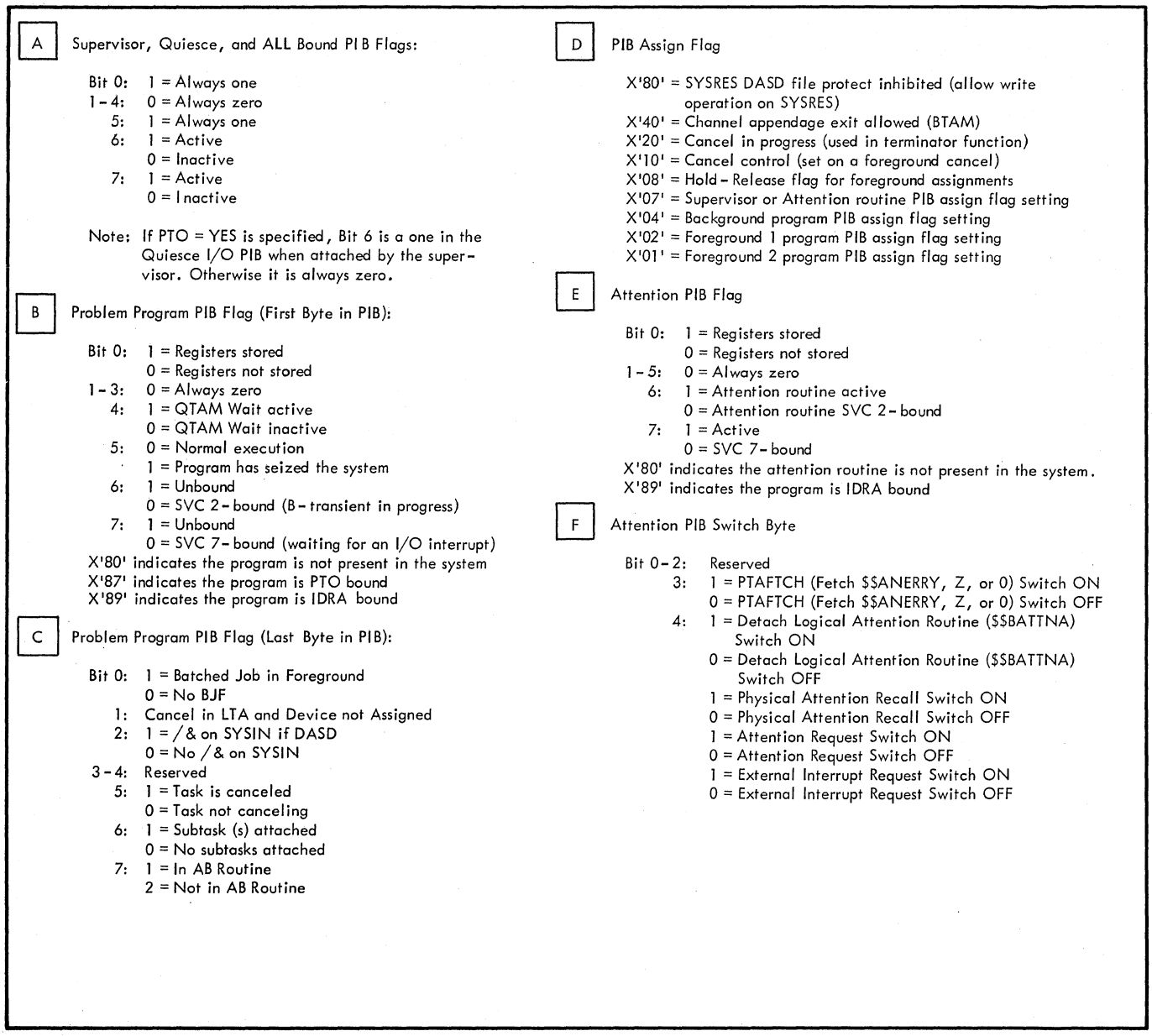


Figure 19. PIB Flag Expansions

Byte Number ➡	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	= 16 Byte Length
All Bound PIB	Reserved						H'16' Priority of All Bound PIB (Lowest)		Reserved				H'0' All Bound PIB Displacement		Reserved		
Background PIB	Address of BG Comm. Region		System LUB Index		Reserved		Priority of BG PIB ④		Address of Termination ECB, if any, or F'0'				X'0010' BG PIB Displacement		Reserved		
FG2 PIB ①	Adress of Area Comm. Region ②		System LUB Index		Reserved		Priority of F2 PIB ④		Address of Termination ECB, if any, or F'0'				X'0020' F2 PIB Displacement		Reserved		
FG1 PIB ①	Address of Area Comm. Region ②		System LUB Index		Reserved		Priority of F1 PIB ④		Address of Termination ECB, if any, or F'0'				X'0030' F1 PIB Displacement		Reserved		
Attention PIB	Address of BG Comm. Region		0	0	Reserved		H'3' Priority of Attention PIB		F'0'				X'0040' Attention PIB Displacement		Reserved		
Quiesce I/O PIB	Reserved						H'2' Priority of Quiesce I/O PIB		F'0'				X'0050' Quiesce PIB Displacement		Reserved		
Supervisor PIB	Reserved						H'1' Priority of Supervisor PIB (Highest)		F'0'				X'0060' Supervisor PIB Displacement		Reserved		
Subtask PIB ③	Address of Area Comm. Region		System LUB Index		Reserved		Priority of Subtask ④		ECB Address for Subtask, or F'0'				PIB Displacement of Maintask		Reserved		

① Generated only if MPS is specified.

② Always background communications region except when MPS=BJF.

③ Total of nine subtasks generated, and oniy when AP is specified.

④ Is filled in with halfword indicating the relative priority of task in the system (range H'4' to H'15', the lower the number the higher the priority).

Bytes 124 and 125 (X'7C' - '7D') of the communications region contain the address of the second part of the PIB table. Label PIB2AD identifies the first byte of the table.

Figure 20. Second Part of PIB Table

Decimal Displacement	Label	Byte Length	Description
			(TEBV Status Block portion of TEBV Table, see Note 1)
0	TEBLEN	1	Length of TEBV Error Block (for each Error Block generated)
1	TSBLEN	1	Length of TEBV Status Block (4, 6, or 22 bytes, see Note 1)
2	EVARTH	1	EVA Read Error Threshold
3	EVAWTH	1	EVA Write Error Threshold
...	.....	...	... ...
4	TEBSTAT	1	DASD ESTV File Status
5	TEBUDC	1	ESTVFLE Label Update Counter
...	.....	...	... ...
6	TEBDEV	1	Data Set Device Code
7	UPXTNT	4	Disk Address of Upper Extent of Data Set (cchh)
11	TEBRPT	1	Number of Records per Track
12	NXTESR	5	Disk Address of Next Available Space for Data Record (cchhr)
17	ESTVLABL	5	Pointer to ESTVFLE Label in VTOC (cchhr)
...	.....	...	... ...
			(TEBV Error Block portion of TEBV Table, see Note 2)
22	TEBV	1	Status Indicator (giving status of posting and writing error conditions)
23		1	Usage Indicator (X'00'=TEBV Error Block in use, X'FF'=Error Block generated but not serving any tape unit)
24		1	Retry Counter
25		1	Permanent Read Errors
26		1	Temporary Read Errors
27		1	Temporary Write Errors
28		1	Erase Gaps
29		1	Noise Blocks
30		1	Permanent Write Errors
31		1	Cleaner Actions
32		2	Number of Start I/Os
34		6	Volume Serial Number (volume ID)
...	.....	...	... ...
40	(repeat bytes 22-39 for each TEBV Error Block)		

Figure 21. TEBV Table Showing Status Block and Error Blocks (Part 1 of 2)

Note 1: The TEBV (Tape Error Block by Volume) Table is composed of one Status Block and (n) Error Blocks, and is addressed symbolically by label TEBVTAB.

Supervisor generation options in the FOPT macro determine the size of the TEBV Status Block at generation time:

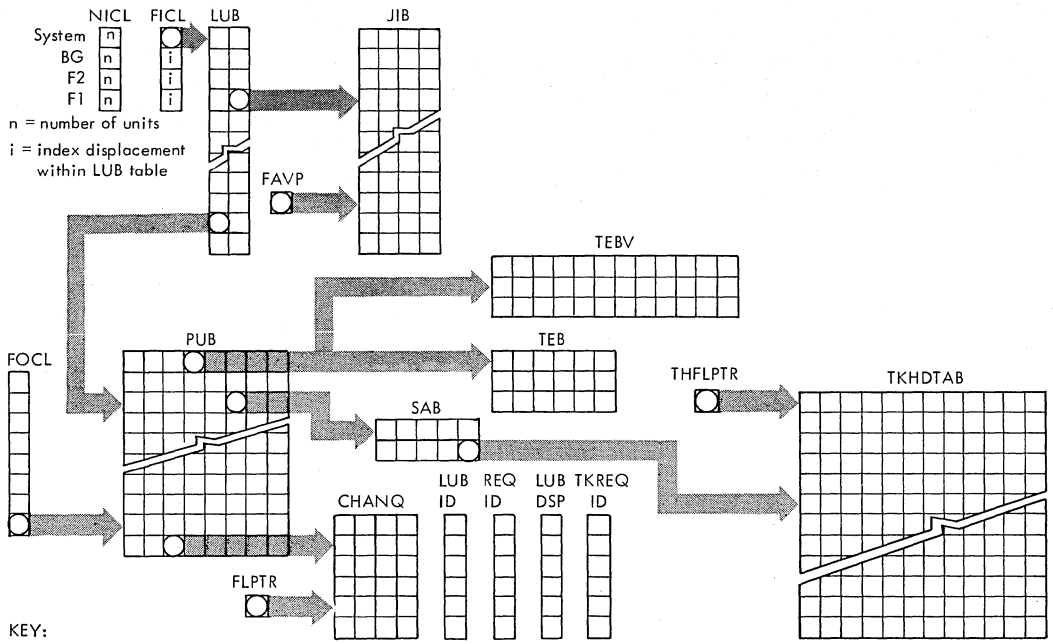
- When EVA is chosen without ESTV, the TEBV Status Block is four bytes long (bytes 0-3), followed by TEBV Error Blocks, so that bytes 4-21 are omitted.

- When ESTV output is to SYSLOG, the TEBV Status Block is six bytes long (bytes 0-5), followed by TEBV Error Blocks, so that bytes 6-21 are omitted.

- When ESTV output is to DASD, the TEBV Status Block is 22 bytes long (bytes 0-21, such as shown in this Figure), followed by TEBV Error Blocks.

Note 2: The number of TEBV Error Blocks generated corresponds to the (n) parameter of the FOPT macro for TEB, TEBV, or EVA options. A TEBV Error Block always contains 18 bytes, as shown in bytes 22-39 of this Figure. Therefore, the TEBV Table is composed of one TEBV Status Block (with its byte length dependent on supervisor generation options, as described in Note 1), followed by (n) number of 18-byte TEBV Error Blocks.

Figure 21. TEBV Table Showing Status Block and Error Blocks (Part 2 of 2)



KEY:

NICL (Number in Class) : The first byte contains the number of system class units. The second, third, and fourth bytes contain the number of programmer class units (BG, F2, F1) (Figure 21).

FICL (First in Class) : The first byte points to the first system class unit in the LUB table. (Always the first LUB table entry.) The second byte points to the first programmer class unit in the LUB table BG area. The third points to the first programmer class unit in the LUB table F2 area. The fourth points to the first programmer class unit in the LUB table F1 area (Figure 21).

LUB (Logical Unit Block) Table : The first byte points to a PUB table entry (if the logical unit is assigned) or contains X'FF'. The second byte points to a JIB table entry or contains X'FF' (Figure 21).

PUB (Physical Unit Block) Table : The first two bytes contain the channel and unit address of the physical device; the third a CHANQ pointer; the fourth a TEB pointer; the fifth device type codes; the sixth a device characteristic code or a SAB pointer; the seventh the channel scheduler flag; and the eighth has the job control flag (See Figure 19).

FOCL (First on Channel List) : The first byte points to the first PUB (highest priority) on channel zero. The next byte points to the first PUB (highest priority) on channel one, etc. A hexadecimal FF indicates the associated channel is not supported.

TEB (Tape Error Block by Unit) : One TEB is built for each tape unit at supervisor generation time if tape error statistics by unit are required (Figure 16).

TEBV (Tape Error Block by Volume) : One TEBV is built for each tape unit at supervisor generation time if tape error statistics by volume are required (Figure 21).

FAVP (First Available Pointer) : A one-byte pointer to the next available JIB entry.

JIB (Job Information Block) : The first two bytes contain extent or LUB information. The third contains ownership and JIB flags. The fourth contains JIB chaining information (See Supervisor PLM - GY24 - 5151).

CHANQ (Channel Queue) Table : The first byte contains the chain field (a pointer to the next in queue). The last three bytes contain the CCB address (Figure 18).

LUBID (LUB Identification) : A one-byte pointer to the LUB making the I/O request.

REQID (Requestor Identification) : A one-byte pointer to the program containing the CCB (Figure 18).

LUBDSP (LUB Displacement) : A one-byte value equal to the absolute LUB number (CCB byte 7).

FLPTR (Free List Pointer) : A one-byte pointer to the next free entry in the channel queue (Figure 18).

SAB (Seek Address Block) : A four-byte (BCCH) address that is the current disk address of the device plus a fifth byte that contains a Track Hold Table pointer of X'FF'. If the Track Hold function is not supported, the fifth byte contains X'00'.

TKHDTAB (Track Hold Table) : The first byte contains a pointer to the next available entry (or X'FF'); bytes 2 - 4 have CCB address of the requesting task; bytes 5 - 10 have a disk address (BBCCHH) of track being held; byte 11 has key of owning track; and byte 12 has two uses: bit 0=1 means a task is waiting for the track, and bits 4 - 7 count the number of holds on the track. (See Supervisor PLM - GY24-5151). Note: The number of holds is one more than the value of bits 4 - 7 of the last byte.

THFLPTR (Track Hold Free List Pointer) : A one-byte pointer to the next free entry in the Track Hold Table.

TKREQID (Track Requestor Identification) : A one-byte pointer to the PIB of the task requesting I/O.

Figure 22. I/O Table Interrelationship

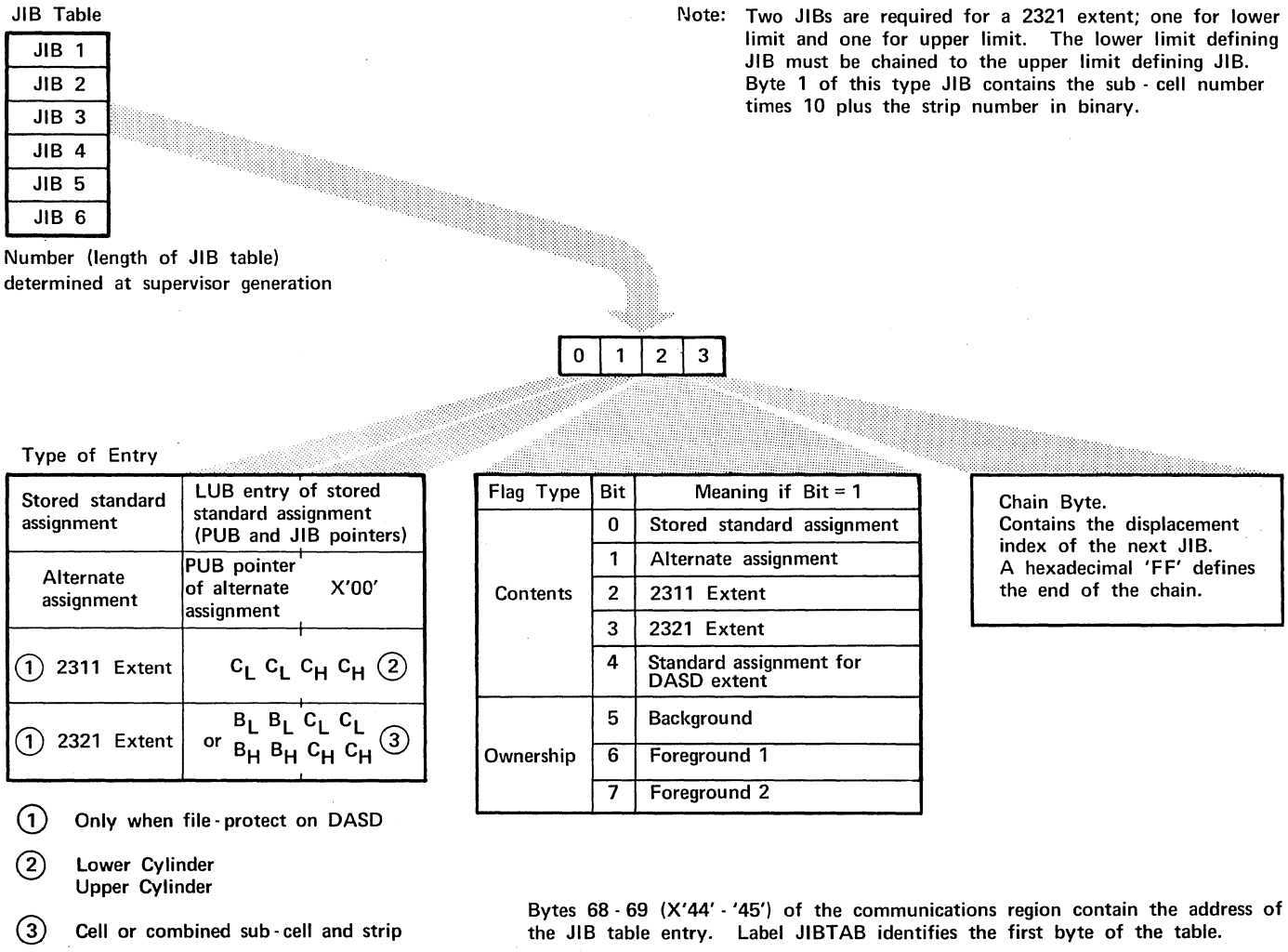


Figure 23. Job Information Block (JIB) Table

Displacement	Label	Description
0-3	ACCTWK1* (ACCTABLE)*	Work area used in SIO update.
4-7	ACCTWK2	Work area used with ACCTWK1 in start/stop time routine.
8-11	ACCTSVPT	Job card pointer; address of job card field following jobname.
12	ACCTPART	ID of partition in charge (partition switch name).
13	ACCTRES2	Reserved.
14-15	ACCTLEN	Length of SIO area = 6n + 1, where n = number of devices for this partition in SYSGEN option JA = (n1, n2, n3).
16-21	ACCTLOAD	Label area instruction; moves JAI label area address to OPEN/CLOSE transients.
22-23	ACCTRES3	Reserved.
24-27	ACCTLADD	Address of alternate label area.
28-31	ACCTCPUT	Counter for CPU time elapsed in a jobstep, counted in 300ths of a second.
32-35	ACCTOVHT	Counter for overhead time; time not charged to any partition.
36-39	ACCTBNDT	Counter for all-bound time; system wait state time divided between running partitions.
40-47	ACCTSVJN	Save area for job name during simulated EOJ.
		JOB ACCOUNTING TABLE (user's portion of Partition Table)
48-55	ACCTJBNM	Job name; taken from job card.
56-71	ACCTUSRS	User information; 16 bytes from Job card.
72-73	ACCTPTID	Partition ID; 'BG', 'F2', or 'F1' in EBCDIC format.
74	ACCTCNCL	Cancel code; see Cancel Codes and Messages (Figure 32).
75	ACCTYPER	Type of record: 'S' = job step, 'L' = last step of job.
76-83	ACCTDATE	Date in format specified at SYSGEN (MM/DD/YY or DD/MM/YY).
84-87	ACCTSTRT	Start time of job, in packed decimal (OHHMMSSF; F = sign).
88-91	ACCTSTOP	Stop time of job, in same format as ACCTSTRT.
92-95	ACCTRES	Reserved.
96-103	ACCTEXEC	Phase name; taken from execute card.
104-107	ACCTHICR	High core address of active program phase, from COMREG.
108-111	ACCTIMES	CPU time elapsed in a job step; counted in 300ths of a second.
112-115	------	Overhead time; elapsed time not charged to any partition, in 300ths of a second.
116-119	------	All-bound time; system wait state time divided between running partitions, in 300ths of a second.
120	ACCTSIOS	SIO tables: 6 bytes for each device specified by SYSGEN options, as follows: 2 bytes for device address (Ocuu), 4 bytes for count of SIOs in current jobstep.
———	------	Overflow byte: normally X'20', but is X'30' if more devices are used within a partition than specified by SYSGEN options.

*Note: DSECT ACCTABLE symbolically addresses the JAI Partition Tables with labels as shown. Each partition in which JAI is supported has its own JAI Partition Table, labeled ACCTBG, ACCTF2, ACCTF1, for active partitions BG, F2, and F1 respectively.

Figure 24. Job Accounting Interface Partition Table

Displacement	Label	Description
0-15	(ACCTCOMN) ACCTSVRG	Temporary register save area.
16-17	ACCTSVRX	Save area for remainder of overhead counter times distributed by partition on exit.
18-19	ACCTSVRE	Save area for remainder of all-bound counter times distributed by partition on entry.
20-23	ACCTPCNT	Count of partitions using JAI.
24	ACCTSAID	Owner of physical transient area*.
25	ACCTFAID	Interrupted program*.
26	ACCTRAID	Active program*.
27	ACCTSWCH	Accounting switches: if bit = 1, true; if bit = 0, not true. bit 0 - cancel accounting bit 1 - no active partitions bit 2 - catalog in process bit 3 - alternate label area bit 4 - IPL indicator bit 5 - $JOBACCT in F1 bit 6 - $JOBACCT in F2 bit 7 - $JOBACCT in BG
28-31	ACCTIME	Start time of current accounting interval, in complement format.
32-33	ACCTRESC	Reserved.
34-35	ACCTUSEP	Address of user save area (ACCTUSER).
36-39	ACCTBLES	Address of BG Job Accounting Table.
40-43	- - - - -	Address of F2 Job Accounting Table if BJF; otherwise zero.
44-47	- - - - -	Address of F1 Job Accounting Table if BJF; otherwise zero.
48-53	ACCTSEAS	Seize blocks; serve as overlapped Event Control Blocks. TS Bit | (reserved) | Wait Bit 1 | PIK 1 | Wait Bit 2 | PIK 2 1st ECB 2nd ECB TS Bit: X'00' = no $JOBACCT running X'FF' = $JOBACCT active
54-55	ACCTUSEL	Length of user save area, set with 4th operand of global AG39.

*Note: X'00' = all bound, X'10' = BG, X'20' = F2, X'30' = F1, X'40' = overhead and FG if SPI.

Figure 25. Job Accounting Interface Common Table (ACCTCOMN)

Figure 26. RMS Machine Check Record on SYSREC

Record 1	0 Machine Check ID (X'10')	1-2 Record Sequence Counter	3 CPU Model Number	4 Reserved	5-7 CPU Serial	8-9 CPU ID	10-11 MCEL Length	12-15 Date	16-19 Time of day in 300ths of a second Timer units	20-27 Reserved	28-35 Job ID	36-43 Machine Check Old PSW	44-77 Reserved	78 Damage Assessment	79 End of Record Indicator (X'FF')
Record 2	0-3 (Same as Record 1)			4 Sub Class Codes	5 Tense Codes	6-7 Error and Validity Codes	8-11 Extended Logout Length	12-19 Reserved	20-23 Failing Storage Address	24-25 Region Code ECC Information	26-27 Region Code Control Word Address	28-78 Reserved			79 (Same as Record 1)
Record 3	0-3 (Same as Record 1)			4-48 Reserved									49-78 Floating Point Register Area		79 (Same as Record 1)
Record 4	0-3 (Same as Record 1)			4-5 Floating Point Regs Save Area (Continued)		6-69 General Registers Save Area								70-78 Control Registers Save Area	79 (Same as Record 1)
Record 5	0-3 (Same as Record 1)			4-58 Control Registers Save Area							59-78 MCEL (Machine Check Extended Logout)				79 (Same as Record 1)
Record 6-17	0-3 (Same as Record 1)			4-78 MCEL (Machine Check Extended Logout)											79 (Same as Record 1)
Record 18	0-3 (Same as Record 1)			4-75 MCEL (Machine Check Extended Logout)										76-78 Reserved (X'000000')	79 (Same as Record 1)

*Note: The Model 155 uses Records 1-18, as shown. The Model 145 uses only Records 1-8. In Record 8 for the Model 145, Bytes 0-3 are the same as Record 1, Bytes 4-25 contain the remainder of the MCEL (Machine Check Extended Logout) area, and the remainder of the record is not used.

0	4	8	12	16	20	24	28	32
LD00SLOT (S$RAST00) (1) (2)	LD01SLOT (S$RAST01) (3)	LD02SLOT (S$RAST02) (5)	LD03SLOT (S$RAST03) (5)	LD04SLOT (S$RAST04) (5)	LD05SLOT (S$RAST05) (5)	LD06SLOT (S$RAST06) (5)	LD07SLOT (S$RAST07) (5)	LD08SLOT (S$RAST08) (5)
36	40	44	48	50	52	54	56	57
LD09SLOT (S$RAST09) (5)	LD10SLOT (S$RAST10) (4)	LD11SLOT (S$RAST11) (5)	RASCCB Residual Count	RASTIB Transmission information	---- CCW Status bytes	---- SYSRES LUB	RASCCBF RAS CCB indicator	---- RAS Fetch CCWs address
60	64	72	80	88	96	103	104	106
---- CCW stored address	RASCCWS RAS seek CCW	RASRCG RAS search CCW	RASTIC TIC CCW	RASREAD CCW to read module into RTA	RASEEK Seek Address	RTAOWN Index into load list for RTA owner	MCPIK PIK of task interrupted by machine check	RTAID RTA I/O requestor ID
107	108	112	116	120	124	128	132	136
ERPID WTOR request return load index	RASRETR RTA return address after I/O operation	RASIOA RAS I/O request address	RASFCHA RAS FETCH request address	ERPIBA ERPIB queue address	RASDEQA CCB DEQ routine address	XCANRASA RAS cancel routine address	CCENTADR Channel Check entry address	RASRES SYSRES I/O address
138	140	142	144	148	156	160	164	
RASREC SYSREC I/O address	RASLOG SYSLOG device address	RASEMIOA Emergency SIO address	RASCQDSP CCB look-up routine address	SUPRETR Save area for registers 9 and 10	SUPBB Base address (X'1000') for supervisor	SUPBC Base address (X'2000') for supervisor	SUPBD Base address (X'3000') for supervisor	
168 (HIR - Hardware Instruction Retry accumulators)				180 (ECCMAIN - Main storage error accumulators)				
168	170	172	176	180	182	184	188	
HIRACNT HIR accumulated count	HIR1CNT Count threshold value	HIR1TIME Time of day for first error of group	HIRLTIME Time threshold in clock units	ECMACNT Accumulated ECC count for main storage	ECMLCNT Count threshold value	ECM1TME Time of day for first error of count	ECMLTME Time threshold in clock units	
192	194	196	197	198	199	200	201	
RESTARTA Disk restart address	RESTARTP PUB address of unit to be restarted	MCMODE Mode status for machine checks	BUFDEL Count of buffers deleted	RASMSG1 Message byte 1 (6)	RASMSG2 Message byte 2 (7)	EOR Records/track for SYSREC	EOT Tracks/cylinder for SYSREC	

Figure 27. RMS Monitor Table - RASTAB (Part 1 of 2)

Notes:

(1) Areas labeled LDxxSLOT (bytes 0 - 47) are called the Load List and each of the 12 entries are formatted as follows:

BYTE	0	1 2 3
	Flag Byte	Cylinder - Head - Record (disk address of R-transient in the core image directory)

(2) LD00SLOT flag byte:

Bit	Flag	Description
0	X'80'	$$RAST00 module activated.
1	X'40'	Machine check analysis to be performed.
2	X'20	Channel check analysis to be performed.
3	X'10'	Active I/O units are valid.
4	X'08'	System termination situation.
5	X'04	Reserved.
6	X'02'	Reserved.
7	X'01'	Attempt made to record in system termination situation.

(3) LD01SLOT flag byte:

Bit	Flag	Description
0	X'80'	$$RAST01 module activated.
1	X'40'	Build and record channel check records.
2-7	- -	Reserved.

(4) LD10SLOT flag byte:

Bit	Flag	Description
0	X'80'	$$RAST10 module activated.
1	X'40'	Refetch calling module after issuing message.
2-7	- -	Reserved.

(5) LDxxSLOT flag byte:

Bit	Flag	Description
0	X'80'	$$RASTxx module activated; that is, should be fetched.
1-7	- -	Reserved.

(6) RASMSG1:

Bit	Flag	Description
0-3	- -	Reserved.
4	X'08'	Timer damage.
5	X'04'	ECC in Quiet mode.
6	X'02'	Reserved.
7	X'01'	MCAR repair failed.

(7) RASMSG2:

Bit	Flag	Description
0	X'80'	Check damage.
1	X'40'	Last track on SYSREC.
2	X'20'	C40 buffer pages deleted.
3	X'10'	Soft machine checks disabled.
4	X'08'	ECC MCI disabled.
5	X'04'	SYSREC full-run EREP.
6	X'02'	Error on SYSREC at BBCCHHR.
7	X'01'	Soft machine check.

Figure 27. RMS Monitor Table - RASTAB (Part 2 of 2)

RASLINK

0 (Decimal Displacement) CPUID	8 RASDMC	9 RASFLAGS	10 MCFLAGS	11 RASMODEL	12 RASTABA	16 RASBASE
CPU ID field	Damaged Channel byte	RAS flag byte	Machine Check flags	CPU Model	RAS Table (RASTAB) address	Base address for RAS Monitor
XXXXXXXX	X	X	X	X	XXXX	XXXX

Key to RAS Linkage Area displacements:

0 CPU ID field.

8 Address of damaged channel, or X'FF' if no channel damaged.

9 RAS Flag byte:

bit	flag	description
0	X'80'	RAS active
1	X'40'	RAS SIO flag
2	X'20'	RTA in control
3	X'10'	RAS I/O delayed
4	X'08'	Channel check on error SIO
5	X'04'	Reserved
6	X'02'	Channel check on SIO
7	X'01'	I/O active for SIO

10 Machine Check Flags:

bit	flag	description
0-4	——	Reserved
5	X'04'	Hard machine check
6	X'02'	All machine records built
7	X'01'	All channel check records built

11 Largest CPU Model.

12 Address of RAS Table (RASTAB).

16 Address used for base register in RAS Monitor Program.

Figure 28. RMS Linkage Area (RASLINK)

Displacement	0	1	2	3	4-11	12-15	16-19	20-27	28-43	44-51
Label	CCKEY	CCN1	CCN2	CCMOD	CCCPUID	CCDATE	CCTIME	CCNAME	CCAIOU	CCFCCW
Record 1 *	Record ID	Record Number	Total Records	CPU Model Code	CPU ID Information	Date	Time of Day	Job Name	Active I/O Units	Failing CCW

Displacement	52-59	60-63	64-65	66	67-69	70-73	74-77	78	79
Label	CCCSW	CCECSW	CCDEVTYP	CCCHID	CCCUA	CCMPI		CCSYSCON	CCGUARD
Record 1 (continued)	CSW	Extended CCW	Device Type	Channel ID	Control Unit	Multi-processing Information	Reserved	System Condition Byte	Guard Byte X'FF'

Displacement	0-3	4-78	79
Record 2	Same as Record 1	Channel Logout Area	Same as Record 1

Displacement	0-3	4-24	25-78	79
Record 3	Same as Record 1	Channel Logout Area	Unused	Same as Record 1

* Note: Only Record 1 is written for the Model 155.
Record 1 is also addressed symbolically as CCREC with the logout data area, Bytes 4-78, addressed as CCLOGD.

Figure 29. RMS Channel Check Record on SYSREC

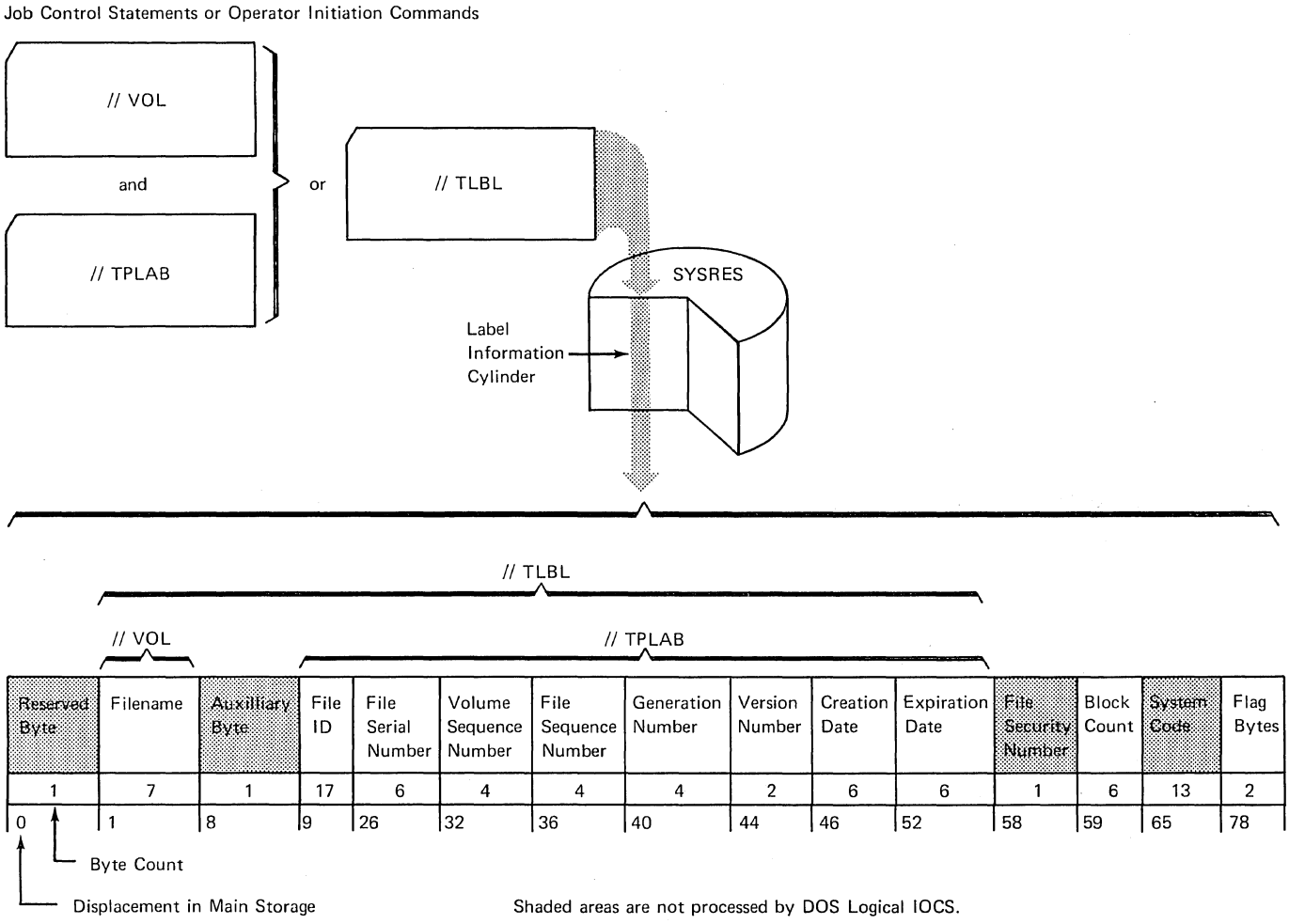


Figure 30. Format of SYSRES Tape Label Information

1	2	3		4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19		Field
DLBL-EXTENT Indicator	Filename	DA/IS Switch	File ID		Format ID	File Serial Number	Volume Seq. No.	Creation Date	Expiration Date	Reserved	Open Code	System Code	Volume Serial Number	EXTENT Type	EXTENT Seq. No.	Extent Lower Limit	Extent Upper Limit	Logical (Symbolic) Unit Address	2321 Lower Cell	2321 Upper Cell	Another Extent if DA or ISFMS
1	7	1	44		1	6	2	3	3	2	1	13	6	1	1	4	4	2	1	1	Bytes
0	1	8	9		53	54	60	62	65	68	70	71	84	90	91	92	96	100	102	103	Displacement

Field	Name	Description
1.	DLBL-EXTENT	SD Bit 0: 1 = Next extent on a new pack. Bit 1: 1 = Last extent. Bit 2: 1 = Bypass extent. Bit 3: 1 = New volume on same unit. Bit 4: 1 = Extent limits omitted. Bit 5: 1 = Extent converted to DASD address. Bit 6: 1 = No EXTENT/XTENT card. Bit 7: 1 = Unused. DA or ISFMS Number of extents.
2.	Filename	
3.	DA/IS Switch	Bits 0-3: Unused. Bit 4: 1 = Extent limits omitted. Bit 5: 1 = Extent converted to DASD address. Bit 6 & 7: Unused.
4.	File ID	File identifier including generation and version numbers. If field is missing on DLBL card, Filename padded with blanks is inserted.
5.	Format ID	Numeric 1 is inserted.
6.	File Serial No.	Volume serial number from first extent.
7.	Volume Seq. No.	Always initialized to X'0001'.
8.	Creation Date	Initialized with 3 bytes of X'00'.
9.	Expiration Date	If date is in the form YYDDD, it is converted to YDD. If date is in retention period form, 1 to 4 characters, the field is padded with binary zeros.
10.	Reserved	The retention period, if specified is converted to a 2-byte number and inserted in this field.
11.	Open Code	DLBL type: S = Sequential D = Direct Access C or E = Indexed sequential File Management System where: C = Load create function E = Load extend function
12.	System Code	Initialized to contain DOS/360 VER 3. This field is not processed by DOS.
13.	Volume Serial No.	Volume serial number for extent.
14.	Extent Type	Same codes as in Format - 1 label: X'00' = Next three fields do not indicate any extent. X'01' = Prime data area (ISFMS) or consecutive area, etc., (that is the extent containing the user's data records). X'02' = Overflow area of an ISFMS file. X'04' = Cylinder index or master index of an ISFMS file. X'40' = User label track area. X'8n' = Shared cylinder indicator, where n = 1, 2, or 4.
15.	Extent Seq. No.	Number of extent as determined by the extent card sequence.
16. & 17.	Extent Lower and Upper Limits	Before the OPEN, DLBL/EXTENT information is in the relative track form of HHNNT followed by three bytes of binary zeros. HH = Relative (to 0) start address in tracks. NN = Number of tracks. T = 0 or upper track number for split cylinder in SD files. Following an OPEN on DLBL/EXTENT cards, or whenever DLAB/XTENT cards are used, the extent lower and upper limits are each in the CCHH format.
18.	Logical (Symbolic) Unit Address	This 2-byte field identifies the logical unit with the same code as that used in a CCB. The first byte identifies the unit class: X'00' = System Logical Unit X'01' = Programmer Logical Unit The second byte identifies the logical unit within its class. Thus X'0003' denotes SYSLST and X'0103' denotes SYS003.
19.	2321 Lower Cell 2321 Upper Cell	2321 extent lower and upper cell limit. This 2-byte field contains zeros for 2311/2314/2319 disk.

Note: For Sequential Disk files, a complete 104-byte block is repeated for each new EXTENT.
For Direct Access and ISFMS files, only fields 13 through 18 are repeated for each EXTENT.

Figure 31. SYSRES DASD Label Information

The index gives the relationship of core-image phase names, relocatable module names, microfiche labels, and microfiche identification numbers to each other.

An asterisk indicates the microfiche label. If the microfiche label differs from both the phase and the module name, it is so indicated in parentheses.

When a phase or module takes up more than one microfiche card, the identification number of only the first card is shown.

For the complete microfiche cross-reference index, see Introduction to DOS Logic listed in the Preface.

Core Image Phase Name	Relocatable Module Name	Card ID
$$A$IPL1*	None	CTL.001.00
$$A$IPL2 ($$A$IPL1)	None	CTL.001.00
$IPLRT2	IJBIPL*	CTL.201.00
$IPLRT3	IJBIPL3 (IJBIPL)	CTL.201.00
$IPLRT4	IJBIPL4 (IJBIPL)	CTL.201.00
$JOBACCT*[1]	$JOBACCT	CTL.187.25
$JOBCTLA	IJBJC1*	CTL.202.00
$JOBCTLD	IJBJC2*	CTL.203.00
$JOBCTLF	IJBJC5*	CTL.206.00
$JOBCTLG	IJBJC3*	CTL.204.00
$JOBCTLJ	IJBJC4*	CTL.205.00
$JOBCTLK	IJBJC6*	CTL.207.00
$JOBCTLM	IJBJC7*	CTL.208.00
$JOBCTLN	IJBJC8*	CTL.208.50

[1]$JOBACCT is a dummy entry to be replaced by a user-written routine.

GLOSSARY

For a more complete list of data processing terms, refer to IBM Data Processing Techniques, A Data Processing Glossary, GC20-1699.

American National Standard Label Format: The tape file format used when the label is written in the ASCII mode.

ASCII (American National Standard Code for Information Interchange): A 128-character, 7-bit code. The high-order bit in the System/360 8-bit environment is zero.

block prefix: An optional, 0-99 byte field preceding an ASCII record. It contains user specified data or, for variable-length (format D) records, the physical record length.

CCH (Channel Check Handler): A feature that assesses System/370 channel errors to determine if the system can continue operations.

channel inboard error: An error that occurs between one I/O device and the central processing unit.

data conversion: The process of changing data from one form of representation to another.

DOS Volume Statistics: A facility that monitors and records the number of temporary read and write errors on currently accessed tape volumes. This facility has two options, Error Statistics by Tape Volume (ESTV) and Error Volume Analysis (EVA).

EREP (Environmental Recording, Editing, and Printing): A program that processes the data contained on the system recorder file.

ESTV (Error Statistics by Tape Volume): One of the two options of the DOS Volume Statistics. With ESTV support, the system collects data on tape errors by volume for any tape volumes used by the system.

EVA (Error Volume Analysis): One of the two options of the DOS Volume Statistics. With this option, the system issues a message to the operator when a number of temporary read or write errors (specified by the user at system generation time) has been exceeded on a currently accessed tape volume.

file section number: For ASCII files, the number indicates the order of a volume in a given file or multifile set. The first file must be numbered 0001. The ASCII file section number is equivalent to the EBCDIC volume sequence number.

I/O (Input/Output) error logging: The process of recording OBR and SDR records on the system recorder file.

job accounting interface: A function that accumulates accounting information for each job step to: charge usage of the system, help plan new applications, and help supervise system operation more efficiently.

load point: The beginning of the recording area on a reel of magnetic tape.

MCAR (Machine Check Analysis and Recording): A feature that records System/370 machine check interrupt error information on the system recorder file and then attempts to recover from the interrupt.

MCI (Machine Check Interrupt): The interrupt that occurs if the central processing unit fails to operate.

MCRR (Machine Check Recording and Recovery): The recording of pertinent data on the system recorder file after either a machine check interrupt or a channel inboard error occurred on System/360 Model 30, Model 40, or Model 50.

nonstandard labels: Labels that do not conform to the System/360 standard label specifications. They can be any length, need not have a specified identification, and do not have a fixed format.

OBR (Outboard Recorder): A feature that records pertinent data on the system recorder file when an unrecoverable I/O error occurs.

PCIL (Private Core Image Library): A file referenced in the same manner and for the same purposes as the system core image library, but distinct from the system core image library. PCIL increases available core image library space to enable compiling, linkage editing, and executing in the foreground partition, when a private core image library is assigned to that foreground partition.

RDE: See Reliability Data Extractor.

Reliability Data Extractor (RDE): A function that provides hardware reliability data that is analyzed by IBM.

RMS (Recovery Management Support): A feature for System/370 that consists of the MCAR (Machine Check Analysis and Recording) and CCH (Channel Check Handler) functions. RMS gathers information about System/370 hardware reliability and attempts certain error recovery operations. RMS is a part of the entire reliability, availability, and serviceability support for System/370.

SDR (Statistical Data Recorder): A feature that records the cumulative error status of an I/O device on the system recorder file.

system recorder file: The file that is used to record hardware reliability data.

Indexes to systems reference library manuals are consolidated in the publication DOS Master Index, GC24-5063. For additional information about any subject listed below, refer to other publications listed for the same subject in the Master Index.

IBM **Technical Newsletter**

File No. S360-36 (DOS Release 26)

Base Publ. No. GY24-5086-4

This Newsletter No. GN33-8686

Date: October 28, 1971

Previous Newsletter Nos. None

DOS IPL AND JOB CONTROL
PROGRAM LOGIC MANUAL

This Technical Newsletter, a part of Release 26 of the IBM Disk Operating System, provides replacement pages for your publication. These replacement pages remain in effect for subsequent DOS releases unless specifically altered. Pages to be inserted and/or removed are:

Cover- 10	137-140
41, 42	167,168
61, 62	215,216
67, 68	221,222
77, 78	225,226
121,122	231,232
125,126	235-238

A change to an illustration is indicated by a vertical line to the left of the change.

Summary of Amendments

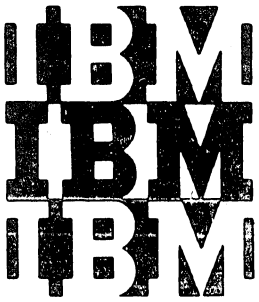
This TNL contains documentation changes for enhancements to OLTEP. It also contains maintenance changes and technical corrections.

Note: Please insert this page in your publication to provide a record of changes.

IBM Laboratory, Publications Dept., Uithoorn Netherlands

PRINTED IN U.S.A.

File Number S360-36 (DOS)
Order No. GY24-5086-4



Program Logic

DOS IPL and Job Control

Program Number 360N-CL-453

This reference publication describes the internal logic of the IBM Operating System, Initial Program Load (IPL) and Job Control Programs. It is intended for use by persons involved in program maintenance and by system programmers who are altering the program design. Program logic information is not needed for normal operation of the IPL and Job Control Programs. It is designed to be used as a supplement to the program listing.

Effective use of this manual requires an understanding of IBM System/360 or System/370 operation and of the IBM Disk Operating System control and service programs, macro instructions, and operating procedures. Reference publications for this information are listed in the Preface of this manual.

For titles and abstracts of other associated publications, see the IBM System/360 and System/370 Bibliography, GA22-6822.

DOS Release 26

Fifth Edition (June, 1971)

This publication was formerly titled IBM System/360 Disk Operating System IPL and Job Control Programs. Although titles of some DOS publications (including this one) have been simplified, the change does not affect the contents of the publications.

This edition, with Technical Newsletter GN33-8686, applies to Release 26 of the IBM Disk Operating System and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the specifications herein; before using this publication in connection with the operation of IBM systems, consult the latest System/360 and System/370 SRL Newsletter, GN20-0360, for the editions that are applicable and current.

This edition is a major revision of, and obsoletes, GY24-5086-3.

Summary of Amendments

This edition contains maintenance changes, and it reflects Job Accounting Interface, OLTEP, Data Set Security, PCIL (Private Core Image Library), MCAR/CCH support, and RDE (Reliability Data Extractor). The IBM 3211 Printer and the IBM 1255/1259 Magnetic Character Readers are also supported in this edition.

Technical Newsletter GN33-8686 contains maintenance changes and technical corrections.

The flowchart symbols used in this manual conform with American National Standards Institute, Inc., flowcharting standards. See Appendix C for an explanation of the new symbols.

Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Request 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 Dept., P.O. Box 24, Uithoorn, Netherlands. Comments become the property of IBM.

This Program Logic Manual (PLM) is a detailed guide to the IBM Disk Operating System IPL and Job Control programs. It supplements the program listings by providing descriptive text and flowcharts.

Note: Although titles of some DOS publications have been simplified, the change does not affect the contents of the publications.

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

- Introduction to DOS Logic, GY24-5017.
- DOS Linkage Editor, GY24-5080.
- DOS Supervisor and Related Transients, GY24-5151.
- DOS Logical Transients, GY24-5152.
- DOS System Service Programs, GY24-5153.
- DOS Librarian, GY24-5079.

Prerequisite publications that will aid in the use of this manual are:

- IBM System/360 Principles of Operation, GA22-6821.
- DOS System Control and Service, GC24-5036.
- IBM System/360 Disk and Tape Operating Systems, Assembler Specifications, GC24-3414.

Publications related in subject matter to the seven system control PLMs are:

- DOS Supervisor and I/O Macros, GC24-5037.
- DOS System Generation, GC24-5033.
- DOS Operating Guide, GC24-5022.
- DOS Messages, GC24-5074.
- DOS Data Management Concepts, GC24-3427.

Titles and abstracts of other related publications are listed in the IBM System/360 and System/370 Bibliography, GA22-6822.

This manual consists of four major sections. The first section is an introduction to the IPL and Job Control programs. The next two sections are a discussion of contents of the IPL and Job Control phases. The last section of the manual, the appendixes, contains label lists, error messages, charts, and tables for use as references in analyzing program details.

The detailed flowcharts are identified by letters AA through ZZ. Numerals such as 00 for the program level flowcharts identify the more general flowcharts.

CHARTS

Chart 00. Disk Operating System Program Flow

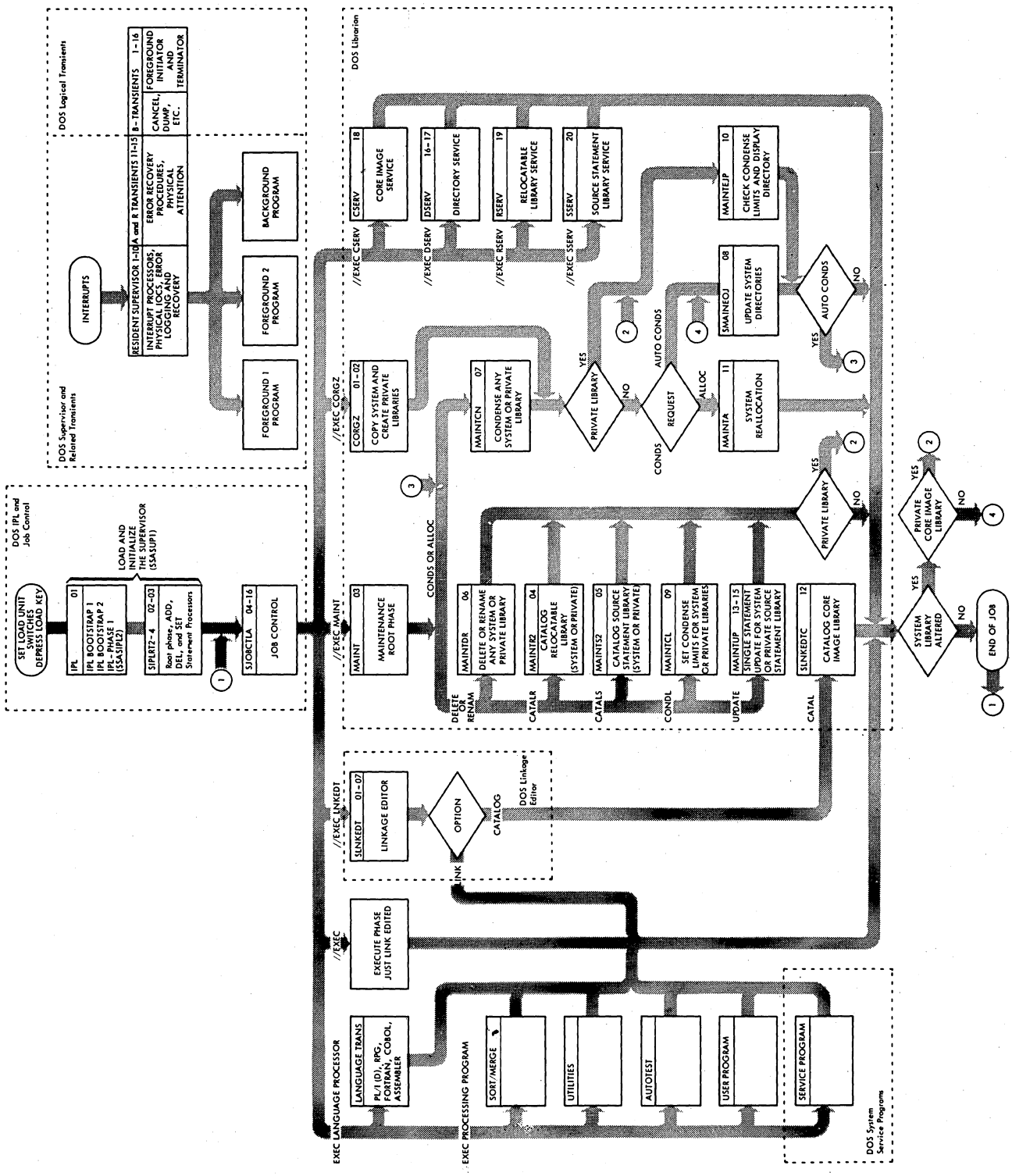


Chart AE. $$A$IPL2 - Move I/O Tables
Refer to Chart 01.

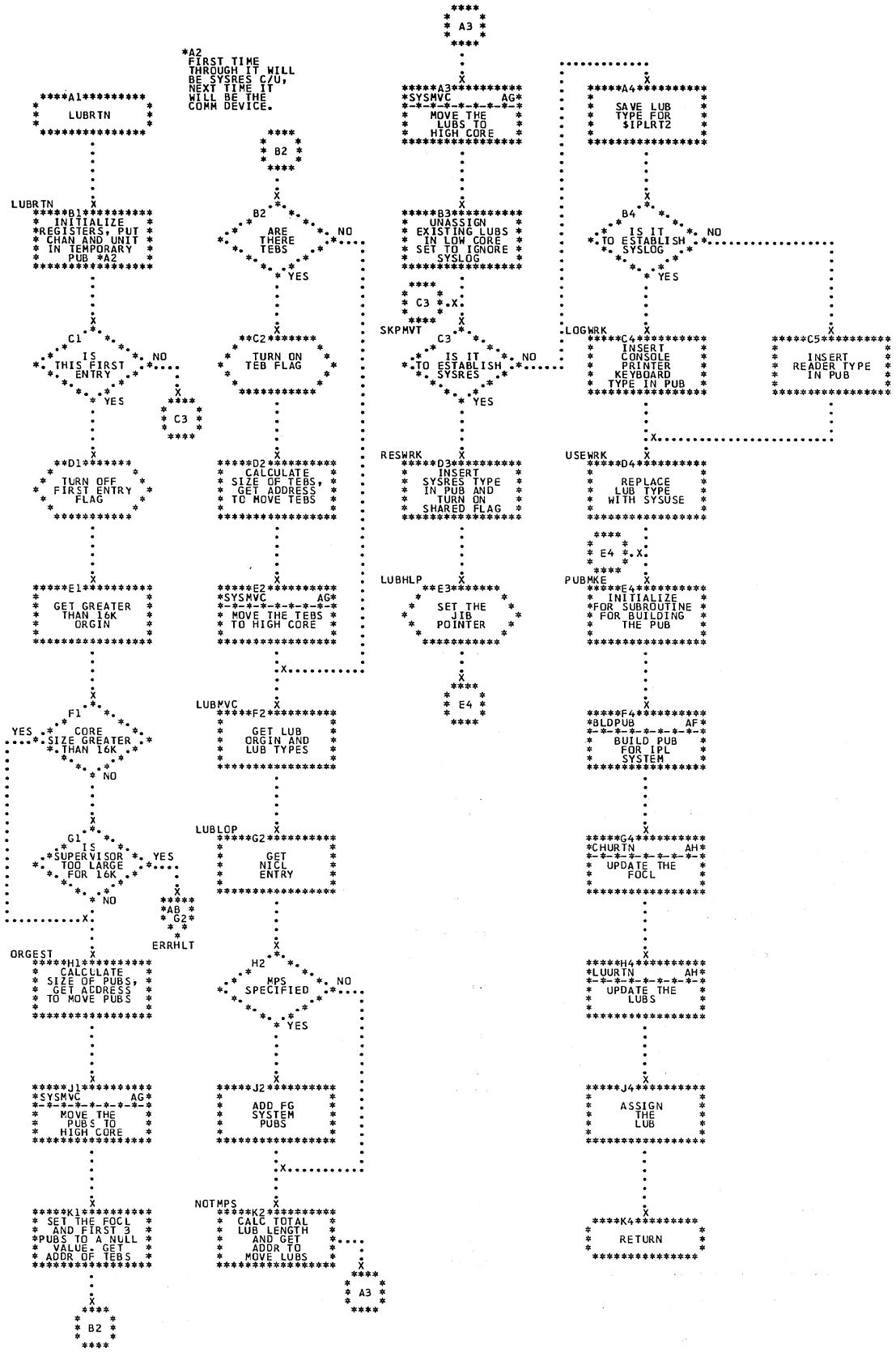


Chart AF. $$A$IPL2 - Build PUB Table
Refer to Chart 01.

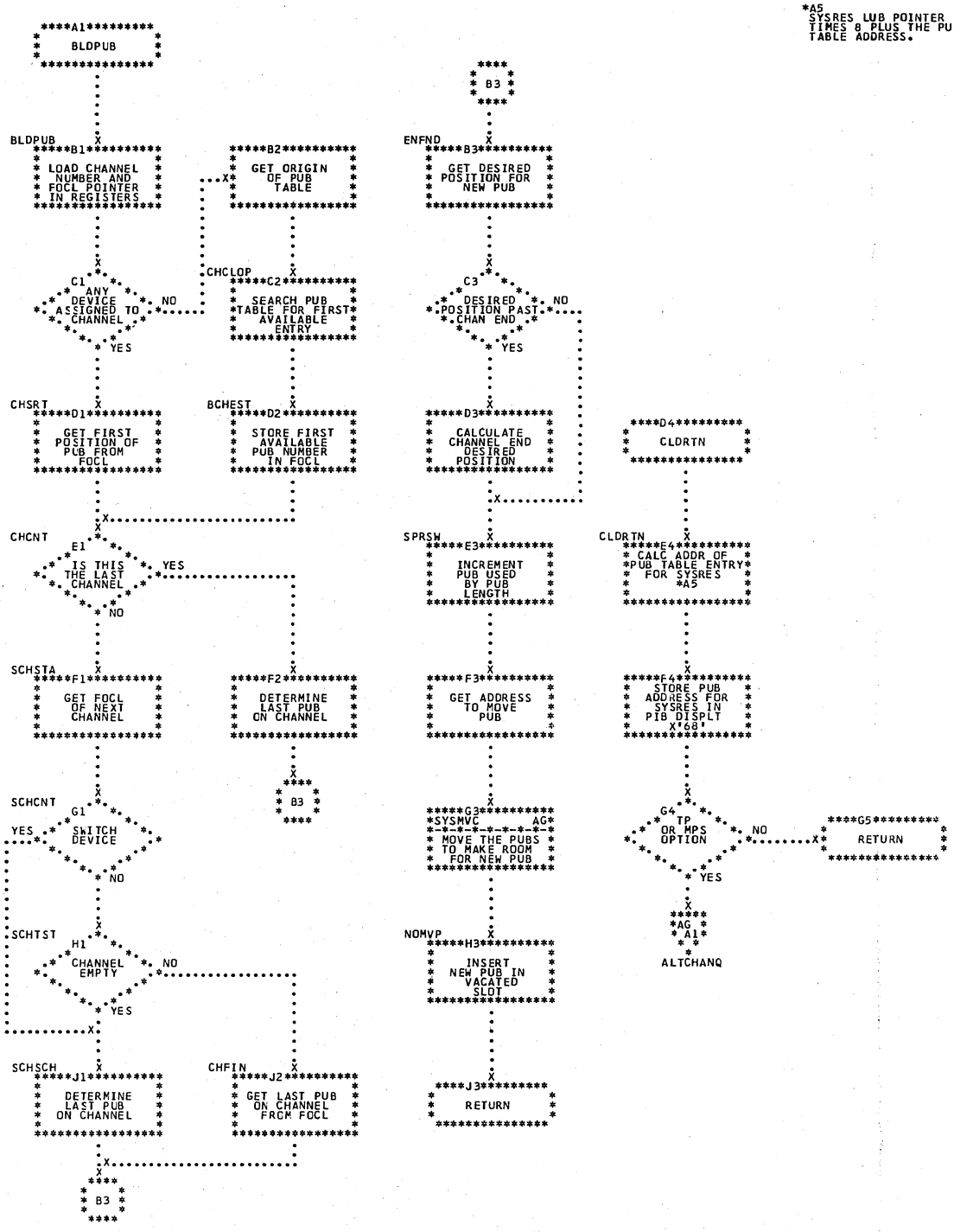


Chart AZA. $IPLRT4 - Find PUB and I/O Subroutines
Refer to Chart 03.

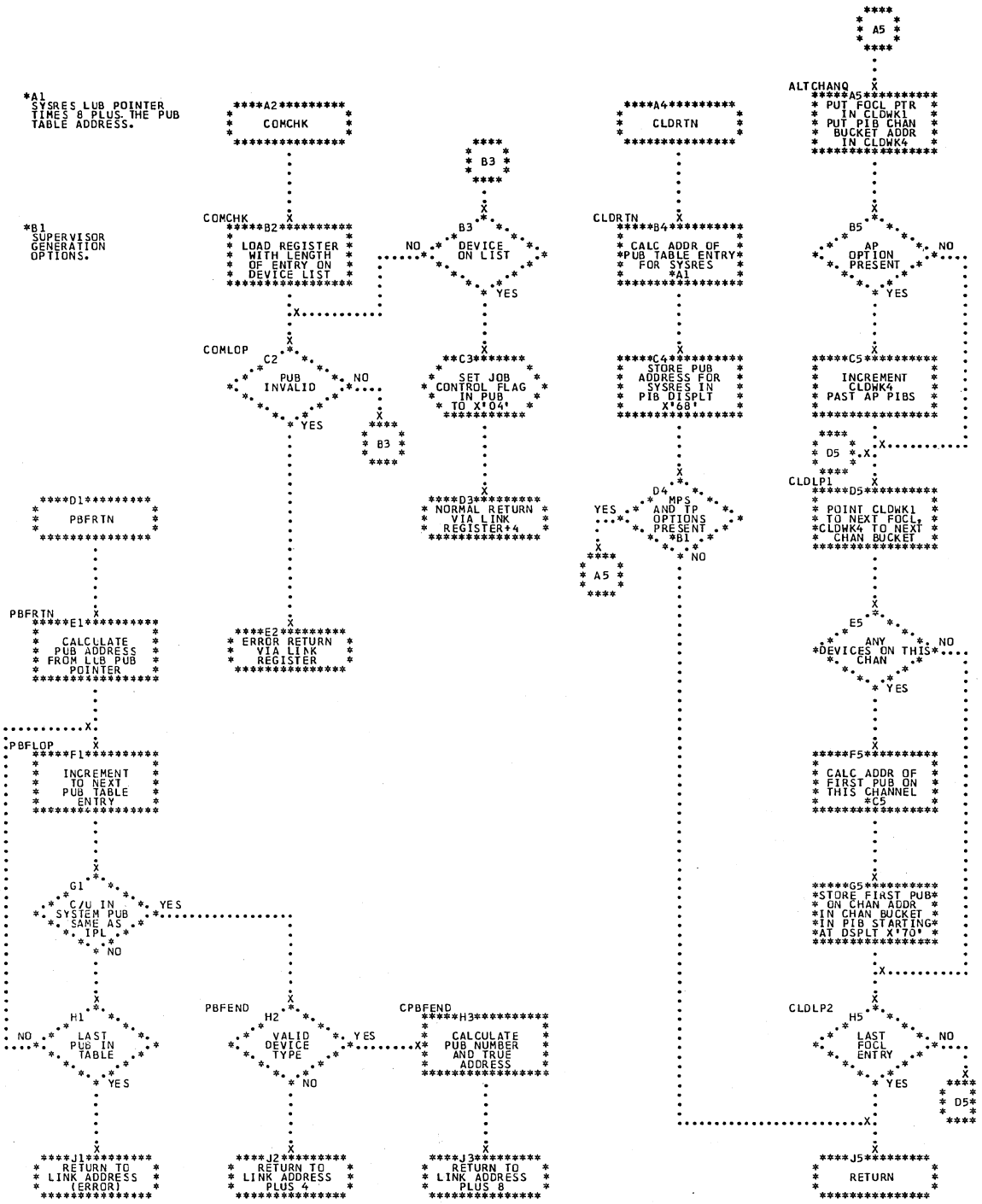


Chart AZB. $IPLRT4 - Date and Time Subroutines
Refer to Chart 03.

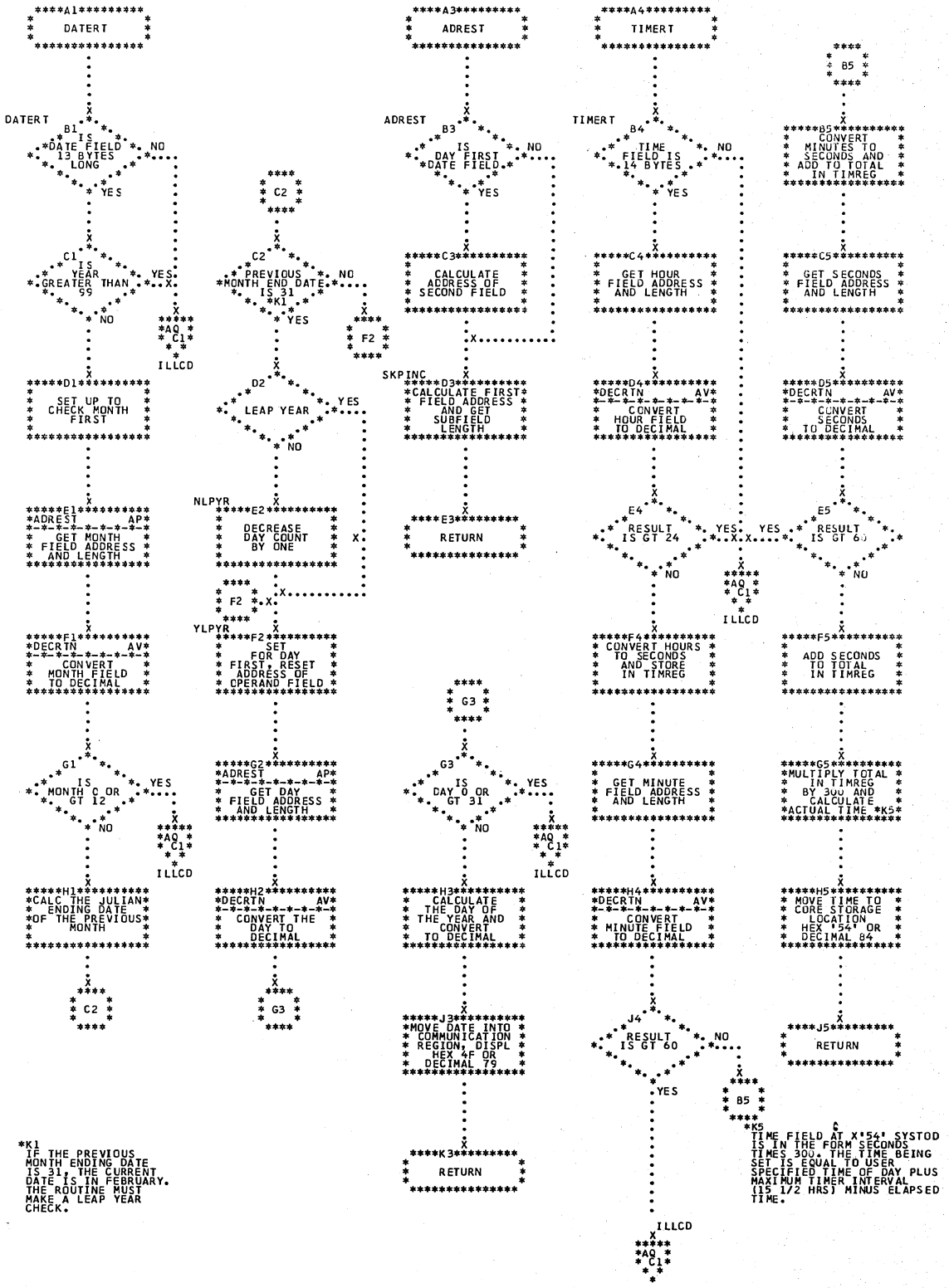


Chart BB. $JOBCTLA - Initialization and Control Statement Read
Refer to Chart 04.

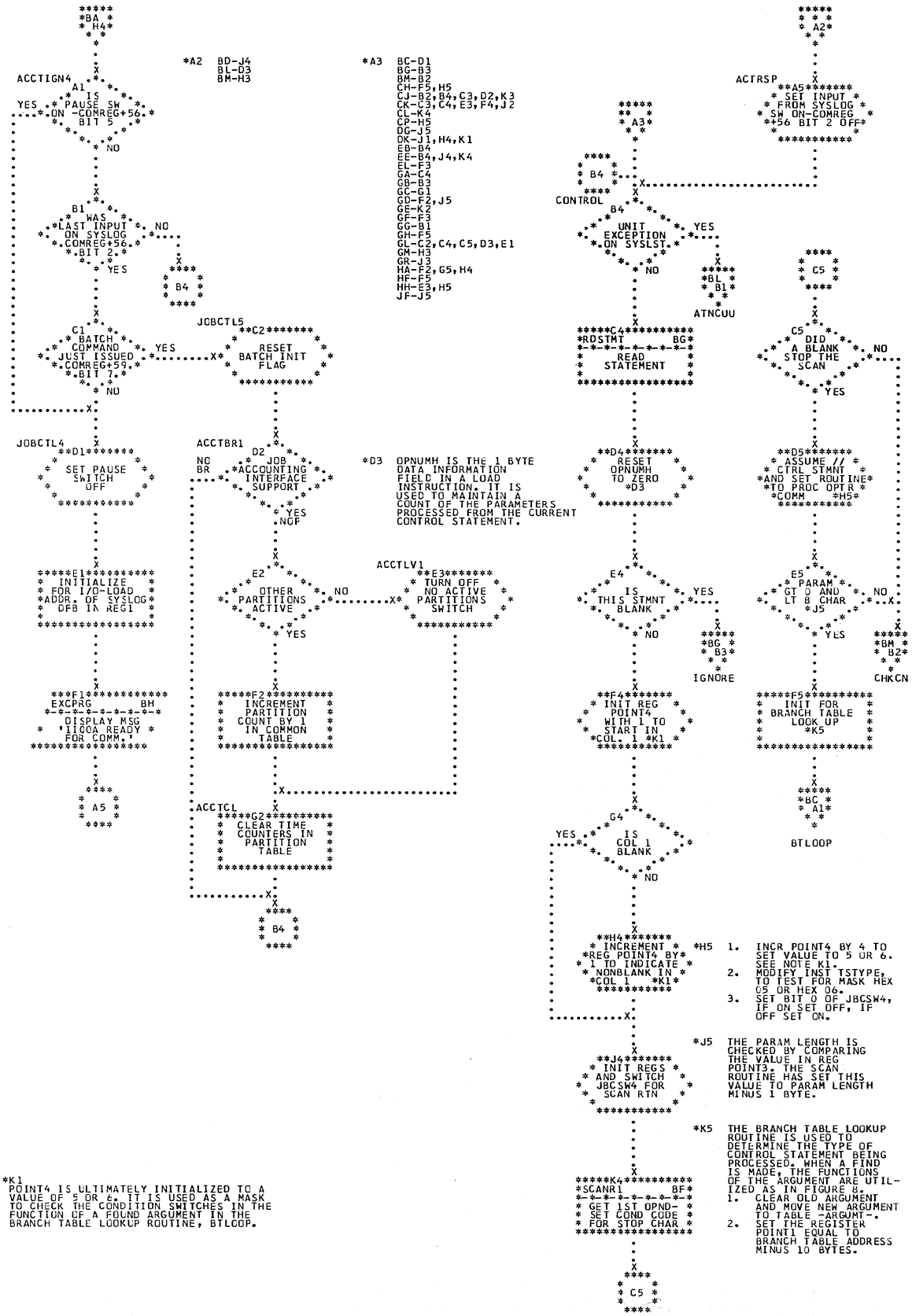


Chart BC. $JOBCTLA - Phase Vector Table Lookup
Refer to Chart 04.

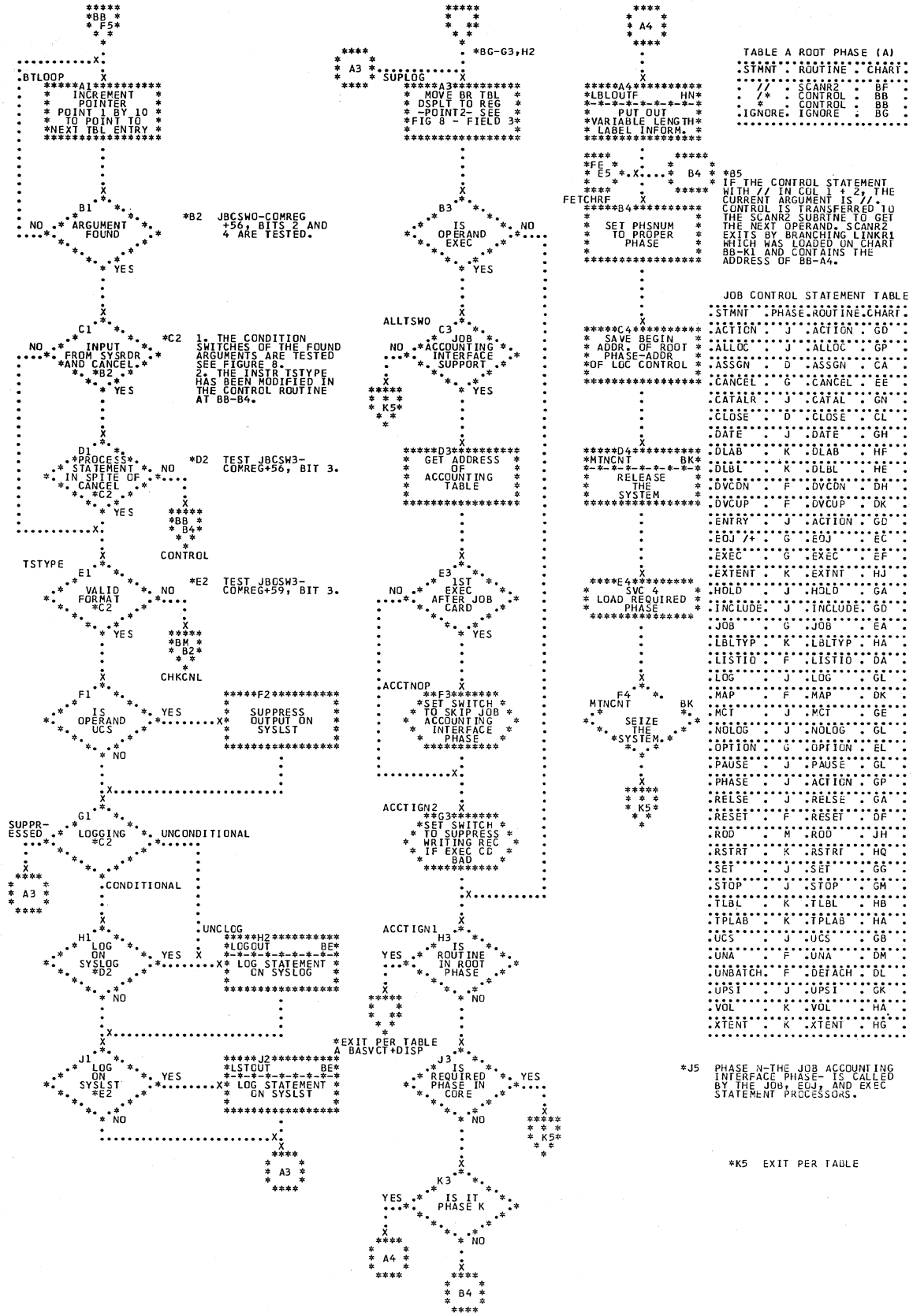


Chart BM. $JOBCTLA - Error Subroutines (Part 2 of 2)
Refer to Chart 04.

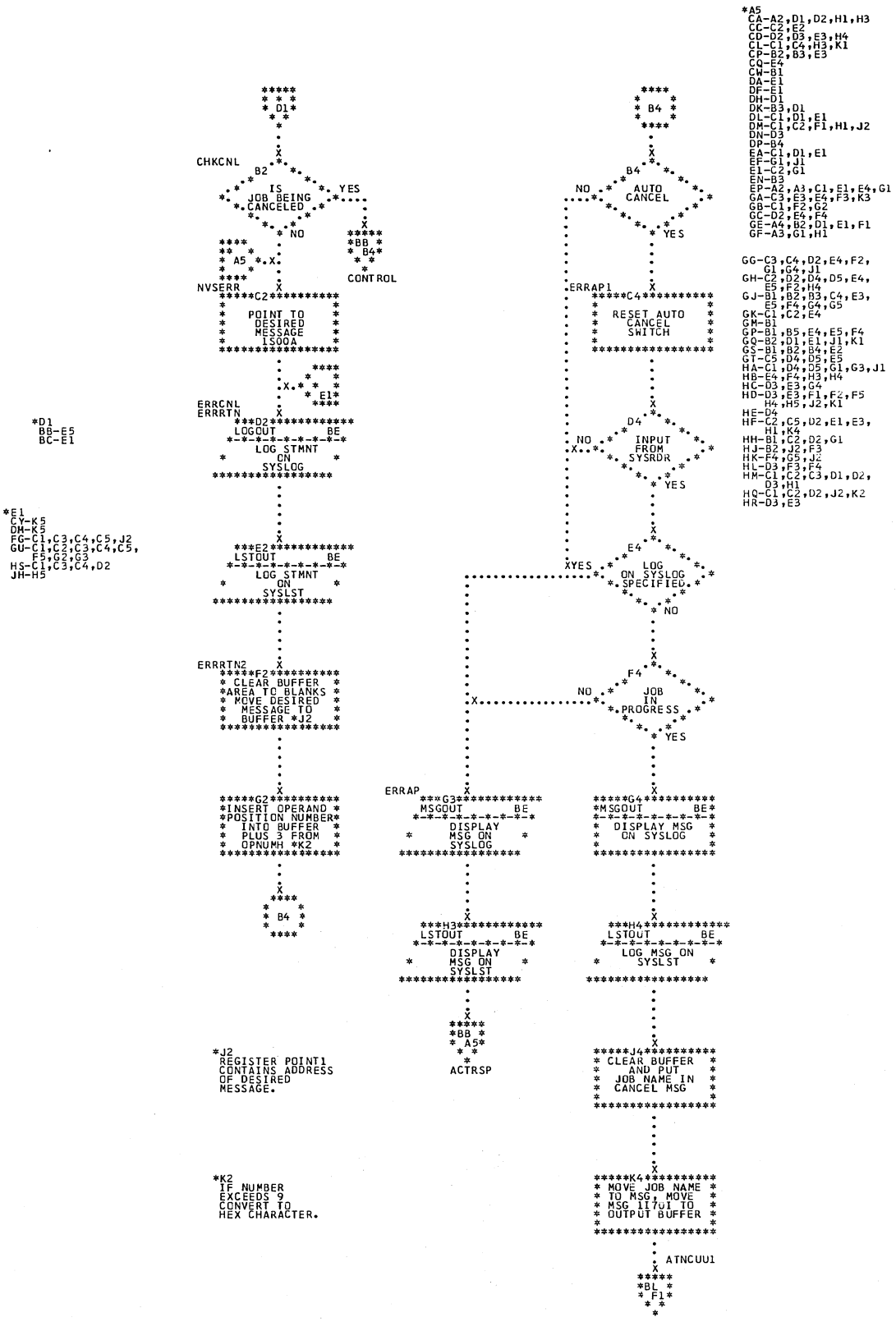


Chart BN. $JOBCTLA - Relocation Subroutines
Refer to Chart 04.

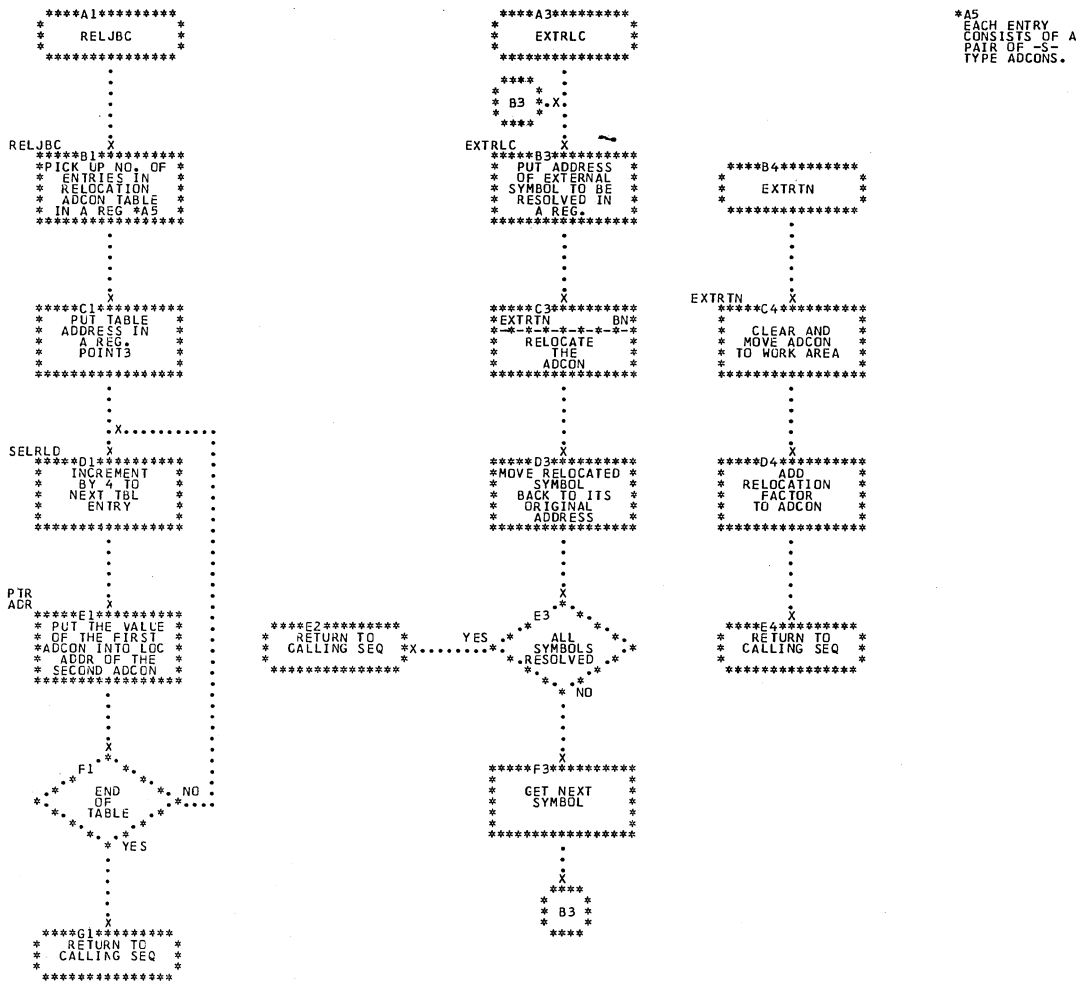


Chart EC. $JOBCTLG - /& Statement Processor (Part 1 of 3)
Refer to Chart 09.

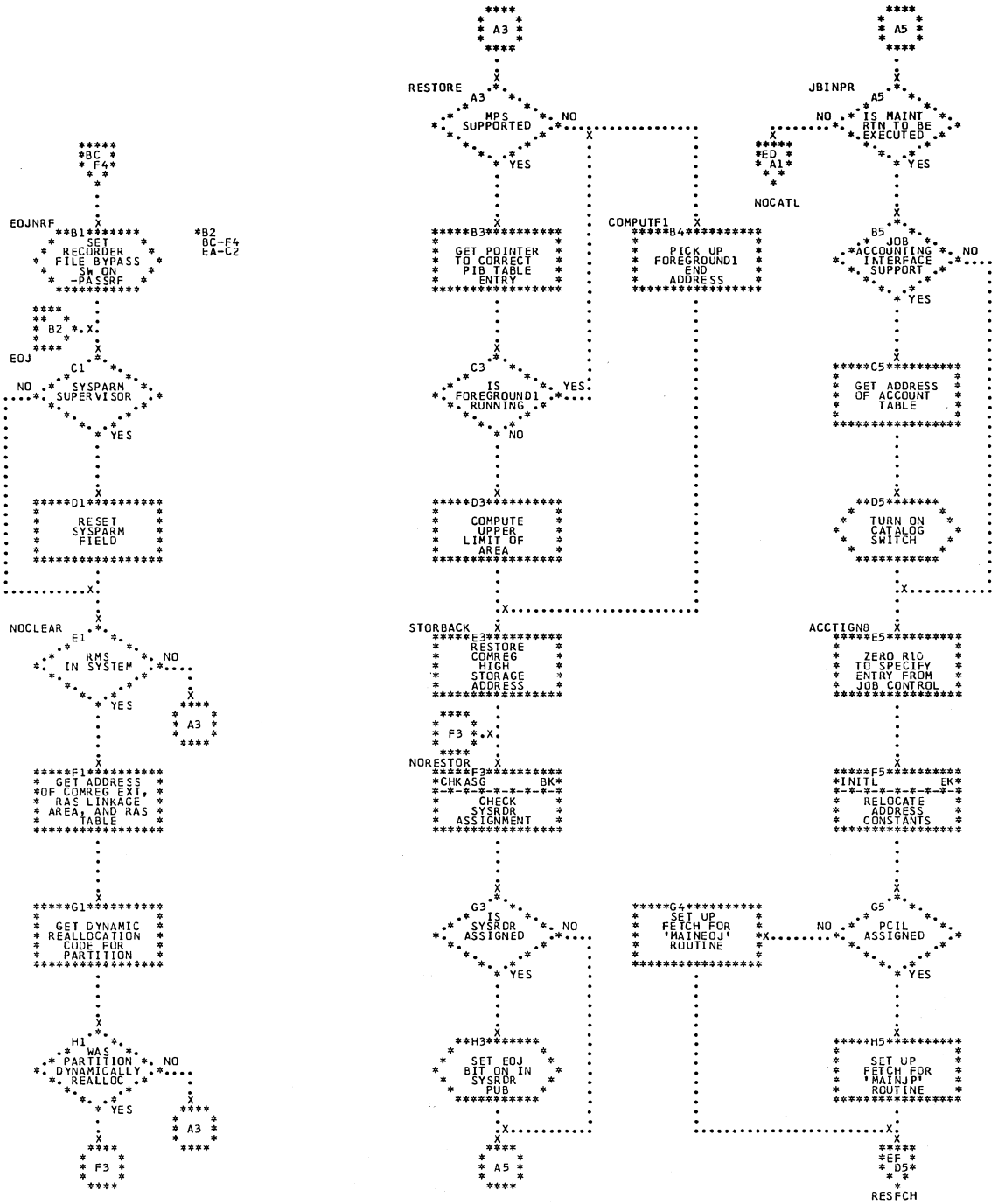


Chart ED. $JOBCTLG - /& Statement Processor (Part 2 of 3)
Refer to Chart 09.

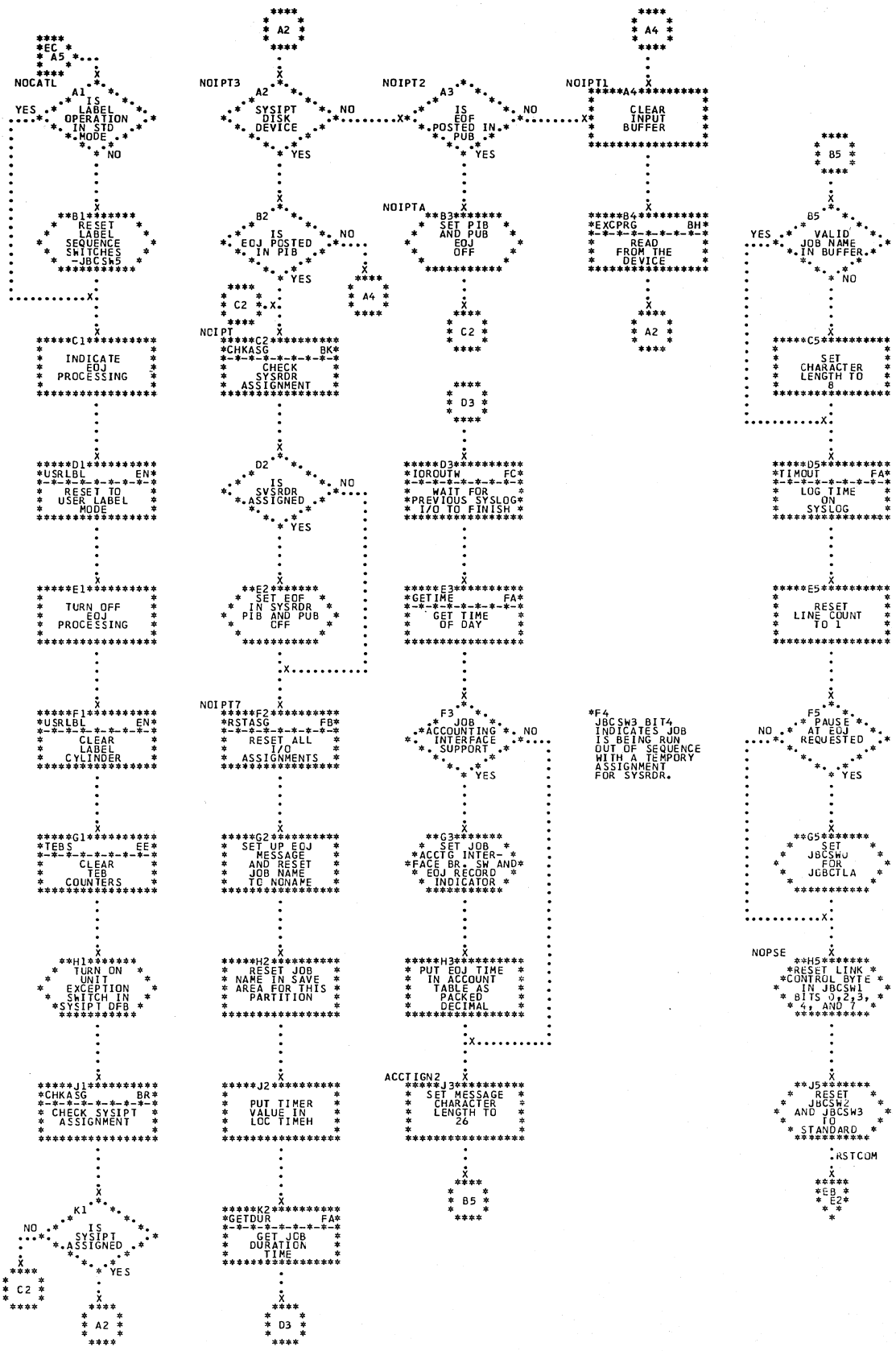


Chart EG. $JOBCTLG - EXEC Statement Processor (Part 2 of 5)
Refer to Chart 09.

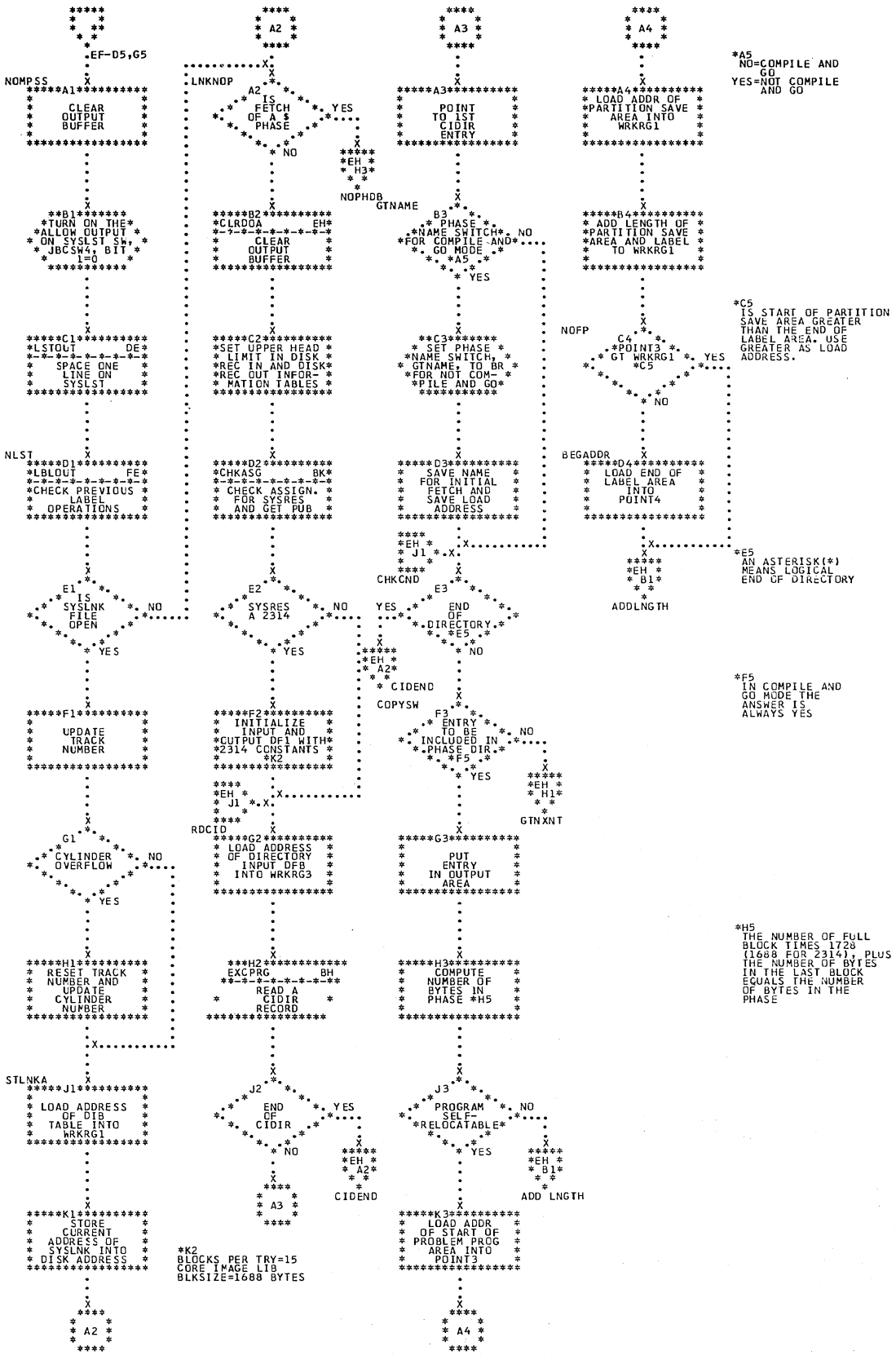


Chart EH. $JOBCTLG - EXEC Statement Processor (Part 3 of 5)
Refer to Chart 09.

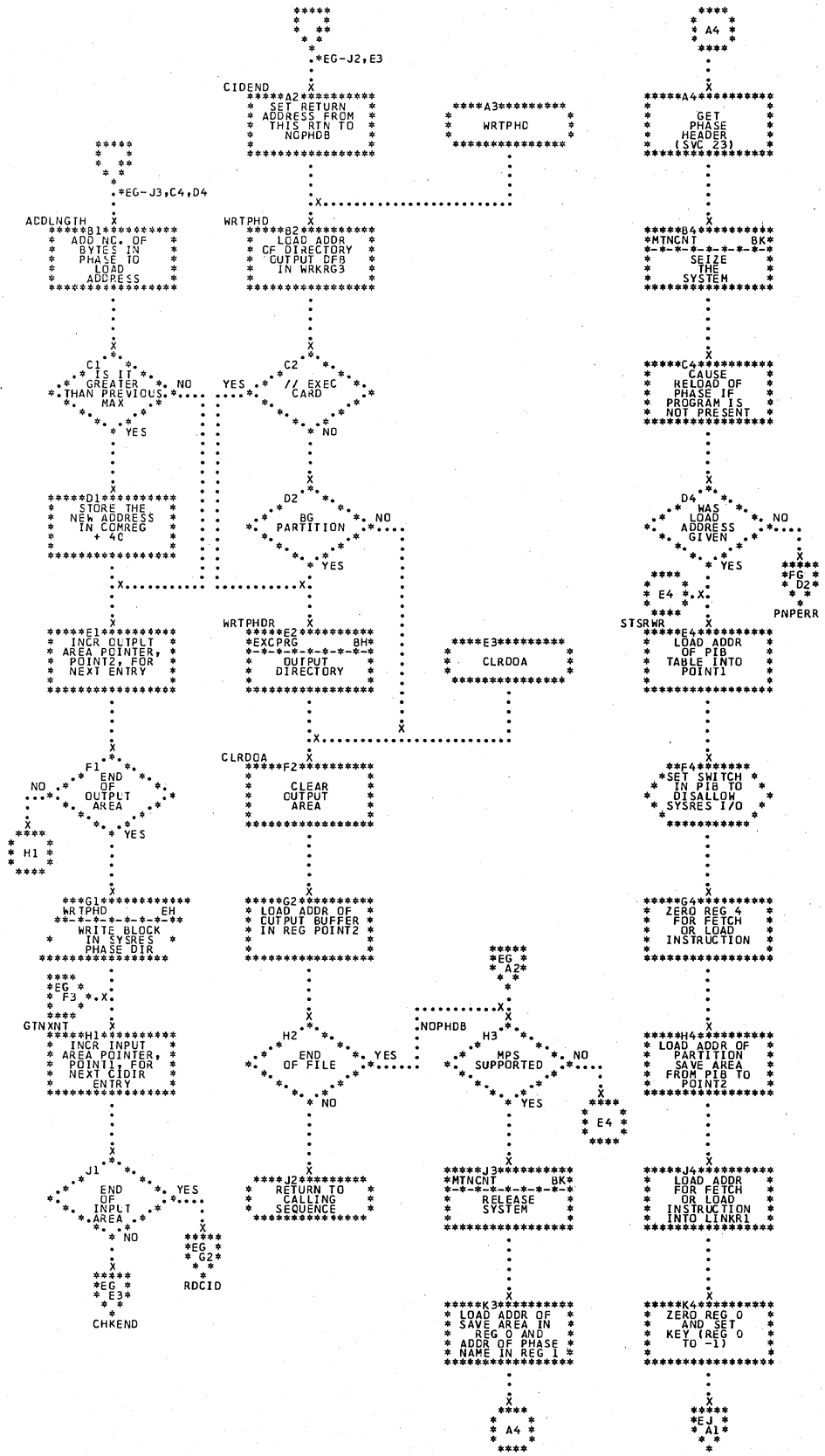


Chart FE. $JOBCTLG - Label Processing Subroutines (Part 1 of 2)
Refer to Charts 08 and 09.

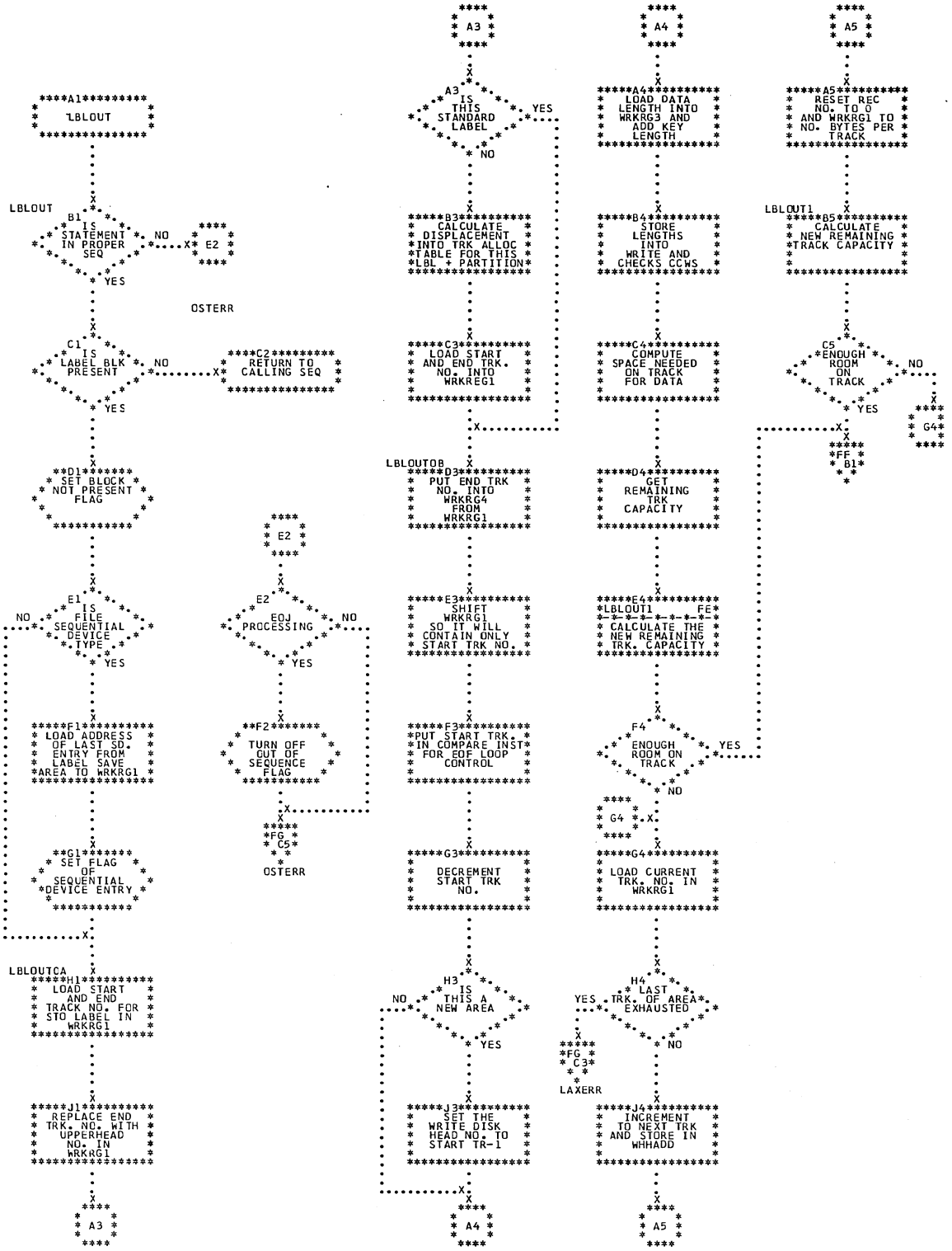


Chart FF. $JOBCTLG - Label Processing Subroutines (Part 2 of 2)
Refer to Charts 08 and 09.

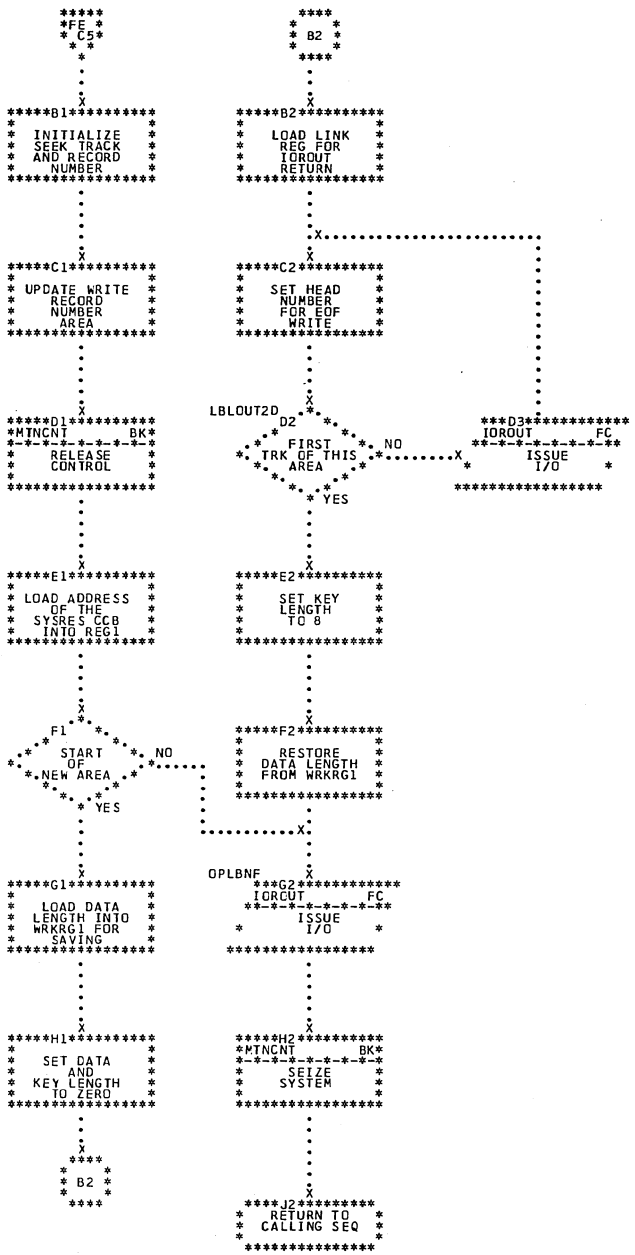


Chart FG. $JOBCTLG - Error Subroutines
Refer to Charts 08 and 09.

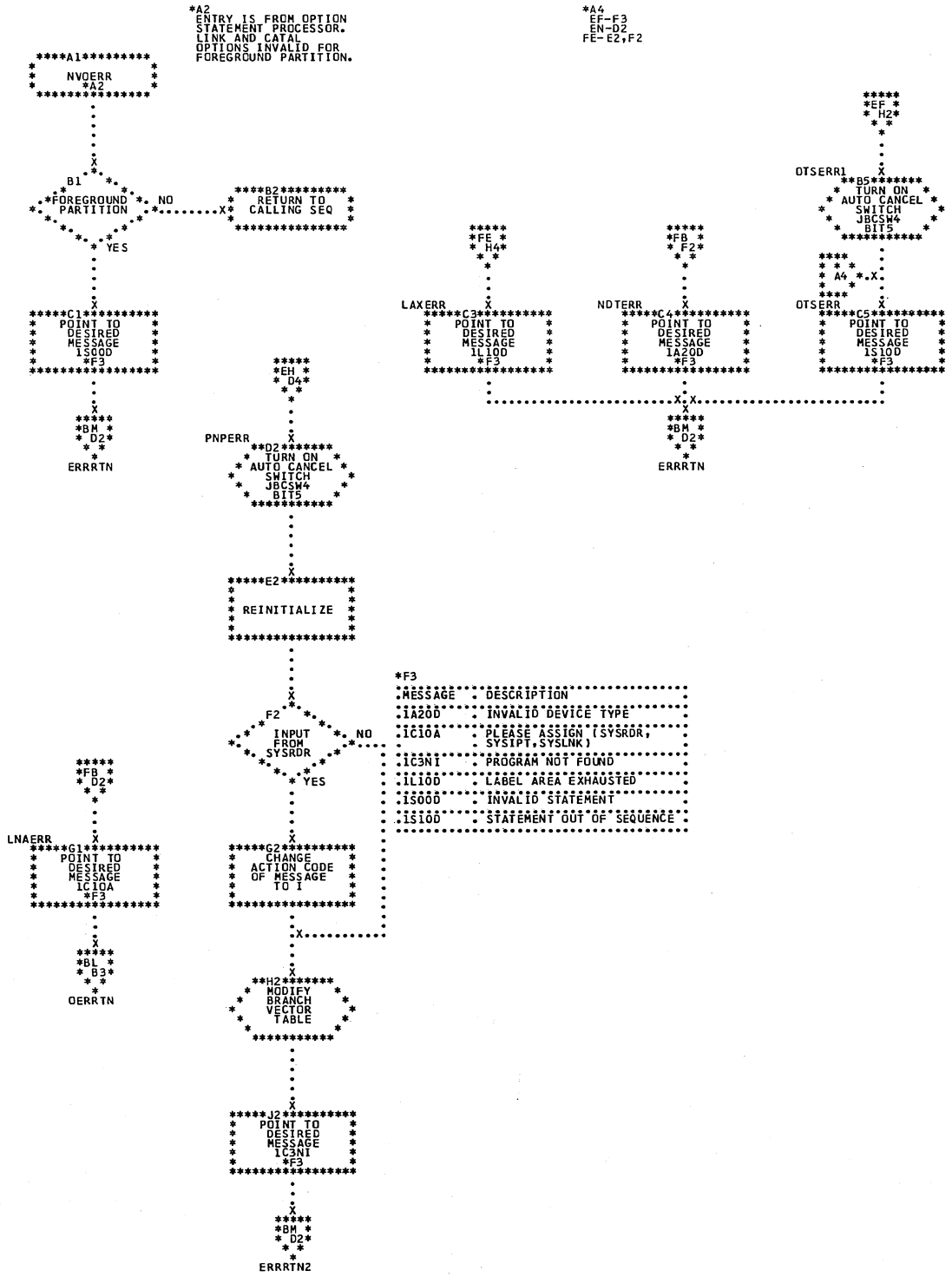


Chart GA. $JOBCTLJ - RELSE and HOLD Statement Processors
Refer to Chart 11.

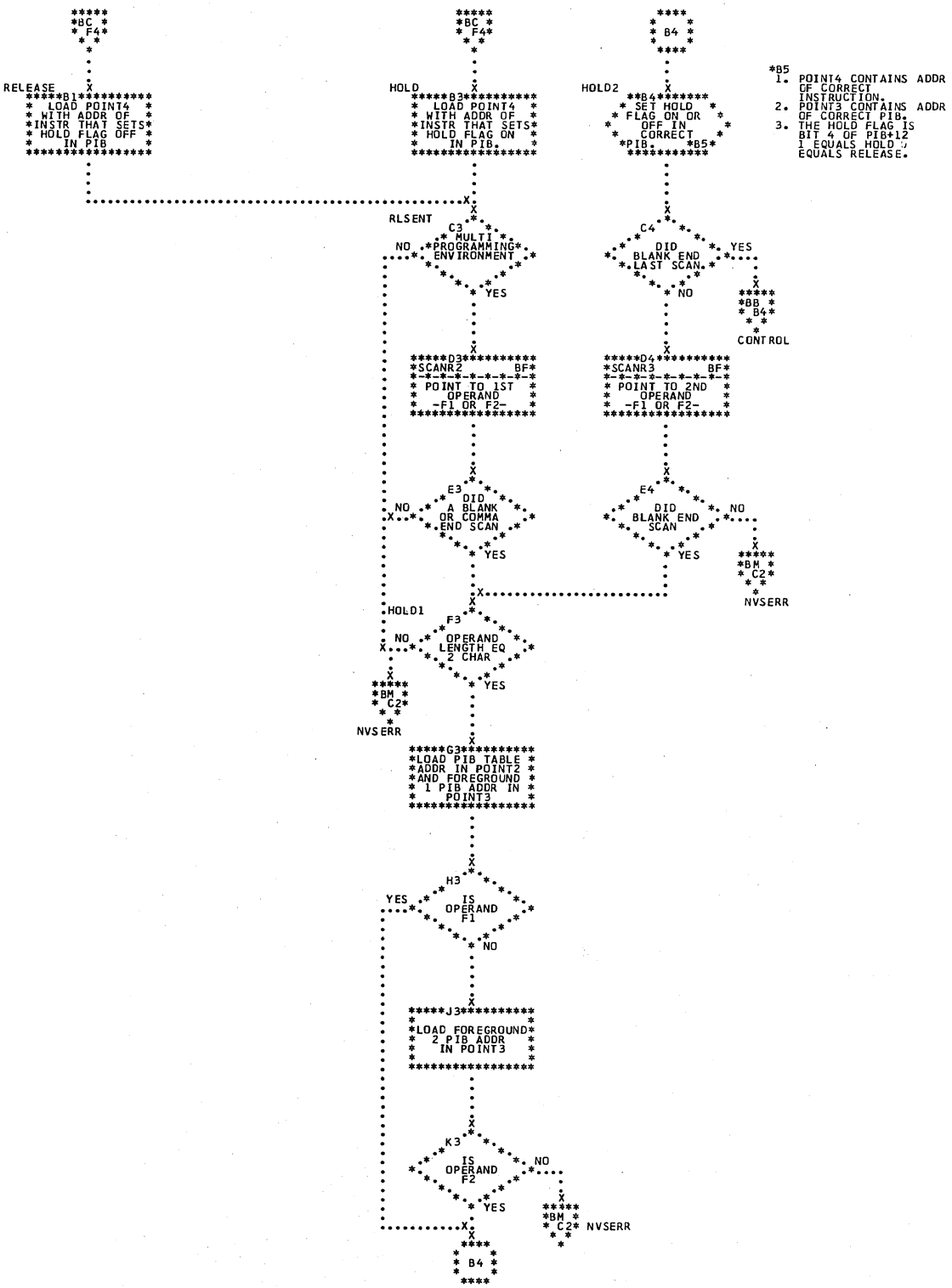


Chart HJ. $JOBCTLK - EXTENT Statement Processor (Part 1 of 3)
Refer to Chart 14.

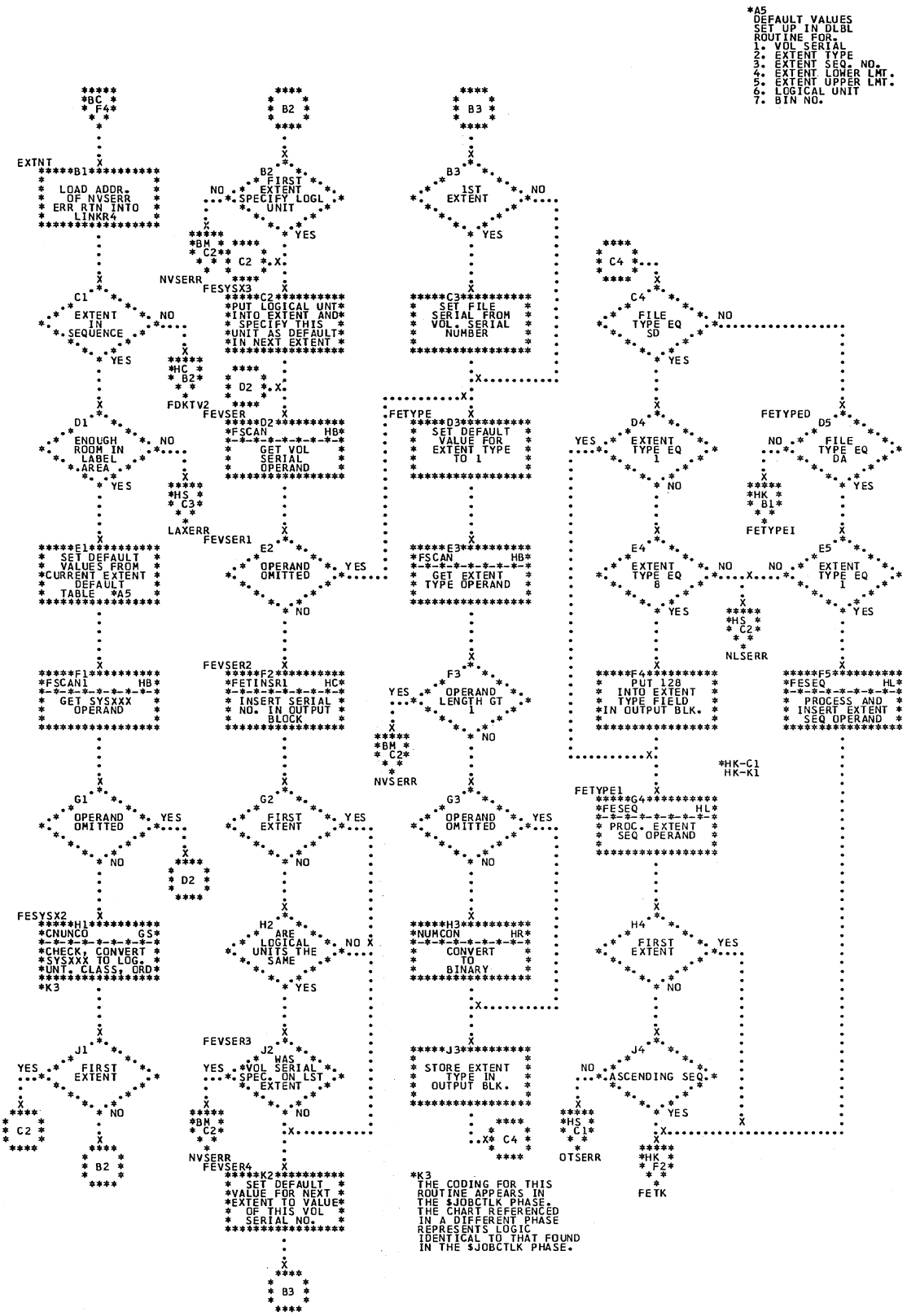
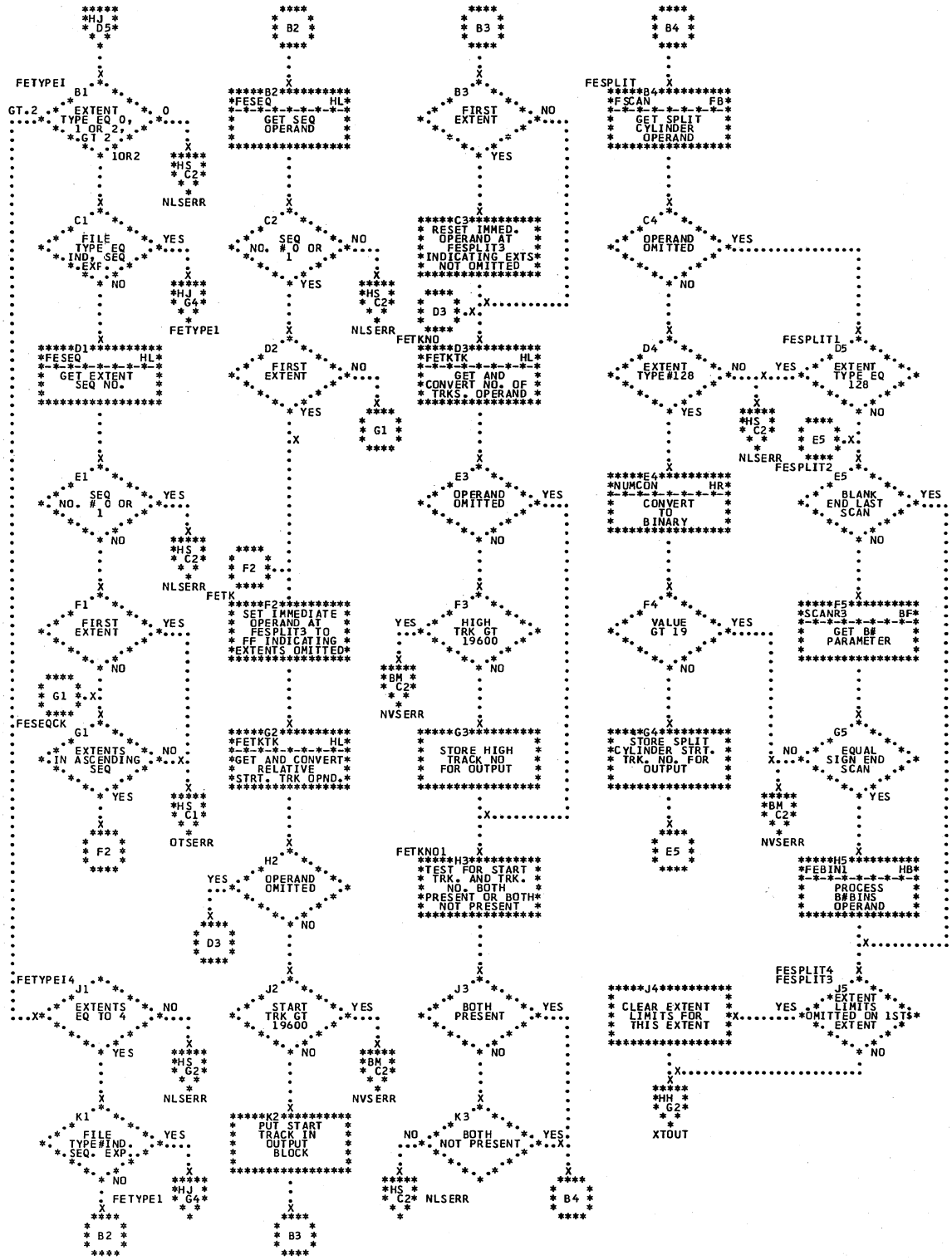
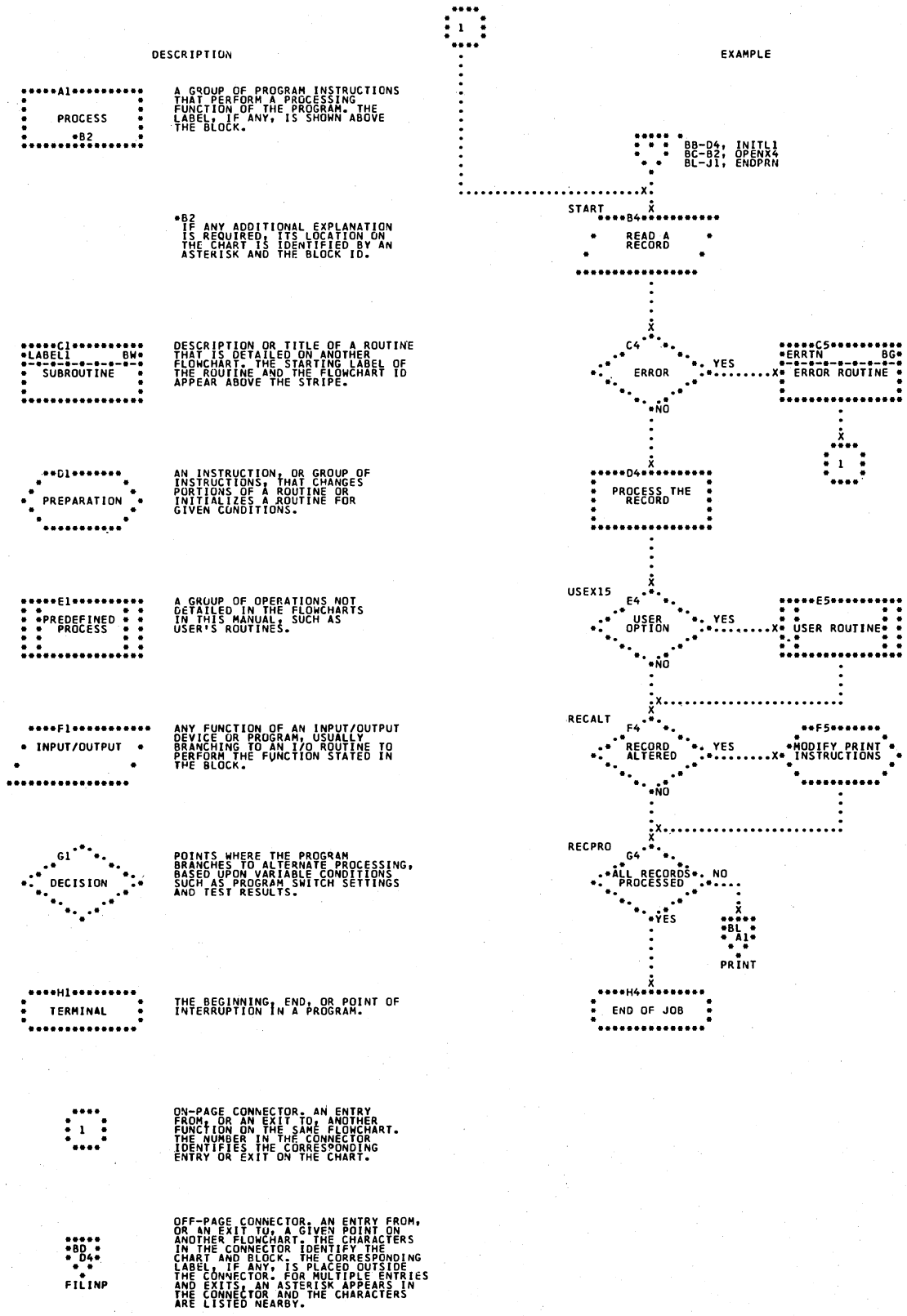


Chart HK. $JOBCTLK - EXTENT Statement Processor (Part 2 of 3)
Refer to Chart 14.



Message	Phase	Chart ID
0I00A	$$A$IPL2	AB
0I01A	$$A$IPL2	AD
0I10A	$IPLRT2	AK
0I11I	$IPLRT2	AQ
0I12I	$IPLRT3	AT
0I13I	$IPLRT3	AR
0I14I	$IPLRT3	AV
0I15I	$IPLRT3	AR
0I16A	$IPLRT4	AX
0I17A	$IPLRT4	AX
0I18A	$IPLRT2	AK
0I20I	$IPLRT4	AY
0I22I	$IPLRT4	AZC
0I23I	$IPLRT4	AZ
0I24A	$IPLRT3	AU
0I25I	$IPLRT4	AY
1A0nD	$JOBCTLD	CC,CD,CE, CG,CH,CJ
1A1nD	$JOBCTLD	CA,CB
1A2nD	$JOBCTLD	CA,CB,CH, CL,CP
1A20D	$JOBCTLG	FB
	$JOBCTLJ	GB,GE
1A3nD	$JOBCTLD	CG,CR
1A4nD	$JOBCTLD	CA,CL,CH,CW
	$JOBCTLF	DD,DF
1A40D	$JOBCTLJ	GS
	$JOBCTLK	HQ
1A5nD	$JOBCTLD	CP
	$JOBCTLF	DP
1A50D	$JOBCTLJ	GS
1A6nD	$JOBCTLD	CA
	$JOBCTLF	DM
1A7nD	$JOBCTLD	CA
	$JOBCTLF	DK
1A70D	$JOBCTLJ	GE
1A90D	$JOBCTLF	DL
1A9nD	$JOBCTLD	CE
1C00A	$JOBCTLA	BL
1C10A	$JOBCTLA	BL
	$JOBCTLG	FG
	$JOBCTLJ	GD
1C30A	$JOBCTLJ	GB,GM
1C3nI	$JOBCTLG	FG
1C70D	$JOBCTLA	BE
1C80D	$JOBCTLA	BL
1I00A	$JOBCTLA	BB
1I50I	$JOBCTLA	BL
1I70I	$JOBCTLA	BM
1I83A	$JOBCTLM	JB
1I84A	$JOBCTLM	JA,JJ
1I85A	$JOBCTLM	JD
1I86A	$JOBCTLM	JD,JF,JG,JM
1I87A	$JOBCTLM	JE
1I88I	$JOBCTLJ	GJ
1I89I	$JOBCTLM	JP,JQ
1I90D	$JOBCTLM	JP
1I91D	$JOBCTLM	JQ
1I92D	$JOBCTLM	JQ
1I93I	$JOBCTLM	JN
1L00D	$JOBCTLK	HF,HG,HJ, HK,HL,HM
1L10D	$JOBCTLG	FE
	$JOBCTLK	HG,HJ,HP
1P00D	$JOBCTLG	GQ
1S00A	$JOBCTLA through $JOBCTLM	See BM-B1
1S00D	$JOBCTLG	FG
	$JOBCTLM	JH
1IS03I	$JOBCTLJ	GN
1S10D	$JOBCTLG	EF,EN,FE
	$JOBCTLJ	GD,GN
	$JOBCTLK	HA,HC,HF,HG, HJ,HK,HP

APPENDIX C: EXPLANATION OF FLOWCHART SYMBOLS



Key to Communications Region Displacement:

56 Job control byte

Bit 0: 1 = Job Accounting Interface (JA) not supported
0 = Job Accounting Interface (JA) is supported
1: 1 = Return to caller on LIOCS disk open failure
0 = Do not return to caller on LIOCS disk open failure
2: 1 = Job control input from SYSRDR
0 = Job control input from SYSLOG
3: 1 = Job control output on SYSLOG
0 = Job control output not on SYSLOG
4: 1 = Cancel job
0 = Do not cancel job
5: 1 = Pause at end-of-job step
0 = No pause at end-of-job step
6: 1 = SYSLOG is not a 1052
0 = SYSLOG is a 1052
7: 1 = SYSLOG is assigned to the same device as SYSLST
0 = SYSLOG is not assigned to the same device as SYSLST

57 Linkage control byte

Bit 0: 1 = SYSLNK open for output
0 = SYSLNK not open for output
1: 1 = $ or FG program phase deleted, renamed, or cataloged (flag bit for $MAINEOJ)
2: 1 = Allow EXEC
0 = Suppress EXEC
3: 1 = Catalog linkage editor output
0 = Do not catalog linkage editor output
4: 1 = Supervisor has been updated
0 = Supervisor has not been updated
5: 1 = Executing in AUTOTEST mode
0 = Not executing in AUTOTEST mode
6: 1 = Reallocate or condense in progress
7: 1 = Fetch $MAINEOJ at end of job to update system directory
0 = Do not fetch $MAINEOJ at end of job for update

58 Language processor control byte. This is a set of switches used to specify nonstandard language translator options. The switches within the byte are controlled by job control OPTION statements and when set to 1, override standard options. The format of this byte is identical to the stnadard option byte (displacement 54) with one exception: Bit 7 in this byte is used to indicate to LIOCS that the rewind and unload option has been specified.

59 Job duration indicator byte

Bit 0: 1 = Within a job condition
0 = Outside a job condition
1: 1 = Dump on an abnormal end-of-job condition
0 = No dump on abnormal EOJ
2: 1 = Pause at EOJ step
0 = No pause at EOJ
} Set by Attention Routine for Job Control
3: 1 = Job control output on SYSLST
0 = Output not on SYSLST
4: 1 = Job is being run out of sequence with a temporary assignment for SYSRDR
0 = Conditions for 1 setting not met
5: 1 = PCIL is being condensed
0 = PCIL is not being condensed
6: Reserved
7: 1 = Batch command just issued
0 = Condition for 1 setting did not occur

Figure 10. Supervisor Communications Region (Part 4 of 5)

Key to Communications Region Displacements:

60 Binary disk address of the volume label area (label cylinder).

62 → 76 As illustrated (Figures for information blocks, I/O tables, and pointers begin at Figure 21 which refers to more detailed Figures).

78 Set to the value nn specified in the LINES = nn parameter of the STDJC macro.

79 The format of the system date contained within this field is determined by the IPL program from information supplied in the date convention byte (displacement 53). Bytes 85-87 contain the day count.

88 Bytes reserved for use by LIOCS. Transient dump programs insert a key to indicate to the LIOCS end-of-volume routine, $$BCMT07, that it was called by a B-transient.

90 Address of the first part of the program information block (PIB) table. (See Figures 18 and 19).

92 ID number of the last checkpoint. Temporary indicator of file protected DASD. Used at IPL time, when DASDFP is specified.

94 Length of the LUBID queue (in bytes). This equals the number of channel queue entries. It can also be used to access the REQID, LUBDSP, and TKREQID queues: (See GY24-5151).

96 Address of disk I/O position data. This is the starting address of the disk information block (DIB) table (See Figure 17).

98 Address of the beginning of the error recovery block. The error recovery block contains addresses of error recovery exits, error recovery queue information that can be used by physical transients routines, and defines storage for the error queue entries (See Figure 43 - GY24-5151).

100 → 104 Option Tables. (See GY24-5151).

106 Key of the program (BG, F2, or F1) that has timer support.

108 Address of LUBID queue. (See GY24-5151).

110 Logical Transient Key (LTK) contains the same value as the PIK (PID) (Displacement 46) when the logical transient is requested. When the transient area is not in use, LTK is equal to zero. The SVC 2 routine sets the LTK. The SVC 11 routine resets the LTK.

112 Supervisor constants:

DOLLARBO	(4 bytes) = C'$$BO'
SSKADR	(5 bytes) = XL5'0'
LTAREA	(3 bytes) = Adcon of LTSVPT, logical transient save pointer

124 Address of second part of program information block (PIB) table (See Figure 20).

126 Address of PDTABB, table of DTF addresses for MICR support.

128 Address of QTAM vector table (IJLQTTAD).

132 Address of background communications region.

134 Option Indicator Byte

Bit 0: 1 = MCRR indicated for OBR writer
0 = No MCRR indicated for OBR writer
1: 1 = EU interface active
0 = EU interface not active
2: 1 = TP request
0 = No TP request
3: 1 = Supervisor support for only 9-track tape
0 = Supervisor does not support 9-track tape exclusively
4: Reserved
5: 1 = RETAIN/370 support generated
0 = RETAIN/370 support not generated
6-7: Reserved

135 System Configuration Byte 2

Bit 0: 1 = PCIL supported
0 = PCIL not supported
1-7: Reserved

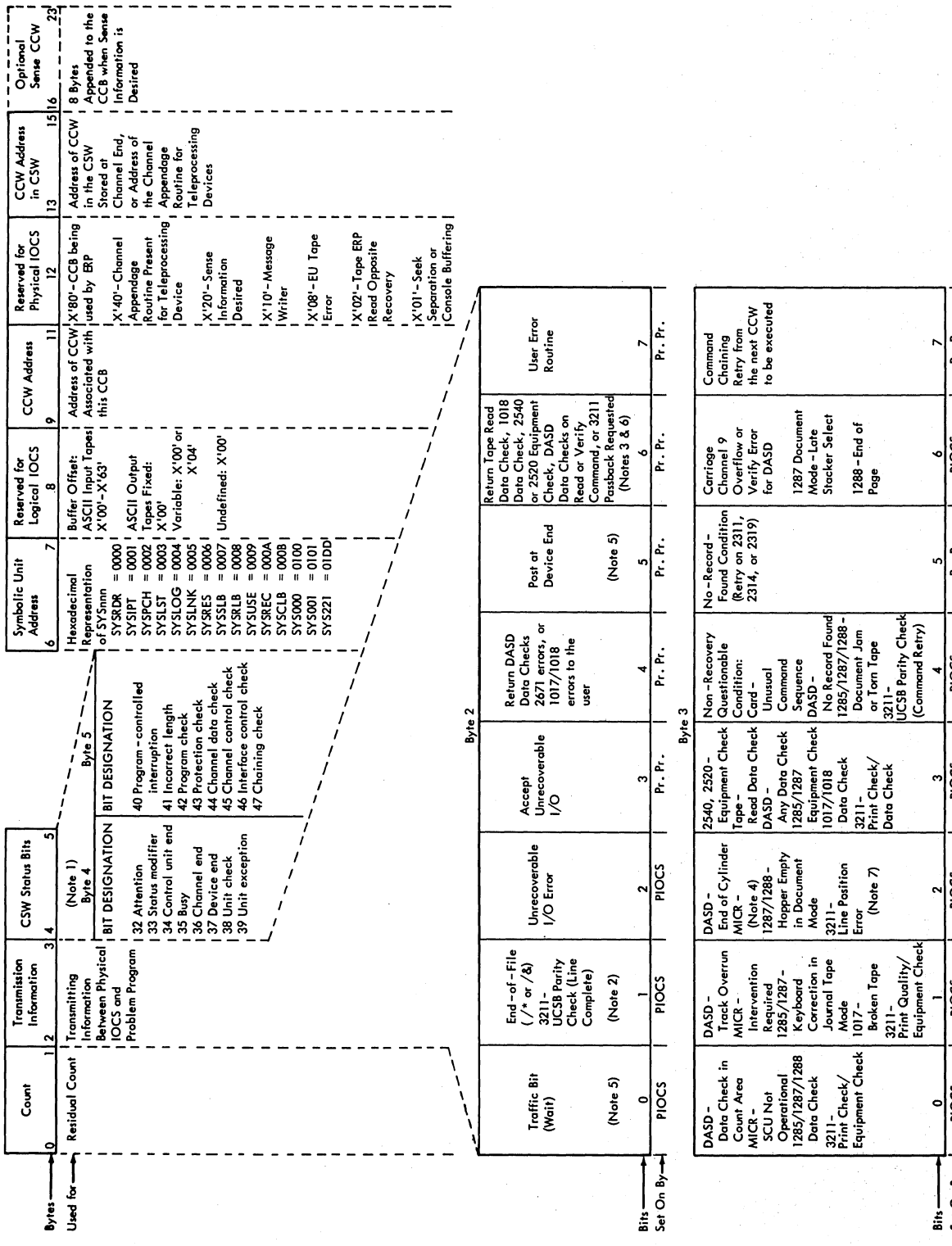
136 Pointer to communications region extension (See Figure 11).

Figure 10. Supervisor Communications Region (Part 5 of 5)

Key to SDR Communications Region Displacements:

32	Address of SDR accumulator area which contains half-byte counters and accumulated error conditions.
36	Address of SDR unit switches. SDR switch byte (1 for each PUB): X'80' - Update operations complete X'40' - Counters on external file overflowed X'20' - I/O error during write X'08' - SDR update half-byte counters routine required X'04' - Update SDR record routine required Other - Reserved When entry contains X'01000000', indicates MCRR, no SDR supported.
40	Reserved.
44	SDR1 register save area.
72	Mask formats for interpretive error accumulator, SDR1: X'FF' - End of update X'FE' - Bypass counter X'FD' - Set up 'OR' condition to previous counter X'FC' - Ignore list item Other - Test bit in error queue
76	Used by the interpretive error accumulator routine to process list passed by OBR/SDR A-transient.
96	Used by the interpretive error accumulator routine.
104	Used by the interpretive error accumulator routine for address alignment.
108	Executed by the interpretive error accumulator routine.
112	Loop counter for the SDR counter update.
116	Save area for pointers to entries in the SDR error queue.
118	Work area where half byte error counters are unpacked and updated.
136	List of devices passed to the SDR processor and $$ANERAD.
156	Used by SDR/OBR recorder phases to pass error message displacements and disk error addresses in event of an error.
160	Entry point from OBR/SDR A-transients. Branches to label SDRMM.
164	Pointer into the OBR/SDR unit switches. Status posted by recorder phases. (See byte 36).
168	OBR and SDR records formatted by the recorder phases.

Figure 12. SDR Communications Region - SDRTABLE (Part 2 of 2)



Note 1. Bytes 4 and 5 contain the status bytes of the Channel Status Word (Bits 32-47). If byte 2, bit 5 is on and device end results as a separate interrupt, device end will be ORed into CCB byte 4.
Note 2. Indicates /* or /& statement encountered on SYSRDR or SYSIPT. Byte 4, bit 7 (unit exception) is also on.
Note 3. DASD data checks on count not returned.
Note 4. For 1255/1259/1270/1275/1412/1419, disengage. For 1275/1419D, I/O Error in external interrupt routine (channel data check or busout check).
Note 5. The traffic bit (Byte 2, bit 0) is normally set on at channel end to signify that the I/O was completed. If byte 2, bit 5 has been set on, the traffic bit and bits 2 and 6 in byte 3 will be set on at device end. Also see Note 1.
Note 6. 1018 ERP does not support the Error Correction Function.
Note 7. This error occurs as an equipment check, data check, or FCB parity check.

Figure 13. Command Control Block (CCB)

PIB TABLE

Byte Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	= 16 Byte Length
All Bound PIB	Flag Byte See A *	Reserved	SP Prefix		Branch Instruction to the All Bound Routine							Reserved					
Problem Program PIB (Note 1)	Flag Byte See B *	Cancel Code **	SYSLOG ID (BG, F2, or F1)		NOP Instruction (CR)	Address of the Partition Save Area			Number of Core Blocks (Note 2)	Address of the Origin of the Partition			PIB Assign Flag See D *	User LUB Index	Number of Program LUBs	Flag Byte See C *	
Attention PIB	Flag Byte See E *	Cancel Code **	SYSLOG ID (AR)		Branch Code (BC)	Active=Address of Save Area Inactive=Remainder of BC Instruction			Switch Byte See F *	Logical Transient Bucket (contains save area address)			X'07' See D *	Reserved	Address of the Logical Transient		
Quiesce PIB	Flag Byte See A *	Cancel Code **	C'/&'		Branch Instruction to Quiesce I/O Routine				Scratch Byte X'00'	Channel PUB Table Index Values X'00'	X'04'	X'08'	X'0C'	X'10'	X'14'	X'18'	
Supervisor PIB	Flag Byte See A *	Cancel Code **	SP Prefix		Branch Instruction to General Exit Routine				Address of SYSRES PUB		Length of Error Queue Entry		Constants to Clear Bytes 2-5 of CCB X'1F'	X'05'	X'00'	X'00'	
Subtask PIB for AP (Note 3)	Flag Byte See B *	Cancel Code **	SYSLOG ID (BG, F2, or F1)		NOP Instruction	Address of the Save Area			Number of Core Blocks (Note 2)	Address of the Origin of the Main Task			PIB Assign Flag See D *	User LUB Index	Number of LUBs	Flag Byte See C *	

Note 1: Three problem program PIBs are built in this sequence when the MPS or BJF feature is selected as a generation option: { Background PIB, Foreground 2 PIB, Foreground 1 PIB }

When a batch-only environment is established at generation time, the All Bound and Foreground PIBs are excluded from the table, and only one (BG) problem program PIB is built. However, the X'20' bytes that F2 and F1 PIBs normally occupy (between PIBBG and PIBAR) are filled with 32 bytes of DIBs data.

Note 2: Number is in multiples of 2K for F2 and F1. BG is always 10K (X'0A').

Note 3: Total of nine subtask PIBs are generated, and only when AP is specified at generation time.

* See Figure 16 for flag byte expansions A, B, C, D, E and F.

** See Supervisor PLM, GY24-5151

Bytes 90 and 91 (X'5A'-'5B') of the communications region contain the address of the first part of the PIB Table. Label PIBTAB identifies the first byte of the table.

Figure 18. First Part of PIB Table

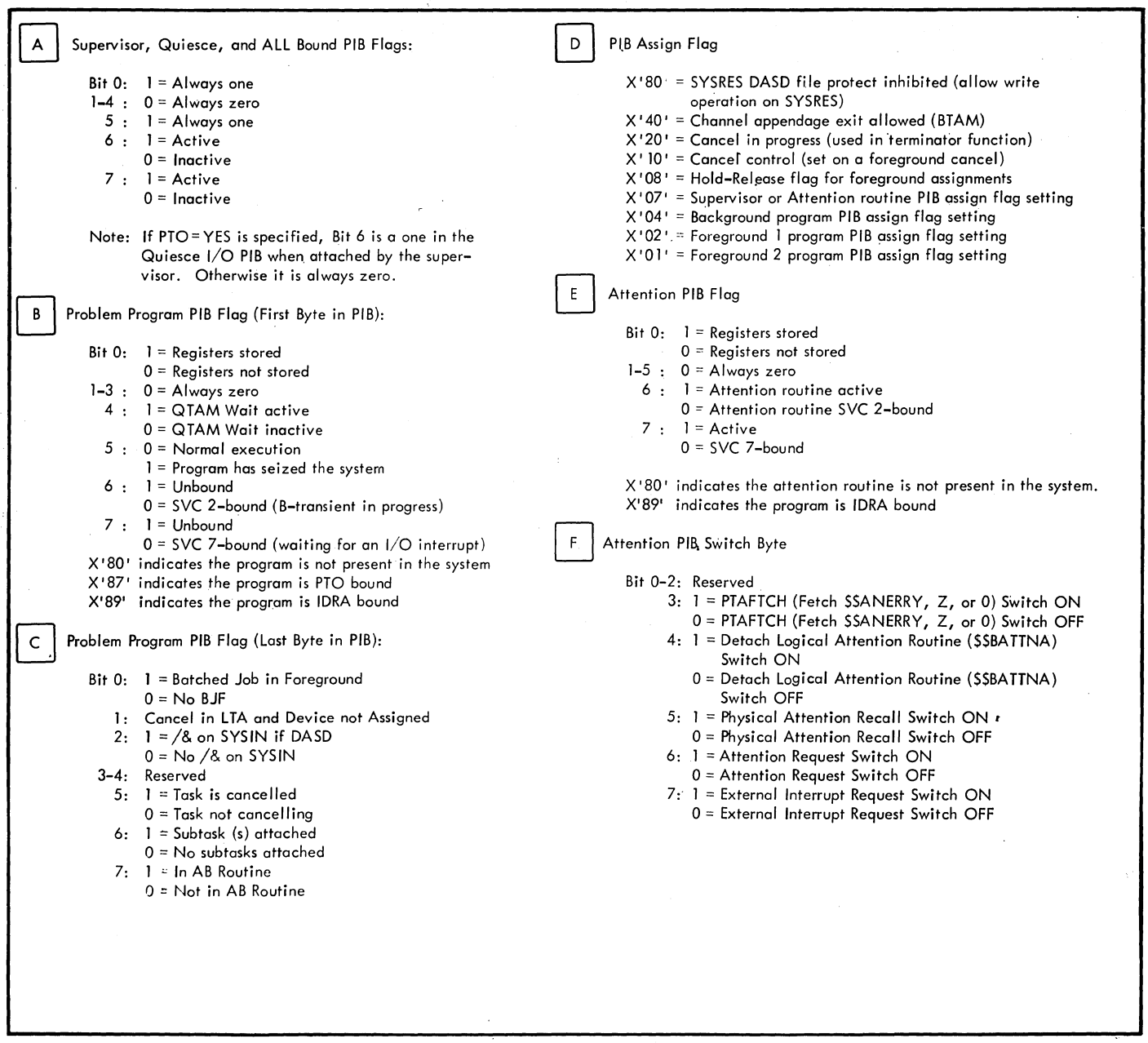


Figure 19. PIB Flag Expansions

Note 1: The TEBV (Tape Error Block by Volume) Table is composed of one Status Block and (n) Error Blocks, and is addressed symbolically by label TEBVTAB.

Supervisor generation options in the FOPT macro determine the size of the TEBV Status Block at generation time:

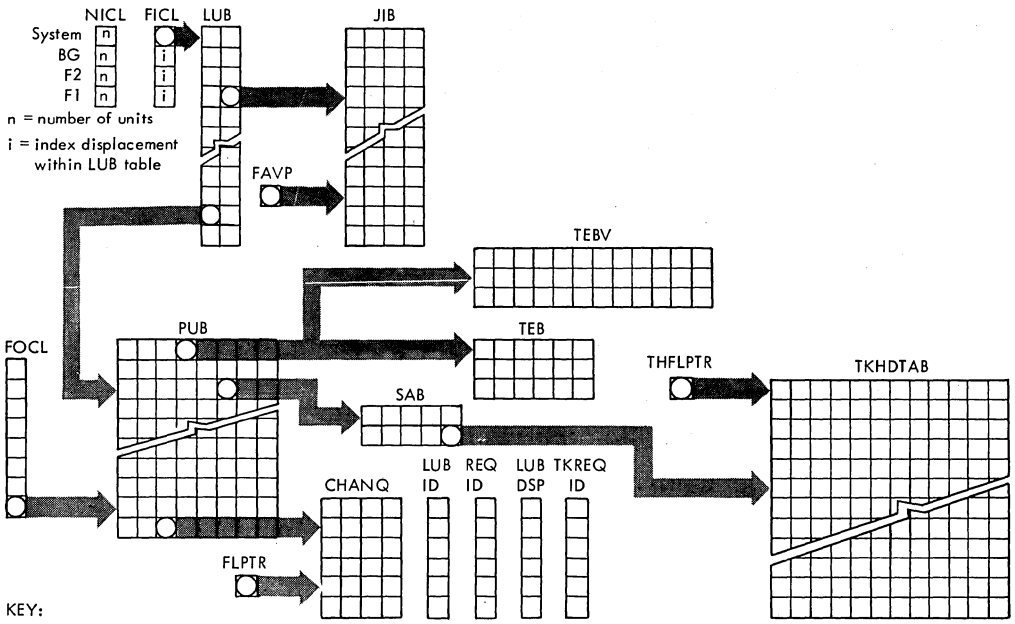
- When EVA is chosen without ESTV, the TEBV Status Block is four bytes long (bytes 0-3), followed by TEBV Error Blocks, so that bytes 4-21 are omitted.

- When ESTV output is to SYSLOG, the TEBV Status Block is six bytes long (bytes 0-5), followed by TEBV Error Blocks, so that bytes 6-21 are omitted.

- When ESTV output is to DASD, the TEBV Status Block is 22 bytes long (bytes 0-21, such as shown in this Figure), followed by TEBV Error Blocks.

Note 2: The number of TEBV Error Blocks generated corresponds to the (n) parameter of the FOPT macro for TEB, TEBV, or EVA options. A TEBV Error Block always contains 18 bytes, as shown in bytes 22-39 of this Figure. Therefore, the TEBV Table is composed of one TEBV Status Block (with its byte length dependent on supervisor generation options, as described in Note 1), followed by (n) number of 18-byte TEBV Error Blocks.

Figure 21. TEBV Table Showing Status Block and Error Blocks (Part 2 of 2)



KEY:

NICL (Number in Class) : The first byte contains the number of system class units. The second, third, and fourth bytes contain the number of programmer class units (BG, F2, F1) (Figure 15).

FICL (First in Class) : The first byte points to the first system class unit in the LUB table. (Always the first LUB table entry.) The second byte points to the first programmer class unit in the LUB table BG area. The third points to the first programmer class unit in the LUB table F2 area. The fourth points to the first programmer class unit in the LUB table F1 area (Figure 15).

LUB (Logical Unit Block) Table : The first byte points to a PUB table entry (if the logical unit is assigned) or contains X'FF'. The second byte points to a JIB table entry or contains X'FF' (Figure 15).

PUB (Physical Unit Block) Table : The first two bytes contain the channel and unit address of the physical device; the third a CHANQ pointer; the fourth a TEB pointer; the fifth device type codes; the sixth a device characteristic code or a SAB pointer; the seventh the channel scheduler flag; and the eighth has the job control flag (See Figure 14).

FOCL (First on Channel List) : The first byte points to the first PUB (highest priority) on channel zero. The next byte points to the first PUB (highest priority) on channel one, etc. A hexadecimal FF indicates the associated channel is not supported.

TEB (Tape Error Block by Unit) : One TEB is built for each tape unit at supervisor generation time if tape error statistics by unit are required (Figure 16).

TEBV (Tape Error Block by Volume) : One TEBV is built for each tape unit at supervisor generation time if tape error statistics by volume are required (Figure 21).

FAVP (First Available Pointer) : A one-byte pointer to the next available JIB entry.

JIB (Job Information Block) : The first two bytes contain extent or LUB information. The third contains ownership and JIB flags. The fourth contains JIB chaining information (See Figure 23).

CHANQ (Channel Queue) Table : The first byte contains the chain field (a pointer to the next in queue). The last three bytes contain the CCB address.*

LUBID (LUB Identification) : A one-byte pointer to the LUB making the I/O request.*

REQID (Requestor Identification) : A one-byte pointer to the program containing the CCB.*

LUBDSP (LUB Displacement) : A one-byte value equal to the absolute LUB number (CCB byte 7).*

FLPTR (Free List Pointer) : A one-byte pointer to the next free entry in the channel queue.*

SAB (Seek Address Block) : A four-byte (BCCH) address that is the current disk address of the device plus a fifth byte that contains a Track Hold Table pointer of X'FF'. If the Track Hold function is not supported, the fifth byte contains X'00'.

TKHDTAB (Track Hold Table) : The first byte contains a pointer to the next available entry (or X'FF'); bytes 2 - 4 have CCB address of the requesting task; bytes 5 - 10 have a disk address (BBCCHH) of track being held; byte 11 has key of owning track; and byte 12 has two uses: bit 0=1 means a task is waiting for the track, and bits 4 - 7 count the number of holds on the track.* Note: The number of holds is one more than the value of bits 4-7 of the last byte.

THFLPTR (Track Hold Free List Pointer) : A one-byte pointer to the next free entry in the Track Hold Table.*

TKREQID (Track Requestor Identification) : A one-byte pointer to the PIB of the task requesting I/O.*

*See Supervisor PLM, GY24-5151

Figure 22. I/O Table Interrelationship

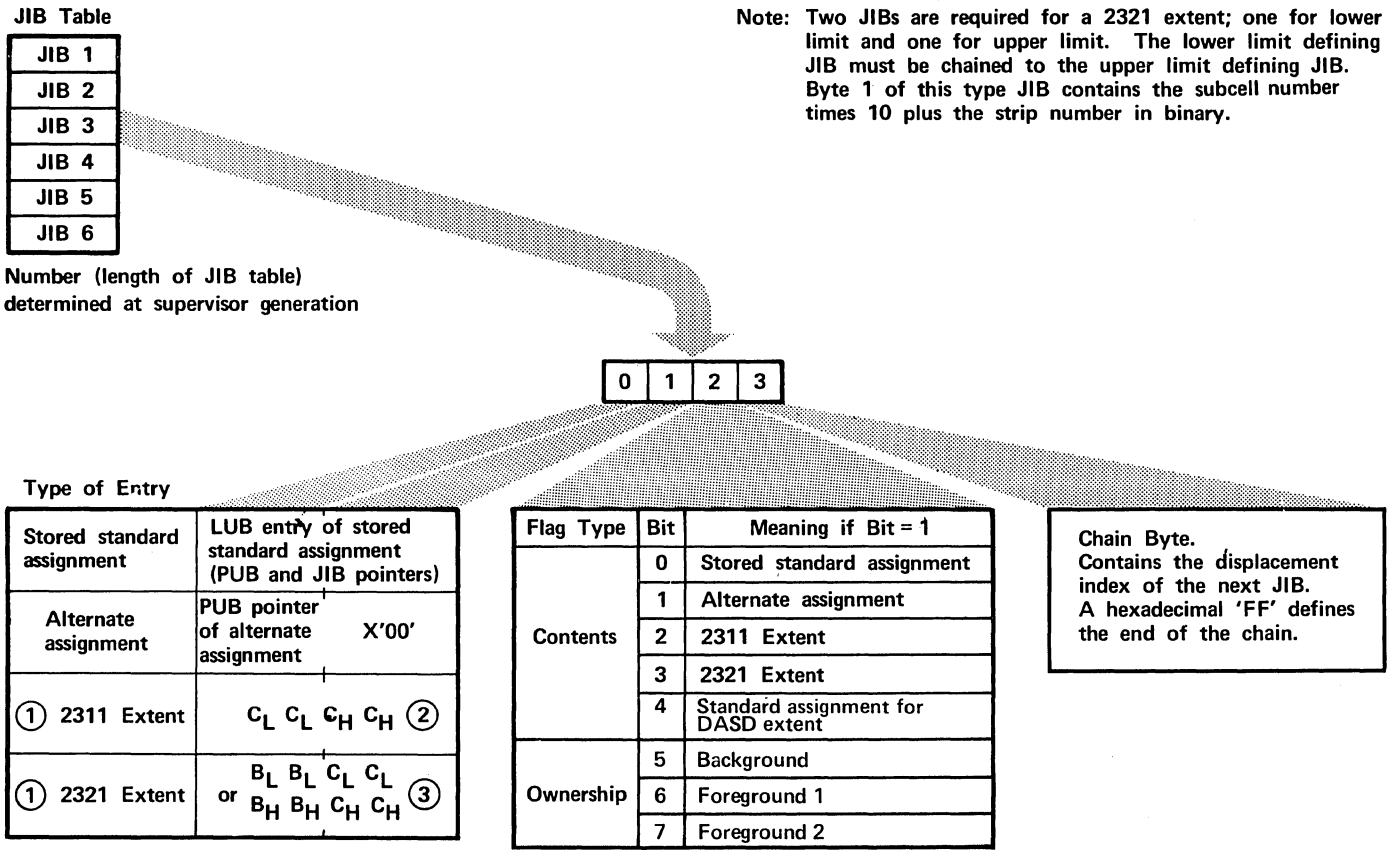


Figure 23. Job Information Block (JIB) Table

Displacement	Label	Description
0-3	ACCTWK1* (ACCTABLE)*	Work area used in SIO update.
4-7	ACCTWK2	Work area used with ACCTWK1 in start/stop time routine
8-11	ACCTSVPT	Job card pointer; address of job card field following jobname.
12	ACCTPART	ID of partition in charge (partition switch name).
13	ACCTRES2	Reserved.
14-15	ACCTLEN	Length of SIO area=6n+1, where n=number of devices for this partition in SYSGEN option JA=(n1, n2, n3).
16-21	ACCTLOAD	Label area instruction; moves JAI label area address to OPEN/CLOSE transients.
22-23	ACCTRES3	Reserved.
24-27	ACCTLADD	Address of alternate label area.
28-31	ACCTCPUT	Counter for CPU time elapsed in a jobstep, counted in 300ths of a second.
32-35	ACCTOVHT	Counter for overhead time; time not charged to any partition.
36-39	ACCTBNDT	Counter for all-bound time; system wait state time divided between running partitions.
40-47	ACCTSVJN	Save area for job name during simulated EOJ.
		JOB ACCOUNTING TABLE (user's portion of Partition Table)
48-55	ACCTJBNM	Job name; taken from job card.
56-71	ACCTUSRS	User information; 16 bytes from Job card.
72-73	ACCTPTID	Partition ID; 'BG', 'F2', or 'F1' in EBCDIC format.
74	ACCTCNCL	Cancel code; see Cancel Codes and Messages (in Supervisor and Related Transients, GY24-5151).
75	ACCTYPER	Type of record: 'S'=job step, 'L'=last step of job.
76-83	ACCTDATE	Date in format specified at SYSGEN (MM/DD/YY or DD/MM/YY).
84-87	ACCTSTRT	Start time of job, in packed decimal (OHHMMSSF; F=sign).
88-91	ACCTSTOP	Stop time of job, in same format as ACCTSTRT
92-95	ACCTRES	Reserved.
96-103	ACCTEXEC	Phase name; taken from execute card.
104-107	ACCTHICR	High core address of active program phase, from COMREG.
108-111	ACCTIMES	CPU time elapsed in a job step; counted in 300ths of a second.
112-115	------	Overhead time; elapsed time not charged to any partition, in 300ths of a second.
116-119	------	All-bound time; system wait state time divided between running partitions, in 300ths of a second.
120	ACCTSIOS	SIO tables: 6 bytes for each device specified by SYSGEN options, as follows: 2 bytes for device address (Ocuu), 4 bytes for count of SIOs in current jobstep.
———	------	Overflow byte: normally X'20', but is X'30' if more devices are used within a partition than specified by SYSGEN options.

*Note: DSECT ACCTABLE symbolically addresses the JAI Partition Tables with labels as shown. Each partition in which JAI is supported has its own JAI Partition Table, labeled ACCTBG, ACCTF2, ACCTF1, for active partitions BG, F2, and F1 respectively.

Figure 24. Job Accounting Interface Partition Table

DOS IPL and Job Control

READER'S COMMENT FORM

GY24-5086-4

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	Accuracy	Index	Illustrations	Appearance	Paper
Organization of the text	Cross-references	Tables	Examples	Printing	Binding

GY24-5086-4

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.

Fold

Fold

FIRST CLASS
PERMIT NO. 1359
WHITE PLAINS, N. Y.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM Corporation
112 East Post Road
White Plains, N.Y. 10601

Attention: Department 813 U

Fold

Fold

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
[U.S.A. only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

CUT ALONG THIS LINE

DOS IPL and Job Control (S360-36) Printed in U.S.A. GY24-5086-4

DOS IPL and Job Control (S360-36) Printed in U.S.A. GY24-5086-4

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
[U.S.A. only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]