IBM System/3
Disk Utility Programs
Logic Manual

Program Number 5702-SC1
5703-SC1

Designed to satisfy requirements of support personnel responsible for maintenance of the IBM System/3 Disk Utility Programs, whether they are used with the IBM System/3 Disk System or the IBM System/3 Model 6.

The manual describes the internal logic of these utility programs:

- Disk Initialization
- Alternate Track Assignment
- Alternate Track Rebuild
- File and Volume Label Display
- File Delete
- Disk Copy/Dump
- Library Maintenance

This manual is intended for use in locating specific areas within a program and relating these areas to the corresponding program listings.

Second Edition (March 1971)

This is a major revision of, and obsoletes, SY21-0503-0. The revisions have been extensive throughout the manual. The manual is now designed to satisfy requirements of support personnel responsible for maintenance of the IBM System/3 Disk Utility programs used with the IBM System/3 Model 10 Disk System or the IBM System/3 Model 6.

This edition applies to Version 3, Modification 0 of the IBM System/3 Disk Utility programs, and to all subsequent versions and modifications, 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 to the IBM Systems, consult the latest IBM System/3 Newsletter, Order Number GN20-2228 for the editions that are applicable and current. (Although this is the first release of IBM System/3 Model 6, it is identified as version 3, modification 0 for purposes of compatibility with the Model 10 Disk System.

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 reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Programming Publications, Department 425, Rochester, Minnesota 55901.

Preface

This Program Logic Manual is designed to satisfy requirements of support personnel responsible for maintenance of the IBM System/3 Disk Utility programs used with the IBM System/3 Disk System or the IBM System/3 Model 6. The manual describes the internal logic of these utility programs:

- Disk Initialization
- Alternate Track Assignment
- Alternate Track Rebuild
- File and Volume Label Display
- File Delete
- Disk Copy/Dump
- Library Maintenance

This publication is intended for use in locating specific areas within a program and relating these areas to the corresponding program listings.

RELATED PUBLICATIONS

These publications are recommended for additional reference:

Disk System

- *IBM System/3 Card and Disk System Components Reference Manual,* GA21-9103.
- *IBM System/3 Disk Systems Data Management and Input/Output Supervisor Logic Manual,* SY21-0512.
- *IBM System/3 Disk Systems System Control Program Logic Manual,* SY21-0502.
- *IBM System/3 Disk System Operation Control Language and Disk Utilities Reference Manual,* GC21-7512.

Model 6

- *IBM System/3 Model 6 Components Reference Manual,* GA34-0001.
- *IBM System/3 Disk Systems Data Management and Input/Output Supervisor Logic Manual,* SY21-0512.
- *IBM System/3 Disk Systems System Control Program Logic Manual,* SY21-0502.
- *IBM System/3 Model 6 Operation Control Language and Disk Utility Programs Reference Manual,* GC21-7516-1.

Contents

PART I. DISK INITIALIZATION 1-1

SECTION 1. INTRODUCTION 1-3
System Requirements 1-3

SECTION 2. METHOD OF OPERATION 1-5

SECTION 3. PROGRAM ORGANIZATION 1-7
Root Phase ($INIT) 1-8
Control Statement Analysis Phase ($INIT1) 1-8
Mainline Phase ($INIT2) 1-9
Surface Analysis Phase ($INSUR) 1-10
Print Phase ($INMSG) 1-11

SECTION 4. DATA AREA FORMATS 1-21
Control Blocks 1-21
Tables 1-22
ICSTL Table 1-22
ISCTBK Table 1-22
ISVBG Table 1-22
ISSCAN Table 1-23
ISUNTB Table 1-23
ISVLTB Table 1-23
IERTAB Table 1-24
IMSGT Table 1-24
ISSYSM Table 1-24
ISCTLO Table 1-24
ALTIN Table 1-25
TBLE Table 1-25
Buffers 1-25
ISIO Read Buffer 1-25
ISSCOT Output Buffer 1-25
IOADDR Write Buffer 1-25
IBFAD Buffer Address Area 1-25
VOLS Write Buffer 1-25
Work Areas 1-26
FCS Work Area 1-26
RIDFCS Work Area 1-26
IRCDE Return Code 1-26
IOSV Save Area 1-26

PART II. ALTERNATE TRACK ASSIGNMENT . . . 2-1

SECTION 1. INTRODUCTION 2-3
System Requirements 2-3

SECTION 2. METHOD OF OPERATION 2-5
Conditional Assignment 2-5
Unconditional Assignment 2-5
Unconditional Cancel Of Prior Assignment 2-6

SECTION 3. PROGRAM ORGANIZATION 2-9
Root Phase ($ALT) 2-10
Control Statement Analysis Phase ($ALT01) 2-10
Mainline Phase ($ALT02) 2-11
Data Transfer Phase ($ALT03) 2-12
Surface Analysis Phase ($ALSUR) 2-12
Print Phase ($ALMSG) 2-13
Sector Print Phase ($ALSEP) 2-13

SECTION 4. DATA AREA FORMATS 2-25
Control Blocks 2-25
Tables 2-26
ASCTBK Table 2-26
ABER Table 2-26
ASSC Table 2-26
ASCTAB Table 2-26
AMTAB Table 2-26
ASLTA Table 2-27
ASSMG Table 2-27
ASOTLY Table 2-27
ALTIN Table 2-27
ATBLE Table 2-27
ATPDL Table 2-28
AMSGT Table 2-28
ASTAB Table 2-28
Buffers 2-28
ABFAD Data Buffer Address Area 2-28
IOADDR Buffer 2-28
ASSCOT Buffer 2-28
ASIOAR Buffer 2-28
Codes 2-28
AATCDE Track Activity Code 2-28
ARCDE Buffer 2-29
ARDTC Buffer 2-29
Work Areas 2-29
FCS Work Area 2-29
RIDFCS Work Area 2-29
PWK1 Work Area 2-29

PART III. ALTERNATE TRACK REBUILD 3-1

SECTION 1. INTRODUCTION 3-3
System Requirements 3-3

SECTION 2. PROGRAM ORGANIZATION 3-5
Mainline Phase ($BUILD) 3-5

SECTION 3. DATA AREA FORMATS 3-9
Control Blocks 3-9
RIOB Control Block 3-9
Tables 3-10
RA Conversion Table 3-10
RTAB Parameter Table 3-10
RADDS Address Table 3-10
Buffers And Work Areas 3-10
RCARD Input Buffer 3-10
RBUFFO Disk Input/Output Area 3-10
RSYOB Output Buffer 3-10

PART IV. FILE AND VOLUME LABEL DISPLAY . . 4-1

SECTION 1. INTRODUCTION 4-3
System Requirements 4-3

SECTION 2. METHOD OF OPERATION 4-5

SECTION 3. PROGRAM ORGANIZATION 4-7
Root and Syntax Scan Phase ($LABEL) 4-8
Mainline Phase ($LABL1) 4-8

SECTION 4. DATA AREA FORMATS 4-13
Tables 4-13
FRCTBK Table 4-13
FSDTB Table 4-13
FSSCAN Table 4-13
FSFLG Table 4-13
FSJWTB Table 4-14
FSCTLO Table 4-14
FSSYSM Table 4-14
FRMNTB Table 4-14
FSSCTB Table 4-15
FVPARM Table 4-15
FDOTL Table 4-15
FDHSP Table 4-15
FDLIST Table 4-15
Buffers 4-16
FSBUF Input Buffer 4-16
FDBUF Print Buffer 4-16
FIOA Read Buffer 4-16
FIOA2 Read Buffer 4-16
FSSCOT Scan Output Area 4-16

PART V. FILE DELETE 5-1

SECTION 1. INTRODUCTION 5-3
System Requirements 5-3

SECTION 2. METHOD OF OPERATION 5-5

SECTION 3. PROGRAM ORGANIZATION 5-7
Root Phase ($DELET) 5-8
Control Statement Analysis Phase ($DELT1) 5-8
Mainline Phase ($DELT2) 5-9

SECTION 4. DATA AREA FORMATS 5-15
Tables 5-15
FRCTBK (FSCTBK) Table 5-15
FSSTB Table 5-15
FSSCAN Table 5-15
FSFLG Table 5-16
FSJWTB Table 5-16
FVTAB Table 5-16
FMOTL Table 5-16
FPRINT Table 5-16
PARMA Table 5-16
PARMB Table 5-17
FMHSP Table 5-17
FSHTB Table 5-17
FRMNTB Table 5-17
FSSCTB Table 5-17
FSCTLO Table 5-18
FSSYSM Table 5-18
Buffers 5-18
FSBUF Input Buffer 5-18
FSSCOT Output Buffer 5-18
FVIO I/O Buffer 5-18
FFIO I/O Buffer 5-18
Work Areas 5-18
FSWRK Work Area 5-18

PART VI. DISK COPY/DUMP 6-1

SECTION 1. INTRODUCTION 6-3
System Requirements 6-3

SECTION 2. METHOD OF OPERATION 6-5
Common Functions 6-5
Copy Pack Function 6-5
Copy File Function 6-8

SECTION 3. PROGRAM ORGANIZATION 6-11
Common Functions 6-11
Root Phase ($COPY) 6-11
Control Statement Analysis Phase ($COSYN) 6-11
Copy Pack Function 6-12
Main Data Transfer Phase ($COPIN) 6-12
Intermediate Phase ($COPIM) 6-13
Copy Files Function 6-14
Initialization Phase ($COFIN) 6-14
Simple Copy Files Phase ($CONDP) 6-14
Copy Non-Indexed Files Phase ($COREC) 6-16
Non-Reorganization Indexed Sequential Files Phase ($COCIO) 6-17
Reorganization Indexed Sequential Files Phase ($CORE) 6-17
Intermediate Control Phase ($COFIM) 6-18
Intermediate Reorganization Phase ($COIRY) . . . 6-18
Intermediate Non-Reorganization Phase ($COIRN) . . 6-19
Intermediate Consecutive Phase ($CONOI) 6-19
Print Phase ($COPRL – Disk System) ($COPRM – Model 6) 6-20
Copy Multi-Volume Direct Files Phase ($COMVD) . . 6-20
Intermediate Multi-Volume Direct Phase ($COIMD) . 6-21
End of Job Phase ($COEOJ) 6-21

SECTION 4. DATA AREA FORMATS 6-47
Common Data Areas 6-47
Tables 6-47
Buffers 6-49
Copy Pack Data Areas 6-50
Control Blocks 6-50
Tables 6-50
Buffers and Work Areas 6-51
Copy File Data Areas 6-51
Control Blocks 6-51
Tables 6-52
Buffers and Work Areas 6-54

PART VII. LIBRARY MAINTENANCE 7-1

SECTION 1. INTRODUCTION 7-3
System Requirements 7-3

SECTION 2. METHOD OF OPERATION 7-5
// ALLOCATE Control Statement 7-5
// RENAME Control Statement 7-5
// DELETE Control Statement 7-8
// COPY Control Statement 7-8
Copy Reader to Source Library 7-8
Copy System . . . . ,7-8
Copy Reader to Object Library (R. Entries) 7-8
Copy Reader to Object Library (O. Entries) 7-9
Copy All Libraries 7-9
Print/Punch Directory or Library 7-9
Copy Source Library to Source Library 7-9
Copy O. Entries or R. Entries to Object Libraries . . 7-9
// END Control Statement 7-9

SECTION 3. PROGRAM ORGANIZATION 7-11
Mainline Librarian Phase ($MAINT) 7-11
Allocate Syntax Scan Phase ($MAALS) 7-13
Allocate Mainline Phase ($MAAMN) 7-14
Allocate Object Library Phase ($MAAOL) 7-14
Allocate Source Library Phase ($MAASL) 7-15
Track Allocate/Deallocate Phase ($MATAD) 7-15
Reallocate Mainline Phase ($MAAMW) 7-16
Find Phase ($MAFND) 7-16
Copy Object Library Phase ($MAACO) 7-18
Copy Source Library Phase ($MAACS) 7-18
Copy Work Pack to Object Library Phase ($MARCO) . 7-19
Copy Work Pack to Source Library Phase ($MARCS) . 7-19
Rename Control Statement Processor Phase ($MAREM). 7-20
Delete Mainline Phase ($MADEM) 7-20
Copy Reader To Source Library Phase ($MACRS) . . 7-21
Copy Object Library – Disk to Disk Phase ($MADRV) . 7-22
Copy System Mainline Phase ($MACSY) 7-24
Object Library Handler Phase ($MAOLH) 7-24
Update PTF Log Sector Phase ($MALOG) 7-25
Copy Reader to Object Library (R. Entries) Phase ($MACRR) 7-25
Readjust Extended End of Object Library Phase ($MAEXE) 7-26
Copy Reader to Object Library (O. Entries) Phase ($MACRO) 7-26
Copy Source Library – Disk to Disk Phase ($MACSL) . 7-28
Print/Punch Mainline Phase ($MAPPM) 7-29
Print Source Directory Phase ($MASDP) 7-30
Print Object Directory Phase ($MAODP) 7-31
Print System Directory Phase ($MASYP) 7-31
Print/Punch Source Library Phase ($MASPT) 7-32
Print/Punch Object Library Phase ($MAOPT) 7-32

SECTION 4. DATA AREA FORMATS 7-105
Control Area (CAREA) 7-105
Object Library Directory 7-110
Source Library Directory 7-110

APPENDIX A. DIRECTORY A-1
Disk Initialization Program A-1
Alternate Track Assignment Program A-1
Alternate Track Rebuild Program A-2
File and Volume Label Display Program A-2
File Delete Program A-3
Disk Copy/Dump Program A-3
Library Maintenance Program A-4

APPENDIX B. FLOWCHARTING TECHNIQUES . . . B-1

INDEX X-1

HOW THIS PUBLICATION IS ORGANIZED

The manual is divided into seven parts – one part for each of the seven programs. The following sections are included, as required, in each part depending on the size and complexity of the program being described.

1. *Introduction* describes the functions and the operational environment of the program.

2. *Method of Operation* describes the functional and data flow of the program.

3. *Program Organization* describes the function of each phase and provides flowcharts and a storage map.

4. *Data Area Formats* describes the formats of areas within the program.

The *Introduction* and *Method of Operation* are designed to give the reader an overview of the functions and flow of the program. The *Program Organization* and *Data Area Formats* are designed to give detailed functional descriptions of each phase and data area comprising a program.

Appendix A contains a directory giving the entry point and synopsis of the phases in each program. Appendix B shows the flowcharting techniques used.

PART I
DISK INITIALIZATION

The IBM System/3 Disk Initialization program performs three functions:

- Primary initialization.
- Secondary initialization.
- Clear.

If the user requests primary disk initialization, the disk pack is initialized to the cylinder capacity of his system configuration. If the user requests secondary disk initialization, the initialized area of the disk pack is expanded to match his upgraded hardware capacity. Clear initializes a disk to hardware capacity without checking active fields.

The only keywords necessary for clear are TYPE-CLEAR, and UNIT-xx. The PID keyword (for internal IBM use only) in the UIN statement is used to initialize packs to half or full capacity. The keywords necessary are TYPE-PRIMARY, UNIT-xx, and PID-capacity (capacity is either half or full).

Note: PID-full is used only to override active file checking and force ERASE-YES.

Both the clear parameter and PID keyword force ERASE-YES. Secondary initialization used with ERASE-YES is invalid. Neither the clear parameter nor the PID keyword cause active file checking.

The Disk Initialization program performs the following operations:

- Performs surface analysis on each track by writing and verifying test data patterns.
- Formats the disk packs:

  1. Writes and identification (ID) field on all sectors. An ID field consists of three bytes: one flag byte, one byte containing the cylinder address, and one byte containing the sector address.

     Note: Alternate tracks are initialized with a flag byte of X‘01’.

  2. Reads an ID field on each track to ensure correct initialization.
  3. Writes the volume label.
  4. Formats the volume table of contents (VTOC) and the outboard recorder/statistical data recorder (OBR/SDR).
  5. Writes an IPL halt in the IPL sector.

- Assigns an alternate track to a defective primary track.
- Saves and restores program protection and OBR/SDR sectors if the unit is the fixed disk on drive (F1) and a valid volume label is present.
- Indicates a disk pack is unusable if:

  1. Cylinder 0, track 0 is defective.
  2. More than six tracks are defective.
  3. All six alternate tracks are defective and a primary track error occurs.
  4. One or more sector ID fields on a track cannot be read.

SYSTEM REQUIREMENTS

The IBM System/3 Disk Initialization program operates using the following minimum system configuration:

Disk System

- IBM 5410 Processing Unit, Model A12
- IBM 5203 Printer, Model A1
- IBM 5424 Multi-Function Card Unit, Model A
- IBM 5444 Disk Storage Drive, Model 2

Model 6

- IBM 5406 Processing Unit, Model B2 (Includes the keyboard with eight command keys)

- IBM 5213 Printer, Model 1

- IBM 5444 Disk Storage Drive, Model 2

The following optional features are supported:

- IBM 5496 Data Recorder, Model 1 with System/3 Model 6 On-line Feature (Used as the Sysin device)

- IBM 2265 Display Station, Model 2 (Cathode Ray Tube; used as Syslog device with IBM 5406 Processing Unit, Models B3 and B4 only)

Section 2. Method of Operation

Figure 1-1 shows the operational flow for the Disk Initialization program.

When the user loads the Disk Initialization program, the Root phase receives control. The Root phase loads the Control Statement Analysis phase into storage and passes control to it. The Control Statement Analysis phase reads the control statements and checks them for validity. The phase carries this out by sending three tables of keyword information (ISSCAN, ISUNTB, and ISVLTB) to the Keyword Syntax Scan routine. This routine checks for valid control statement keywords and returns the information in the Scan Output Buffer (ISSCOT) to the Control Statement Analysis phase. The Control Statement Analysis phase then sends two tables of parameters, ISCTBK and ISVBG, to the Root phase (ISSYSM and ISCTLO, two tables of halt information, are also passed to the Halt/ Syslog transient. See note which follows.)

The Root phase then passes control and the ICSTL table of parameters (ISCTBK and ISVBG) to the Mainline phase. The Mainline phase checks ICSTL to determine the types of initialization needed and if that initialization is valid. The Mainline phase then calls the Surface Analysis phase and passes an area of alternate track information (TBLE, and ALTIN) to it. ALTIN contains information about the status of alternate tracks; TBLE contains alternate track addresses. An I/O control blcok (IOB) is passed to the Surface Analysis phase to be used for writing and verifying the sector ID. The Surface Analysis phase tests the tracks for surface errors by writing and verifying the sector ID fields and a test data pattern. This phase performs its own error recovery procedure for surface errors. The write buffer address (IBFAD) is used for a special error recovery procedure. If error recovery is not needed, the write buffer IOADDR is used. The phase passes a return code (IRCDE), updated alternate track information (ALTIN), and control to the Mainline phase.

The Mainline phase assigns the alternate tracks and writes the volume label. The volume label for primary initialization is written in the VOLS output buffer. If secondary initialization is requested, the original volume label is saved and updated to new pack capacity in the IOSV buffer. The Mainline phase has two input/output control blocks (IOB and IOBV) which control the reading and writing for this program. After completion of processing, the Mainline phase passes a table of messages (IMSGT) to the Print phase. The Print phase logs the messages of actions taken.

Note: The EOJ transient performs the end of job requirements. The Mainline phase passes a table of halt messages (IERTAB) to the Halt/Syslog transient. The Halt/Syslog transient generates all halts due to errors during the program and prints all error messages. Halts are displayed on the Message Display Unit.

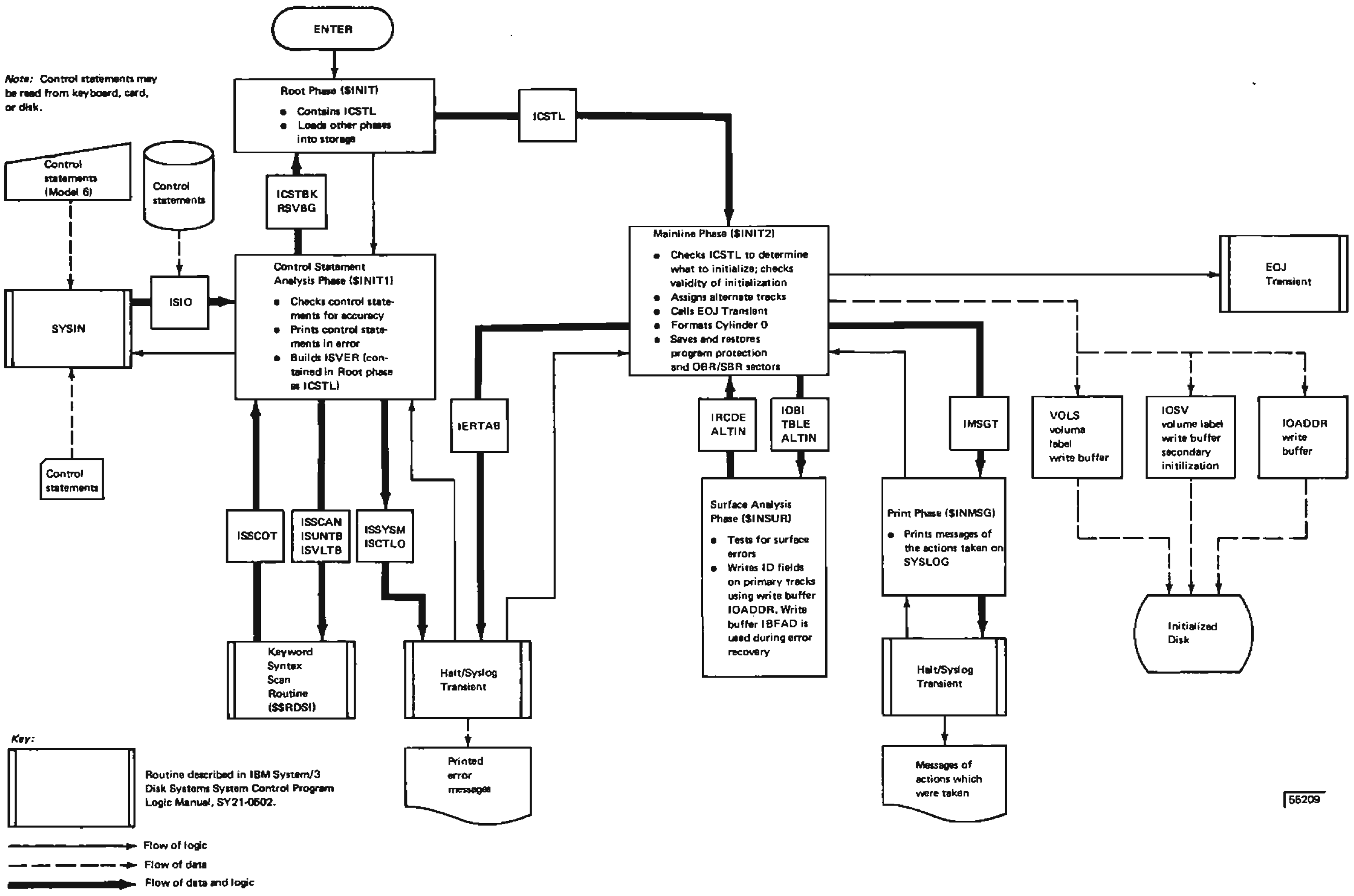

ENTER
Root Phase ($INIT)
• Contains ICSTL
• Loads other phases into storage
ICSTL
Note: Control statements may be read from keyboard, card, or disk.
Control statements (Model 6)
Control statements
ICSTBK RSVBG
SYSIN
ISIO
Control Statement Analysis Phase ($INIT1)
• Checks control statements for accuracy
• Prints control statements in error
• Builds ISVER (contained in Root phase as ICSTL)
Control statements
Mainline Phase ($INIT2)
• Checks ICSTL to determine what to initialize; checks validity of initialization
• Assigns alternate tracks
• Calls EOJ Transient
• Formats Cylinder 0
• Saves and restores program protection and OBR/SBR sectors
EOJ Transient
IERTAB
IRCDE ALTIN
IOBI TBLE ALTIN
IMSGT
VOLS volume label write buffer
IOSV volume label write buffer secondary initilization
IOADDR write buffer
ISSCOT
ISSCAN ISUNTB ISVLTB
ISSYSM ISCTLO
Surface Analysis Phase ($INSUR)
• Tests for surface errors
• Writes ID fields on primary tracks using write buffer IOADDR. Write buffer IBFAD is used during error recovery
Print Phase ($INMSG)
• Prints messages of the actions taken on SYSLOG
Initialized Disk
Keyword Syntax Scan Routine ($$RDSI)
Halt/Syslog Transient
Halt/Syslog Transient
Key:
Routine described in IBM System/3 Disk Systems System Control Program Logic Manual, SY21-0502.
Printed error messages
Messages of actions which were taken
55209
Flow of logic
Flow of data
Flow of data and logic

Figure 1-1. Operational Diagram for the Disk Initialization Program

Section 3. Program Organization

This section contains a functional description of each phase included in the IBM System/3 Disk Initialization program. Flowcharts are also provided where necessary. Figure 1-2 shows a storage map of the program.

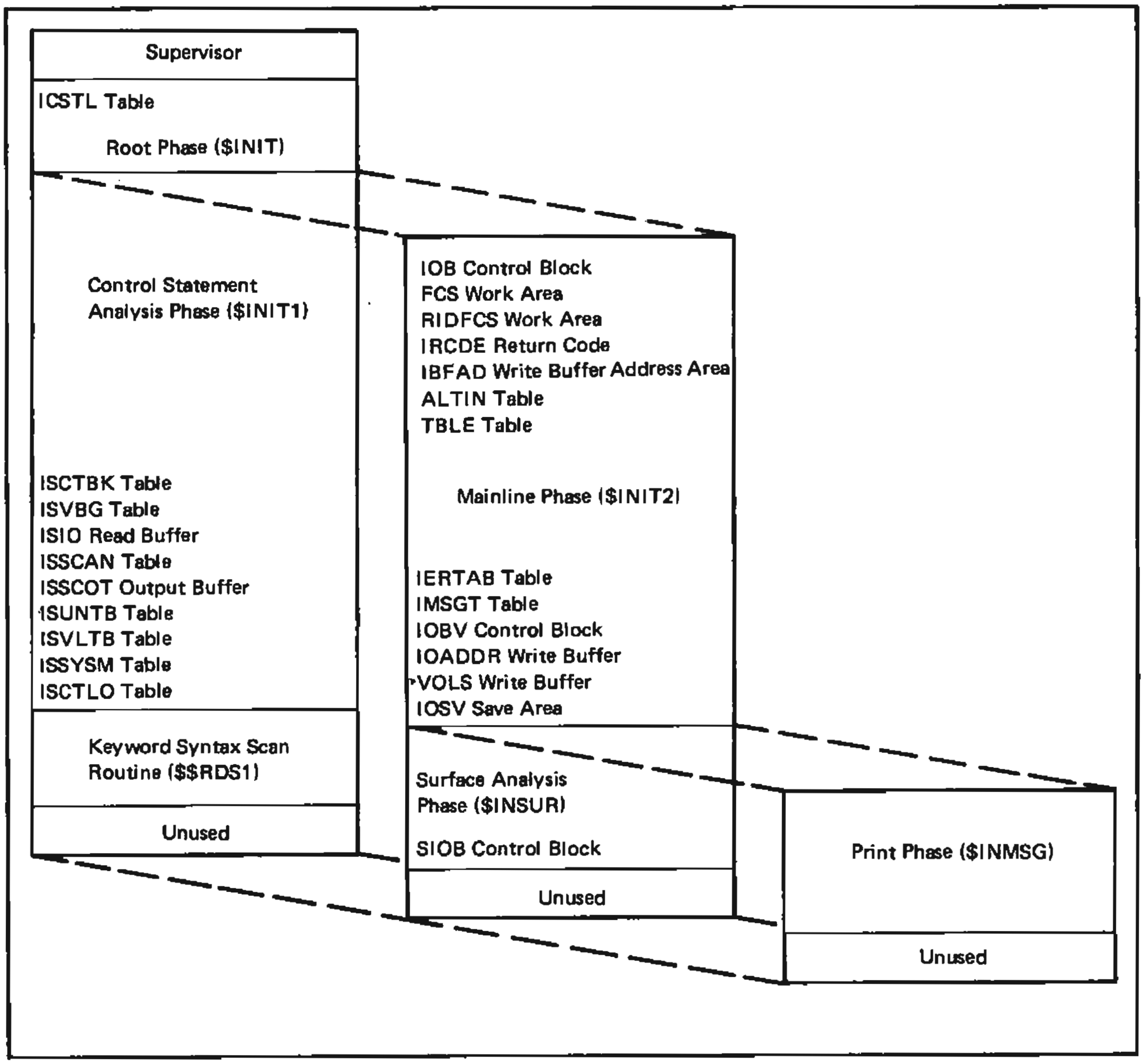


Figure 1-2. Storage Map of the Disk Initialization Program

Root Phase ($INIT)

Entry Point: IROOT

Chart: None

Functions:

- Loads the Control Statement Analysis phase and the Mainline phase into main storage.
- Maintains a table of constants (ICSTL) built by the Control Statement Analysis phase.

Input: Table of constants (ISCTBK and ISVBG) from Control Statement Analysis phase ($INIT1)

Output: Table of constants (ICSTL) to the Mainline phase ($INIT2)

Exits:

- Normal
  1. Supervisor to load the phases
  2. Entry point of the phases called
- Error – None

Control Statement Analysis Phase ($INIT1)

Entry Point: IDSYN

Chart: AA

Functions:

- Reads control statements into an input buffer (ISIO) from the procedures library or from the current system input device.

  Note: If a control statement error occurs and the statement is read from the procedures library, a retry is not allowed.

- Converts all numeric control statement operands from decimal to hexadecimal notation.
- Passes three tables of keyword information parameters (ISSCAN, ISUNTB, and ISVLTB) to the Keyword Syntax Scan routine. The control statements are then checked for validity and proper sequence:

  1. The // UIN control statement is checked for proper UNIT, TYPE, ERASE, VERIFY, and PID parameters.

     Note: The following halts are associated with the parameter for PID keyword (for Customer Engineer use only):

     Disk System

Halt	*Log*	*Reason and Recovery*
5U		Reason - Error in 'UIN' statement.
	UI 5U P1	Parameter for PID keyword is missing.
	UI 5U P2	Parameter for PID keyword is a sublist which is not allowed.
	UI 5U P3	Parameter for PID keyword is not either half or full.
		Recovery - Correct statement and re-enter.

Model 6

Halt	*Log*	*Reason and Recovery*
BCD12		Reason - Error in 'UIN' statement.
	UI BCD12 5PI	Parameter for PID keyword is missing.
	UI BCD12 5P2	Parameter for PID keyword is a sublist which is not allowed.
	UI BCD12 5P3	Parameter for PID keyword is not either half or full.
		Recovery - Correct statement and re-enter.

2. The // VOL control statement is checked for proper ID and PACK parameters

- Builds two tables of parameters (ISCTBK and ISVBG) from the information supplied in the control statements and passes it to the Root phase.

Input: Control statements (// UIN, // VOL, // END)

Output:

- Two tables of parameters (ISCTBK and ISVBG) to the Root phase
- Table of constants (ISCTLO) to the Halt/Syslog transient to print control statements.

Exits:

- Normal
  1. Root phase (SINIT) after building keyword parameter tables
  2. Keyword Syntax Scan routine (SSRDS1) to check for valid keywords.
- Error – Halt/Syslog transient to print error messages (ISSYSM)

Mainline Phase ($INIT2)

Entry Point: IMAIN

Chart: AB

Functions:

- Obtains the parameters from the table ICSTL.
- Reads the volume label.
- Determines the validity of the requested initialization:
  1. If a volume label was not found, only primary or clear initialization can occur and ERASE-YES is forced. ERASE-YES is forced if the pack has to be re-initialized during one run.
  2. The pack capacity is determined from the hardware sense bytes.
  3. If secondary initialization is requested, determines the first track to be initialized.
- Initializes the alternate track information table (ALTIN) and saves this information and the volume label for secondary initialization.
- Does a special space request allocation to check for active files.
- Passes a parameter table in XR1 (IOB) to the Surface Analysis phase and checks the return code (IRCDE) it receives from the phase.
- Saves and restores the program protection sectors (cylinder 0, track 1, sectors 12, 13, and 14) and the OBR/SBR sectors (cylinder 0, track 0, sectors 3 through 8) if the fixed disk on drive 1 (F1) is to be initialized and the volume label has no errors.
- Assigns alternate tracks:
  1. The phase writes the ID fields on the assigned alternate tracks.
  2. If a primary track is assigned to an alternate track, the ID field of the primary track contains the address of the alternate track, and the alternate track contains the address of the defective primary track.

- Writes the volume label for primary initialization and updates the volume label for secondary initialization.
- Formats VTOC, OBR/SDR, and IPL sectors.

Input:

- Table of constants (ICSTL) from the Root phase
- Return code (IRCDE) from the Surface Analysis phase

Output:

- Initialized Disk
- Table of messages (ISMGT) to Print phase
- IOB, ALTIN, and TBLE containing information to Surface Analysis phase.

Exits:

- Normal
  1. EOJ transient to terminate job.
  2. Surface Analysis phase to pass track information
  3. Print phase to log message of actions taken
- Error - Halt/Syslog transient to print error messages

Surface Analysis Phase ($INSUR)

Entry Point: SURFAN

Chart: AC

Functions:

- Writes and verifies sector addresses using two test patterns (X'55' and X'00') the number of times specified in the VERIFY parameter.
- Provides a special error recovery procedure for surface errors by trying to write the data ten times.
- Updates alternate track information table (ALTIN) in the Mainline phase.
- Passes a return code (IRCDE) to the Mainline phase:
  1. X'00' – No error
  2. X'01' – Unusable pack (cylinder 0 in error, more than six defective tracks, or defective track ID cannot be read)
  3. X'02' – Alternate track defective
  4. X'04' – Hardware error
  5. X'08' – Alternate track assignment valid

Input:

- XR1 points to an IOB defined in the Mainline phase
- XR2 points to write buffer IOADDR

Output:

- Return code IRCDE to the Mainline phase
- Updated ALTIN information

Exits:

- Normal – ERCHK in the Mainline phase ($INIT2)
- Error – None

Print Phase ($INMSG)

Entry Point: ALMSG

Chart: AD

Functions:

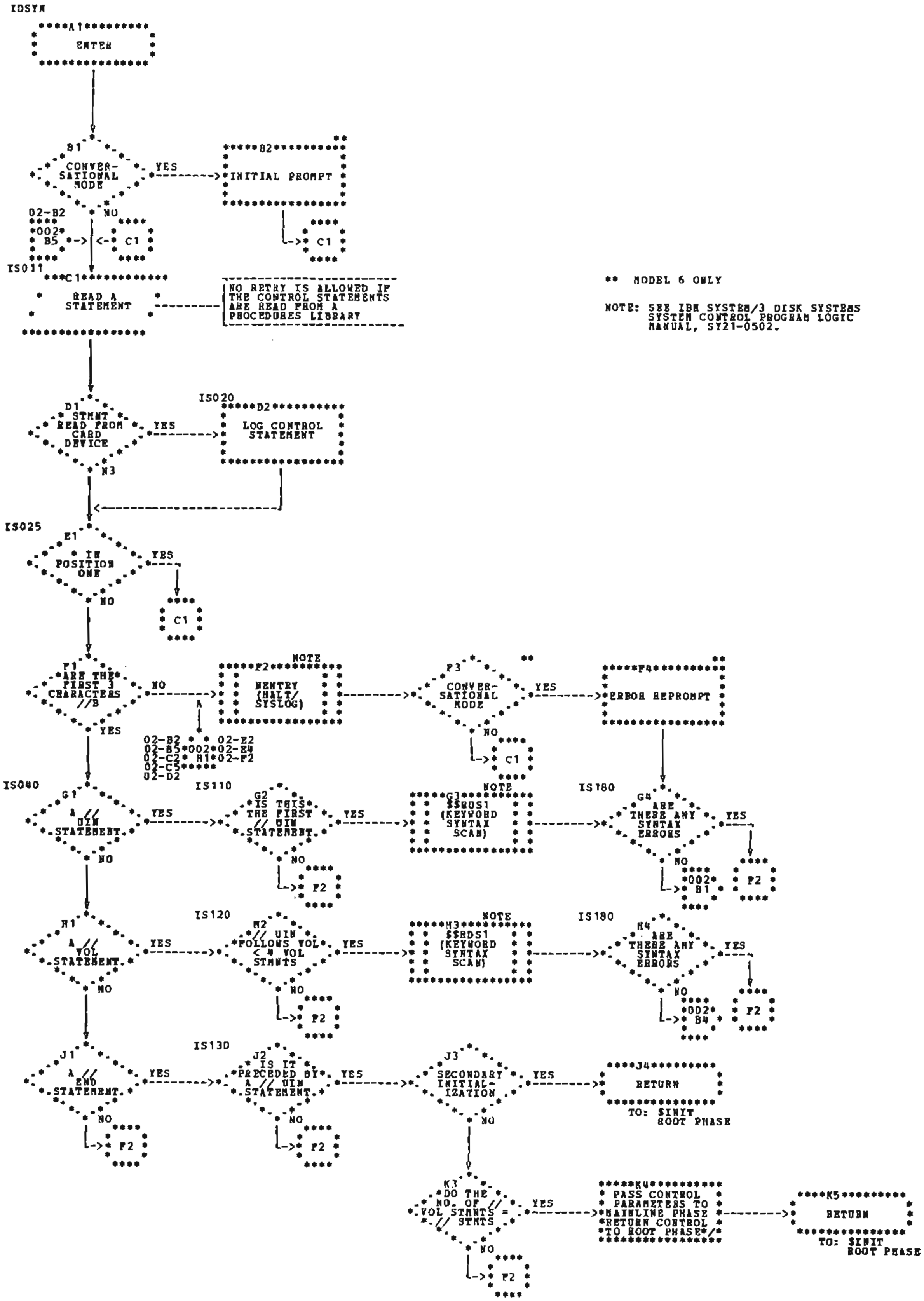
- Logs the results of a successful alternate track assignment.
- Prints addresses of primary and alternate tracks.
- Logs results of initialization.

Input: Table of messages (IMSGT) to be printed from the Mainline phase

Output: Printed messages

Exit:

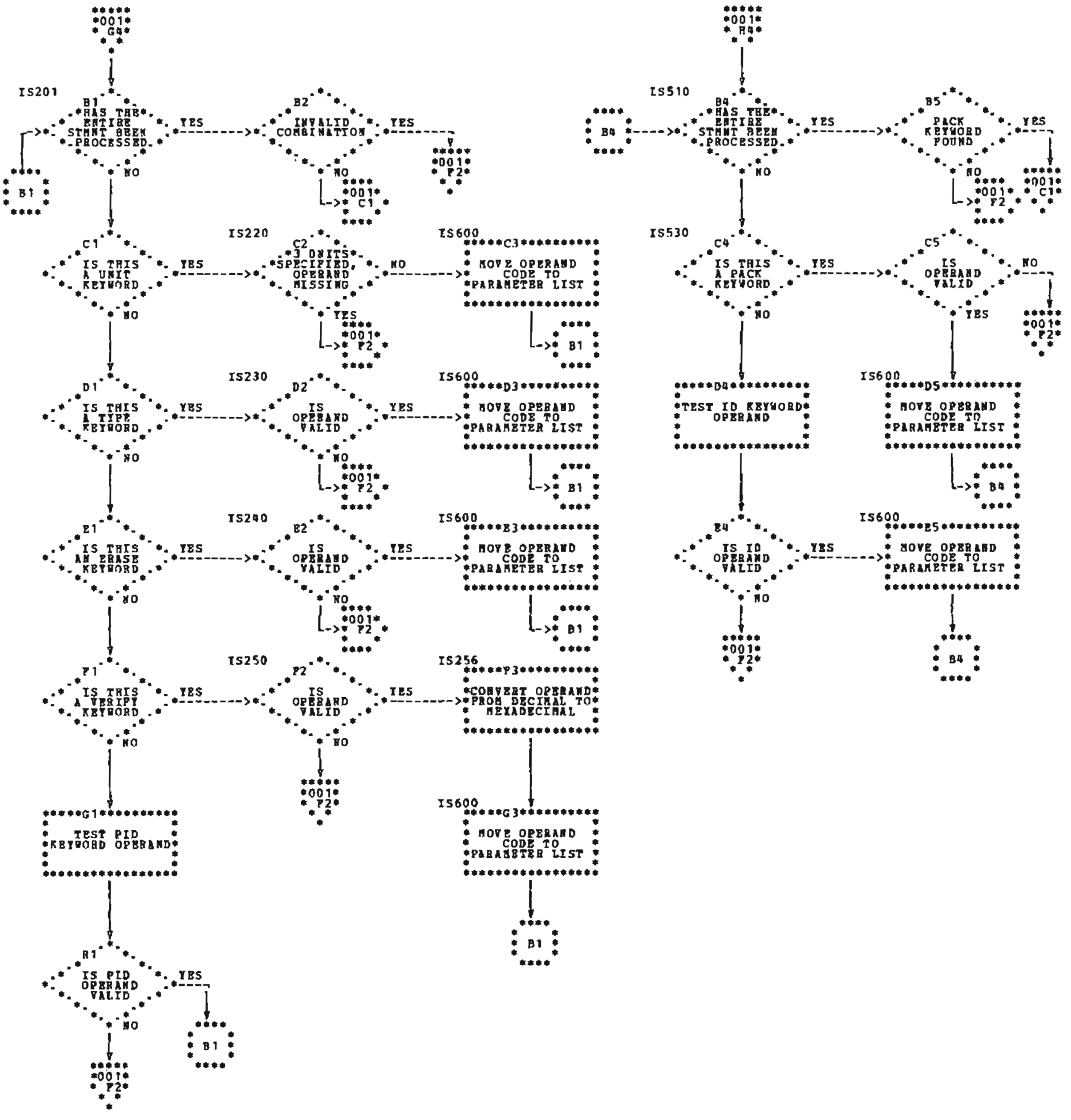
- Normal
  1. Mainline phase ($INIT2) after messages are printed
  2. Halt/Syslog transient to print messages of actions taken
- Error – None



** MODEL 6 ONLY

NOTE: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart AA (Part 1 of 2). Control Statement Analysis Phase



001 G4
IS201
B1 HAS THE ENTIRE STMNT BEEN PROCESSED
YES
NO
B1
B2 INVALID COMBINATION
YES
NO
001 F2
001 C1
C1 IS THIS A UNIT KEYWORD
YES
NO
IS220
C2 3 UNITS SPECIFIED, OPERAND MISSING
NO
YES
001 F2
IS600
C3 MOVE OPERAND CODE TO PARAMETER LIST
B1
D1 IS THIS A TYPE KEYWORD
YES
NO
IS230
D2 IS OPERAND VALID
YES
NO
001 F2
IS600
D3 MOVE OPERAND CODE TO PARAMETER LIST
B1
E1 IS THIS AN ERASE KEYWORD
YES
NO
IS240
E2 IS OPERAND VALID
YES
NO
001 F2
IS600
E3 MOVE OPERAND CODE TO PARAMETER LIST
B1
F1 IS THIS A VERIFY KEYWORD
YES
NO
IS250
F2 IS OPERAND VALID
YES
NO
001 F2
IS256
F3 CONVERT OPERAND FROM DECIMAL TO HEXADECIMAL
IS600
G3 MOVE OPERAND CODE TO PARAMETER LIST
B1
G1 TEST PID KEYWORD OPERAND
H1 IS PID OPERAND VALID
YES
B1
NO
001 F2
001 H4
IS510
B4
B4 HAS THE ENTIRE STMNT BEEN PROCESSED
YES
NO
B5 PACK KEYWORD FOUND
YES
001 C1
NO
001 F2
IS530
C4 IS THIS A PACK KEYWORD
YES
NO
C5 IS OPERAND VALID
NO
001 F2
YES
D4 TEST ID KEYWORD OPERAND
IS600
D5 MOVE OPERAND CODE TO PARAMETER LIST
B4
E4 IS ID OPERAND VALID
YES
NO
001 F2
IS600
E5 MOVE OPERAND CODE TO PARAMETER LIST
B4

Chart AA (Part 2 of 2). Control Statement Analysis Phase

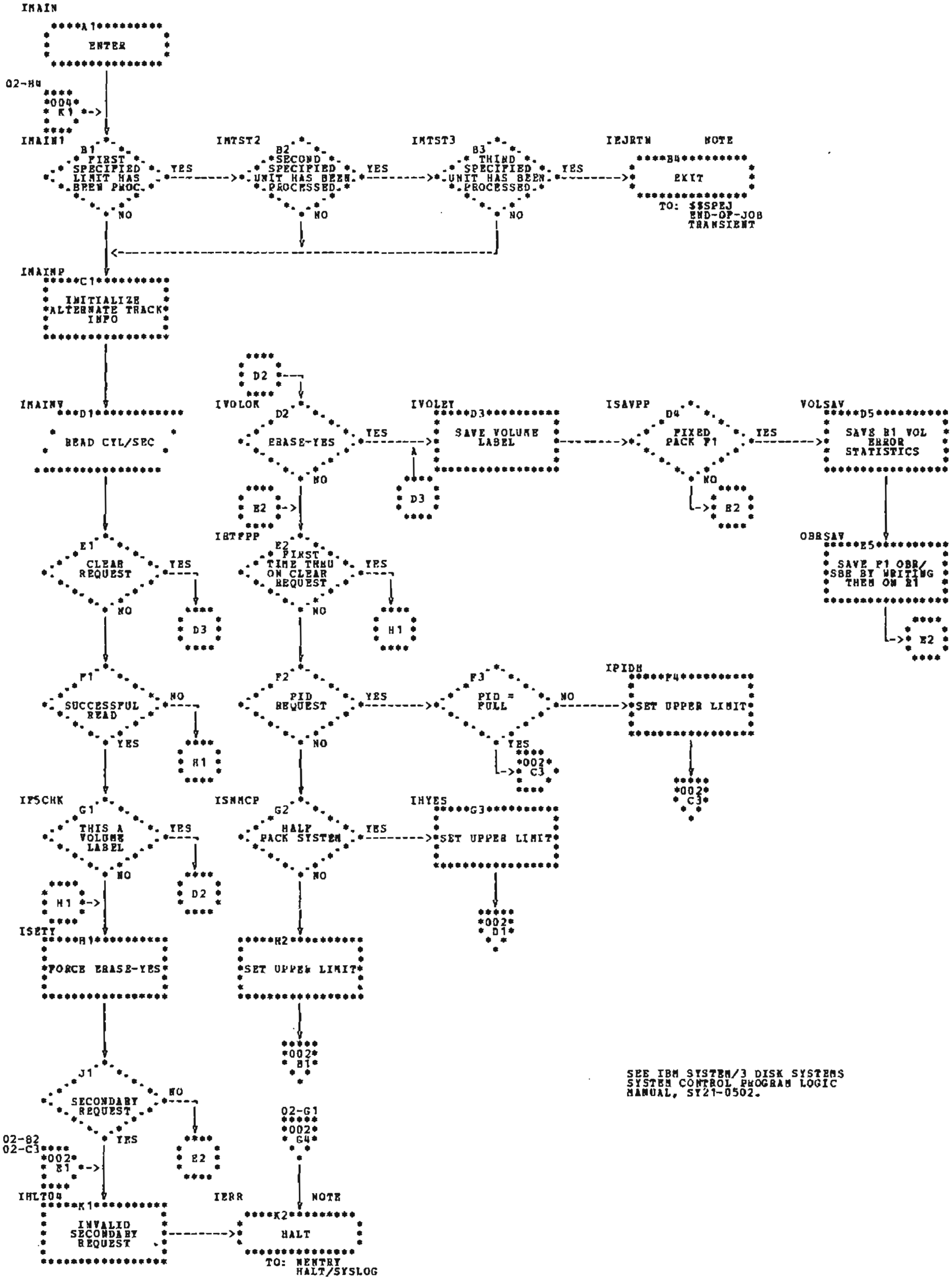


Chart AB (Part 1 of 4). Mainline Phase

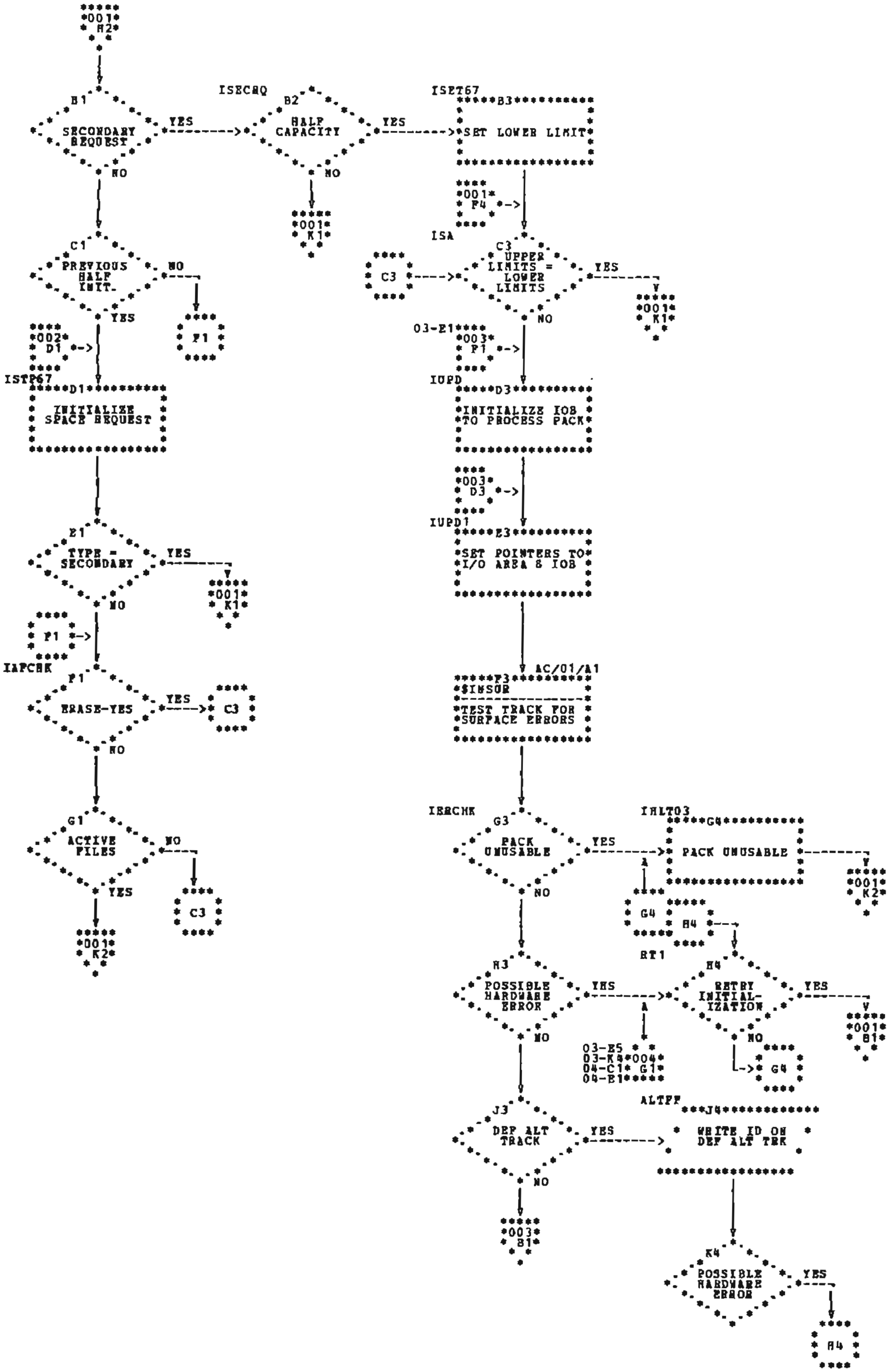


Chart AB (Part 2 of 4). Mainline Phase

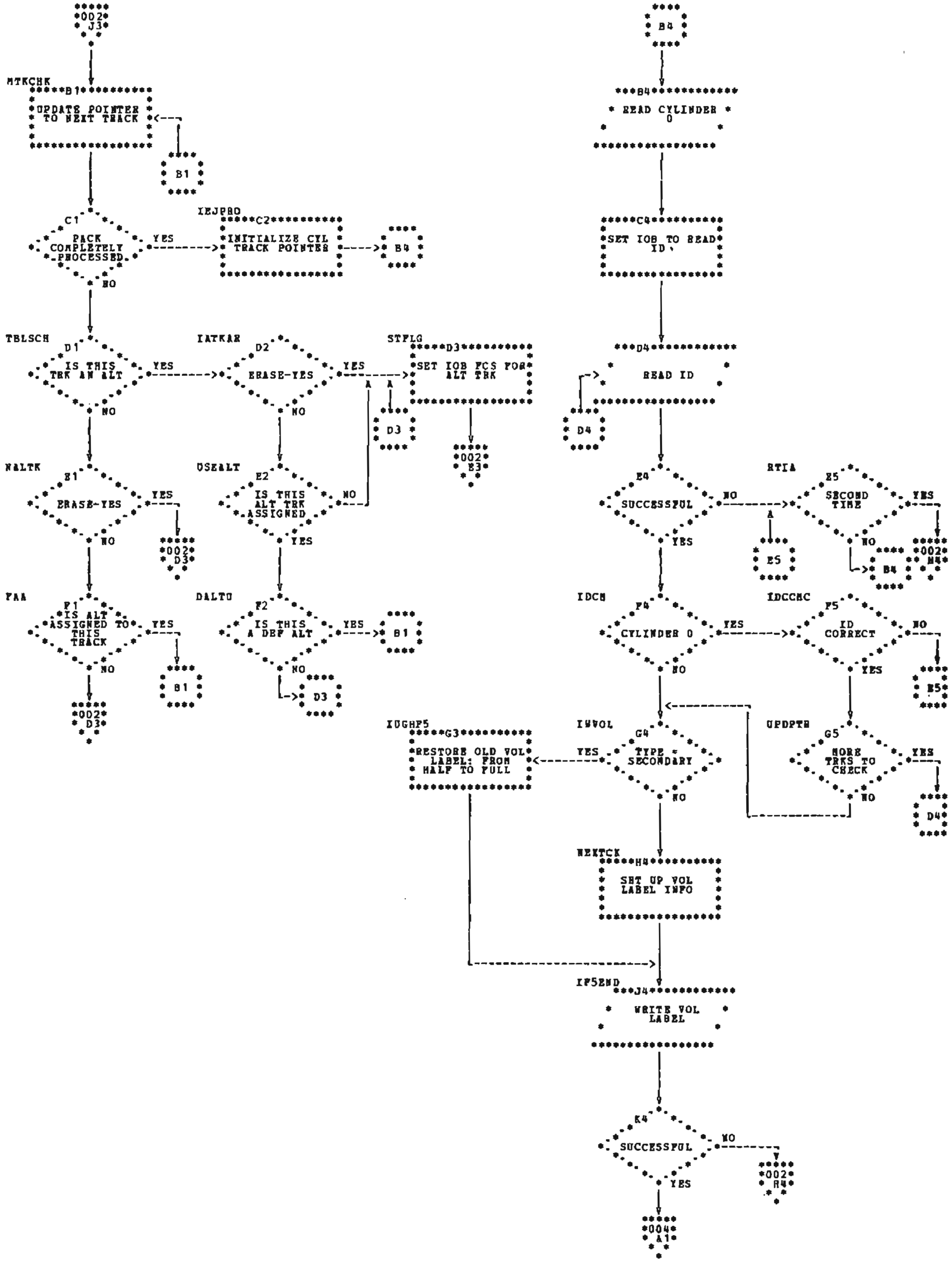


Chart AB (Part 3 of 4). Mainline Phase

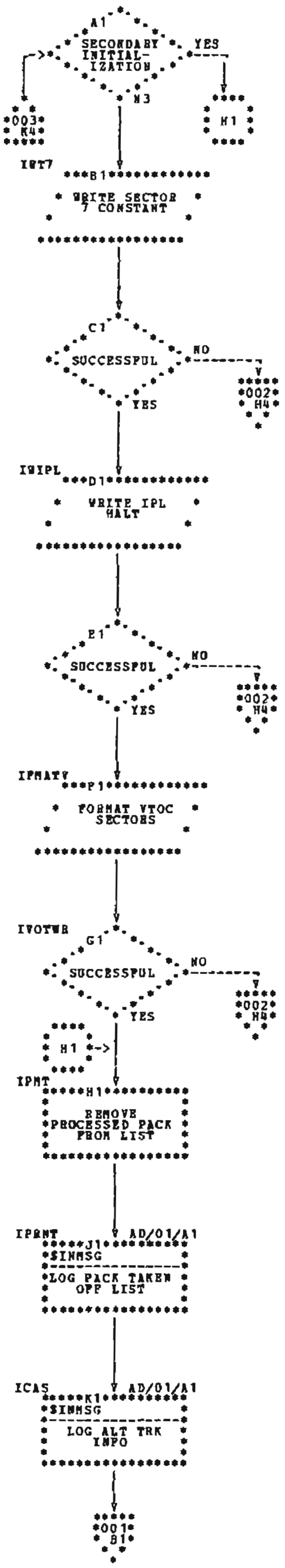


Chart AB (Part 4 of 4). Mainline Phase

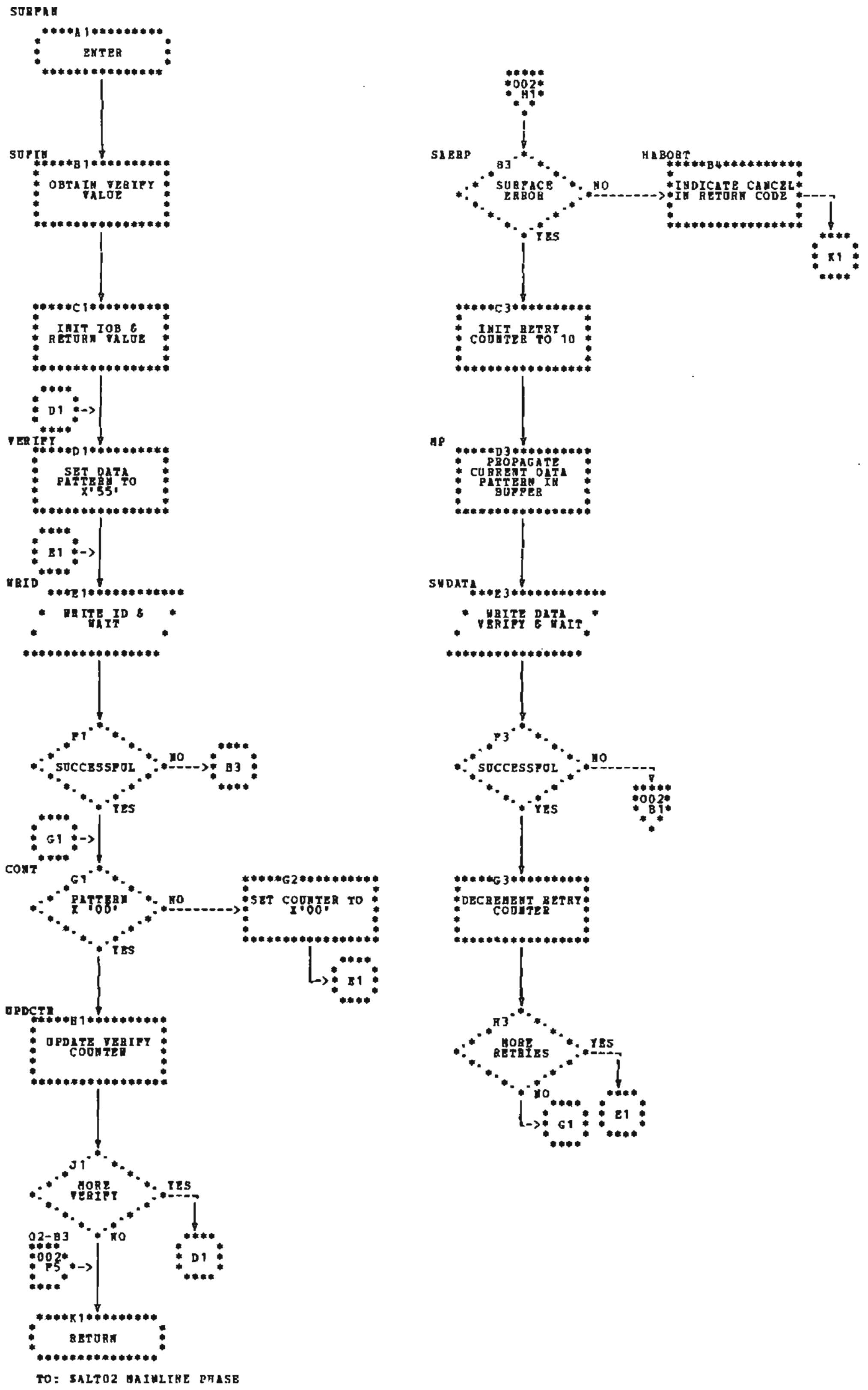


Chart AC (Part 1 of 2). Surface Analysis Phase

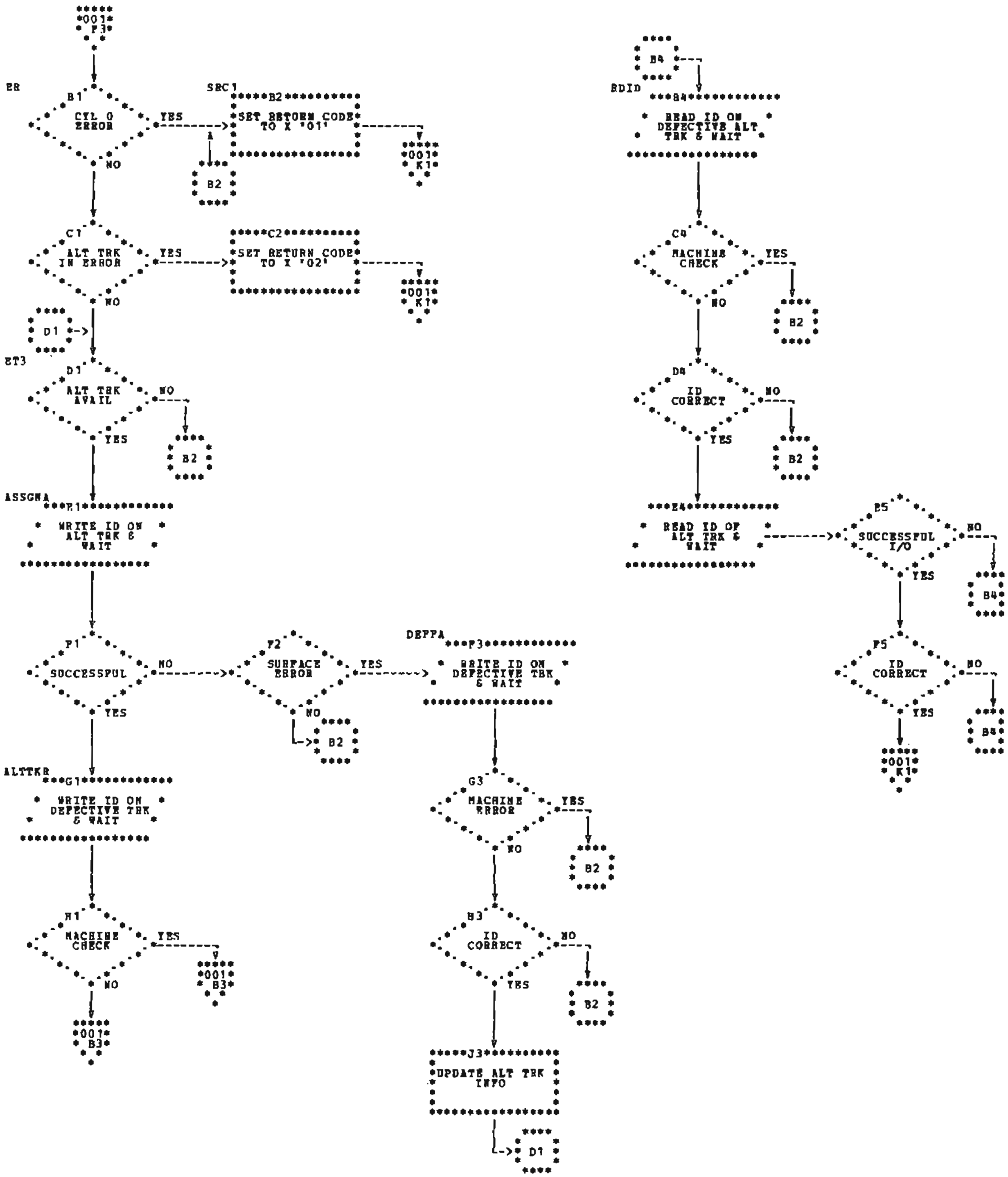
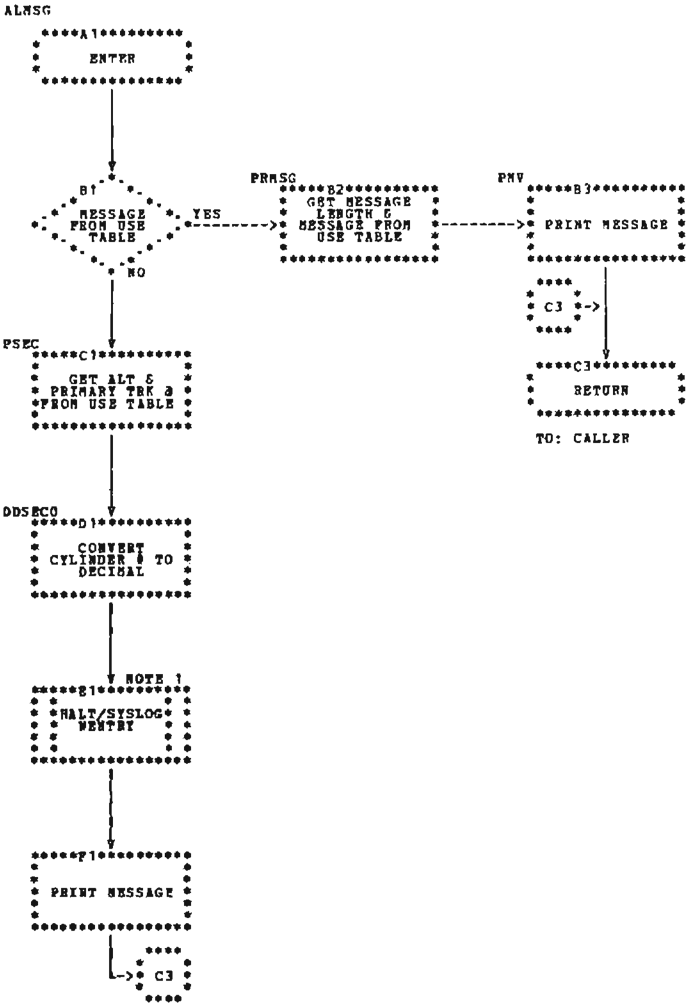


Chart AC (Part 2 of 2). Surface Analysis Phase



NOTE 1: SEE IBM SYSTEM/3
DISK SYSTEMS SYSTEM
CONTROL PROGRAM
LOGIC MANUAL SY21-0502

Chart AD. Print Phase

Section 4. Data Area Formats

This section describes the format and contents of the control blocks, tables, and data areas used by the IBM System/3 Disk Initialization program. See the storage map in Figure 1-2 for the relative location of these data areas.

CONTROL BLOCKS

The Disk Initialization program requires three input/output control blocks:

1. IOB
2. IOBV
3. SIOB

The IOB, defined by the Mainline phase, is used to read cylinder 0, track 0, sector 2, and to write and verify the ID fields of each track. It is located before the entry point (IMAIN) of the Mainline phase. A byte labeled IOBFLG can be set to the following indications:

- Bit 0 – The program will handle its own error recovery procedures.
- Bit 2 – No error logging on outboard recorder/statistical data recorder (OBR/SDR).
- Bit 3 – The data area FCS will be used for comparisons during the disk IOS SCAN instructions.
- Bit 4 – The program does not use data management.

The IOB also contains the control information passed to the Surface Analysis phase by the Mainline phase.

The IOBV, defined by the Mainline phase, is used to write the volume label. It is located at the end of the Mainline phase.

The SIOB is used for writing data during surface error recovery procedures in the Surface Analysis phase. It is located at the end of the Surface Analysis phase. A byte labeled SIOBFL is set to indicate that the phase will handle surface error recovery procedures.

Byte positions and contents for the input/output blocks labeled IOB, IOBV, or SIOB are:

Byte	*Contents*
0-1	Chain pointer used and updated by IOS to queue requests
2	Completion code
3	Q byte indicating operation
4	R control byte
5	Cylinder address in hexadecimal
6	Sector address in hexadecimal
7	Number of sectors involved in operation (N-1)
8-9	Data area address
10-11	Sense bytes used by IOS
12	Error counter used by IOS
13	Flag byte X‘80’ = No error recovery X‘40’ = No verification X‘20’ = No error logging X‘10’ = Write ID operation X‘08’ = No DTF X‘04’ = No load I/O for disk file date register X‘02’ = Equipment check X‘01’ = Alternate track in process
14-15	Register 1 (XR1) save area
16-17	ARR save area

TABLES

ICSTL Table

ICSTL is a 58-byte table of parameters of control statement information maintained by the Root phase (Figure 1-3). The Control Statement Analysis phase ($INIT) builds two tables, ISCTBK and ISVBG, and passes them to the Root phase where they are maintained as ICSTL. ICSTL is located at the beginning of the Root phase.

Byte	Label	Contents
0	IBVER	VERIFY option
1	ITYPE	TYPE option 1. P = Primary 2. S = Secondary 3. C = Clear
2	IERASE	ERASE option 1. Y = Yes 2. N = No
3	IPID	PID option 1. F = Full 2. H = Half
4	IUN1	Hardware address of first unit (Q code)
5	IUN2	Hardware address of second unit (Q code)
6	IUN3	Hardware address of third unit (Q code)
7-12	IVSER1	Volume serial number of first unit
13-22	IOID1	Owner ID of first unit
23-28	IVSER2	Volume serial number of second unit
29-38	IOID2	Owner ID of second unit
39-44	IVSER3	Volume serial number of third unit
45-54	IOID3	Owner ID of third unit
55	INSEC	Number of sectors for error recovery
56-57	IBUFAD	Address of error recovery buffer

Figure 1-3. ICSTL Table

ISCTBK Table

ISCTBK, a 7-byte table of parameters defined by the Control Statement Analysis phase, contains the requested parameters on the control statements (Figure 1-4). ISCTBK, along with ISVBG, is passed to the Root phase which maintains these parameters under the label ICSTL. ISCTBK is located at the beginning of the Control Statement Analysis phase.

ISVBG Table

ISVBG, a 48-byte table of parameters, is built in the Control Statement Analysis phase (Figure 1-5). This table, along with ISCTBK, is passed to the Root phase to be maintained under the label ICSTL. ISVBG is located after ISCTBK at the beginning of the Control Statement Analysis phase.

Byte	Label	Contents
0	ISVER	VERIFY option
1	ISTYP	TYPE option 1. P = Primary 2. S = Secondary 3. C = Clear
2	ISERAS	ERASE option 1. Y = Yes 2. N = No
3	ISPID	PID option 1. F = Full 2. H = Half
4	ISUT1	Hardware address of first unit
5	ISUT2	Hardware address of second unit
6	ISUT3	Hardware address of third unit

Figure 1-4. ISCTBK Table

ISSCAN Table

ISSCAN is an 8-byte table of parameters passed to the Keyword Syntax Scan routine by the Control Statement Analysis phase (Figure 1-6). ISSCAN is located at the end of the Control Statement Analysis phase.

ISUNTB Table

ISUNTB, a 29-byte table of keyword parameters, is passed to the Keyword Syntax Scan routine by the Control Statement Analysis phase. It is located in the Control Statement Analysis phase. ISUNTB contains the parameters for the // UIN control statement. If a // UIN control statement is read, the address of ISUNTB (ISPMAD) is placed in the ISSCAN parameter table. The byte positions and contents are:

Byte	*Contents*
0	X'14'
1-4	'UNIT'
5	X'24'
6-9	'TYPE'
10	X'36'
11-15	'ERASE'
16	X'46'
17-22	'VERIFY'
23	X'53'
24-26	'PID'
27-28	X'FFFF'

ISVLTB Table

ISVLTB, a 10-byte table of keyword parameters, is passed to the Keyword Syntax Scan routine by the Control Statement Analysis phase. It is located near the end of the Control Statement Analysis phase. ISVLTB contains the parameters for the // VOL control statement. If a // VOL statement is read, the address of this table is placed in the ISSCAN table (ISPMAD). The byte positions and contents are:

Byte	*Contents*
0	X'14'
1-4	'PACK'
5	X'22'
6-7	'ID'
8-9	X'FFFF'

Byte	Label	Contents
0-15	ISVL1	Volume serial number and owner ID of first unit
16-31	ISVL2	Volume serial number and owner ID of second unit
32-47	ISVL3	Volume serial number and owner ID of third unit

Figure 1-5. ISVBG Table

Byte	Label	Contents
0-1	ISSTAD	Address of where to start scan
2-3	None	Address of where to end scan
4-5	None	Address of scan output area
6-7	ISPMAD	Address of keyword parameter table

Figure 1-6. ISSCAN Table

IERTAB Table

IERTAB is a 7-byte table of halt information passed to the Halt/Syslog transient by the Mainline phase (Figure 1-7). It is located near the end of the Mainline phase.

IMSGT Table

The IMSGT table contains messages of actions taken which are to be printed. The table is passed from the Mainline phase to the Print phase. It is located near the end of the Mainline phase.

ISSYSM Table

ISSYSM, a 7-byte table for halt messages, is passed by the Control Statement Analysis phase to the Halt/Syslog transient (Figure 1-8). It is located near the end of the Control Statement Analysis phase.

ISCTLO Table

ISCTLO is a 5-byte table of constants passed by the Control Statement Analysis phase to the Halt/Syslog transient to print control statements. It is located at the end of the Control Statement Analysis phase. The byte positions and contents are:

Byte	*Contents*
0-1	X'0000' – Indicates output only; log control statements
2	X'60' – Indicates 96-byte length
3-4	Address of I/O area (input buffer ISIO)

Byte	Label	Contents
0	IERTAB	B'11010110' – Indicates halt and system print
1	None	C'I' – Indicates Disk Initialization program
2-3	IHSTX	Disk System – Characters to be used for halt display lights Model 6 – Code to be converted for halt display lights
4	ISABT	Operator options X'01' = Immediate cancel X'02' = Control cancel X'04' = Retry X'08' = Proceed
5-6	OSIBM	C'ƀƀ' – Indicates submessage area

Figure 1-7. IERTAB Table

Byte	Label	Contents
0	ISSYM	B'11010110' – Indicates halt and system print
1	None	C'I' – Indicates Disk Initialization program
2-3	ISHALT	Disk System – Characters to be used for halt display lights Model 6 – Code to be converted for halt display lights
4	None	Operator options X'01' = Immediate cancel X'02' = Control cancel X'04' = Retry X'08' = Proceed
5-6	ISSUB	C'ƀƀ' – Indicates submessage area

Figure 1-8. ISSYSM Table

ALTIN Table

This 12-byte table, defined by the Mainline phase, contains two bytes of information on each of the six alternate tracks. ALTIN is located after the FCS work area at the beginning of the Mainline phase. It is initialized to zeros and may be modified by the Surface Analysis phase. Each of the 2-byte areas within the table contains one of the following hexadecimal configurations:

Contents	*Meaning*
X'00XX'	Alternate track is available for assignment
X'FFXX'	Alternate track is defective and may not be used
X'NNNN'	Address of the primary track to which the alternate track is assigned

TBLE Table

This 12-byte area, defined by the Mainline phase, is located at the beginning of the phase and contains the addresses of the alternate tracks. The byte positions and contents are:

Byte	*Contents*
0-1	X'0100' = Track number one (upper)
2-3	X'0180' = Track number one (lower)
4-5	X'0200' = Track number two (upper)
6-7	X'0280' = Track number two (lower)
8-9	X'0300' = Track number three (upper)
10-11	X'0380' = Track number three (lower)

BUFFERS

ISIO Read Buffer

ISIO, a 96-byte read buffer used by the Control Statement Analysis Phase, contains the control statement images. It is located at the end of the Control Statement Analysis phase.

ISSCOT Output Buffer

ISSCOT, a 96-byte scan output buffer, contains the parameters found on the control statements. It is passed from the Keyword Syntax Scan routine to the Control Statement Analysis phase. It is located at the end of the Control Statement Analysis phase.

IOADDR Write Buffer

IOADDR is a 256-byte write buffer used by the Mainline and Surface Analysis phases for writing, reading, and verifying data on disk. It is located at the end of the Mainline phase.

IBFAD Write Buffer Address Area

IBFAD, a 2-byte area, contains the address of a write buffer used during the error recovery procedure of the Surface Analysis phase. It is located in the Mainline phase.

VOLS Write Buffer

VOLS is a 256-byte write buffer used by the Mainline phase to write the volume label on disk. It is located at the end of the Mainline phase. The byte positions and contents of this buffer are:

Byte	*Contents*
0-2	VOL
3-9	Volume serial number
10-81	Initialized to zero
82-91	Owner ID
92-100	Device constants
101-112	Alternate track information (ALTIN table)
113-214	Initialized to zero
215-238	Suspected defective tracks; indicators are initialized to X'FF'
239-255	Initialized to zero

WORK AREAS

FCS Work Area

The FCS work area, located in the Mainline phase immediately following the control block IOB, is a 3-byte area initialized to zeros. It is comprised of a flag byte, a cylinder address, and a sector address. Bit 3 of the IOBFLG byte in the IOB indicates that FCS is to be used during the IOS SCAN instruction. The IOB also contains a cylinder and sector address of a primary track on which a write or read operation is to occur. The FCS contains the same address as the IOB if the track is operative. If the track is defective FCS contains the address of the assigned alternate track. In this case the FCS flag byte contains a X'01' to indicate that the alternate track is an alternate track (X'02' indicates a defective primary track; X'03' indicates a defective alternate track).

RIDFCS Work Area

The RIDFCS work area, located in the Mainline phase following the FCS, is a 3-byte area initialized to zeros. During a read ID operation, IOS will place the address of the track that was actually read (either a primary track or an alternate track) in this data area.

IRCDE Return Code

IRCDE, a 1-byte data area initialized to zero, contains a return code passed from the Surface Analysis phase to the Mainline phase after surface analysis has occurred. The possible codes are:

Code	*Meaning*
X'00'	No error
X'01'	Unusable pack (cylinder 0 in error, more than six defective tracks, or defective track ID cannot be read)
X'02'	Alternate track defective
X'04'	Hardware error
X'08'	Alternate track assignment valid

IOSV Save Area

IOSV, a 256-byte save area in the Mainline phase, contains the original volume label if secondary initialization is requested. This area is located at the beginning of the Mainline phase.

Note: This area overlays the Root phase.

PART II
ALTERNATE TRACK ASSIGNMENT

The IBM System/3 Alternate Track Assignment program attempts to recover from read or write error conditions encountered on a track located on a previously initialized disk pack. There are three ways this program can be run:

1. *Conditional assignment* transfers data from a suspected defective track to an available alternate track. Surface analysis is then performed on the original track. If the track is defective, the ID fields of the defective track and an alternate track are set to reflect an alternate track assignment. If the track is not defective, data is transferred back to the original track; the alternate track is available for further assignment.

2. *Unconditional assignment* bypasses surface analysis and unconditionally assigns an alternate track to the suspected defective track.

3. *Unconditional cancel of prior assignment* attempts to return data located on an assigned alternate track back to the original track. The alternate track is available for further use.

After an assignment or cancel of prior assignment has been executed, the program performs the following functions:

- Updates the volume label (cylinder 0, track 0, sector 2).
- Prints data from the defective sectors to allow the user to check the data for accuracy (assignment only).
- Logs results of action taken.

SYSTEM REQUIREMENTS

The IBM System/3 Alternate Track Assignment program uses the following minimum system configuration:

Disk System

- IBM 5410 Processing Unit, Model A12
- IBM 5203 Printer, Model A1
- IBM 5424 Multi-Function Card Unit, Model A
- IBM 5444 Disk Storage Drive, Model 2

Model 6

- IBM 5406 Processing Unit, Model B2 (Includes the keyboard with eight command keys)
- IBM 5213 Printer, Model 1
- IBM 5444 Disk Storage Drive, Model 2

The following optional features are supported:

- IBM 5496 Data Recorder, Model 1 with System/3 Model 6 On-line Feature (Used as the Sysin device)
- IBM 2265 Display Station, Model 2 (Cathode Ray Tube, used as Syslog device with IBM 5406 Processing Unit, Model B3 and B4 only)

Figure 2-1 shows the operational flow of the Alternate Track Assignment program.

When the user loads the Alternate Track Assignment program, the Root phase receives control. The Root phase then loads the Control Statement Analysis phase into main storage, which reads the control statements (ASIOAR) and checks their validity. The Control Statement Analysis phase passes control and parameter list ASSC to the Keyword Syntax Scan routine ($$RDS1) which scans the control statements for the keywords indicated in ASSC. Output from the Keyword Syntax Scan routine is located in the ASSCOT buffer. The Control Statement Analysis phase then builds a table of constants (ASCTBK) and returns control to the Root phase. The Root phase maintains this table of constants under the label ABER.

The Root phase loads the Mainline phase which performs one of the following functions:

1. Conditional assignment

2. Unconditional assignment

3. Unconditional cancel of prior assignment

CONDITIONAL ASSIGNMENT

If the user requests a conditional assignment, the Mainline phase obtains the addresses of the suspected defective tracks from the volume label, builds table ATPDL, and writes the ID field on an alternate track. It then loads $ALT03 to transfer data from the suspected defective track to an alternate track. Control information is passed to the phase in the AIOB control block. After the transfer, the Mainline phase regains control and loads the Surface Analysis phase to write the ID field on the suspected defective track. The AIOB input/output control block contains the control information required by the Surface Analysis phase (see *Section 4. Data Area Formats* for a description of the control block).

The Surface Analysis phase tests the specified tracks for read or write error conditions, and performs special error recovery procedures in case of surface errors. The SIOB input/output block and the write buffer (pointed to by the address at ABFAD) are used in this procedure. Otherwise, the IOADDR write buffer is used. After surface analysis, control is passed back to the Mainline phase along with a return code indicating the status of the track tested. If the track was defective, the alternate track is assigned and the ID field is set to reflect the assignment.

The Mainline phase uses the IOADDR write buffer and the AIOB control block to write the ID fields on the alternate track. Two tables containing the status of the alternate tracks (ALTIN) and their addresses (ATBLE) are defined by the Mainline phase. If the track was not defective, the Data Transfer phase is called to return data to the primary track. The Mainline phase passes a table of messages (AMSGT) to the Print phase indicating the results of the assignment. If the data transferred was in error, the Data Transfer phase also passes a table (ASTAB) to the Sector Print phase consisting of the sectors and data in error so that the user can check the data. A conditional assignment is automatically performed after each unconditional request.

UNCONDITIONAL ASSIGNMENT

If the user requests an unconditional assignment, the Mainline phase receives the addresses of the tracks to be processed from the table of constants contained in the Root phase (ABER) and writes the ID field on an alternate track. The Data Transfer phase is then loaded to transfer data from the requested track to the alternate track.

After data transfer, the Mainline phase regains control and loads the Surface Analysis phase to write the ID field on the primary track. If data in error is transferred, the Sector Print phase receives a table from the Data Transfer phase to print the data so it may be checked for accuracy. Control is returned to the Mainline phase which passes control to the Print phase to print the results of the assignment.

UNCONDITIONAL CANCEL OF PRIOR ASSIGNMENT

If the user requests an unconditional cancel of prior assignment, the addresses of the tracks to be cancelled are obtained from the table of constants maintained by the Root phase. The Surface Analysis phase is called to test the primary track. The Data Transfer phase is then called to attempt a data transfer to the primary track; it contains a special error recovery procedure and attempts the transfer eight times if not successful the first time. A return code in the ARDTC buffer, is passed to the Mainline phase indicating whether the attempt was successful or not.

If the track is still defective, the user may:

- Force the assignment.
- Retry the transfer.
- Terminate the run.
- Continue to the next request.
- Write the updated volume label after the request has been performed.

Note: The EOJ transient performs end-of-job requirements. The Halt/Syslog transient performs all halts due to errors throughout the program.

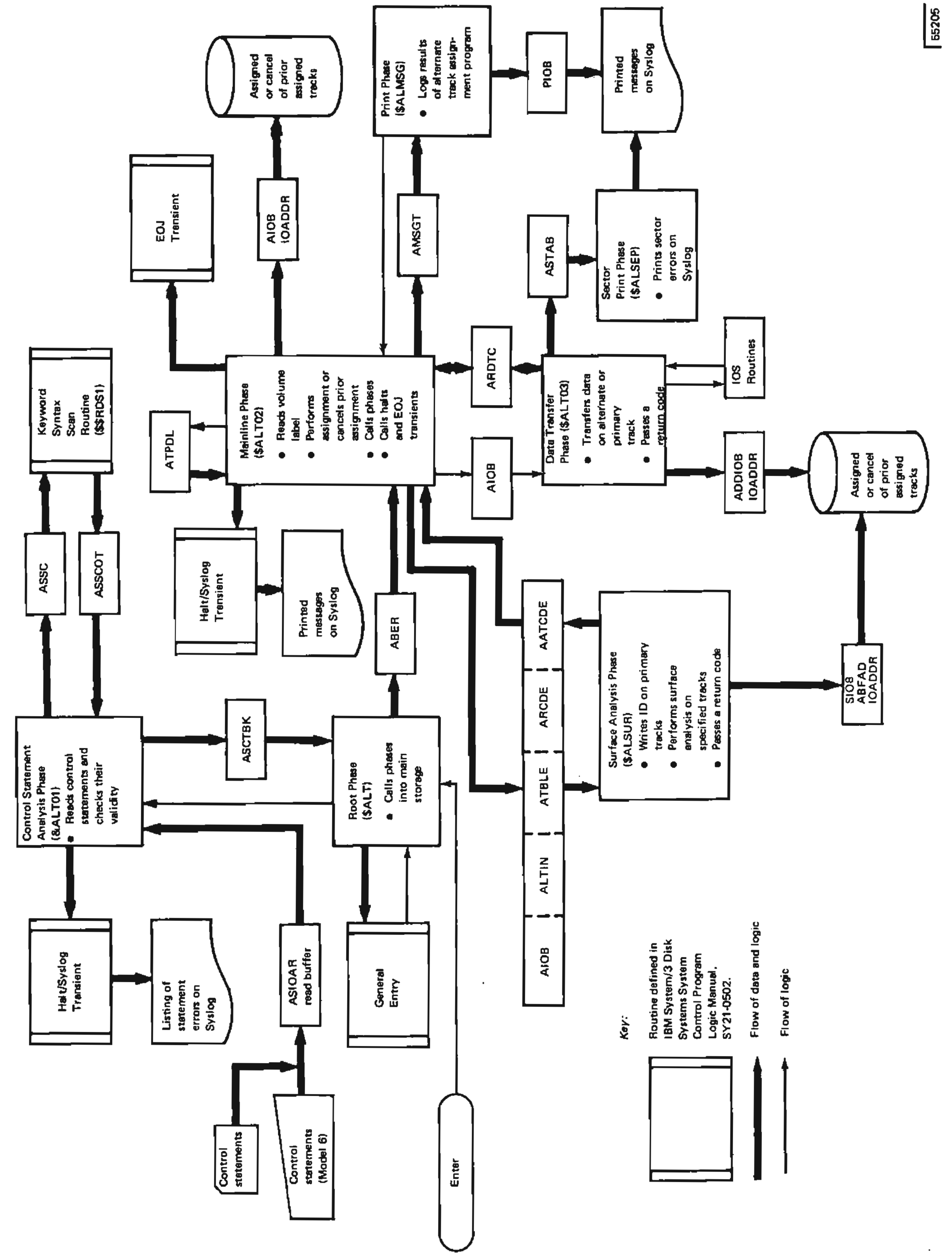

55205
Control statements
Control statements (Model 6)
Enter
ASIOAR read buffer
Halt/Syslog Transient
Listing of statement errors on Syslog
General Entry
Control Statement Analysis Phase ($ALT01)
● Reads control statements and checks their validity
Root Phase ($ALT)
● Calls phases into main storage
ASCTBK
ASSC
ASSCOT
Keyword Syntax Scan Routine ($$RDS1)
ABER
Halt/Syslog Transient
Printed messages on Syslog
ATPDL
Mainline Phase ($ALT02)
● Reads volume label
● Performs assignment or cancels prior assignment
● Calls phases
● Calls halts and EOJ transients
EOJ Transient
AIOB IOADDR
Assigned or cancel of prior assigned tracks
AMSGT
Print Phase ($ALMSG)
● Logs results of alternate track assignment program
PIOB
Printed messages on Syslog
ARDTC
AIOB
Data Transfer Phase ($ALT03)
● Transfers data on alternate or primary track
● Passes a return code
ASTAB
Sector Print Phase ($ALSEP)
● Prints sector errors on Syslog
IOS Routines
ADDIOB IOADDR
Assigned or cancel of prior assigned tracks
AIOB
ALTIN
ATBLE
ARCDE
AATCDE
Surface Analysis Phase ($ALSUR)
● Writes ID on primary tracks
● Performs surface analysis on specified tracks
● Passes a return code
SIOB ABFAD IOADDR
Key:
Routine defined in IBM System/3 Disk Systems System Control Program Logic Manual, SY21-0502.
Flow of data and logic
Flow of logic

Figure 2-1. Operational Diagram for the Alternate Track Assignment Program

Section 3. Program Organization

This section contains a functional description of each phase included in the IBM System/3 Alternate Track Assignment program. Flowcharts are also provided where necessary. Figure 2-2 shows a storage map of the program.

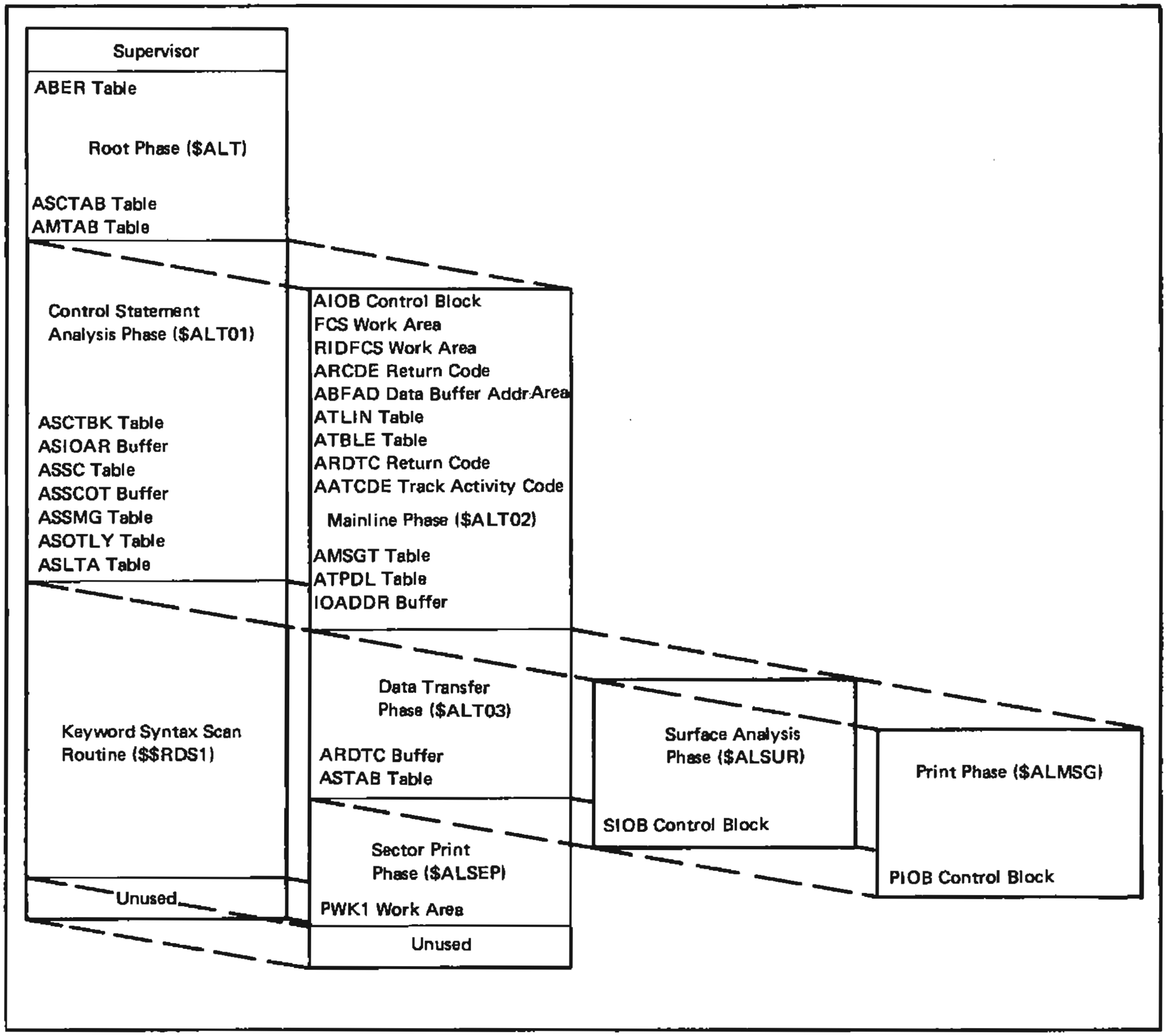


Figure 2-2. Storage Map of the Alternate Track Assignment Program

Root Phase ($ALT)

Entry Point: AROOT

Chart: None

Functions:

- Loads the Control Statement Analysis phase and the Mainline phase into main storage.
- Maintains a table of constants (ABER) built by the Control Statement Analysis phase.

Input: Table of constants (ASCTBK) from the Control Statement Analysis phase ($ALT01)

Output: Table of constants (ABER) to the Mainline phase

Exits:

- Normal
  1. Supervisor to load the phases
  2. Entry point of phase called
- Error – None

Control Statement Analysis Phase ($ALT01)

Entry Point: ASASYN

Chart: BA

Functions:

- Reads in control statements from SYSIN or the job scheduler work area.
- Checks control statements for validity and proper sequence.
- Builds a table of constants (ASCTBK) with information obtained from the Keyword Syntax Scan routine.

Input: Control statements read in from the keyboard, card, disk, or procedure library

Output: Table of constants (ASCTBK) to Root phase

Exits:

- Normal – Root phase
- Error – Halt/Syslog transient

Mainline Phase ($ALT02)

Entry Point: AMAIN

Chart: BB

Functions:

- Reads the volume label of the disk pack to be processed.
- Performs one of the following requests:

1. Conditional assignment:

   a. Removes any duplicate entries from the conditional request list (ATPDL) in the volume label.

   b. Passes information to the Data Transfer phase to transfer data from a primary track to an alternate track and checks the return code (ARDTC).

   c. Passes information (AATCDE) to the Surface Analysis phase to perform surface analysis on the primary track and checks the return code (ARCDE).

   d. Passes information to the Data Transfer phase to transfer data back to the primary track if the primary track was not defective.

   e. Assigns the alternate track to the defective track if the primary track was defective.

   f. Determines whether an assigned alternate track is defective. If it is, an attempt is made to assign another track.

   g. Updates alternate track information in the volume label.

   h. Passes a table of messages (AMSGT) to the Print phase.

2. Unconditional assignment:

   a. Passes information to the Data Transfer phase to transfer data from a suspected defective track to an alternate track and checks the return code (ARCDE).

   b. Assigns an alternate track to the suspected defective track.

   c. Determines whether an assigned alternate track is defective. If it is, an attempt is made to assign another alternate track.

   d. Updates the alternate track information in the volume label.

   e. Passes a table of messages (AMSGT) to the Print phase.

3. Unconditional cancel of prior assignment:

   a. Determines if the request is valid. (There must have been a previous assignment if there is to be a cancel).

   b. Passes information to the Data Transfer phase to transfer data back to the primary track and checks the return code (ARCDE).

   c. Updates the alternate track information.

   d. Passes a table of messages (AMSGT) to the Print phase. If the track is still defective the user may (1) force the cancel of prior assignment, (2) retry the transfer, (3) terminate the run, (4) continue to next request.

Input:

- Table of constants (ABER) from the Root phase
- Return code from the Surface Analysis phase
- Return code from the Data Transfer phase

Output:

- Control information to the Data Transfer phase and the Surface Analysis phase
- Table of messages (AMSGT) to the Print phase (ALMSG)
- XR1 pointing to AIOB for Data Transfer phase and Surface Analysis phase

Exits:

- Normal
  1. Surface Analysis phase
  2. Data Transfer phase
  3. Print phase
  4. EOJ transient
- Error – Halt/Syslog transient

Data Transfer Phase ($ALT03)

Entry Point: ADATX

Chart: BC

Functions:

- Transfers data by sector from a primary track to alternate track. This phase attempts to transfer data eight times if the first transfer is not successful and a hardware error has not occurred.
- Returns data to the primary track.
- Passes a table (ASTAB) to the Sector Print phase ($ALSEP) to print sector in error.

Input: XR1 points to input/output block AIOB in the Mainline phase

Output:

- Return code to Mainline phase
- Message to Sector Print phase

Exits:

- Normal
  1. Mainline phase
  2. Sector Print phase
- Error – Halt/Syslog transient

Surface Analysis Phase ($ALSUR)

Entry Point: SURFAN

Chart: BD

Functions:

- Writes and verifies sector addresses and test patterns (X'55' and X'00') on specified tracks.
- Provides a special error recovery procedure for surface errors; tries writing data ten times.
- Sets the flag byte of a defective track to indicate that it is defective.
- Performs surface analysis and indicates if the primary track is defective for conditional assignment and unconditional cancel of prior assignment.
- Writes ID field on a defective primary track to indicate an assignment.
- Passes a return code to the Mainline phase:

  X'10' – No errors

  X'11' – Unusable pack

  X'12' – Alternate track defective

  X'14' – Hardware error encountered

  X'18' – Valid alternate track assigned

Input:

- XR1 points to input/output block AIOB defined by the Mainline phase
- XR2 points to input/output buffer IOADDR

Output: Return code to Mainline phase

Exits:

- Normal – AERCHK in the Mainline phase
- Error – None

Print Phase ($ALMSG)

Entry Point: ALMSG

Chart: BE

Functions:

- Prints the results of a successful assignment and the addresses of the primary track and the assigned alternate track.
- Prints the results of an unconditional assignment.
- Prints the results of an unconditional cancel of prior assignment.

Input: Table of messages (AMSGT) from the calling phase

Output: Printed messages

Exits:

- Normal – Mainline phase
- Error – None

Sector Print Phase ($ALSEP)

Entry Point: ALSEP

Chart: BF

Functions: Prints the address of the sector in error and the data on that sector if a data error occurred during an alternate track assignment.

Input: Table (ASTAB) from the Data Transfer phase

Output: Sector in error is printed

Exits:

- Normal – Data Transfer phase
- Error – None

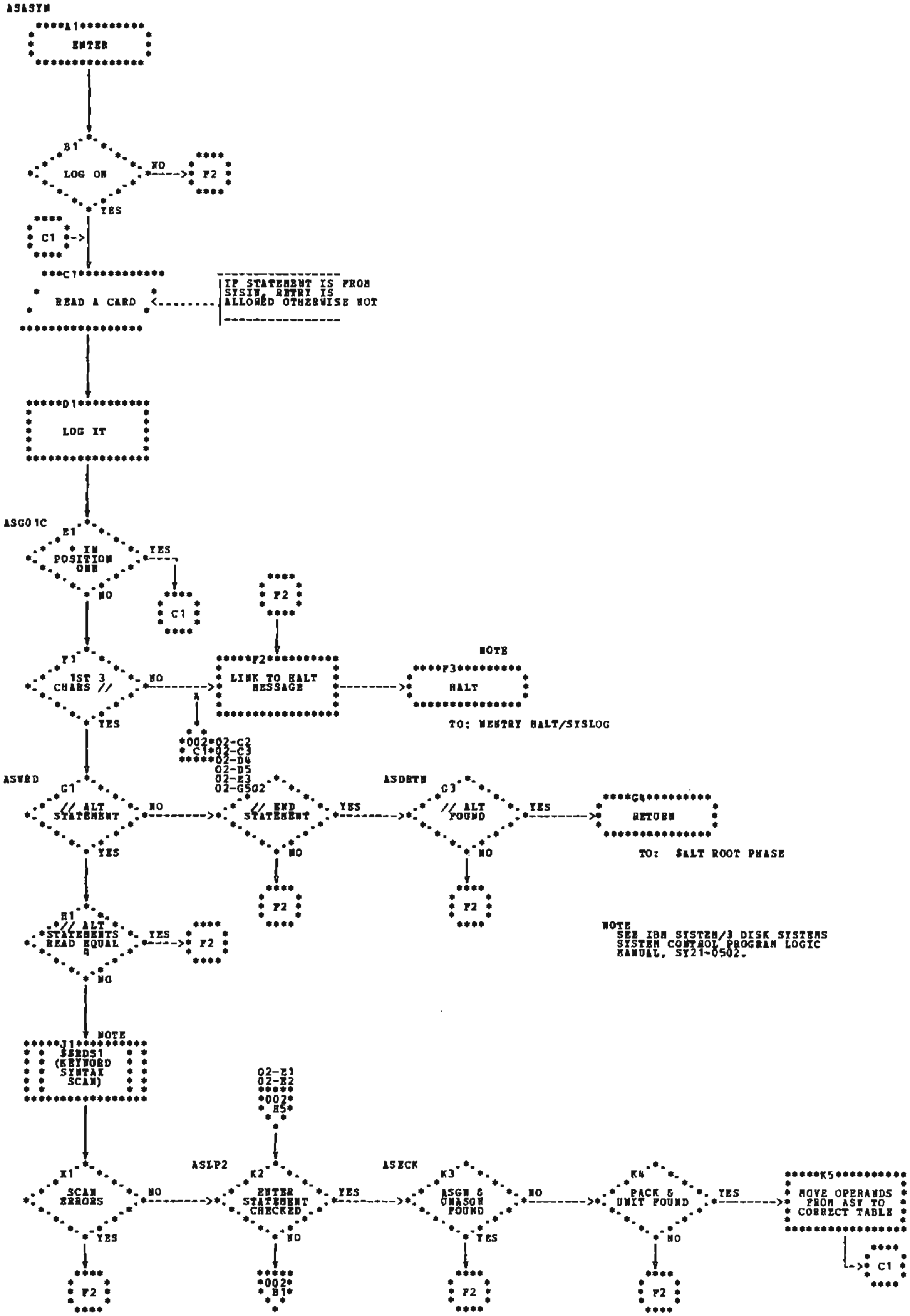


Chart BA (Part 1 of 2). Control Statement Analysis Phase

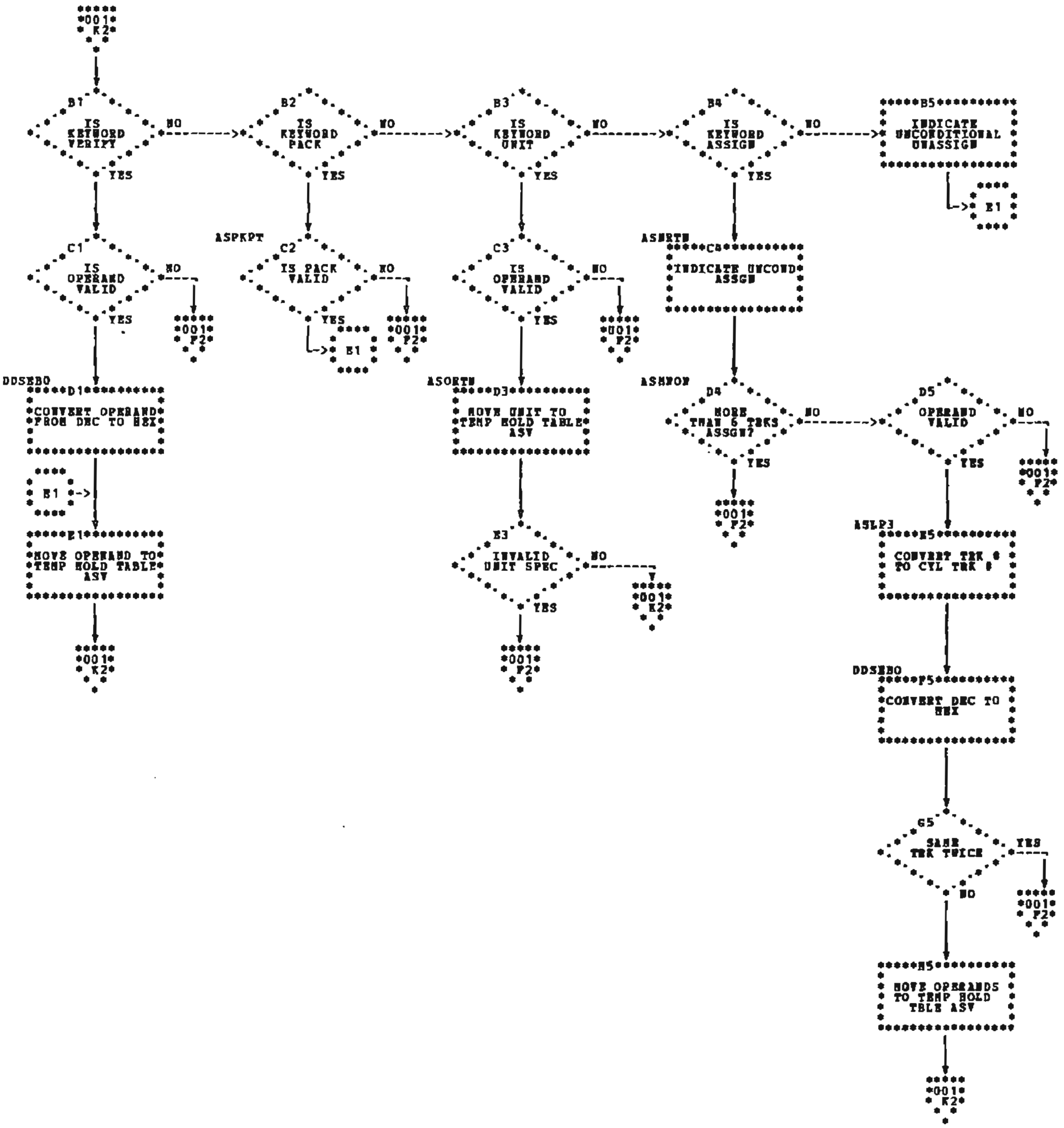


Chart BA (Part 2 of 2). Control Statement Analysis Phase

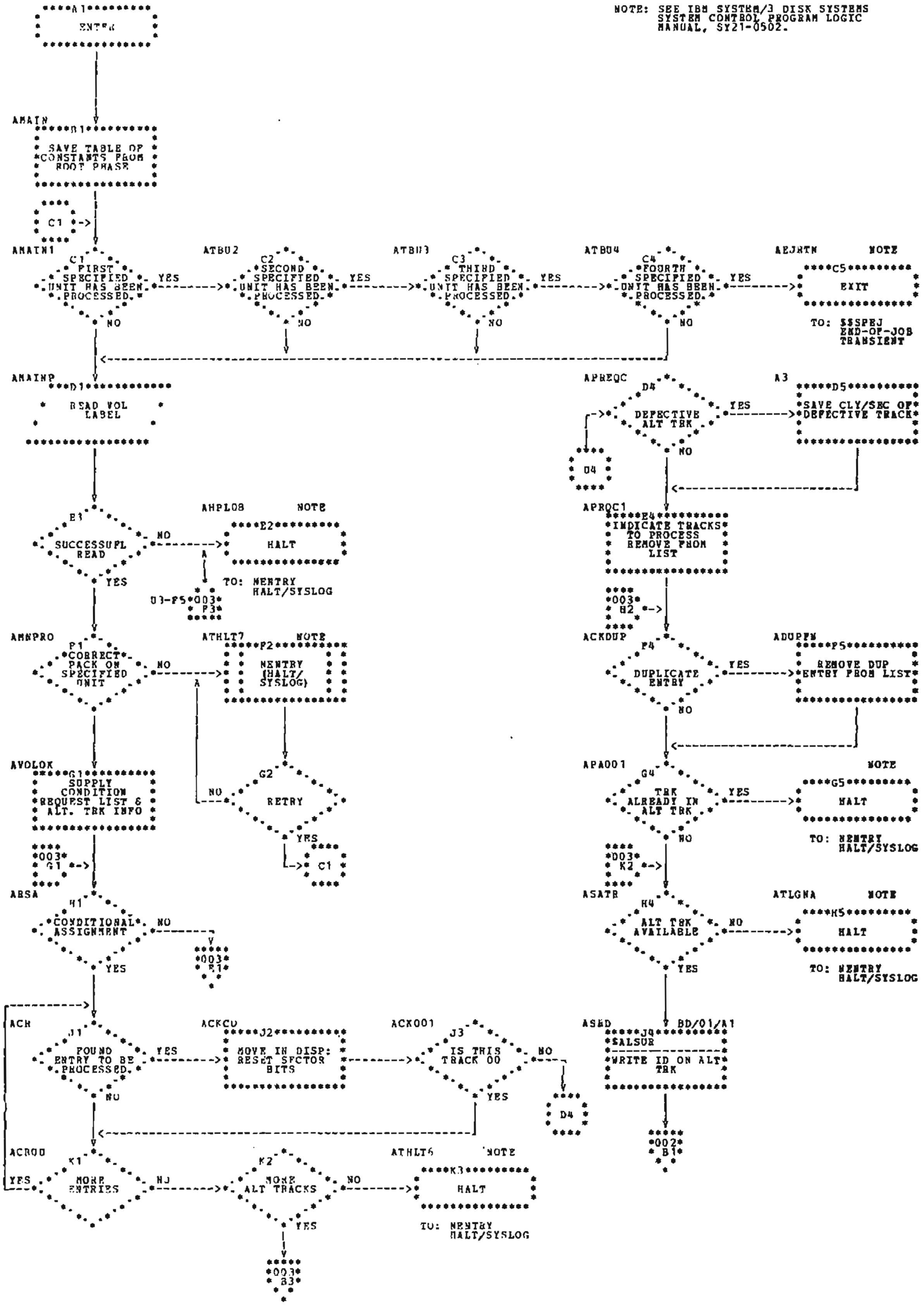
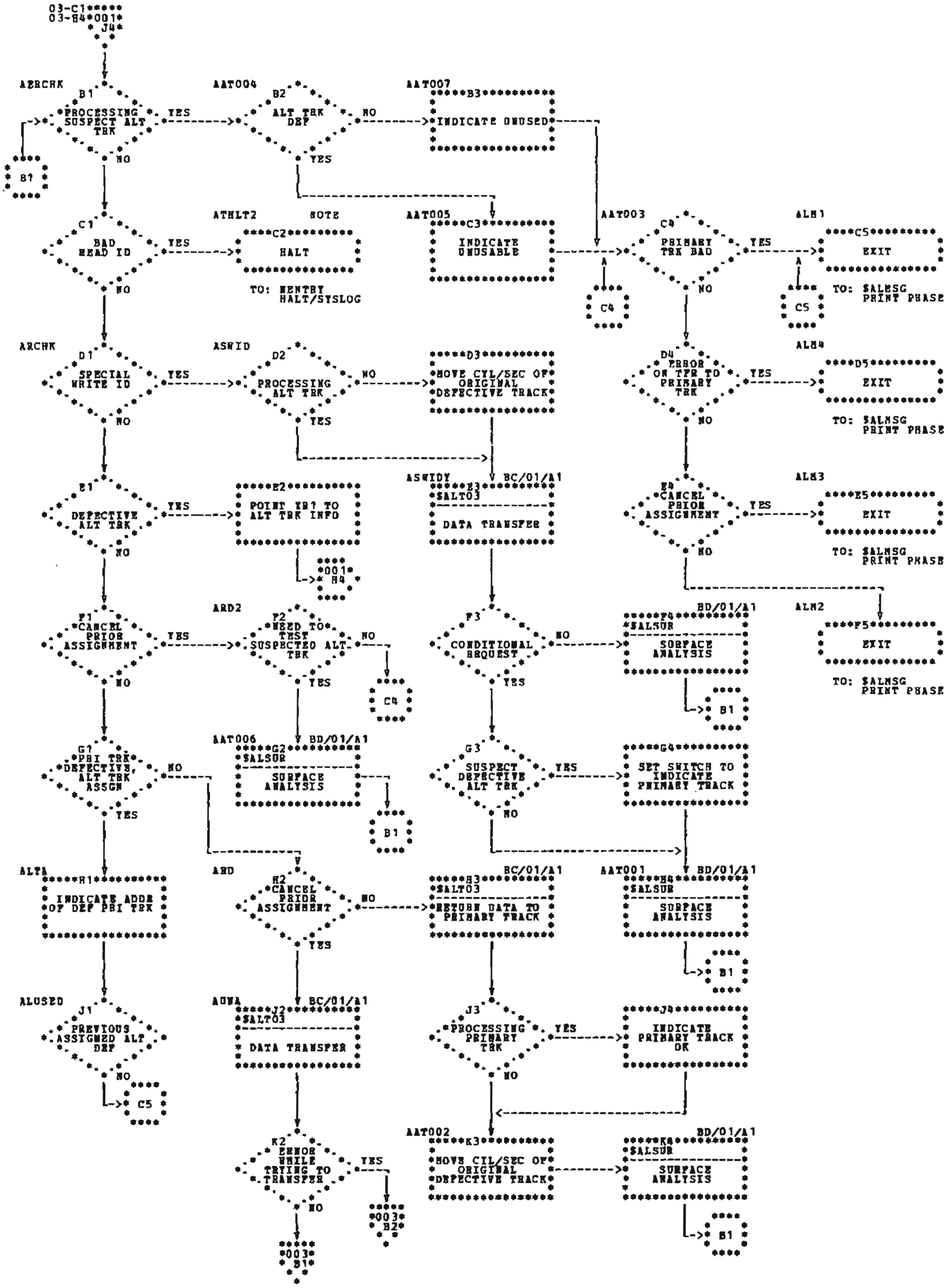


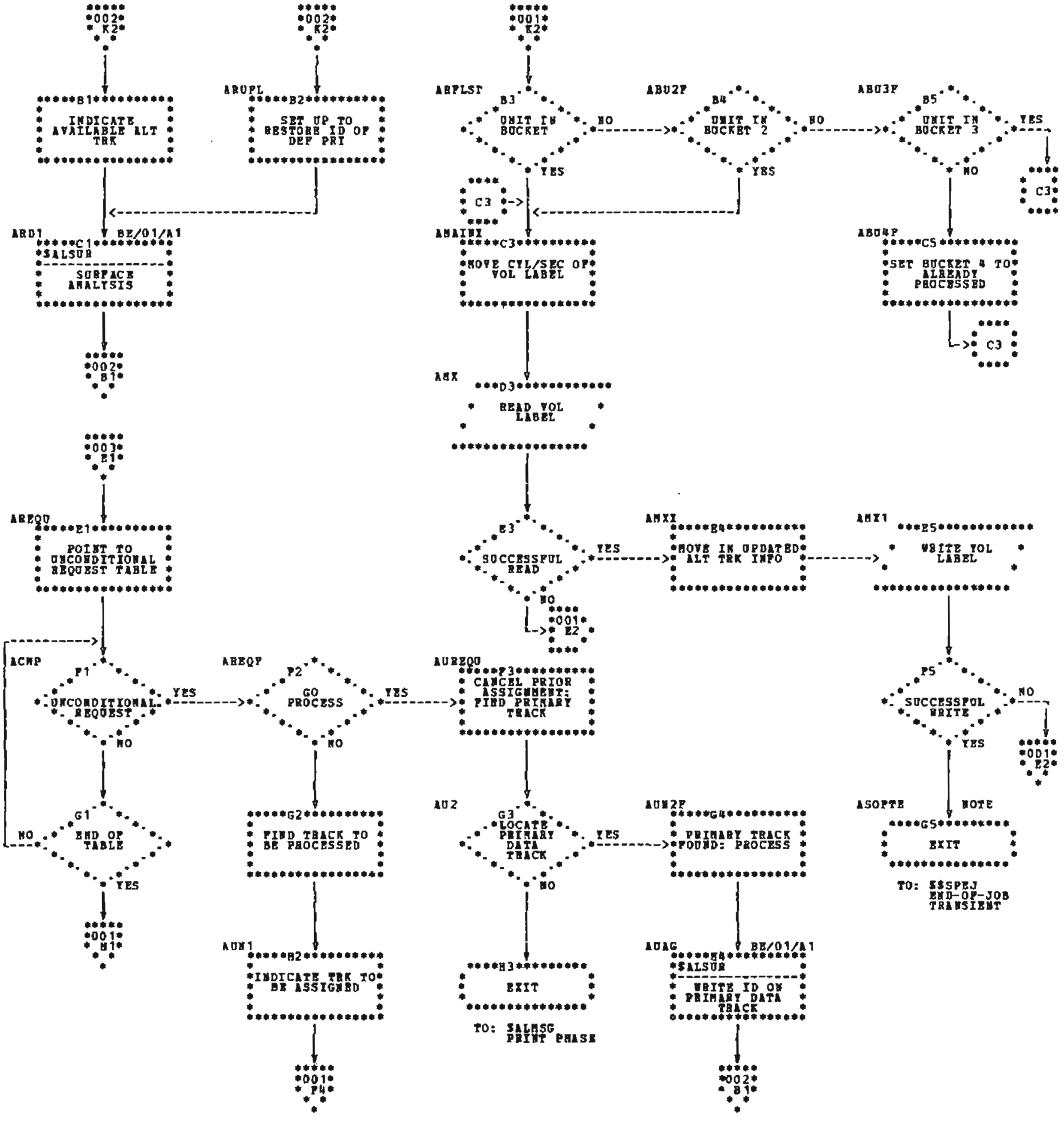
Chart BB (Part 1 of 3). Mainline Phase

NOTE: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.



03-C1
03-H4 001 J4
AERCHK B1 PROCESSING SUSPECT ALT TRK YES NO
B1
AAT004 B2 ALT TRK DEF NO YES
AAT007 B3 INDICATE UNUSED
C1 BAD READ ID YES NO
ATHLT2 NOTE C2 HALT
TO: RENTRY HALT/SYSLOG
AAT005 C3 INDICATE UNUSABLE
C4
AAT003 C4 PRIMARY TRK BAD YES NO
ALN1 C5 EXIT
C5
TO: $ALMSG PRINT PHASE
ARCHK D1 SPECIAL WRITE ID YES NO
ASWID D2 PROCESSING ALT TRK NO YES
D3 MOVE CYL/SEC OF ORIGINAL DEFECTIVE TRACK
D4 ERROR ON TFR TO PRIMARY TRK YES NO
ALN4 D5 EXIT
TO: $ALMSG PRINT PHASE
E1 DEFECTIVE ALT TRK YES NO
E2 POINT XR1 TO ALT TRK INFO
001 H4
ASWIDY BC/01/A1 E3 $ALT03 DATA TRANSFER
E4 CANCEL PRIOR ASSIGNMENT YES NO
ALN3 E5 EXIT
TO: $ALMSG PRINT PHASE
F1 CANCEL PRIOR ASSIGNMENT YES NO
ARD2 F2 NEED TO TEST SUSPECTED ALT TRK NO YES
C4
F3 CONDITIONAL REQUEST NO YES
BD/01/A1 F4 $ALSUR SURFACE ANALYSIS
B1
ALN2 F5 EXIT
TO: $ALMSG PRINT PHASE
G1 PRI TRK DEFECTIVE, ALT TRK ASSGN NO YES
AAT006 BD/01/A1 G2 $ALSUR SURFACE ANALYSIS
B1
G3 SUSPECT DEFECTIVE ALT TRK YES NO
G4 SET SWITCH TO INDICATE PRIMARY TRACK
ALTA H1 INDICATE ADDR OF DEF PRI TRK
ARD H2 CANCEL PRIOR ASSIGNMENT NO YES
BC/01/A1 H3 $ALT03 RETURN DATA TO PRIMARY TRACK
AAT001 BD/01/A1 H4 $ALSUR SURFACE ANALYSIS
B1
ALUSED J1 PREVIOUS ASSIGNED ALT DEF NO
C5
ADNA BC/01/A1 J2 $ALT03 DATA TRANSFER
J3 PROCESSING PRIMARY TRK YES NO
J4 INDICATE PRIMARY TRACK OK
K2 ERROR WHILE TRYING TO TRANSFER YES NO
003 B2
003 B1
AAT002 K3 MOVE CYL/SEC OF ORIGINAL DEFECTIVE TRACK
BD/01/A1 K4 $ALSUR SURFACE ANALYSIS
B1

Chart BB (Part 2 of 3). Mainline Phase



NOTE: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart BB (Part 3 of 3). Mainline Phase

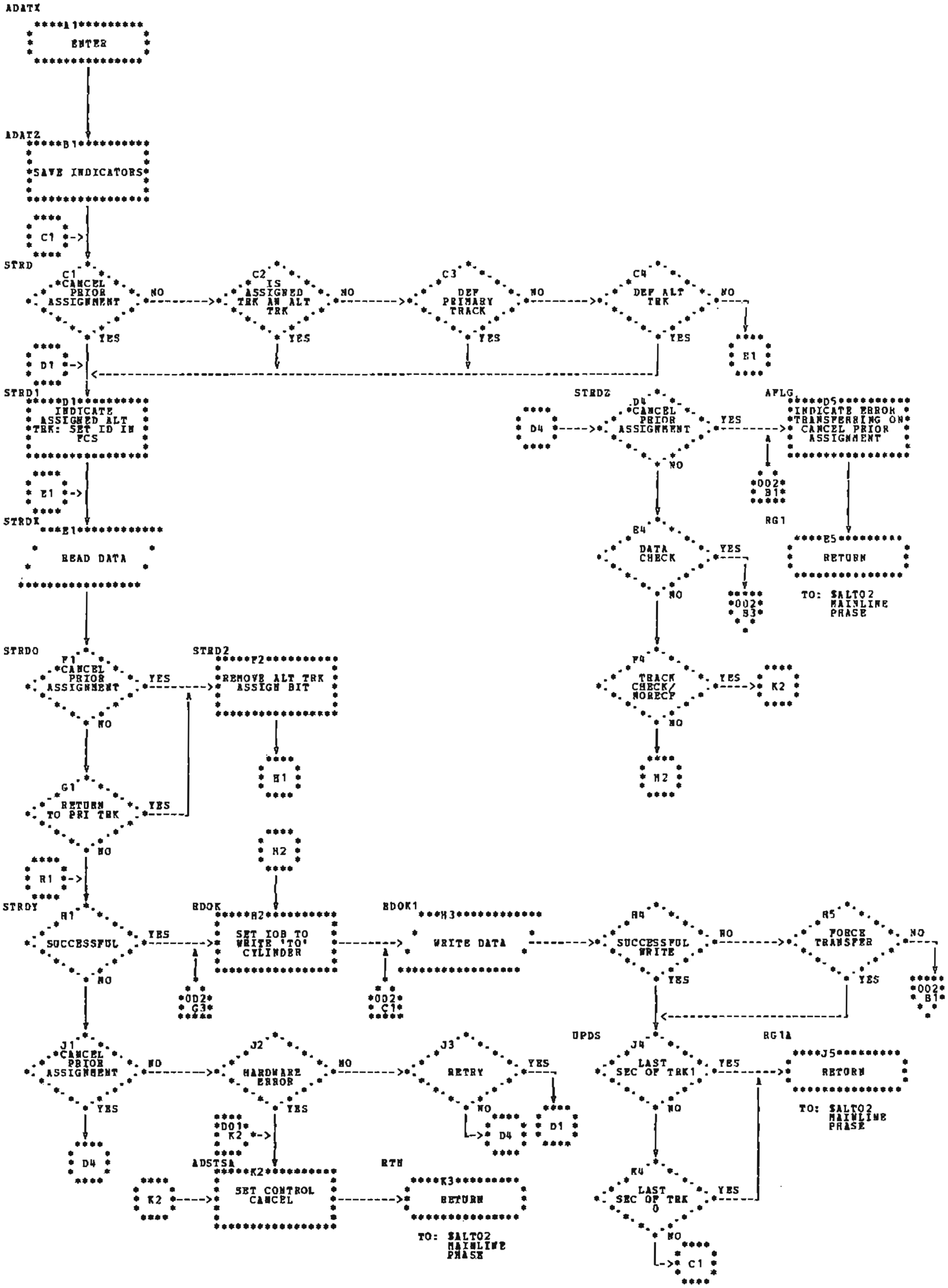


Chart BC (Part 1 of 2). Data Transfer Phase

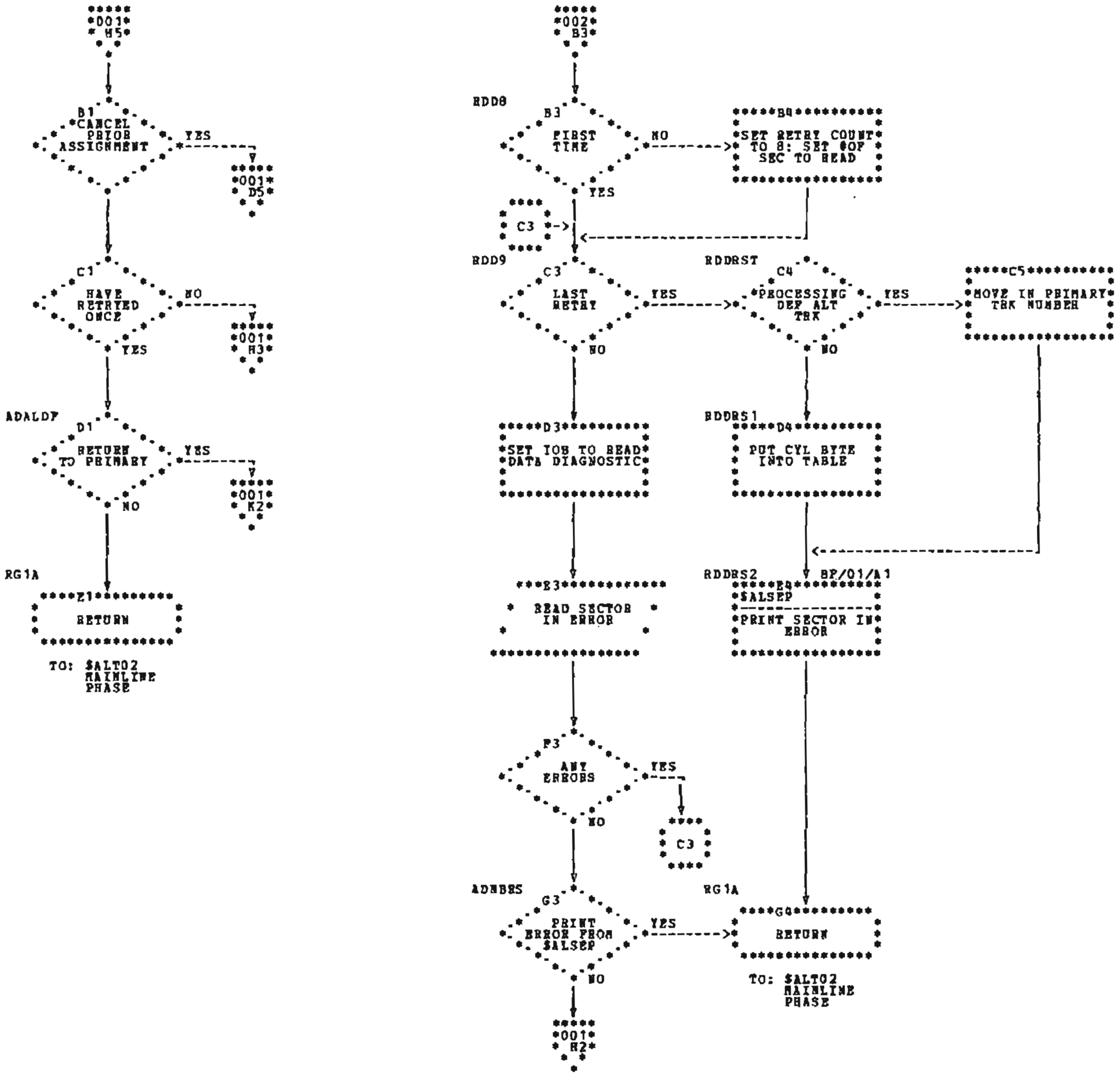


Chart BC (Part 2 of 2). Data Transfer Phase

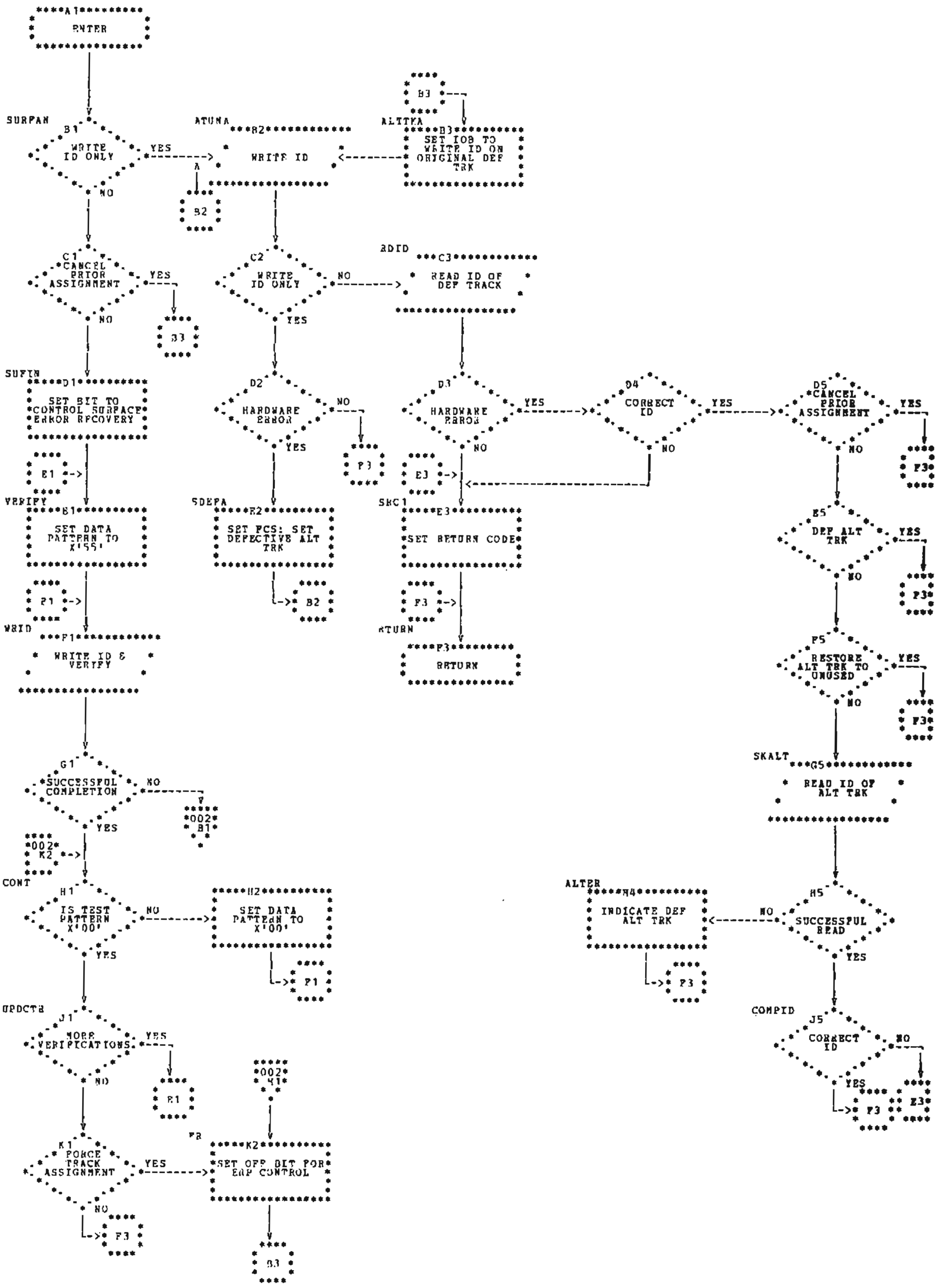


Chart BD (Part 1 of 2). Surface Analysis Phase

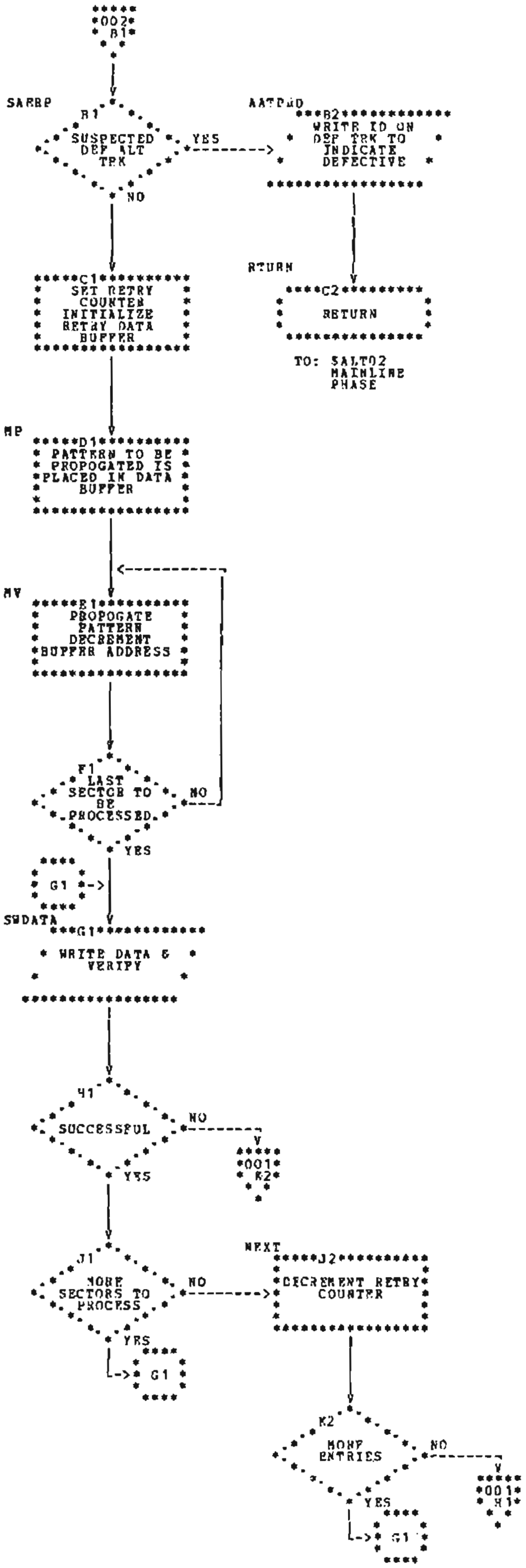
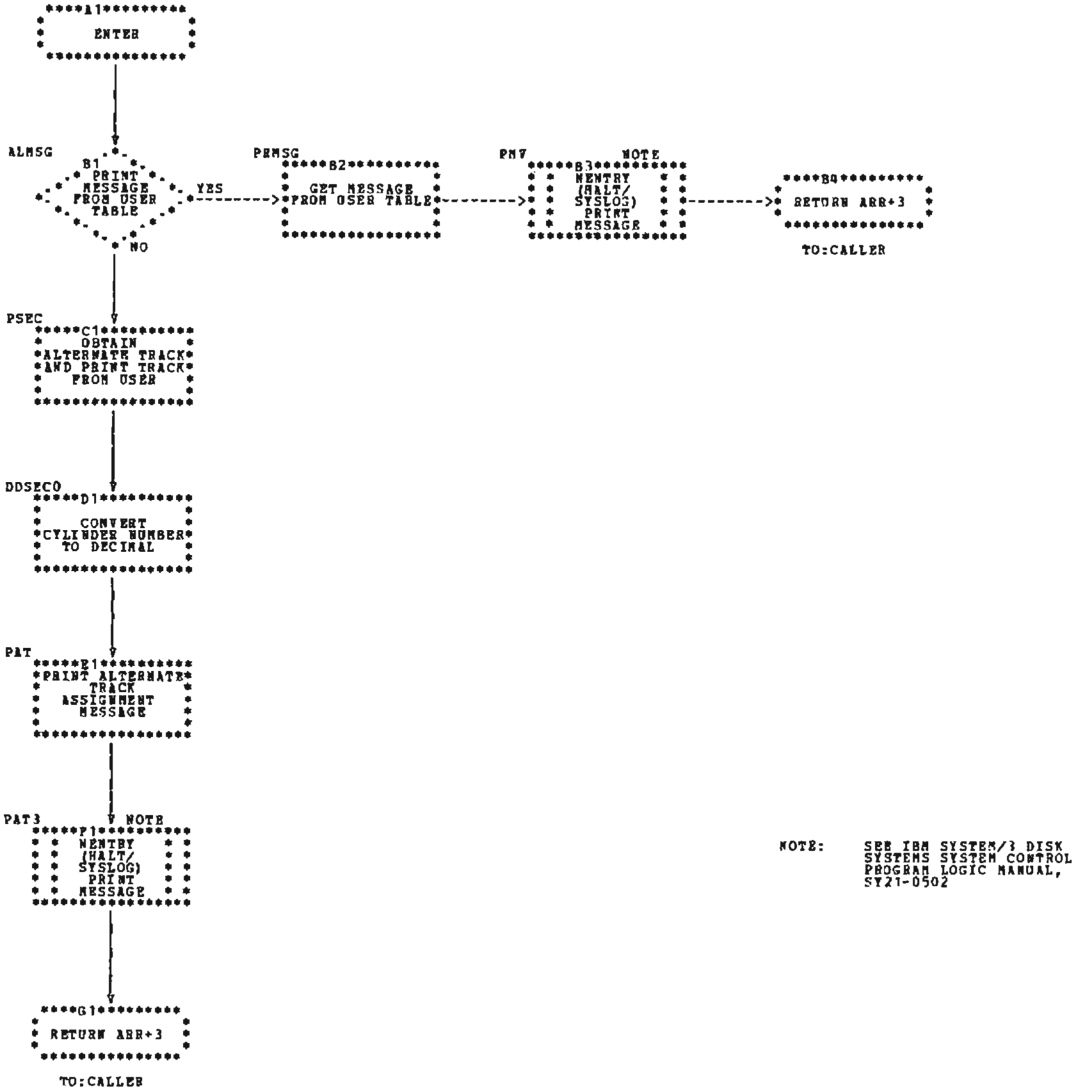
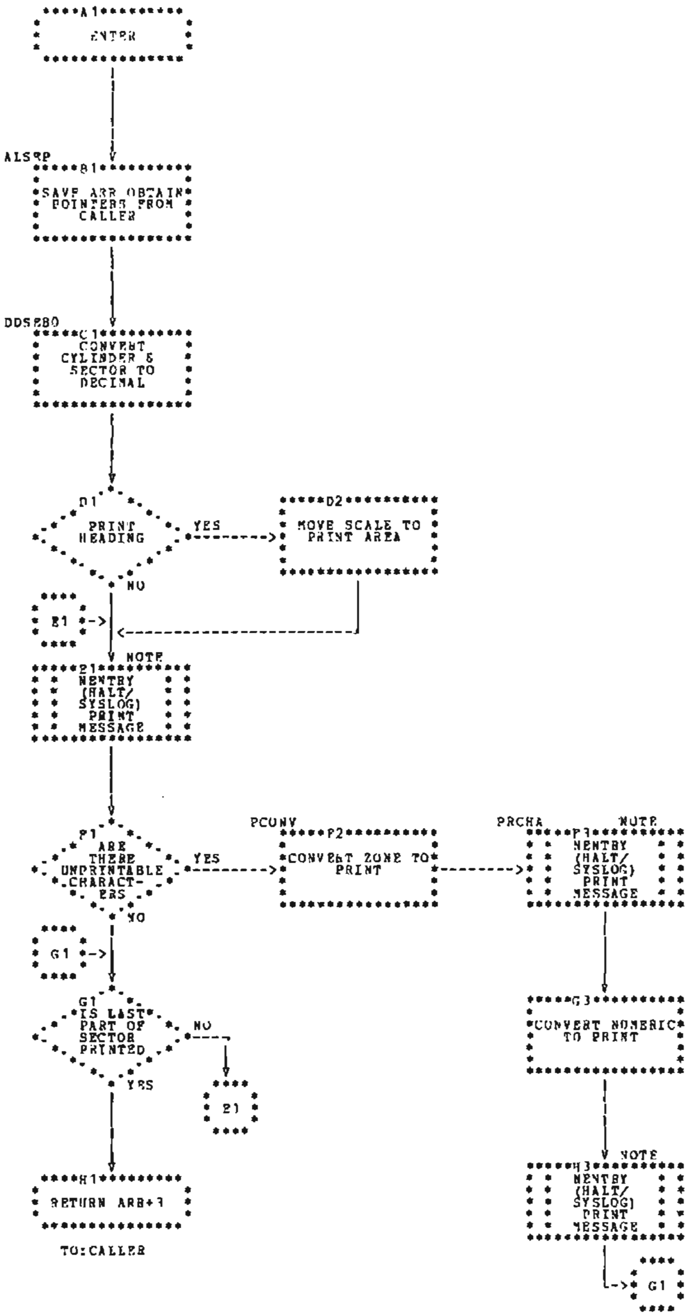

002
B1
SARRP
B1
SUSPECTED DEF ALT TRK
YES
NO
AATPRD
B2
WRITE ID ON DEF TRK TO INDICATE DEFECTIVE
RTURN
C2
RETURN
TO: SALT02 MAINLINE PHASE
C1
SET RETRY COUNTER INITIALIZE RETRY DATA BUFFER
MP
D1
PATTERN TO BE PROPOGATED IS PLACED IN DATA BUFFER
MV
E1
PROPOGATE PATTERN DECREMENT BUFFER ADDRESS
F1
LAST SECTOR TO BE PROCESSED
NO
YES
G1
SWDATA
G1
WRITE DATA & VERIFY
H1
SUCCESSFUL
NO
YES
001
K2
J1
MORE SECTORS TO PROCESS
NO
YES
G1
NEXT
J2
DECREMENT RETRY COUNTER
K2
MORE ENTRIES
NO
YES
001
H1
G1

Chart BD (Part 2 of 2). Surface Analysis Phase


A1
ENTER
ALMSG
B1
PRINT MESSAGE FROM USER TABLE
YES
NO
PRMSG
B2
GET MESSAGE FROM USER TABLE
PN7
NOTE
B3
MENTRY (HALT/ SYSLOG) PRINT MESSAGE
B4
RETURN ARR+3
TO:CALLER
PSEC
C1
OBTAIN ALTERNATE TRACK AND PRINT TRACK FROM USER
DDSECO
D1
CONVERT CYLINDER NUMBER TO DECIMAL
PAT
E1
PRINT ALTERNATE TRACK ASSIGNMENT MESSAGE
PAT3
NOTE
F1
MENTRY (HALT/ SYSLOG) PRINT MESSAGE
G1
RETURN ARR+3
TO:CALLER

NOTE: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502

Chart BE. Print Phase


A1
ENTER
ALSRP
B1
SAVE ARR OBTAIN POINTERS FROM CALLER
DDSEBO
C1
CONVERT CYLINDER & SECTOR TO DECIMAL
D1
PRINT HEADING
YES
NO
D2
MOVE SCALE TO PRINT AREA
E1
NOTE
E1
NENTRY (HALT/ SYSLOG) PRINT MESSAGE
F1
ARE THERE UNPRINTABLE CHARACT- ERS
YES
NO
PCONV
F2
CONVERT ZONE TO PRINT
PRCHA
NOTE
F3
NENTRY (HALT/ SYSLOG) PRINT MESSAGE
G1
G1
IS LAST PART OF SECTOR PRINTED
NO
YES
E1
G3
CONVERT NUMERIC TO PRINT
NOTE
H3
NENTRY (HALT/ SYSLOG) PRINT MESSAGE
G1
H1
RETURN ARR+3
TO:CALLER

NOTE: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502

Chart BF. Sector Print Phase

This section describes the format and contents of various control blocks, tables, and other data areas used by the IBM System/3 Alternate Track Assignment program. See the storage map in Figure 2-2 for the relative location of these data areas.

CONTROL BLOCKS

The Alternate Track Assignment program requires two input/output control blocks:

1. AIOB
2. SIOB

The AIOB, defined by the Mainline phase, is used to read cylinder 0, track 0, sector 2 and perform surface analysis. It is passed to the Surface Analysis phase and Data Transfer phase which use the control information. A byte labeled IOBFLG can be set to the following indications:

- Bit 0 – The program will handle its own error recovery procedures.
- Bit 2 – No error logging.
- Bit 3 – Data area FCS will be used for comparisons during the disk IOS SCAN instruction and that FSC will contain the ID to write for a WRITE ID instruction.
- Bit 4 – The program does not use data management.

The SIOB, defined by the Surface Analysis phase, is used to write data during surface error recovery procedures.

Byte positions and contents for the input/output blocks labeled AIOB and SIOB are:

Byte	*Contents*
0-1	Chain pointer used and updated by IOS to queue requests
2	Completion code
3	Q byte indicating operation
4	R control byte
5	Cylinder address in hexadecimal
6	Sector address in hexadecimal
7	Number of sectors involved in operation (N-1)
8-9	Data area address
10-11	Sense bytes used by IOS
12	Error counter used by IOS
13	Flag byte X‘80’ = No error recovery X‘40’ = No verification X‘20’ = No error logging X‘10’ = Write ID operation X‘08’ = No DTF X‘04’ = No load I/O for disk file date register X‘02’ = Equipment check X‘01’ = Alternate track in process
14-15	Register 1 (XR1) save area
16-17	ARR save area

TABLES

ASCTBK Table

ASCTBK, an 84-byte table of constants, is passed by the Control Statement Analysis phase to the Root phase. The Root phase maintains the table under the label ABER. The table is comprised of 21-bytes of information for each // ALT statement. (There is a maximum of four // ALT statements in the input stream.) The 21-byte sections of the table are labeled ASV1, ASV2, ASV3, and ASV. ASV is also used as a temporary hold area before moving the constants into the other sections of the table. Each section contains such information as the number of requested verifications, the type of assignment, the unit and volume serial numbers of the packs to be processed, and the track numbers requiring unconditional processing.

ABER Table

ABER, an 84-byte table located in the Root phase and passed to the Mainline phase, maintains the parameters initialized by the Control Statement Analysis phase under the label ASCTBK.

ASSC Table

ASSC is an 8-byte table of parameters passed to the Keyword Syntax Scan routine by the Control Statement Analysis phase. The byte positions and contents are:

Byte	*Contents*
0-1	Address of where to start scan
2-3	Address of where to end scan
4-5	Address of scan output area (ASSCOT)
6-7	Address of keyword parameter table (ASLTA) for the // ALT statement

ASCTAB Table

ASCTAB is a 10-byte table used by the Root phase to load the Control Statement Analysis phase into main storage. The byte positions and contents are:

Byte	*Contents*
0	C'O' – Object library
1-6	C'$ALT01' – Program name to call
7	X'00'
8-9	Address of where to put the object program (ALTEND)

AMTAB Table

AMTAB is a 10-byte table used by the Root phase to load the Mainline phase into main storage. The byte positions and contents are:

Byte	*Contents*
0	C'O' – Object library
1-6	C'$ALT02' – Program name to call
7	X'00'
8-9	Address of where to put the object program (ALTEND)

ASLTA Table

ASLTA is a 35-byte table containing the keyword parameters of the // ALT card defined in the Control Statement Analysis phase and used by the Keyword Syntax Scan routine. The byte positions and contents are:

Byte	*Contents*
0	X'16'
1-6	'VERIFY'
7	X'24'
8-11	'PACK'
12	X'34'
13-16	'UNIT'
17	X'46'
18-23	'ASSIGN'
24	X'58'
25-32	'UNASSIGN'
33-34	X'FFFF'

ASSMG Table

ASSMG is a table of halt messages to be logged on the Halt/Syslog transient (Figure 2-3).

Byte	**Label**	**Contents**
0	ASSMG	B'11010110' – Indicates halt and system print
1	None	C'A' – Indicates Alternate Track Assignment program
2-3	ASHMSG	Disk System – Characters to be used for halt display lights Model 6 – Code to be converted for halt display lights
4	None	Operator options X'01' = Immediate cancel X'02' = Control cancel X'04' = Retry X'08' = Proceed
5-6	ASUBMG	C'ƀƀ' – Indicates submessage area

Figure 2-3. ASSMG Table

ASOTLY Table

ASOTLY is a halt table used by the Control Statement Analysis phase to print control statements that are in error. The byte positions and contents are:

Byte	*Contents*
0-1	X'0000' – Indicates output only; log control statement
2	X'60' – Indicates 96-byte length
3-4	Address of I/O area (ASIOAR)

ALTIN Table

This 12-byte table contains two bytes of information concerning each of the six alternate tracks. It is initialized to zeros by the Mainline phase and may be modified by the Surface Analysis phase. Each of the 2-byte areas within the table contains one of the following hexadecimal configurations:

Contents	*Meaning*
X'00XX'	Alternate track is available for assignment
X'FFXX'	Alternate track is defective and may not be used
X'NNNN'	Address of the defective original track to which the alternate track is assigned

ATBLE Table

This 12-byte table, used by the Mainline phase, contains the addresses of the alternate tracks. The byte positions and contents are:

Byte	*Contents*
0-1	X'0100' = Track number one (upper)
2-3	X'0180' = Track number one (lower)
4-5	X'0200' = Track number two (upper)
6-7	X'0280' = Track number two (lower)
8-9	X'0300' = Track number three (upper)
10-11	X'0380' = Track number three (lower)

ATPDL Table

This 24-byte area is a conditional assignment request list. It contains a 2-byte entry for each request. X'NNNN' is the cylinder and sector of the suspected bad track. X'FFFF' indicates no conditional assignment has been requested.

AMSGT Table

AMSGT is a table of messages sent to the Print phase (ALMSG) by the Mainline phase ($AST02).

ASTAB Table

The ASTAB table is a 6-byte table used by the Data Transfer phase to pass the location of a defective sector to the Sector Print phase.

BUFFERS

ABFAD Data Buffer Address Area

ABFAD is a 2-byte area which contains the address of a data buffer (I/O work area) to be used during the error recovery procedure of the Surface Analysis phase.

IOADDR Buffer

IOADDR is a 256-byte data buffer used by the Mainline phase, the Surface Analysis phase, and the Data Transfer phase to write and verify data on disk.

ASSCOT Buffer

ASSCOT is a 96-byte output area for the Keyword Syntax Scan routine.

ASIOAR Buffer

ASIOAR is a 96-byte input buffer used by the Control Statement Analysis phase to read the control statement images.

CODES

AATCDE Track Activity Code

AATCDE is a 1-byte area containing a track activity code. The code is passed from the Mainline phase to the Surface Analysis phase. The code contains one of the following hexadecimal configurations:

Contents	*Meaning*
X'01'	Suspected defective alternate track in process
X'02'	Processing primary track that was assigned to the alternate track
X'04'	Primary track all right
X'08'	Processing suspected alternate track
X'10'	Alternate track defective
X'40'	Primary track still defective

ARCDE Return Code

ARCDE, a 1-byte data area initialized to X'10', contains a return code passed between the Surface Analysis phase and the Mainline phase. The possible codes are:

Code	*Meaning*
X'10'	No errors
X'11'	Unusable pack
X'12'	Alternate track defective
X'14'	Hardware error encountered
X'18'	Valid alternate track assigned

ARDTC Return Code

ARDTC, a 1-byte area initialized to zero, contains a return code passed between the Data Transfer phase and the Mainline phase. The return code indicates whether an attempt to transfer data to the primary track was successful or not. The possible codes are:

Code	*Meaning*
X'80'	Force unassign
X'40'	Defective alternate track
X'10'	Error in transferring data to primary track
X'08'	Bad read ID
X'02'	Error in transferring data to alternate track
X'01'	Return to primary track

WORK AREAS

FCS Work Area

The FCS work area, located in the Mainline phase immediately following the control block IOB, is a 3-byte area initialized to zeros. It is comprised of a flag byte, a cylinder address, and a sector address. Bit 3 of the IOBFLG byte in the IOB indicates that FCS is to be used during the IOS SCAN instruction. The IOB also contains the cylinder and sector address of the primary track on which a write or read operation is to occur. The FCS contains the address of the assigned alternate track. In this case the FCS flag byte contains a X'01' to indicate that the assigned track is an alternate track (X'02' indicates a defective primary track; X'03' indicates a defective alternate track).

RIDFCS Work Area

The RIDFCS work area, located in the Mainline phase following the FCS work area, is a 3-byte area initialized to zeros. During a read ID operation, IOS will place the address of the track that was actually read (either a primary track or an alternate track) in this data area.

PWK1 Work Area

PWK1 is a 96-byte work area used by the Sector Print phase to convert unprintable characters to printable characters.

PART III
ALTERNATE TRACK REBUILD

The IBM System/3 Alternate Track Rebuild program corrects data in error which was transferred from a defective track to an alternate track by the IBM System/3 Alternate Track Assignment program.

The Alternate Track Rebuild program obtains the location of the incorrect data from the // REBUILD control statement. If no alternate track was assigned to the suspected defective track, no data may be entered and a halt occurs. If an alternate track was assigned, the program replaces the incorrect data with hexadecimal data provided by the user on data statements following the control statement.

A // PATCH control statement may be used in the same manner as the // REBUILD control statement; however, no check is made for an alternate track assignment. The // PATCH statement has the same format and parameters as the // REBUILD statement.

SYSTEM REQUIREMENTS

The IBM System/3 Alternate Track Rebuild program uses the following minimum system configuration:

Disk System

- IBM 5410 Processing Unit, Model A12
- IBM 5203 Printer, Model A1
- IBM 5424 Multi-Function Card Unit, Model A
- IBM 5444 Disk Storage Drive, Model 2

Model 6

- IBM 5406 Processing Unit, Model B2 (Includes the keyboard with eight command keys)
- IBM 5213 Printer, Model 1
- IBM 5444 Disk Storage Drive, Model 2

The following optional features are supported:

- IBM 5496 Data Recorder, Model 1 with System/3 Model 6 On-line Feature (Used as the Sysin device)
- IBM 2265 Display Station, Model 2 (Cathode Ray Tube; used as Syslog device with IBM 5406 Processing Unit, Model B3 and B4 only)

This section contains a functional description of each phase included in the IBM System/3 Alternate Track Rebuild program. Figure 3-1 shows a storage map of the program.

Mainline Phase ($BUILD)

Entry Point: ROPEN

Chart: CA

Functions:

- Initializes constants and builds an input/output control block (RIOB) to read the specified sector of disk and write the corrections on the sector.
- Reads control statements and determines if they are in sequence. If they are not in sequence, a halt occurs.
- Determines whether a statement is a control statement or a data statement.

  1. Control statement – The data is converted from decimal to hexadecimal representation by the Hex Conversion routine within the Mainline phase. The parameters (RTAB) and a table of work area addresses (RADDS) are passed to the Keyword Syntax Scan routine. The Mainline phase then analyzes the table that is returned by the routine. A halt occurs if the parameters are invalid.

  2. Data statement – The statement is scanned for hexadecimal numbers. The Hex Conversion routine ensures that the correct hexadecimal data is written on the specified sector.

Input:

- // REBUILD or // PATCH control statement
- Data statement
- // END control statement

Output:

- Corrected sector of disk
- Printout of control statements

Exits:

- Normal – EOJ transient
- Error – Halt/Syslog transient

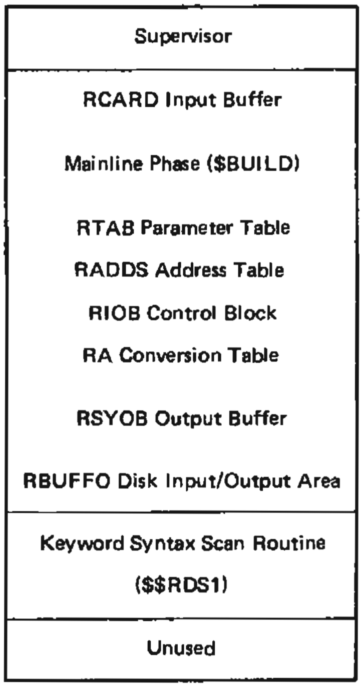
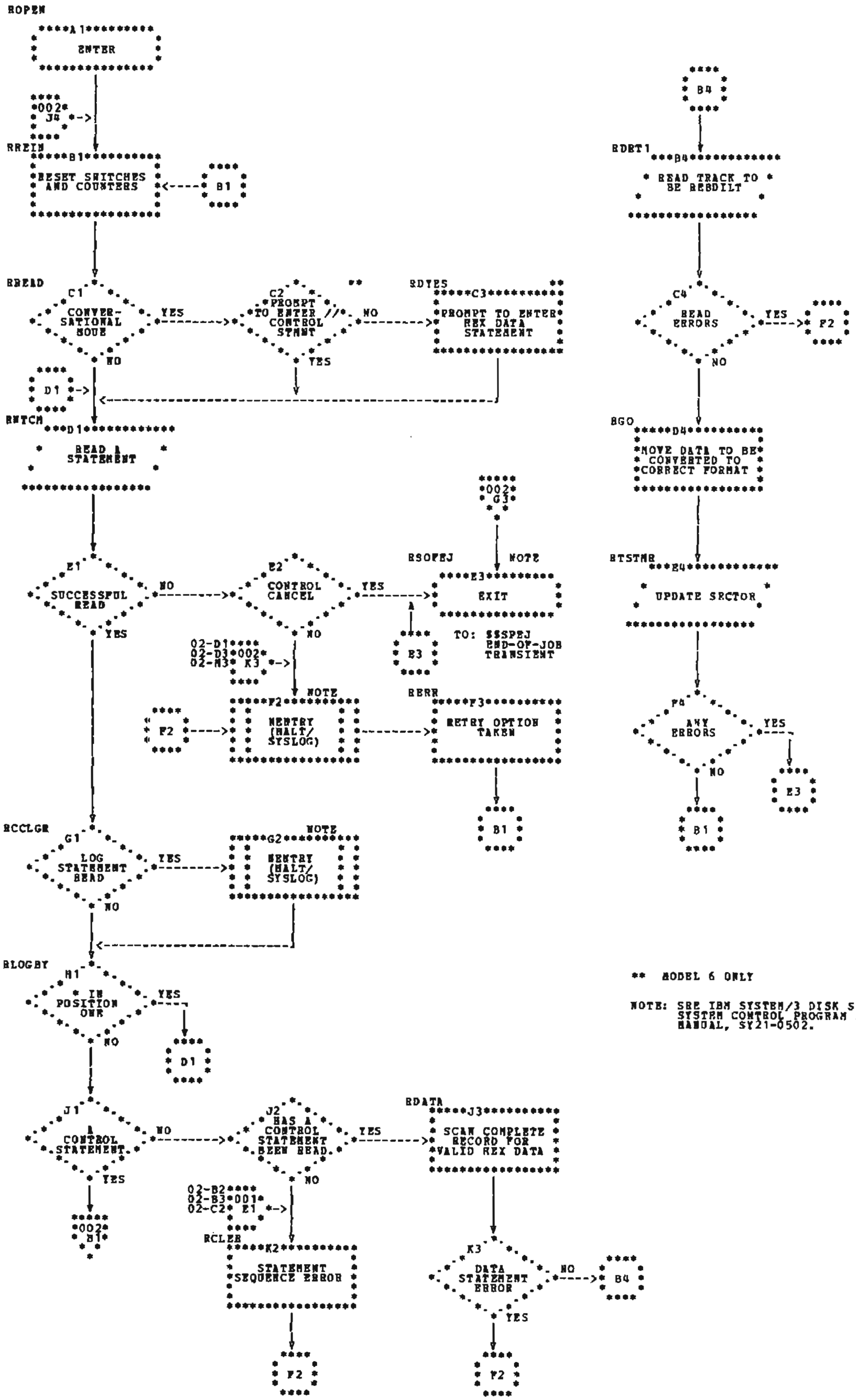


Figure 3-1. Storage Map for the Alternate Track Rebuild Program



** MODEL 6 ONLY

NOTE: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart CA (Part 1 of 2). Mainline Phase

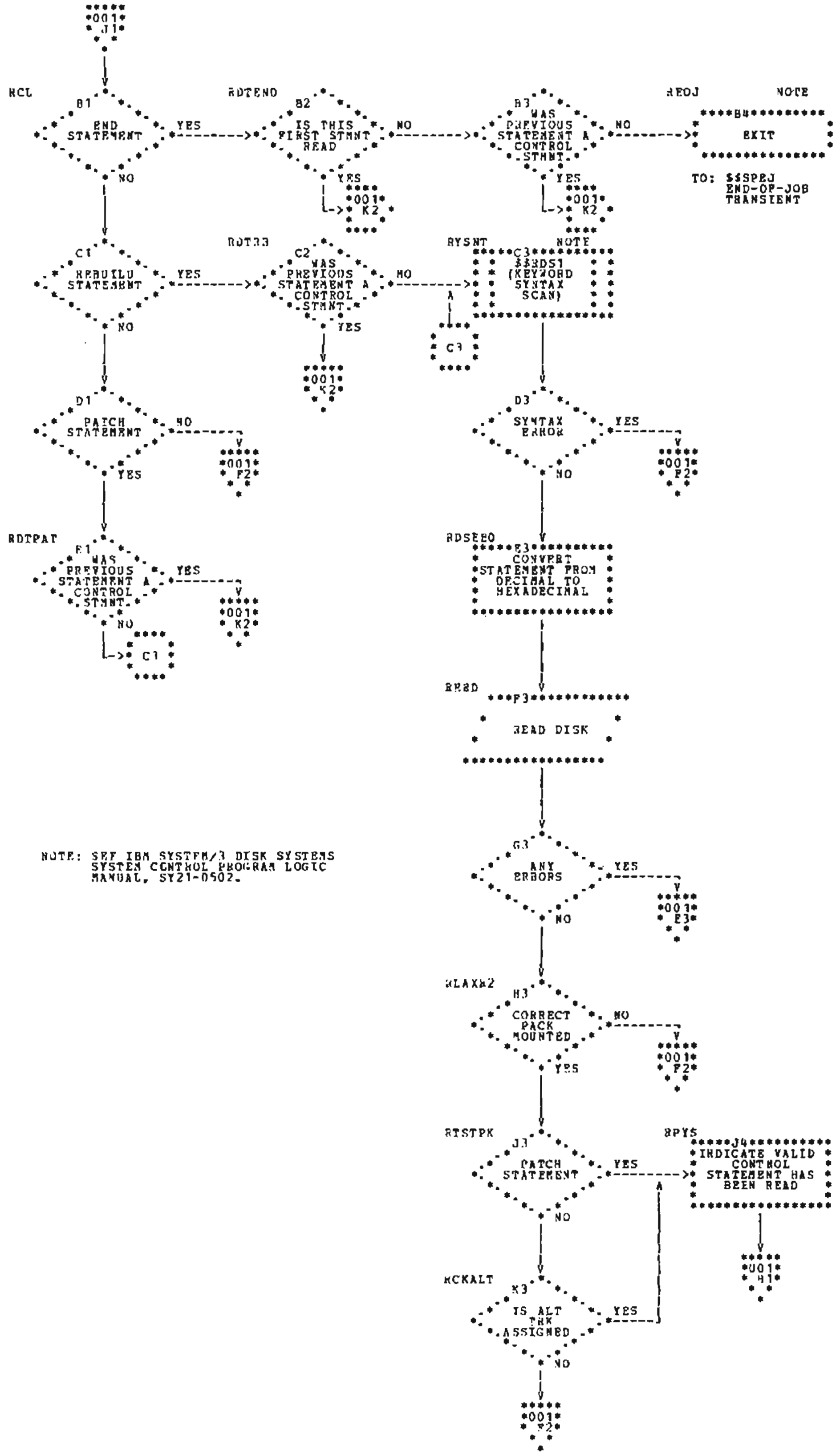


Chart CA (Part 2 of 2). Mainline Phase

Section 3. Data Area Formats

This section describes the format and contents of the control blocks, tables and data area used by the IBM System/3 Alternate Track Rebuild program. See the storage map in Figure 3-1 for the relative location of these data areas.

CONTROL BLOCKS

RIOB Control Block

RIOB is an input/output control block used to read the specified sector on the control statement and to write the corrected data on the sector. RIOB is located near the end of the Mainline phase. The byte positions and contents are:

Byte	*Contents*
0-1	Chain pointer used by IOS to queue requests
2	Completion code
3-4	Q and R bytes
5	Cylinder address in hexadecimal
6	Sector address in hexadecimal
7	Number of sectors involved in operation
8-9	Data area address
10-11	Sense bytes
12	Error byte
13	Flag byte X'80' = No error recovery X'40' = No verification X'20' = No error logging X'10' = Write ID operation X'08' = No DTF X'04' = No load I/O for disk file data register X'02' = Equipment check X'01' = Alternate track in process
14-17	IOS work area

TABLES

RA Conversion Table

RA is a 4-byte table pointed to by XR2 at entrance to the Convert Decimal to Hexadecimal routine. It contains a 2-byte address of the decimal representation to be converted and a 2-byte hexadecimal output address.

RTAB Parameter Table

RTAB, a 30-byte parameter table, contains the valid parameters which are passed from the Keyword Syntax Scan routine. RTAB is located near the end of the Mainline phase. The byte positions and contents are:

Byte	*Contents*
0	X'14'
1-4	'PACK'
5	X'24'
6-9	'UNIT'
10	X'35'
11-15	'TRACK'
16	X'46'
17-22	'LENGTH'
23	X'54'
24-27	'DISP'
28-29	X'FFFF'

RADDS Address Table

RADDS, an 8-byte table of addresses passed to the Keyword Syntax Scan routine, contains the addresses of the valid parameter table, output area, and the area to scan. It is located near the end of the Mainline phase. The byte positions and contents are:

Byte	*Contents*
0-1	Address of where to start scan
2-3	Address of where to end scan
4-5	Address of scan work area
6-7	Address of keyword parameter table

BUFFERS AND WORK AREAS

RCARD Input Buffer

RCARD is a 96-byte area used as the statement input buffer and conversion work area. It is located at the beginning of the Mainline phase.

RBUFFO Disk Input/Output Area

RBUFFO is a 256-byte disk input/output buffer. It is located at the end of the Mainline phase.

RSYOB Output Buffer

RSYOB is a 96-byte output buffer used by the Keyword Syntax Scan routine. It is located near the end of the Mainline phase.

PART IV
FILE AND VOLUME LABEL DISPLAY

The IBM System/3 File and Volume Label Display program prints the contents of:

- The entire volume table of contents (VTOC) which contains information on all data files on the disk.
- Certain volume label information such as the pack ID, owner ID, number of available alternate tracks, track numbers that have alternates assigned to them, device capacity, beginning and ending tracks of the libraries, and available space on the pack.
- Specified file labels in the VTOC containing information about the specified files.

SYSTEM REQUIREMENTS

The IBM System/3 File and Volume Display program uses the following minimum system configuration:

Disk System

- IBM 5410 Processing Unit, Model A12
- IBM 5203 Printer, Model A1
- IBM 5424 Multi-Function Card Unit, Model A
- IBM 5444 Disk Storage Drive, Model 2

Model 6

- IBM 5406 Processing Unit, Model B2
  (Includes the keyboard with eight command keys)
- IBM 5213 Printer, Model 1
- IBM 5444 Disk Storage Drive, Model 2

The following optional features are supported:

- IBM 5496 Data Recorder, Model 1 with System/3 Model 6 On-line Feature (Used as the Sysin device)
- IBM 2265 Display Station, Model 2 (Cathode Ray Tube; used as Syslog device with IBM 5406 Processing Unit, Models B3 and B4 only)

The general flow of the File and Volume Label Display program is shown in Figure 4-1.

The Root and Syntax Scan phase ($LABEL) reads the control statements into main storage (FSBUF). The control statements are read by SYSIN if they are on cards or by the job scheduler if they are in the procedure library. $LABEL loads the Keyword Syntax Scan routine ($$RDS1) and passes the parameter list FSSCAN to it. The control statements are checked for validity and the parameter list FSSCOT is passed back to $LABEL which checks for proper sequence.

$LABEL builds a table of constants (FRCTBK) which is used by the Mainline phase ($LABL1). $LABL1 is then loaded into storage. The volume label is read using the input/output area FIOA and the pack ID is stored in the table of constants. At this point, the flow of logic is dependent upon the LABEL operand specified on the DISPLAY statement.

- If the control statement indicates that the VTOC is to be displayed, $LABL1 prints the volume label information from FDBUF. It prints label headings, reads the VTOC index using the input/output area FIOA, and prints the file labels found via the index and the scratch chain. The scratch chain pointed to by the VTOC index, contains files defined with a scratch attribute. It then calls the EOJ transient.

- If the control statement indicates that a file label is to be displayed, $LABL1 prints the label headings, reads the VTOC index, and searches for the specified filenames. If the filenames are found, the file labels are read into the buffer FIOA2 and printed from FDBUF. The EOJ transient is then called. If the specified filenames are not found, a halt occurs.

Note: All control statements and halt messages are logged by the Halt/Syslog transient.

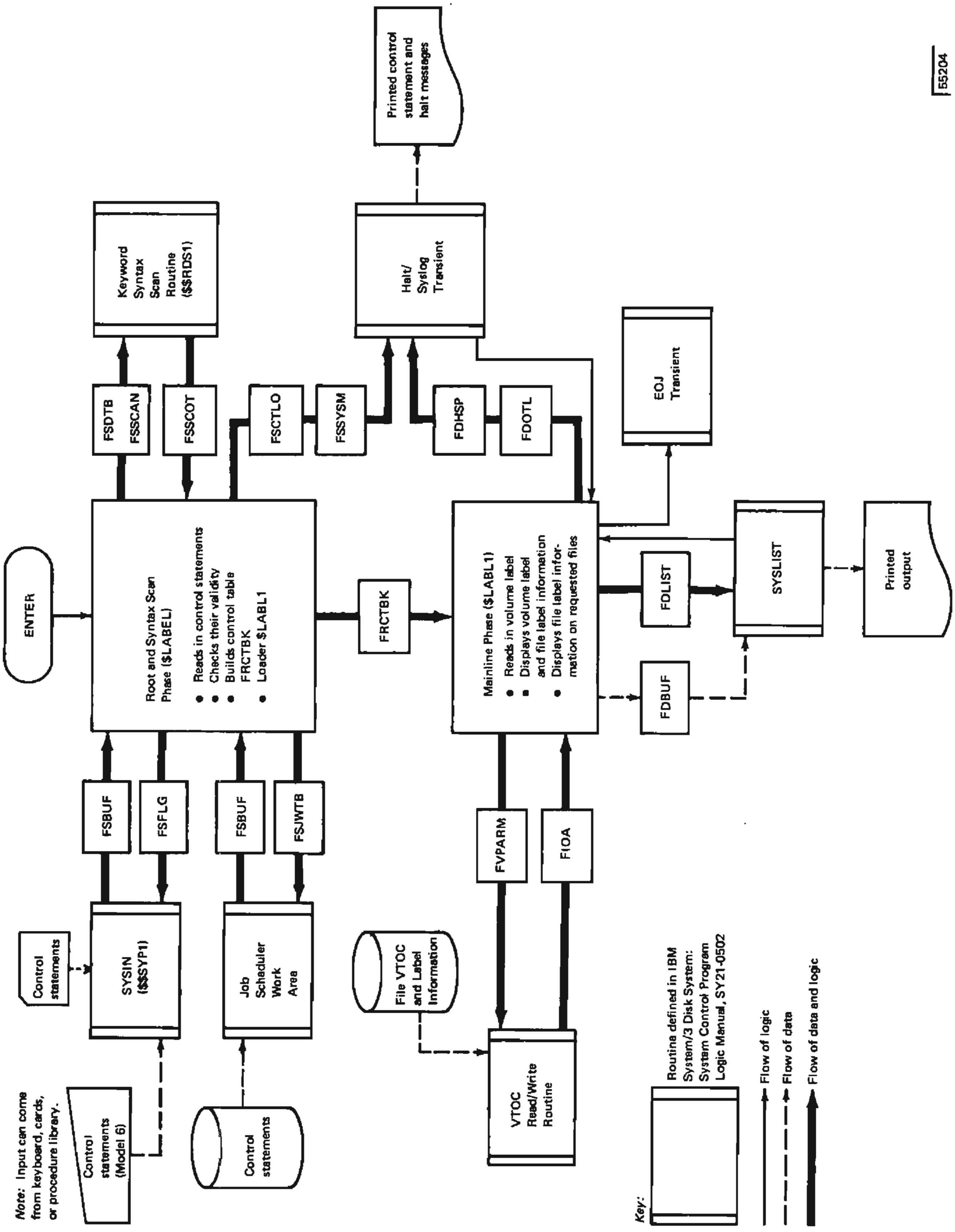

Note: Input can come from keyboard, cards, or procedure library.
Control statements (Model 6)
Control statements
Control statements
SYSIN ($$SYP1)
Job Scheduler Work Area
File VTOC and Label Information
VTOC Read/Write Routine
FSBUF
FSFLG
FSBUF
FSJWTB
FVPARM
FIOA
ENTER
Root and Syntax Scan Phase ($LABEL)
• Reads in control statements
• Checks their validity
• Builds control table FRCTBK
• Loader $LABL1
FRCTBK
Mainline Phase ($LABL1)
• Reads in volume label
• Displays volume label and file label information
• Displays file label information on requested files
FDBUF
FDLIST
SYSLIST
Printed output
FSDTB
FSSCAN
FSSCOT
Keyword Syntax Scan Routine ($$RDS1)
FSCTLO
FSSYSM
FDHSP
FDOTL
Halt/ Syslog Transient
Printed control statement and halt messages
EOJ Transient
55204
Key:
Routine defined in IBM System/3 Disk System: System Control Program Logic Manual, SY21-0502
Flow of logic
Flow of data
Flow of data and logic

Figure 4-1. Operational Diagram for the File and Volume Label Display Program

This section contains a functional description and flowchart of each phase included in the IBM System/3 File and Volume Label Display program. Figure 4-2 shows a storage map of the program.

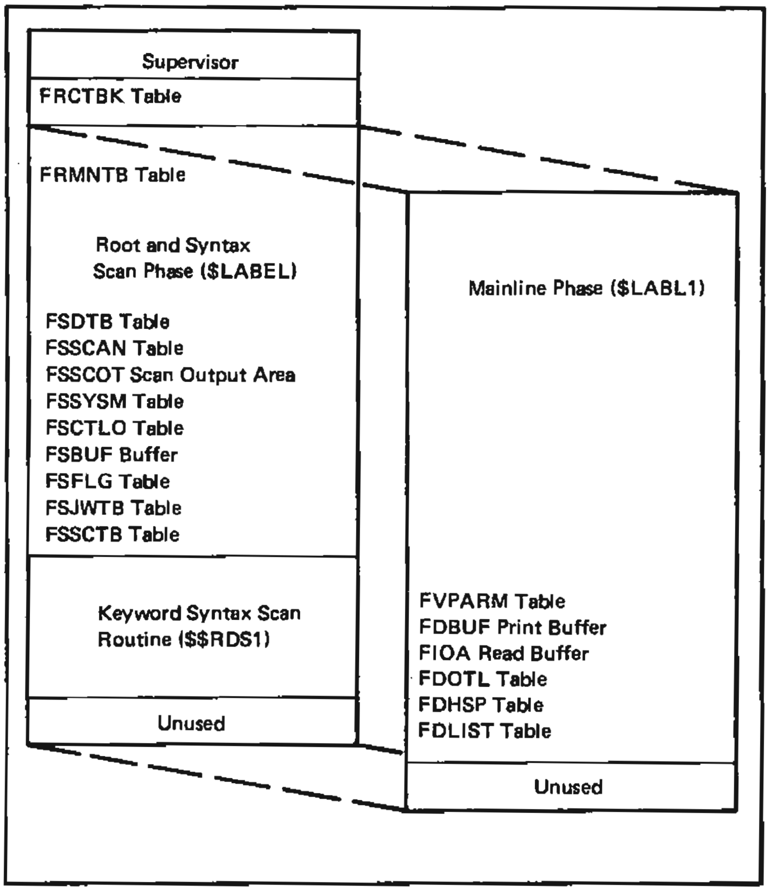


Figure 4-2. Storage Map of the File and Volume Label Display Program

Root and Syntax Scan Phase ($LABEL)

Entry Point: FILSTN

Chart: DA

Functions:

- Reads control statements. If an error occurs, the Halt/Syslog transient logs the control statement along with the proper error message if a logging device is on. A retry is allowed only if the control statements were read from SYSIN.
- Checks the validity and proper sequence of the control statements.
- Prints all control statements if a logging device is on.
- Builds and passes a table of constants (FRCTBK) to the Mainline phase ($LABL1).
- Loads and passes control to the Mainline phase ($LABL1).

Input: Control statements

Output:

- Table of constants (FRCTBK) to the Mainline phase
- Printed control statements

Exits:

- Normal – Mainline phase ($LABL1)
- Error – Halt/Syslog transient

Mainline Phase ($LABL1)

Entry Point: DISPLY

Chart: DB

Functions:

- Reads in volume label to obtain pack ID.
- Converts and displays file information for all files and, if the control statement was LABEL-VTOC, volume label information in readable format.
- Converts and displays file information for all requested files in readable format if the control statement was not LABEL-VTOC. If a file label is specified and more than one file contains the same name, all files with that name are printed.

Input: Table of constants (FRCTBK) from Root and Syntax Scan phase

Output: Printed file information

Exits:

- Normal – EOJ transient
- Error – Halt/Syslog transient

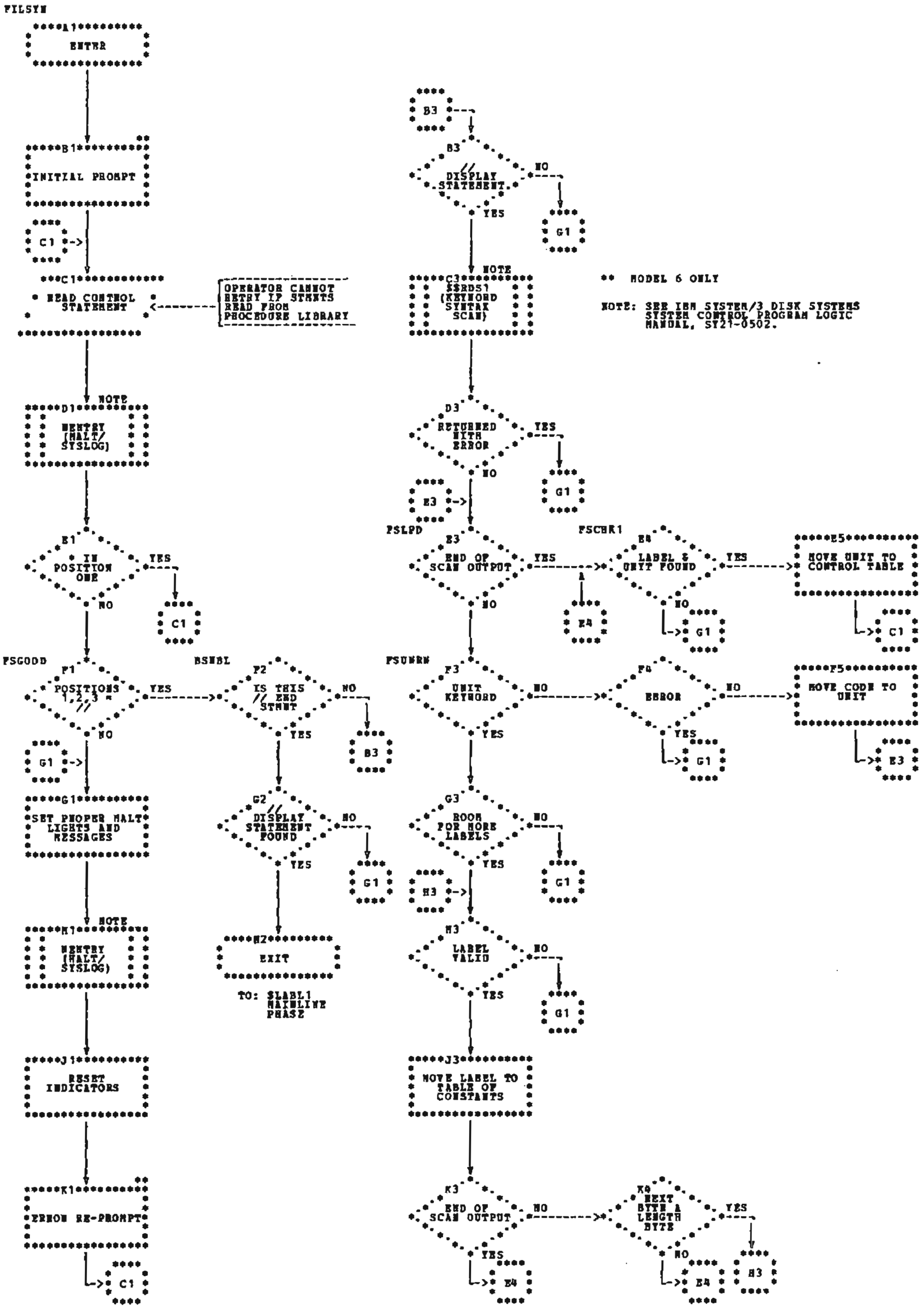


Chart DA. Root and Syntax Scan Phase

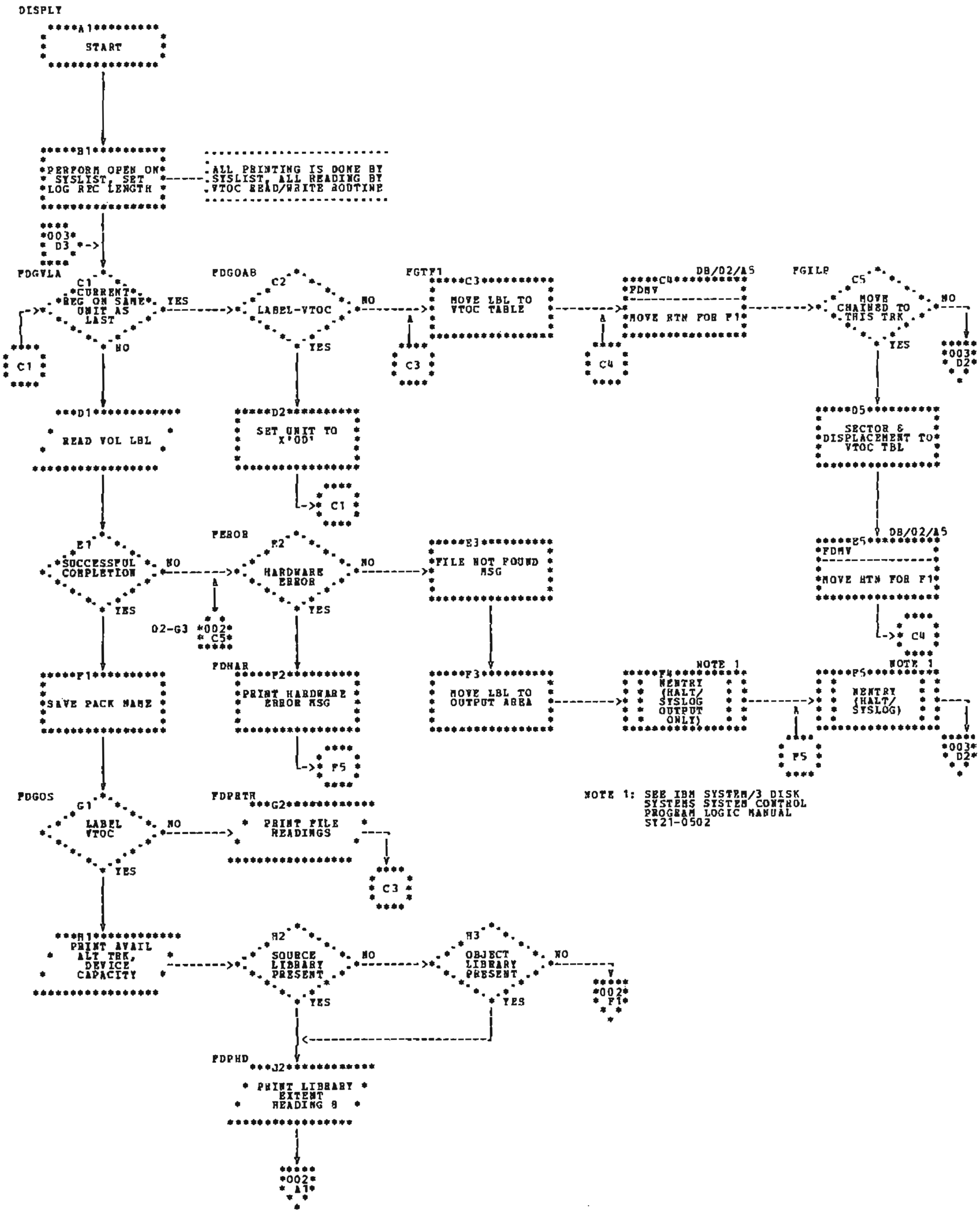


Chart DB (Part 1 of 3). Mainline Phase

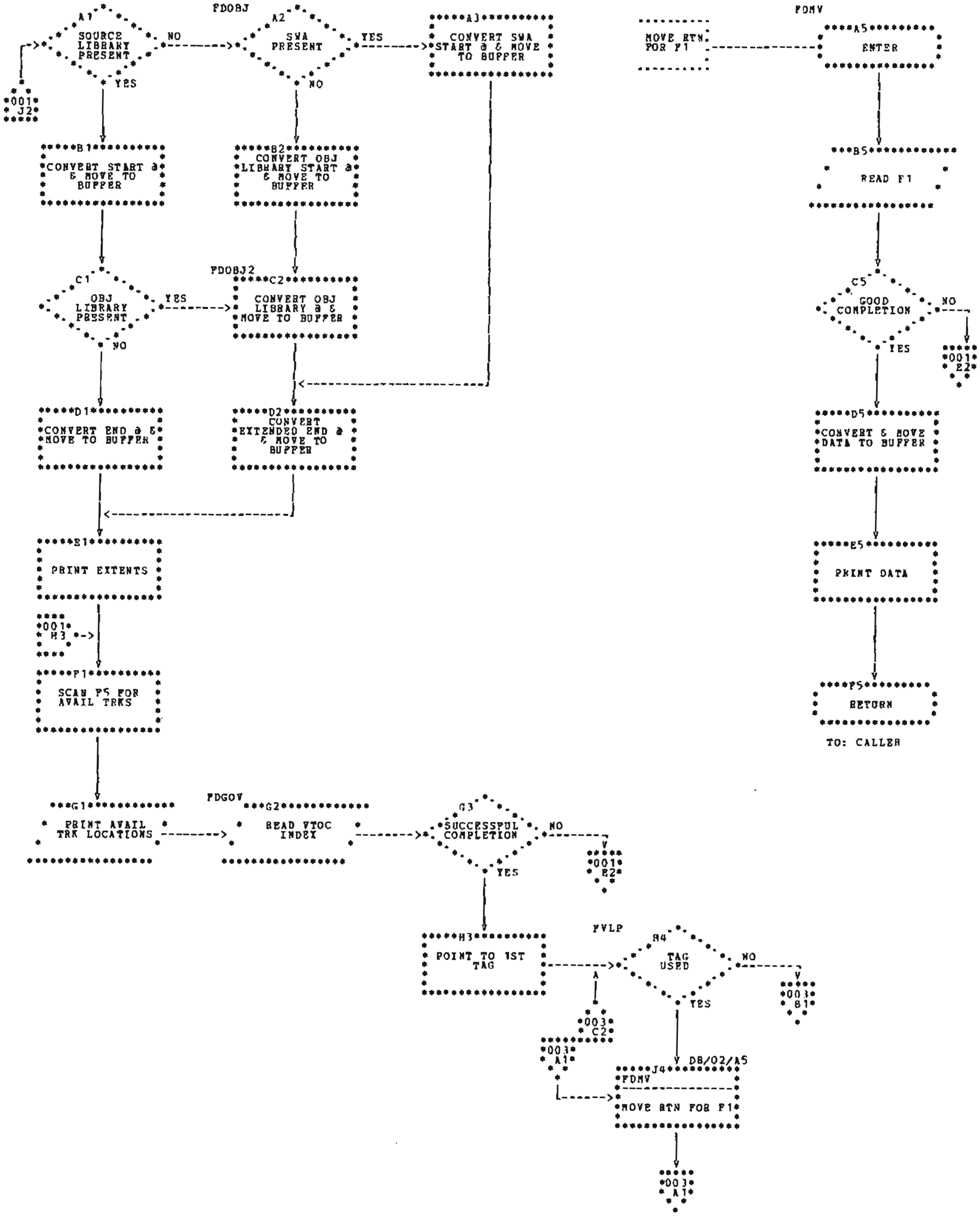


Chart DB (Part 2 of 3). Mainline Phase

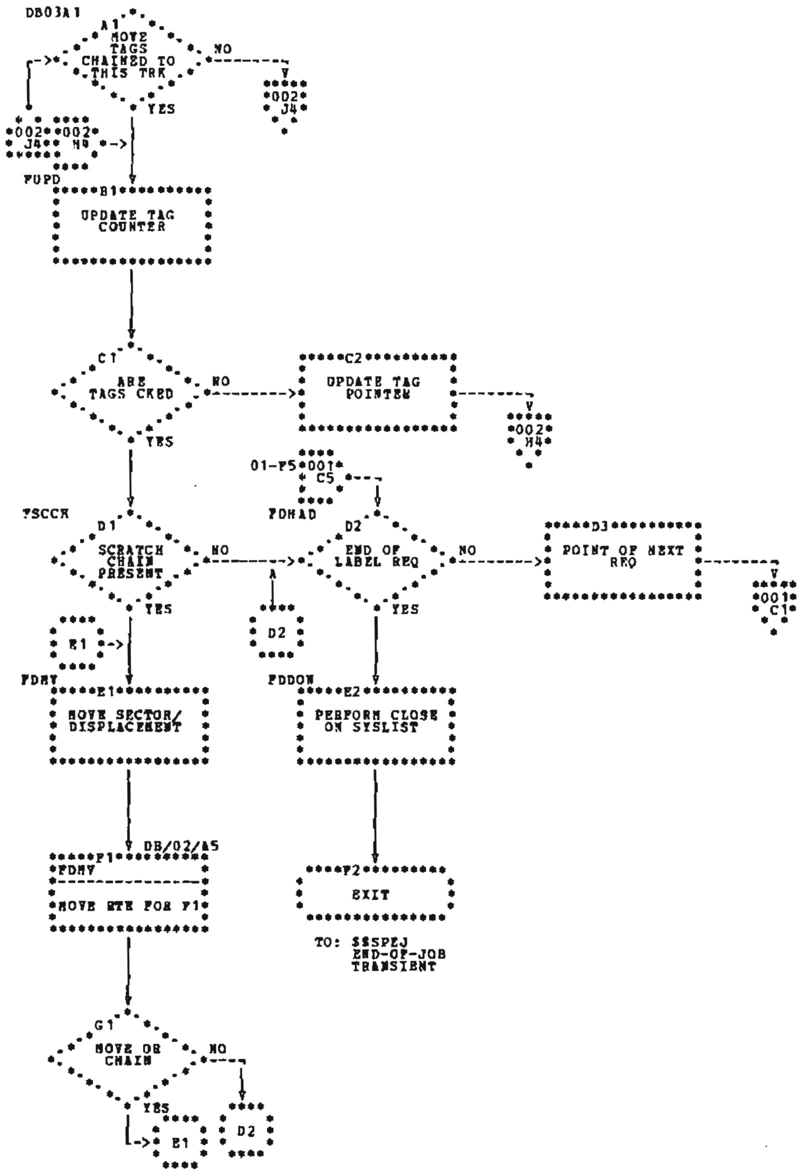


Chart DB (Part 3 of 3). Mainline Phase

Section 4. Data Area Formats

This section describes the format and contents of the control blocks, tables, and data areas used by the IBM System/3 File and Volume Label Display program. Figure 4-2 shows a storage map with the relative position of tables and buffers within each phase.

TABLES

FRCTBK Table

FRCTBK is a 180-byte table initialized to blanks. It is built by the Root and Syntax Scan phase for the Mainline phase and contains the file labels to be displayed. It can contain a maximum of 20 file labels. It is located in storage at the beginning of the Root and Syntax Scan phase. The Mainline phase is loaded at the end of the FRCTBK table.

FSDTB Table

FSDTB is a 13-byte keyword parameter table for the DISPLAY control statement in the Root and Syntax Scan phase. The byte positions and contents are:

Byte	*Contents*
0	X'14'
1-4	'UNIT'
5	X'25'
6-10	'LABEL'
11-12	X'FFFF'

FSSCAN Table

FSSCAN is an 8-byte table of addresses passed by the Root and Syntax Scan phase to the Keyword Syntax Scan routine (Figure 4-3).

FSFLG Table

FSFLG is a 7-byte parameter table passed by the Root and Syntax Scan phase to SYSIN if the control statements are not read from the procedure library. The byte positions and contents are:

Byte	*Contents*
0	Flag byte X'60' = Hardware error X'40' = Good completion X'09' = Single buffer
1-2	Address of input buffer
3-4	Address of input buffer
5-6	Address of work area

Byte	Label	Contents
0-1	FSSTAD	Address of where to start scan
2-3	None	Address of where to end scan (FSCEND)
4-5	None	Address of scan output area (FSSCOT)
6-7	FSAPTB	Address of keyword parameter table (FSDTB)

Figure 4-3. FSSCAN Table

FSJWTB Table

FSJWTB, a 7-byte parameter table, is passed by the Root and Syntax Scan phase to the job scheduler work area if the control statements are located on disk (Figure 4-4).

Byte	Label	Contents
0-1	FSFB	X'4860'
2	FSCMC	X'00'
3-4	None	Address of input buffer (FSBUF)
5-6	None	Not used

Figure 4-4. FSJWTB Table

FSCTLO Table

FSCTLO is a 5-byte table passed to the Halt/Syslog transient from the Root and Syntax Scan phase for logging the control statements (Figure 4-5).

Byte	Label	Contents
0-1	FSCTLO	X'0000' - Indicates output only; log control statement
2	None	X'60' - Indicates 96-byte length
3-4	None	Address of I/O area (FSBUF)

Figure 4-5. FSCTLO Table

FSSYSM Table

FSSYSM is a 7-byte table passed to the Halt/Syslog transient from the Root and Syntax Scan phase for halts and error messages (Figure 4-6).

Byte	Label	Contents
0	FSSYSM	B'11010110' – Indicates halt and system print
1	None	C'F' – Indicates File Delete program
2-3	FSHALT	Disk System – Characters to be used for halt display lights Model 6 – Code to be converted for halt display lights
4	None	Operator options X'01' = Immediate cancel X'02' = Control cancel X'04' = Retry X'08' = Proceed
5-6	FSSUB	C'ƀƀ' – Indicates submessage area

Figure 4-6. FSSYSM Table

FRMNTB Table

FRMNTB, a 10-byte table in the Root and Syntax Scan phase ($LABEL), contains the constant #LABL1 and the address of where to load the Mainline phase. The byte positions and contents are:

Byte	*Contents*
0	X'00'
1-6	C'$LABL1'
7	X'00'
8-9	Address of where to load the Mainline phase (FRMLD)

FSSCTB Table

FSSCTB, a 10-byte table in the Root and Syntax Scan phase ($LABEL), contains the constant 0$$RDS1 and the address where to load the Keyword Syntax Scan routine. The byte positions and contents are:

Byte	*Contents*
0-6	C'0$$RDS1'
7	X'00'
8-9	Address of where to load the Keyword Syntax Scan routine (FSENT)

FVPARM Table

FVPARM, a 17-byte parameter table, contains control information for the VTOC Read/Write routine. It is passed from the Mainline phase to the VTOC Read/Write routine in the supervisor. The byte positions and contents are:

Byte	*Contents*
0	X'02'
1	Unit
2	Return code
3-4	Address of I/O area
5-6	Address of work area
7-8	Sector and displacement for format 1
9-16	Label for format 1

FDOTL Table

FDOTL is a 5-byte table passed to the Halt/Syslog transient from the Mainline phase. Bytes 0-2 contain X'000008' and bytes 3-4 have the address of the output buffer (FDBUF).

FDHSP Table

FDHSP is a 7-byte table passed to the Halt/Syslog transient from the Mainline phase for halts and error messages (Figure 4-7).

Byte	Label	Contents
0	FDHSP	B'11010110' – Indicates halt and system print
1	None	C'F' – Indicates File Delete program
2-3	FDHLT	Disk System – Characters to be used for halt display lights Model 6 – Code to be converted for halt display lights
4	None	Operator options X'01' = Immediate cancel X'02' = Control cancel X'04' = Retry X'08' = Proceed
5-6	FDSUB	C'NF' – Indicates submessage area

Figure 4-7. FDHSP Table

FDLIST Table

FDLIST, a 7-byte table of parameters passed from the Mainline phase to SYSLIST, contains control information for printing (Figure 4-8).

Byte	Label	Contents
0	FDOP	Print and skip space
1	FDSKBP	Skip before printing
2-3	None	Space before and skip after printing
4	FDSPAP	Space 1 after printing
5-6	None	Address of FDBUF print buffer

Figure 4-8. FDLIST Table

BUFFERS

FSBUF Input Buffer

FSBUF, a 96-byte input buffer, contains the control statement images in the Root and Syntax Scan phase.

FDBUF Print Buffer

FDBUF, a 96-byte print buffer, contains images from the Mainline phase to be printed by SYSLIST.

FIOA Read Buffer

FIOA is a 512-byte area used by the Mainline phase for two functions:

1. As a 256-byte read buffer for the volume label.
2. As a 512-byte read buffer for the VTOC index.

FIOA2 Read Buffer

FIOA2 is a 64-byte read buffer (the last 64 bytes of FIOA) used by the Mainline phase to read the file labels.

FSSCOT Scan Output Area

FSSCOT is a 128-byte area containing the operands of the valid keywords. It is passed from the Keyword Syntax Scan routine ($$RDS1) to the Root and Syntax Scan phase.

PART V
FILE DELETE

The IBM System/3 File Delete program allows the user to delete files no longer used or needed which are listed in the volume table of contents (VTOC). The File Delete program deletes:

- Specified file(s) listed in the VTOC.
- All files listed in the VTOC.

If a SCRATCH statement is specified, the program writes the file labels on the scratch chain. The scratch chain, pointed to by the VTOC index, contains files defined as scratch files (files which may be overlaid by other files). If the REMOVE statement is specified, the file labels are removed from the VTOC.

SYSTEM REQUIREMENTS

The IBM System/3 File Delete program uses the following minimum system configuration:

Disk System

- IBM 5410 Processing Unit, Model A12
- IBM 5203 Printer, Model A1
- IBM 5424 Multi-Function Card Unit, Model A
- IBM 5444 Disk Storage Drive, Model 2

Model 6

- IBM 5406 Processing Unit, Model B2 (Includes the keyboard with eight command keys)
- IBM 5213 Printer, Model 1
- IBM 5444 Disk Storage Drive, Model 2

The following optional features are supported:

- IBM 5496 Data Recorder, Model 1 with System/3 Model 6 On-line Feature (Used as the Sysin device)
- IBM 2265 Display Station, Model 2 (Cathode Ray Tube) (Used with the IBM 5406 Processing Unit, Models B3 and B4 only) (Used as the Syslog device)

Figure 5-1 shows an operational diagram of the File Delete program.

When the user loads the File Delete program, the Root phase receives control. It loads the Control Statement Analysis phase which reads in the control statements using the input buffer FSBUF. The control statements are checked for validity by calling the Keyword Syntax Scan routine ($$RDS1). A table of keyword parameters (FSCTBK) is built by the Control Statement Analysis phase from the output (FSSCOT) of $$RDS1. FSCTBK is passed to the Root phase and maintained under the label FRCTBK. The Root phase again receives control and passes FRCTBK to the Mainline phase. The Mainline phase calls the VTOC Read/Write routine to read the volume label, VTOC index, and file labels. It gives an S retain type to all P or T retain types if the request was LABEL-VTOC. It gives an S retain type to specified files if the request is only for specified file labels. The flow of logic then depends upon which control statement was read by the Control Statement Analysis phase.

If a SCRATCH statement were read, the Job Scheduler Work Area Put routine writes the file labels with their new retain type in the scheduler work area. When the Control Statement Analysis phase reads a valid END statement, it calls the EOJ transient which writes the file labels onto the scratch chain.

If a REMOVE statement containing the parameter DATA-YES was read, data for the files is removed. The Job Scheduler Work Area Put routine then writes the file labels and attributes set to remove the labels in the scheduler work area. When the Control Statement Analysis phase reads a valid END statement, it calls the EOJ transient which removes the file labels from the VTOC.

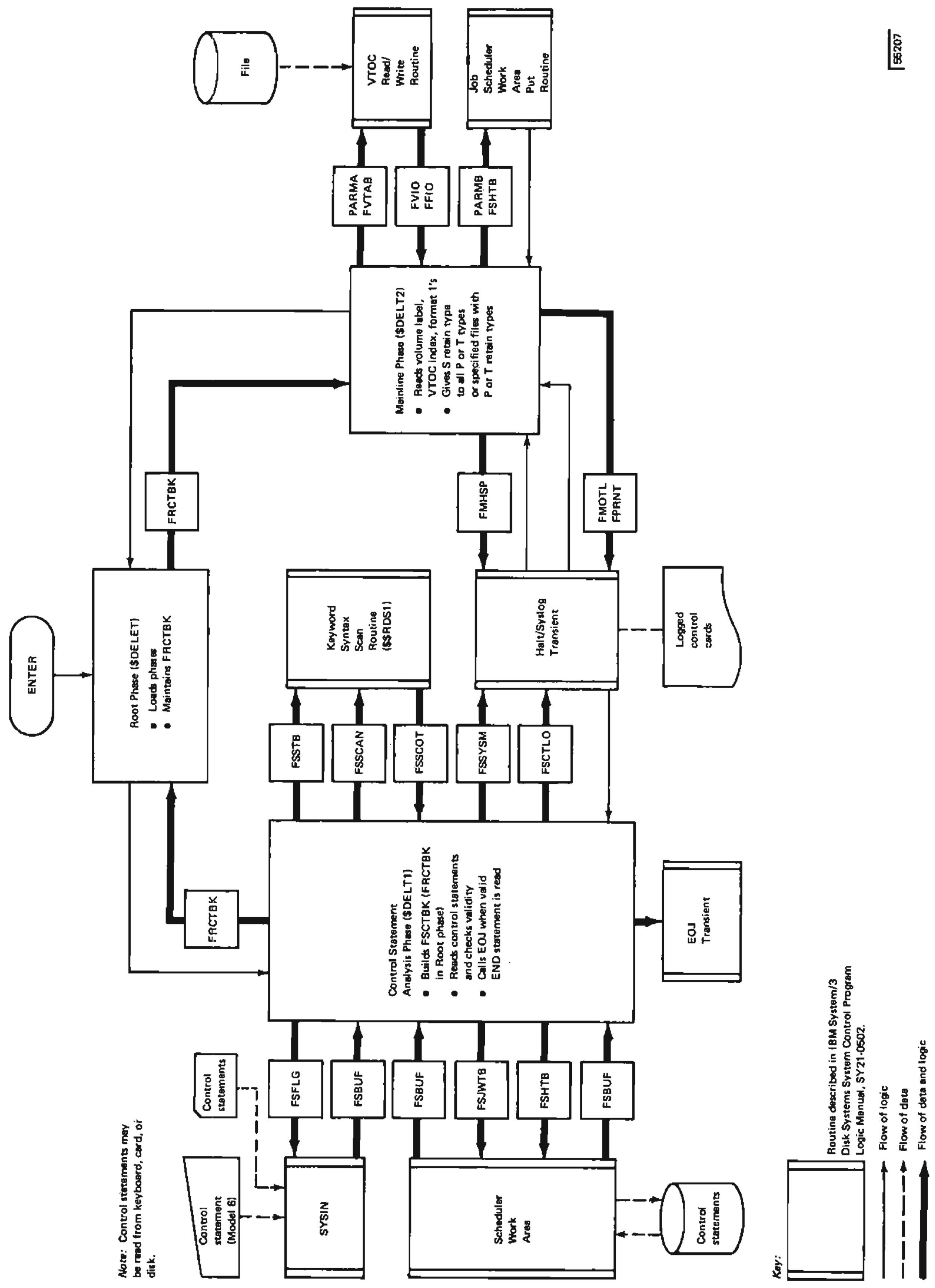

ENTER
Root Phase ($DELET)
• Loads phases
• Maintains FRCTBK
FRCTBK
FRCTBK
Mainline Phase ($DELT2)
• Reads volume label, VTOC index, format 1's
• Gives S retain type to all P or T types or specified files with P or T retain types
File
VTOC Read/ Write Routine
Job Scheduler Work Area Put Routine
PARMA FVTAB
FVIO FFIO
PARMB FSHTB
FMHSP
FMOTL FPRNT
Keyword Syntax Scan Routine ($$RDS1)
Halt/Syslog Transient
Logged control cards
FSSTB
FSSCAN
FSSCOT
FSSYSM
FSCTLO
Control Statement Analysis Phase ($DELT1)
• Builds FSCTBK (FRCTBK in Root phase)
• Reads control statements and checks validity
• Calls EOJ when valid END statement is read
EOJ Transient
FSFLG
FSBUF
FSBUF
FSJWTB
FSHTB
FSBUF
Note: Control statements may be read from keyboard, card, or disk.
Control statement (Model 6)
Control statements
SYSIN
Scheduler Work Area
Control statements
Key:
Routine described in IBM System/3 Disk Systems System Control Program Logic Manual, SY21-0502.
Flow of logic
Flow of data
Flow of data and logic
55207

Figure 5-1. Operational Diagram for the File Delete Program

Section 3. Program Organization

The following section contains a function description of the phases comprising the IBM System/3 File Delete program. Flowcharts are provided where necessary. Figure 5-2 shows a storage map of the program.

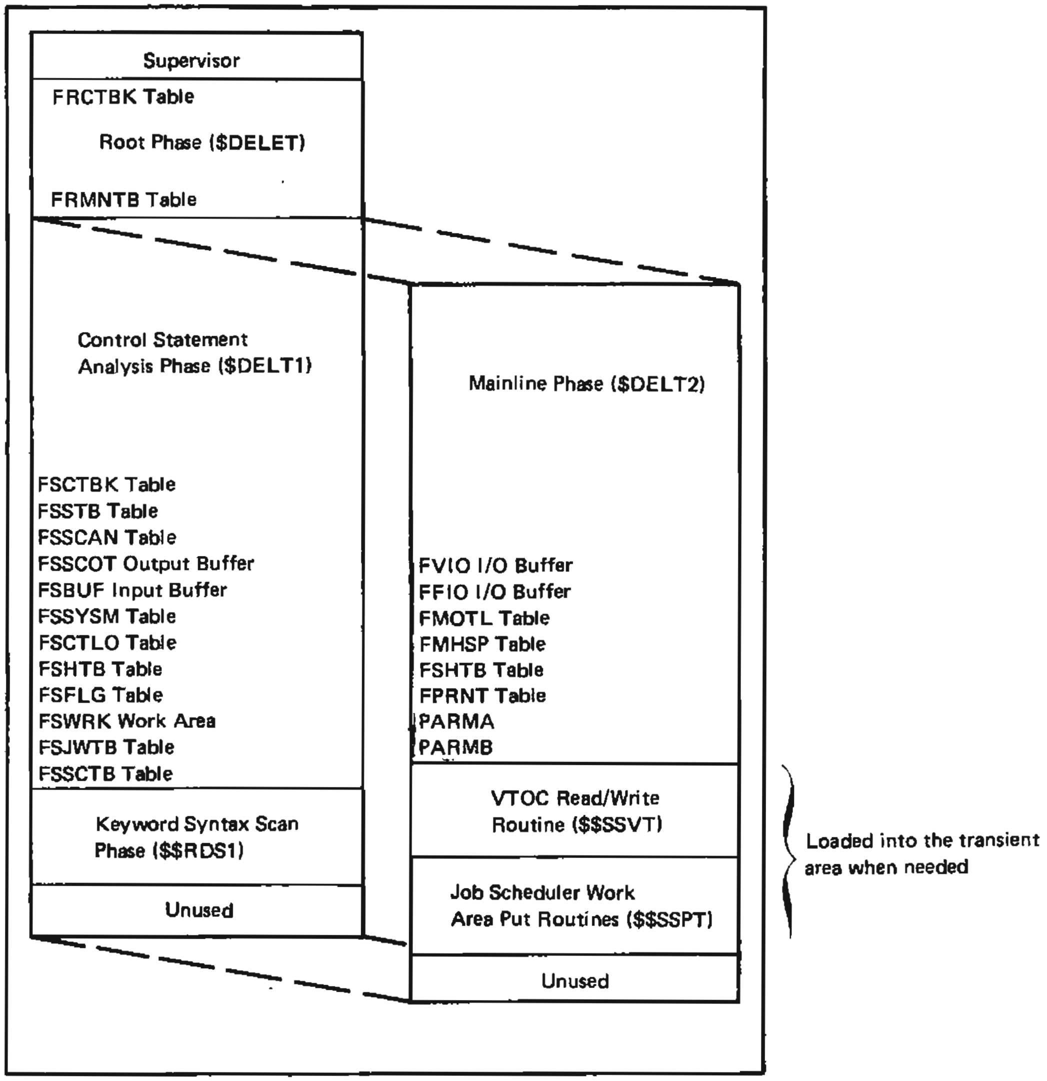


Figure 5-2. Storage Map of the File Delete Program

Root Phase ($DELET)

Entry Point: FLBRS

Chart: None

Functions:

- Provides a save area for FRCTBK. FRCTBK is a table of keyword parameters passed between phases.
- Loads and passes control to the Control Statement Analysis phase ($DELET) and to the Mainline phase ($DELT2).

Input: Table of keyword parameters (FSCTBK) from Control Statement Analysis phase ($DELT1).

Output: Table of keyword parameters (FRCTBK) to the Mainline phase

Exits:

- Normal
  1. Supervisor to load phases
  2. Entry point of phases called
- Error – None

Control Statement Analysis Phase ($DELT1)

Entry Point: FILSYN

Chart: EA

Functions:

- Reads in control statements.
- Checks the validity of the control statements.
- Logs the control statements if a logging device is on.
- Builds a table of keyword parameters (FSCTBK) and passes it to the Root phase.

Input: Control statements

Output: Table of keyword parameters (FSCTBK) to the Root phase

Exits:

- Normal
  1. Root phase after table of keyword parameters (FSCTBK) is built
  2. EOJ transient
- Error – Halt/Syslog transient

Mainline Phase ($DELT2)

Entry Point: FDEL

Chart: EB

Functions:

- If the control statement reads LABEL-VTOC:

  1. All the format 1s (F1s) on the VTOC index with a P or T retain type are read in.

  2. Data is removed for all specified files, if a REMOVE control statement with DATA-YES is read.

  3. The F1s with a P or T retain type are given an S retain type.

  4. The F1s are written out in the scheduler work area.

- If the control statement does not read LABEL-VTOC:

  1. The specified F1s on the VTOC index are read in.

  2. Data is removed for all specified files, if a REMOVE control statement with DATA-YES is read.

  3. The F1s are given an S retain type.

  4. The F1s are written out in the scheduler work area.

  Note: If a pack has more than one file with the same label, a DATE option can be specified to delete one of the files. If no DATE option is specified, all the files with that label are deleted.

Input:

- Table of keyword parameters from the Root phase

- Volume label, VTOC index, and F1s from disk

Output: F1s in the scheduler work area

Exits:

- Normal – Root phase

- Error – Halt/Syslog transient

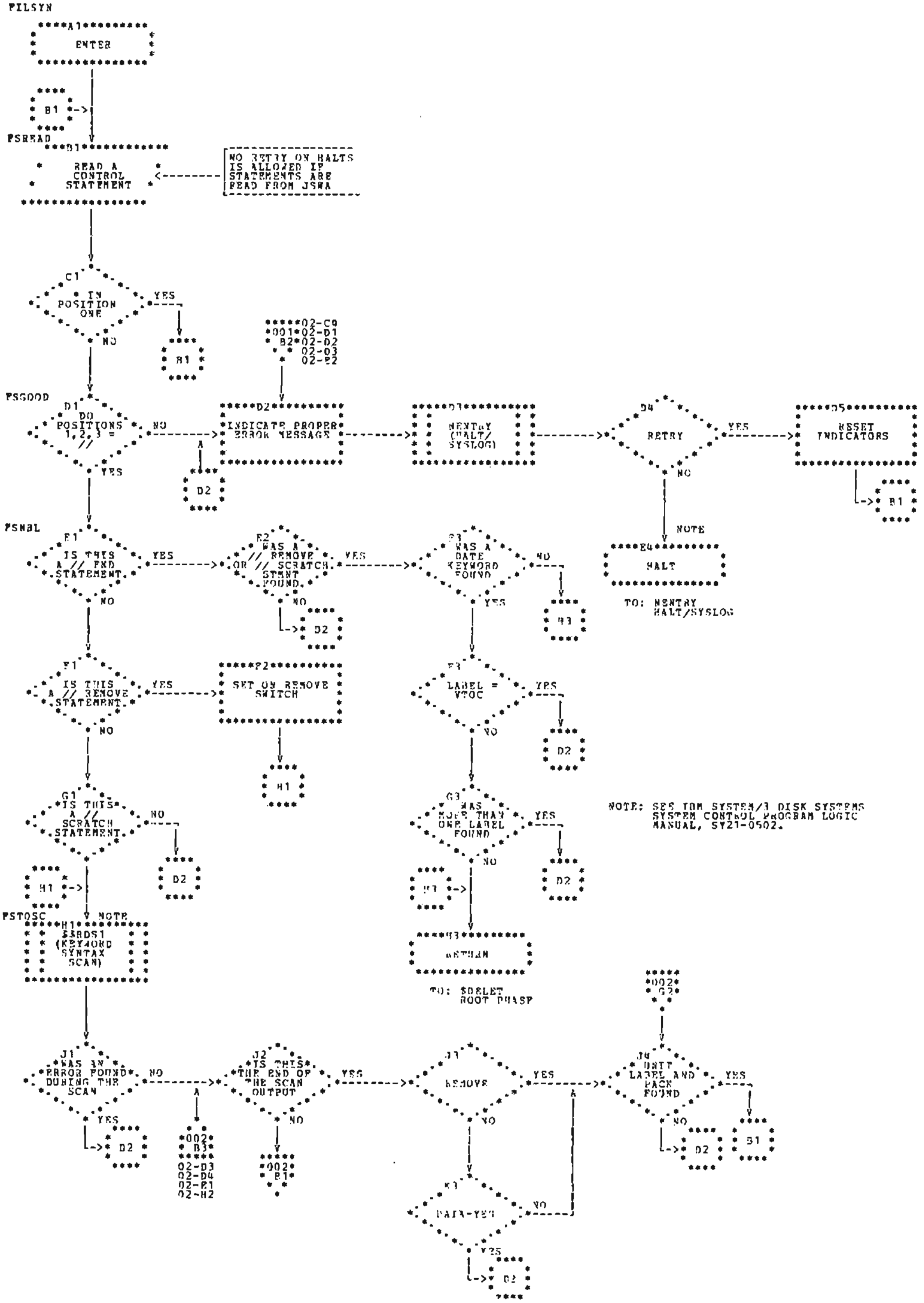


Chart EA (Part 1 of 2). Control Statement Analysis Phase

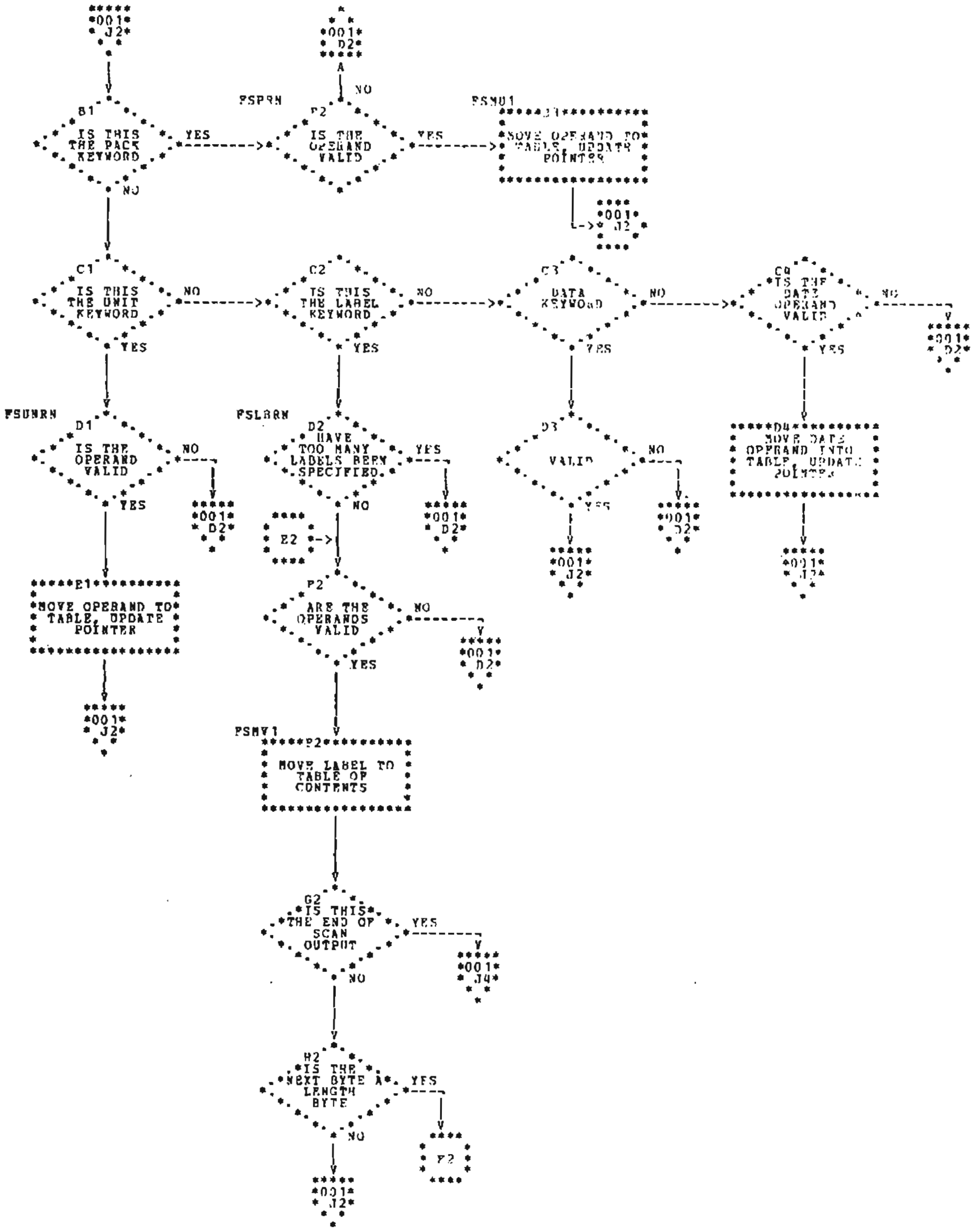
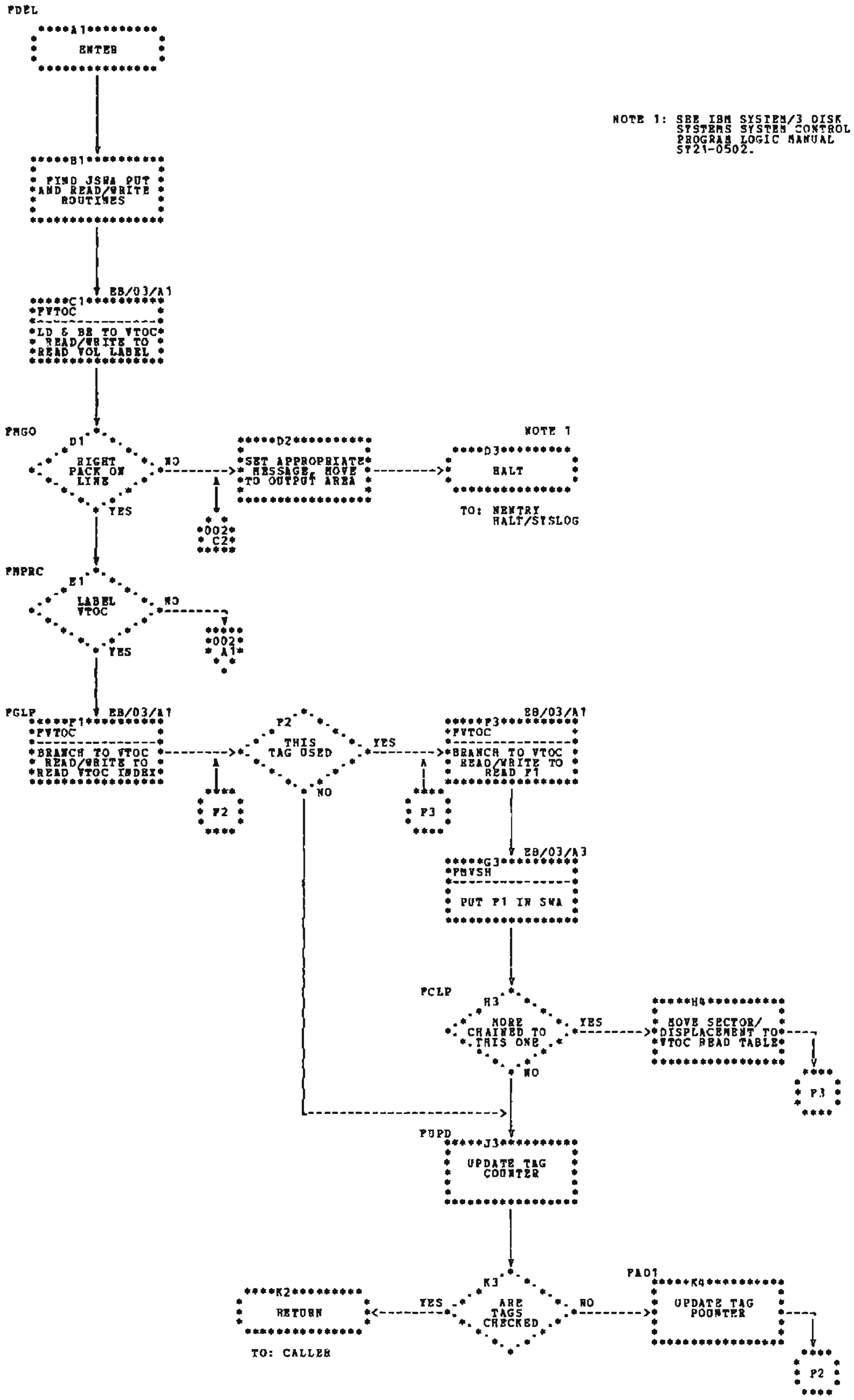
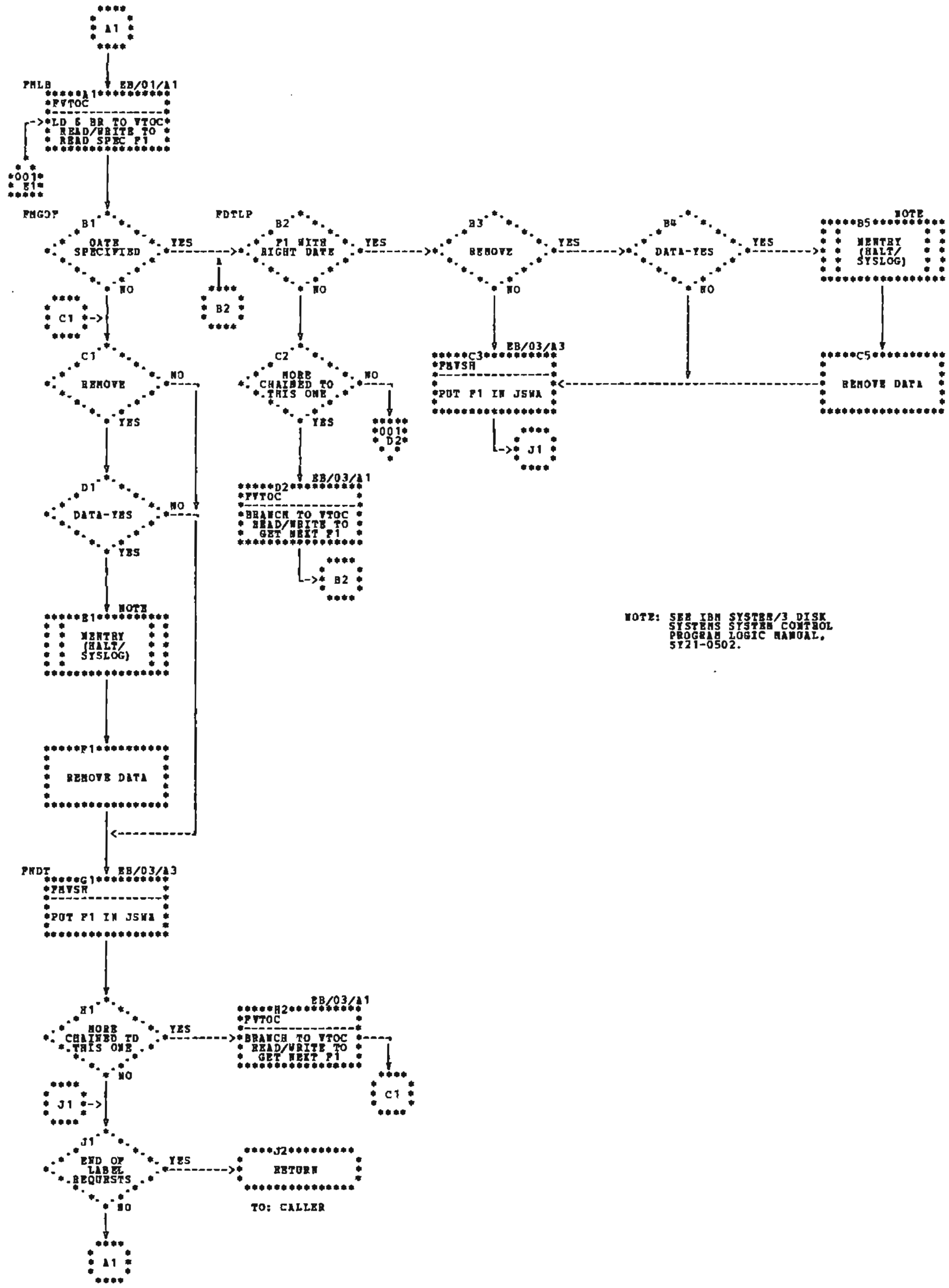


Chart EA (Part 2 of 2). Control Statement Analysis Phase



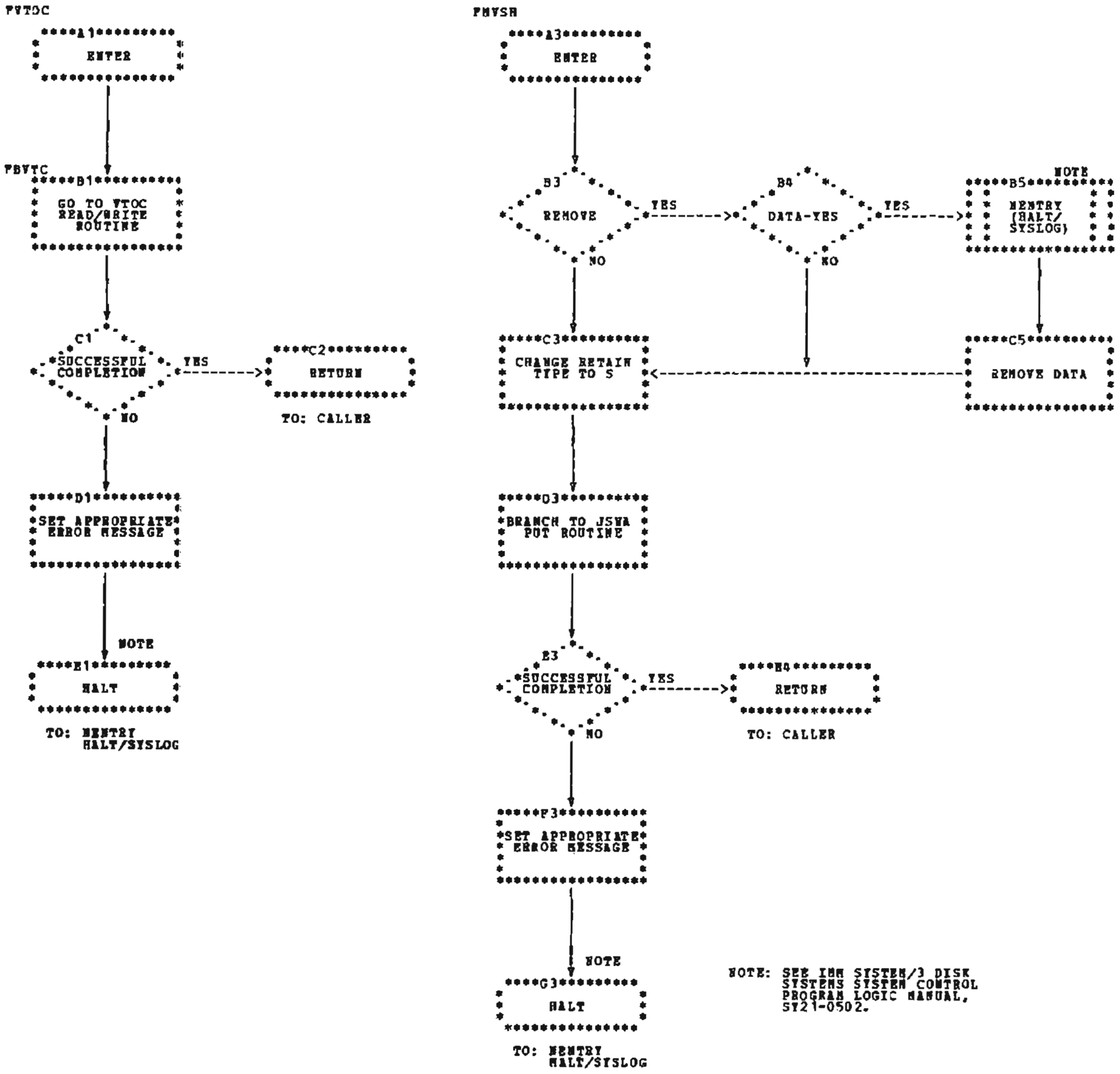
NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502.

Chart EB (Part 1 of 3). Mainline Phase



NOTE: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart EB (Part 2 of 3). Mainline Phase



NOTE: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart EB (Part 3 of 3). Mainline Phase

Section 4. Data Area Formats

This section describes the format and contents of the control blocks, tables, and data areas used by the IBM System/3 File and Delete program. Figure 5-2 shows a storage map with the relative position of tables and buffers within each phase.

TABLES

FRCTBK (FSCTBK) Table

FSCTBK is a 264-byte keyword parameter table filled by the Control Statement Analysis phase. It is moved to the Root phase where it is maintained as FRCTBK and used by the Mainline phase (Figure 5-3).

FSSTB Table

FSSTB, a 27-byte table, contains the parameters of the SCRATCH statement. It is passed by the Control Statement Analysis phase to the Keyword Syntax Scan routine. The byte positions and contents are:

Byte	*Contents*
0	X'14'
1-4	'PACK'
5	X'24'
6-9	'UNIT'
10	X'35'
11-15	'LABLE'
16	X'44'
17-20	'DATE'
21	X'54'
22-25	'DATA'
26	X'FF'

FSSCAN Table

FSSCAN is a 8-byte table of addresses passed by the Control Statement Analysis phase to the Keyword Syntax Scan routine (Figure 5-4).

Byte	Label	Contents
0	FRUNIT	Unit
1	FRVTOC	VTOC: 1. Y = Yes 2. N = No
2-7	FRPAK	Pack ID
8-13	FRDATE	Date
14-261	FRLABL	File labels
262	X'FF'	End of labels
263	FLBCT	Function byte for Job Scheduler Work Area Put routine

Figure 5-3. FRCTBK (FSCTBK) Table

Byte	Label	Contents
0-1	FSSTAD	Address of where to start scan
2-3	None	Address of where to end scan
4-5	None	Address of scan output area
6-7	FSAPIB	Address of keyword parameter table

Figure 5-4. FSSCAN Table

FSFLG Table

FSFLG is a 7-byte parameter table passed by the Control Statement Analysis phase to SYSIN if the control statements are not located on disk. The byte positions and contents are:

Byte	*Contents*
0	Flag byte X'60' = Hardware error X'40' = Good completion X'09' = Single buffer
1-2	Address of input buffer
3-4	Address of input buffer
5-6	Address of work area

FSJWTB Table

FSJWTB is a 7-byte parameter table passed by the Control Statement Analysis phase to the Job Scheduler Work Area Read routine to read control statements from disk (Figure 5-5).

FVTAB Table

FVTAB, a 17-byte parameter table, contains control information for the VTOC Read/Write routine. It is passed from the Mainline phase to the VTOC Read/Write routine in the supervisor. The byte positions and contents are:

Byte	*Contents*
0	X'02'
1	Unit
2	Return code
3-4	Address of I/O area
5-6	Address of work area
7-8	Sector and displacement for format 1
9-16	Label for format 1

FMOTL Table

FMOTL is a 5-byte table passed to the Halt/Syslog transient from the Mainline phase. Bytes 0-2 contain X'000009' and bytes 3-4 have the address of the output buffer (FDBUF).

FPRNT Table

FPRNT, a 5-byte table passed to the Halt/Syslog transient from the Mainline phase. The byte positions and contents are:

Byte	*Contents*
0-1	X'0000' = Control bytes
2	X'2A' = Message length
3-4	Address of the message

PARMA Table

PARMA, a 12-byte table in the Mainline phase ($DELT2), contains the constant O$$SSVT. The constant is used to find O$$SSVT. The byte positions and contents are:

Byte	*Contents*
0	C'O' = Library
1-6	C'O$$SSVT' = Name of module to be loaded
7	X'00' = Control byte
8-11	X'40404040'

Byte	Label	Contents
0-1	FSFB	X'4860'
2	FSCMC	X'00'
3-4	FSJWTB+4	Address of input buffer
5-6	None	Not used

Figure 5-5. FSJWTB Table

PARMB Table

PARMB, a 12-byte table in the Mainline phase ($DELT2), contains the constant O$$SSPT. The constant is used to find O$$SSPT. The byte positions and contents are:

Byte	*Contents*
0	C'O' = Library
1-6	C'O$$SSPT' = Name of module to be loaded
7	X'00' = Control byte
8-11	X'40404040'

FMHSP Table

FMHSP is a 7-byte table passed to the Halt/Syslog transient from the Mainline phase for halts and error messages (Figure 5-6).

Byte	Label	Contents
0	FMHSP	B'11010110' – Indicates halt and system print
1	None	C'F' – Indicates File Delete program
2-3	FMHLT	Disk System – Characters to be used for halt display lights Model 6 – Code to be converted for halt display lights
4	None	Operator options X'01' = Immediate cancel X'02' = Control cancel X'04' = Retry X'08' = Proceed
5-6	FMSUB	C'ƀƀ' – Indicates submessage area

Figure 5-6. FMHSP Table

FSHTB Table

FSHTB is a 5-byte parameter table passed to the Job Scheduler Work Area Put routine (Figure 5-7). The JSWA Put routine writes the file labels in the scheduler work area.

Byte	Label	Contents
0	FBT1	X'08' – Indicates function byte
1	FBT2	X'40' – Indicates length
2	FCOM	Completion code
3-4	FSHTB+4	Address of I/O area

Figure 5-7. FSHTB Table

FRMNTB Table

FRMNTB, a 10-byte table in the Root phase ($DELET), contains the constant $DELT2 and the address where to load the Mainline phase. The byte positions and contents are:

Byte	*Contents*
0	X'00'
1-6	C'$DELT2'
7	X'00'
8-9	Address of where to load the Mainline phase (FREND)

FSSCTB Table

FSSCTB, a 10-byte table in the Control Statement Analysis phase, contains the constant 0$$RDS1 and the address of where to load the Keyword Syntax Scan routine. The byte positions and contents are:

Byte	*Contents*
0-6	C'0$$RDS1'
7	X'00'
8-9	Address of where to load the Keyword Syntax Scan routine (FSENT)

FSCTLO Table

FSCTLO is a 5-byte table passed to the Halt/Syslog transient from the Control Statement Analysis phase for logging the control statements (Figure 5-8).

Byte	Label	Contents
0-1	FSCTLO	X'0000' Indicates output only, log control statement
2	None	X'60' - Indicates 96-byte length
3-4	None	Address of I/O area (FSBUF)

Figure 5-8. FSCTLO Table

FSSYSM Table

FSSYSM is a 7-byte table passed to the Halt/Syslog transient from the Control Statement Analysis phase for halts and error messages (Figure 5-9).

Byte	Label	Contents
0	FSSYSM	B'11010110' — Indicates halt and system print
1	None	C'F' — Indicates Fiel Delete program
2-3	FSHALT	Disk System — Characters to be used for halt display lights Model 6 — Code to be converted for halt display lights
4	None	Operator options X'01' = Immediate cancel X'02' = Control cancel X'04' = Retry X'08' = Proceed
5-6	FSSUB	C'ƀƀ' = Indicates submessage area

Figure 5-9. FSSYSM Table

BUFFERS

FSBUF Input Buffer

FSBUF, a 96-byte input buffer in the Control Statement Analysis phase, contains control statement images. The last byte is labeled FSCEND.

FSSCOT Output Buffer

FSSCOT, a 128-byte output area, contains the valid keyword operands. It is passed from the Keyword Syntax Scan routine to the Control Statement Analysis phase.

FVIO I/O Buffer

FVIO, a 512-byte input/output buffer in the Mainline phase, is used to read the volume label and VTOC index.

FFIO I/O Buffer

FFIO, a 64-byte input/output buffer in the Mainline phase, is used to read the file labels.

WORK AREAS

FSWRK Work Area

FSWRK is a 47-byte work area defined in the tables of constants FSFLG of the Control Statement Analysis phase ($DELT1). The address of FSWRK is passed to SYSIN when FSFLG is passed to SYSIN by the Control Statement Analysis phase.

PART VI
DISK COPY/DUMP

The IBM System/3 Disk Copy/Dump program has two unique functions: copy pack and copy file. They are used to:

- Copy the contents of one disk onto another.
- Copy a data file from one disk to another.
- Copy a data file from one location to another on the same disk.
- Print the contents of a data file.

If the user requests the copy pack function of the Disk Copy/Dump program, the entire contents of one disk pack are transferred to another disk pack without regard to file organization or type. If the user requests the copy file function of the Disk Copy/Dump program, the selected files may be copied or have records deleted; indexed files may be reorganized or records deleted. Printed copies of the data files may also be requested by the copy file function.

SYSTEM REQUIREMENTS

The IBM System/3 Disk Copy/Dump program uses the following minimum system configuration:

Disk System

- IBM 5410 Processing Unit, Model A12
- IBM 5203 Printer, Model A1
- IBM 5424 Multi-Function Card Unit, Model A
- IBM 5444 Disk Storage Drive, Model 2

Model 6

- IBM 5406 Processing Unit, Model B2 (Includes the keyboard with eight command keys)
- IBM 5213 Printer, Model 1
- IBM 5444 Disk Storage Drive, Model 2

The following optional features are supported:

- IBM 5496 Data Recorder, Model 1 with System/3 Model 6 On-line Feature (Used as the Sysin device)
- IBM 2265 Display Station, Model 2 (Cathode Ray Tube; used as Syslog device with IBM 5406 Processing Unit, Models B3 and B4 only)

COMMON FUNCTIONS

Figure 6-1 shows an operational diagram of the common function of the Disk Copy/Dump program.

When the user loads the Disk Copy/Dump program, the Root phase ($COPY) receives control. The Root phase passes control to the Control Statement Analysis phase ($COSYN). The Control Statement Analysis phase reads the control statements from SYSIN (using the information from the CSYSIN table) or the procedures library into the CSCARD input buffer. If the logging device is on, CSCTLP is passed to the Halt/Syslog transient and the control statements are printed. The control statements are then checked for validity.

Two tables of keyword parameter information (CSPKTL and CSFILT) are built by the Control Statement Analysis phase. The address of CSPKTL or CSFILT (depending on whether a copy pack or a copy file type of control statement is read) is placed in CSSYNT. CSSYNT is passed to the Keyword Syntax Scan routine ($$RDS1). The Keyword Syntax Scan routine builds a table of parameters and passes it to the Control Statement Analysis phase in a buffer labeled CSSYOT. CSCTL is built by the Control Statement Analysis phase and passed to the Root phase.

The Root phase then uses either the copy pack function or the copy file function.

Note: CSERRB is passed by the Control Statement Analysis phase to the Halt/Syslog transient to display any error messages.

COPY PACK FUNCTION

Figure 6-2 shows an operational diagram of the copy pack function.

If the copy pack function is requested, the Root phase passes CSCTL to the Main Data Transfer phase ($COPIN) or the Intermediate phase ($COPIM). The Intermediate phase receives control if intermediate copying from one pack on the removable disk on drive one (R1) to the second pack on R1 is required. If intermediate copying is not required, control is passed to the Main Data Transfer phase.

When control is passed to the Main Data Transfer phase, the size of the I/O area is calculated and the pack sizes are compared. The output pack is checked for active files. Tracks 0 and 1 are copied using a work area called CDPIWK, a buffer labeled CDPIBF, and control block CDPIOB. Cylinders from cylinder 4 to the upper limit are copied. The disk pack is read (using control block CDPIOB) by half track, full track, or full cylinder into an output area labeled CDPIBF. The program then exits to the EOJ transient.

If control is passed to the Intermediate phase, space for a temporary file on the fixed disk on drive one (F1) is obtained to copy cylinder 0. The rest of disk pack is copied and read using control blocks CDIIOB, CDOIOB, and CDFIOB, and the buffer labeled CDIBUF. The program exits to the EOJ transient.

Note: CERRTB and COPHLT are halt tables, identical in format, which are passed to the Halt/Syslog transient. The Halt/Syslog transient generates all halts due to errors and displays error messages.

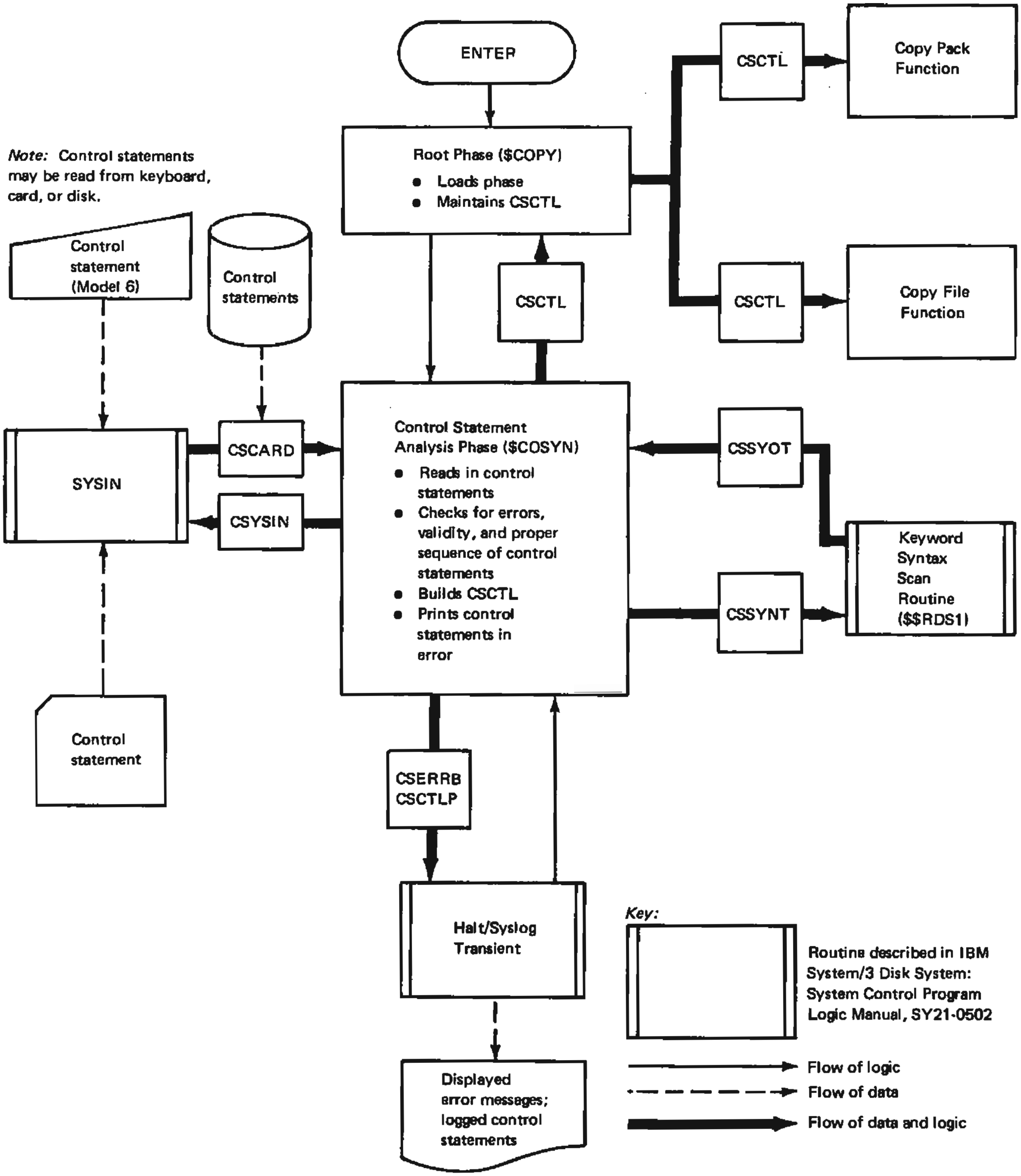

ENTER
Root Phase ($COPY)
● Loads phase
● Maintains CSCTL
CSCTL
Copy Pack Function
CSCTL
Copy File Function
CSCTL
Note: Control statements may be read from keyboard, card, or disk.
Control statement (Model 6)
Control statements
SYSIN
CSCARD
CSYSIN
Control Statement Analysis Phase ($COSYN)
● Reads in control statements
● Checks for errors, validity, and proper sequence of control statements
● Builds CSCTL
● Prints control statements in error
CSSYOT
CSSYNT
Keyword Syntax Scan Routine ($$RDS1)
Control statement
CSERRB
CSCTLP
Halt/Syslog Transient
Displayed error messages; logged control statements
Key:
Routine described in IBM System/3 Disk System: System Control Program Logic Manual, SY21-0502
Flow of logic
Flow of data
Flow of data and logic

Figure 6-1. General Diagram for the Common Functions of the Disk Copy/Dump Program

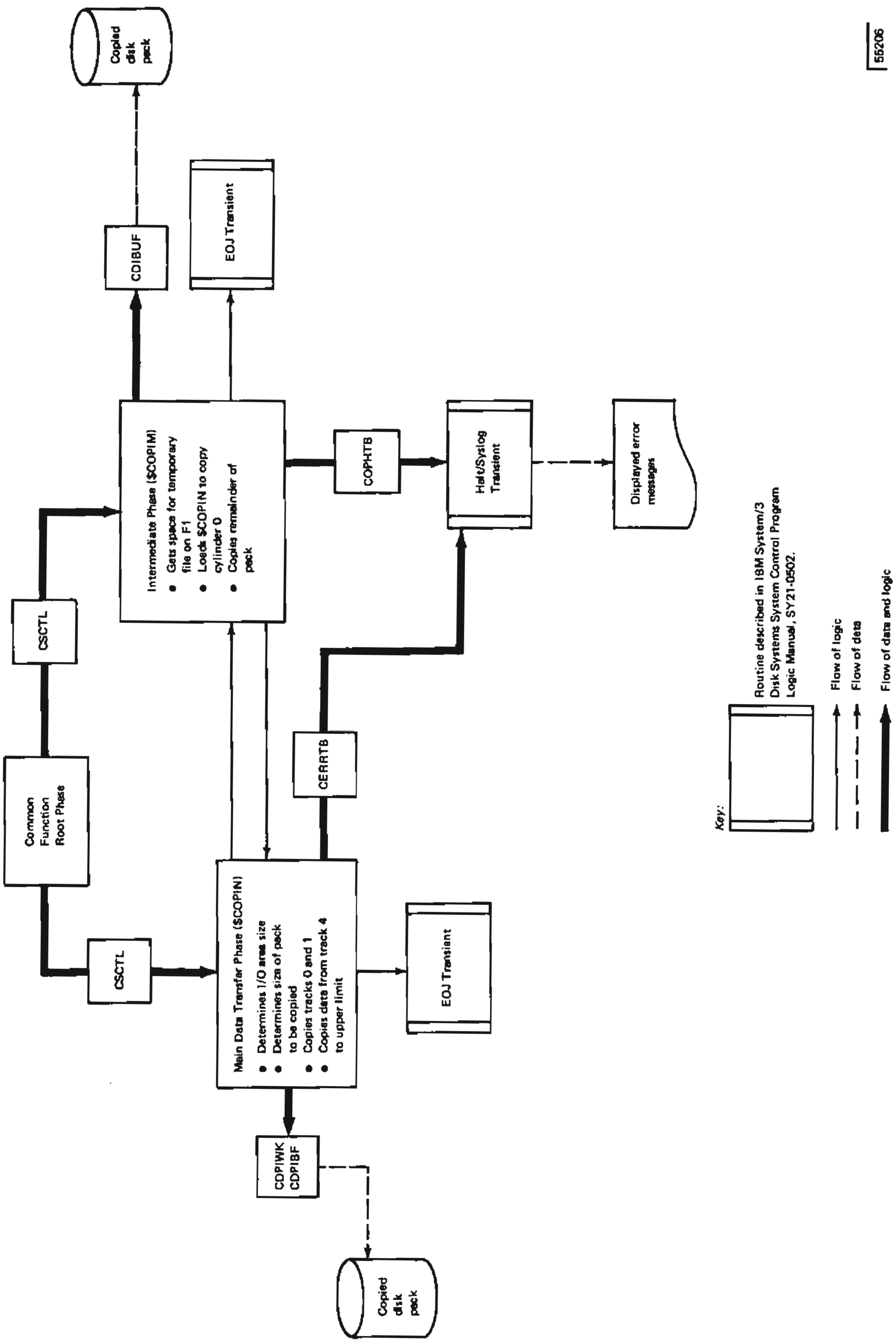

Copied disk pack
CDIBUF
EOJ Transient
CSCTL
Intermediate Phase ($COPIM)
• Gets space for temporary file on F1
• Loads $COPIN to copy cylinder 0
• Copies remainder of pack
COPHTB
Halt/Syslog Transient
Displayed error messages
Common Function Root Phase
CERRTB
CSCTL
Main Data Transfer Phase ($COPIN)
• Determines I/O area size
• Determines size of pack to be copied
• Copies tracks 0 and 1
• Copies data from track 4 to upper limit
EOJ Transient
CDPIWK
CDPIBF
Copied disk pack
Key:
Routine described in IBM System/3 Disk Systems System Control Program Logic Manual, SY21-0502.
Flow of logic
Flow of data
Flow of data and logic
55206

Figure 6-2. Operational Diagram for the Copy Pack Function of the Disk Copy/Dump Program

COPY FILE FUNCTION

Figure 6-3 shows an operational diagram of the copy file function.

If the copy file function is requested, the Initialization phase ($COFIN) receives control and CSCTL is passed to it by the Root phase. This phase does a special open to the input and output files (using CFIN and CFOUT) to obtain record length and file type. It analyzes the attributes, finds storage sizes, and saves record lengths. This information is placed in a DTF and stored in the Root phase immediately following CSCTL. Control is then passed to one of the following five phases. Register 1 points to CSCTL and the DTFs and is input to each of these phases:

- *Simple Copy Files ($CONDP)* copies an indexed, non multi-volume direct, or consecutive file that does not require printing or deletion of a record or reorganizing. This phase uses CONEND as a record area for one logical record.

- *Copy Non-Indexed Files ($COREC)* copies a consecutive or non multi-volume direct file and deletes and/or prints a record.

- *Non-Reorganization Indexed Sequential File ($COCIO)* prevents reorganization of an indexed file and allows deletion and printing of a record. This phase uses COIEND as a buffer for one logical record.

- *Reorganization Indexed Sequential Files ($COREI)* copies an indexed sequential file, allows deletion and printing of a record and reorganization. This phase uses COIEND as a buffer for one logical record.

- *Multi-Volume Direct Files ($COMVD)* copies a multi-volume direct file and allows printing.

- *Intermediate Control ($COFIM)* receives control if intermediate copying (from one pack on R1 to another pack on R1) is requested. The Intermediate Control phase receives the DTFs and opens input one. It then determines if main storage space is available and preallocates space on the fixed disk on drive one (F1). The F1 DTF is opened and added to the DTFs from the Root phase. The Intermediate Control phase then passes control to one of four phases. Register 1 points to CSCTL and the DTFs and is input to each of these phases. Each phase uses CFIOB as a control block and the CFVLBF buffer for reading the volume label.

  1. *Intermediate Reorganization ($COIRY)* copies and reorganizes an indexed sequential file, allows record deletion from an output file, and allows printing with no output on disk.

  2. *Intermediate Non-Reorganization ($COIRN)* copies an indexed sequential file with no reorganization and allows record deletion from the output file.

  3. *Intermediate Consecutive ($CONOI)* copies a consecutive or non multi-volume direct file and allows record deletion (if not direct) and printing with output on disk.

  4. *Intermediate Multi-Volume Direct ($COIMD)* copies multi-volume direct files and allows printing.

- *End of Job Phase ($COEOJ)* puts the correct volume sequence number and file type in the output if copying only one volume of a multi-volume file.

Note 1: All phases except Simple Copy Files, Initialization, and Intermediate Control phases call the Print phase. The Print phase loads SYSLIST and passes a table of information (CPLIST) and print buffer CPBUF to it. The Print phase prints a record and changes unprintable characters to vertical hexadecimal characters. Control is then returned to the calling phases. The Print phase is loaded by the Non-Reorganization Indexed Sequential Files phase.

Note 2: All phases except Initialization, Intermediate Control, and Print phases exit to the EOJ transient or to the EOJ phase which then exits to the EOJ transient.

Note 3: CFERR, CORHLT, CRIHLT, and CHTTB are halt tables which are identical in format and are passed to the Halt/Syslog transient. The Halt/Syslog transient generates all halts due to errors and displays error messages. If the logging device is on, the Copy Non-Indexed Files phase passes CNOTAB to the Halt/Syslog transient to print error messages on the logging device.

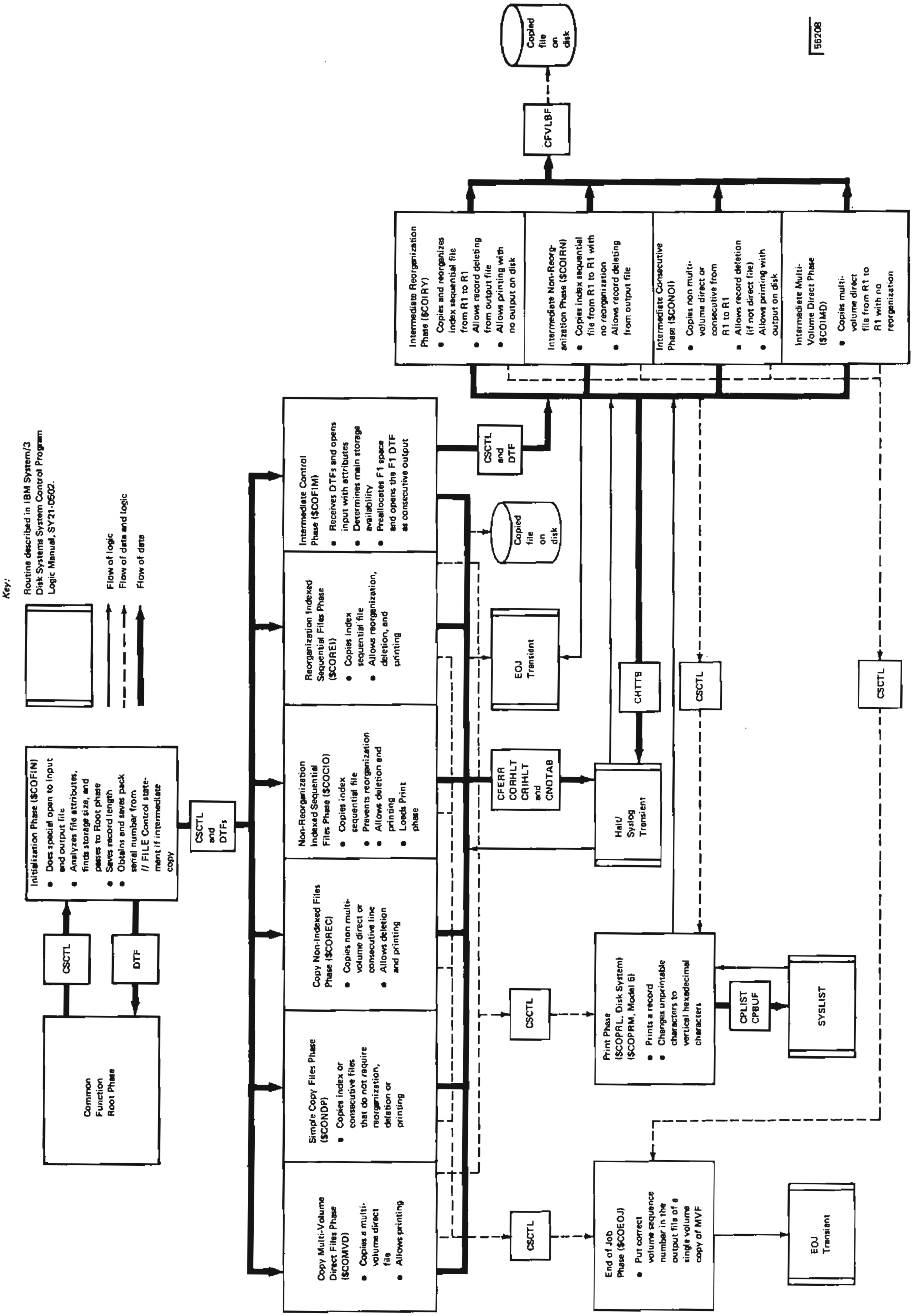

Key:
Routine described in IBM System/3 Disk Systems System Control Program Logic Manual, SY21-0502.
Flow of logic
Flow of data and logic
Flow of data
Common Function Root Phase
CSCTL
DTF
Initialization Phase ($COFIN)
• Does special open to input and output file
• Analyzes file attributes, finds storage size, and passes to Root phase
• Saves record length
• Obtains and saves pack serial number from // FILE Control statement if intermediate copy
CSCTL and DTFs
Copy Multi-Volume Direct Files Phase ($COMVD)
• Copies a multi-volume direct file
• Allows printing
Simple Copy Files Phase ($CONDP)
• Copies index or consecutive files that do not require reorganization, deletion or printing
Copy Non-Indexed Files Phase ($COREC)
• Copies non multi-volume direct or consecutive line
• Allows deletion and printing
Non-Reorganization Indexed Sequential Files Phase ($COCIO)
• Copies index sequential file
• Prevents reorganization
• Allows deletion and printing
• Loads Print phase
Reorganization Indexed Sequential Files Phase ($COREI)
• Copies index sequential file
• Allows reorganization, deletion, and printing
Intermediate Control Phase ($COFIM)
• Receives DTFs and opens input with attributes
• Determines main storage availability
• Preallocates F1 space and opens the F1 DTF as consecutive output
CSCTL and DTF
Copied file on disk
EOJ Transient
CFERR CORHLT CRIHLT and CNOTAB
Halt/Syslog Transient
CHTTB
CSCTL
CSCTL
CSCTL
Print Phase ($COPRL, Disk System) ($COPRM, Model 6)
• Prints a record
• Changes unprintable characters to vertical hexadecimal characters
CPLIST CPBUF
SYSLIST
CSCTL
End of Job Phase ($COEOJ)
• Put correct volume sequence number in the output file of a single volume copy of MVF
EOJ Transient
Intermediate Reorganization Phase ($COIRY)
• Copies and reorganizes index sequential file from R1 to R1
• Allows record deleting from output file
• Allows printing with no output on disk
Intermediate Non-Reorganization Phase ($COIRN)
• Copies index sequential file from R1 to R1 with no reorganization
• Allows record deleting from output file
Intermediate Consecutive Phase ($CONOI)
• Copies non multi-volume direct or consecutive from R1 to R1
• Allows record deletion (if not direct file)
• Allows printing with output on disk
Intermediate Multi-Volume Direct Phase ($COIMD)
• Copies multi-volume direct file from R1 to R1 with no reorganization
CFVLBF
Copied file on disk
56208

Figure 6-3. Operational Diagram for the Copy Files Function of the Disk Copy/Dump Program

This section contains a functional description of each phase included in the IBM System/3 Copy/Dump program. Storage maps and flowcharts are provided where necessary.

COMMON FUNCTIONS

Initially the Root phase receives control and loads the other phases into storage. The control statements are read and checked for validity. A table of constants (CSCTL) containing information found on the control statements is built and maintained by this group of phases. This table, CSCTL, is used by both the copy file and copy pack functions.

Root Phase ($COPY)

Entry Point: CROOT

Chart: None

Functions:

- Calls the various phases into main storage.
- Maintains a table of constants (CSCTL) that is passed to various phases.

Input: Table of constants (CSCTL) from Control Statement Analysis phase

Output: Table of constants (CSCTL) to various phases

Exits:

- Normal
  1. Supervisor to load the phases
  2. Entry point of the phase called
- Error — None

Control Statement Analysis Phase ($COSYN)

Entry Point: CSBEG

Chart: FA

Functions:

- Reads control statements into input buffer CSCARD.

  Note: If the control statements are read from the job scheduler work area, no retry is allowed on halts.
- Checks for errors and proper sequence of the control statements.
- Builds a table of constants (CSCTL) containing information found on the control statements and passes it to the Root phase.

Input: Control statements read from SYSIN or the job scheduler work area

Output:

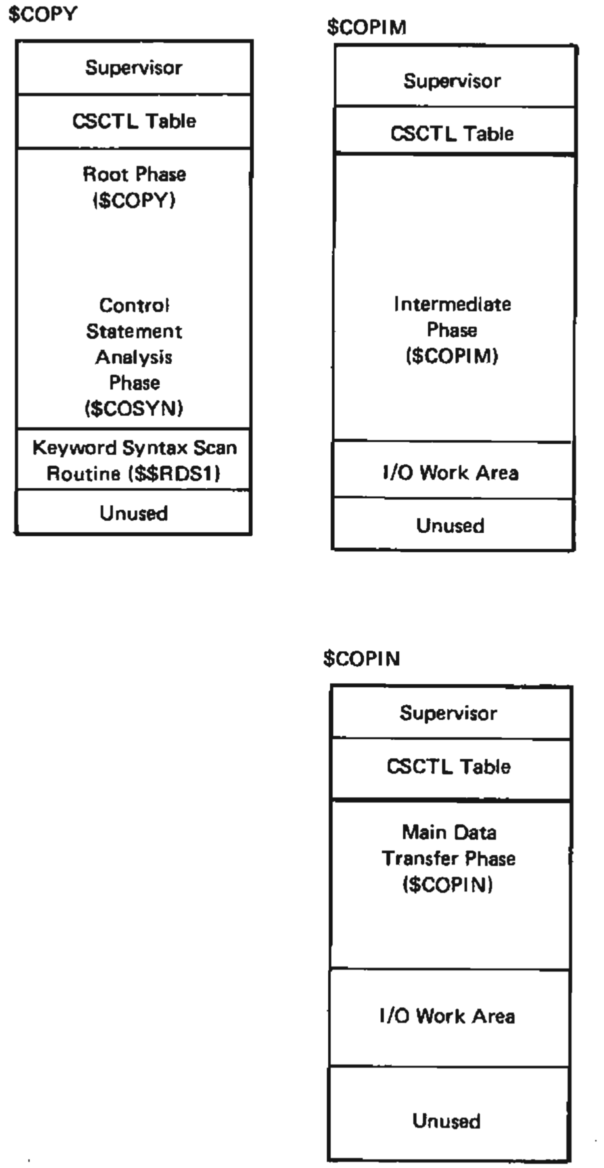
- Table of constants (CSCTL)
- Printed control statements if the logging device is on

Exits:

- Normal
  1. Root phase after building table of constants CSCTL
  2. Keyword Syntax Scan routine to check for valid keyword parameter
- Error — Halt/Syslog transient

COPY PACK FUNCTION

The copy pack function transfers the entire contents of one disk to another without regard to data type, format, or contents. The receiving pack must not contain any active files and must be the same logical size. Figure 6-4 is a storage map for the copy pack function.



Note: In each phase there is a dark line. The area below the line is newly loaded by that phase. The area above the line remains the same as it was in the preceding phase.

Figure 6-4. Storage Maps of the Copy Pack Function of the Disk Copy/Dump Program

Main Data Transfer Phase ($COPIN)

Entry Point: COPINT

Chart: FB

Functions:

- Determines the size of the input/output area that may be used. The larger the input/output area, the faster the copy may occur. The minimum area is 3,072 bytes and the maximum is 12,288 bytes (full cylinder).
- Checks the size of the pack to be copied.
- Changes the last byte of the output IPL label (track 0, sector 0) to equal the Q code of the output pack:
  1. X'A0' if R1 (removable disk on drive 1)
  2. X'A8' if F1 (fixed disk on drive 1)
- Copies all of track 0 and 1 except for:
  1. Volume serial number
  2. Owner ID
  3. Alternate track information
  4. Outboard data recorder/statistical data recorder (OBR/SDR); program protection sectors
- Determines whether the output volume contains active files. If active files are present, a halt occurs.
- Writes out a volume serial number of (,,,,,,) and directory pointers of 'FFFF'; saves the good serial number and pointers in an unused area of the label.

Input: Table of constants (CSCTL) from the Control Statement Analysis phase

Output: Tracks 0 and 1, cylinder 4 to last customer cylinder copied and updated

Exits:

- Normal – EOJ transient to end job
- Error – Halt/Syslog transient

Intermediate Phase ($COPIM)

Entry Point: COPIM

Chart: FC

Functions:

- Determines the size of the input/output area that may be used. The larger the input/output area, the faster the copy may occur. The minimum area is 3,072 bytes and the maximum is 12,288 bytes (full cylinder).
- Checks the size of the pack to be copied.
- Copies cylinder 0.
- Copies all of track 0 and 1 except for:

  1. Volume serial number
  2. Owner ID
  3. Alternate track information
  4. Outboard data recorder/statistical data recorder (OBR/SDR); program protection sectors

- Obtains space on fixed disk on drive 1 (F1) to use as a temporary file.
- Halts and issues a message to operator to mount the correct pack.
- Determines whether the output volume contains active files. If active files are present, a halt occurs.
- Writes out the good volume label, destroys the label of (,,,,,,), and restores the pointers that were saved.

Input: Table of constants pointed to be XR1

Output: Copied disk

Exits:

- Normal – EOJ transient to end job
- Error – Halt/Syslog transient

COPY FILES FUNCTION

The copy files function analyzes the file attributes to determine (1) if a data file is to be copied from one disk to another or from R1 to R1 on different packs (intermediate copying) (2) the type of file to be copied; indexed, direct, consecutive, and whether printing, reorganization, or deletion is required. The correct phase needed to fulfill the request is then loaded and the file is copied. Figure 6-5 is a storage map for the copy files function.

Initialization Phase ($COFIN)

Entry Point: COFI

Chart: FD

Functions:

- Determines the type of copy file requested and the phases needed to fulfill the request.
- Determines the storage size available for the program.
- Obtains the record length and passes it to the Root phase to be saved.

Input: XR1 points to a table of constants (CSCTL) from the Root phase

Output:

- DTFs stored in the Root phase
- XR1 points to table of constants (CSCTL)

Exits:

- Normal – Supervisor
- Error – Halt/Syslog transient

Simple Copy Files Phase ($CONDP)

Entry Point: $CONDP

Chart: FE

Functions: Copies the following files when printing, deleting, or re-organizing is not required:

- Indexed sequential
- Consecutive
- Direct (when not multi-volume)

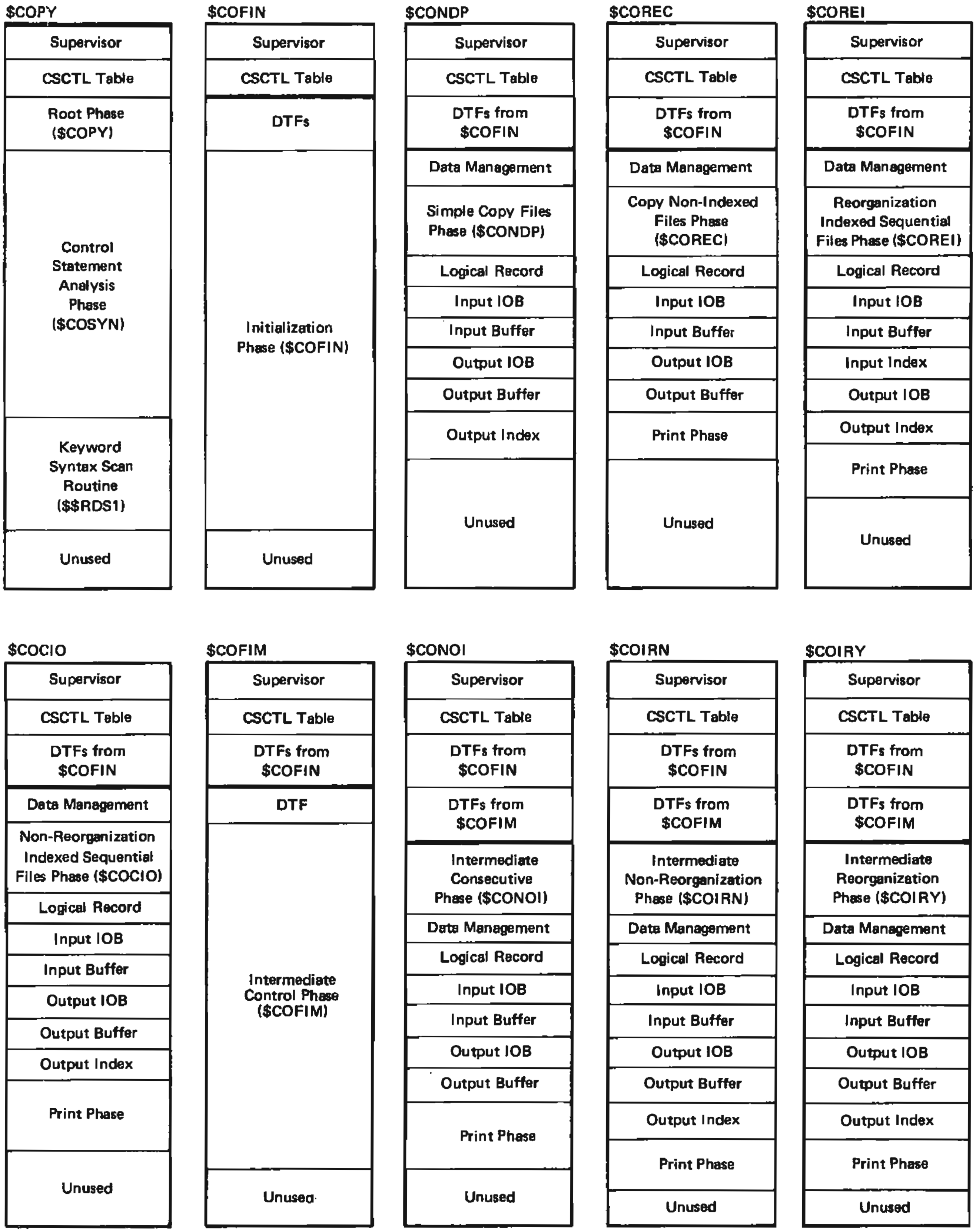
Input:

- Disk file to be copied
- XR1 points to a table of constants (CSCTL) from the Root phase
- DTFs immediately following the CSCTL table

Output: Copied file on disk

Exits:

- Normal
  1. Supervisor to open and close files, load halt transient, and load EOJ transient
  2. EOJ transient routine to end job
  3. EOJ phase ($COEOJ) if one volume of multivolume file
- Error – Halt/Syslog transient



Note: In each phase there is a dark line. The area below the line is newly loaded by that phase. The area above the line remains the same as it was in the preceding phase.

Figure 6-5. Storage Maps of the Copy Files Function of the Disk Copy/Dump Program (Part 1 of 2)

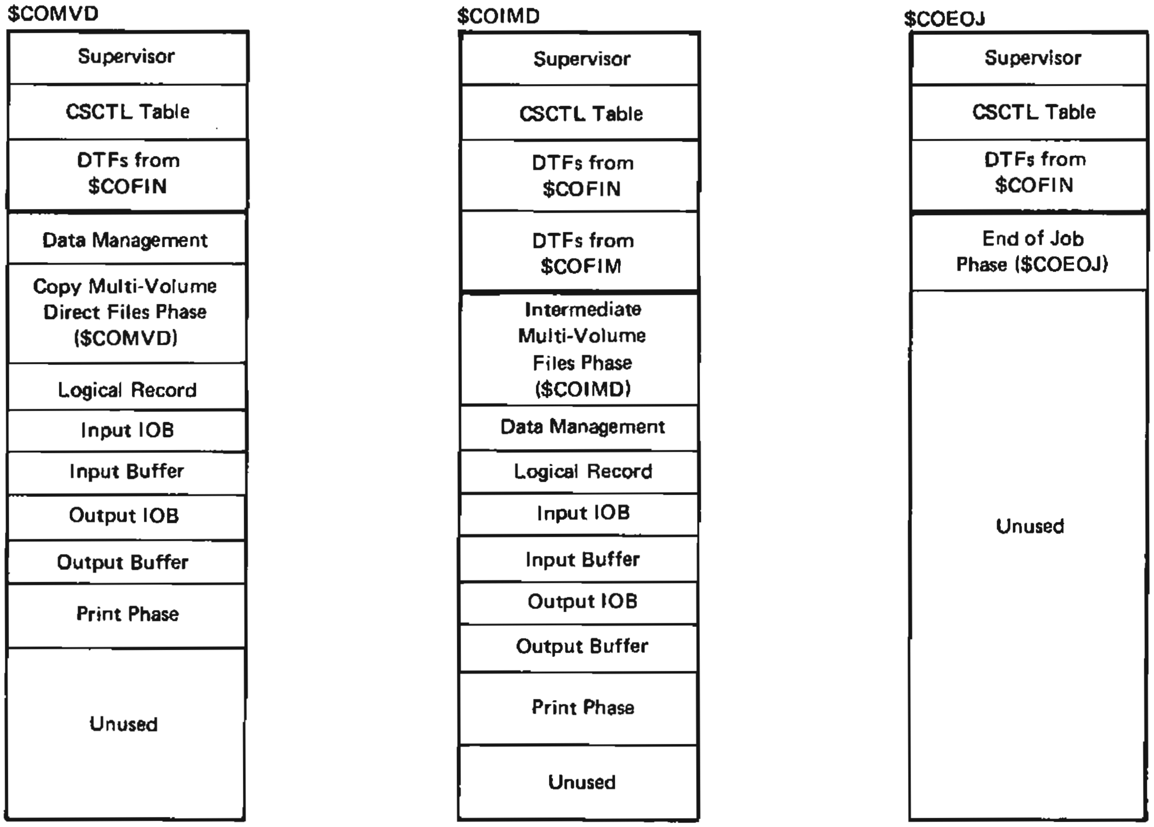


Figure 6-5. Storage Maps of the Copy Files Function of the Disk Copy/Dump Program (Part 2 of 2)

Copy Non-Indexed Files Phase ($COREC)

Entry Point: COREC

Chart: FF

Functions:

- Copies a consecutive file or a direct file that is not multi-volume.
- Deletes a record if specified (consecutive file only).
- Prints a record if specified.

Input:

- XR1 points to a table of constants (CSCTL) from the Root phase
- DTFs immediately following the CSCTL table

Output:

- Copied file on disk
- Printed file

Exits:

- Normal
  1. Print phase to print a record
  2. EOJ transient to end job
  3. EOJ phase ($COEOJ) if one volume of multi-volume file
- Error – Halt/Syslog transient

Non-Reorganization Indexed Sequential Files Phase ($COCIO)

Entry Point: COINDX

Chart: FG

Functions:

- Copies an indexed sequential file.
- Allows the file to be reorganized.
- Allows specified records to be deleted.
- Allows records to be printed.

Input:

- XR1 points to a table of constants (CSCTL) passed from the Root phase
- DTFs immediately following the CSCTL table

Output:

- Copied file on disk
- Printed file

Exits:

- Normal
  1. Print phase to print record
  2. EOJ transient to end job
  3. EOJ phase ($COEOJ) if one volume of multi-volume file
- Error – Halt/Syslog transient

Reorganization Indexed Sequential Files Phase ($COREI)

Entry Point: COINDX

Chart: FG

Functions:

- Copies an indexed sequential file.
- Prevents reorganization.
- Allows records to be deleted.
- Allows records to be printed.

Input:

- XR1 points to a table of constants (CSCTL) from the Root phase
- DTFs immediately following the CSCTL table

Output:

- Copied file on disk
- Printed file

Exits:

- Normal
  1. Print phase to print record
  2. EOJ transient routine to end job
  3. EOJ phase ($COEOJ) if one volume of multi-volume file
- Error – Halt/Syslog transient

Intermediate Control Phase ($COFIM)

Entry Point: CFBEG

Chart: FH

Functions:

- Receives the DTFs and opens input with the correct attributes.
- Determines if enough main storage is available for the program.
- Preallocates space on the fixed disk on drive 1 (F1) and opens the F1 DTF as consecutive output.

Input:

- XR1 points to a table of constants (CSCTL) passed from the Root phase
- DTF immediately following the CSCTL table

Output:

- F1 DTF and constants added to CSCTL table
- DTFs immediately following the CSCTL table

Exits:

- Normal – Entry point of phase called
- Error – Halt/Syslog transient

Intermediate Reorganization Phase ($COIRY)

Entry Point: CFBEG

Chart: FI

Functions:

- Copies and reorganizes an indexed sequential file from one pack of the removable disk on drive one (R1) to another pack on R1.
- Allows records to be deleted from the output file.
- Allows printing to be specified with output on disk.

Input:

- Indexed file on the input pack on R1
- XR1 points to the table of constants (CSCTL) from the Root phase.
- DTF immediately following the CSCTL table

Output: Reorganized file on the output R1

Exits:

- Normal
  1. Halt/Syslog transient for mounting packs
  2. Disk IOS routine for reading the volume label
  3. Print phase to print record
  4. EOJ transient to end job
  5. EOJ phase ($COEOJ) if one volume of multivolume file
- Error – Halt/Syslog transient

Intermediate Non-Reorganization Phase ($COIRN)

Entry Point: CFBEG

Chart: FI

Functions:

- Copies an indexed sequential file from one pack on R1 to another pack on R1 with no reorganization.
- Allows records to be deleted from the output file.

Input:

- Indexed file on the input pack
- XR1 points to a table of constants (CSCTL) from the Root phase
- DTF immediately following the CSCTL table

Output: Indexed file on the output pack

Exits:

- Normal
  1. Halt/Syslog transient for mounting packs
  2. Disk IOS for reading the volume label
  3. Print phase to print a record
  4. EOJ transient to end job
  5. EOJ phase ($COEOJ) if one volume of multi-volume file
- Error – Halt/Syslog transient

Intermediate Consecutive Phase ($CONOI)

Entry Point: CFBEG

Chart: FI

Functions:

- Copies a non-indexed file and a direct file that is not multi-volume from one pack on removable disk on drive one (R1) to another pack on R1.
- Allows records to be deleted from the output file (consecutive file only).
- Allows printing with output on disk.

Input:

- Consecutive file or a direct file that is not multi-volume on the input pack
- XR1 points to a table of constants (CSCTL) from the Root phase.
- DTF immediately following the CSCTL table

Output: Consecutive file or a direct file that is not multi-volume on the output pack on R1

Exits:

- Normal
  1. Halt/Syslog transient for mounting packs
  2. Disk IOS routine for reading the volume label
  3. Print phase to print record
  4. EOJ transient to end job
  5. EOJ phase ($COEOJ) if one volume of multi-volume file
- Error – Halt/Syslog transient

Print Phase ($COPRL – Disk System)
($COPRM – Model 6)

Entry Point: COPRT

Chart: FJ

Functions:

- Prints a record with its key or number.
- Indicates if a record is deleted.
- Prints a scale at the top and bottom of each page.
- Prints the record count.

Input:

- Record address
- Record length
- Record key or number
- An indication in the CSCTL table of constants when a record is deleted

Output: Printed records with a record count

Exits:

- Normal
  1. SYSLIST to print a record
  2. Calling routine once record is printed
- Error – Halt/Syslog transient

Copy Multi-Volume Direct Files Phase ($COMVD)

Entry Point: COREC

Chart: FK

Functions:

- Copies a multi-volume direct file.
- Prints a record if specified.

Input:

- XR1 points to a table of constants (CSCTL) from the Root phase
- DTFs immediately following the CSCTL table

Output:

- Copied file on disk
- Printed file

Exits:

- Normal
  1. Print phase to print a record
  2. EOJ transient to end job
  3. EOJ phase ($COEOJ) for volume sequence number
- Error – Halt/Syslog transient

Intermediate Multi-Volume Direct Phase ($COIMD)

Entry Point: CFBEG

Chart: FI

Functions:

- Copies a multi-volume direct file from the pack on the removable disk on drive one (R1) to another pack on R1.
- Allows printing with output on disk.

Input:

- Multi-Volume direct file on the input pack
- XR1 points to a table of constants (CSCTL) from the Root phase
- DTF immediately following the CSCTL table

Output: Multi-Volume direct file on the output pack on R1

Exits:

- Normal
  1. Halt/Syslog transient for mounting packs
  2. Disk IOS routine for reading the volume label
  3. Print phase to print record
  4. EOJ phase ($COEOJ) for volume sequence number
- Error – Halt/Syslog transient

End of Job Phase ($COEOJ)

Entry Point: COEOJ

Chart: FL

Function: Puts the volume sequence number and file type in the output file when copying one volume of a multi-volume file.

Input:

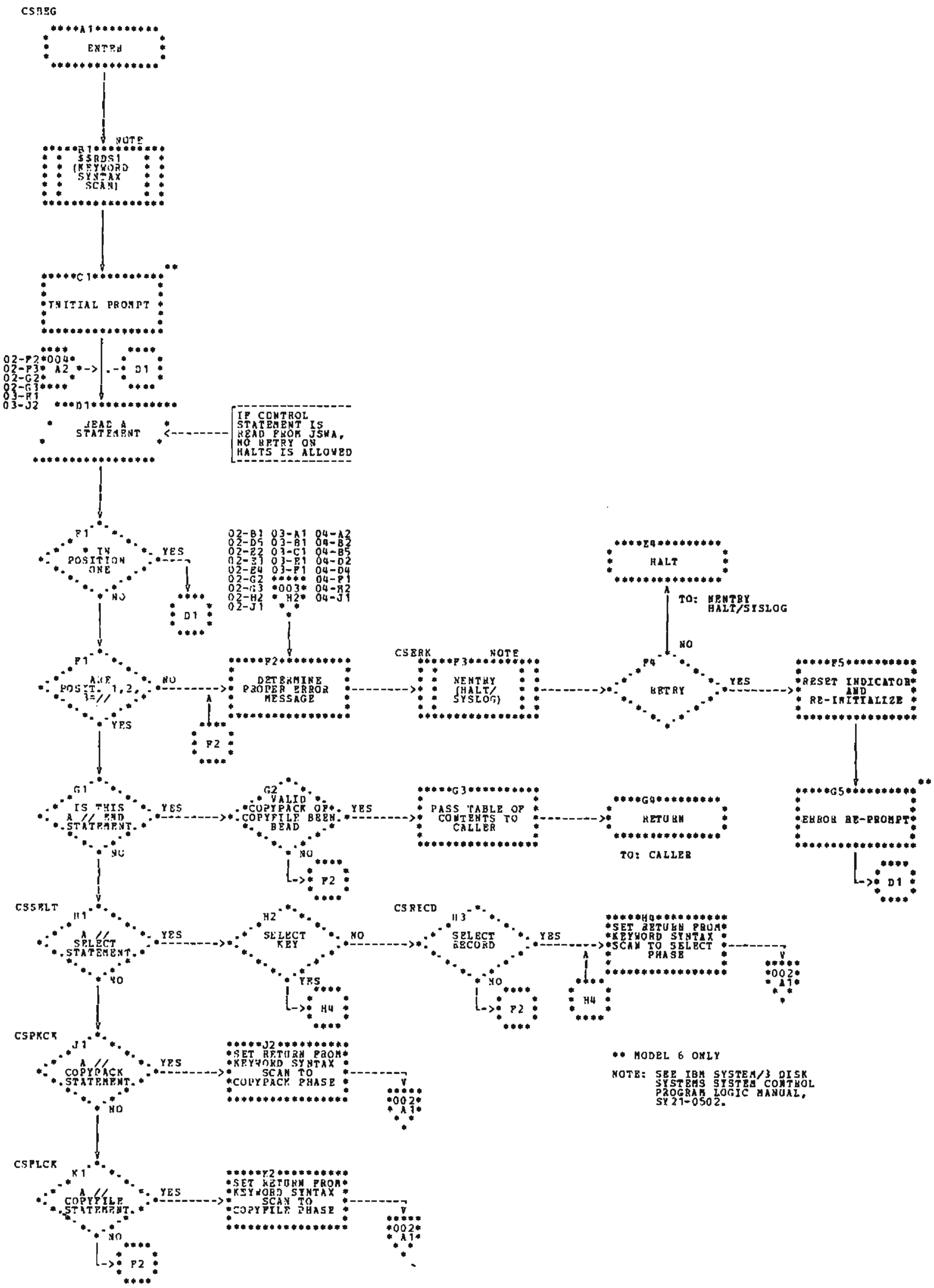
- XR1 points to a table of constants (CSCTL) from the Root phase
- Reads F1 from the scheduler work area
- Reads F7 from the scheduler work area if an indexed file

Output:

- File type and the volume sequence number in F1 of the output file
- Volume sequence number in F7 of the output file if an indexed file

Exits:

- Normal – EOJ transient
- Error – Not applicable



** MODEL 6 ONLY

NOTE: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart FA (Part 1 of 4). Control Statement Analysis Phase

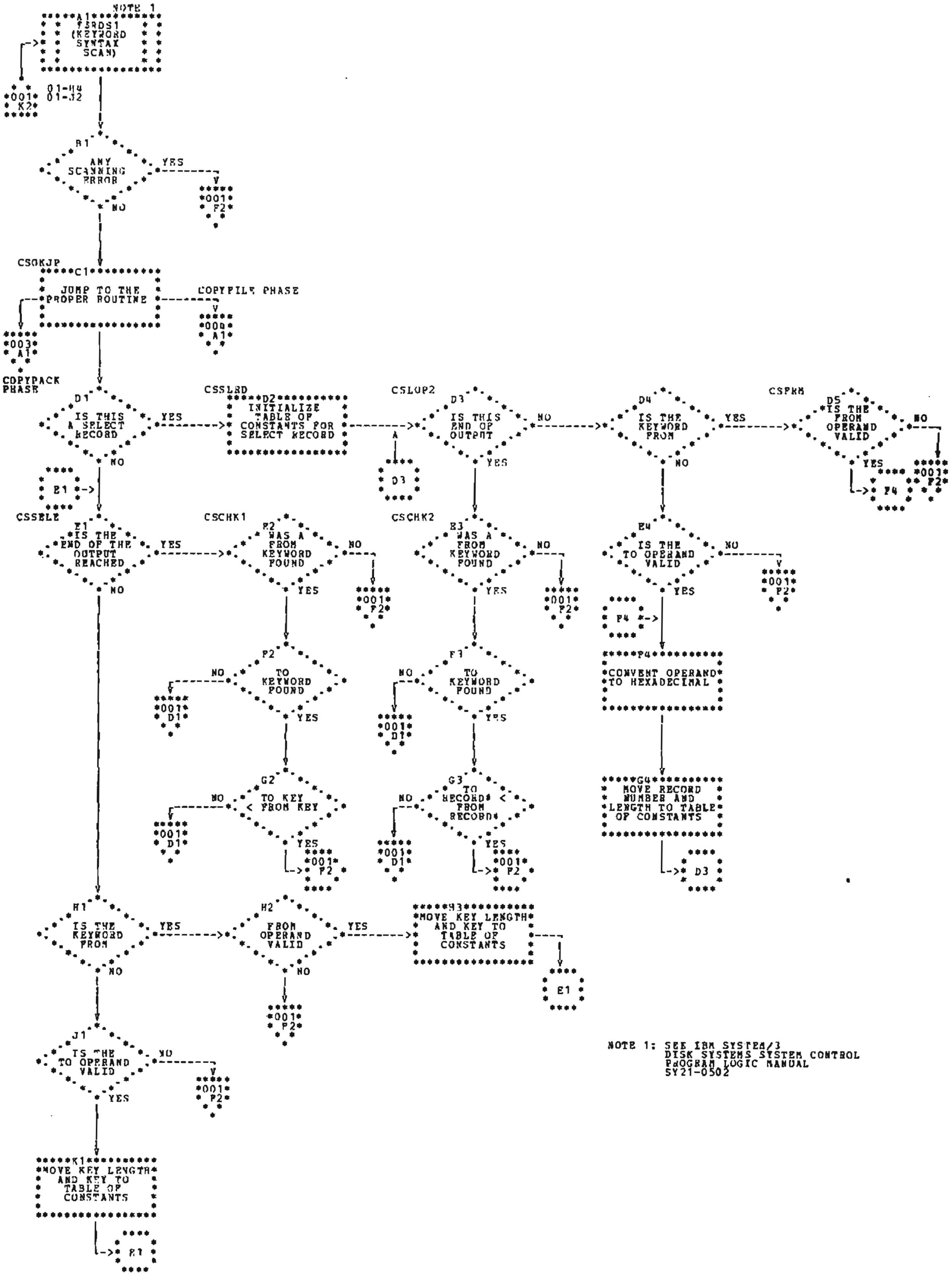


Chart FA (Part 2 of 4). Control Statement Analysis Phase

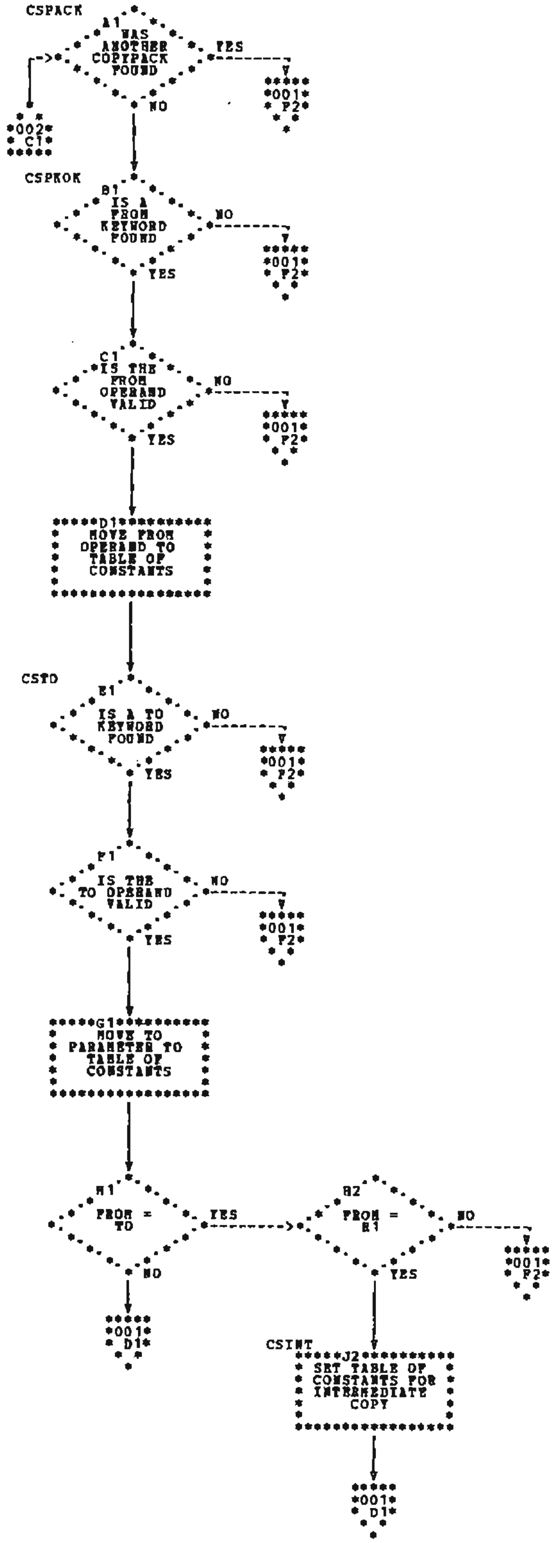


Chart FA (Part 3 of 4). Control Statement Analysis Phase

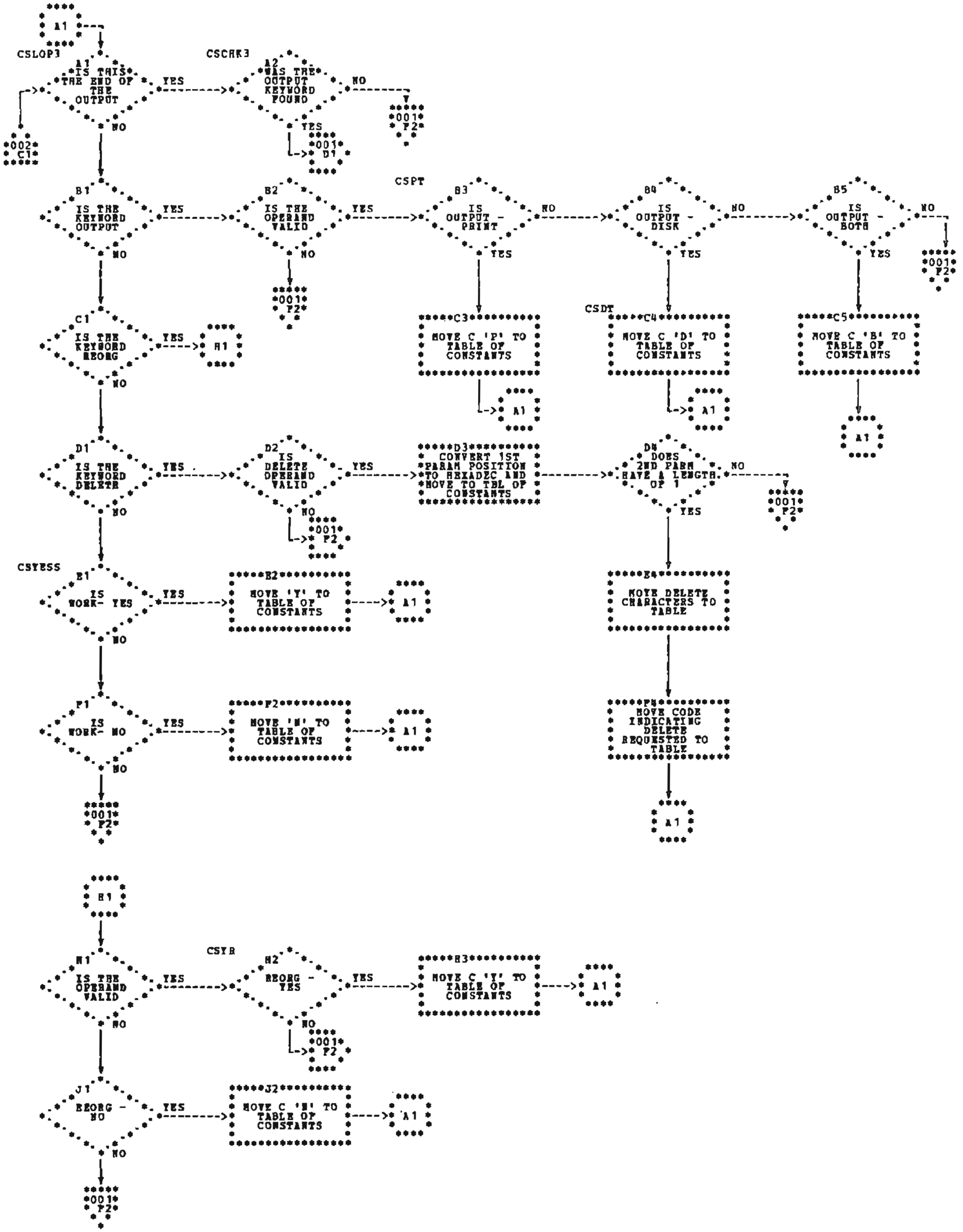


Chart FA (Part 4 of 4). Control Statement Analysis Phase

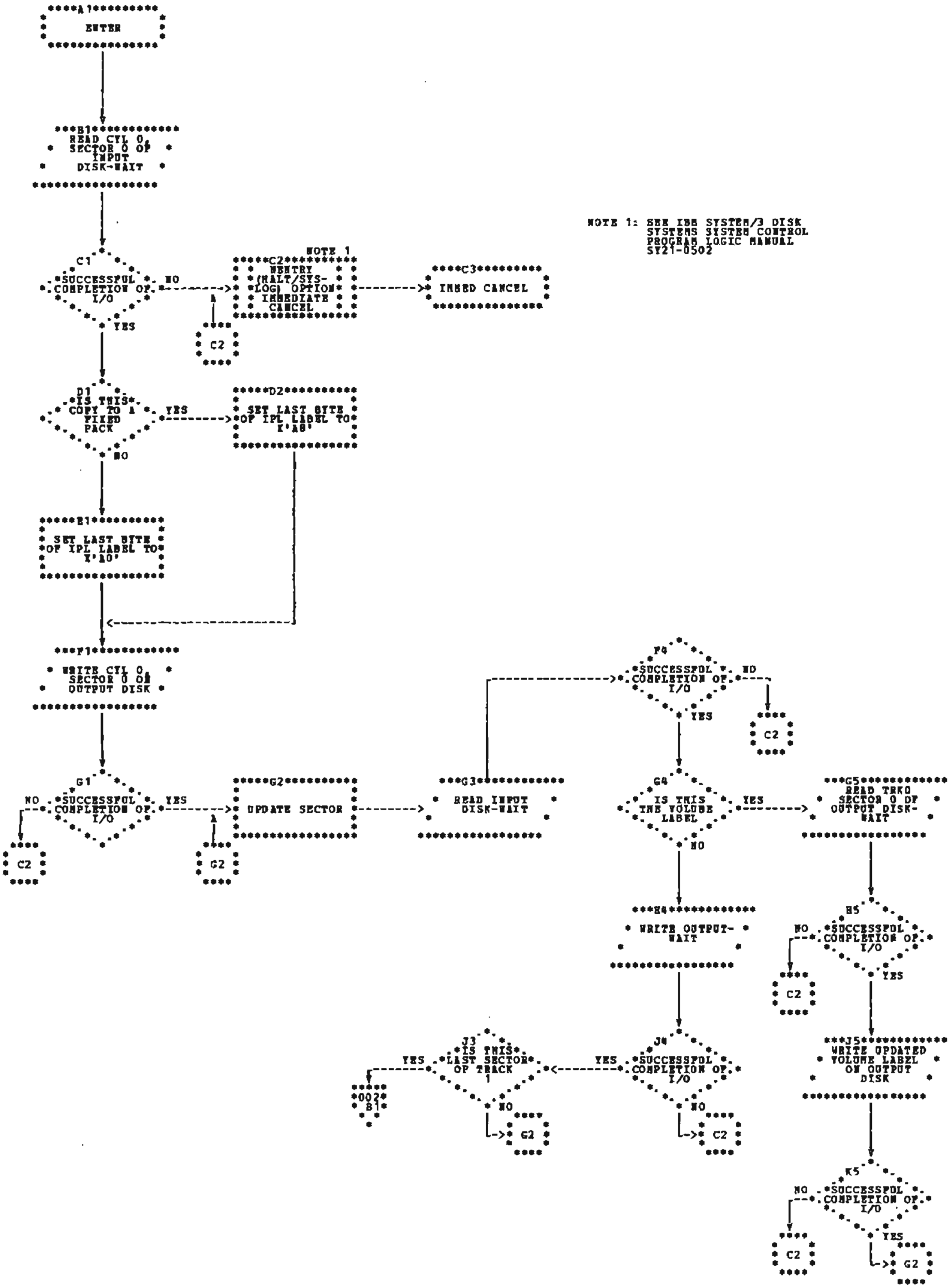
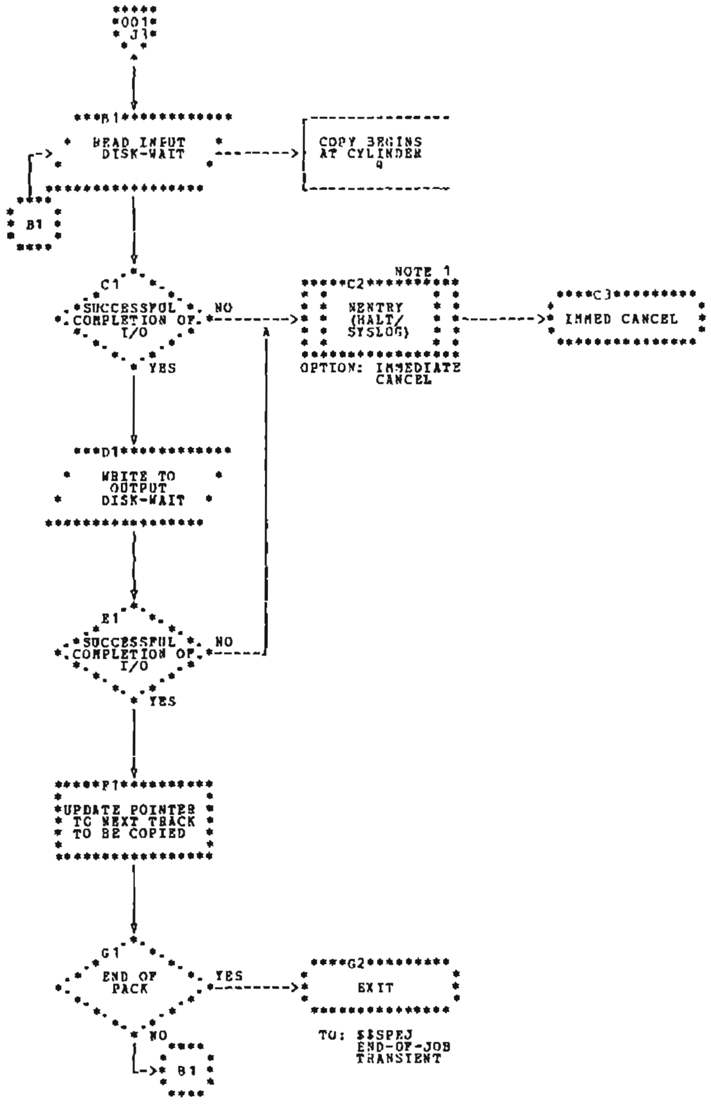
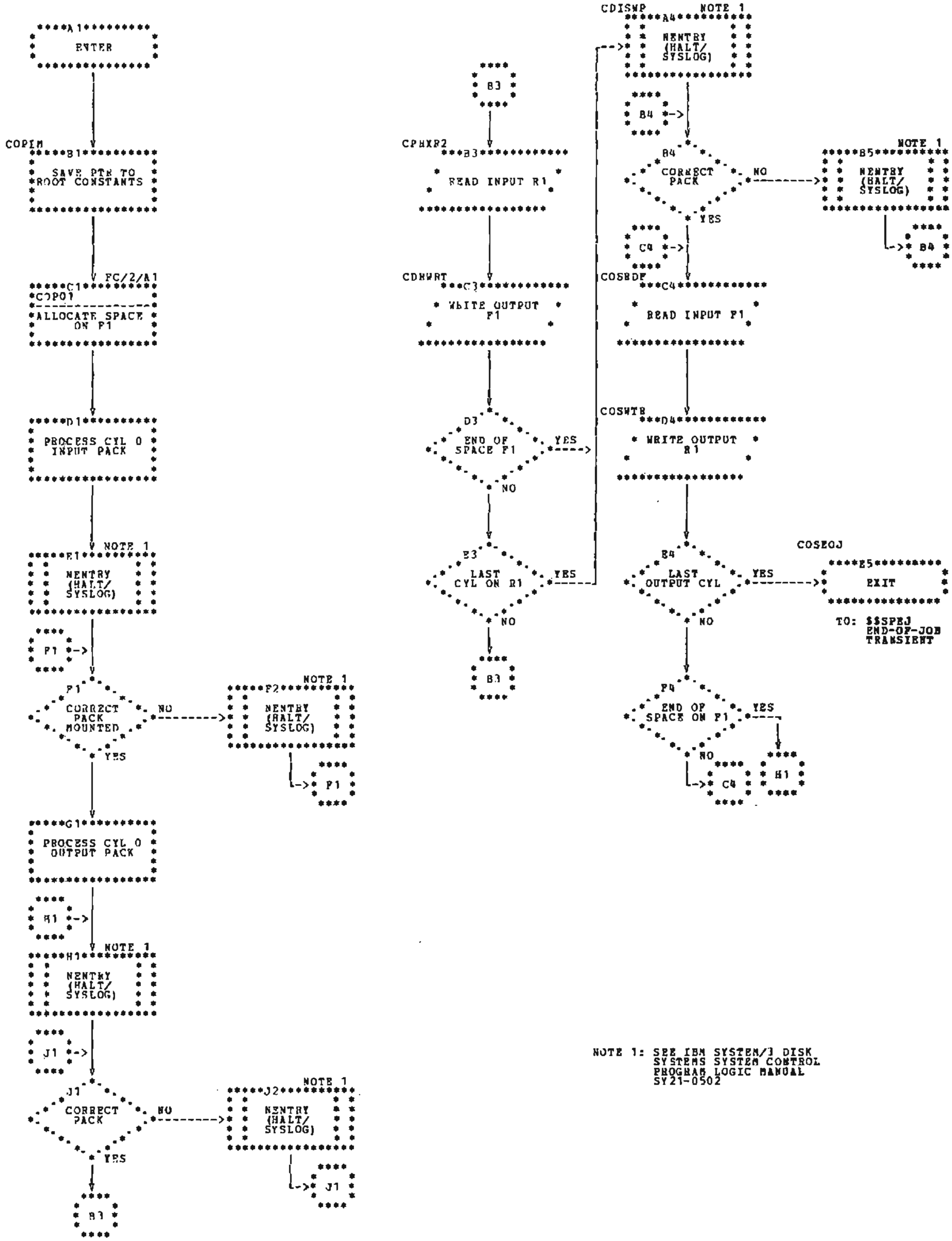

A1
ENTER
B1
READ CYL 0, SECTOR 0 OF INPUT DISK-WAIT
C1
SUCCESSFUL COMPLETION OF I/O
NO
YES
NOTE 1
C2
MENTRY (HALT/SYS-LOG) OPTION IMMEDIATE CANCEL
C2
C3
IMMED CANCEL
NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502
D1
IS THIS COPY TO A FIXED PACK
YES
NO
D2
SET LAST BYTE OF IPL LABEL TO X'A8'
E1
SET LAST BYTE OF IPL LABEL TO X'A0'
F1
WRITE CYL 0, SECTOR 0 ON OUTPUT DISK
G1
SUCCESSFUL COMPLETION OF I/O
NO
YES
C2
G2
UPDATE SECTOR
G2
G3
READ INPUT DISK-WAIT
F4
SUCCESSFUL COMPLETION OF I/O
NO
YES
C2
G4
IS THIS THE VOLUME LABEL
YES
NO
G5
READ TRK0 SECTOR 0 OF OUTPUT DISK-WAIT
H4
WRITE OUTPUT-WAIT
H5
SUCCESSFUL COMPLETION OF I/O
NO
YES
C2
J3
IS THIS LAST SECTOR OF TRACK 1
YES
NO
002 B1
G2
J4
SUCCESSFUL COMPLETION OF I/O
YES
NO
C2
J5
WRITE UPDATED VOLUME LABEL ON OUTPUT DISK
K5
SUCCESSFUL COMPLETION OF I/O
NO
YES
C2
G2

Chart FB (Part 1 of 2). Main Data Transfer Phase


001
J1
B1
READ INPUT DISK-WAIT
COPY BEGINS AT CYLINDER 4
B1
C1
SUCCESSFUL COMPLETION OF I/O
NO
YES
NOTE 1
C2
NENTRY (HALT/ SYSLOG)
OPTION: IMMEDIATE CANCEL
C3
IMMED CANCEL
A
D1
WRITE TO OUTPUT DISK-WAIT
E1
SUCCESSFUL COMPLETION OF I/O
NO
YES
F1
UPDATE POINTER TO NEXT TRACK TO BE COPIED
G1
END OF PACK
YES
NO
G2
EXIT
TO: $$SPEJ END-OF-JOB TRANSIENT
B1

NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502

Chart FB (Part 2 of 2). Main Data Transfer Phase



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502

Chart FC (Part 1 of 2). Intermediate Phase

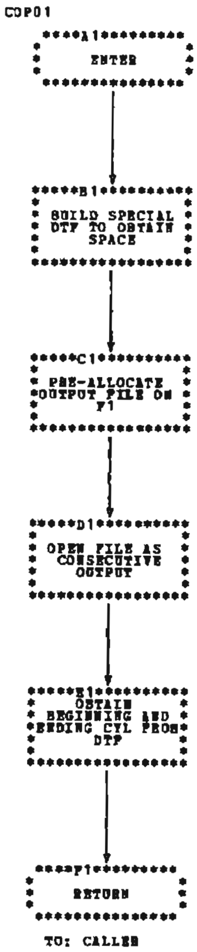
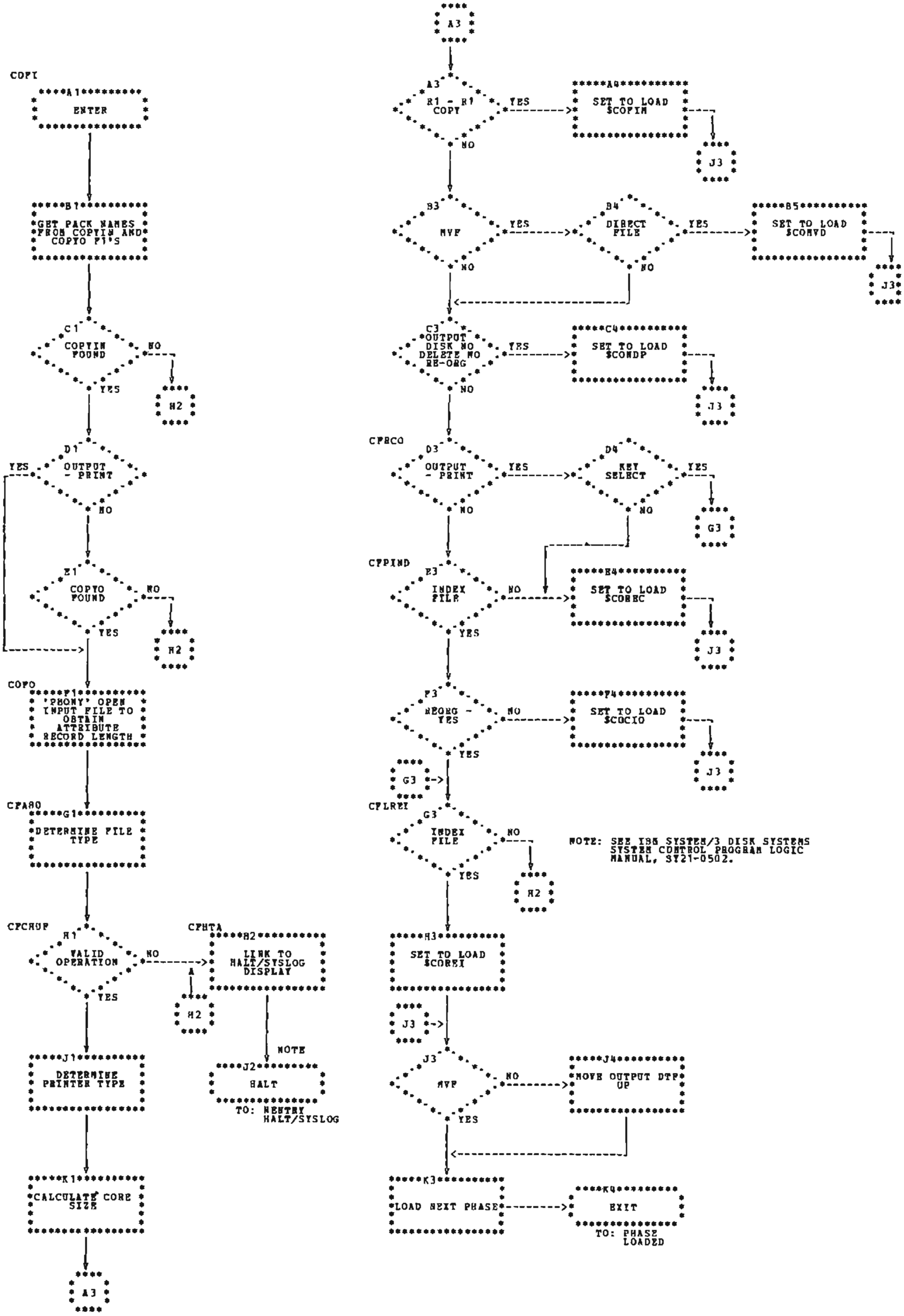

COP01
A1
ENTER
B1
BUILD SPECIAL DTF TO OBTAIN SPACE
C1
PRE-ALLOCATE OUTPUT FILE ON F1
D1
OPEN FILE AS CONSECUTIVE OUTPUT
E1
OBTAIN BEGINNING AND ENDING CYL FROM DTF
F1
RETURN
TO: CALLER

Chart FC (Part 2 of 2). Intermediate Phase



NOTE: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart FD. Initialization Phase

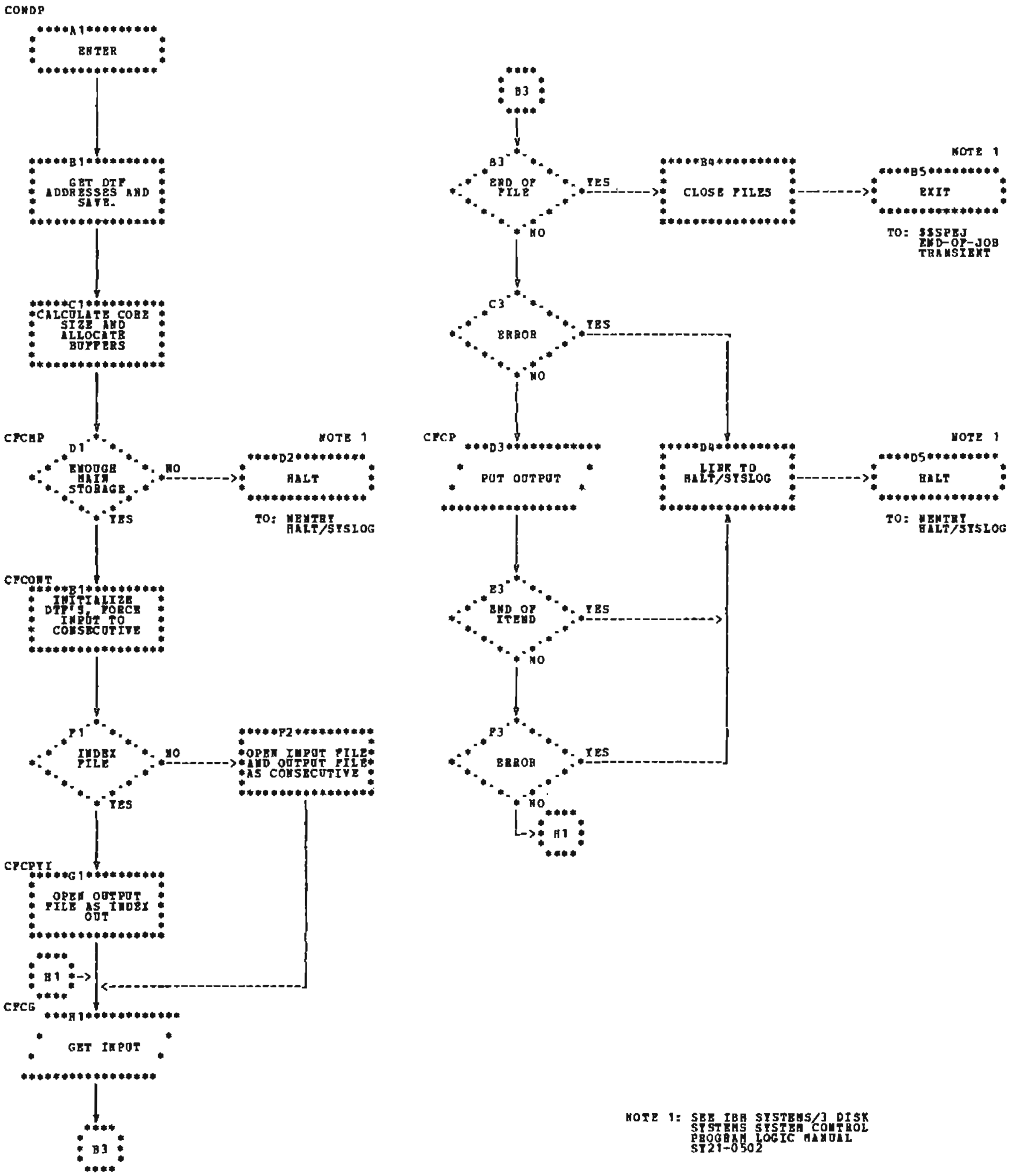

CONDP
A1
ENTER
B1
GET DTF ADDRESSES AND SAVE.
C1
CALCULATE CORE SIZE AND ALLOCATE BUFFERS
CFCHP
D1
ENOUGH MAIN STORAGE
NO
YES
NOTE 1
D2
HALT
TO: NENTRY HALT/SYSLOG
CFCONT
E1
INITIALIZE DTF'S, FORCE INPUT TO CONSECUTIVE
F1
INDEX FILE
NO
YES
F2
OPEN INPUT FILE AND OUTPUT FILE AS CONSECUTIVE
CFCPYI
G1
OPEN OUTPUT FILE AS INDEX OUT
H1
CFCG
H1
GET INPUT
B3
B3
B3
END OF FILE
YES
NO
B4
CLOSE FILES
NOTE 1
B5
EXIT
TO: $$SPEJ END-OF-JOB TRANSIENT
C3
ERROR
YES
NO
CFCP
D3
PUT OUTPUT
D4
LINK TO HALT/SYSLOG
NOTE 1
D5
HALT
TO: NENTRY HALT/SYSLOG
E3
END OF XTEND
YES
NO
F3
ERROR
YES
NO
H1
NOTE 1: SEE IBM SYSTEMS/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502

Chart FE. Simple Copy Files Phase

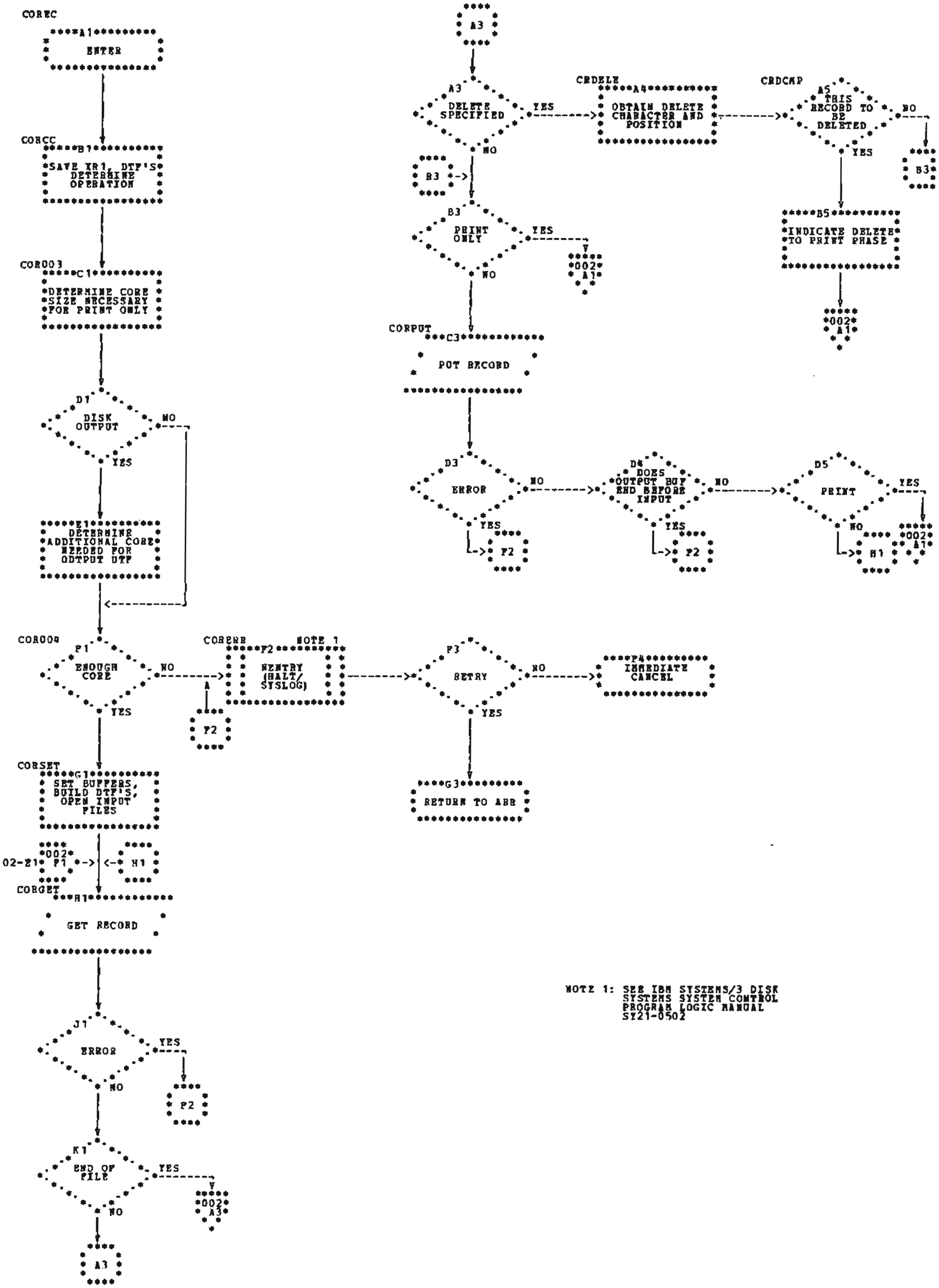
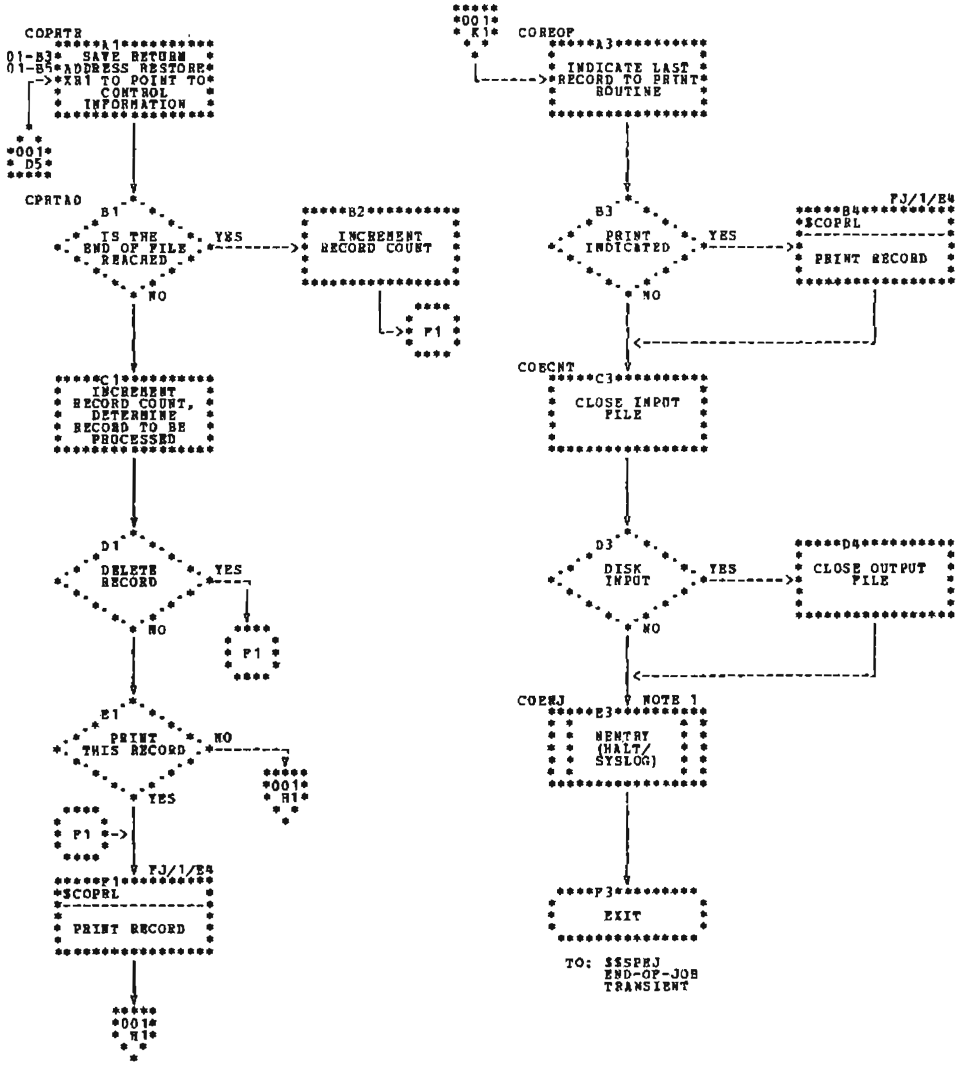

CORBC
A1
ENTER
CORCC
B1
SAVE XR1, DTF'S DETERMINE OPERATION
COR003
C1
DETERMINE CORE SIZE NECESSARY FOR PRINT ONLY
D1
DISK OUTPUT
NO
YES
E1
DETERMINE ADDITIONAL CORE NEEDED FOR OUTPUT DTF
COR004
F1
ENOUGH CORE
NO
YES
CORERR
F2
NOTE 1
NENTRY (HALT/ SYSLOG)
F2
CORSET
G1
SET BUFFERS, BUILD DTF'S, OPEN INPUT FILES
002 F1
02-E1
H1
CORGET
H1
GET RECORD
J1
ERROR
YES
NO
F2
K1
END OF FILE
YES
NO
002 A3
A3
A3
A3
DELETE SPECIFIED
YES
NO
CRDELE
A4
OBTAIN DELETE CHARACTER AND POSITION
CRDCMP
A5
THIS RECORD TO BE DELETED
NO
YES
B3
B3
B3
PRINT ONLY
YES
NO
002 A1
B5
INDICATE DELETE TO PRINT PHASE
002 A1
CORPUT
C3
PUT RECORD
D3
ERROR
NO
YES
F2
D4
DOES OUTPUT BUF END BEFORE INPUT
NO
YES
F2
D5
PRINT
YES
NO
H1
002 A1
F3
RETRY
NO
YES
F4
IMMEDIATE CANCEL
G3
RETURN TO ABR
NOTE 1: SEE IBM SYSTEMS/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502

Chart FF (Part 1 of 2). Copy Non-Indexed Files Phase



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502

Chart FF (Part 2 of 2). Copy Non-Indexed Files Phase

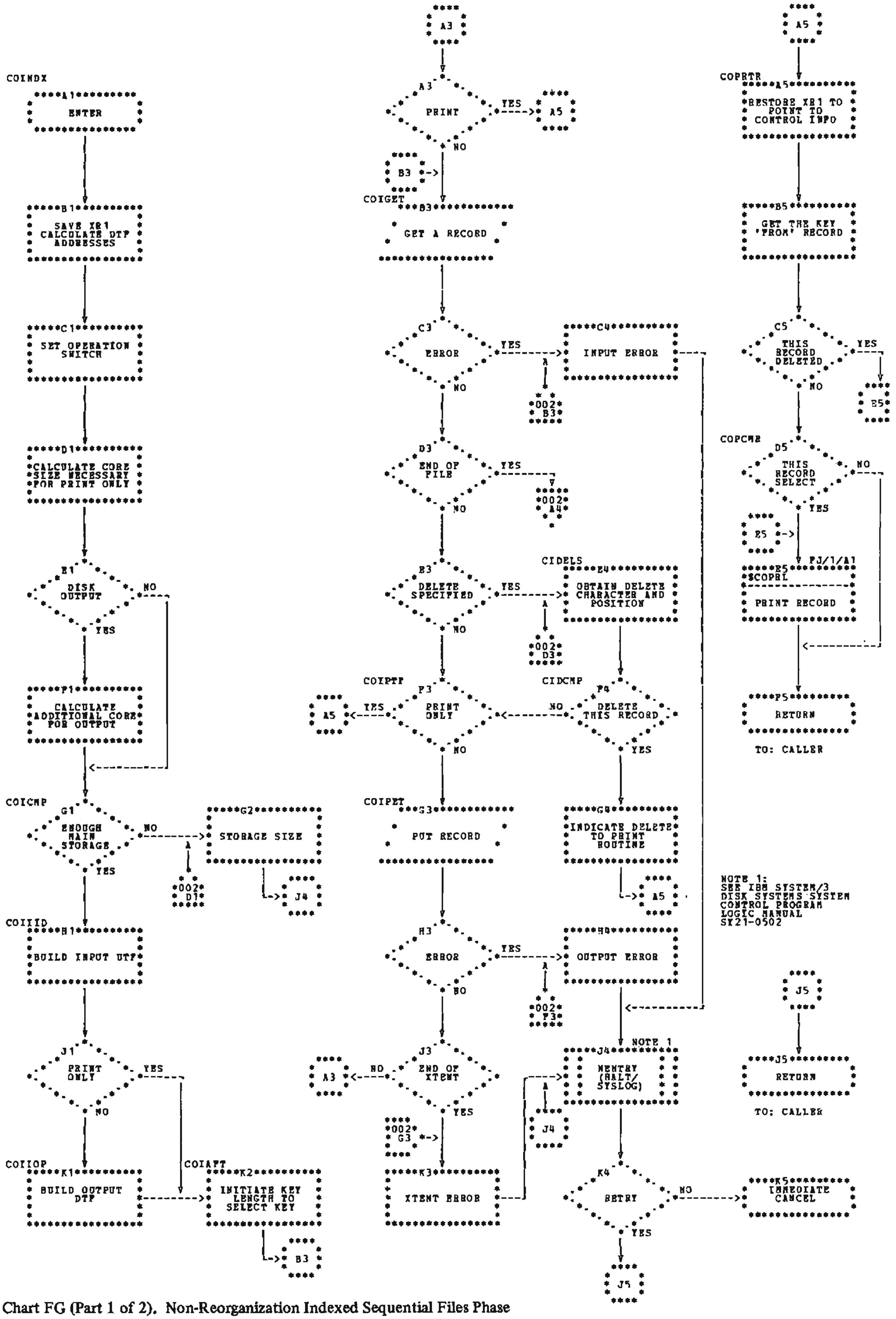


Chart FG (Part 1 of 2). Non-Reorganization Indexed Sequential Files Phase

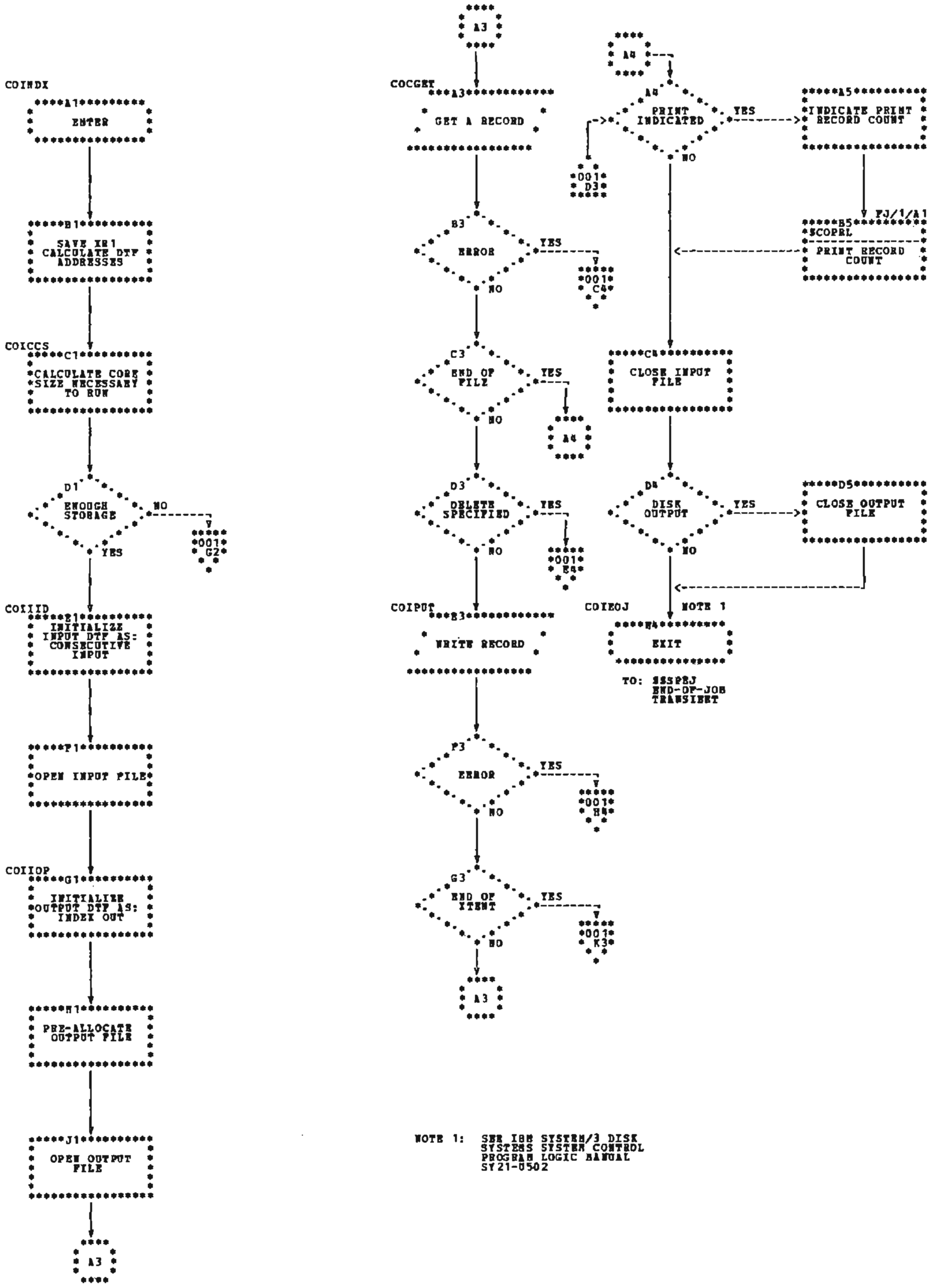


Chart FG (Part 2 of 2). Non-Reorganization Indexed Sequential Files Phase

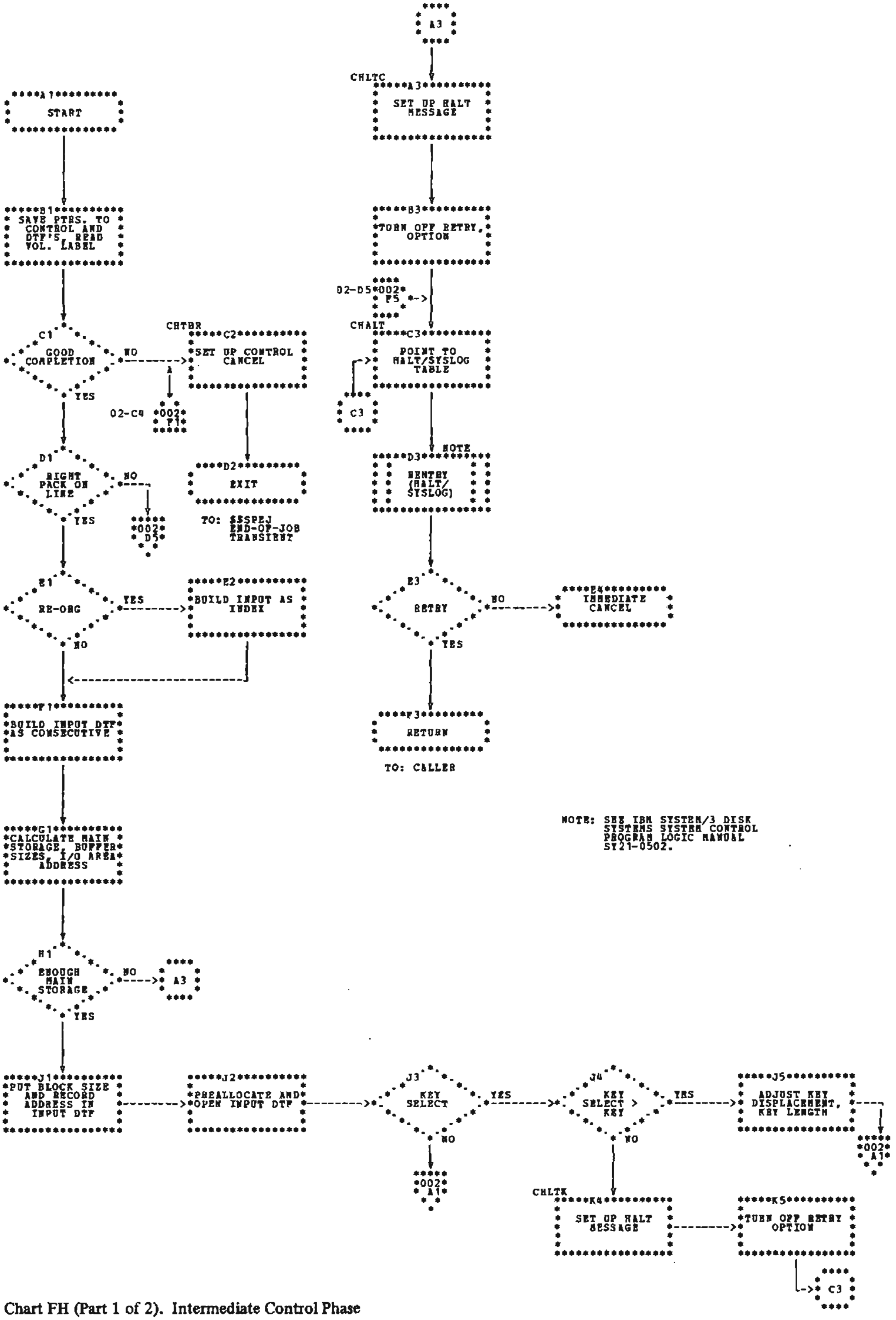


Chart FH (Part 1 of 2). Intermediate Control Phase

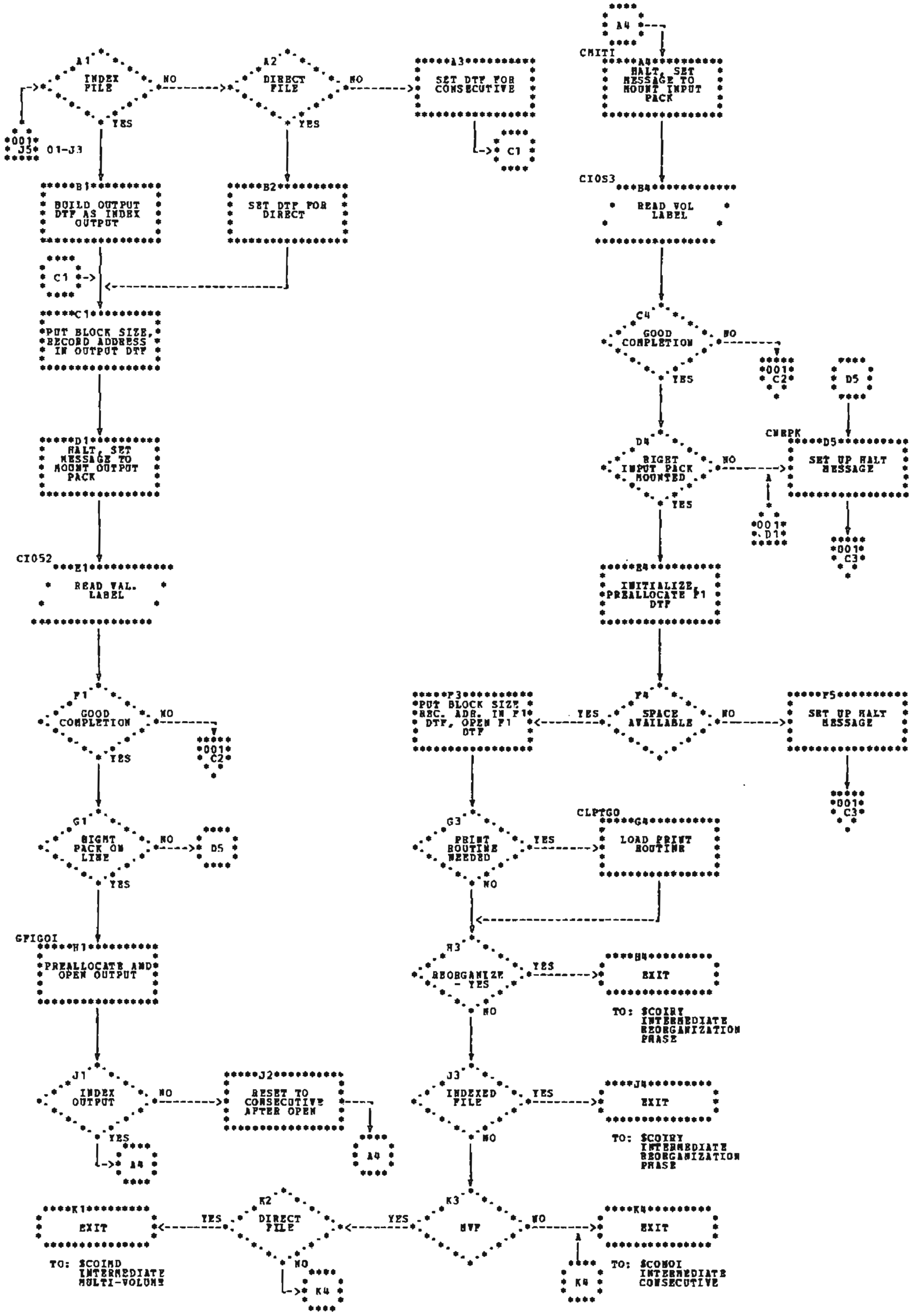

A1 INDEX FILE
NO
YES
001 J5 01-J3
A2 DIRECT FILE
NO
YES
A3 SET DTF FOR CONSECUTIVE
C1
A4
CMITI
A4 HALT, SET MESSAGE TO MOUNT INPUT PACK
B1 BUILD OUTPUT DTF AS INDEX OUTPUT
B2 SET DTF FOR DIRECT
CIOS3
B4 READ VOL LABEL
C1
C1 PUT BLOCK SIZE, RECORD ADDRESS IN OUTPUT DTF
C4 GOOD COMPLETION
NO
001 C2
YES
D5
D1 HALT, SET MESSAGE TO MOUNT OUTPUT PACK
D4 RIGHT INPUT PACK MOUNTED
NO
YES
001 D1
CMRPK
D5 SET UP HALT MESSAGE
001 C3
CIO52
E1 READ VAL. LABEL
E4 INITIALIZE, PREALLOCATE F1 DTF
F1 GOOD COMPLETION
NO
001 C2
YES
F3 PUT BLOCK SIZE REC. ADR. IN F1 DTF, OPEN F1 DTF
YES
F4 SPACE AVAILABLE
NO
F5 SET UP HALT MESSAGE
001 C3
G1 RIGHT PACK ON LINE
NO
D5
YES
G3 PRINT ROUTINE NEEDED
YES
CLPTGO
G4 LOAD PRINT ROUTINE
NO
GFIGOI
H1 PREALLOCATE AND OPEN OUTPUT
H3 REORGANIZE - YES
YES
H4 EXIT
TO: $COIRT INTERMEDIATE REORGANIZATION PHASE
NO
J1 INDEX OUTPUT
NO
J2 RESET TO CONSECUTIVE AFTER OPEN
A4
YES
A4
J3 INDEXED FILE
YES
J4 EXIT
TO: $COIRT INTERMEDIATE REORGANIZATION PHASE
NO
K1 EXIT
TO: $COIMD INTERMEDIATE MULTI-VOLUME
YES
K2 DIRECT FILE
NO
K4
YES
K3 MVF
NO
K4 EXIT
K4
TO: $CONOI INTERMEDIATE CONSECUTIVE

Chart FH (Part 2 of 2). Intermediate Control Phase

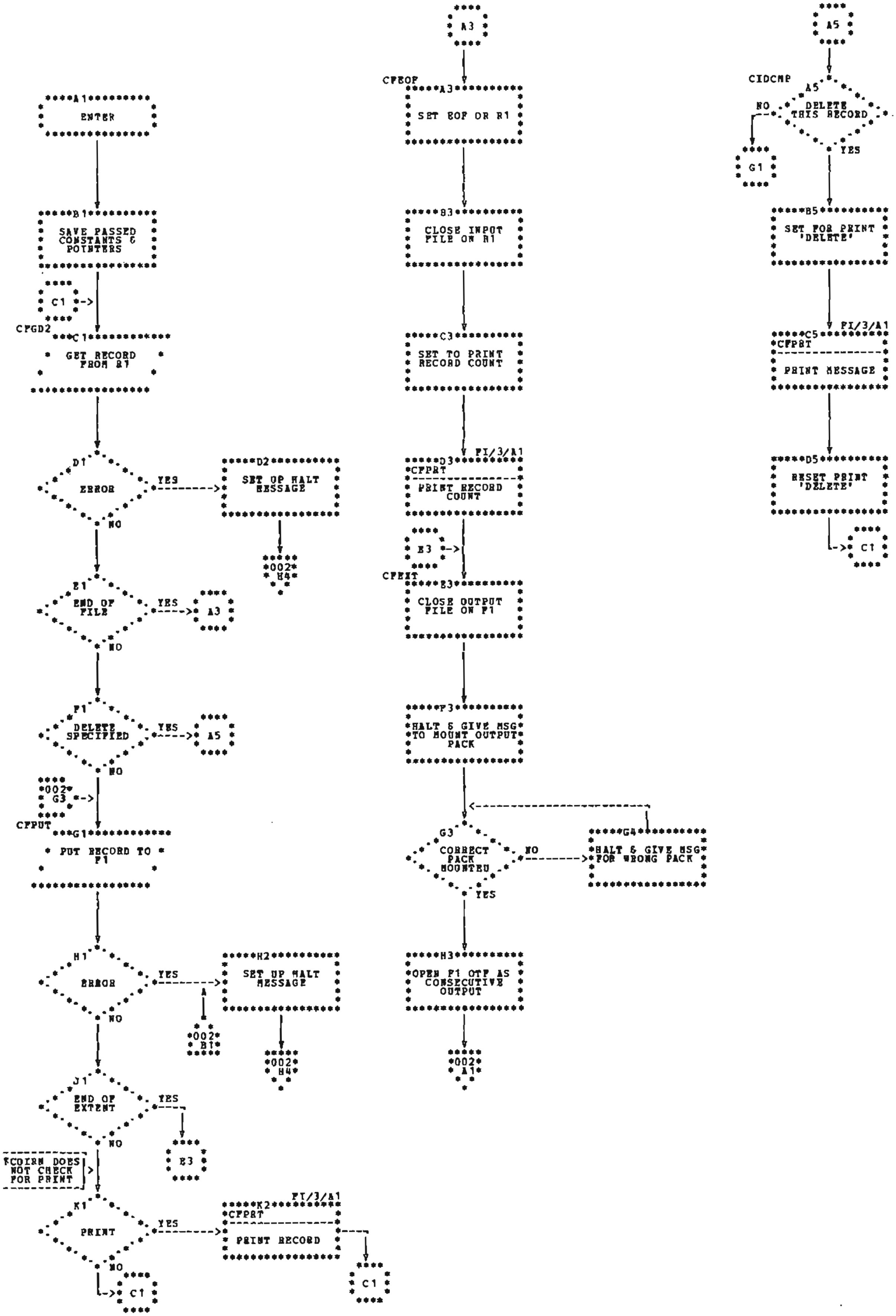


Chart FI (Part 1 of 3). Intermediate Phases

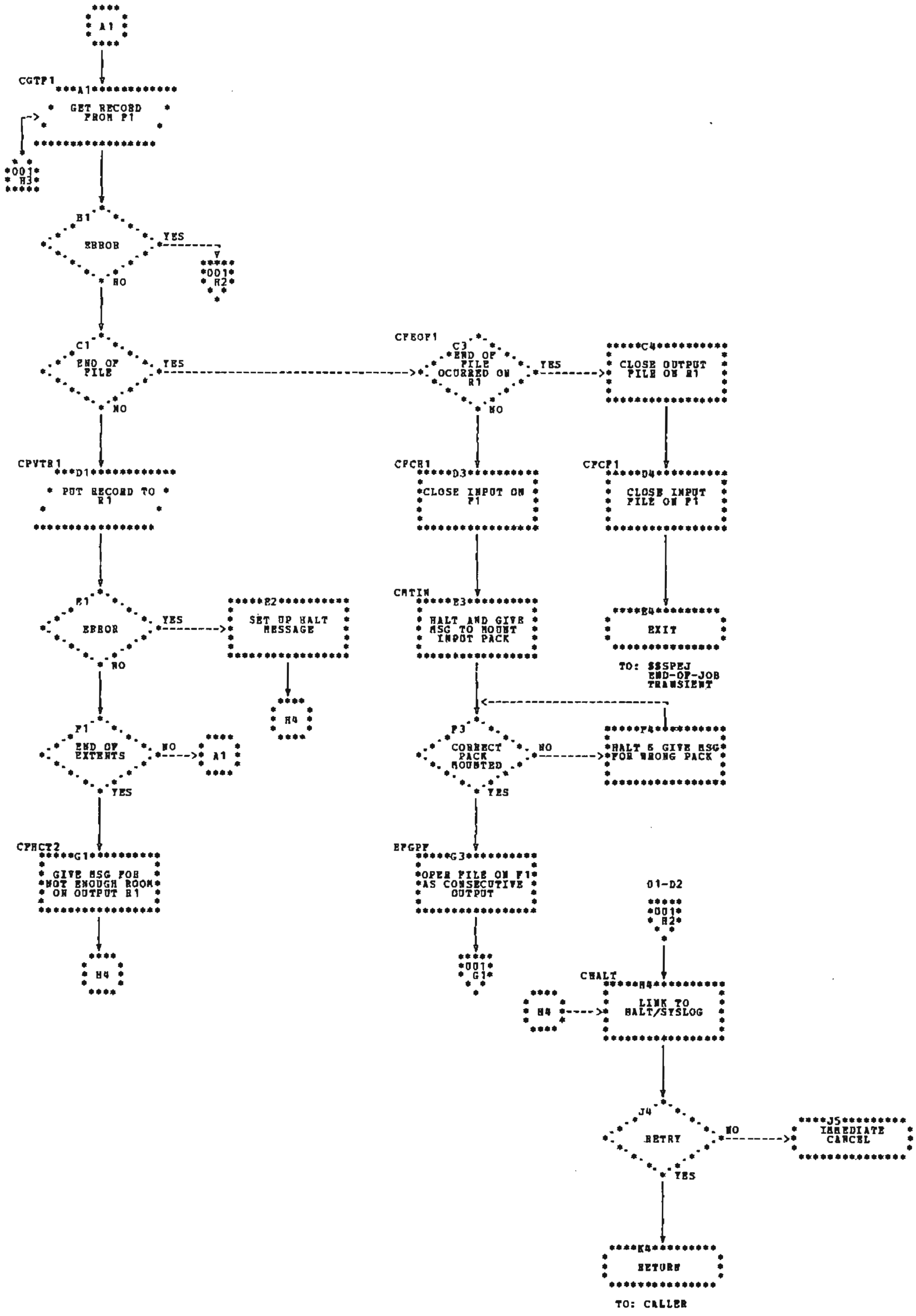


Chart FI (Part 2 of 3). Intermediate Phases

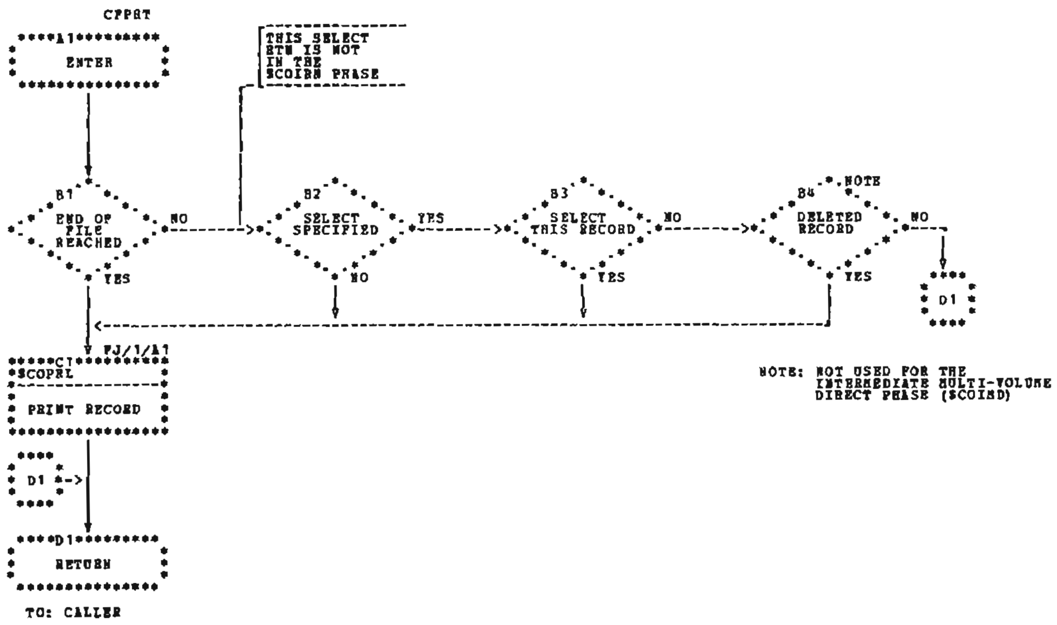
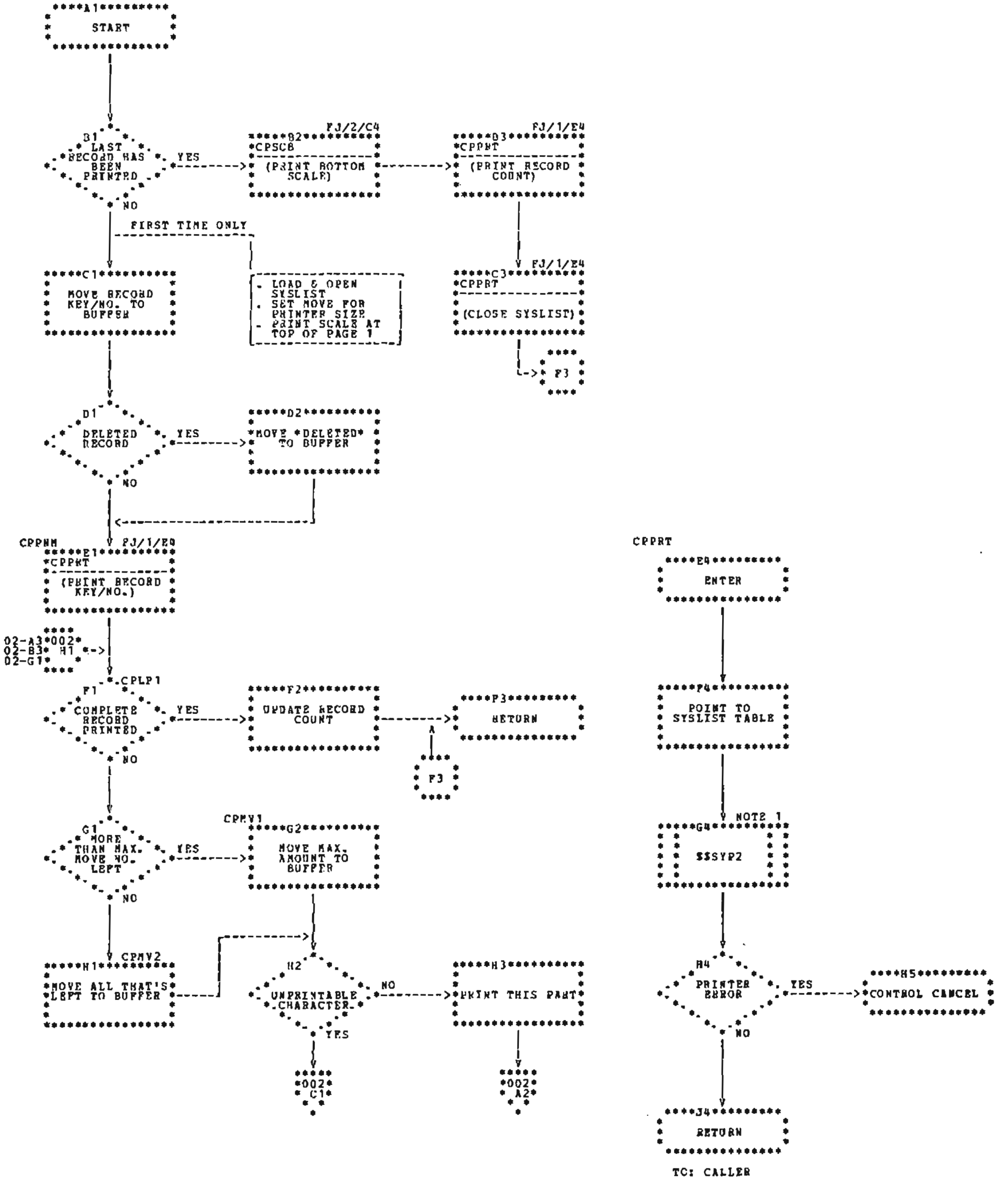


Chart FI (Part 3 of 3). Intermediate Phases



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502

Chart FJ (Part 1 of 2). Print Phase

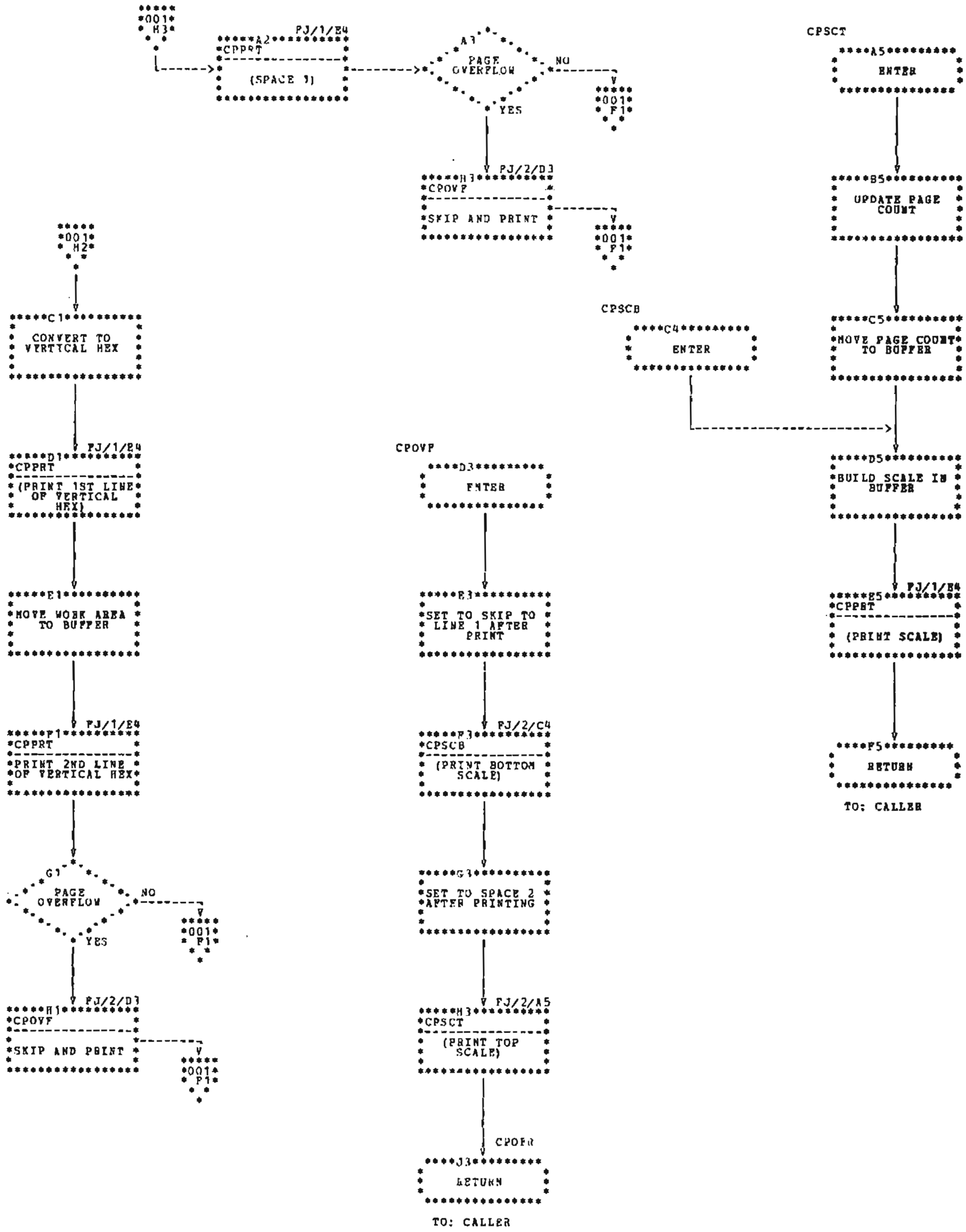
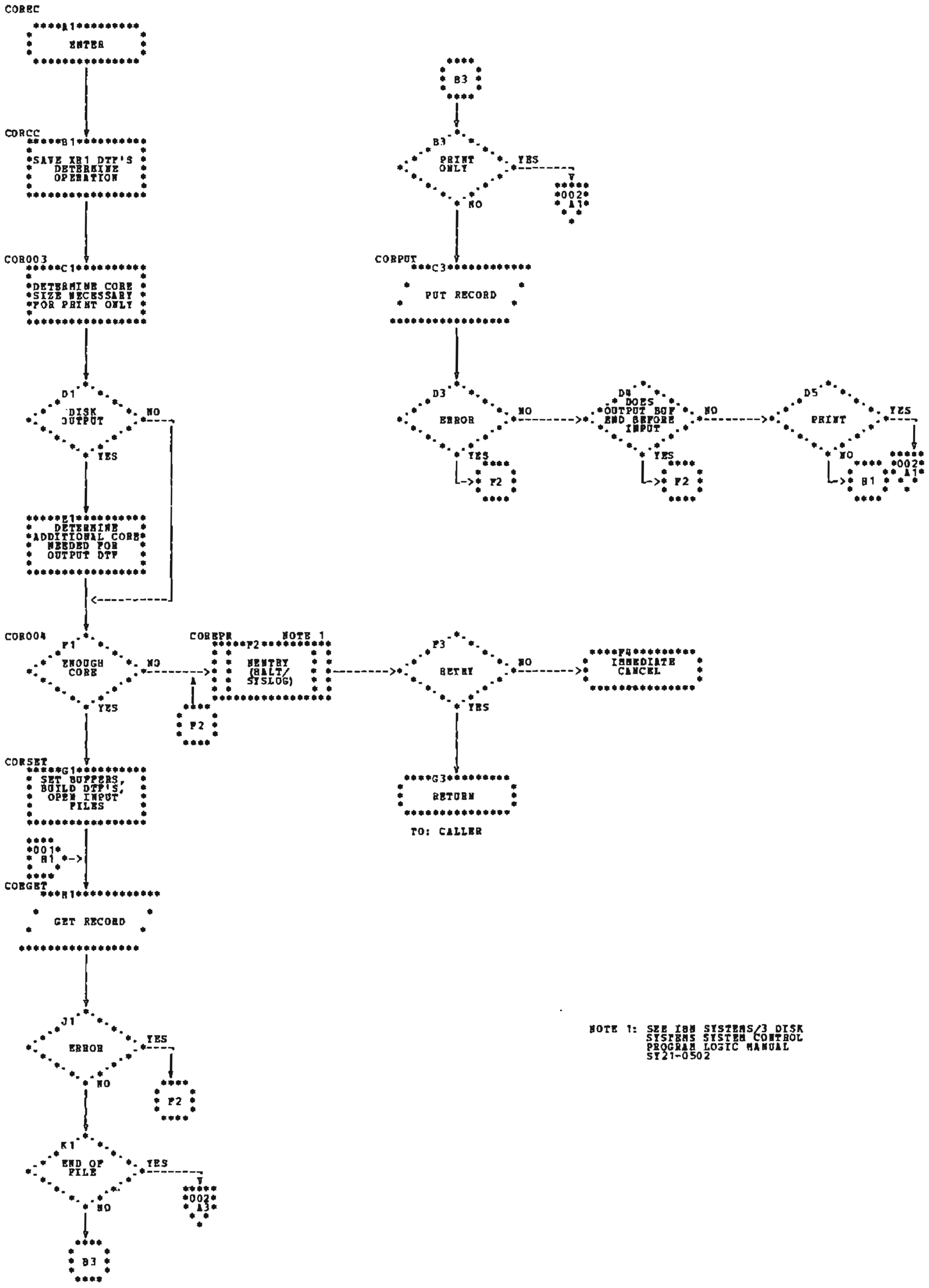
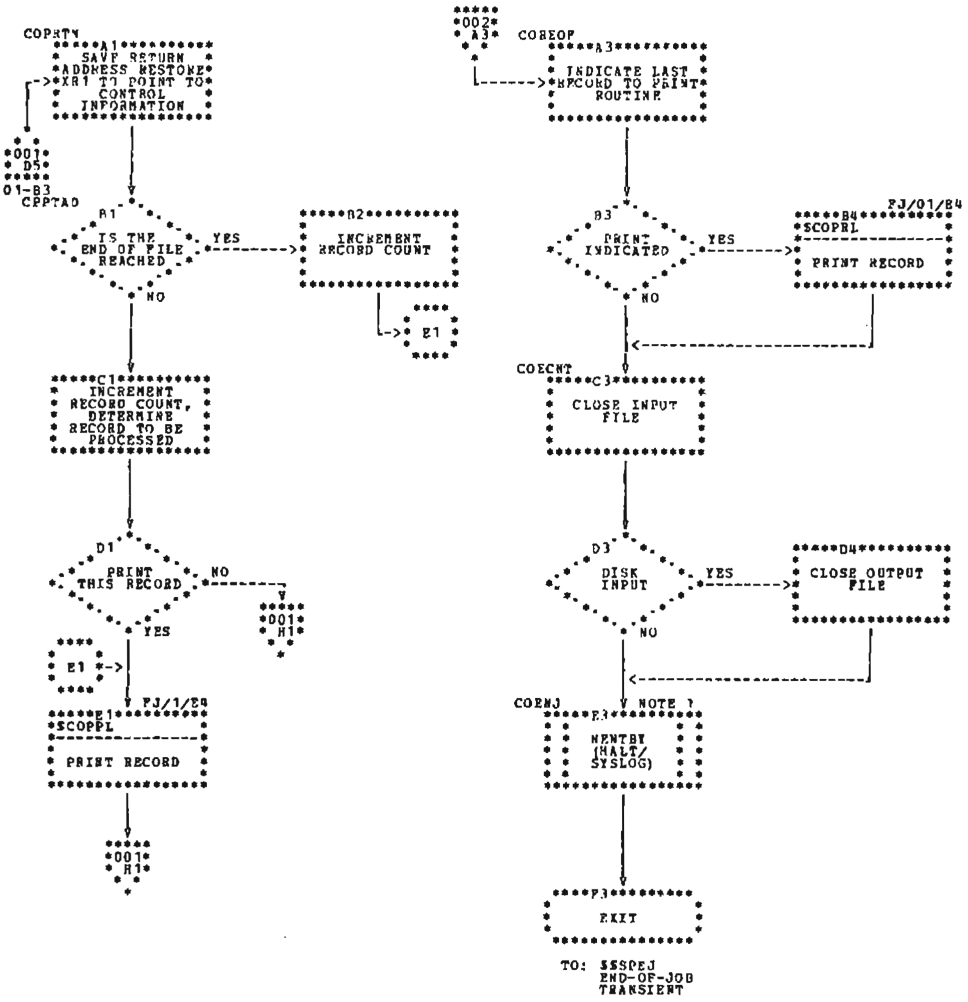


Chart FJ (Part 2 of 2). Print Phase



NOTE 1: SEE IBM SYSTEMS/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502

Chart FK (Part 1 of 2). Copy Multi-Volume Direct Files Phase



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502

Chart FK (Part 2 of 2). Copy Multi-Volume Direct Files Phase

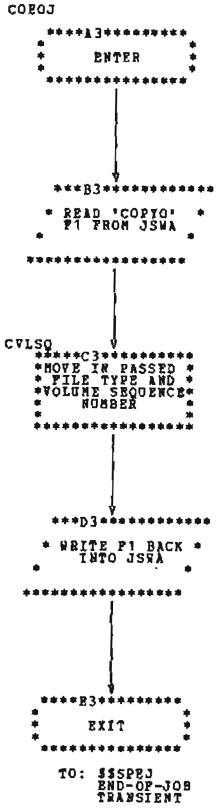

COEOJ
A3
ENTER
B3
READ 'COPYO' F1 FROM JSWA
CVLSQ
C3
MOVE IN PASSED FILE TYPE AND VOLUME SEQUENCE NUMBER
D3
WRITE F1 BACK INTO JSWA
E3
EXIT
TO: $$SPEJ END-OF-JOB TRANSIENT

Chart FL. End of Job Phase

Section 4. Data Area Formats

This section describes the format and contents of the control blocks, tables and data areas used by the IBM System/3 Copy/Dump program. This section is divided into three parts: common data areas, copy pack data areas, and copy file data areas.

COMMON DATA AREAS

The common data areas are used by the common function.

Tables

CSCTL Table

CSCTL is an 83-byte table of constants containing control statement parameter information (Figure 6-6). This table is built by the Control Statement Analysis phase and maintained by the Root phase. It is located at the beginning of the Root phase.

Byte	Label	Contents
0	CTYP	Type of control statement 1. P = Copy pack 2. F = Copy file
1	CIMD	Type of copying 1. I = Intermediate 2. ƀ = Non-intermediate
2	CFM	'FROM' Q code
3	CTO	'TO' Q code
4	CSROG	Reorganization option 1. Y = Yes 2. N = No
5	CSOPT	Output option 1. P = Print 2. D= Disk 3. B = Both
6	CSDEL	D = Delete
7	CSCHR	Chracter to be deleted
8-9	CSPOS	Record position used to check for delete
10	CSTYP	Select option 1. K = Key Select 2. R = Record Select
11	CSLG	Length of key
12-40	CSFKY	Key FROM
41-69	CSTKY	Key TO
These bytes are filled in by the Copy File Initialization phase ($COFIN) after a pre-open.		
70	CTYPE	Type of file 1. I = Index 2. C= Consecutive 3. D= Direct
71	CFMVF	M = Multi-volume file
72	CFEOF	E = End of file
73-74	CRECL	Record length
75-76	COREZ	Storage size of this program level
The following bytes are maintained for the Print phase.		
77	CFDP	This record to be deleted 1. N = No 2. Y = Yes
78	CFPRE	Last record to be printed 1. N = No 2. Y = Yes
79-80	CFPA	Address of first byte of record
81-82	CFENDC	End of storage

Figure 6-6. CSCTL Table of Constants

CSERRB Table

CSERRB is a 7-byte parameter table passed to the Halt/Syslog transient for halts due to errors (Figure 6-7). It is located in the Control Statement Analysis phase.

CSCTLP Table

CSCTLP is a 5-byte table used by the Halt/Syslog transient to log control statements. It is located in the Control Statement Analysis phase. The byte positions and contents for this table are:

Byte	*Contents*
0-1	X'0000' – Indicates output only; log control card
2	X'60' – Indicates 96-byte length
3-4	Address of the I/O area

CSSYNT Table

CSSYNT is an 8-byte table of addresses passed to the Keyword Syntax Scan routine by the Control Statement Analysis phase. It contains the address of the area to be scanned, the output area and the keyword parameter table (Figure 6-8). It is located in the Control Statement Analysis phase.

Byte	Label	Contents
0	CSERRB	B'11010110' – Indicates halt and system print
1	None	C'C' – Indicates Disk Copy/Dump program
2-3	CSHALT	Disk System – Characters to be used for halt display lights Model 6 – Code to be converted for halt display lights
4	CSOPTN	Operator options X'01' = Immediate cancel X'02' = Control cancel X'04' = Retry X'08' = Proceed
5-6	CSSUBM	C'ƀƀ' – Indicates submessage area

Figure 6-7. CSERRB Halt Table

Byte	Label	Contents
0-1	CSIOBG	Address of where to start scan
2-3	CSIOED	Address of where to end scan (CSCARD+95)
4-5	None	Address of scan routine output area (CSSYOT)
6-7	CSTBLS	Address of keyword parameter table

Figure 6-8. CSSYNT Address Table

CSPKTL Table

CSPKTL is a 10-byte keyword parameter table containing the parameters from the COPYPACK and SELECT control statements. If a COPYPACK or SELECT control statement is read, the address of CSPKTL is placed in the table CSSYNT. CSPKTL is located in the Control Statement Analysis phase. The byte positions and contents for this table are:

Byte	*Contents*
0-3	X'14'
4	'FROM'
5-6	X'22'
7	'TO'
8-9	X'FFFF'

CSFILT Table

CSFILT is a 27-byte keyword parameter table containing the parameters from the COPYFILE control statement. If a COPYFILE control statement is read, the address of CSFILT is placed in CSSYNT. CSFILT is located in the Control Statement Analysis phase. The byte positions and contents for this table are:

Byte	*Contents*
0-5	X'16'
6	'OUTPUT'
7-11	X'25'
12	'REORG'
13-18	X'36'
19	'DELETE'
20-23	X'44'
24-25	'WORK'
26	X'FFFF'

CSYSIN Table

CSYSIN is a 7-byte table used to read control statements from SYSIN. It is located in the Control Statement Analysis phase. The byte positions and contents for this table are:

Byte	*Contents*
0	X'09' – Operation code
1-2	Address of I/O area (CSCARD)
3-4	Address of I/O area (CSCARD)
5-6	Address of work area (CSYSWK)

Buffers

CSSYOT Output Buffer

CSSYOT is a 96-byte output buffer used by the Keyword Syntax Scan routine to pass a modified parameter list to the Control Statement Analysis phase.

CSCARD Input Buffer

CSCARD is a 96-byte input buffer containing the control statement images. It is located at the end of the Control Statement Analysis phase.

COPY PACK DATA AREAS

The following data areas are used by the copy pack function.

Control Blocks

CDPIOB Control Block

This control block is used by the Main Data Transfer phase. The Main Data Transfer phase uses CDPIOB to read track 0 and 1 from the input pack and write it on the output pack, and to read data beginning at cylinder 4 from the input pack and write it on the output pack.

CDIIOB Control Block

CDIIOB is used by the Intermediate phase to read the input pack located on the removable disk on drive on (R1).

CDOIOB Control Block

CDOIOB is used by the Intermediate phase to write the output pack located on the removable disk on drive one (R1).

CDFIOB Control Block

CDFIOB Control Block is used by the Intermediate phase to read in the volume label and to read/write on fixed disk on drive one (F1).

Tables

CDCTL Table

CDCTL is a 6-byte table located in the Initialization phase. See Figure 6-9 for its contents.

Byte	Label	Contents
0-1	CDPISC	Address of constant area built by the Keyword Syntax Scan routine
2	CDPIIS	Input disk size
3	CDPIOS	Output disk size
4-5	CDPIOA	Size of I/O buffer CDPMBF

Figure 6-9. CDCTL Table

COPHLT Table

COPHLT is a 7-byte halt table passed to the Halt/Syslog transient from the Intermediate phase (Figure 6-10).

Byte	Label	Contents
0	COPHLT	B'11010110' – Indicates halt and system print
1	None	C'C' – Indicates Disk Copy/Dump program
2-3	COPHTX	Disk System – Characters to be used for halt display lights Model 6 – Code to be converted for halt display lights
4	COPHTO	Operator options X'01' = Immediate cancel X'02' = Control cancel X'04' = Retry X'08' = Proceed
5-6	COPHTM	C'ƀƀ' – Indicates submessage area

Figure 6-10. COPHLT Halt Table

CERRTB Table

CERRTB is a 7-byte halt table passed to the Halt/Syslog transient from the Initialization phase (Figure 6-11).

Byte	Label	Contents
0	CERRTB	B'11010110' – Indicates halt and system print
1	None	C'C' – Indicates Disk Copy/Dump program
2-3	None	Disk System – Characters to be used for halt display lights Model 6 – Code to be converted for halt display lights
4	CDOPTN	Operator options X'01' = Immediate cancel X'02' = Control cancel X'04' = Retry X'08' = Proceed
5-6	CERM	C'ƀƀ' – Indicates submessage area

Figure 6-11. CERRTB Halt Table

Buffers and Work Areas

CDPIBF I/O Buffer

CDPIBF is a 256-byte disk input/output buffer located in the Initialization phase.

CDPMBF I/O Buffer

CDPMBF is a disk input/output buffer used by the Main Data Transfer phase. This area has a default size of 3,072 bytes for half track reads unless otherwise specified. CDPMBF is located in the Main Data Transfer phase.

CDIBUF I/O Buffer

CDIBUF is a disk input/output buffer used by the Intermediate phase.

COPY FILE DATA AREAS

These data areas are used by the copy file function.

Control Blocks

CFIN Control Block

CFIN located in the Initialization phase, is the DTF control block for the input file. The file attributes and the record length are filled in by data management.

CFOUT Control Block

CFOUT is the DTF control block for the output file. It is located in the Initialization phase.

CFIOB Control Block

CFIOB is the IOB used to read the volume label. It is used by the Intermediate Control, Intermediate Reorganization, Intermediate Non-Reorganization, and Intermediate Consecutive phases.

CF1DTF Control Block

CF1DTF is the DTF control block for the fixed disk on drive one (F1). It is located in the Intermediate Control phase. For the format and contents of a DTF, see *IBM System/3 Disk Systems Data Management Input/Output Supervisor Logic Manual,* SY21-0512.

Tables

CFERR Table

CFERR is a 7-byte halt table passed to the Halt/Syslog transient from the Simple Copy Files and Initialization phases (Figure 6-12).

Byte	Label	Contents
0	CFERR	B'11010110' — Indicates halt and system print
1	None	C'C' — Indicates Disk Copy/Dump program
2-3	CFERH	Disk System — Characters to be used for halt display lights Model 6 — Code to be converted for halt display lights
4	CFERO	Operator options X'01' = Immediate cancel X'02' = Control cancel X'04' = Retry X'08' = Proceed
5-6	CFERM	X'b̸b̸' — Indicates submessage area

Figure 6-12. CFERR Table

CORHLT Table

CORHLT is a 7-byte halt table passed to the Halt/Syslog transient from the Copy Non-Indexed Files phase (Figure 6-13).

Byte	Label	Contents
0	CORHLT	B'11010110' — Indicates halt and system print
1	None	C'C' — Indicates Disk Copy/Dump program
2-3	None	Disk System — Characters to be used for halt display lights Model 6 — Code to be converted for halt display lights
4	CORHTA	Operator options X'01' = Immediate cancel X'02' = Control cancel X'04' = Retry X'08' = Proceed
5-6	CORHTM	C'b̸b̸' — Indicates submessage area

Figure 6-13. CORHLT Table

CRIHLT Table

CRIHLT is a 7-byte halt table passed to the Halt/Syslog transient from the Non-Reorganization Indexed Sequential Files and Reorganization Indexed Sequential Files phases (Figure 6-14).

Byte	Label	Contents
0	CRIHLT	B'11010110' – Indicates halt and system print
1	None	C'C' – Indicates Disk Copy/Dump program
2-3	None	Disk System – Characters to be used for halt display lights Model 6 – Code to be converted for halt display lights
4	None	Operator options X'01' = Immediate cancel X'02' = Control cancel X'04' = Retry X'08' = Proceed
5-6	CRIHTM	C'ƀƀ' – Indicates submessage area

Figure 6-14. CRIHLT Table

CHTTB Table

CHTTB is a 7-byte halt table passed to the Halt/Syslog transient from the Intermediate Control, Intermediate Reorganization, Intermediate Non-Reorganization, and Intermediate Consecutive phases (Figures 6-15).

Byte	Label	Contents
0	CHTTB	B'11010110' – Indicates halt and system print
1	None	C'C' – Indicates Disk Copy/Dump program
2-3	CHLT	Disk System – Characters to be used for halt display lights Model 6 – Code to be converted for halt display lights
4	None	Operator options X'01' = Immediate cancel X'02' = Control cancel X'04' = Retry X'08' = Proceed
5-6	CSUB	C'ƀƀ' – Indicates submessage area

Figure 6-15. CHTTB Table

CNOTAB Table

CNOTAB, a 5-byte table passed to the Halt/Syslog transient, is used to log error messages on the logging device. It is located in the Copy Non-Indexed Files phase.

CPLIST Table

CPLIST is a 7-byte parameter table passed to SYSLIST by the Print phase (Figure 6-16). It is located in the Print phase.

Byte	Label	Contents
0	CPOP	X'10' = Operation code
1	CPSKBP	X'01' = Skip before printing
2	CPSKBP	X'00' = Space before printing
3	CPSKAP	X'00' = Skip after printing
4	CPSKAP	X'01' = Space after printing
5-6	CBFAD	Address of logical record

Figure 6-16. CPLIST Table

Buffers and Work Areas

COIEND Buffer

COIEND contains one logical record. The size of the buffer is variable and is determined by the record length. It is located in the Non-Reorganization Indexed Sequential Files and the Reorganization Indexed Sequential Files phases.

CPBUF Buffer

CPBUF is a print buffer passed to SYSLIST by the Print phase. This 106-byte buffer contains a logical record.

CFVLBF Buffer

CFVLBF is a 256-byte area used to read in the volume label. It is used by the Intermediate Control, Intermediate Reorganization, Intermediate Non-Reorganization, and Intermediate Consecutive phases.

CONEND Record Area

CONEND contains the I/O area. The size of the area is determined by main storage size. CONEND is located in the Simple Copy Files phase.

CPWRK Conversion Area

CPWRK is a work area used by the Print phase to convert unprintable characters to vertical hexadecimal.

PART VII
LIBRARY MAINTENANCE

The Library Maintenance program processes object and source libraries in any one of these ways:

- Allocates space for the library.
- Deletes library entries.
- Copies the library or entries within the libraries.
- Renames the library entries.

These functions are provided by four control statements:

1. // ALLOCATE
2. // DELETE
3. // COPY
4. // RENAME

The Library Maintenance program scans the control statements and passes control to a phase that performs the desired task. When all of the requested functions have been completed, control is passed to the scheduler.

SYSTEM REQUIREMENTS

The Library Maintenance program uses the following minimum system configuration:

Disk System

- IBM 5410 Processing Unit, Model A12
- IBM 5203 Printer, Model A1
- IBM 5424 Multi-Function Card Unit, Model A
- IBM 5444 Disk Storage Drive, Model 2

Model 6

- IBM 5406 Processing Unit, Model B2 (Includes the keyboard with eight command keys)
- IBM 5213 Printer, Model 1
- IBM 5444 Disk Storage Drive, Model 2

The following optional features are supported:

- IBM 5496 Data Recorder, Model 1 with System/3 Model 6 On-line Feature (Used as the Sysin device)
- IBM 2265 Display Station, Model 2 (Cathode Ray Tube; used as Syslog device with IBM 5406 Processing Unit, Models B3 and B4 only)

When the scheduler passes control to the Library Maintenance program, the Mainline Librarian phase ($MAINT) receives control. This phase processes five types of library maintenance control statements:

1. // ALLOCATE
2. // RENAME
3. // DELETE
4. // COPY
5. // END

The Mainline Librarian phase reads control statements one at a time into an input buffer (LEBUFR). The buffer stores the statement image; from this buffer a control area (CAREA) and a parameter list of valid keywords (LELUXP) are built. The Mainline Librarian phase then calls the Keyword Syntax Scan routine ($$RDS1) which analyzes the statement image and returns a data area of valid parameters (LERETO) to the Mainline Librarian. The Mainline Librarian uses this data area to call the correct control statement processor (Figure 7-1).

// ALLOCATE CONTROL STATEMENT

If the Mainline Librarian has read a // ALLOCATE control statement, control is passed to the Allocate Syntax Scan phase ($MAALS). This phase uses the control area (CAREA), volume label, and the Allocation Initiator phase ($$STAI) to check for the required parameters on the // ALLOCATE statement and to determine if the specified functions can be performed. If the specified functions can be performed, the Allocate Syntax Scan phase ($MAALS) passes control to the Allocate Mainline phase ($MAAMN) or the Reallocate Mainline phase ($MAAMW).

The Allocate Mainline ($MAAMN) passes control to one of three phases:

- Allocate Object Library ($MAAOL) builds the object section of the system directory and allocates the scheduler work area (SWA) and roll-in/roll-out area.
- Allocate Source Library ($MAASL) builds the source section of the system directory and chains the source directory sectors and library sectors.
- Track Allocate/Deallocate ($MATAD) updates the volume label.

The Reallocate Mainline phase ($MAAMW), using the Find phase ($MAFND) to locate the specified entry in either the object or source library passes control to one of five phases:

- Copy Source Library ($MAACS) copies the source library onto the work pack.
- Copy Object Library ($MAACO) copies the object library onto the work pack.
- Copy Work Pack to Object Library ($MARCO) copies the object library entries from the work pack to a reallocated object library.
- Copy Work Pack to Source Library ($MARCS) copies the source library entries from work pack to a reallocated source library.
- Allocate Mainline ($MAAMN) deallocates and allocates libraries.

After processing the // ALLOCATE control statement, control is returned to the Mainline Librarian which will read another control statement.

// RENAME CONTROL STATEMENT

If the Mainline Librarian has read a // RENAME control statement, control is passed to the Rename Control Statement Processor ($MAREM). This processor uses the Find phase ($MAFND) to check for duplicate entry names and to update the library directory to reflect the specified new name. Control is then returned to the Mainline Librarian which will read another control statement.

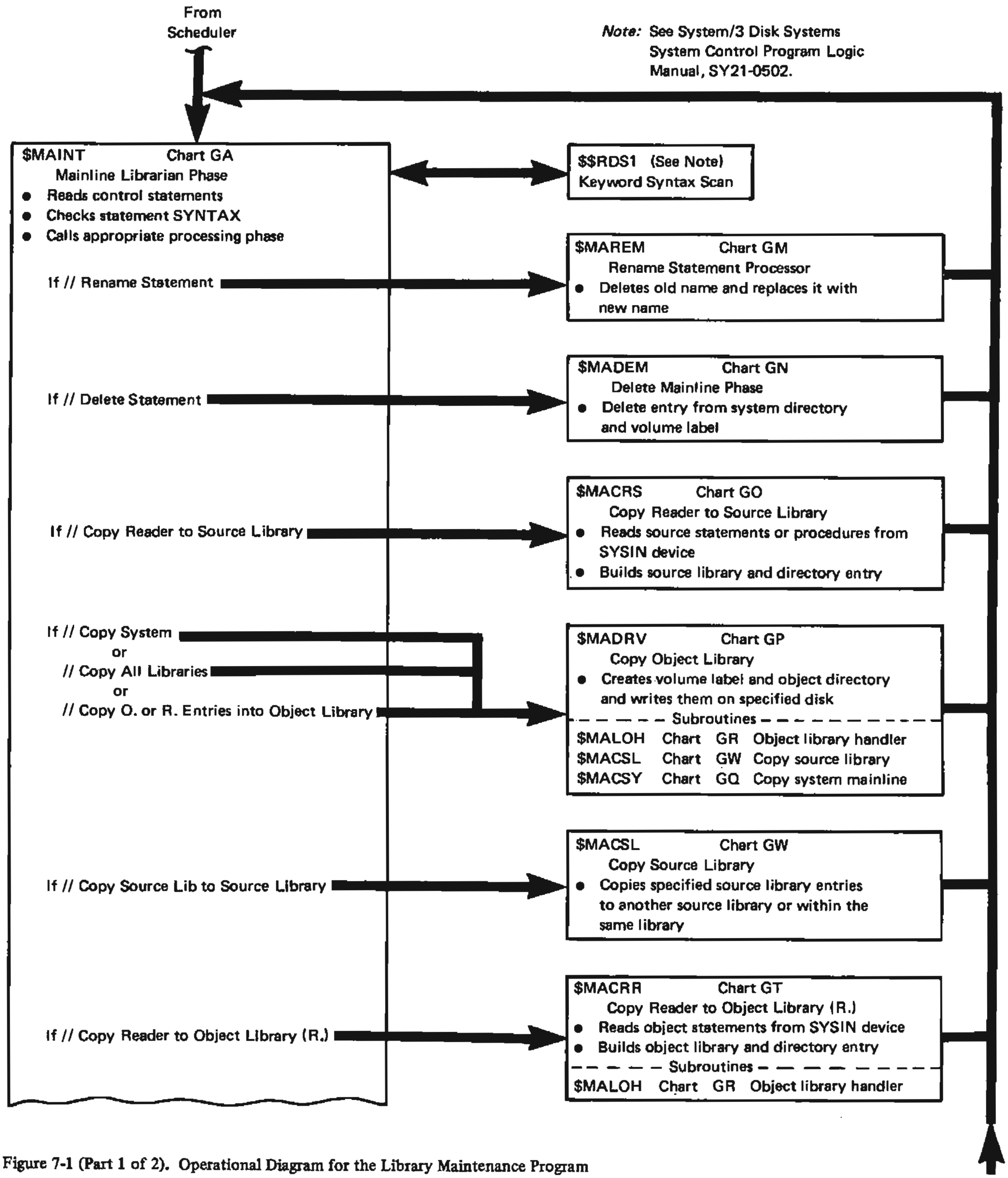

From Scheduler
Note: See System/3 Disk Systems System Control Program Logic Manual, SY21-0502.
$MAINT Chart GA
Mainline Librarian Phase
● Reads control statements
● Checks statement SYNTAX
● Calls appropriate processing phase
If // Rename Statement
If // Delete Statement
If // Copy Reader to Source Library
If // Copy System
or
// Copy All Libraries
or
// Copy O. or R. Entries into Object Library
If // Copy Source Lib to Source Library
If // Copy Reader to Object Library (R.)
$$RDS1 (See Note)
Keyword Syntax Scan
$MAREM Chart GM
Rename Statement Processor
● Deletes old name and replaces it with new name
$MADEM Chart GN
Delete Mainline Phase
● Delete entry from system directory and volume label
$MACRS Chart GO
Copy Reader to Source Library
● Reads source statements or procedures from SYSIN device
● Builds source library and directory entry
$MADRV Chart GP
Copy Object Library
● Creates volume label and object directory and writes them on specified disk
Subroutines
$MALOH Chart GR Object library handler
$MACSL Chart GW Copy source library
$MACSY Chart GQ Copy system mainline
$MACSL Chart GW
Copy Source Library
● Copies specified source library entries to another source library or within the same library
$MACRR Chart GT
Copy Reader to Object Library (R.)
● Reads object statements from SYSIN device
● Builds object library and directory entry
Subroutines
$MALOH Chart GR Object library handler

Figure 7-1 (Part 1 of 2). Operational Diagram for the Library Maintenance Program

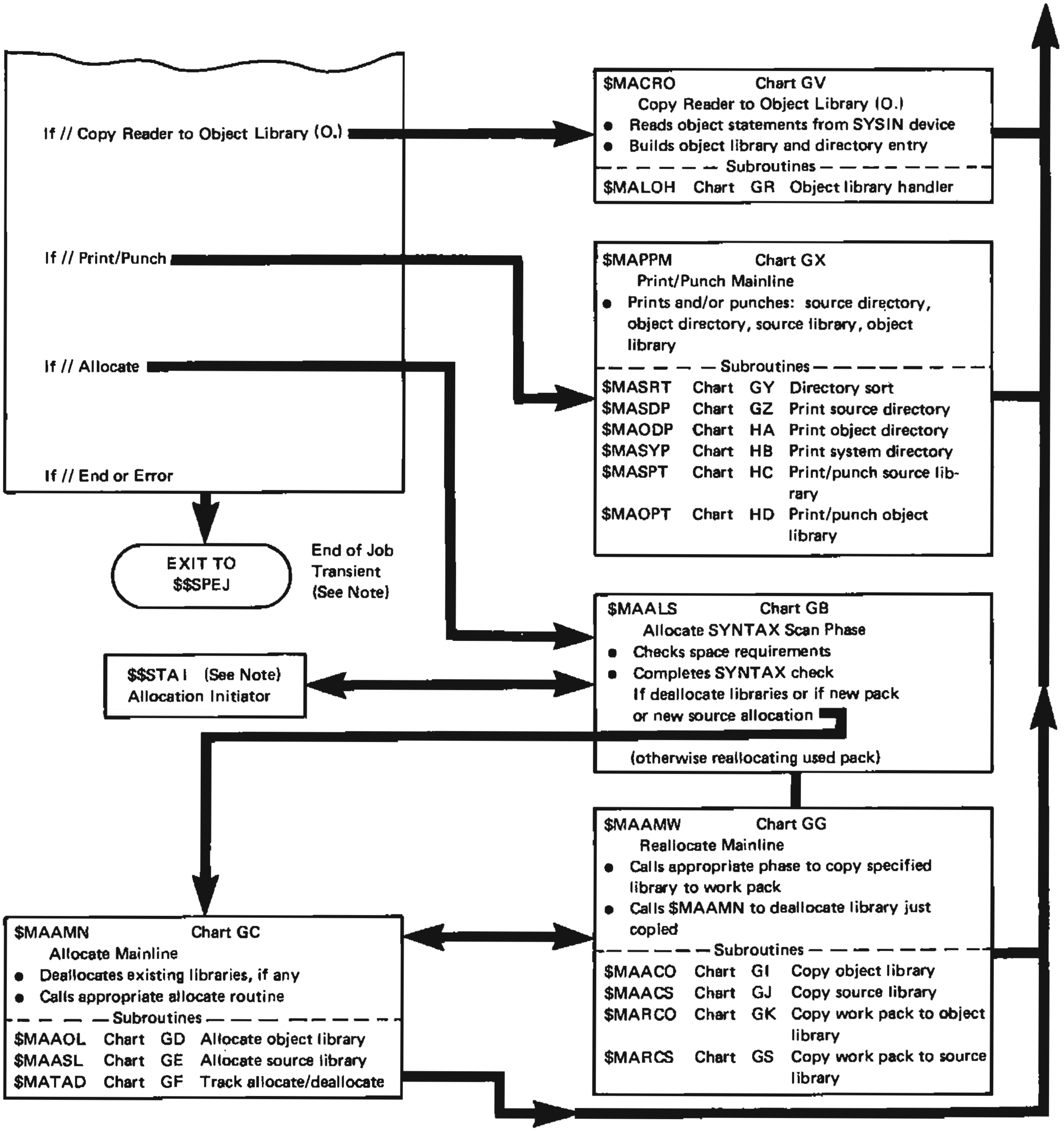

If // Copy Reader to Object Library (O.)
If // Print/Punch
If // Allocate
If // End or Error
EXIT TO $$SPEJ
End of Job Transient (See Note)
$$STA I (See Note)
Allocation Initiator
$MACRO Chart GV
Copy Reader to Object Library (O.)
● Reads object statements from SYSIN device
● Builds object library and directory entry
Subroutines
$MALOH Chart GR Object library handler
$MAPPM Chart GX
Print/Punch Mainline
● Prints and/or punches: source directory, object directory, source library, object library
Subroutines
$MASRT Chart GY Directory sort
$MASDP Chart GZ Print source directory
$MAODP Chart HA Print object directory
$MASYP Chart HB Print system directory
$MASPT Chart HC Print/punch source library
$MAOPT Chart HD Print/punch object library
$MAALS Chart GB
Allocate SYNTAX Scan Phase
● Checks space requirements
● Completes SYNTAX check
If deallocate libraries or if new pack or new source allocation
(otherwise reallocating used pack)
$MAAMW Chart GG
Reallocate Mainline
● Calls appropriate phase to copy specified library to work pack
● Calls $MAAMN to deallocate library just copied
Subroutines
$MAACO Chart GI Copy object library
$MAACS Chart GJ Copy source library
$MARCO Chart GK Copy work pack to object library
$MARCS Chart GS Copy work pack to source library
$MAAMN Chart GC
Allocate Mainline
● Deallocates existing libraries, if any
● Calls appropriate allocate routine
Subroutines
$MAAOL Chart GD Allocate object library
$MAASL Chart GE Allocate source library
$MATAD Chart GF Track allocate/deallocate

Figure 7-1 (Part 2 of 2). Operational Diagram for the Library Maintenance Program

// DELETE CONTROL STATEMENT

If the Mainline Librarian has read a // DELETE control statement, control is passed to Delete Mainline ($MADEM). Delete Mainline uses the Find phase ($MAFND) to delete the specified library entries in either the object, source, or both object and source libraries and to update the system directory. Control is then returned to the Mainline Librarian which will read another control statement.

// COPY CONTROL STATEMENT

The Mainline Librarian also acts as the Copy Mainline and determines which of these eight functions is to be performed:

1. Copy reader to source library.
2. Copy system.
3. Copy reader to object library (R. entries).
4. Copy reader to object library (O. entries).
5. Copy all libraries.
6. Print/punch directory or library.
7. Copy source library to source library.
8. Copy O. entries or R. entries to object library.

Copy Reader to Source Library

Control is passed from the Mainline Librarian to the Copy Reader to Source Library phase ($MACRS). This phase copies the specified program from the reader if (1) space is available and (2) the requested name is unique to the source library. The Source Put phase ($$SYSP) is used to compress the source statement and build the entry. Control is then returned to the Mainline Librarian which will read another control statement. If the requested name is not unique, the duplicate named entry can be deleted following a halt as long as a 'T' does not replace a 'P' type entry.

Copy System

Control is passed from the Mainline Librarian to Copy Object Library – Disk to Disk phase ($MADRV). Control is then passed from the Mainline Librarian to the Copy System Mainline phase ($MACSY). This phase copies the following three items from one system disk pack to the other specified disk pack:

- Initial Program Load (IPL) Bootstrap program.
- System configuration record.
- Nucleus Initialization program (NIP).

$MADRV then calls the Object Library Handler phase ($MAOLH) which locates available space for the entry. $MAOLH and the Find phase ($MAFND) are used to copy all $$ entries. The Update PTF Log Sector phase ($MALOG) updates PTF log sectors after the copy.

Upon completion, control is returned to the Mainline Librarian which will read another control statement.

Copy Reader to Object Library (R. Entries)

Control is passed from the Mainline Librarian to the Copy Reader to Object Library (R. Entries) phase ($MACRR). This phase uses the Find phase ($MAFND) to locate the specified entry and the Object Library Handler ($MALOH) to locate available space for the entry. The Readjust Extended End of Object Library phase ($MAEXE) determines the number of sectors that must be added to the object library. This phase then:

1. Reads and compresses object records from the SYSIN device.
2. Builds an object library entry from the compressed cards.
3. Builds a directory entry for the object library and updates the system directory.

Control is then returned to the Mainline Librarian which will read another control statement.

Copy Reader to Object Library (O. Entries)

Control is passed from the Mainline Librarian to the Copy Reader to Object Library (O. Entries) ($MACRO). This phase uses the Find phase ($MAFND) to locate the entries and the Object Library Handler ($MAOLH) to locate available space for the entries. The Readjust Extended End of Library phase ($MAEXE) determines the number of sectors that must be added to the object library. The Update PTF Log Sector phase ($MALOG) updates PTF log sectors after the copy. This phase then:

1. Reads and compresses object records from the SYSIN device.
2. Builds the object library.
3. Builds the directory entry for the object library.

Control is then returned to the Mainline Librarian which will read another control statement.

Copy All Libraries

Control is passed from the Mainline Librarian to the Copy Object Library – Disk to Disk phase ($MADRV). This phase uses the Find phase ($MAFND) to locate the specified entries and the Object Library Handler ($MAOLH) to locate available space for the library entries. This phase copies the object library entries to the specified disk. The Readjust Extended End of Library phase ($MAEXE) determines the number of sectors to be added to the object library. The Update PTF Log Sector phase ($MALOG) updates PTF log sectors after the copy. The Copy System Mainline phase ($MACSY) copies the system directories. Control is then passed to the Copy Source Library – Disk to Disk phase ($MACSL), which copies the source library entries to the specified disk. Control is then returned to the Mainline Librarian which will read another control statement.

Print/Punch Directory or Library

Control is passed from the Mainline Librarian to the Print/Punch Mainline ($MAPPM). This phase uses the Find phase ($MAFND) to locate the specified entries and the Directory Sort phase ($MASRT) to sort the directories before they are printed. Each library contains a directory which points to the entries within the library. Section 4, *Data Area Formats,* contains the format of the object and source library directories. Control is then passed from the Print/Punch Mainline depending upon the type of request:

- Directory print request. Control is passed to one of three phases:

  1. Print Source Directory ($MASDP).
  2. Print Object Directory ($MAODP).
  3. Print System Directory ($MASYP).

- Print/Punch library request. Control is passed to one of two phases:

  1. Print/Punch Source Library ($MASPT).
  2. Print/Punch Object Library ($MAOPT).

Control is returned from the Print/Punch Mainline to the Mainline Librarian which will read another control statement.

Copy Source Library to Source Library

Using the Find phase ($MAFND) to locate the specified source library entries, the Copy Source Library – Disk to Disk phase ($MACSL) copies the entries either to another source library (renaming them if requested) or within the same library (renaming them). Control is then passed to the Mainline Librarian which will read another control statement.

Copy O. Entries or R. Entries to Object Libraries

Control is passed from the Mainline Librarian to the Copy Object Library – Disk to Disk phase ($MADRV) using the Find phase ($MAFND) and the Object Library Handler ($MAOLH). This phase copies the specified library entries to the specified disk. Control is then returned to the Mainline Librarian which will read another control statement.

// END CONTROL STATEMENT

The // END control statement causes the Mainline Librarian to pass control to the Transient Resolver phase ($$OXRF) which resolves the address of all $$ modules and then passes control to End of Job transient ($$SPEJ). This transient terminates the Library Maintenance program and blanks out the transient queue. Control is passed to the Scheduler End of Job Deallocation phase ($$TMEJ) which deallocates all devices and work areas used by the Library Maintenance program.

The following section contains a description and flowchart for each phase. The phases are discussed in the order they are mentioned in the *Method of Operation* section.

Mainline Librarian Phase ($MAINT)

Entry Point: $MAINT

Storage Map: Figure 7-2

Chart: GA

Functions:

- Reads control statements into input buffer LEBUFR.
- Builds a control area (CAREA) from the control statement parameters.
- Loads the specified system directories.
- Passes control to the Keyword Syntax Scan routine ($$RDS1) to check the functional correctness of the control statements.
- Passes control to the appropriate control statement processing phase.
- Reads the // END control statment and indicates if the Transient Resolver phase ($$OXRF) must be called to resolve addresses for $$ modules.

Input: Five control statements:

1. // ALLOCATE
2. // RENAME
3. // DELETE
4. // COPY
5. // END

Output: Partially built control area (CAREA)

Exit:

- Normal – End of job transient ($$SPEJ) when the // END control statement is read
- Error – Halt/Syslog transient

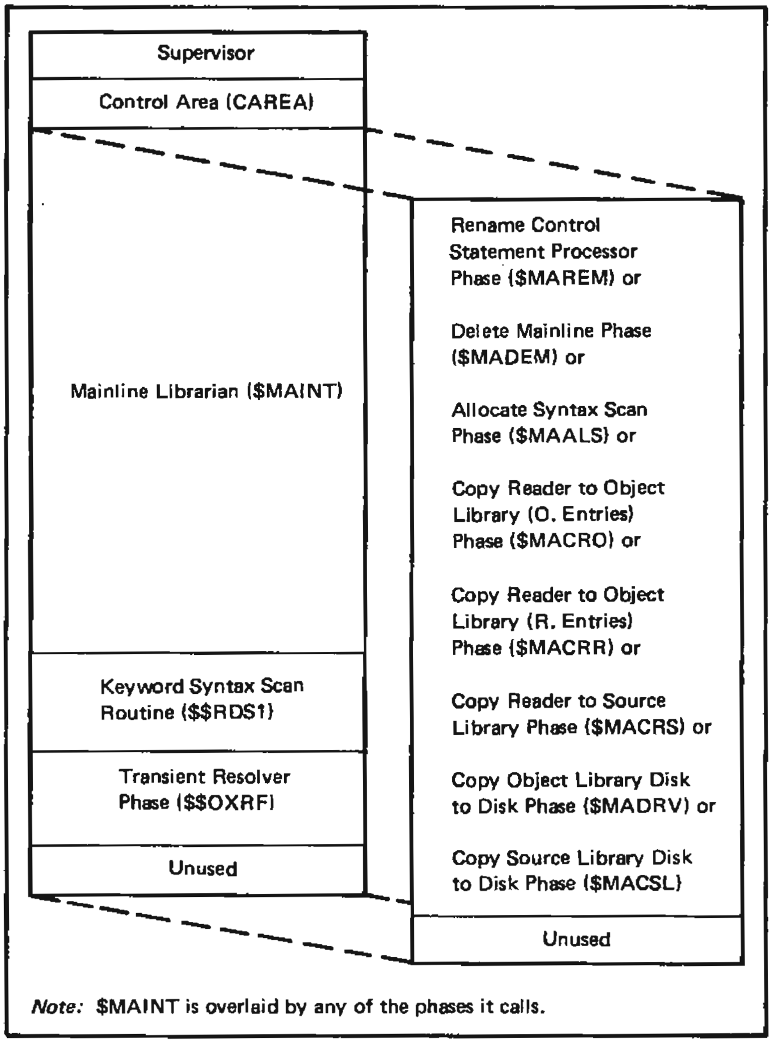

Supervisor
Control Area (CAREA)
Mainline Librarian ($MAINT)
Keyword Syntax Scan Routine ($$RDS1)
Transient Resolver Phase ($$OXRF)
Unused
Rename Control Statement Processor Phase ($MAREM) or
Delete Mainline Phase ($MADEM) or
Allocate Syntax Scan Phase ($MAALS) or
Copy Reader to Object Library (O. Entries) Phase ($MACRO) or
Copy Reader to Object Library (R. Entries) Phase ($MACRR) or
Copy Reader to Source Library Phase ($MACRS) or
Copy Object Library Disk to Disk Phase ($MADRV) or
Copy Source Library Disk to Disk Phase ($MACSL)
Unused
Note: $MAINT is overlaid by any of the phases it calls.

Figure 7-2. Storage Map of the Mainline Librarian Phase ($MAINT)

Allocate Syntax Scan Phase ($MAALS)

Entry Point: $MAALS

Storage Map: Figure 7-3

Chart: GB

Functions:

- Checks for required parameters on the // ALLOCATE statement.
- Checks the availability of the specified pack.
- Checks volume label space requirements.
- Locates the first available space for libraries if the pack does not already contain allocated library space.
- Passes track size of existent libraries to the control area.
- Determines if the object library is being allocated with SYSTEM-YES; if so, the size of the scheduler work area is computed. If the object library is to be reallocated or deallocated, the library is checked to ensure that it is not on the active SYSRES pack.

Input: Partially built control area (CAREA) and volume label

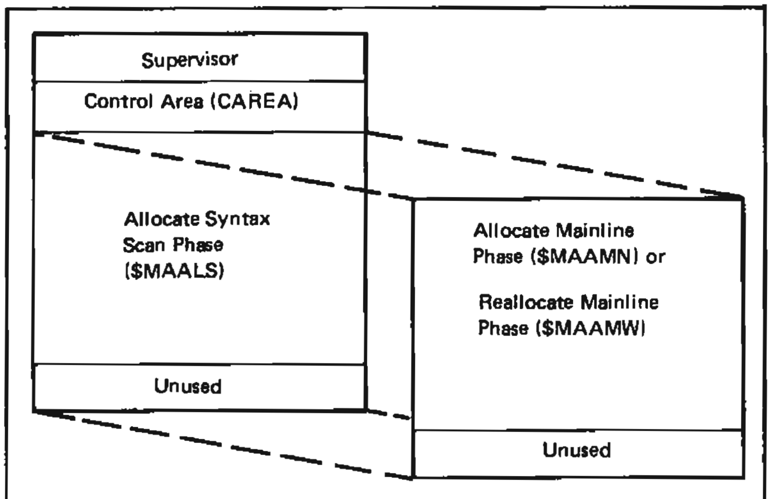
Output: Complete control area (CAREA)

Exit:

- Normal

  1. Allocate Mainline phase ($MAAMN) if no work pack is required
  2. Reallocate Mainline phase ($MAAMW) if a work pack is required

- Error – Halt/Syslog transient


Supervisor
Control Area (CAREA)
Allocate Syntax Scan Phase ($MAALS)
Unused
Allocate Mainline Phase ($MAAMN) or
Reallocate Mainline Phase ($MAAMW)
Unused

Note: $MAALS is overlaid by either $MAAMN or $MAAMW.

Figure 7-3. Storage Map of the Allocate Syntax Scan Phase ($MAALS)

Allocate Mainline Phase ($MAAMN)

Entry Point: $MAAMN

Storage Map: Figure 7-4

Chart: GC

Functions:

- Deallocates:
  1. Source library.
  2. Object library.
  3. Volume label.
- Passes control to other phases to:
  1. Allocate libraries.
  2. Reset volume label entries.

Input: Control area (CAREA) and volume label

Output: Updated volume label and libraries

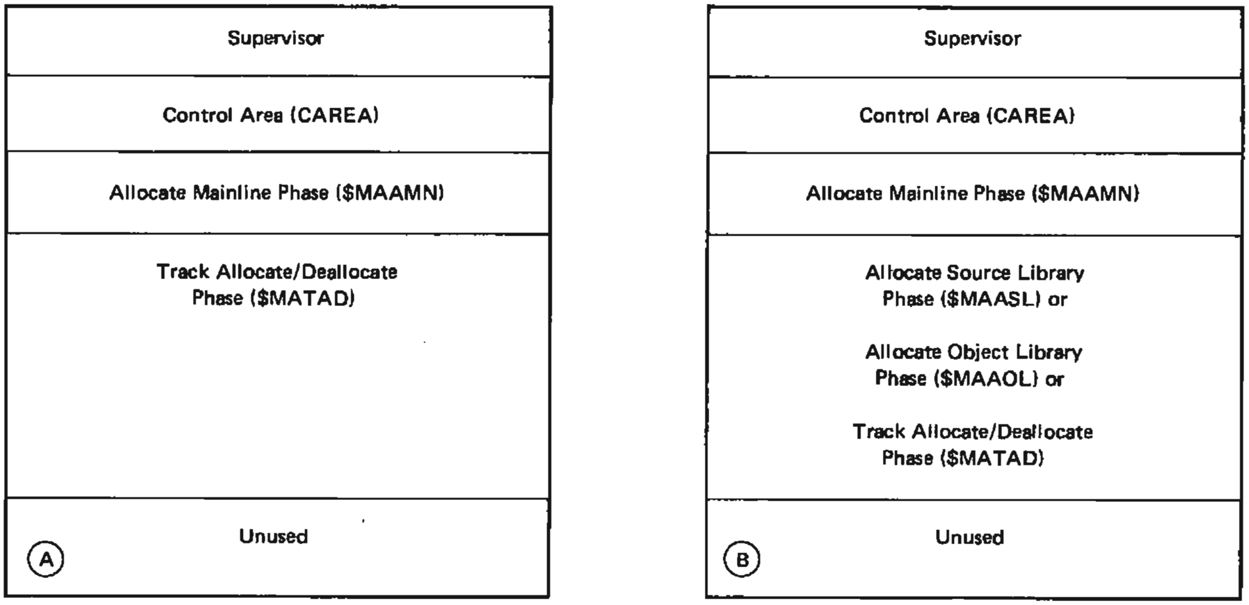
Exit:

- Normal – Mainline Librarian phase ($MAINT) upon completion of specified function
- Error – None

Allocate Object Library Phase ($MAAOL)

Entry Point: $MAAOL

Chart: GD


Supervisor
Control Area (CAREA)
Allocate Mainline Phase ($MAAMN)
Track Allocate/Deallocate Phase ($MATAD)
Unused
A
Supervisor
Control Area (CAREA)
Allocate Mainline Phase ($MAAMN)
Allocate Source Library Phase ($MAASL) or
Allocate Object Library Phase ($MAAOL) or
Track Allocate/Deallocate Phase ($MATAD)
Unused
B

Note: $MAAMN calls in other phases depending upon the function to be performed:

- Deallocate – Insert A is the storage map for the deallocate function.
- Allocate – Insert B is the storage map for the allocate function.

Figure 7-4. Storage Map of the Allocate Mainline Phase ($MAAMN)

Functions:

- Builds the object part of the system directory.
- Allocates a scheduler work area (SWA).

Input:

- System directory
- Control area (CAREA)

Output:

- Updated system directory
- Allocated object library

Exit:

- Normal – Allocate Mainline phase ($MAAMN)
- Error – None

Where Used: $MAAMN

Allocate Source Library Phase ($MAASL)

Entry Point: $MAASL

Chart: GE

Functions:

- Builds the source part of the system directory.
- Chains the library sectors together.
- Indicates the end of the system directory by inserting X'FFFF' in the last sector.

Input: Control area (CAREA) and existing system directory

Output:

- Updated system directory
- Chained library and directory sectors

Exit:

- Normal – Allocate Mainline phase ($MAAMN)
- Error – Halt/Syslog transient

Where Used: $MAAMN

Track Allocate/Deallocate Phase ($MATAD)

Entry Point: $MATAD

Storage Map: Figure 7-4

Chart: GF

Functions:

- Turns off reserved library space indicators.
- Moves in new library start and end addresses.
- Indicates the tracks reserved for libraries.

Input: Existing volume label

Output: Updated volume label

Exit:

- Normal – Allocate Mainline phase ($MAAMN)
- Error – Halt/Syslog transient

Where Used: $MAAMN

Reallocate Mainline Phase ($MAAMW)

Entry Point: $MAAMW

Storage Map: Figure 7-5

Chart: GG

Functions:

- Calls the Find phase ($MAFND) to locate the specified library.
- Calls the appropriate phase to copy the specified library (object or source) onto a work pack.
- Calls the Allocate Mainline phase ($MAAMN) to deallocate the library just copied.

Input: Control area (CAREA)

Output: Reallocated source and/or object libraries

Exit:

- Normal – Mainline Librarian phase ($MAINT)
- Error – Halt/Syslog transient

Find Phase ($MAFND)

Entry Point: $MAFND

Chart: GH

Functions:

- Locates a specified entry in either the object or source libraries.
- Returns the displacement of the entry to the calling phase if the entry was found or returns a code to the calling phase indicating that the entry was not found.

Input:

- IOB
- Buffer with the directory entries
- Control area (CAREA)

Output:

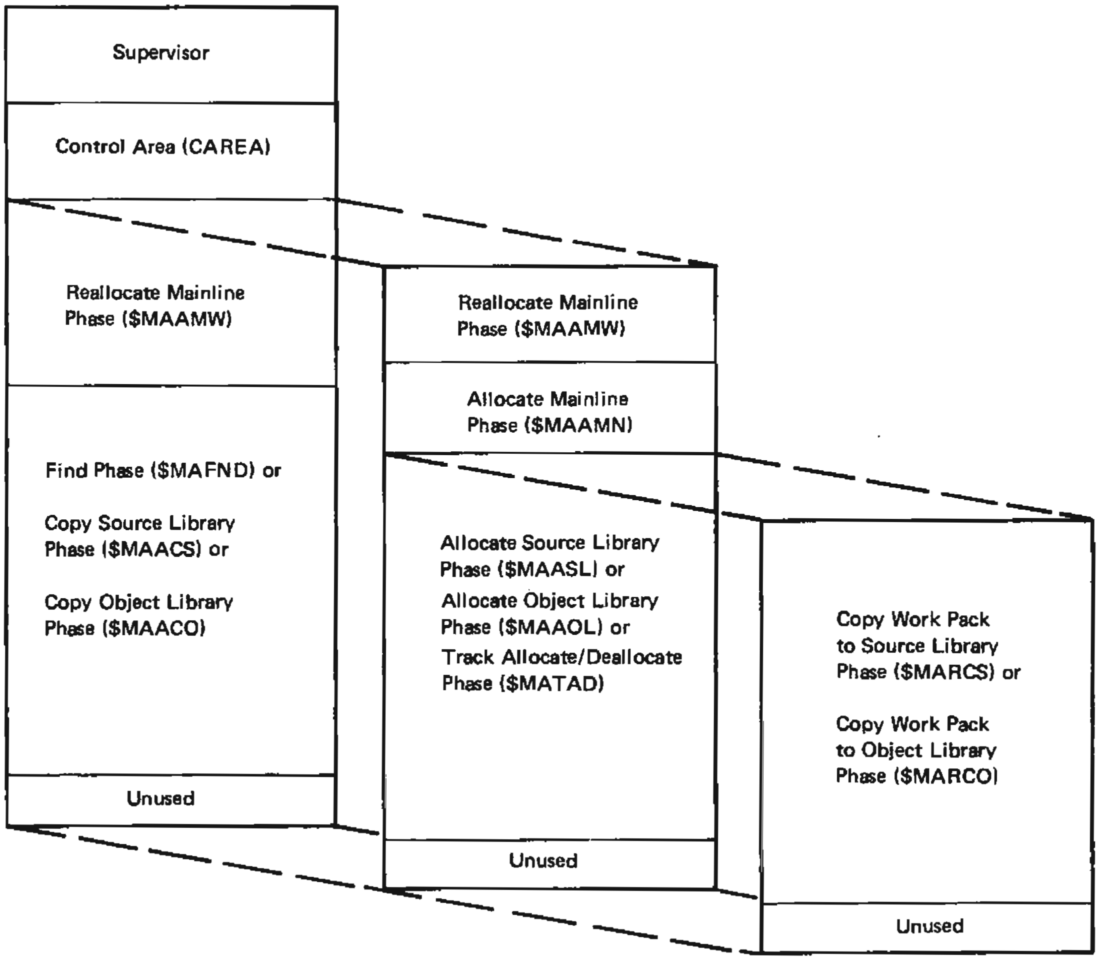
- Location of the specified routine in XR1
- Return code indicating that the routine was not found

Exit:

- Normal – Calling phase
- Error – None

Where Used:

- $MAACS
- $MAACO
- $MARCO
- $MARCS
- $MAAMW
- $MAREM
- $MADEM
- $MADRV
- $MAOLH
- $MAPPM
- $MACRR
- $MACRO
- $MACSL


Supervisor
Control Area (CAREA)
Reallocate Mainline Phase ($MAAMW)
Find Phase ($MAFND) or
Copy Source Library Phase ($MAACS) or
Copy Object Library Phase ($MAACO)
Unused
Reallocate Mainline Phase ($MAAMW)
Allocate Mainline Phase ($MAAMN)
Allocate Source Library Phase ($MAASL) or
Allocate Object Library Phase ($MAAOL) or
Track Allocate/Deallocate Phase ($MATAD)
Unused
Copy Work Pack to Source Library Phase ($MARCS) or
Copy Work Pack to Object Library Phase ($MARCO)
Unused

Note: If entries are to be copied, $MAACS, $MAFND, and $MAACO are loaded to copy the source and object libraries to a work pack. $MAAMN is then loaded which loads $MATAD to allocate the libraries. Then $MAASL, $MAAOL, and $MATAD are loaded to allocate new libraries. $MAAMW regains control and loads $MARCS and $MARCO to copy to source and object libraries from the work pack to the newly created libraries.

Figure 7-5. Storage Map of the Reallocate Mainline Phase ($MAAMW)

Copy Object Library Phase ($MAACO)

Entry Point: $MAACO

Chart: GI

Functions:

- Copies the object library to a work pack when it is being reallocated or when the source library is being allocated and needs more room.
- Reserves one or three tracks for the directory entries and allocates the remaining space for permanent entries.
- Places the start address of the work pack directory in the control area (CAREA).

Input: Control area (CAREA)

Output: Object library copied onto specified work pack

Exit:

- Normal – Reallocate Mainline phase ($MAAMW)
- Error – Halt/Syslog transient

Where Used: $MAAMW

Copy Source Library Phase ($MAACS)

Entry Point: $MAACS

Chart: GJ

Functions:

- Determines the number of directory sectors and chains them together.
- Loads the source library entries onto the work pack following the directory entries.
- Indicates the next available sector for the object library.

Input: Control area (CAREA) containing pointer to work pack

Output: Source library copied onto the work pack

Exit:

- Normal – Reallocate Mainline phase ($MAAMW)
- Error – Halt/Syslog transient

Where Used: $MAAMW

Copy Work Pack to Object Library Phase ($MARCO)

Entry Point: $MARCO

Chart: GK

Function: Copies object library entries from the work pack to a reallocated object library.

Input: Control area (CAREA)

Output:

- Reallocated object library
- Updated volume label

Exit:

- Normal – Reallocate Mainline phase ($MAAMW)
- Error – Halt/Syslog transient

Where Used: $MAAMW

Copy Work Pack to Source Library Phase ($MARCS)

Entry Point: $MARCS

Chart: GL

Functions: Copies the source library from the work pack to a newly allocated source library.

Input: Control area (CAREA)

Output:

- Reallocated source library
- Updated volume label
- Updated system directory

Exit:

- Normal – Reallocate Mainline phase ($MAAMW)
- Error – Halt/Syslog transient

Where Used: $MAAMW

Rename Control Statement Processor Phase ($MAREM)

Entry Point: $MAREM

Storage Map: Figure 7-6

Chart: GM

Functions:

- Builds a parameter list with specified new name.
- Checks for a duplicate name and aborts if duplicate is found.
- Deletes the old name and replaces it with a new name.

Input: Control area (CAREA)

Output: Library directory with specified new name

Exit:

- Normal – Mainline Librarian phase ($MAINT)
- Error – Halt/Syslog transient

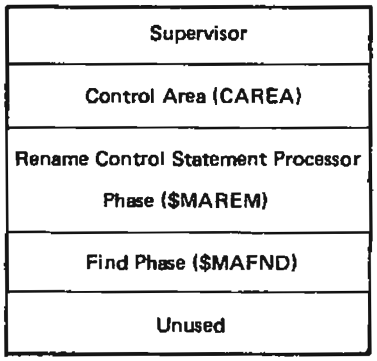


Figure 7-6. Storage Map of the Rename Control Statement Processor Phase ($MAREM)

Delete Mainline Phase ($MADEM)

Entry Point: $MADEM

Storage Map: Figure 7-7

Chart: GN

Functions:

- Deletes specified library entries in the object and/or source libraries.
- Updates the system directory and the volume label to reflect the current status of the libraries.

Input: Control area (CAREA)

Output:

- Updated system directory
- Deleted library entry
- Updated volume label

Exit:

- Normal – Mainline Librarian phase ($MAINT)
- Error – Halt/Syslog transient

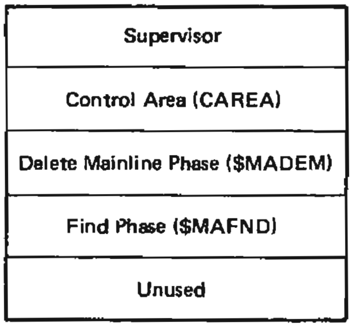


Figure 7-7. Storage Map of the Delete Mainline Phase ($MADEM)

Copy Reader To Source Library Phase ($MACRS)

Entry Point: $MACRS

Storage Map: Figure 7-8

Chart: GO

Functions:

- Determines if the requested name already exists in the source library.
- Determines if there is enough space in the source library for the program that is to be entered there.
- Copies the program into the source library.
- Updates the source and system libraries to reflect the entry of the copied program.

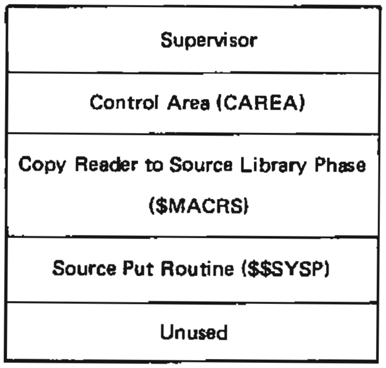

Supervisor
Control Area (CAREA)
Copy Reader to Source Library Phase ($MACRS)
Source Put Routine ($$SYSP)
Unused

Figure 7-8. Storage Map of the Copy Reader to Source Library Phase ($MACRS)

Input:

- Source statements or procedures
- Source library on the specified disk pack

Output:

- Copied program in the source library
- Updated source and system directories

Exit:

- Normal – Mainline Librarian phase ($MAINT)
- Error – Halt/Syslog transient

Copy Object Library — Disk to Disk Phase ($MADRV)

Entry Point: $MADRV

Storage Map: This is the first phase called for three functions:

- Copy system (Figure 7-9)
- Copy all libraries (Figure 7-10)
- Copy entries into the object library (Figure 7-11)

Chart: GP

Functions:

- Calls the Object Library Handler phase ($MAOLH) to locate available space for entry.
- Calls the Find phase ($MAFND) to locate specified entry.
- Creates the volume label and object directory in storage and writes them on the specified disk.

Input: Control area (CAREA)

Output: Specified library copy on the specified disk.

Exit:

- Normal — Mainline Librarian phase ($MAINT)
- Error — Halt/Syslog transient

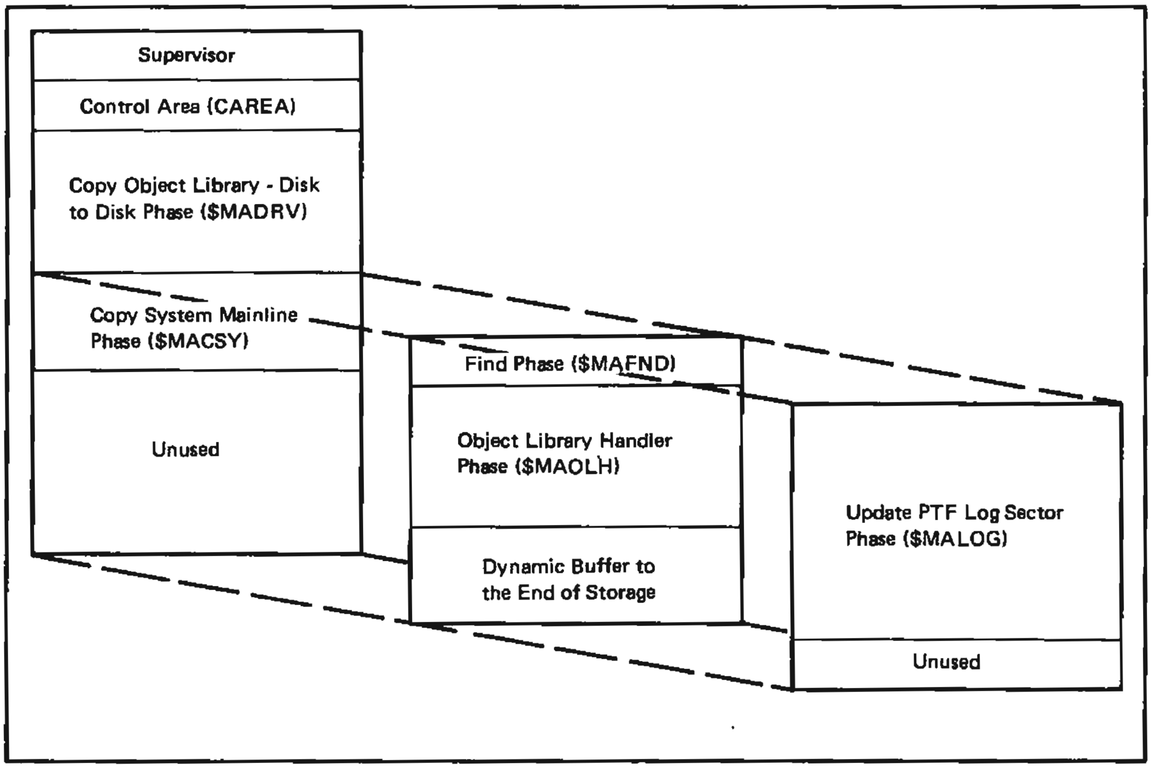

Supervisor
Control Area (CAREA)
Copy Object Library - Disk to Disk Phase ($MADRV)
Copy System Mainline Phase ($MACSY)
Unused
Find Phase ($MAFND)
Object Library Handler Phase ($MAOLH)
Dynamic Buffer to the End of Storage
Update PTF Log Sector Phase ($MALOG)
Unused

Figure 7-9. Storage Map of the Copy Object Library - - Disk to Disk Phase ($MADRV) for the Copy System Function

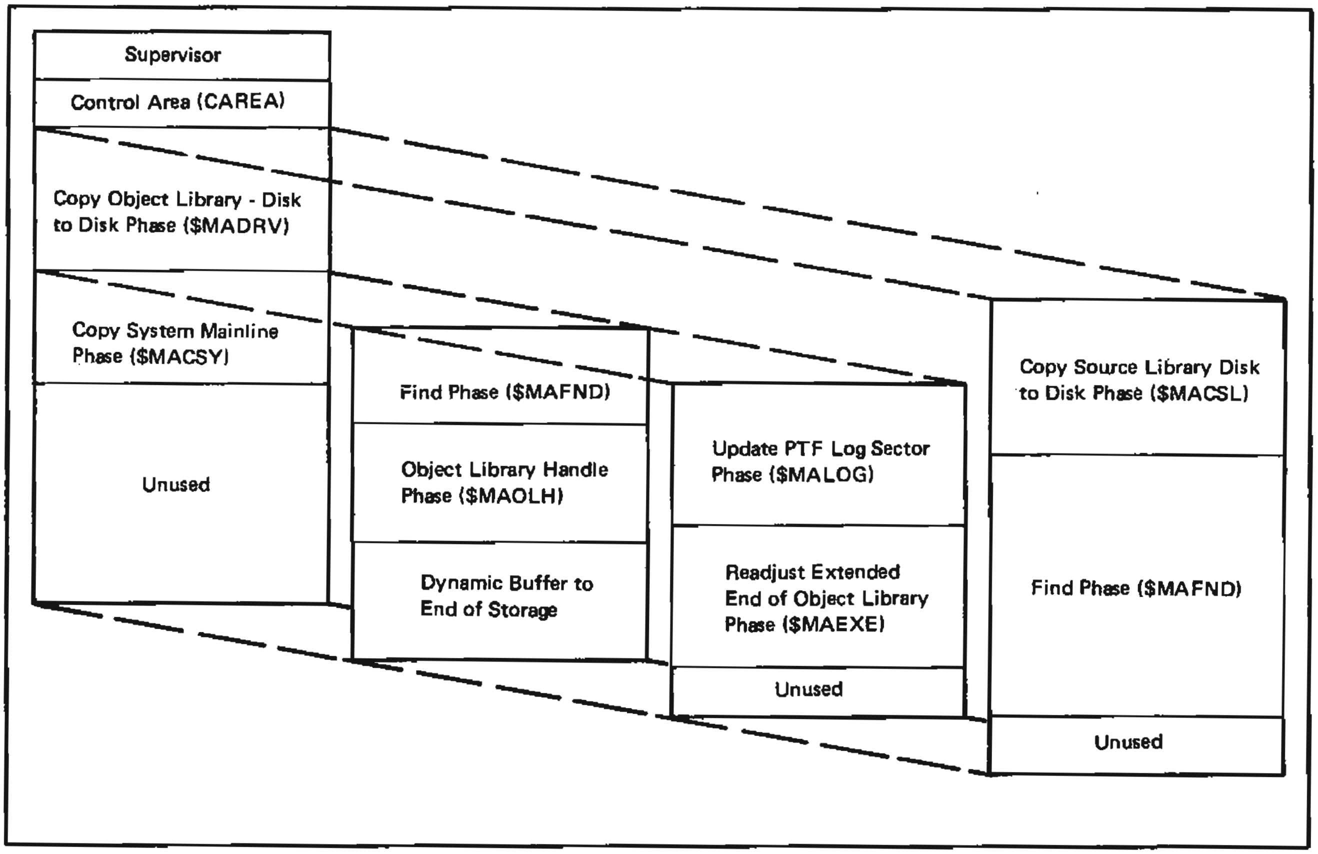

Supervisor
Control Area (CAREA)
Copy Object Library - Disk to Disk Phase ($MADRV)
Copy System Mainline Phase ($MACSY)
Unused
Find Phase ($MAFND)
Object Library Handle Phase ($MAOLH)
Dynamic Buffer to End of Storage
Update PTF Log Sector Phase ($MALOG)
Readjust Extended End of Object Library Phase ($MAEXE)
Unused
Copy Source Library Disk to Disk Phase ($MACSL)
Find Phase ($MAFND)
Unused

Figure 7-10. Storage Map of the Copy Object Library - - Disk to Disk Phase ($MADRV) for the Copy All Libraries Function

Supervisor
Control Area (CAREA)
Copy Object Library — Disk to Disk Phase ($MADRV)
Find Phase ($MAFND)
Object Library Handler Phase ($MAOLH)
Dynamic Buffer to the End of Storage

Figure 7-11. Storage Map of the Copy Object Library - - Disk to Disk Phase ($MADRV) for the Copy Entries to Object Library Function

Copy System Mainline Phase ($MACSY)

Entry Point: $MACSY

Storage Map: Figure 7-9

Chart: GQ

Functions: Copies the following from one system disk pack to another specified disk pack:

1. Initial Program Load (IPL) Bootstrap program.
2. System configuration record.
3. Nucleus Initialization program (NIP).

Input: Control area (CAREA)

Output: The IPL Bootstrap program, system configuration record, and Nucleus Initialization program are copied from the current system disk pack to the other specified disk pack.

Exit:

- Normal – Copy Object Library – Disk to Disk phase ($MADRV)
- Error – Halt/Syslog transient

Object Library Handler Phase ($MAOLH)

Entry Point: $MAOLH

Chart: GR

Functions:

- Loads an entry into the object library from either the reader or from another object library.
- Checks for available space in the object library unless an R. entry is coming from the reader to the object library.
- Dynamically allocates needed space when adding a temporary entry to the object library.
- Checks for duplicate entry names and deletes the old entry name when possible.
- Finds the least amount of available space when adding a permanent entry.
- Deletes temporary entries when adding permanent entries.
- Updates the control area (CAREA) to reflect the addresses of the added/deleted object library entries.

Input:

- XR1 points to an area in which the FIND parameter list will be built
- XR2 points to control area CAREA

Output:

- Address of the object library entry in the control area
- Object library dynamically increased if necessary

Exit:

- Normal – Calling routine
- Error – Halt/Syslog transient

Where Used:

- $MADRV
- $MACRR
- $MACRO

Update PTF Log Sector Phase ($MALOG)

Entry Point: $MALOG

Chart: GS

Function:

- Updates the PTF log sector following a delete or copy.
- If a module with a PTF applied to it is deleted, the PTF is also deleted. All of these PTFs will be deleted; the others will be left adjusted.
- Following a copy, all PTFs pertaining to the modules copied will also be copied onto the beginning of the PTF log sector; all others will be right adjusted. If the sector is full, the PTFs on the right end will be lost.

Input: PTF log sector and control area

Output: Updated PTF log sector

Exit:

- Normal – Calling routine
- Error – Halt/Syslog transient

Where Used:

- $MADRV
- $MACRO
- $MADEM

Copy Reader to Object Library (R. Entries) Phase ($MACRR)

Entry Point: $MACRR

Storage Map: Figure 7-12

Chart: GT

Functions:

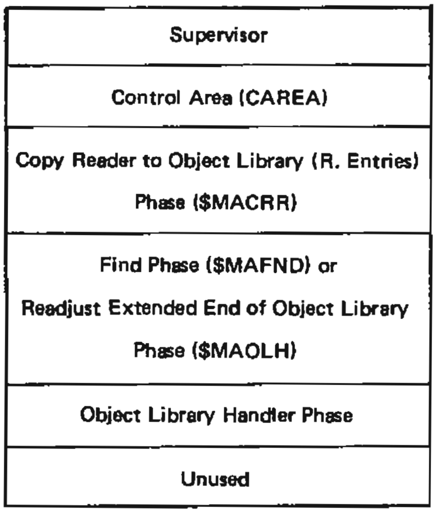
- Reads object statements from the SYSIN device.
- Compresses each statement into 64 bytes.
- Builds an object library from the compressed statements (four statements per sector).
- Builds a directory entry for the object library stating the number of sectors, type, name, disk address, and attribute (temporary or permanent).

Input: Object statements (R. entries) read from SYSIN device.

Output: R. entry in the object library

Exit:

- Normal – Mainline Librarian phase ($MAINT)
- Error – Halt/Syslog transient



Supervisor

Control Area (CAREA)

Copy Reader to Object Library (R. Entries) Phase ($MACRR)

Find Phase ($MAFND) or Readjust Extended End of Object Library Phase ($MAOLH)

Object Library Handler Phase

Unused

Figure 7-12. Storage Map of the Copy Reader to Object Library (R. Entries) Phase ($MACRR)

Readjust Extended End of Object Library Phase ($MAEXE)

Entry Point: $MAEXE

Chart: GU

Functions:

- Reads in the volume label.
- Determines the allocated end of the object library.
- Determines the number of tracks and sectors that need to be added to the allocated library end in order to contain the temporary object library entries.
- Updates both the volume label and system directories.

Input: Volume label

Output:

- Updated volume label
- Updated system directory

Exit:

- Normal – Calling phase
- Error – Halt/Syslog transient

Where Used:

- $MACRR
- $MACRO
- $MADRV

Copy Reader to Object Library (O. Entries) Phase ($MACRO)

Entry Point: $MACRO

Storage Map: Figure 7-13

Chart: GV

Functions:

- Reads the header control statement and uses the information to build a directory entry.
- Reads, compresses, and inserts the object records into the object library.
- Inserts the directory entry into the volume label when the // CEND control statement is read.
- In following text, column descriptions indicate the format of the header, text, relocation dictionary, and end statements for an object directory O. type module being placed in the object library.

Input:

- Header control statement
- Object records
- // CEND control statement

Output:

- O. entry in the object library
- Updated volume label

Exit:

- Normal – Mainline Librarian phase ($MAINT)
- Error – Halt/Syslog transient

The following are descriptions of the header, text, relocation dictionary, and end statements for an object directory O. type module.

Header Statement

Column	*Contents*
1	H
2-4	Blank
5	O (character O)
6-8	Blank
9-14	Name from the first TITLE card
15-17	Blank
18-21	0000 (character zeros)
22-23	Number of text sections in hexadecimal
24-27	Link edit address in hexadecimal
28-29	Displacement of RLD in hexadecimal
30-33	SCA in hexadecimal
34-35	Core size in hexadecimal
36-39	Attributes in hexadecimal
40-41	Level in hexadecimal
42-45	Total number of sectors in hexadecimal
46-85	Blank
86-88	Self check bytes
89-92	First four characters of module name
93-96	Sequence number (always 0001)

Text Statement

Column	*Contents*
1	T
2	Length of text minus one
3-4	Relocated address of first text byte in card
5-85	Text expanded into 4 for 3 format
86-88	Self check bytes
89-92	First four characters of module name
93-96	Sequence number

Relocation Dictionary Statement

Column	*Contents*
1	R
2-4	Blank
5-85	Blanks and RLDs
86-88	Self check bytes
89-92	First four characters of module name
93-96	Sequence number

*Columns 2-85 are in 4 for 3 format

End Statement

Column	*Contents*
1	E
2-35	Blank
36-60	The characters 'LAST'
61-85	Blank
86-88	Self check bytes
89-92	First four bytes of module name
93-96	Sequence number

Copy Source Library – Disk to Disk Phase ($MACSL)

Entry Point: $MACSL

Storage Map: Figure 7-14 (see Figure 7-10 for the copy all libraries function)

Chart: GW

Functions:

- Copies specified source library entries to another disk source library or within the same library.
- Checks all new entry names for duplication.
- Deletes duplicate entry names when possible.

Input:

- Control area (CAREA)
- Volume labels

Output: Updated source library

Exit:

- Normal – Mainline Librarian phase ($MAINT)
- Error – Halt/Syslog transient

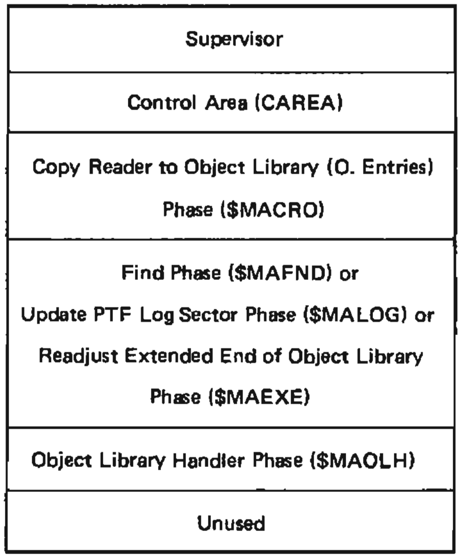

Supervisor
Control Area (CAREA)
Copy Reader to Object Library (O. Entries) Phase ($MACRO)
Find Phase ($MAFND) or
Update PTF Log Sector Phase ($MALOG) or
Readjust Extended End of Object Library Phase ($MAEXE)
Object Library Handler Phase ($MAOLH)
Unused

Figure 7-13. Storage Map of the Copy Reader to Object Library (O. Entries) Phase ($MACRO)

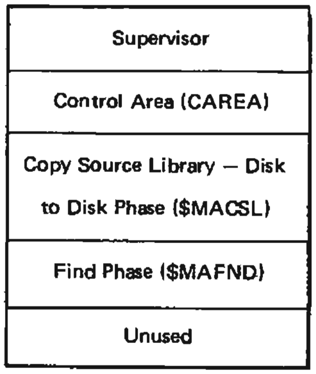


Figure 7-14. Storage Map of the Copy Source Library -- Disk to Disk Phase ($MACSL)

Print/Punch Mainline Phase ($MAPPM)

Entry Point: $MAPPM

Storage Map: Figure 7-15 shows the directory print function. Figure 7-16 shows the library print/punch function.

Chart: GX

Functions: Prints and/or punches:

1. Source directory
2. Object directory
3. Source library
4. Object library

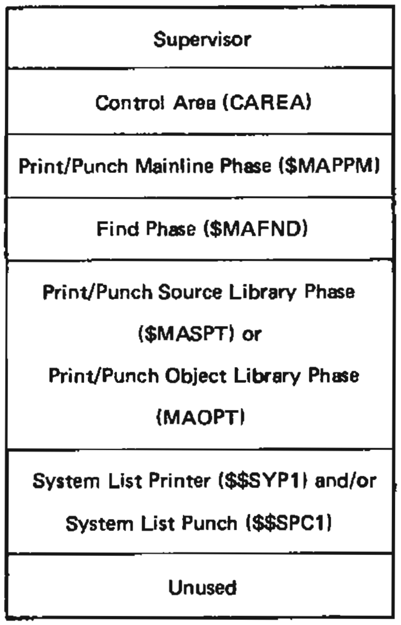

Supervisor
Control Area (CAREA)
Print/Punch Mainline Phase ($MAPPM)
Find Phase ($MAFND)
Print/Punch Source Library Phase ($MASPT) or Print/Punch Object Library Phase (MAOPT)
System List Printer ($$SYP1) and/or System List Punch ($$SPC1)
Unused

Figure 7-16. Storage Map of the Print/Punch Mainline Phase ($MAPPM) for the Library Print/Punch Function

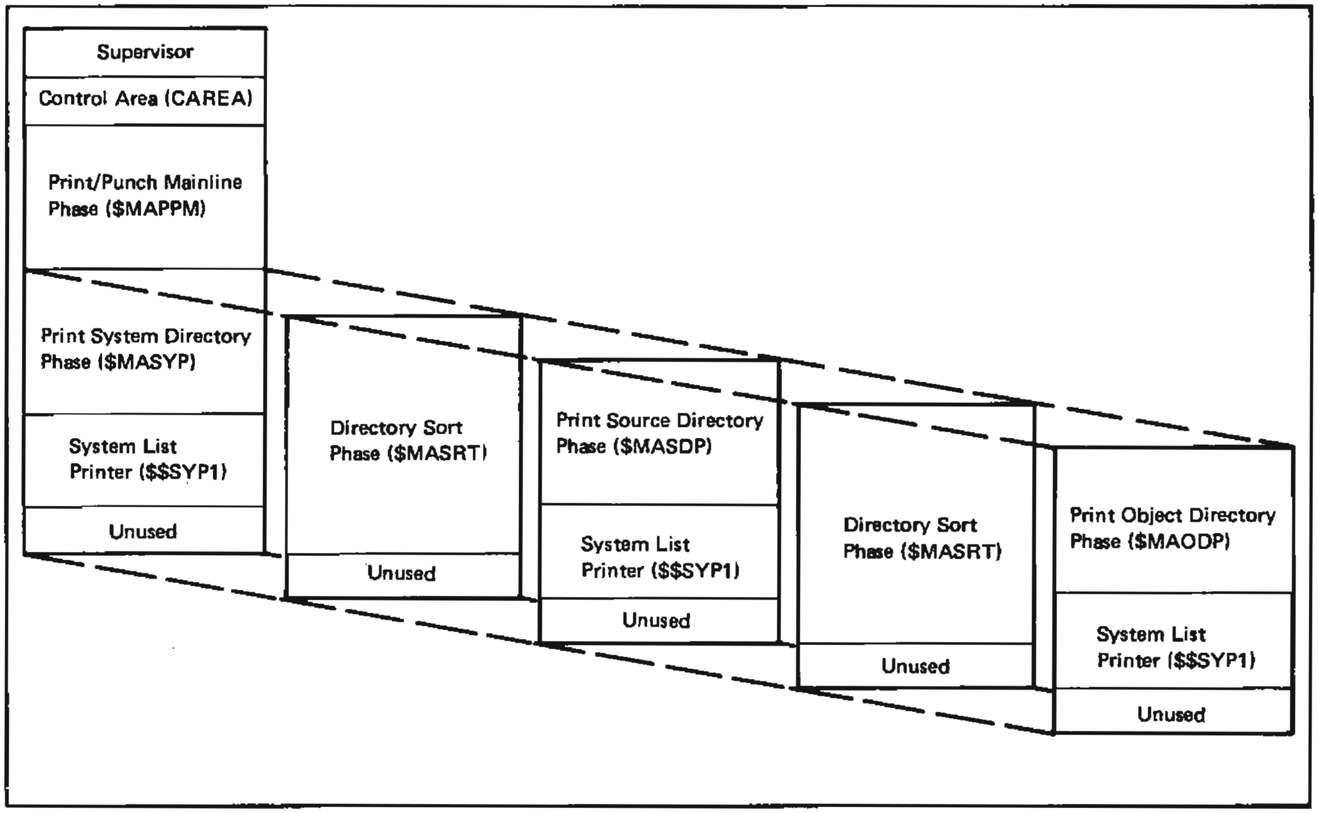

Supervisor
Control Area (CAREA)
Print/Punch Mainline Phase ($MAPPM)
Print System Directory Phase ($MASYP)
System List Printer ($$SYP1)
Unused
Directory Sort Phase ($MASRT)
Unused
Print Source Directory Phase ($MASDP)
System List Printer ($$SYP1)
Unused
Directory Sort Phase ($MASRT)
Unused
Print Object Directory Phase ($MAODP)
System List Printer ($$SYP1)
Unused

Figure 7-15. Storage Map of the Print/Punch Phase ($MAPPM) for the Directory Print Function

Input: XR1 points to a control area specifying the requested module.

Output: Printed and/or punched module

Exit:

- Normal – Mainline Librarian phase ($MAINT)
- Error – Halt/Syslog transient

Directory Sort Phase ($MASRT)

Entry Point: $MASRT

Chart: GY

Functions:

- Sorts all entries in a source or object directory.
- Returns the address of the sorted entries to the calling routine via XR2.

Input:

- XR1 points to the control area (CAREA)
- XR2 points to letter O (object library to be sorted) or S (source library to be sorted)

Output: XR2 points to the address of the first sector sorted

Exit:

- Normal – Calling routine
- Error – Halt/Syslog transient

Where Used:

- $MASPT
- $MAOPT

Print Source Directory Phase ($MASDP)

Entry Point: $MASDP

Chart: GZ

Functions: Prints out the contents of the source directory.

Input: XR2 contains the address of the source directory.

Output: One printed line of formatted information for each directory entry.

Exits:

- Normal – Print/Punch Mainline phase ($MAPPM)
- Error
  - Halt/Syslog transient if retry operation is not specified
  - Mainline Librarian phase ($MAINT) if RETRY operation is specified

Where Used: $MAPPM

Print Object Directory Phase ($MAODP)

Entry Point: $MAODP

Chart: HA

Functions: Prints out the object directory.

Input:

- Contents of the object library
- Address of the directory in a parameter list pointed to by XR2
- Control area pointed to by XR1 which tells which directory is to be used and what pack that directory is on

Output: Formatted directory

Exits:

- Normal – Print/Punch Mainline phase ($MAPPM)
- Error
  - Halt/Syslog transient if a retry operation is not specified
  - Mainline Librarian phase ($MAINT) if a retry operation is specified

Where Used: $MAPPM

Print System Directory Phase ($MASYP)

Entry Point: $MASYP

Chart: HB

Function: Prints the system directory.

Input:

- $MAINT common area
- Address of the common area in XR1

Output: One printed page of system directory information

Exits:

- Normal – Print/Punch Mainline phase ($MAPPM)
- Error
  - Halt/Syslog transient if a retry operation is not specified
  - Mainline Librarian phase ($MAINT) if a retry operation is specified

Where Used: $MAPPM

Print/Punch Source Library Phase ($MASPT)

Entry Point: $MASPT

Chart: HC

Functions: Prints and/or punches modules from the source library.

Input:

- XR1 points to a control area that specifies where the module resides
- XR2 contains the address of the module's directory entry

Output:

- Object deck if punching was specified
- One 96-character printed line per module if printing was specified

Exits:

- Normal – Print/Punch Mainline phase ($MAPPM)
- Error – Halt/Syslog transient

Where Used: $MAPPM

Print/Punch Object Library Phase ($MAOPT)

Entry Point: $MAOPT

Chart: HD

Functions: Prints and/or punches specified object library modules. In preceding text, column descriptions indicate the format of the punched and/or printed modules.

Input:

- XR1 points to a control area which specifies the requested module
- XR2 contains the address of the module's directory entry

Output:

- Object deck if punching was specified
- One 96-character printed line per module if printing was specified.

Exits:

- Normal – Print/Punch Mainline phase ($MAPPM)
- Error – Halt/Syslog transient

Where Used: $MAPPM

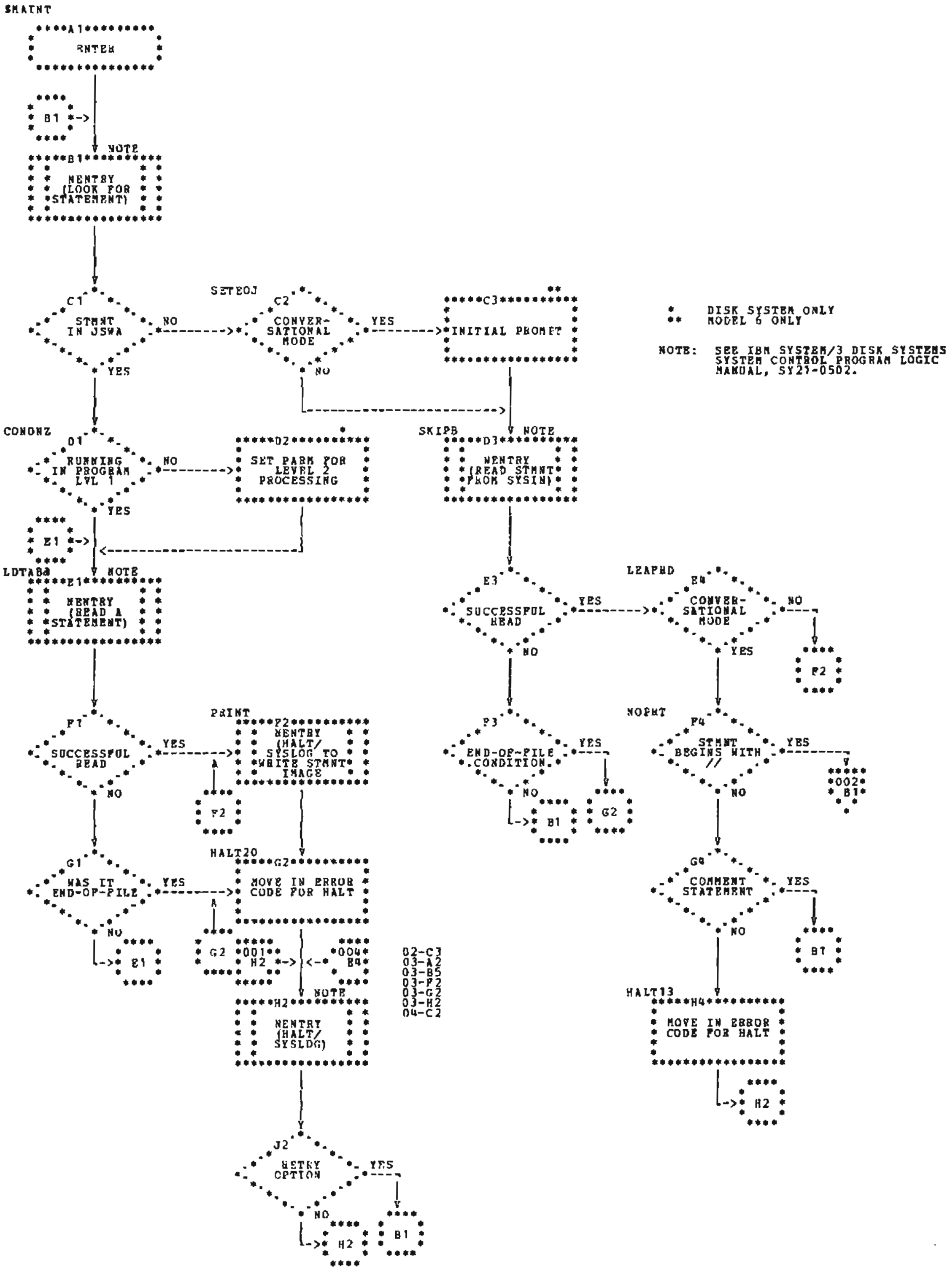
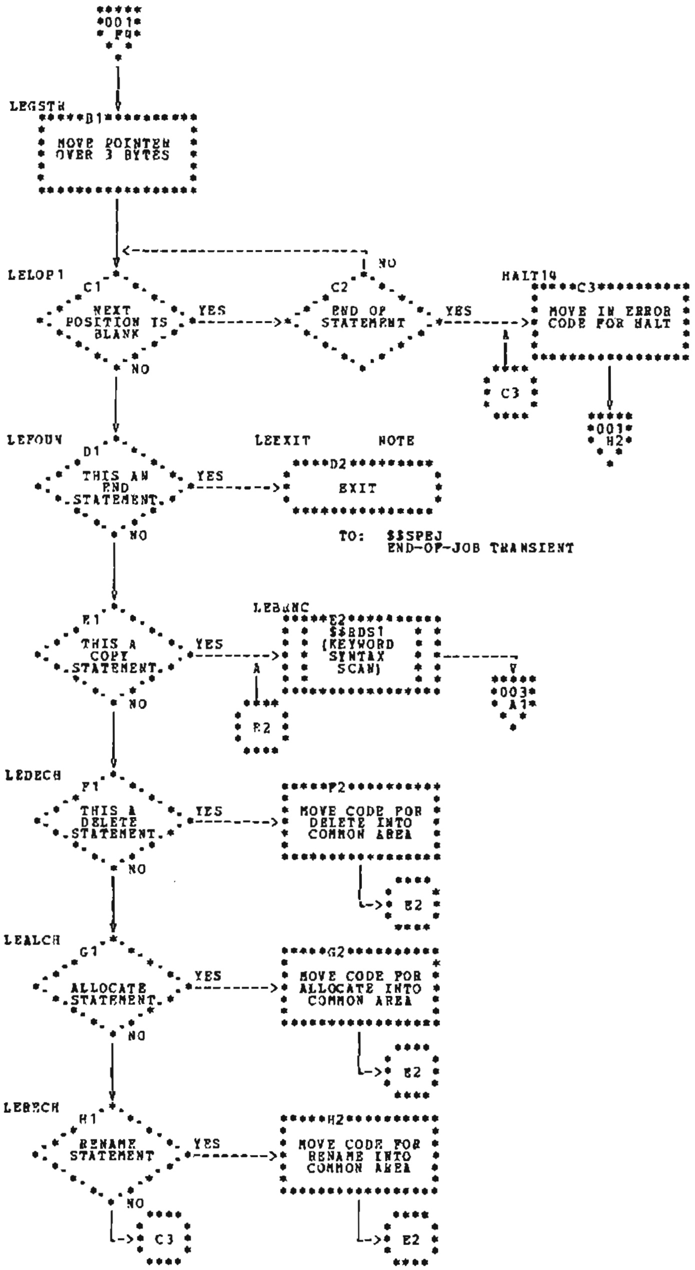


Chart GA (Part 1 of 5). Mainline Librarian Phase



NOTE : SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart GA (Part 2 of 5). Mainline Librarian Phase

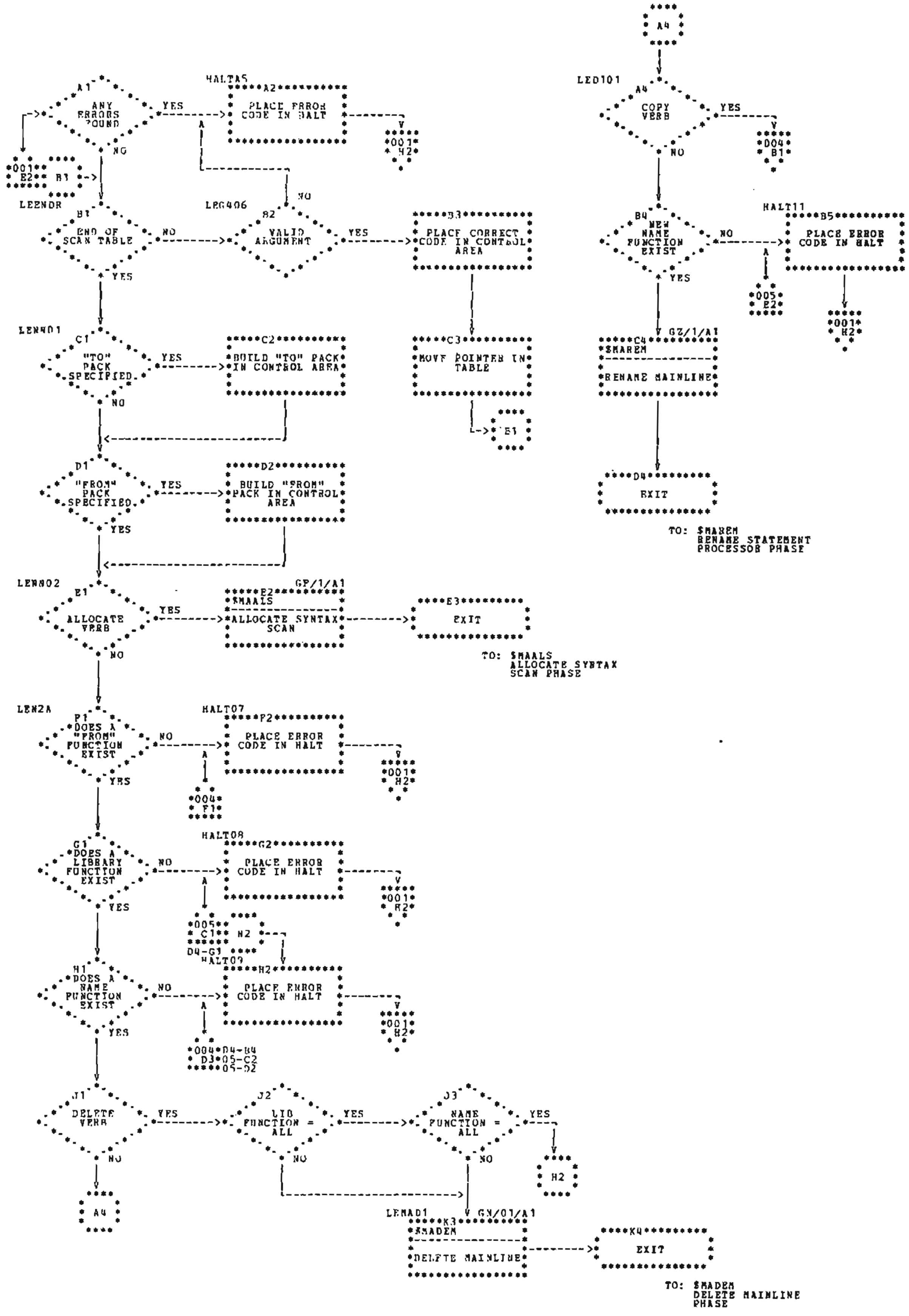


Chart GA (Part 3 of 5). Mainline Librarian Phase

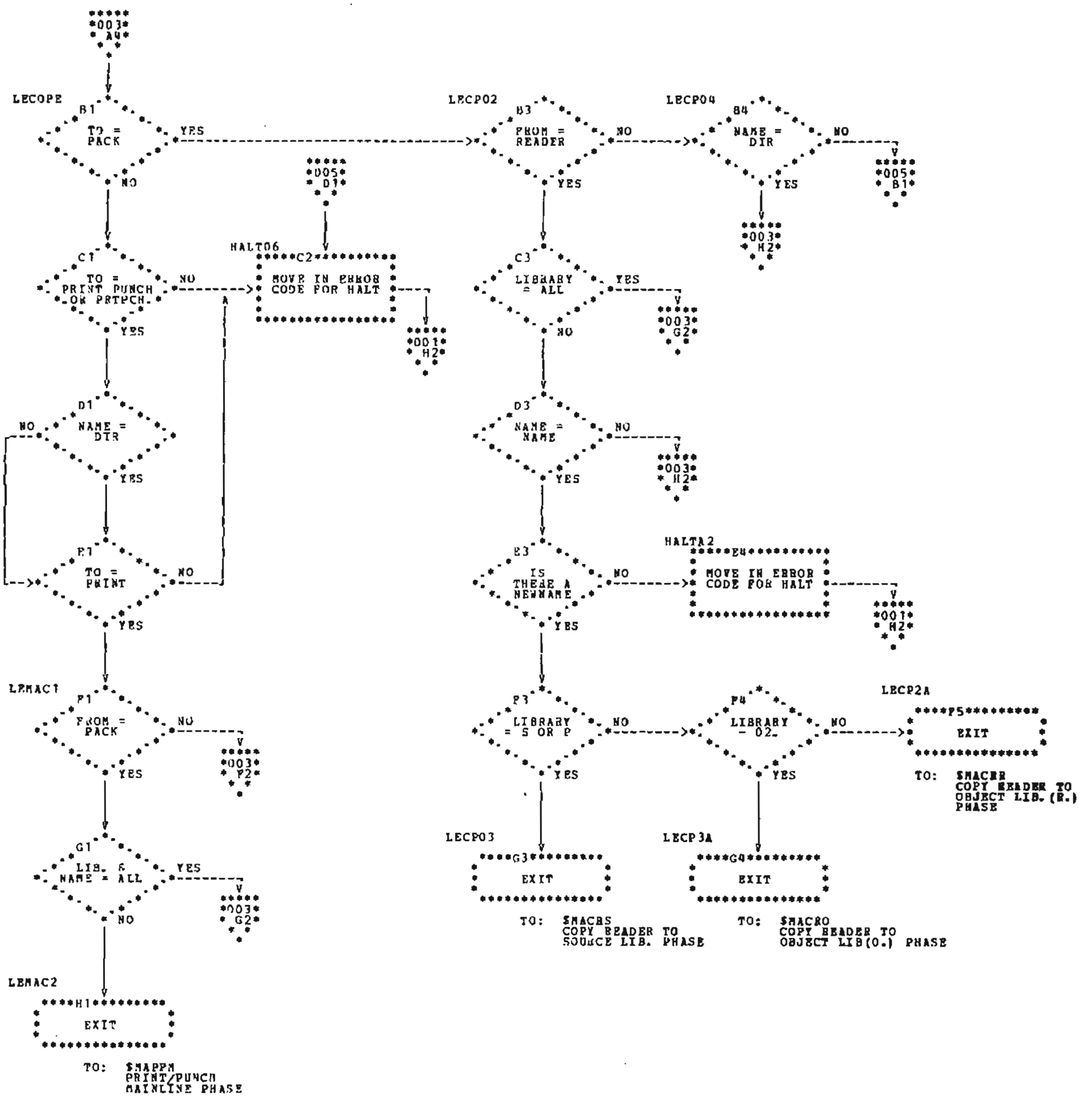


Chart GA (Part 4 of 5). Mainline Librarian Phase

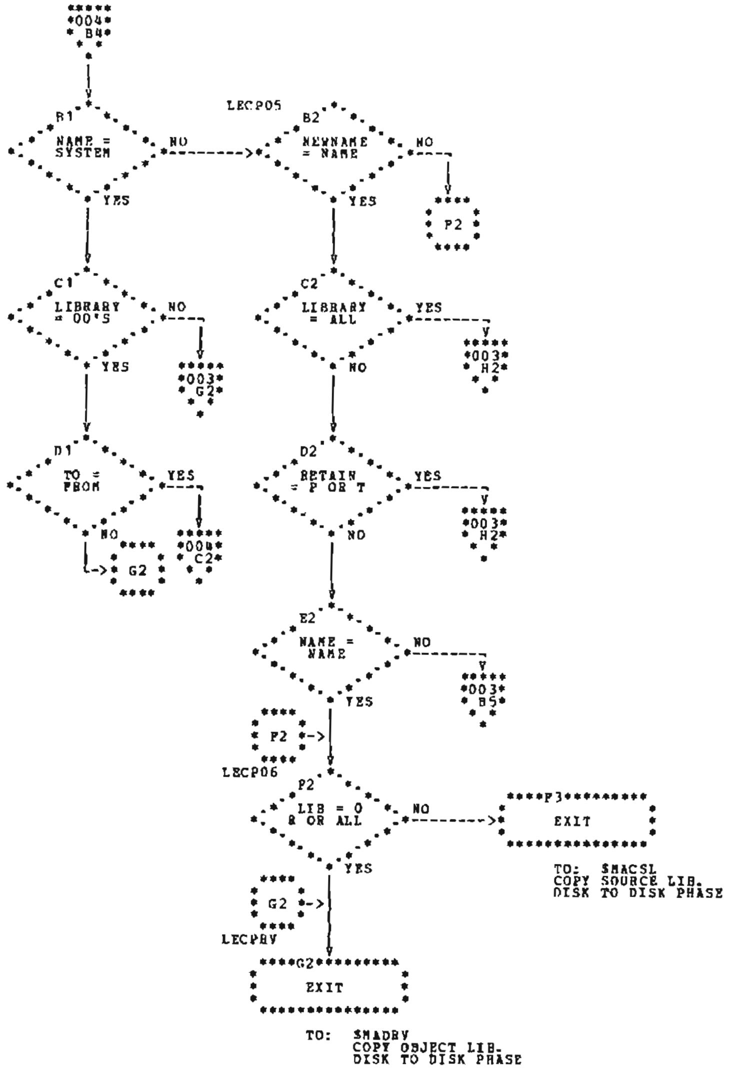


Chart GA (Part 5 of 5). Mainline Librarian Phase

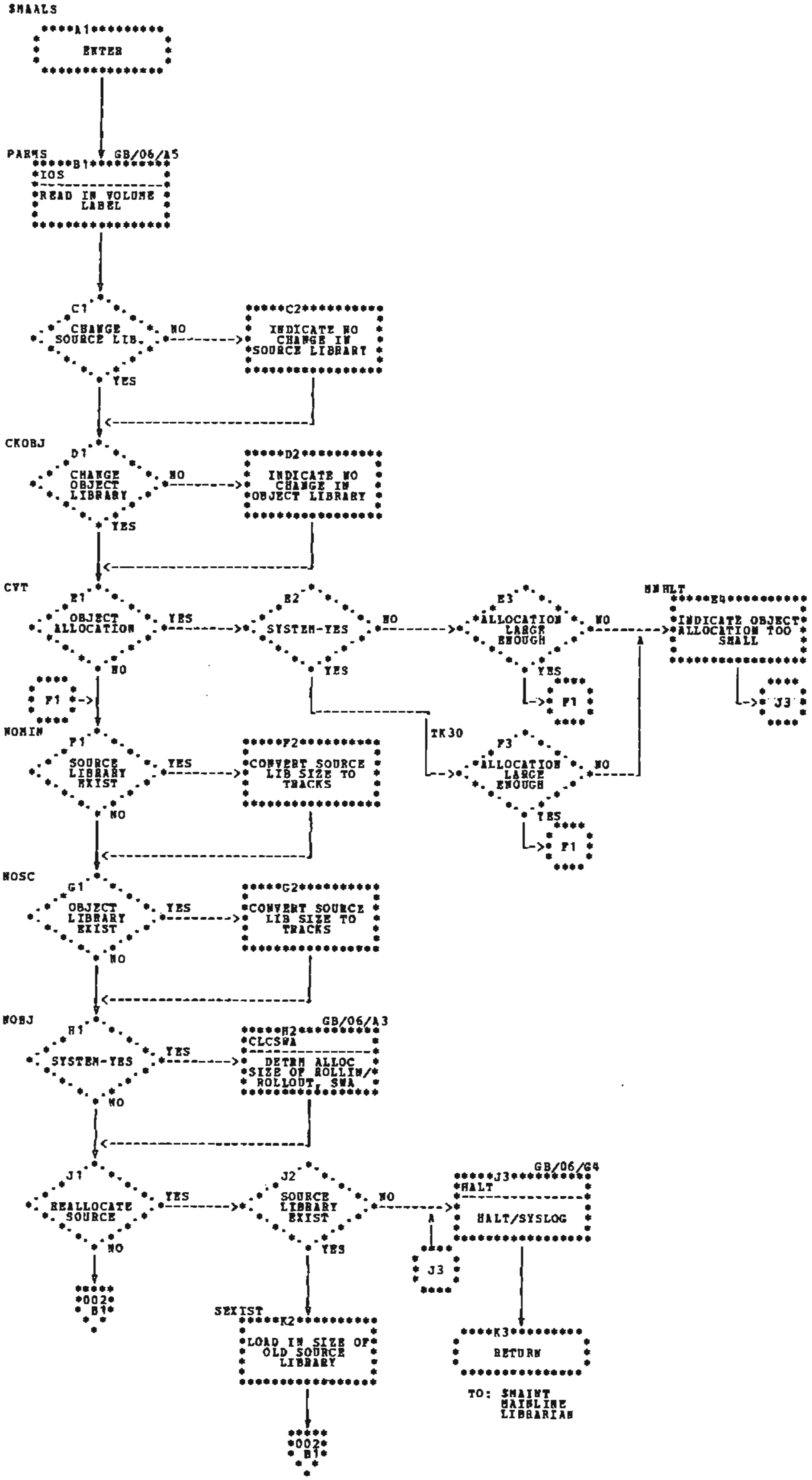


Chart GB (Part 1 of 6). Allocate Syntax Scan Phase

NOTE: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

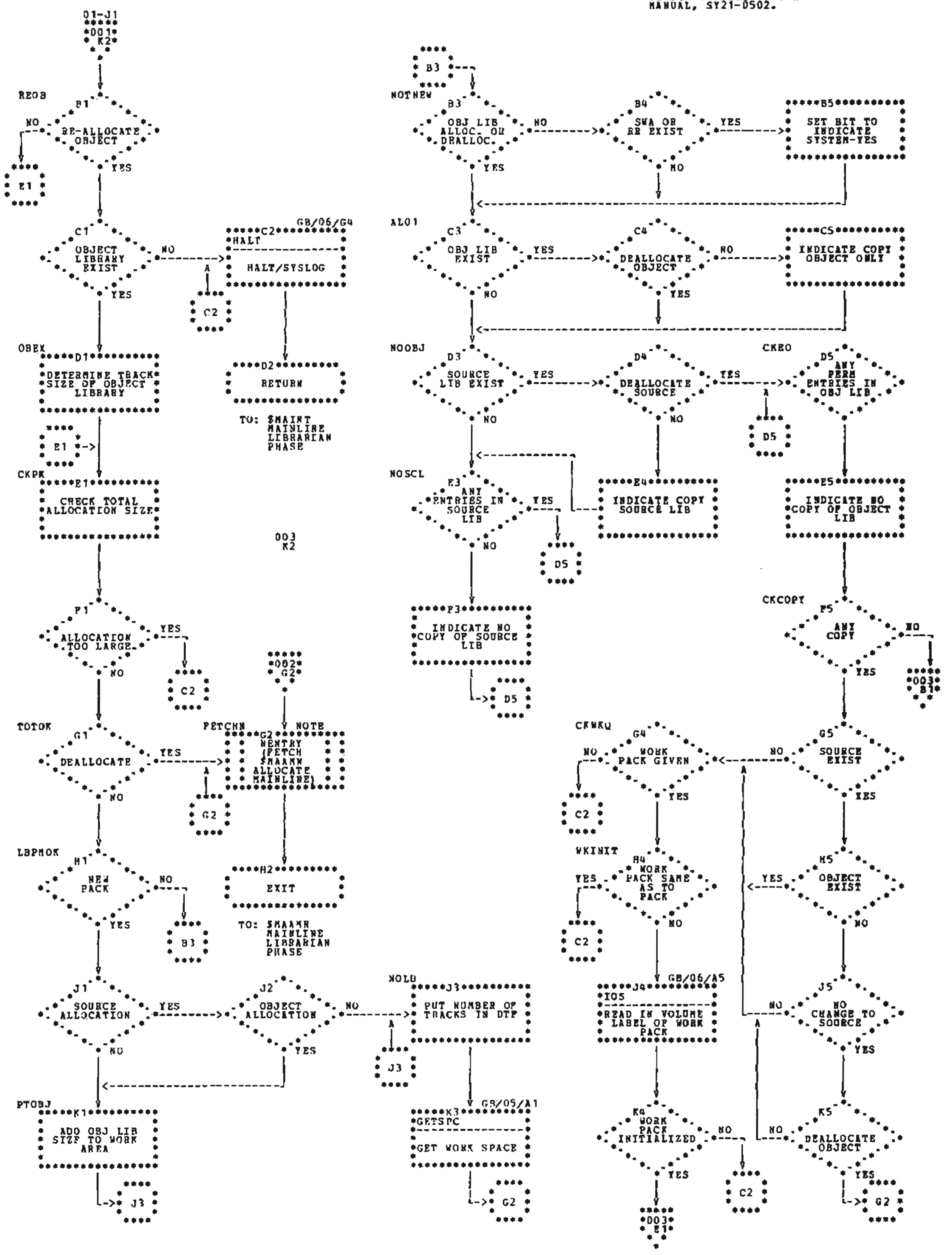


Chart GB (Part 2 of 6). Allocate Syntax Scan Phase

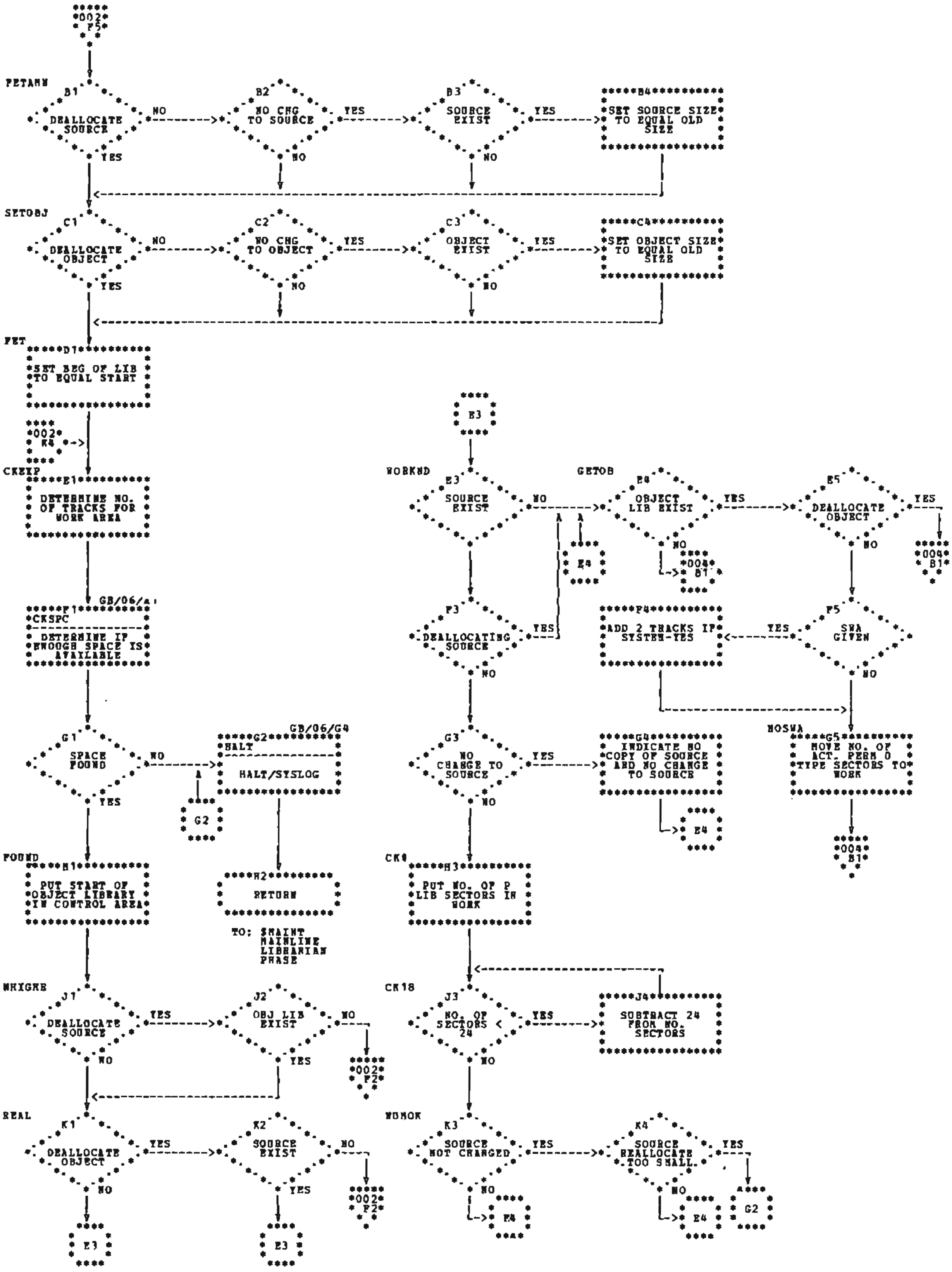


Chart GB (Part 3 of 6). Allocate Syntax Scan Phase

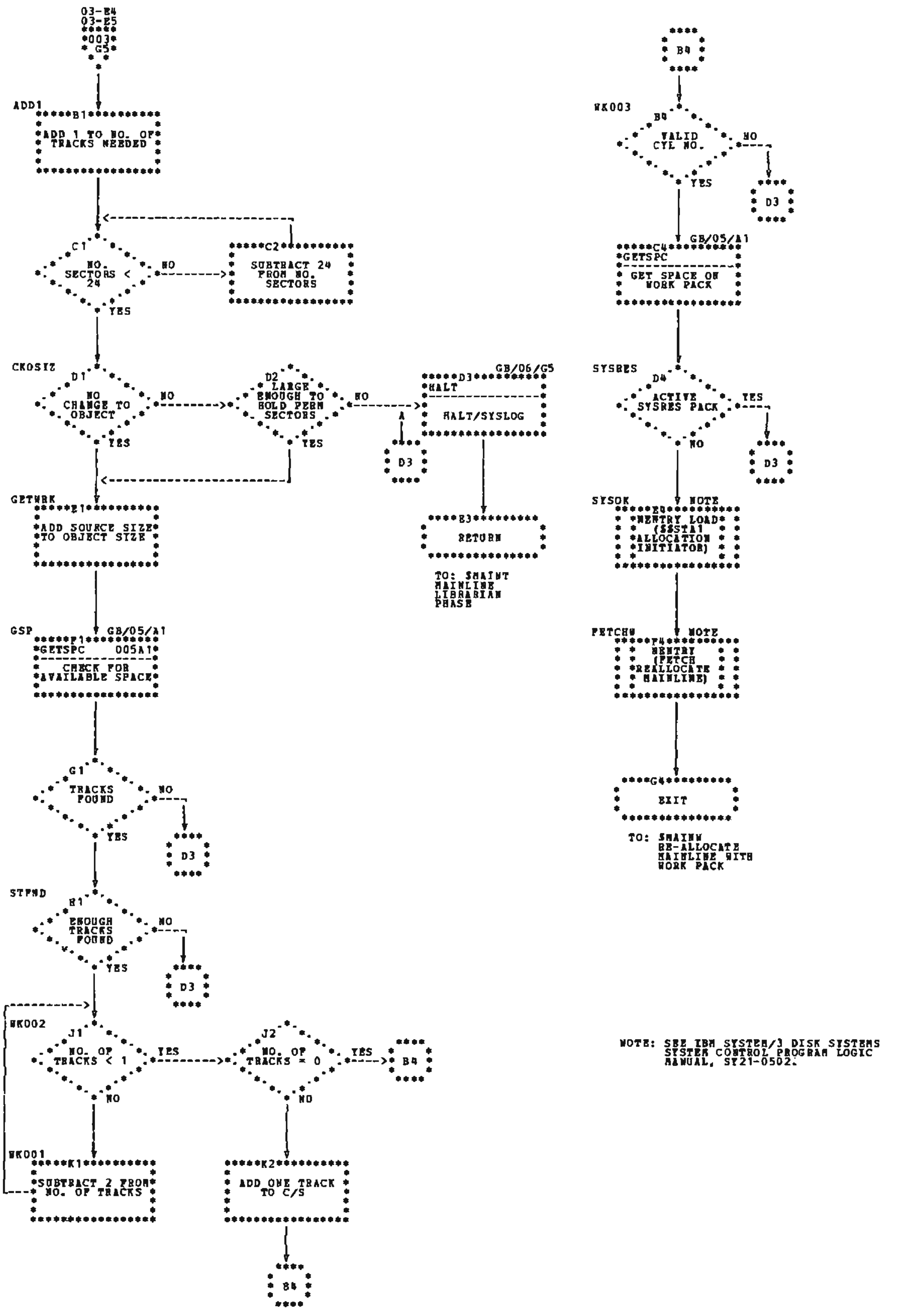


Chart GB (Part 4 of 6). Allocate Syntax Scan Phase

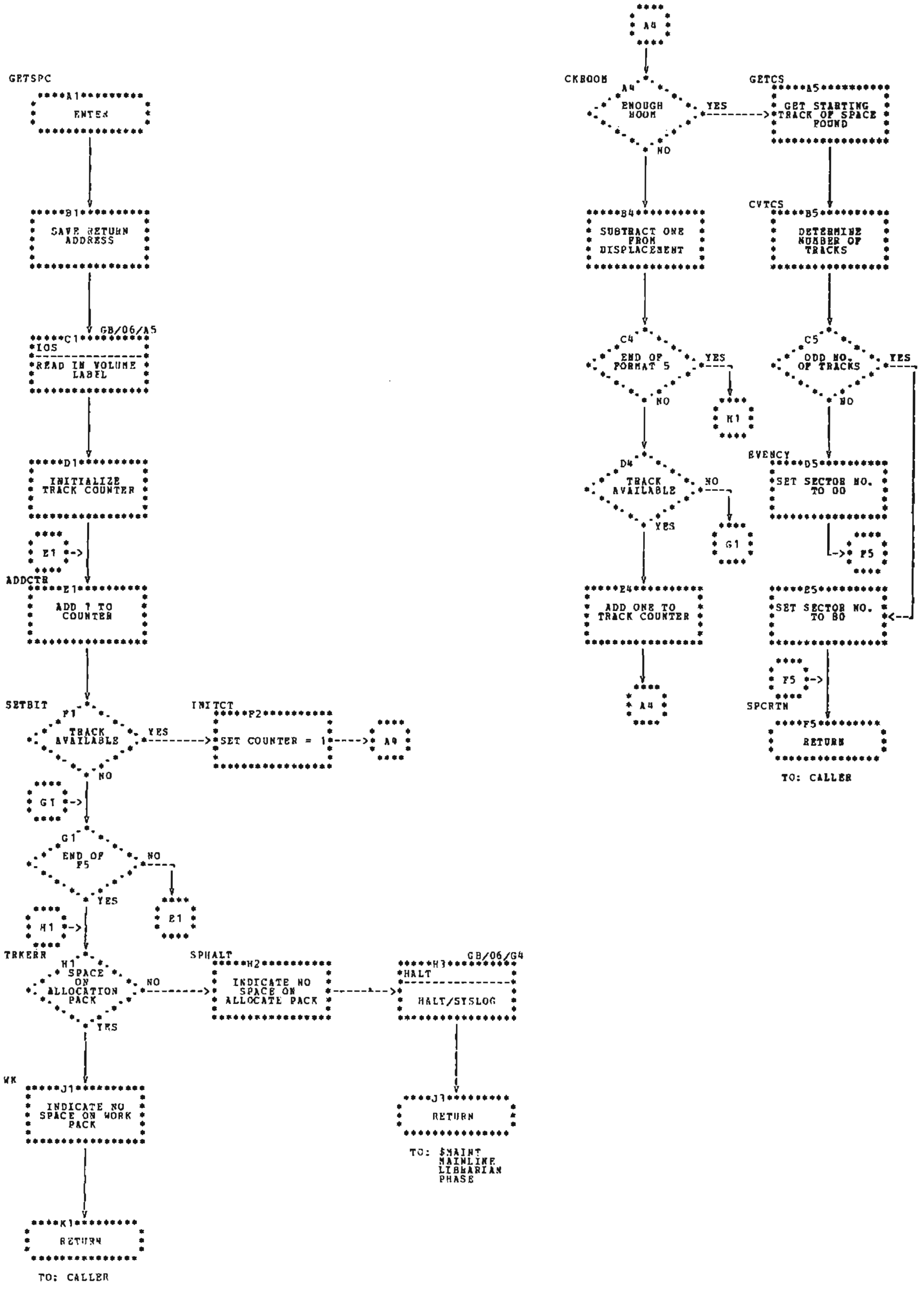
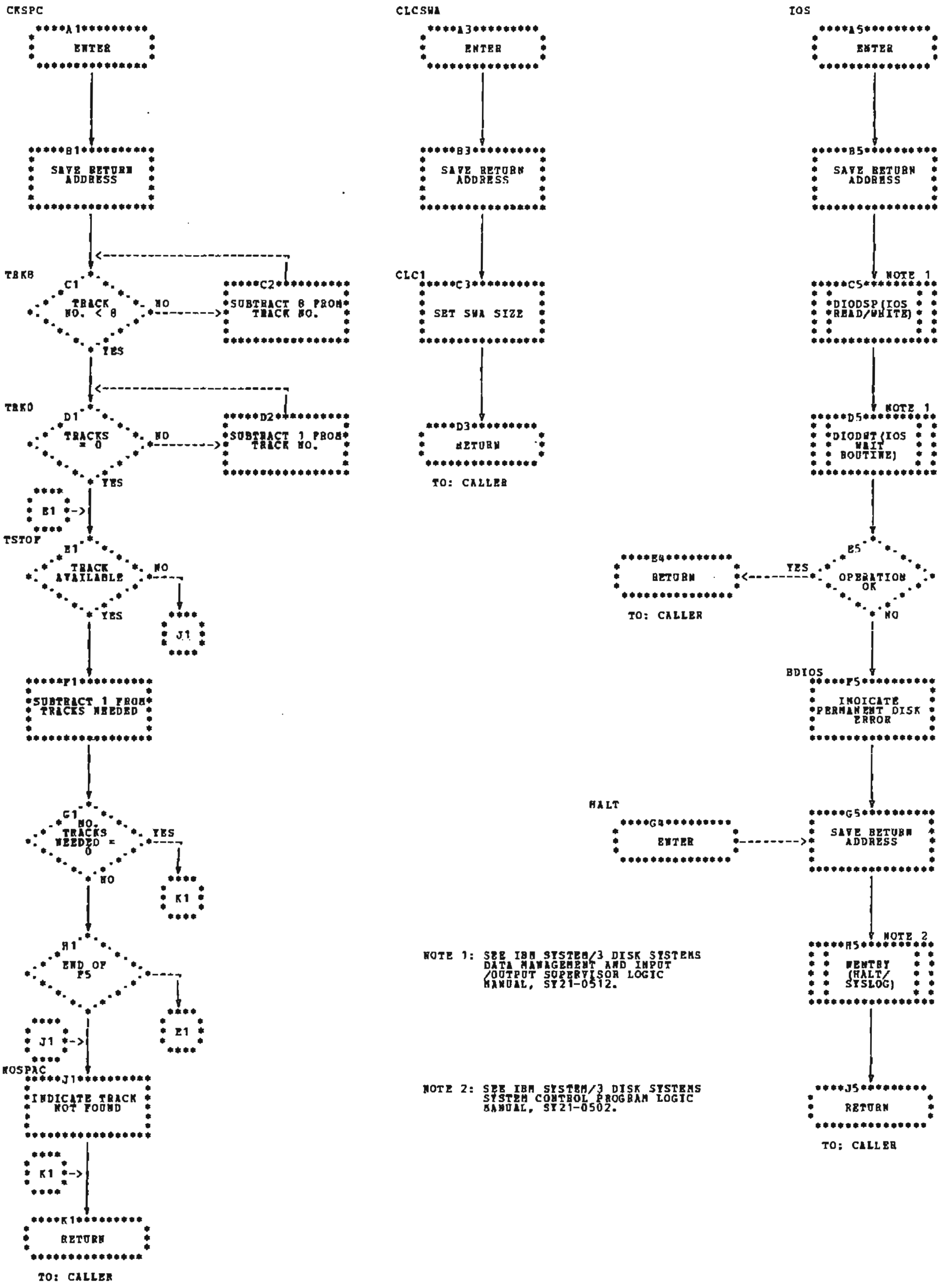


Chart GB (Part 5 of 6). Allocate Syntax Scan Phase



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS DATA MANAGEMENT AND INPUT /OUTPUT SUPERVISOR LOGIC MANUAL, SY21-0512.

NOTE 2: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart GB (Part 6 of 6). Allocate Syntax Scan Phase

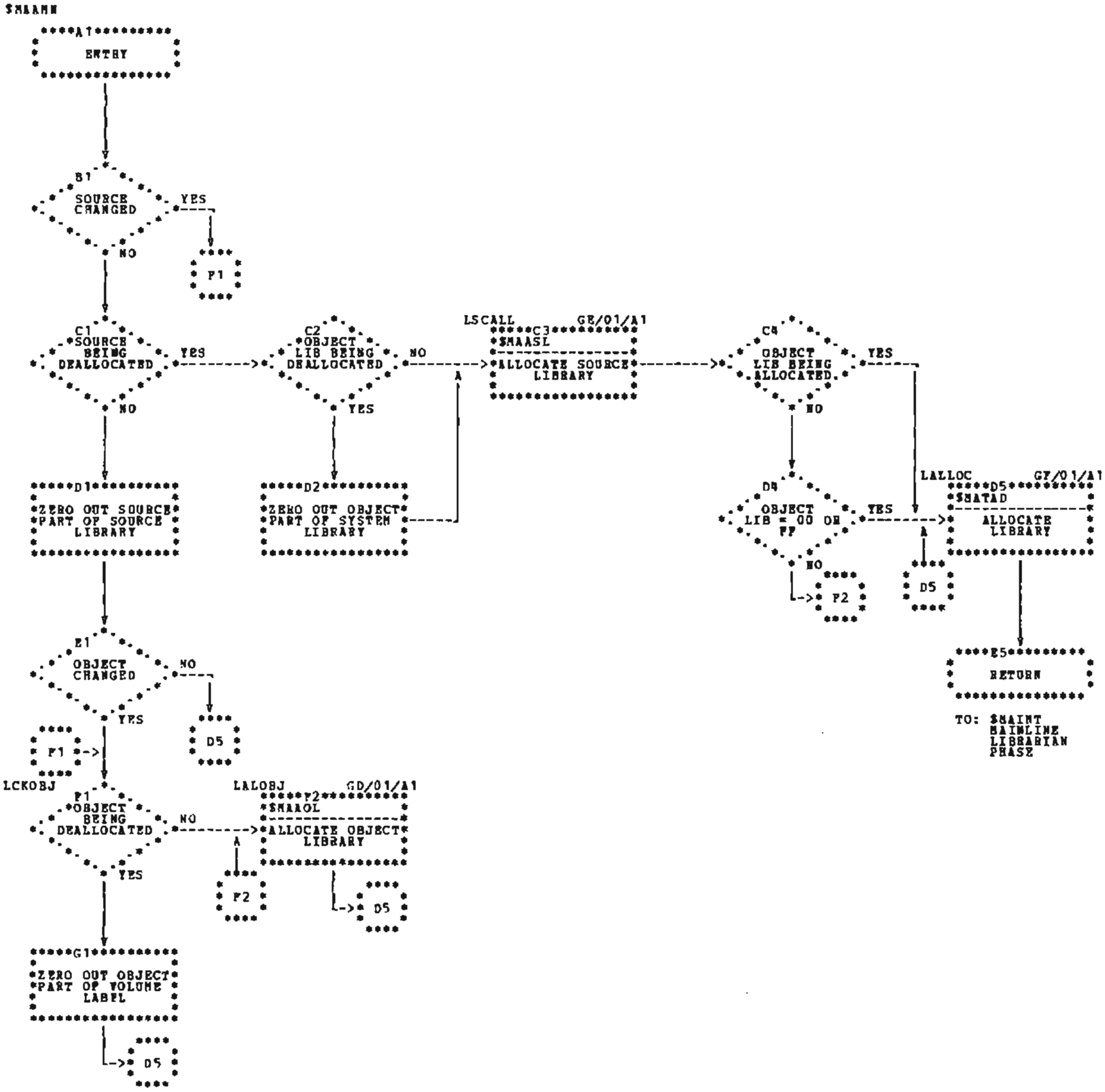


Chart GC. Allocate Mainline Phase

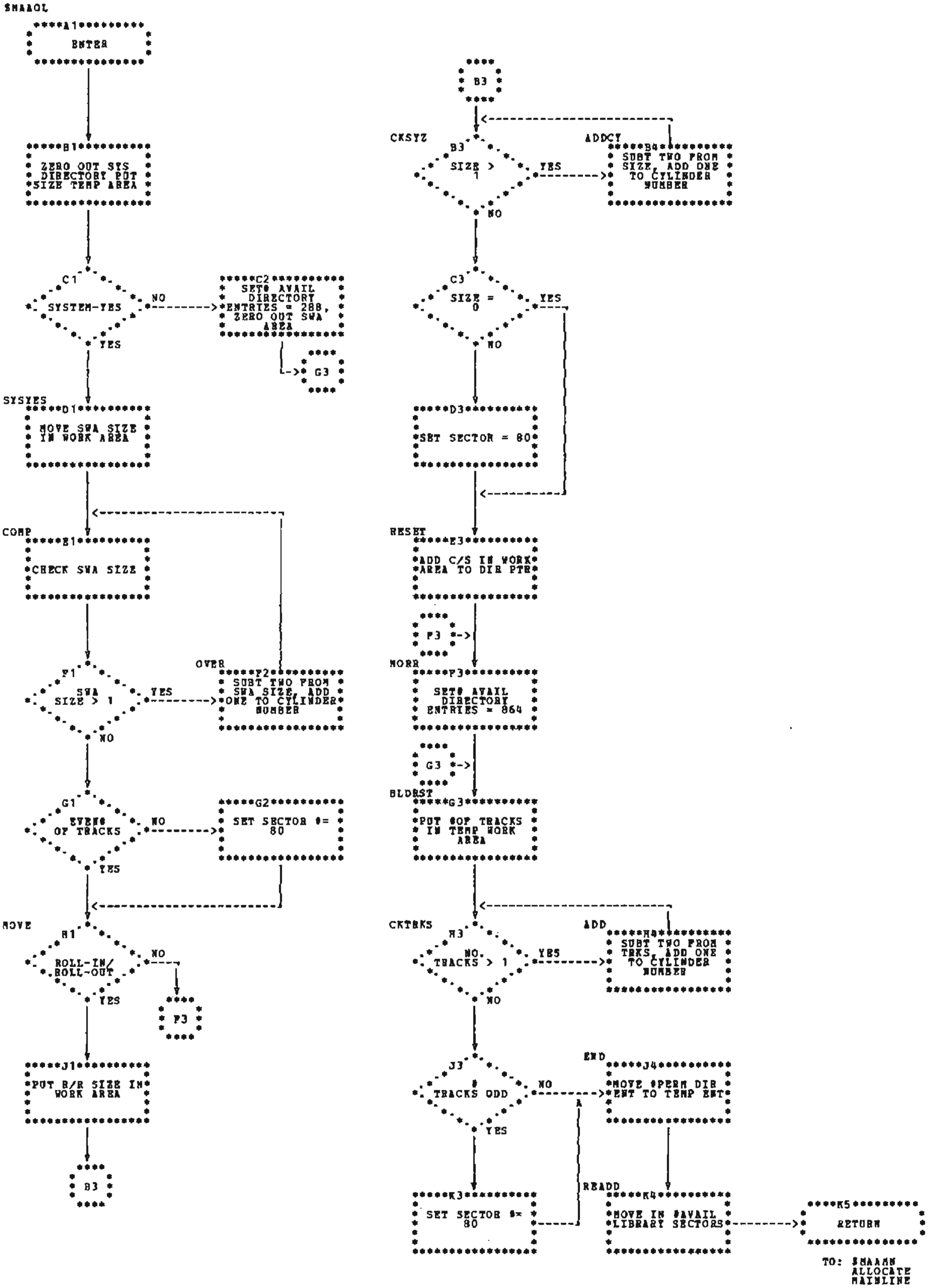

$MAAOL
A1
ENTER
B1
ZERO OUT SYS DIRECTORY PUT SIZE TEMP AREA
C1
SYSTEM-YES
NO
YES
C2
SET# AVAIL DIRECTORY ENTRIES = 288, ZERO OUT SWA AREA
G3
SYSYES
D1
MOVE SWA SIZE IN WORK AREA
COMP
E1
CHECK SWA SIZE
F1
SWA SIZE > 1
YES
NO
OVER
F2
SUBT TWO FROM SWA SIZE, ADD ONE TO CYLINDER NUMBER
G1
EVEN# OF TRACKS
NO
YES
G2
SET SECTOR #= 80
MOVE
H1
ROLL-IN/ ROLL-OUT
NO
YES
F3
J1
PUT R/R SIZE IN WORK AREA
B3
B3
CKSYZ
B3
SIZE > 1
YES
NO
ADDCY
B4
SUBT TWO FROM SIZE, ADD ONE TO CYLINDER NUMBER
C3
SIZE = 0
YES
NO
D3
SET SECTOR = 80
RESET
E3
ADD C/S IN WORK AREA TO DIR PTR
F3
NORR
F3
SET# AVAIL DIRECTORY ENTRIES = 864
G3
BLDRST
G3
PUT #OF TRACKS IN TEMP WORK AREA
CKTRKS
H3
NO. TRACKS > 1
YES
NO
ADD
H4
SUBT TWO FROM TRKS, ADD ONE TO CYLINDER NUMBER
J3
# TRACKS ODD
NO
YES
END
J4
MOVE #PERM DIR ENT TO TEMP ENT
A
K3
SET SECTOR #= 80
READD
K4
MOVE IN #AVAIL LIBRARY SECTORS
K5
RETURN
TO: $MAAMN ALLOCATE MAINLINE

Chart GD. Allocate Object Library Phase

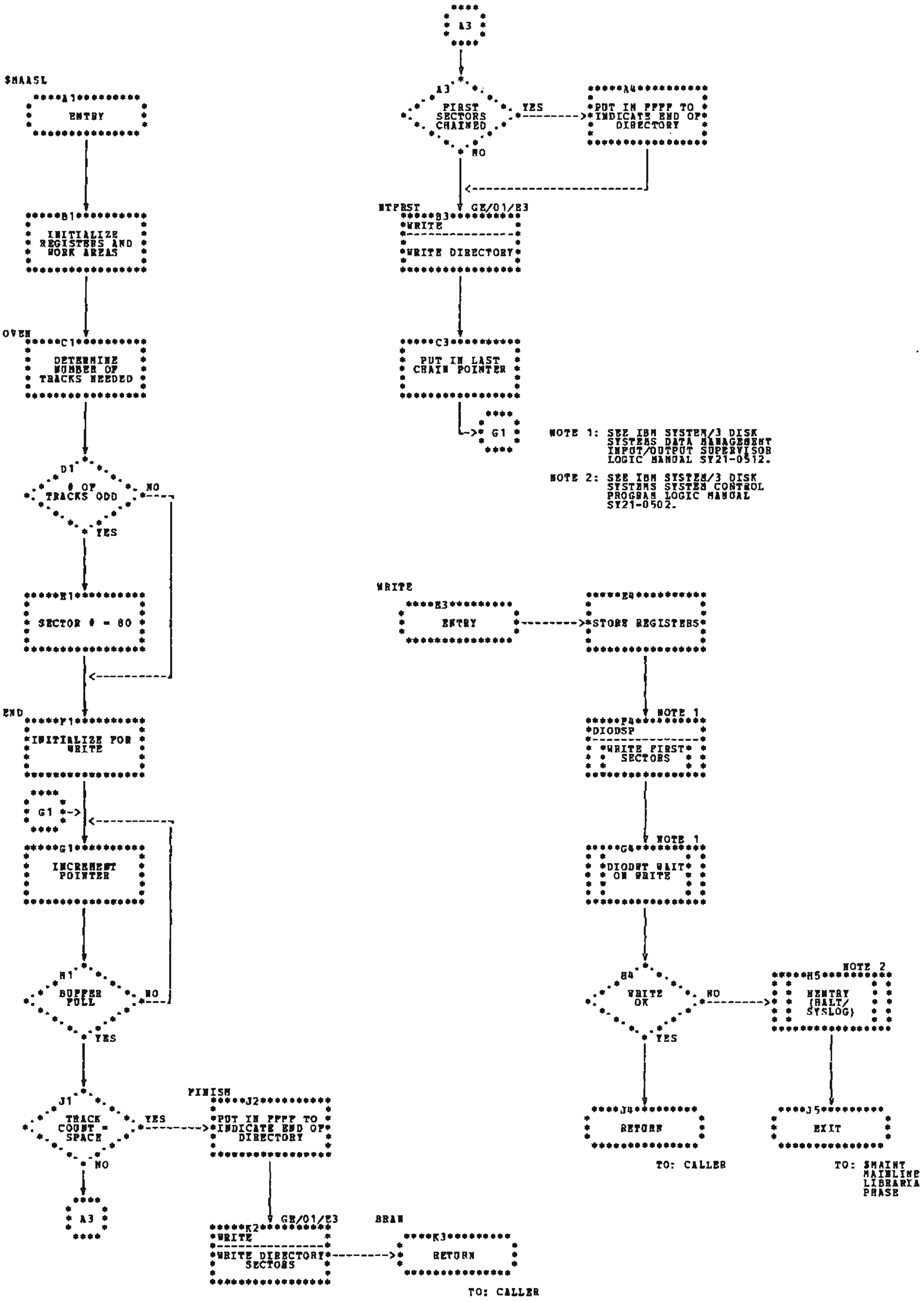


Chart GE. Allocate Source Library Phase

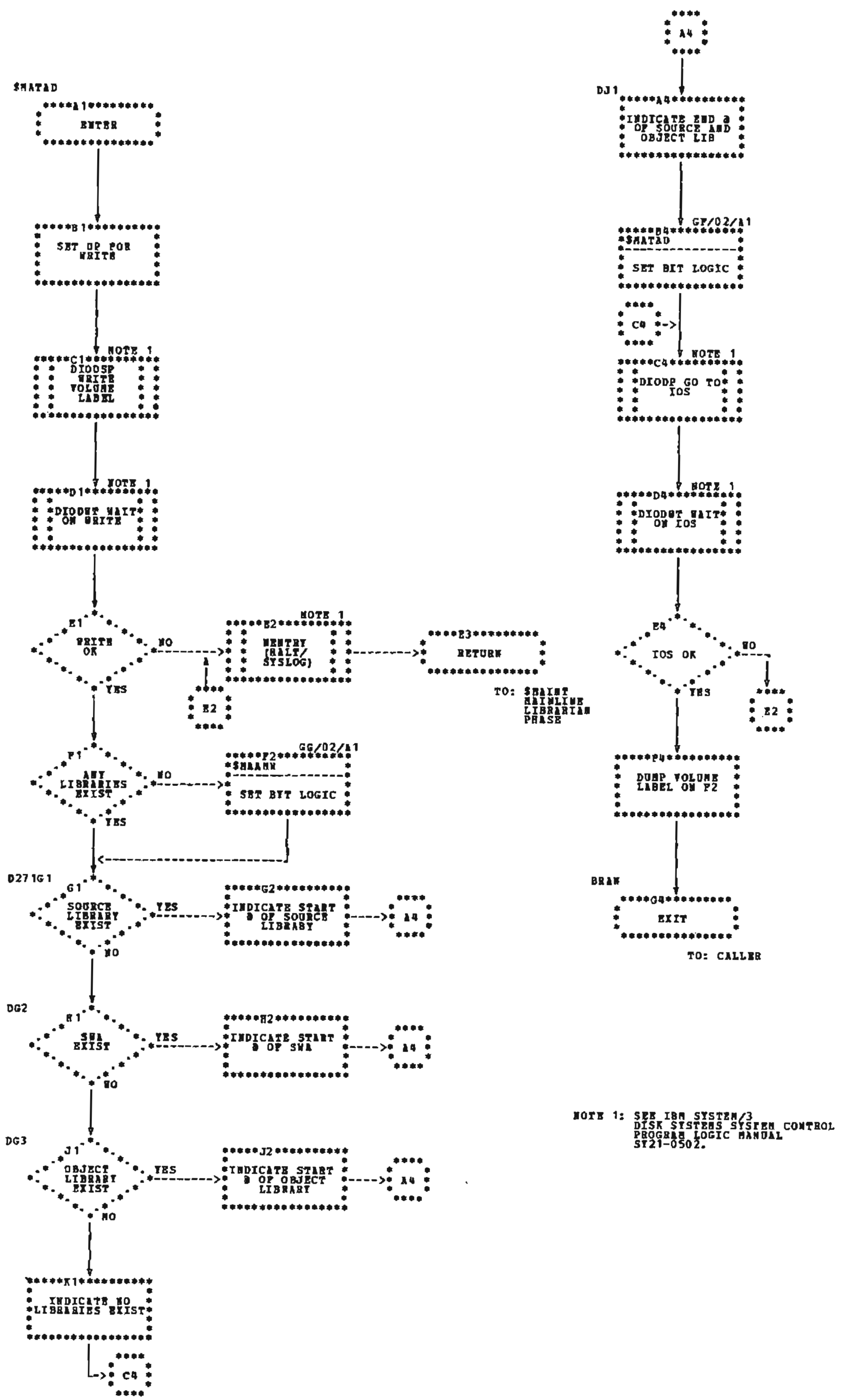


Chart GF (Part 1 of 2). Track Allocate/Deallocate Phase

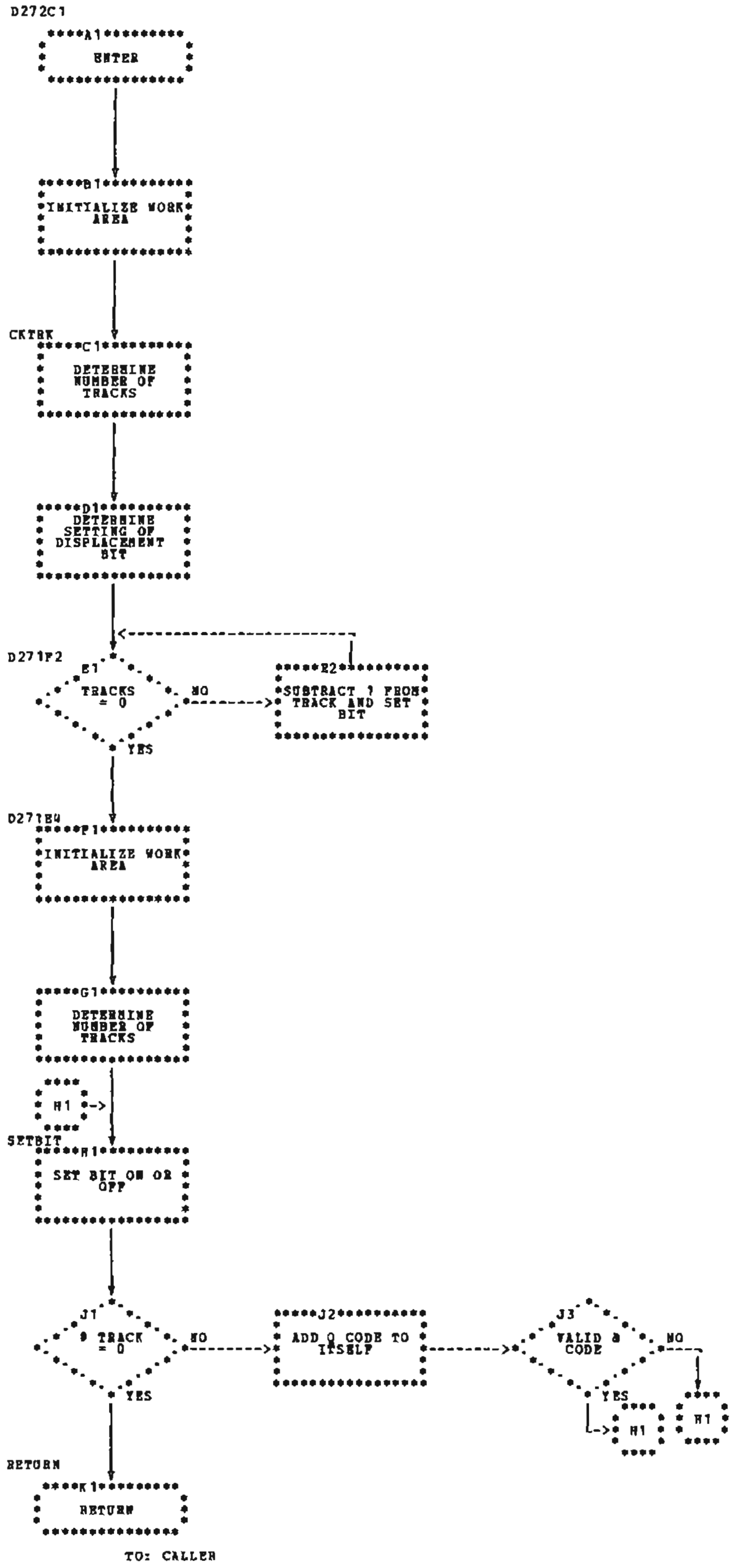


Chart GF (Part 2 of 2). Track Allocate/Deallocate Phase

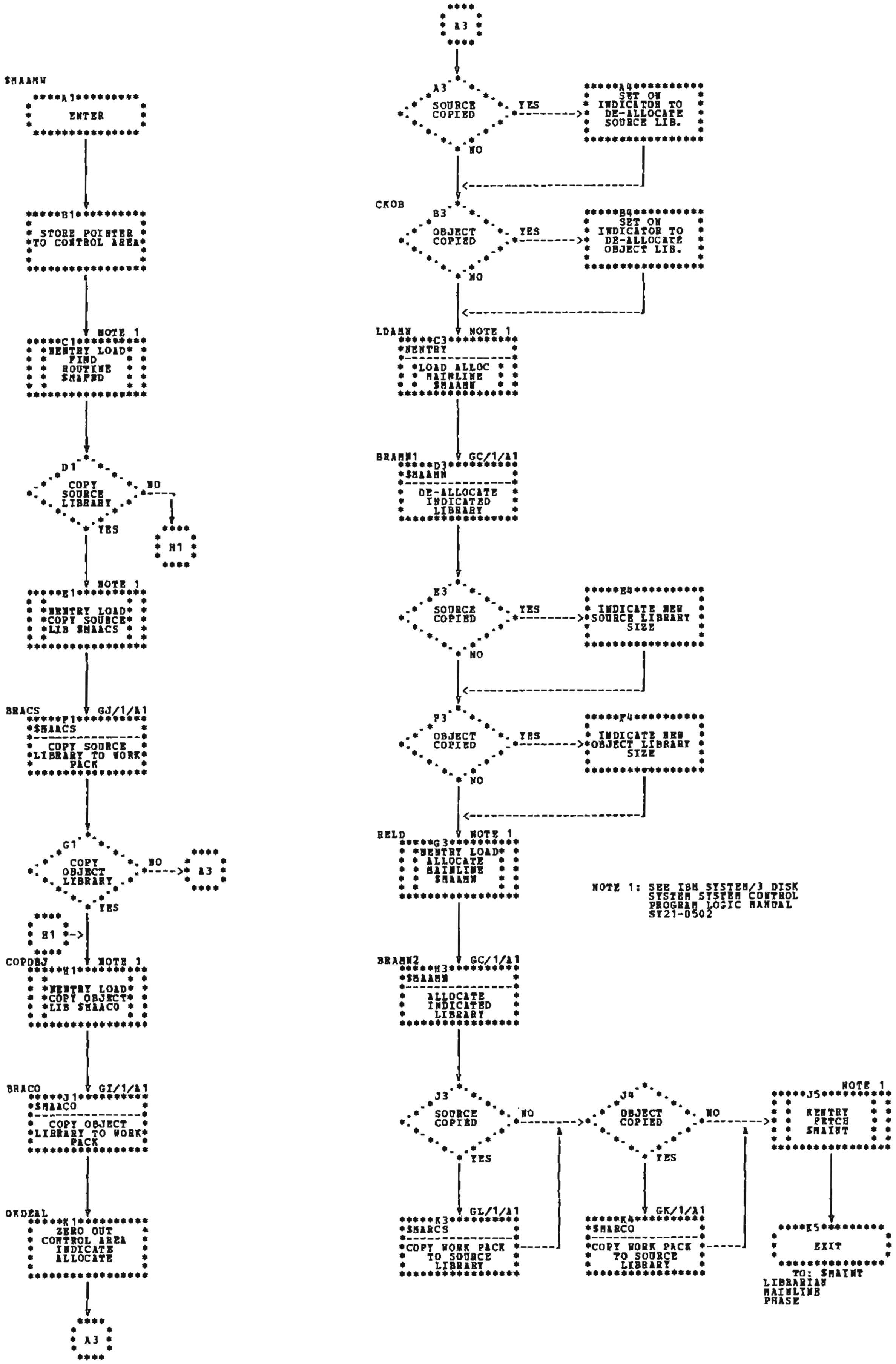
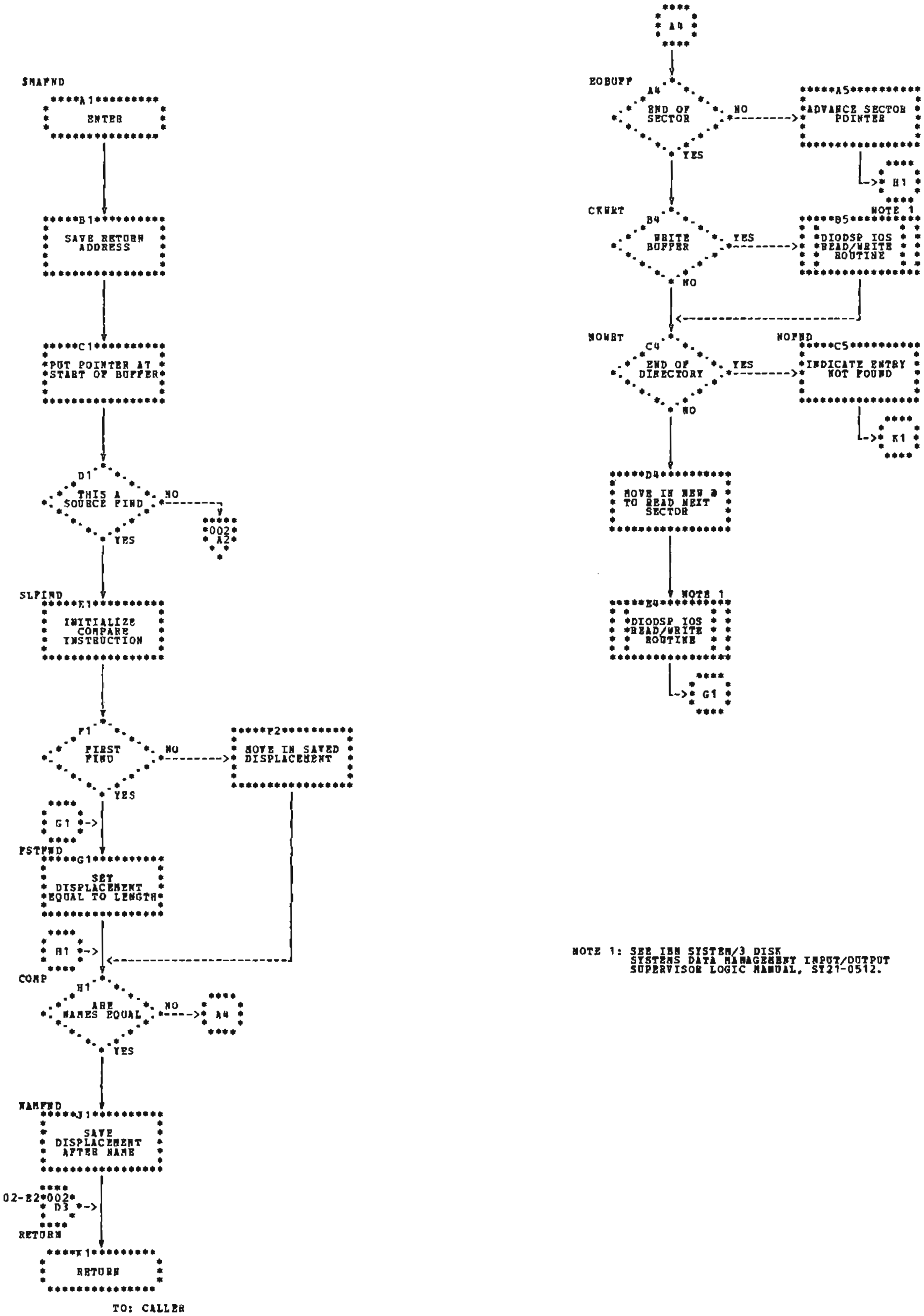
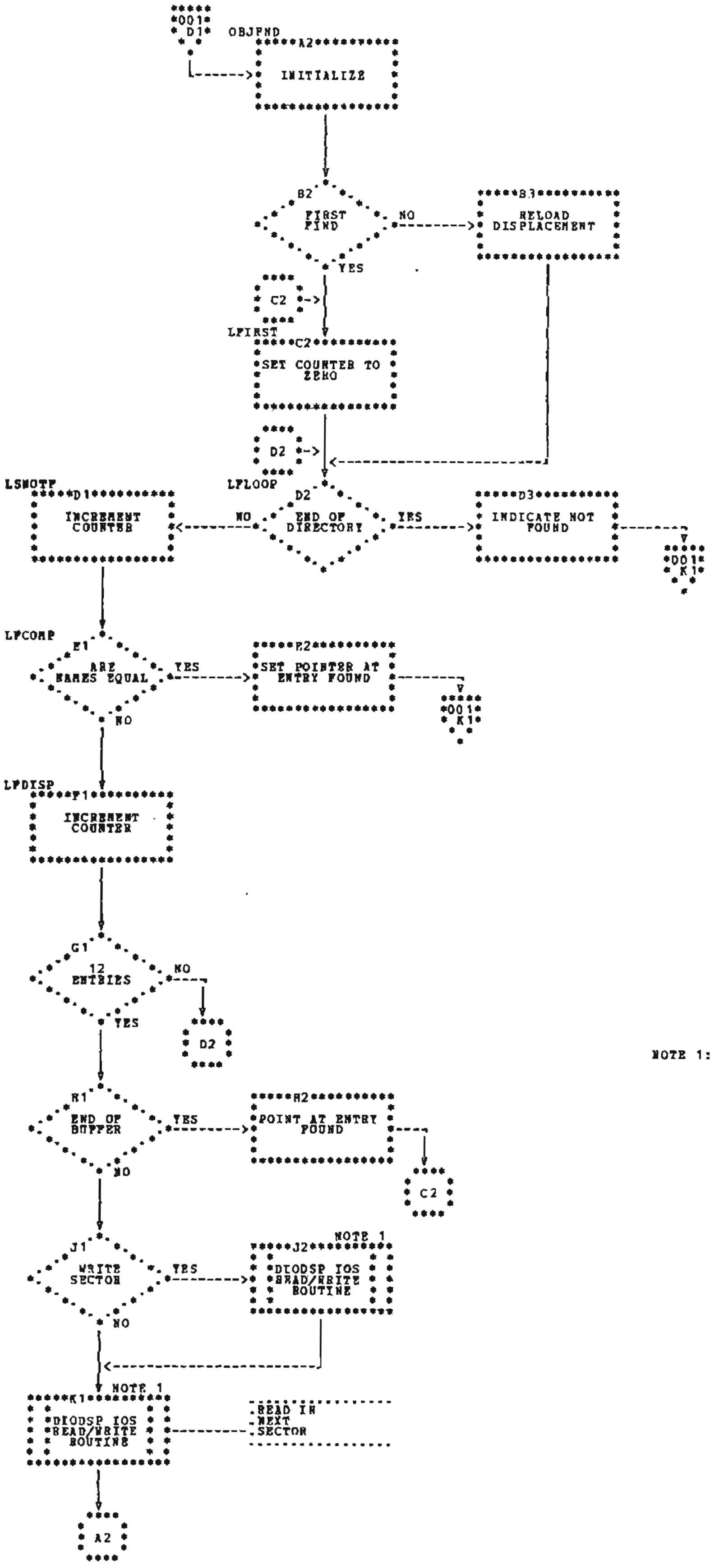

$MAAMN
A1
ENTER
B1
STORE POINTER TO CONTROL AREA
NOTE 1
C1
$ENTRY LOAD FIND ROUTINE $MAFND
D1
COPY SOURCE LIBRARY
NO
YES
H1
NOTE 1
E1
$ENTRY LOAD COPY SOURCE LIB $MAACS
BRACS
GJ/1/A1
F1
$MAACS
COPY SOURCE LIBRARY TO WORK PACK
G1
COPY OBJECT LIBRARY
NO
A3
YES
H1
COPOBJ
NOTE 1
H1
$ENTRY LOAD COPY OBJECT LIB $MAACO
BRACO
GI/1/A1
J1
$MAACO
COPY OBJECT LIBRARY TO WORK PACK
OKDEAL
K1
ZERO OUT CONTROL AREA INDICATE ALLOCATE
A3
A3
A3
SOURCE COPIED
YES
NO
A4
SET ON INDICATOR TO DE-ALLOCATE SOURCE LIB.
CKOB
B3
OBJECT COPIED
YES
NO
B4
SET ON INDICATOR TO DE-ALLOCATE OBJECT LIB.
LDAMN
NOTE 1
C3
$ENTRY
LOAD ALLOC MAINLINE $MAAMN
BRAMN1
GC/1/A1
D3
$MAAMN
DE-ALLOCATE INDICATED LIBRARY
E3
SOURCE COPIED
YES
NO
E4
INDICATE NEW SOURCE LIBRARY SIZE
F3
OBJECT COPIED
YES
NO
F4
INDICATE NEW OBJECT LIBRARY SIZE
RELD
NOTE 1
G3
$ENTRY LOAD ALLOCATE MAINLINE $MAAMN
NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEM SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502
BRAMN2
GC/1/A1
H3
$MAAMN
ALLOCATE INDICATED LIBRARY
J3
SOURCE COPIED
NO
YES
J4
OBJECT COPIED
NO
YES
J5
NOTE 1
$ENTRY FETCH $MAINT
K3
GL/1/A1
$MARCS
COPY WORK PACK TO SOURCE LIBRARY
K4
GK/1/A1
$MARCO
COPY WORK PACK TO SOURCE LIBRARY
K5
EXIT
TO: $MAINT LIBRARIAN MAINLINE PHASE

Chart GG. Reallocate Mainline Phase



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS DATA MANAGEMENT INPUT/OUTPUT SUPERVISOR LOGIC MANUAL, SY21-0512.

Chart GH (Part 1 of 2). Find Phase



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS DATA MANAGEMENT INPUT/OUTPUT SUPERVISOR LOGIC MANUAL, SY21-0512.

Chart GH (Part 2 of 2). Find Phase

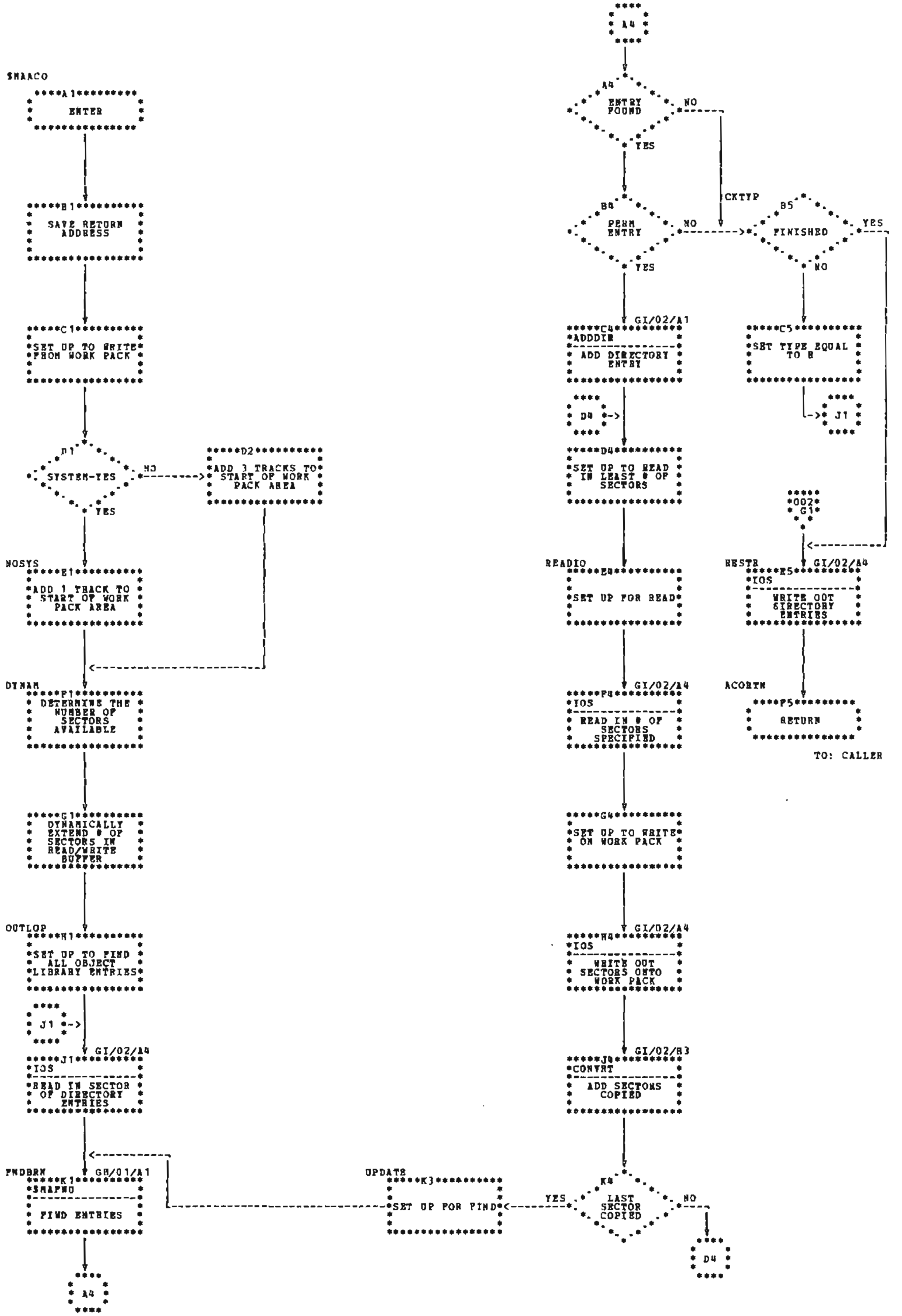
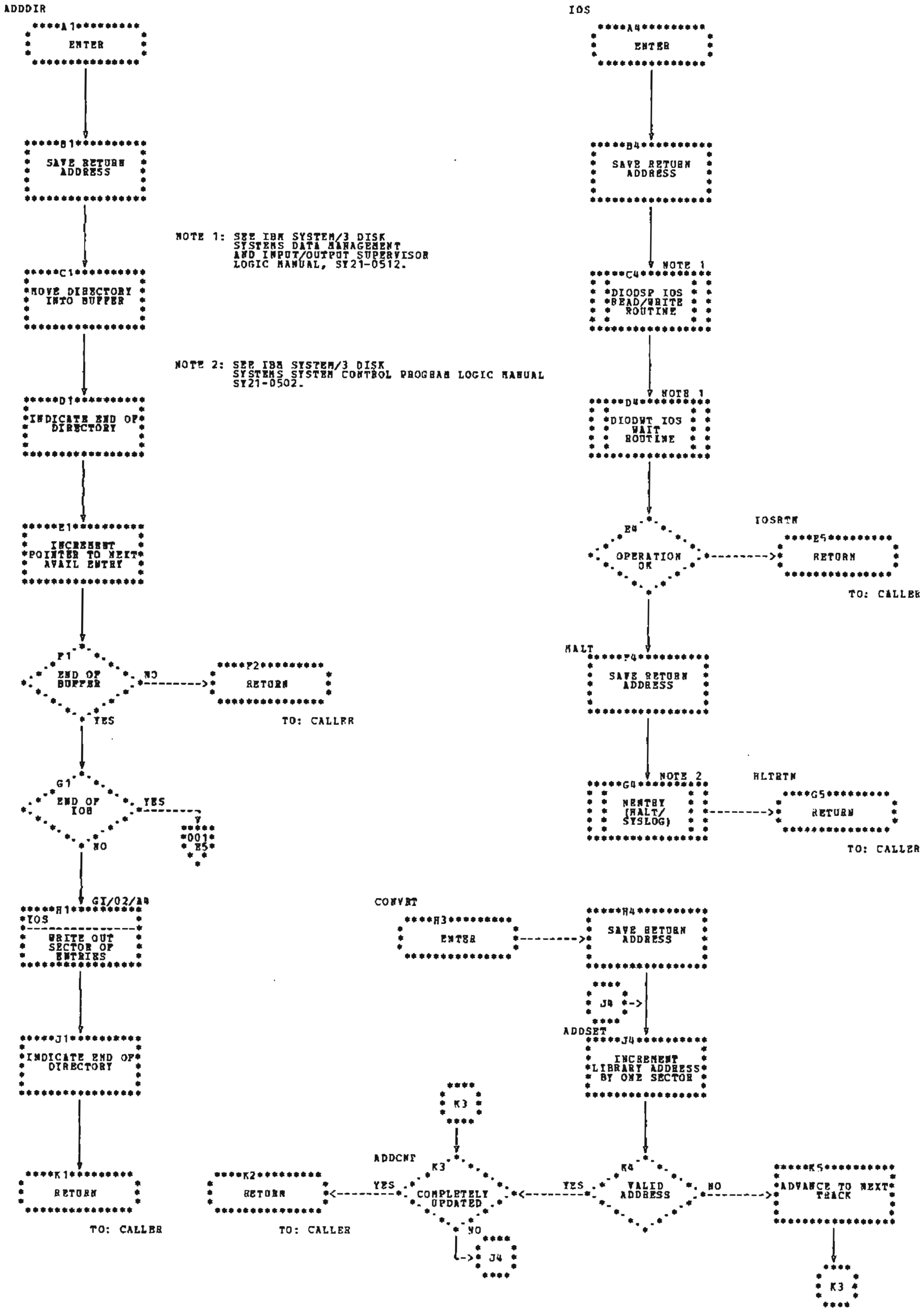


Chart GI (Part 1 of 2). Copy Object Library Phase



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS DATA MANAGEMENT AND INPUT/OUTPUT SUPERVISOR LOGIC MANUAL, SY21-0512.

NOTE 2: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502.

Chart GI (Part 2 of 2). Copy Object Library Phase

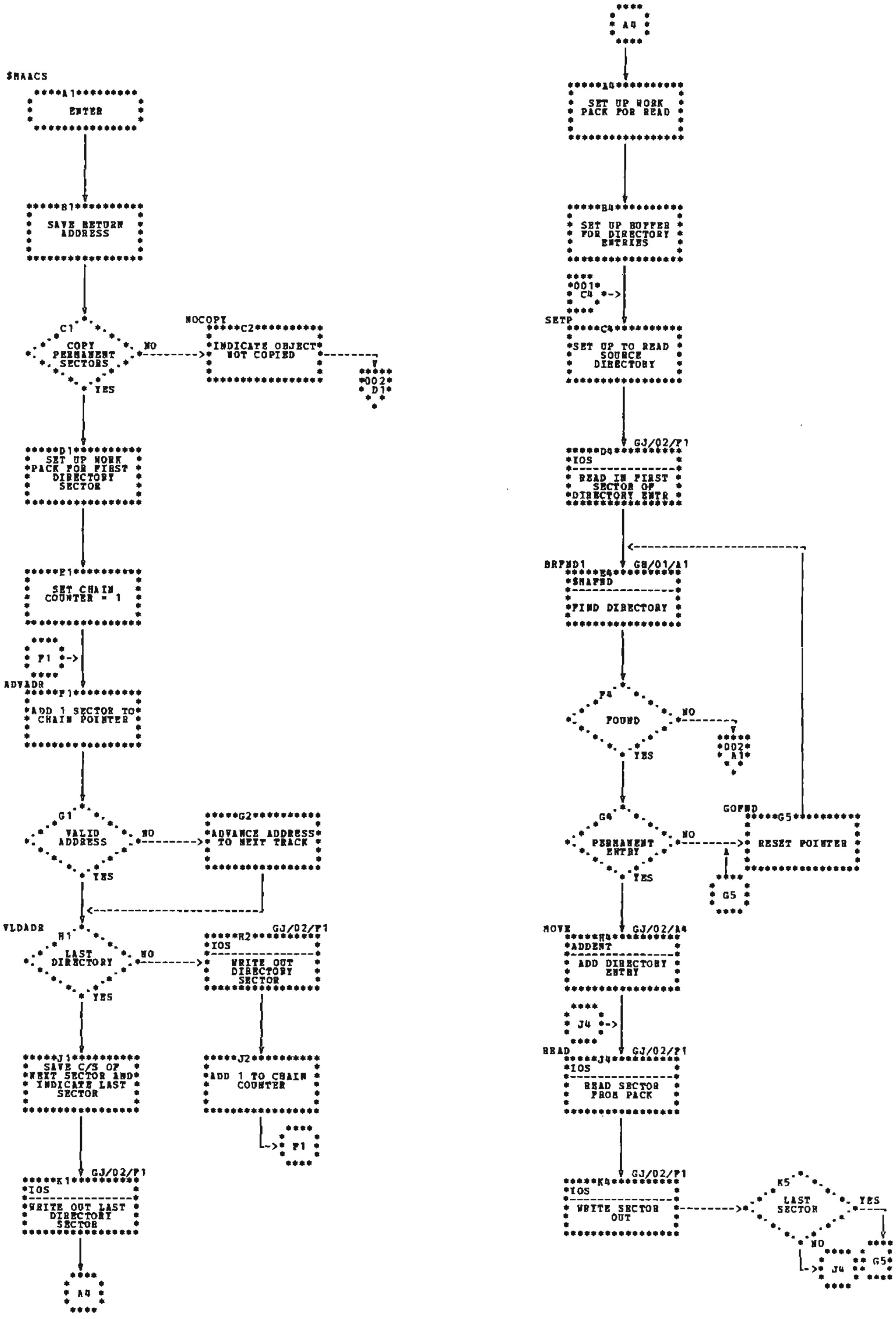
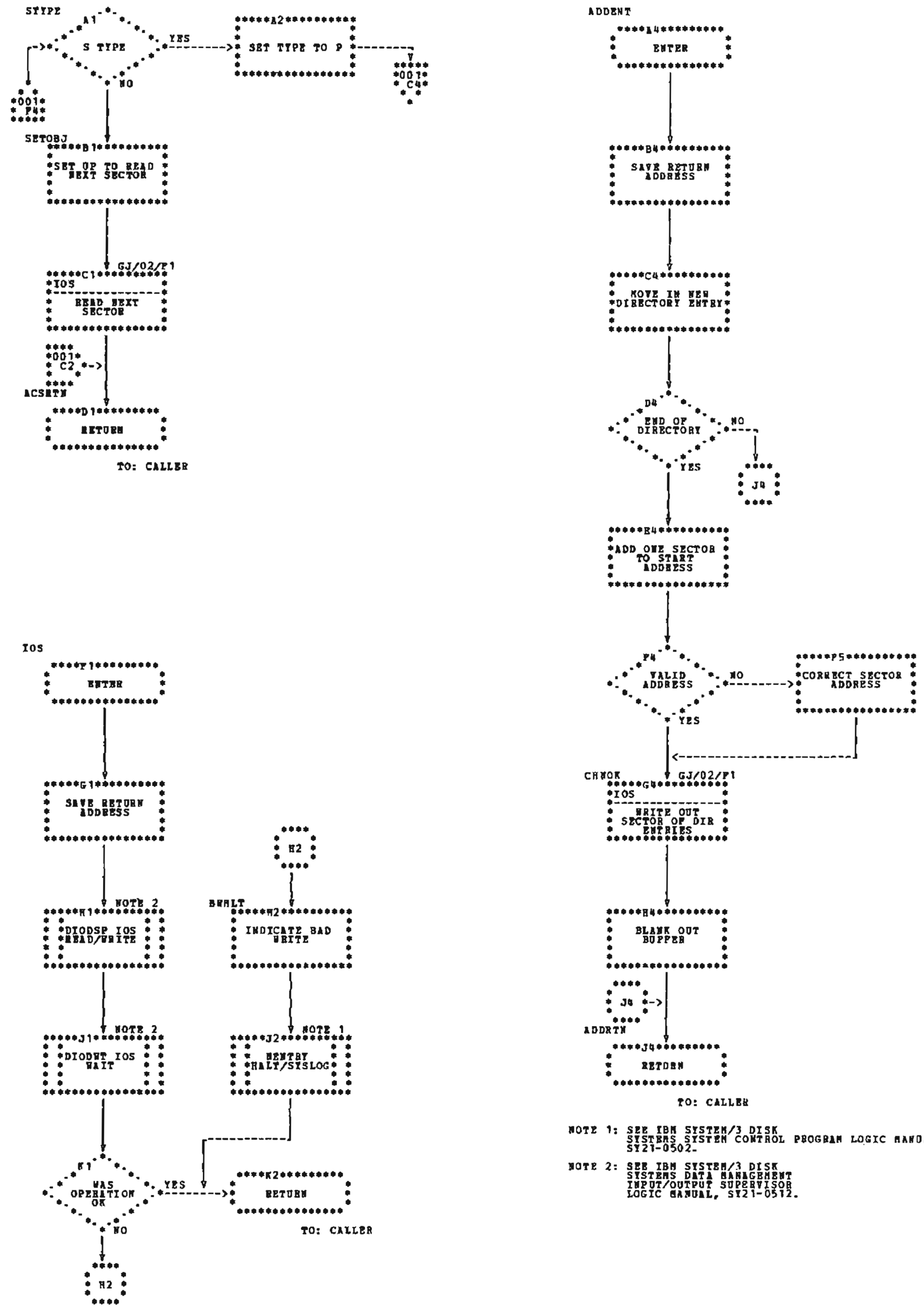


Chart GJ (Part 1 of 2). Copy Source Library Phase



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

NOTE 2: SEE IBM SYSTEM/3 DISK SYSTEMS DATA MANAGEMENT INPUT/OUTPUT SUPERVISOR LOGIC MANUAL, SY21-0512.

Chart GJ (Part 2 of 2). Copy Source Library Phase

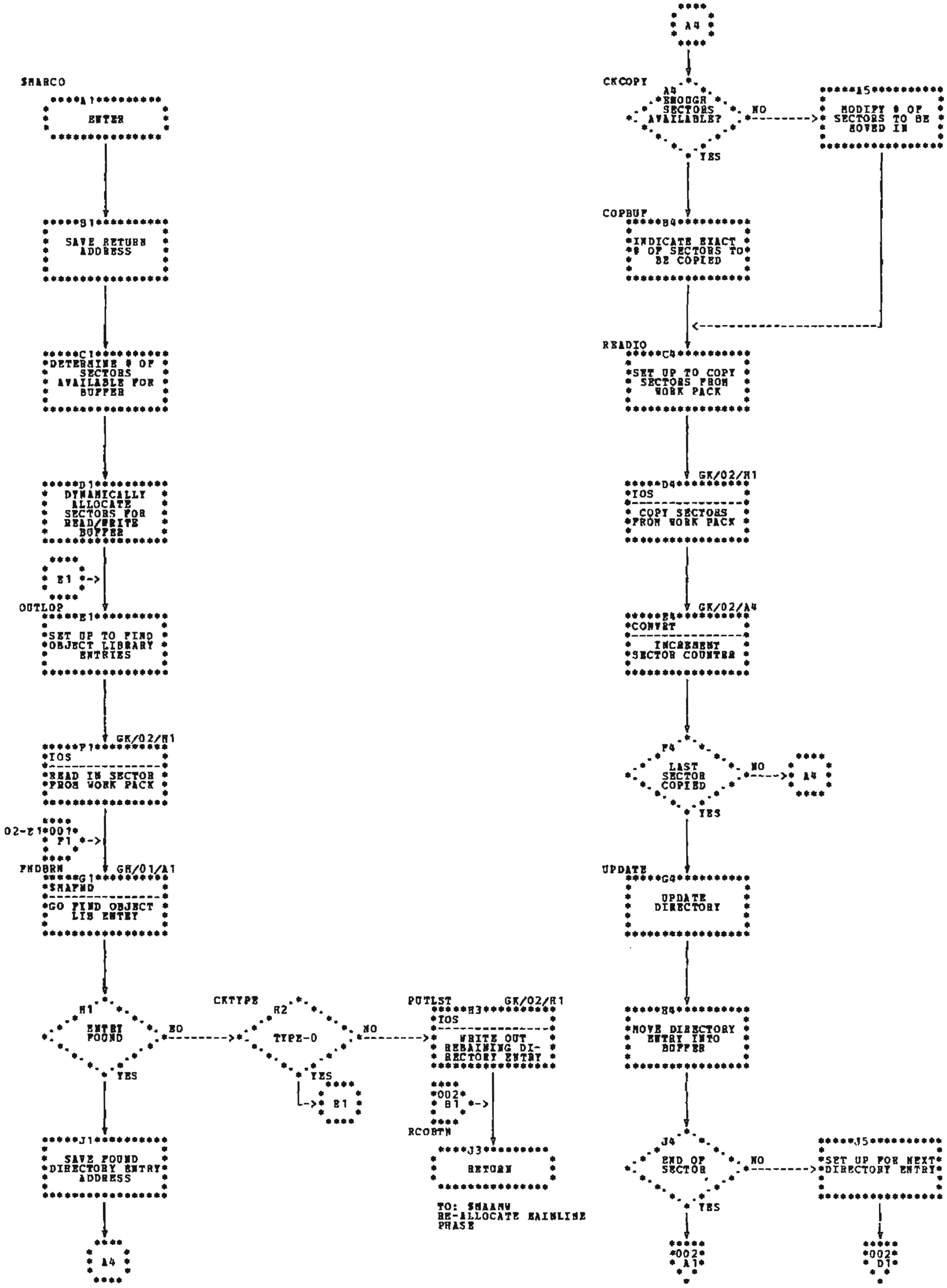
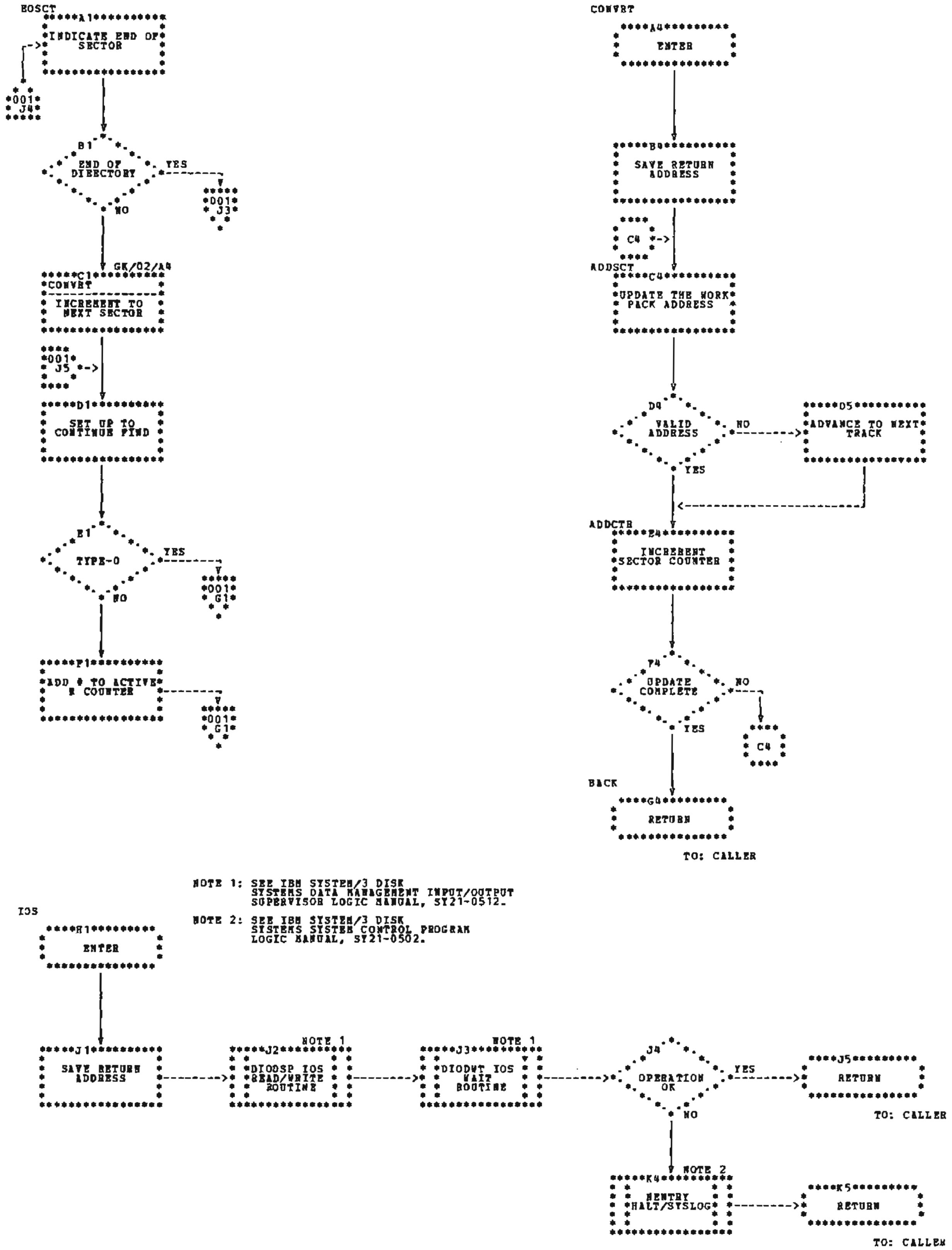

SHARCO
A1
ENTER
B1
SAVE RETURN ADDRESS
C1
DETERMINE # OF SECTORS AVAILABLE FOR BUFFER
D1
DYNAMICALLY ALLOCATE SECTORS FOR READ/WRITE BUFFER
E1 ->
OUTLOP
E1
SET UP TO FIND OBJECT LIBRARY ENTRIES
GK/02/H1
F1
IOS
READ IN SECTOR FROM WORK PACK
02-E1 *001* F1 ->
FNDBRN
GH/01/A1
G1
SHAFND
GO FIND OBJECT LIB ENTRY
H1
ENTRY FOUND
NO
YES
CKTYPE
H2
TYPE-0
NO
YES
E1
PUTLST
GK/02/H1
H3
IOS
WRITE OUT REMAINING DIRECTORY ENTRY
002 B1 ->
RCOBTN
J3
RETURN
TO: SHAAMW RE-ALLOCATE MAINLINE PHASE
J1
SAVE FOUND DIRECTORY ENTRY ADDRESS
A4
A4
CKCOPY
A4
ENOUGH SECTORS AVAILABLE?
NO
YES
A5
MODIFY # OF SECTORS TO BE MOVED IN
COPBUF
B4
INDICATE EXACT # OF SECTORS TO BE COPIED
READIO
C4
SET UP TO COPY SECTORS FROM WORK PACK
GK/02/H1
D4
IOS
COPY SECTORS FROM WORK PACK
GK/02/A4
E4
CONVRT
INCREMENT SECTOR COUNTER
F4
LAST SECTOR COPIED
NO
A4
YES
UPDATE
G4
UPDATE DIRECTORY
H4
MOVE DIRECTORY ENTRY INTO BUFFER
J4
END OF SECTOR
NO
YES
J5
SET UP FOR NEXT DIRECTORY ENTRY
002 A1
002 D1

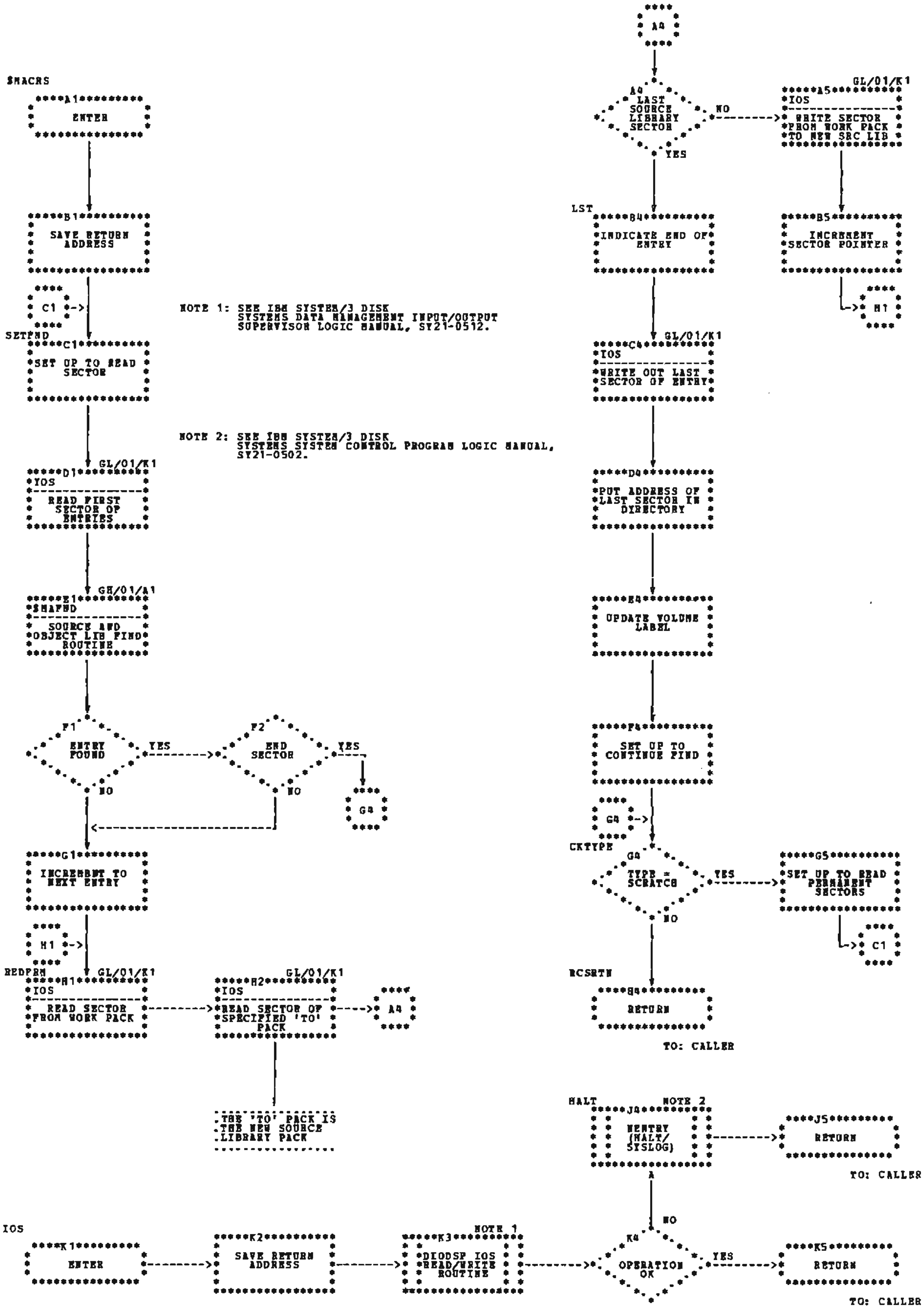
Chart GK (Part 1 of 2). Copy Work Pack to Object Library



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS DATA MANAGEMENT INPUT/OUTPUT SUPERVISOR LOGIC MANUAL, SY21-0512.

NOTE 2: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart GK (Part 2 of 2). Copy Work Pack to Object Library



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS DATA MANAGEMENT INPUT/OUTPUT SUPERVISOR LOGIC MANUAL, SY21-0512.

NOTE 2: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart GL. Copy Work Pack to Source Library Phase

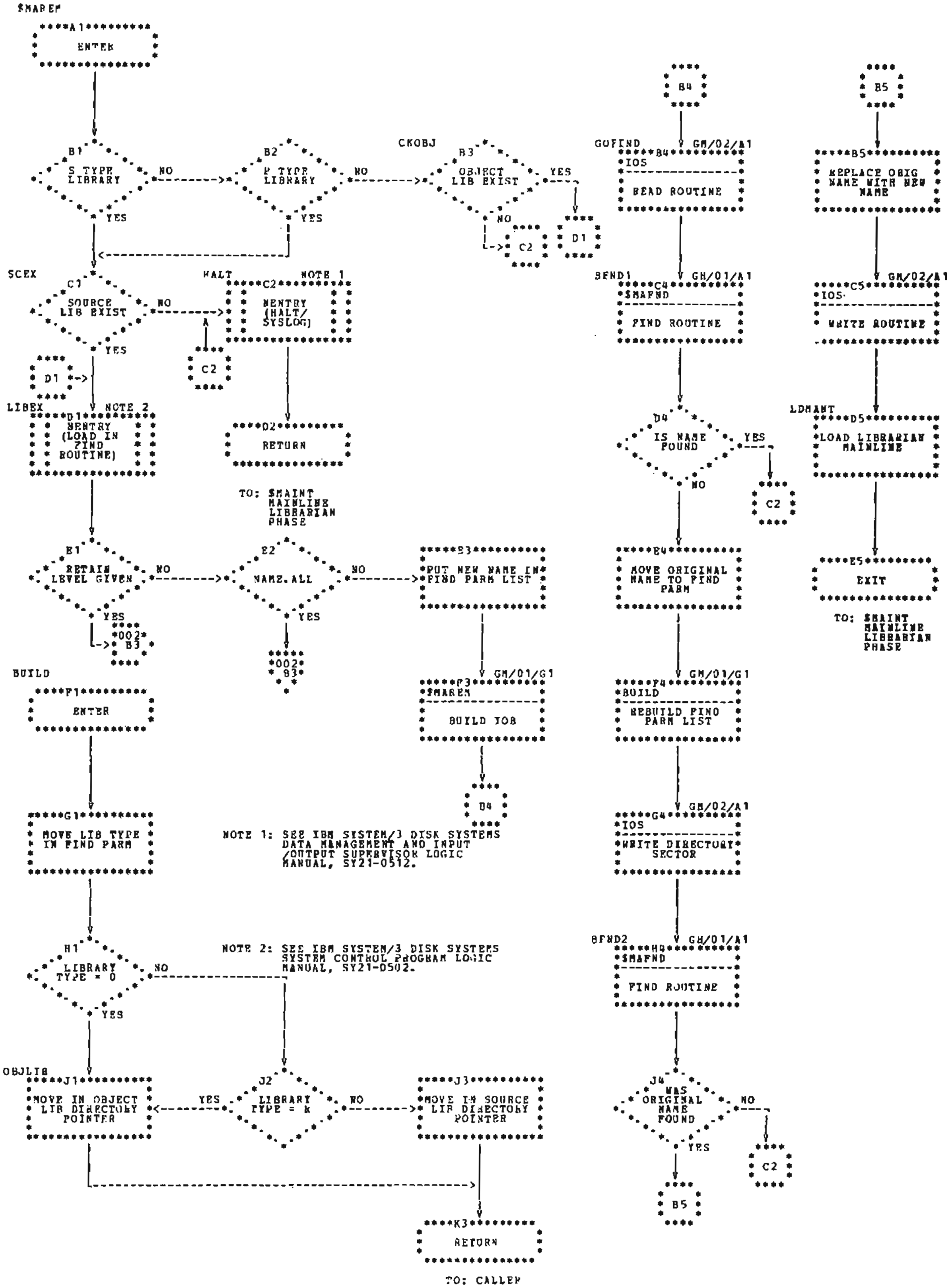
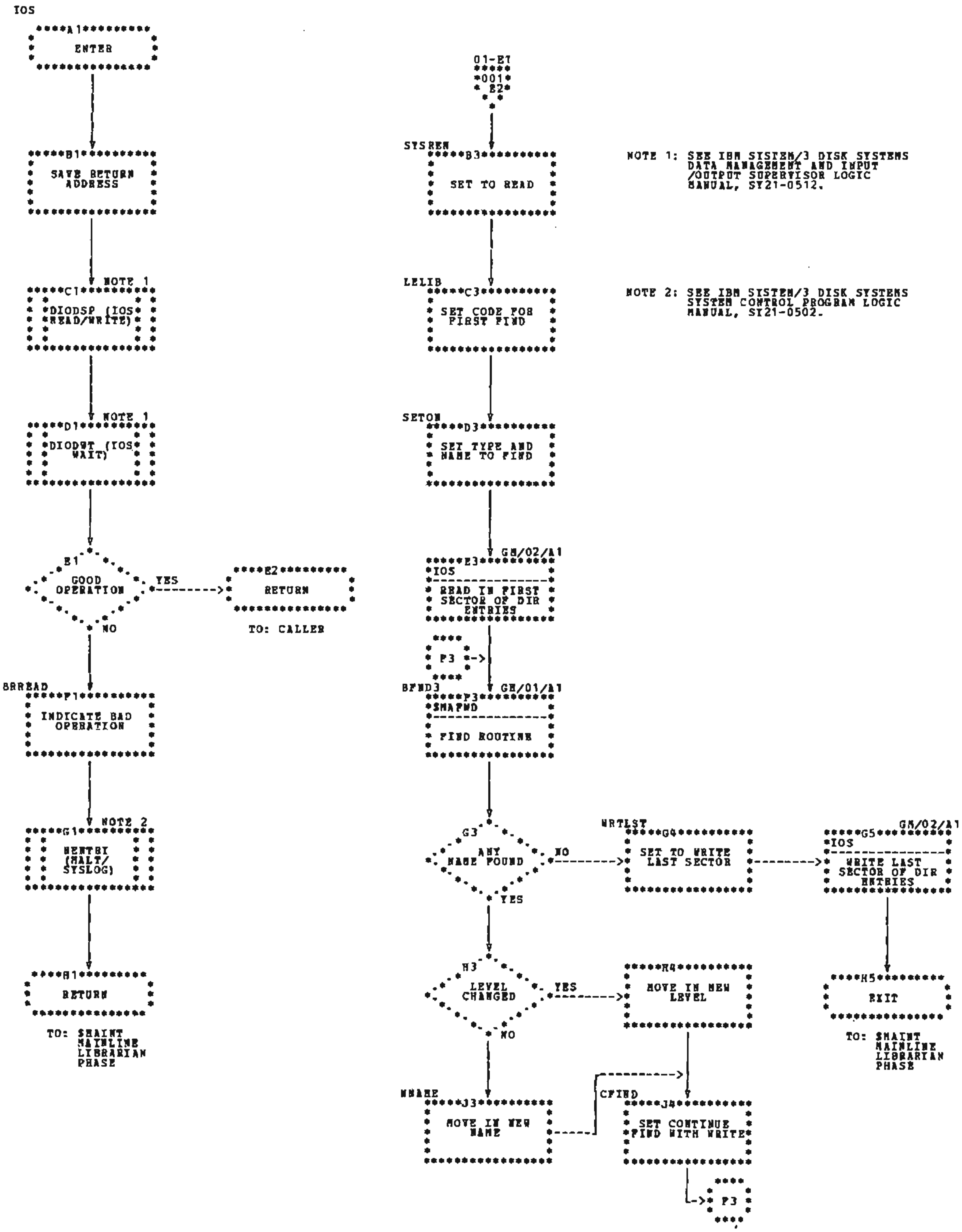


Chart GM (Part 1 of 2). Rename Control Statement Processor Phase



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS DATA MANAGEMENT AND INPUT /OUTPUT SUPERVISOR LOGIC MANUAL, SY21-0512.

NOTE 2: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart GM (Part 2 of 2). Rename Control Statement Processor Phase

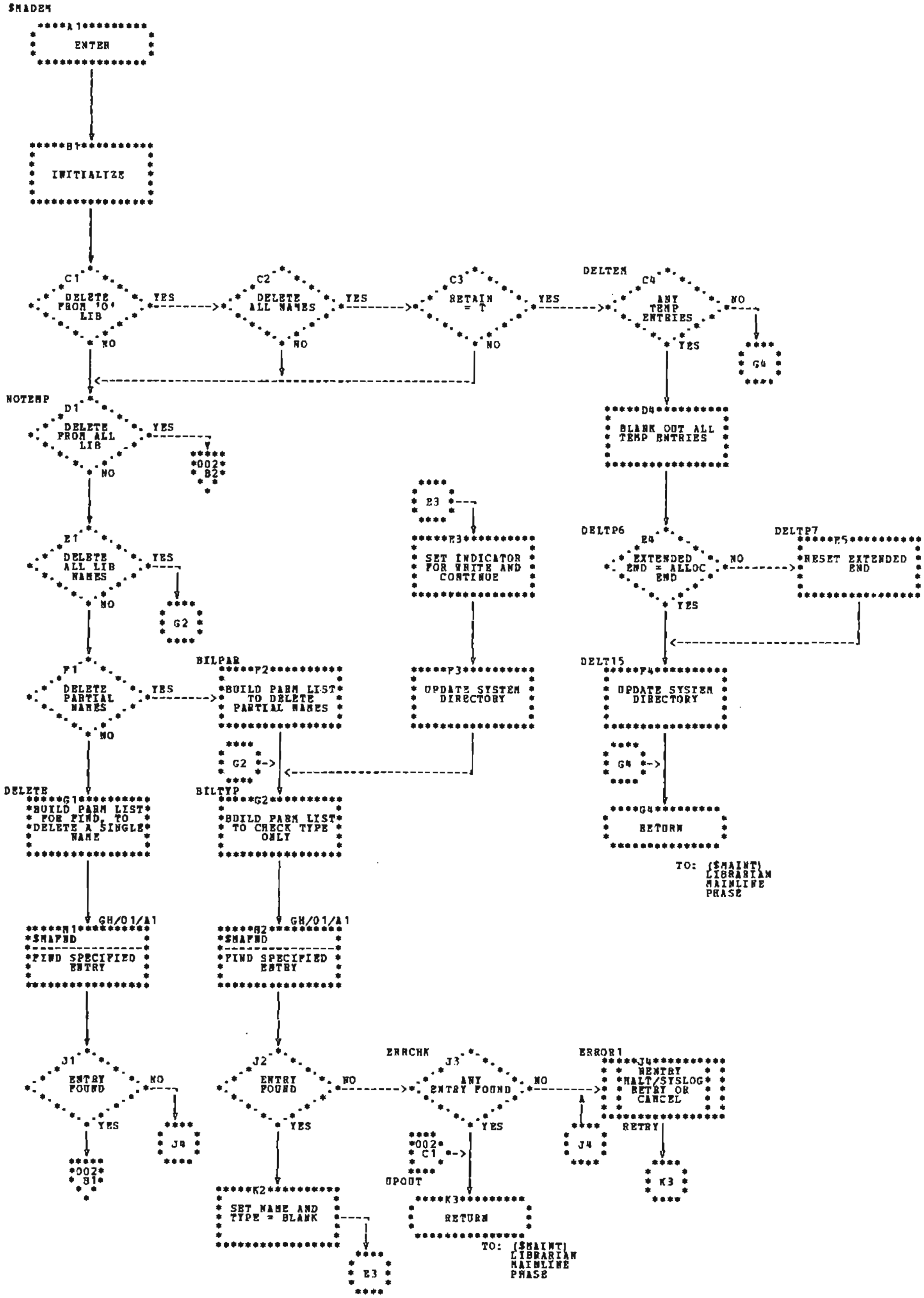


Chart GN (Part 1 of 2). Delete Mainline Phase

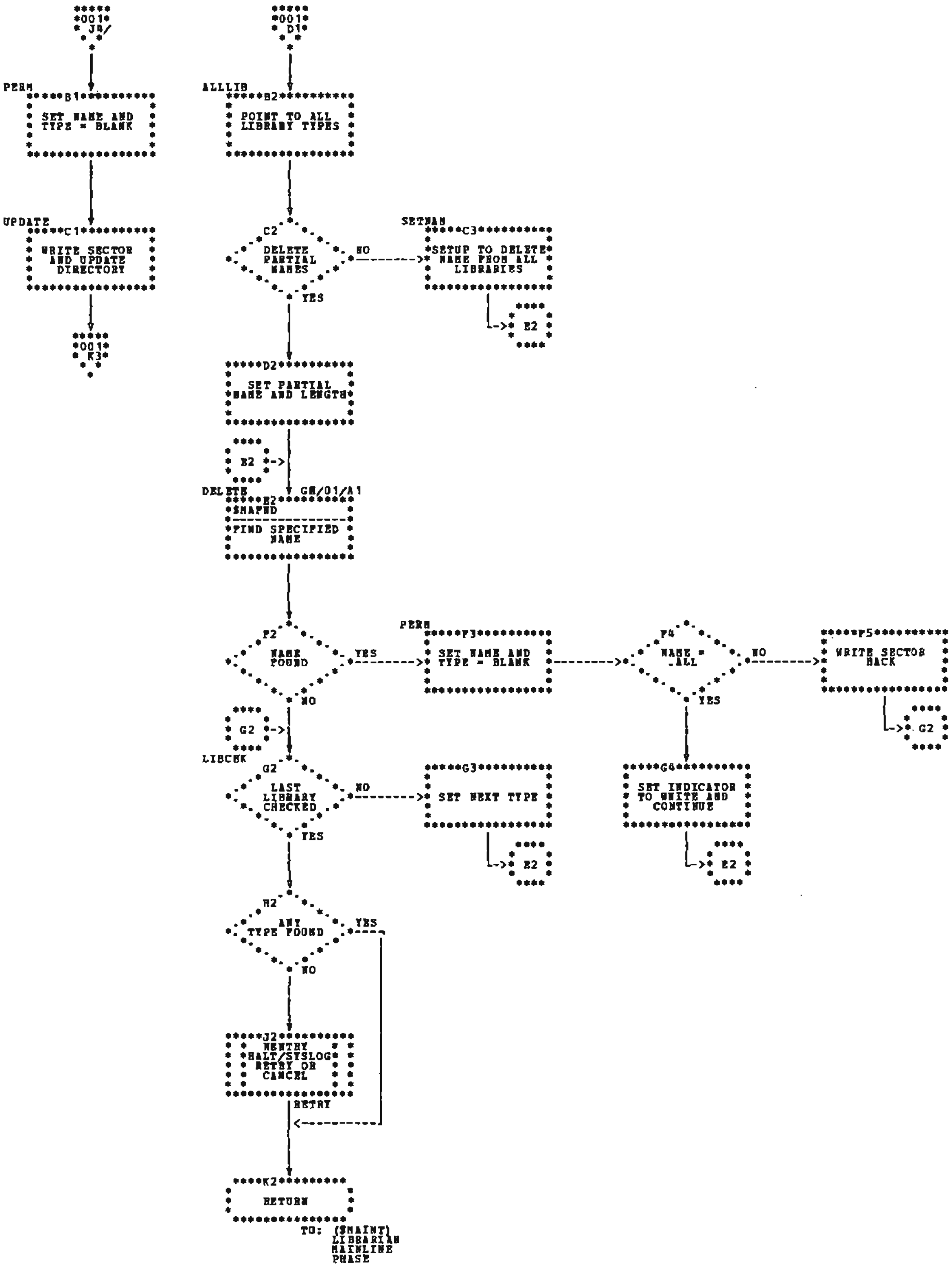
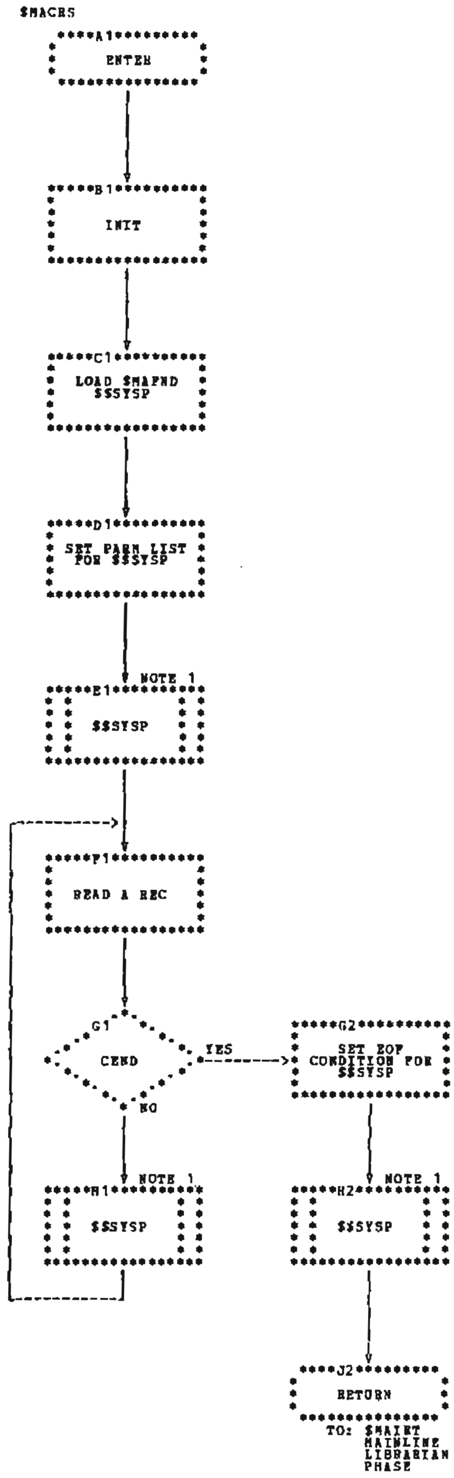
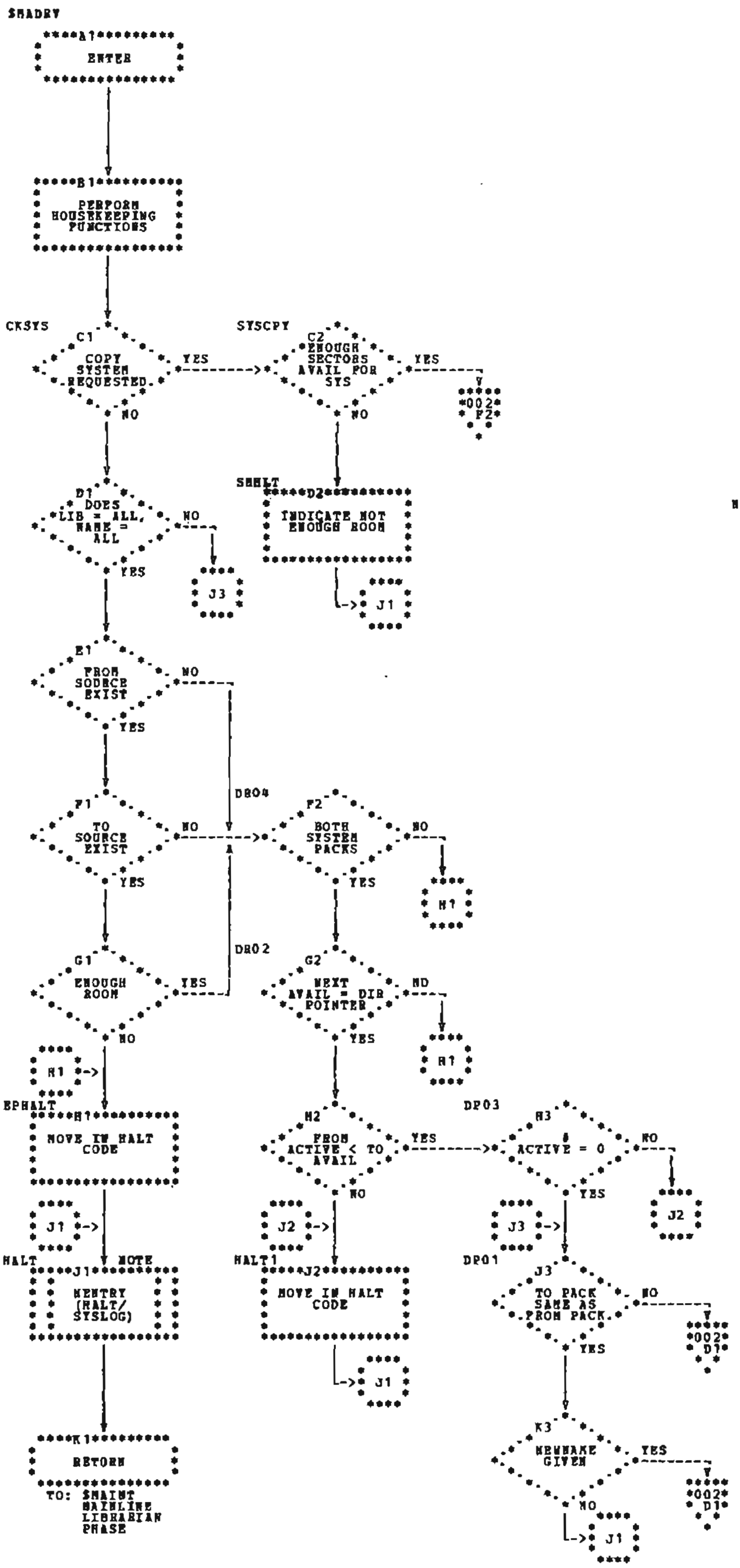

001 J3
001 D1
PERM
B1
SET NAME AND TYPE = BLANK
ALLLIB
B2
POINT TO ALL LIBRARY TYPES
UPDATE
C1
WRITE SECTOR AND UPDATE DIRECTORY
001 K3
C2
DELETE PARTIAL NAMES
NO
YES
SETNAM
C3
SETUP TO DELETE NAME FROM ALL LIBRARIES
E2
D2
SET PARTIAL NAME AND LENGTH
E2
DELETE
GN/01/A1
E2
$MAFND
FIND SPECIFIED NAME
F2
NAME FOUND
YES
NO
PERM
F3
SET NAME AND TYPE = BLANK
F4
NAME = .ALL
NO
YES
F5
WRITE SECTOR BACK
G2
G2
LIBCHK
G2
LAST LIBRARY CHECKED
NO
YES
G3
SET NEXT TYPE
E2
G4
SET INDICATOR TO WRITE AND CONTINUE
E2
H2
ANY TYPE FOUND
YES
NO
J2
MENTRY HALT/SYSLOG RETRY OR CANCEL
RETRY
K2
RETURN
TO: ($MAINT) LIBRARIAN MAINLINE PHASE

Chart GN (Part 2 of 2). Delete Mainline Phase



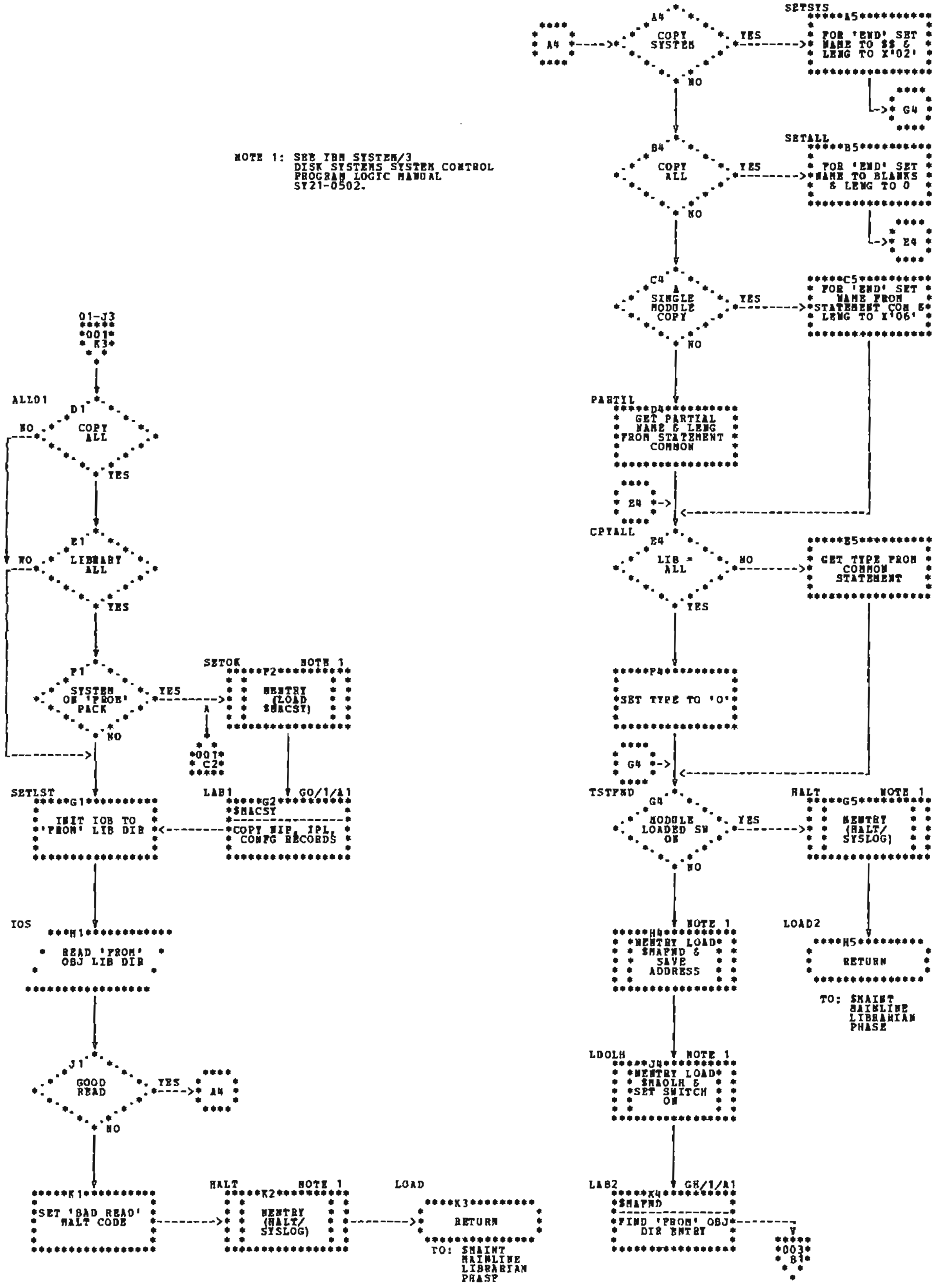
NOTE 1: SEE IBM SYSTEM/3
DISK SYSTEMS SYSTEM CONTROL
PROGRAM LOGIC MANUAL
SY21-0502.

Chart GO. Copy Reader to Source Library Phase



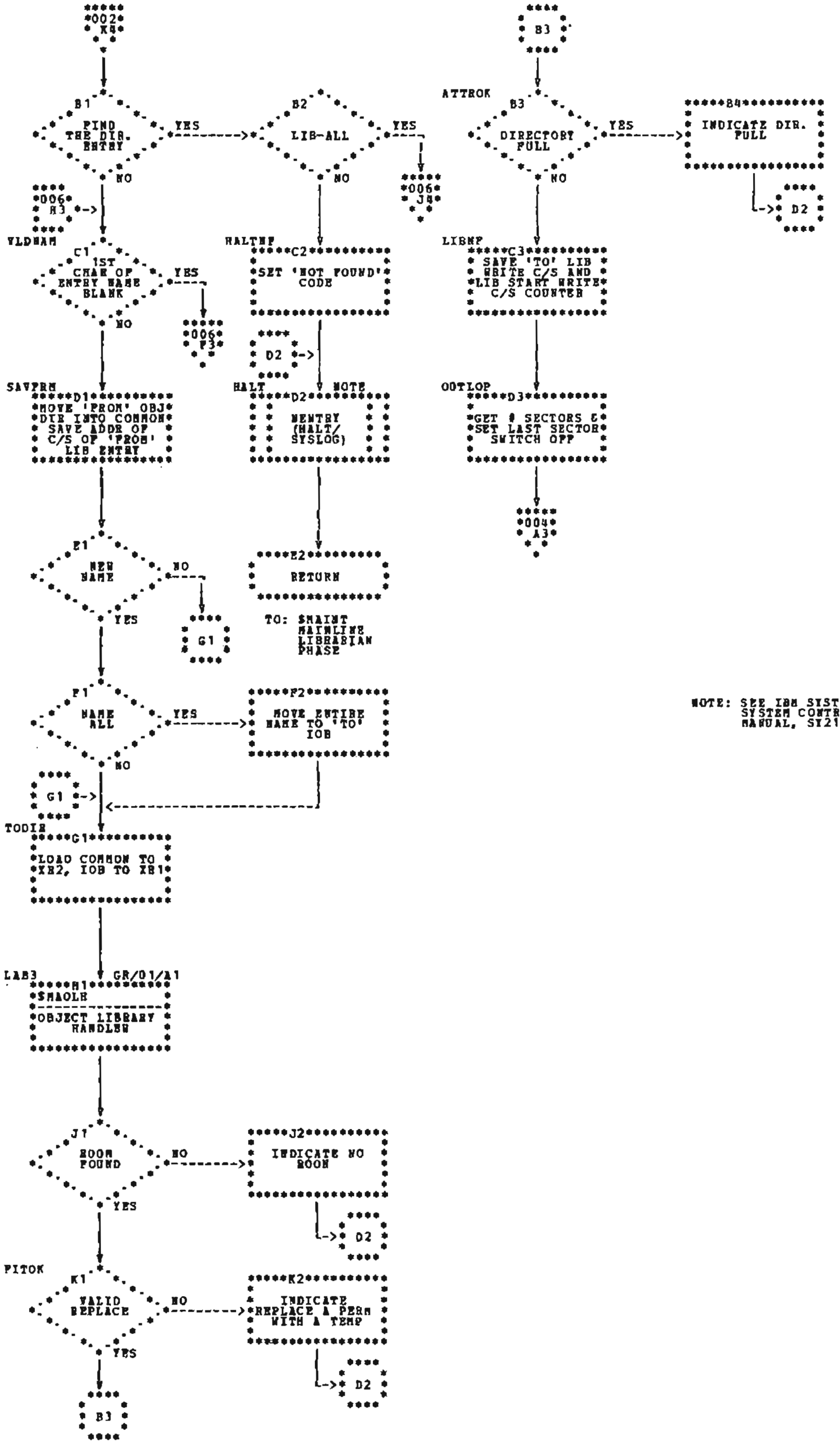
NOTE: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart GP (Part 1 of 7). Copy Object Library–Disk to Disk



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502.

Chart GP (Part 2 of 7). Copy Object Library—Disk to Disk



NOTE: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart GP (Part 3 of 7). Copy Object Library–Disk to Disk

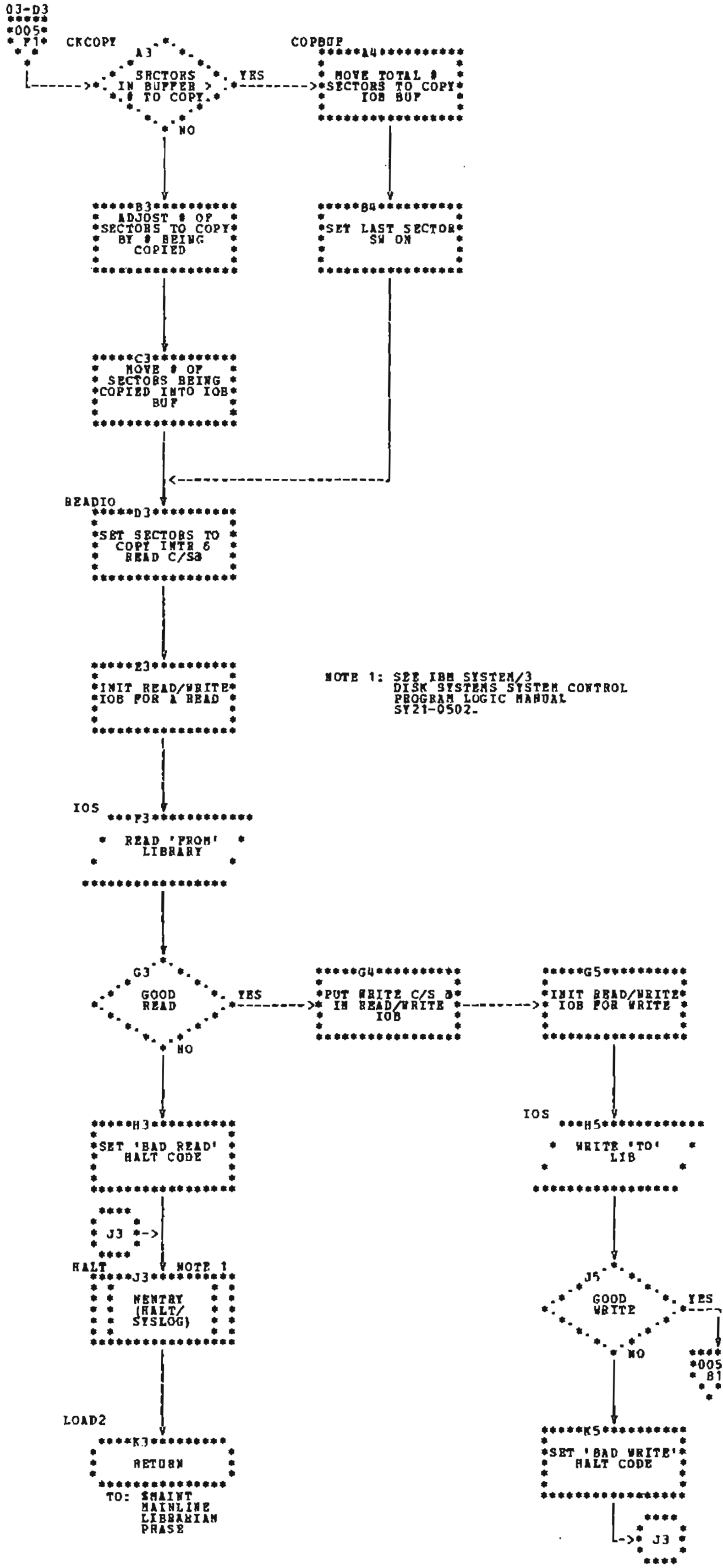

03-D3
005 F1
CKCOPY
A3
SECTORS IN BUFFER > # TO COPY
YES
NO
COPBUF
A4
MOVE TOTAL # SECTORS TO COPY IOB BUF
B3
ADJUST # OF SECTORS TO COPY BY # BEING COPIED
B4
SET LAST SECTOR SW ON
C3
MOVE # OF SECTORS BEING COPIED INTO IOB BUF
READIO
D3
SET SECTORS TO COPY INTR 6 READ C/S@
E3
INIT READ/WRITE IOB FOR A READ
NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502.
IOS
F3
READ 'FROM' LIBRARY
G3
GOOD READ
YES
NO
G4
PUT WRITE C/S @ IN READ/WRITE IOB
G5
INIT READ/WRITE IOB FOR WRITE
H3
SET 'BAD READ' HALT CODE
IOS
H5
WRITE 'TO' LIB
J3
HALT
NOTE 1
J3
NENTRY (HALT/ SYSLOG)
J5
GOOD WRITE
YES
NO
005 B1
LOAD2
K3
RETURN
TO: $MAINT MAINLINE LIBRARIAN PHASE
K5
SET 'BAD WRITE' HALT CODE
J3

Chart GP (Part 4 of 7). Copy Object Library–Disk to Disk

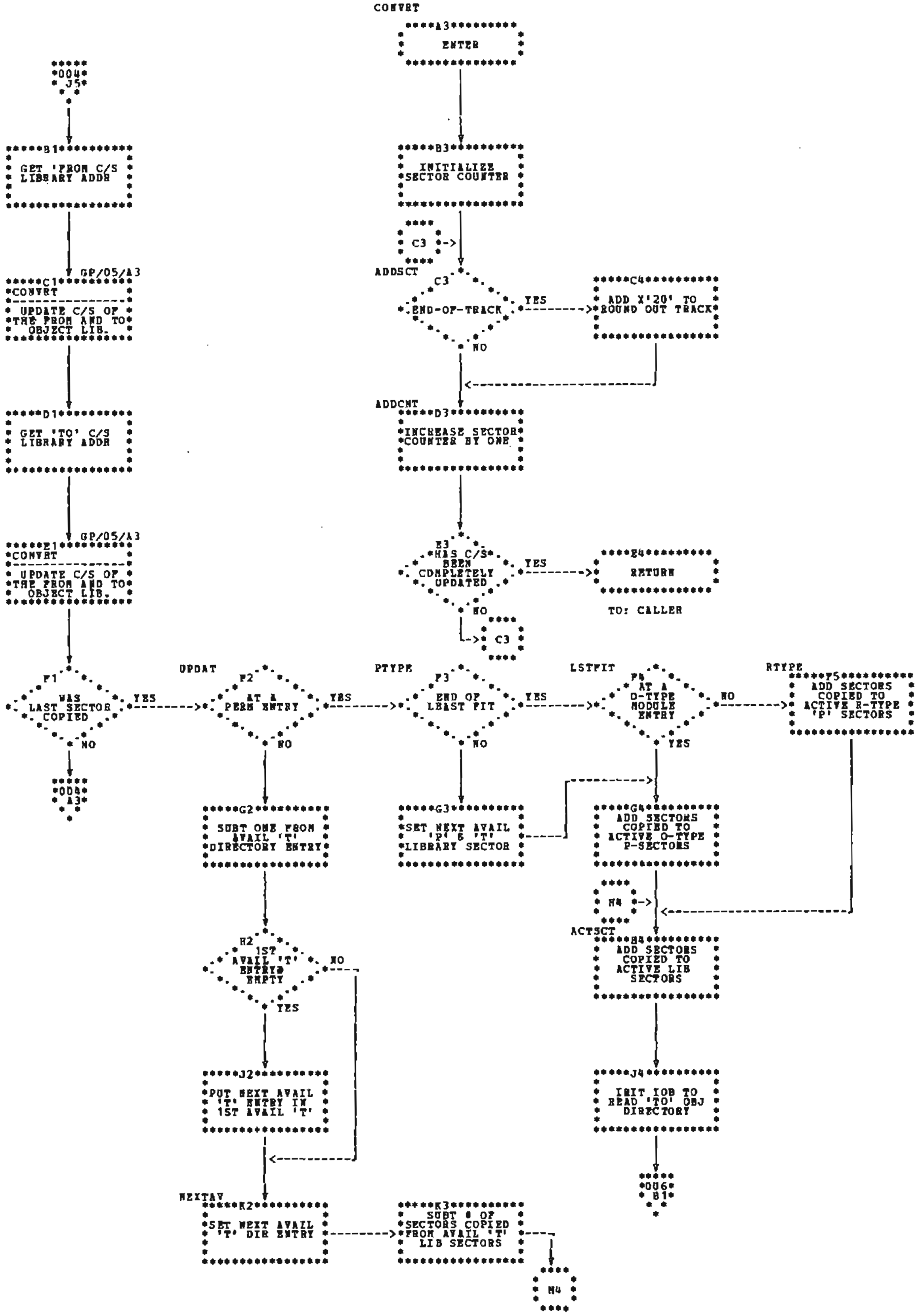

CONVRT
A3
ENTER
B3
INITIALIZE SECTOR COUNTER
C3
ADDSCT
C3
END-OF-TRACK
YES
NO
C4
ADD X'20' TO ROUND OUT TRACK
ADDCNT
D3
INCREASE SECTOR COUNTER BY ONE
E3
HAS C/S BEEN COMPLETELY UPDATED
YES
NO
E4
RETURN
TO: CALLER
C3
004
J5
B1
GET 'FROM C/S LIBRARY ADDR
GP/05/A3
C1
CONVRT
UPDATE C/S OF THE FROM AND TO OBJECT LIB.
D1
GET 'TO' C/S LIBRARY ADDR
GP/05/A3
E1
CONVRT
UPDATE C/S OF THE FROM AND TO OBJECT LIB.
F1
WAS LAST SECTOR COPIED
YES
NO
004
A3
UPDAT
F2
AT A PERM ENTRY
YES
NO
G2
SUBT ONE FROM AVAIL 'T' DIRECTORY ENTRY
H2
1ST AVAIL 'T' ENTRYS EMPTY
NO
YES
J2
PUT NEXT AVAIL 'T' ENTRY IN 1ST AVAIL 'T'
NEXTAV
K2
SET NEXT AVAIL 'T' DIR ENTRY
K3
SUBT # OF SECTORS COPIED FROM AVAIL 'T' LIB SECTORS
H4
PTYPE
F3
END OF LEAST FIT
YES
NO
G3
SET NEXT AVAIL 'P' & 'T' LIBRARY SECTOR
LSTFIT
F4
AT A O-TYPE MODULE ENTRY
NO
YES
G4
ADD SECTORS COPIED TO ACTIVE O-TYPE P-SECTORS
RTYPE
F5
ADD SECTORS COPIED TO ACTIVE R-TYPE 'P' SECTORS
H4
ACTSCT
H4
ADD SECTORS COPIED TO ACTIVE LIB SECTORS
J4
INIT IOB TO READ 'TO' OBJ DIRECTORY
006
B1

Chart GP (Part 5 of 7). Copy Object Library–Disk to Disk

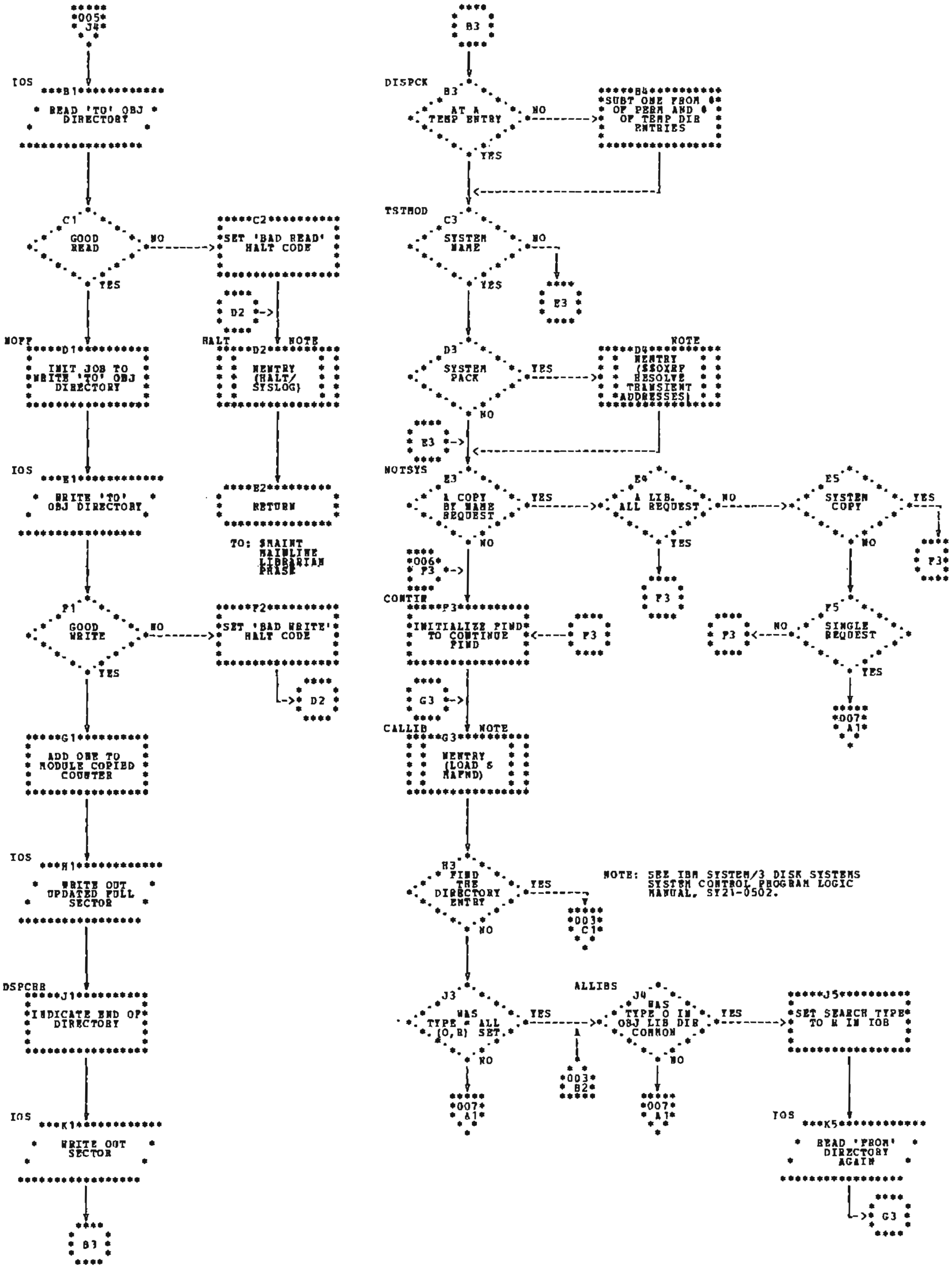


Chart GP (Part 6 of 7). Copy Object Library—Disk to Disk

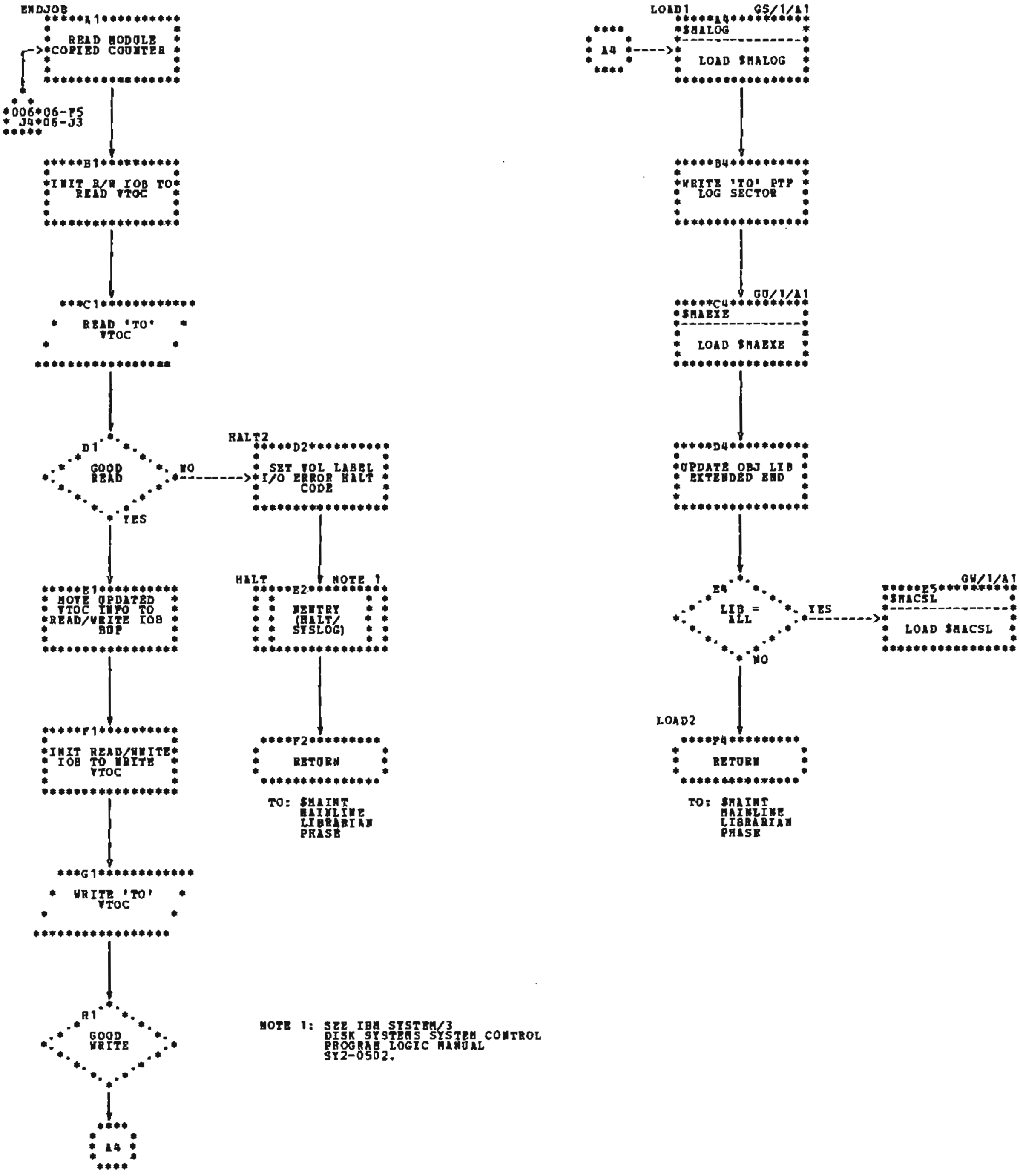
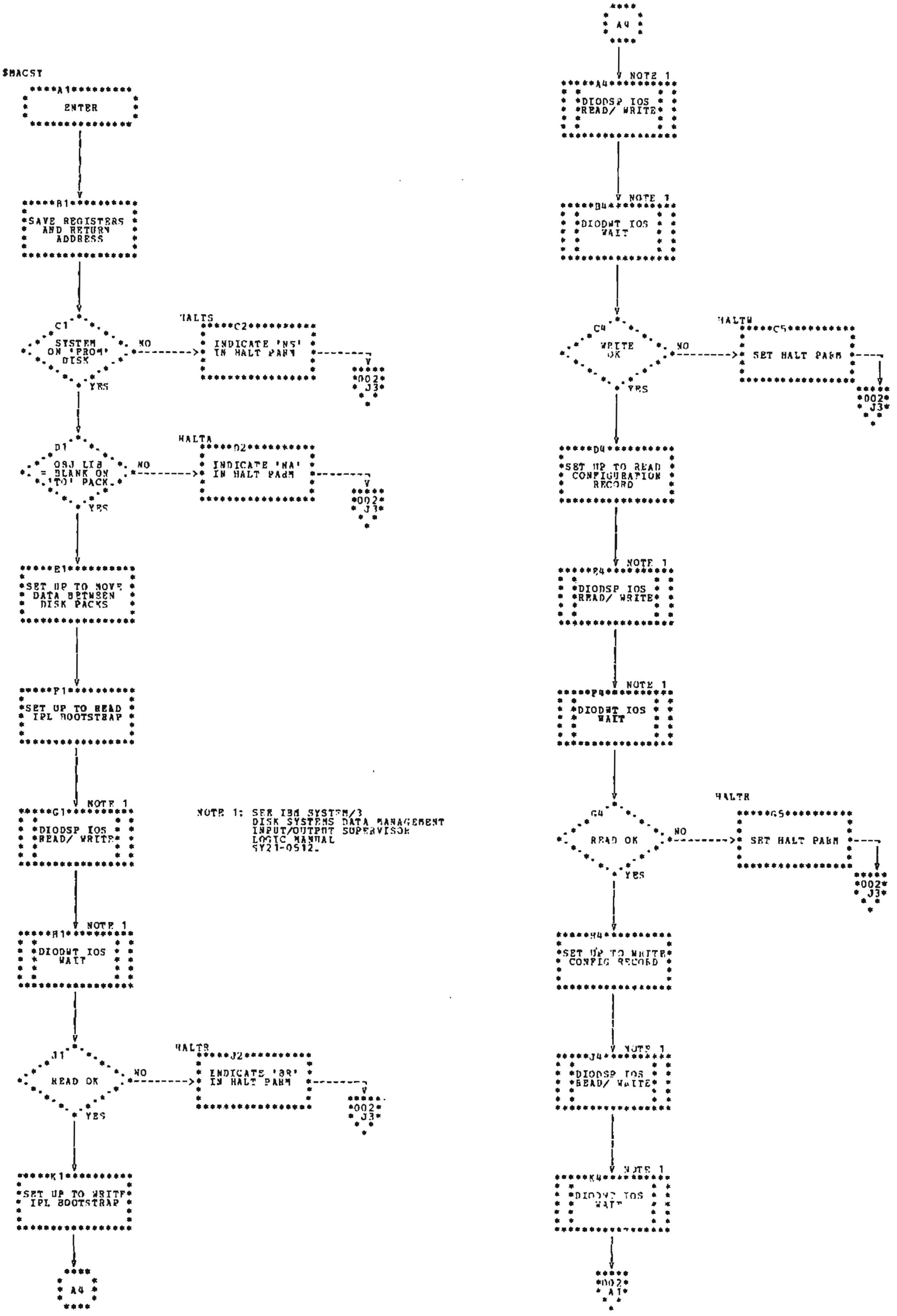
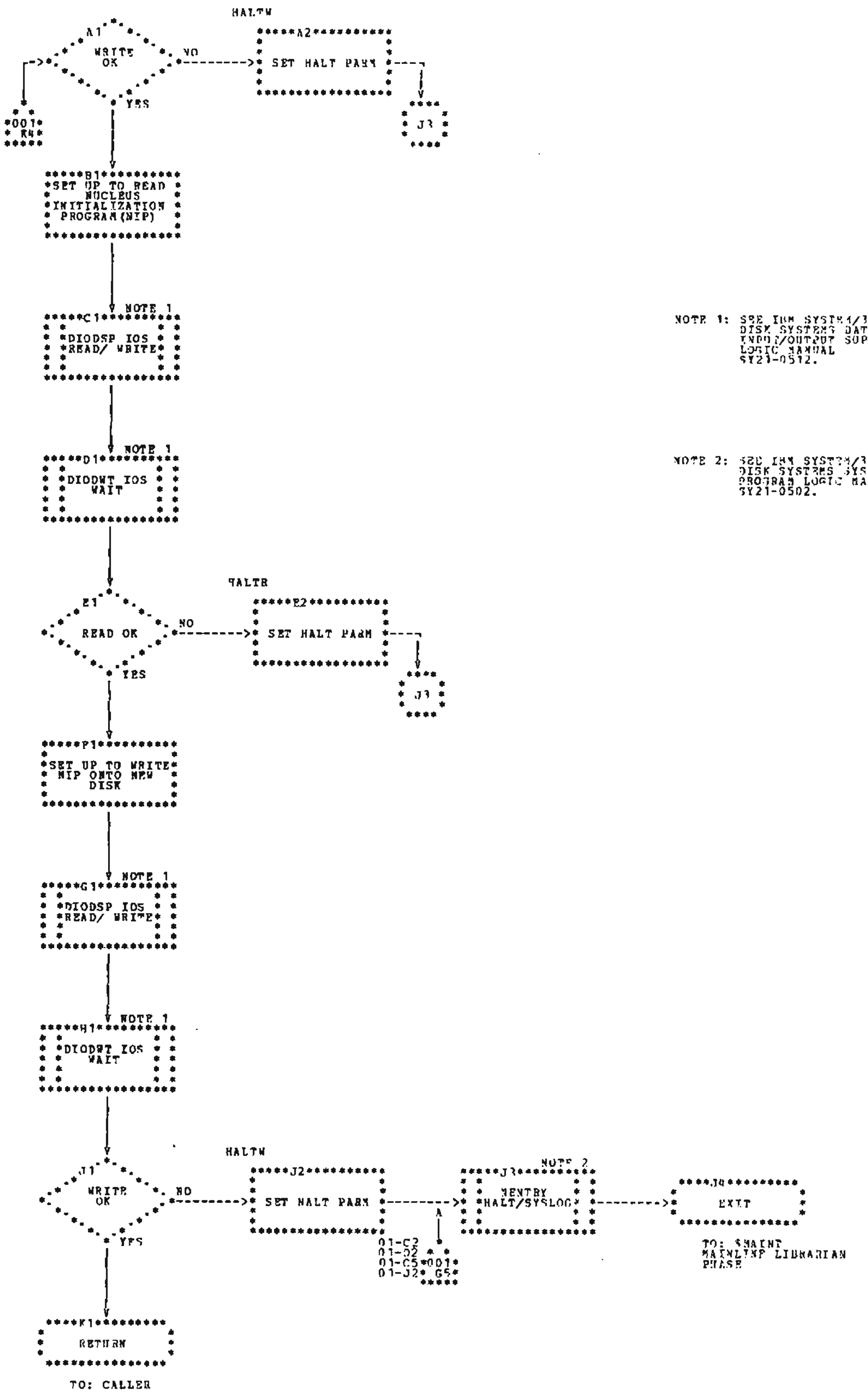

ENDJOB
A1
READ MODULE COPIED COUNTER
006*06-F5
J4*06-J3
B1
INIT R/W IOB TO READ VTOC
C1
READ 'TO' VTOC
D1
GOOD READ
NO
YES
HALT2
D2
SET VOL LABEL I/O ERROR HALT CODE
E1
MOVE UPDATED VTOC INFO TO READ/WRITE IOB BUF
HALT
E2
NOTE 1
$ENTRY (HALT/ SYSLOG)
F1
INIT READ/WRITE IOB TO WRITE VTOC
F2
RETURN
TO: $MAINT MAINLINE LIBRARIAN PHASE
G1
WRITE 'TO' VTOC
H1
GOOD WRITE
A4
NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY2-0502.
LOAD1
GS/1/A1
A4
$MALOG
LOAD $MALOG
B4
WRITE 'TO' PTF LOG SECTOR
GU/1/A1
C4
$MAEXE
LOAD $MAEXE
D4
UPDATE OBJ LIB EXTENDED END
E4
LIB = ALL
YES
NO
GW/1/A1
E5
$MACSL
LOAD $MACSL
LOAD2
F4
RETURN
TO: $MAINT MAINLINE LIBRARIAN PHASE

Chart GP (Part 7 of 7). Copy Object Library–Disk to Disk



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS DATA MANAGEMENT INPUT/OUTPUT SUPERVISOR LOGIC MANUAL SY21-0512.

Chart GQ (Part 1 of 2). Copy System Mainline Phase



NOTE 1: SEE IBM SYSTEM/3
DISK SYSTEMS DATA MANAGEMENT
INPUT/OUTPUT SUPERVISOR
LOGIC MANUAL
SY21-0512.

NOTE 2: SEE IBM SYSTEM/3
DISK SYSTEMS SYSTEM CONTROL
PROGRAM LOGIC MANUAL
SY21-0502.

Chart GQ (Part 2 of 2). Copy System Mainline Phase

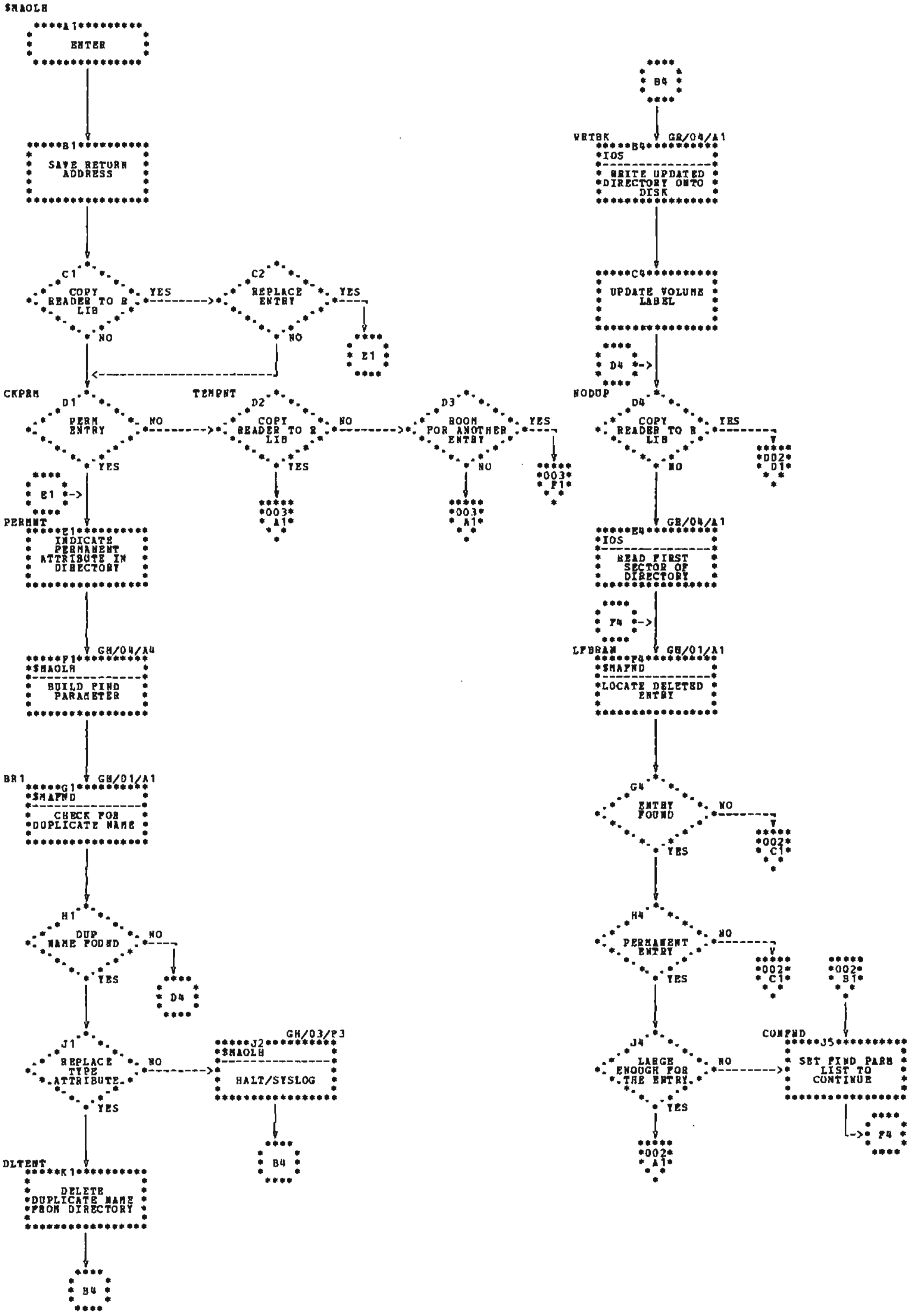


Chart GR (Part 1 of 4). Object Library Handler Phase

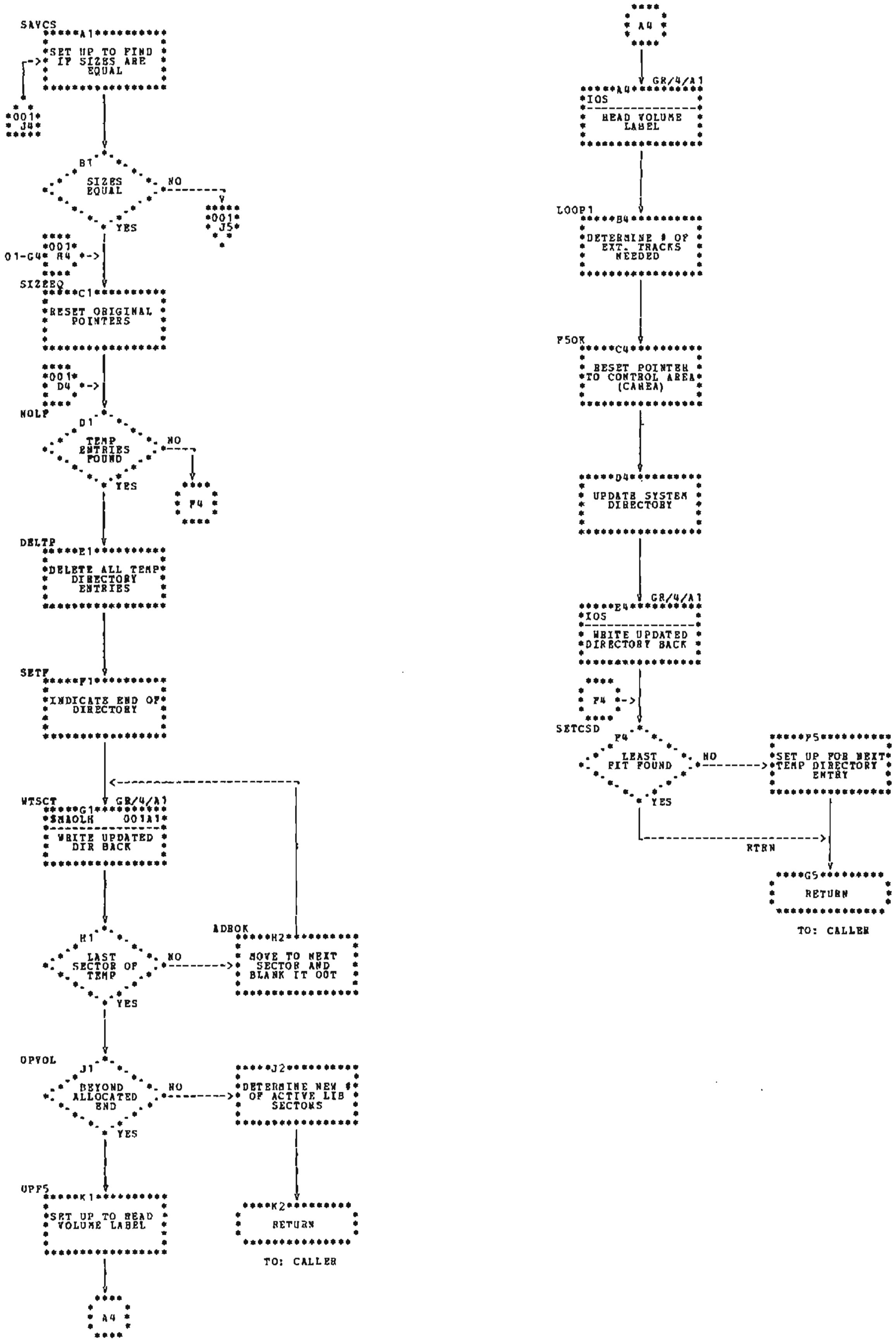
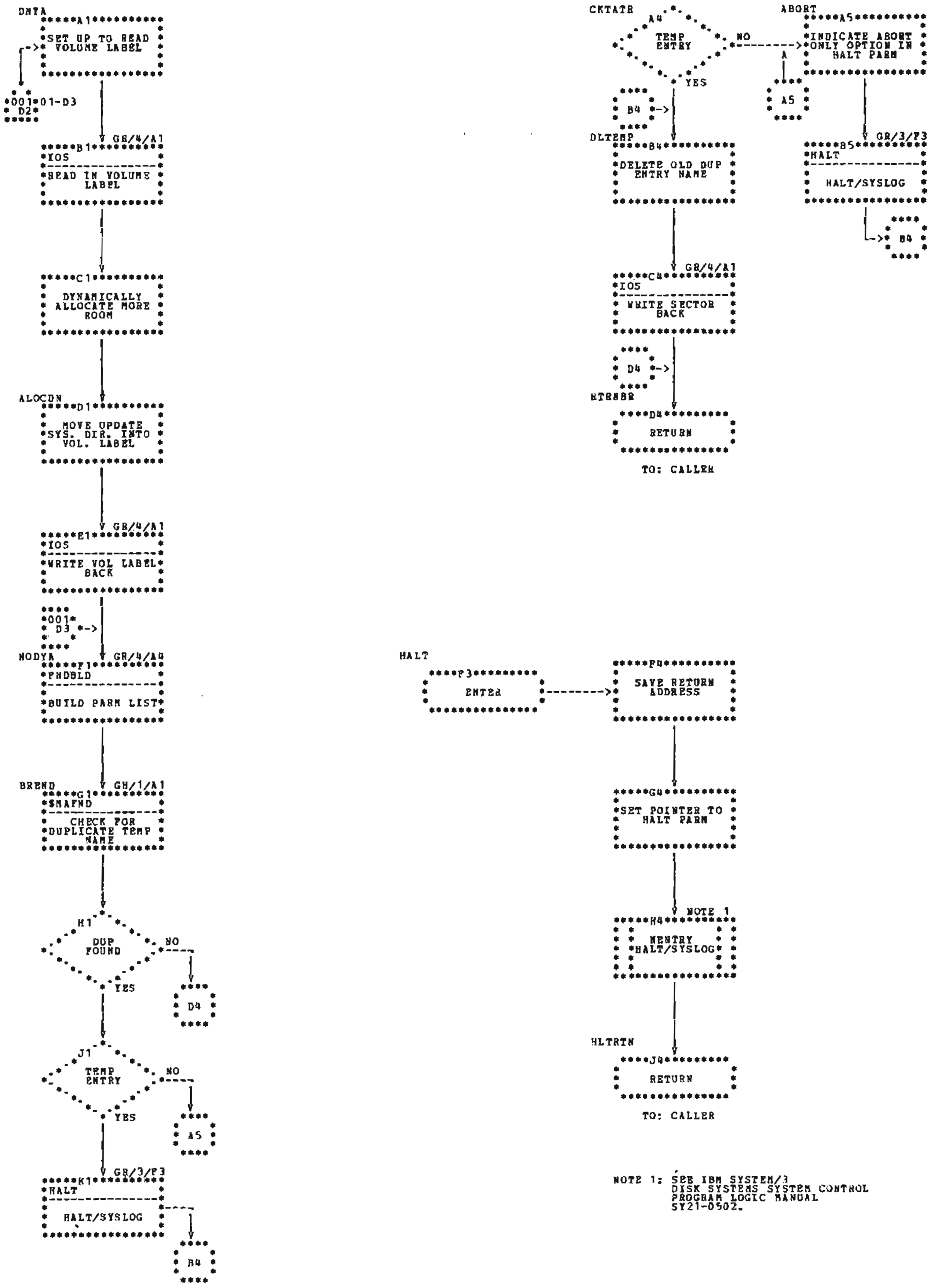
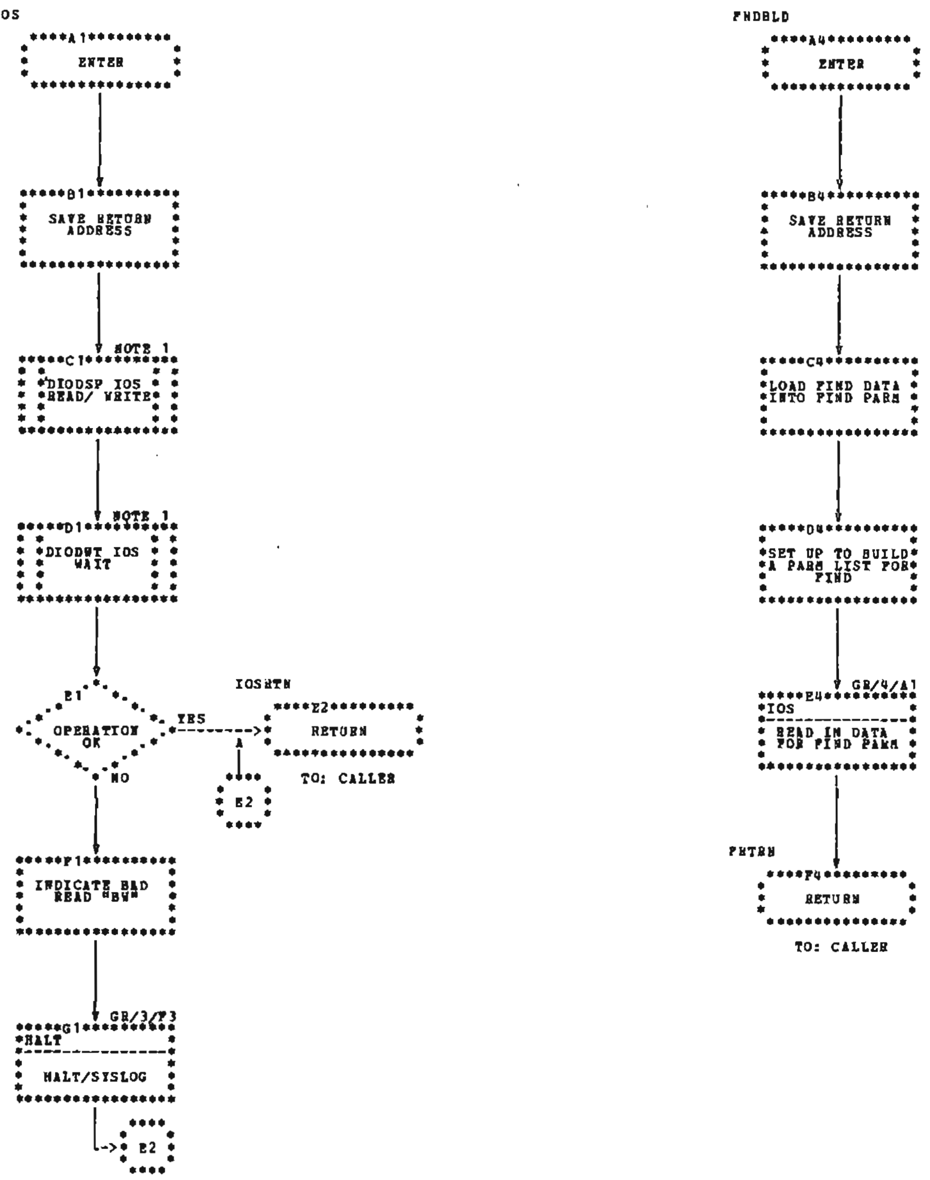


Chart GR (Part 2 of 4). Object Library Handler Phase



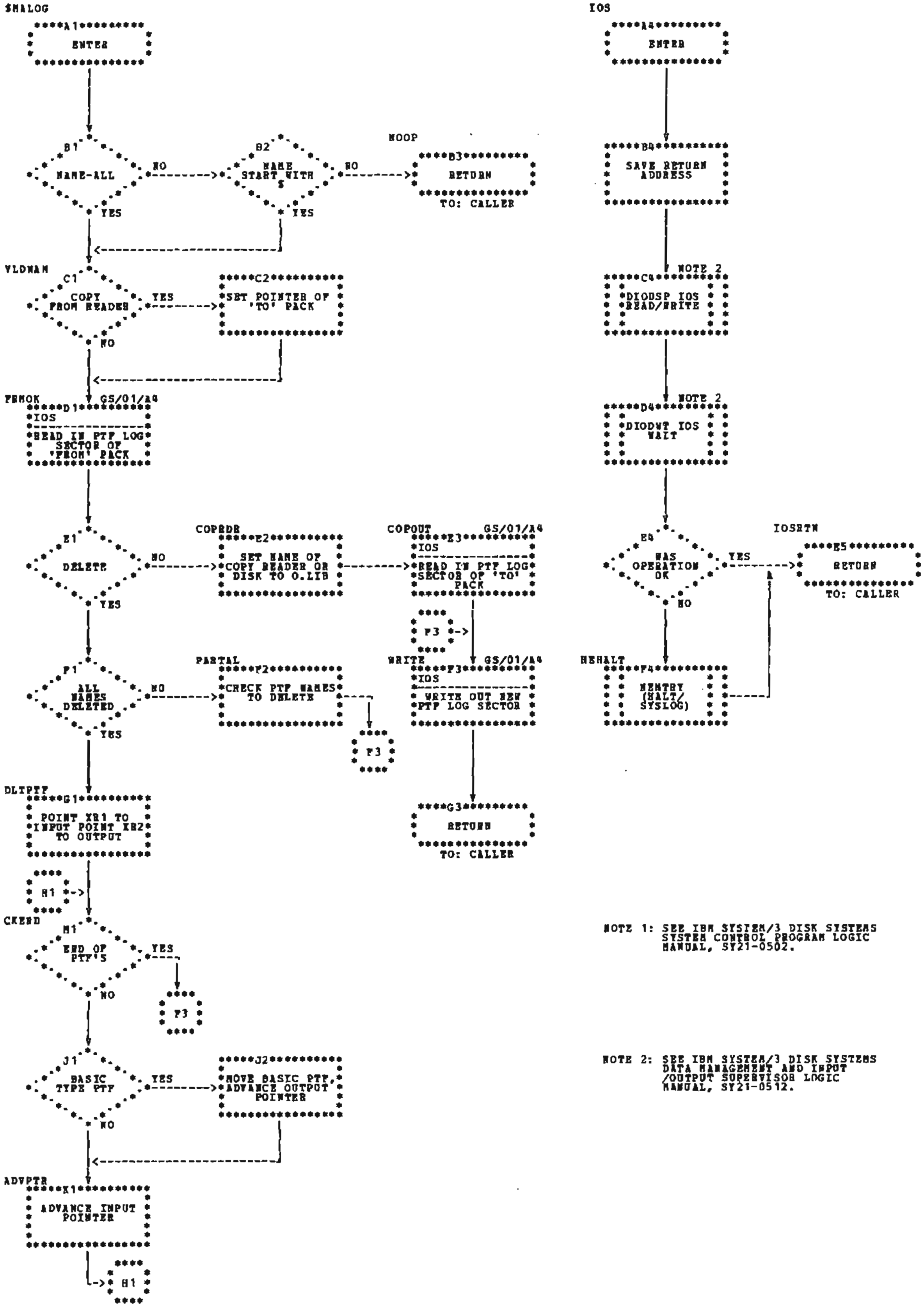
NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502.

Chart GR (Part 3 of 4). Object Library Handler Phase



NOTE 1: SEE IBM SYSTEM/3
DISK SYSTEMS DATA MANAGEMENT
INPUT/OUTPUT SUPERVISOR
LOGIC MANUAL
SY21-0512.

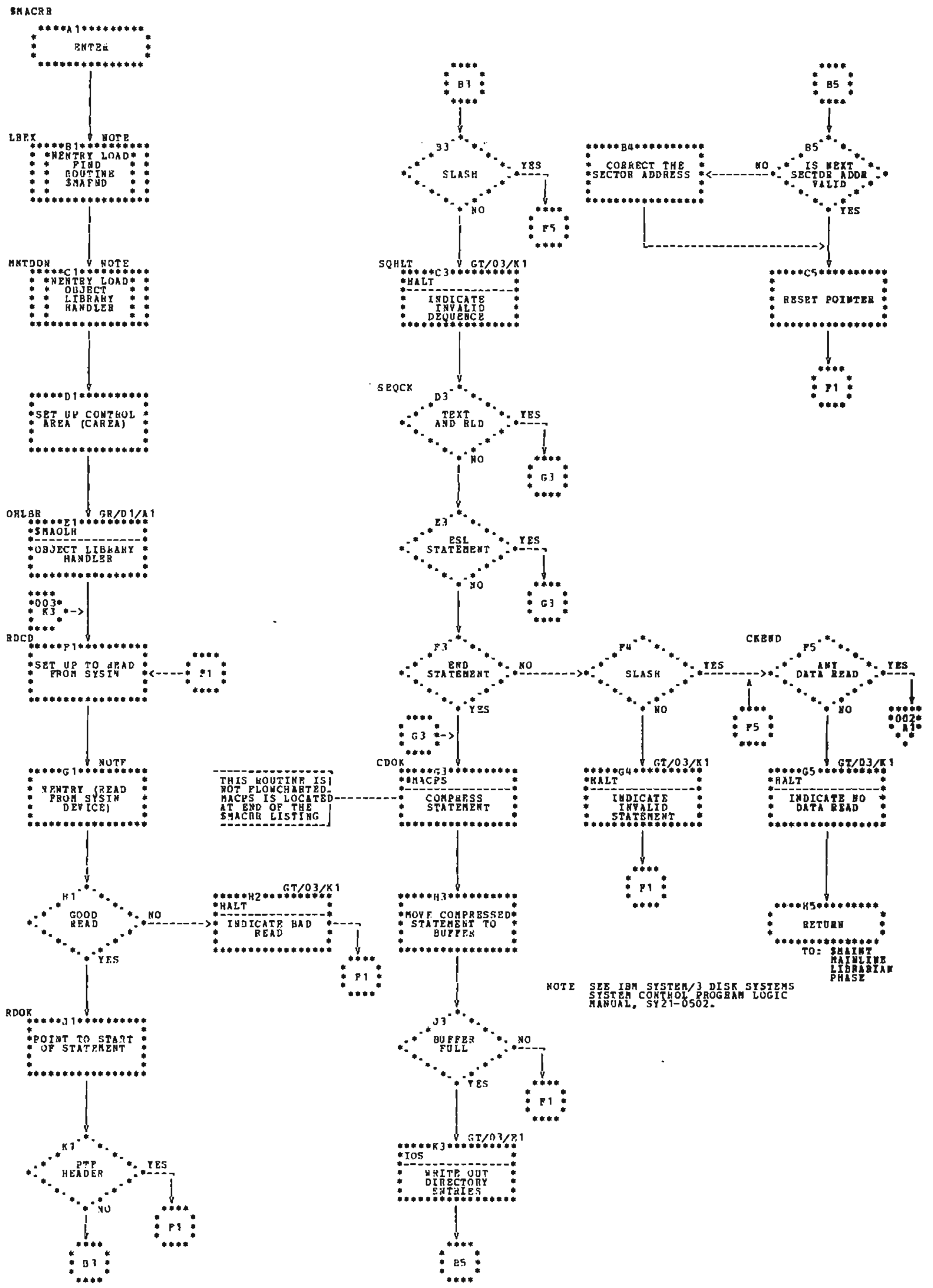
Chart GR (Part 4 of 4). Object Library Handler Phase



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

NOTE 2: SEE IBM SYSTEM/3 DISK SYSTEMS DATA MANAGEMENT AND INPUT /OUTPUT SUPERVISOR LOGIC MANUAL, SY21-0512.

Chart GS. Update PTF Log Sector Phase



NOTE SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart GT (Part 1 of 3). Copy Reader to Object Library (R. Entries) Phase

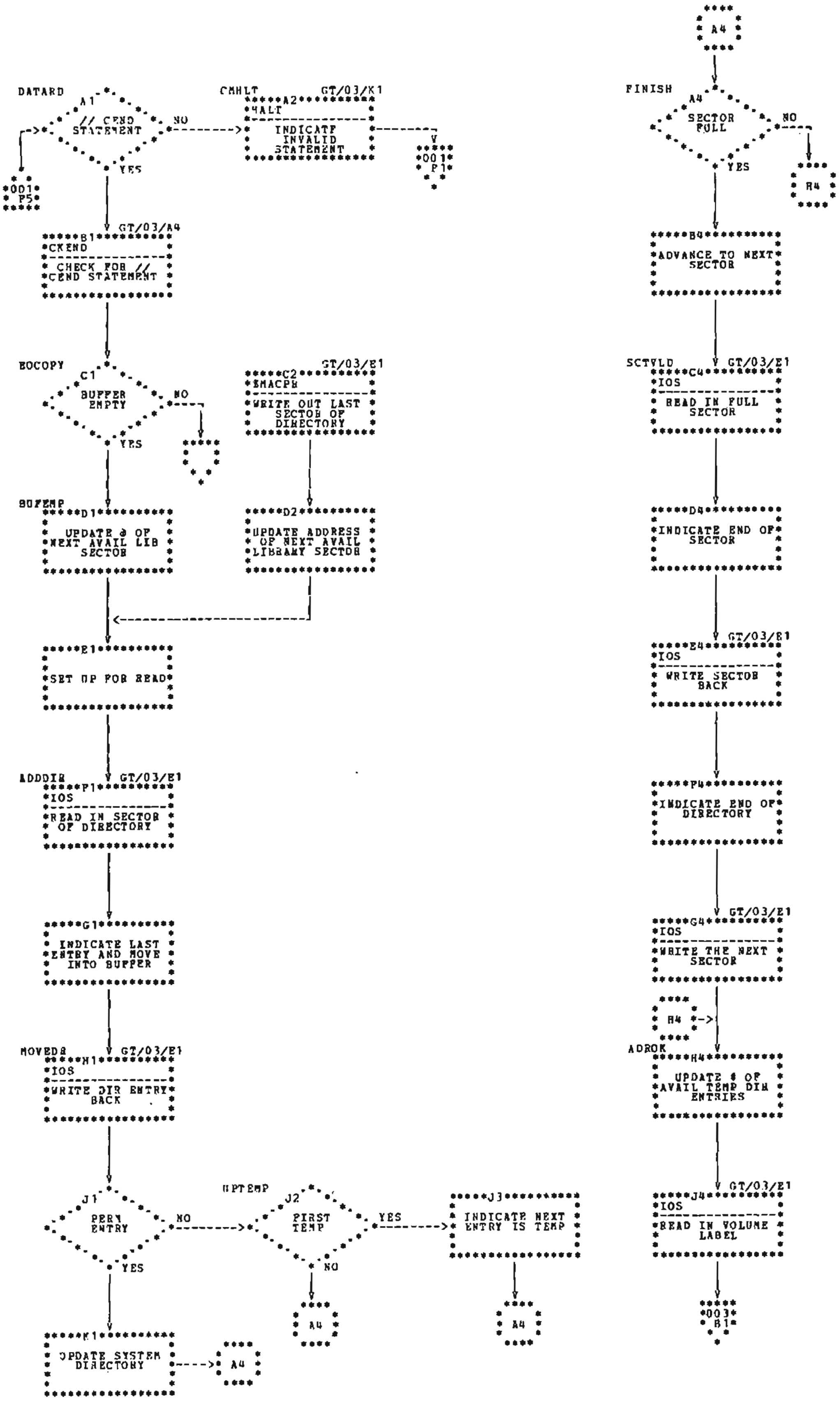
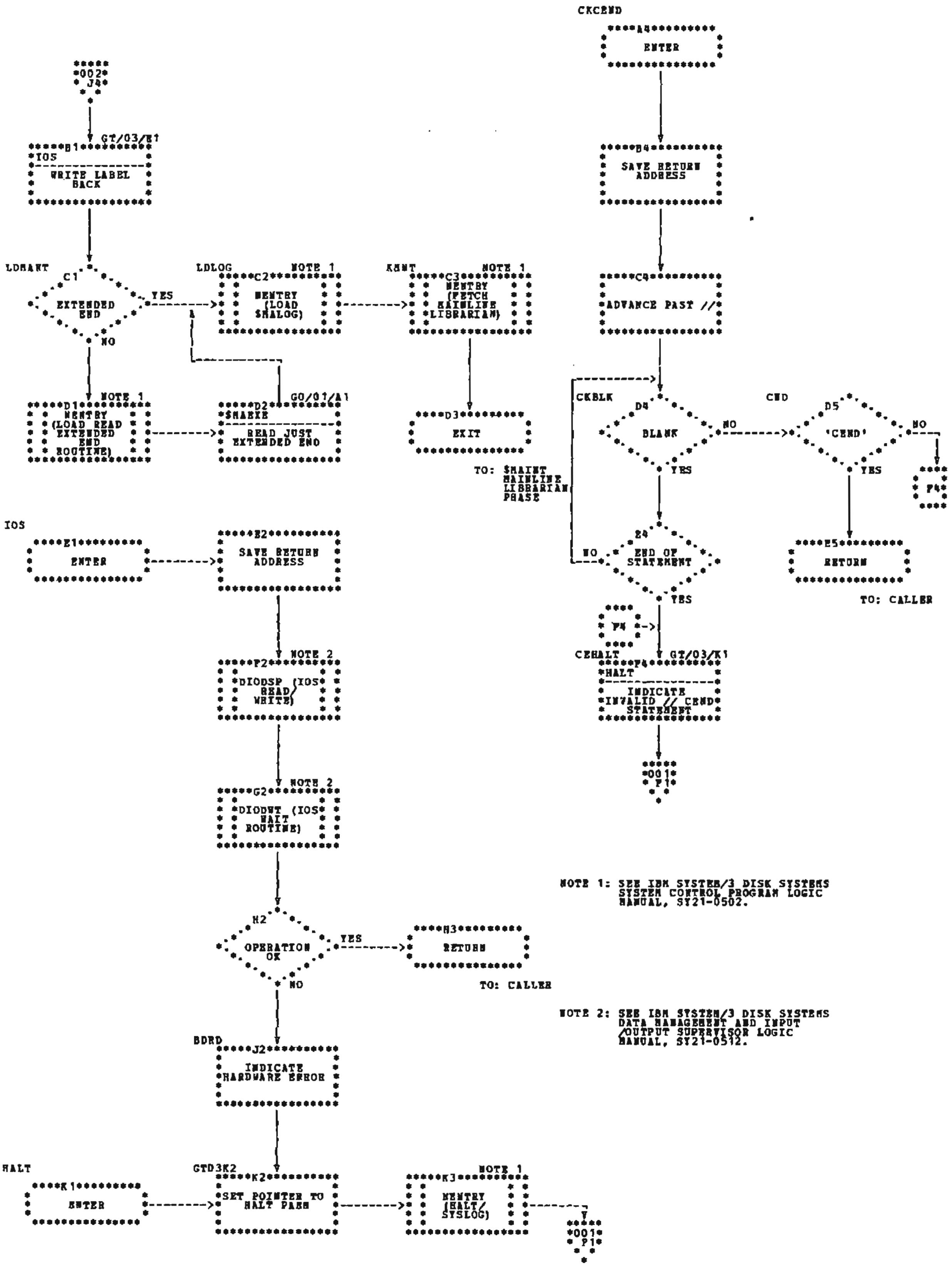


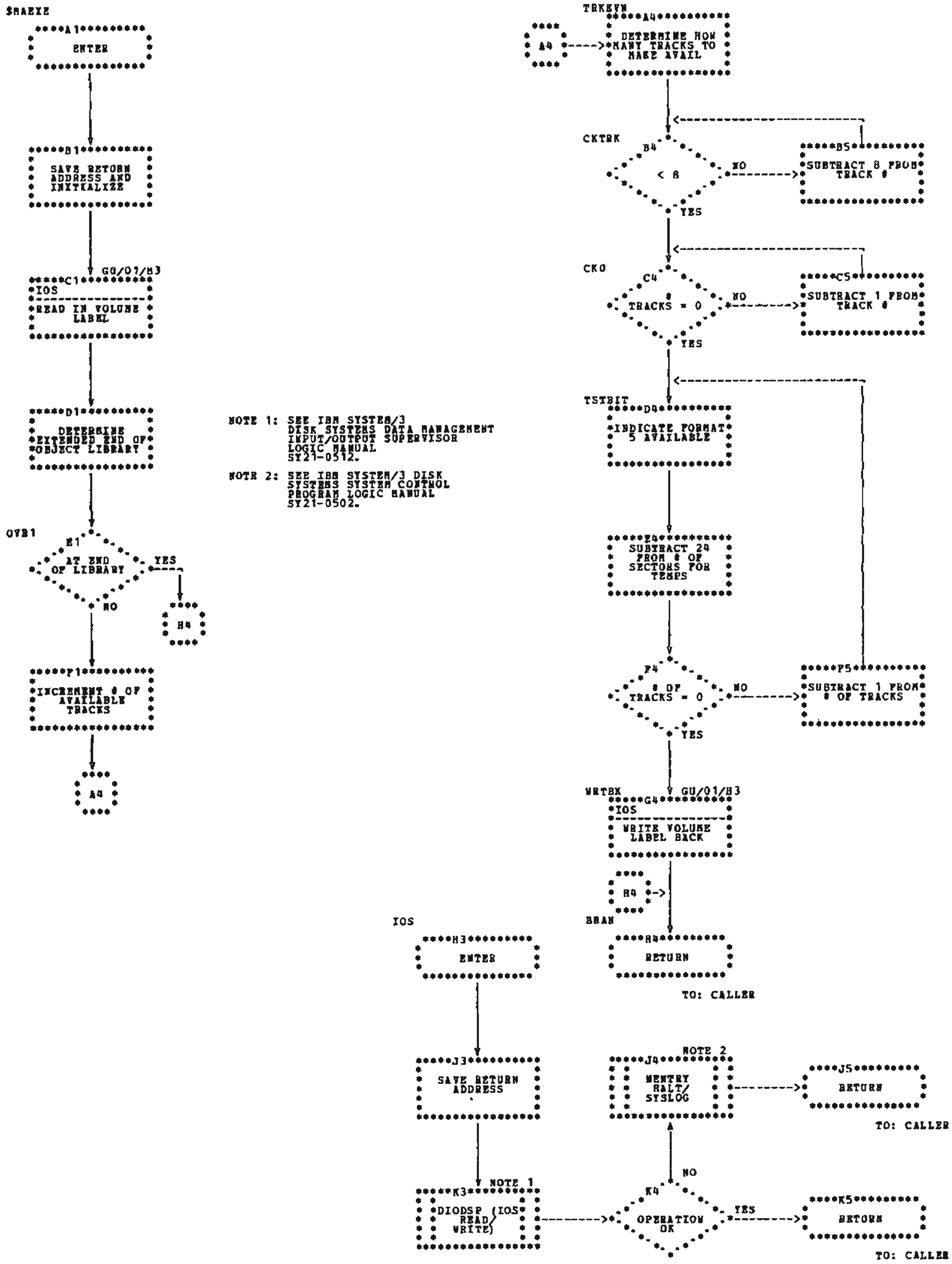
Chart GT (Part 2 of 3). Copy Reader to Object Library (R. Entries) Phase



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

NOTE 2: SEE IBM SYSTEM/3 DISK SYSTEMS DATA MANAGEMENT AND INPUT /OUTPUT SUPERVISOR LOGIC MANUAL, SY21-0512.

Chart GT (Part 3 of 3). Copy Reader to Object Library (R. Entries) Phase



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS DATA MANAGEMENT INPUT/OUTPUT SUPERVISOR LOGIC MANUAL SY21-0512.

NOTE 2: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502.

Chart GU. Readjust Extended End of Object Library Phase

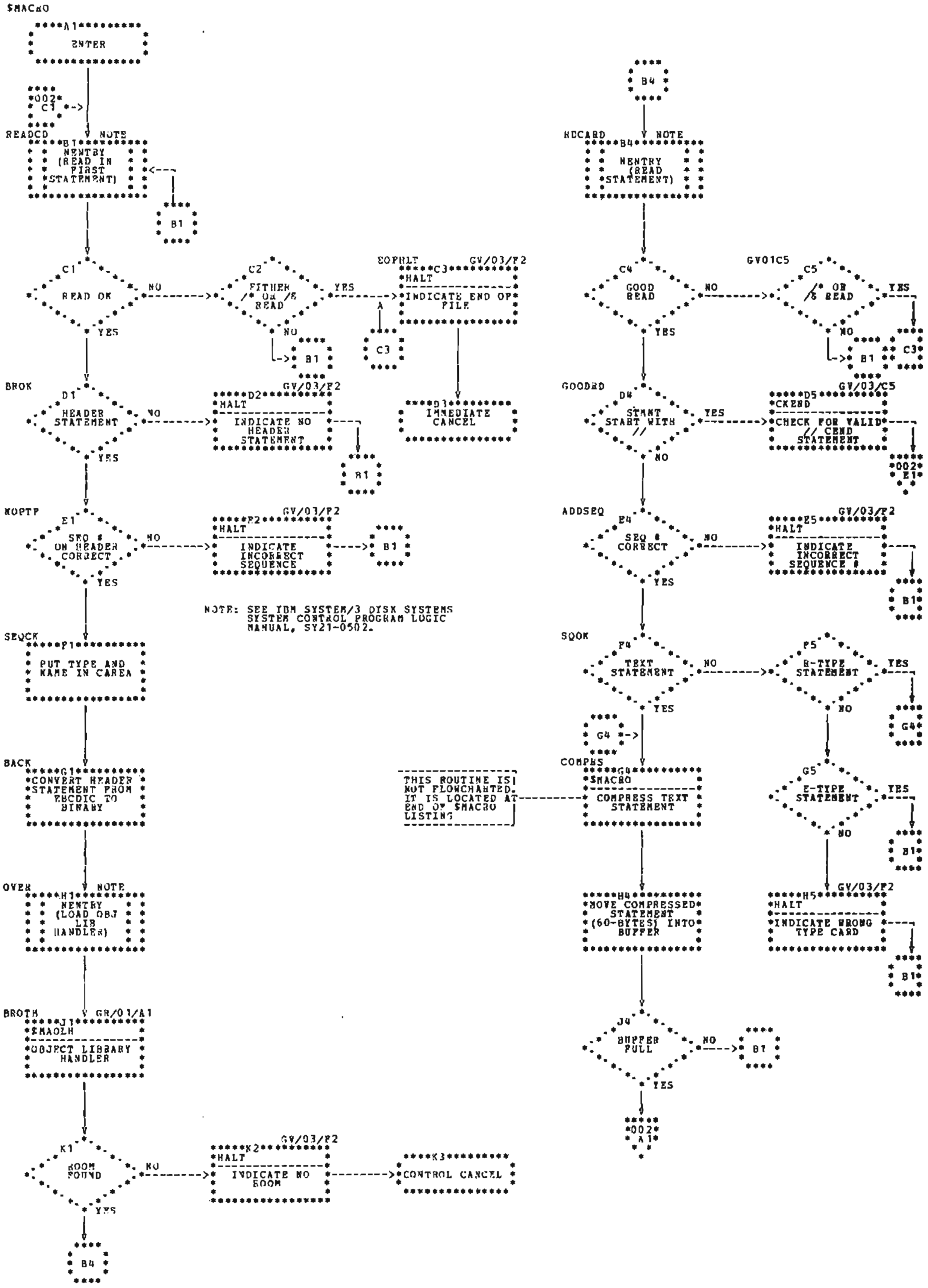


Chart GV (Part 1 of 3). Copy Reader to Object Library (O. Entries) Phase

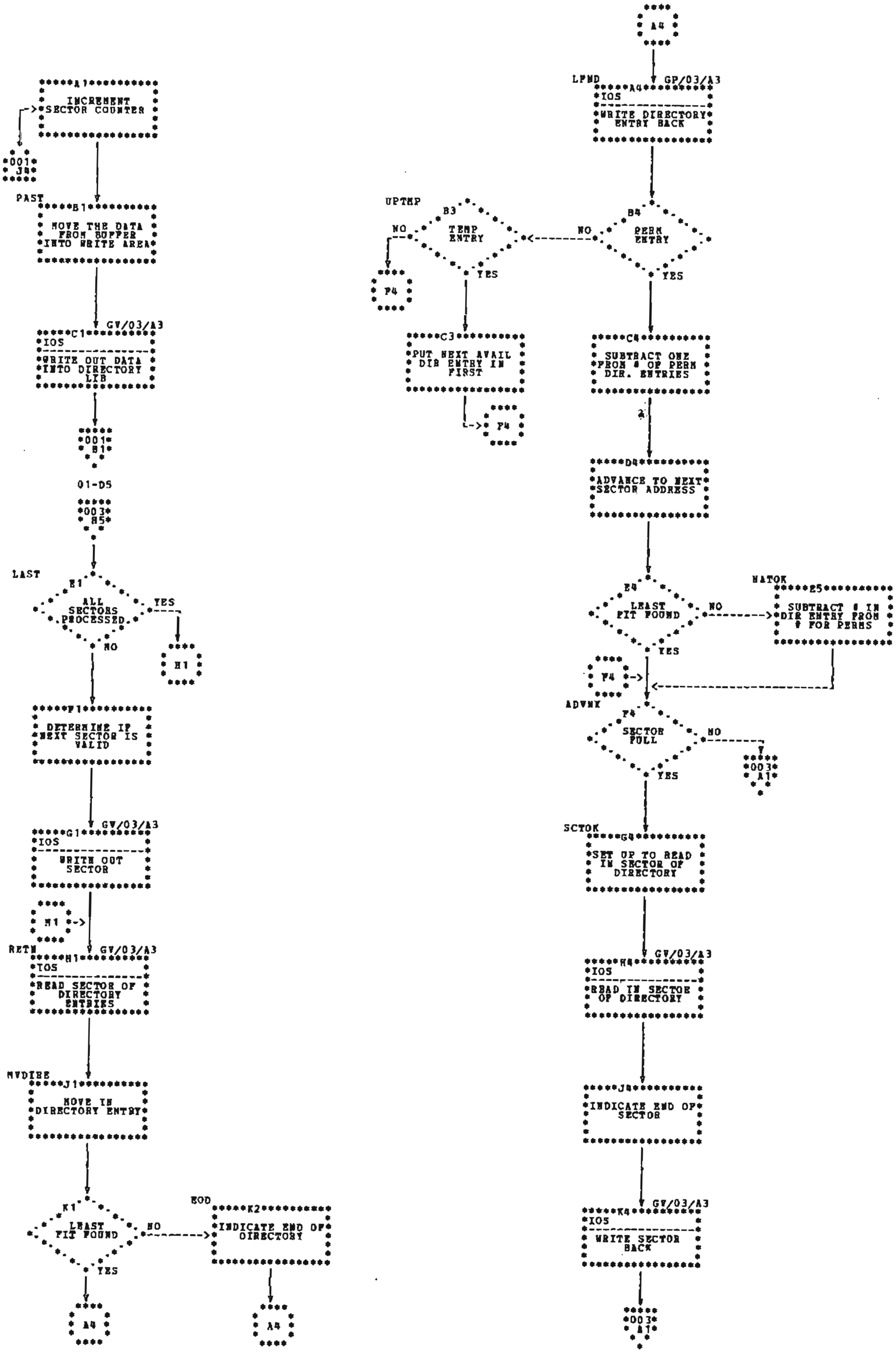
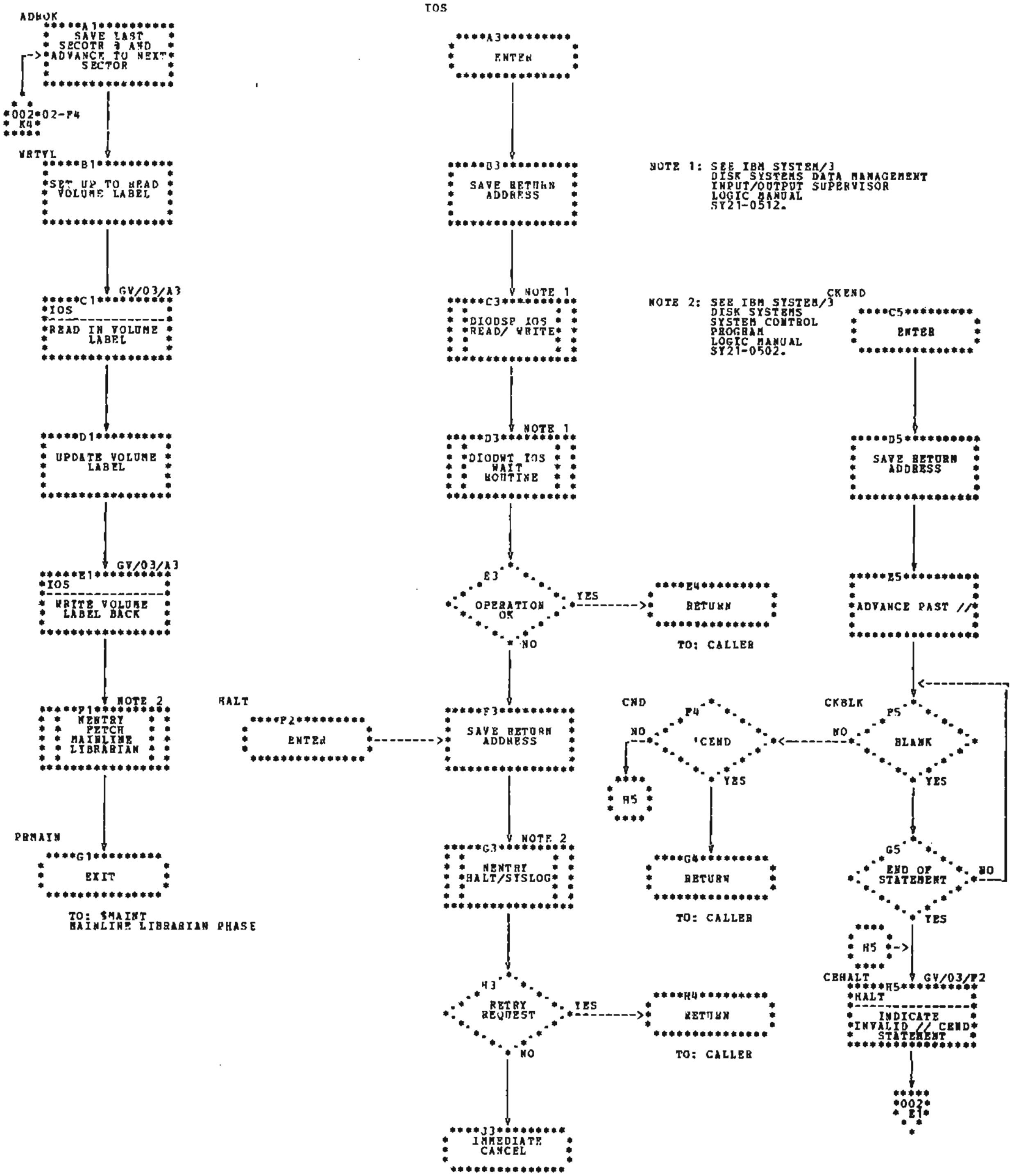


Chart GV (Part 2 of 3). Copy Reader to Object Library (O. Entries) Phase



NOTE 1: SEE IBM SYSTEM/3
DISK SYSTEMS DATA MANAGEMENT
INPUT/OUTPUT SUPERVISOR
LOGIC MANUAL
SY21-0512.

NOTE 2: SEE IBM SYSTEM/3
DISK SYSTEMS
SYSTEM CONTROL
PROGRAM
LOGIC MANUAL
SY21-0502.

Chart GV (Part 3 of 3). Copy Reader to Object Library (O. Entries) Phase

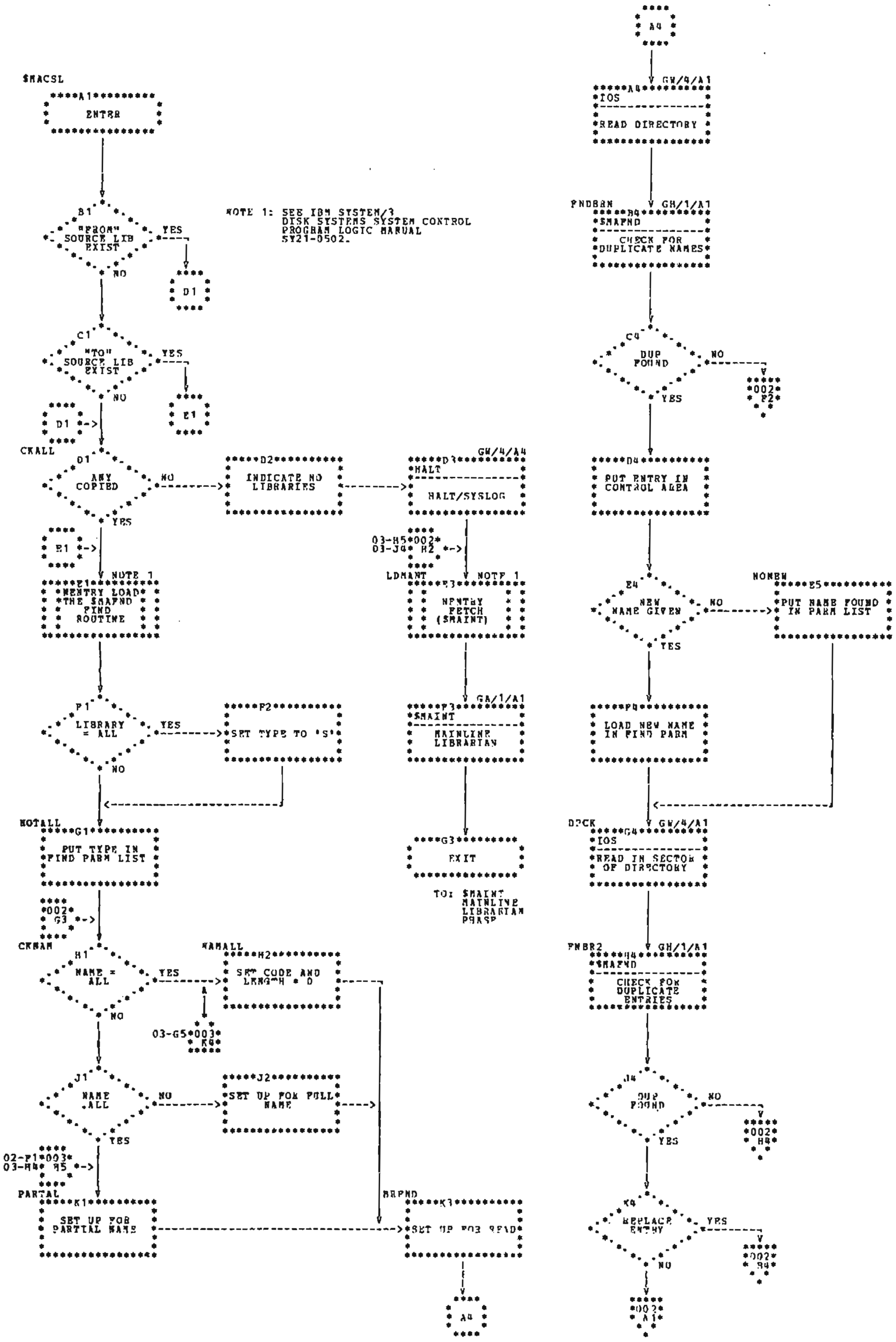


Chart GW (Part 1 of 4). Copy Source Library—Disk to Disk Phase

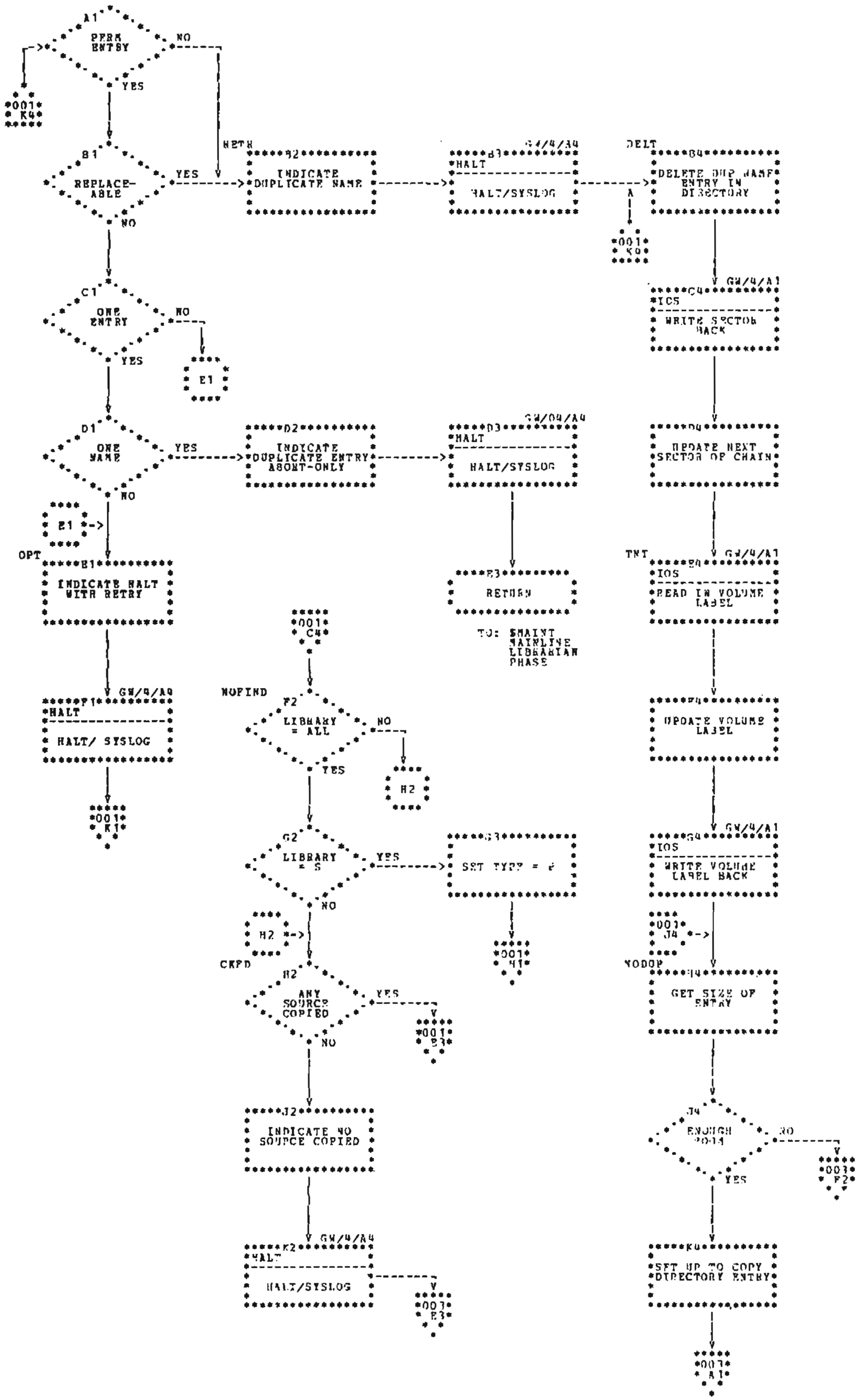


Chart GW (Part 2 of 4). Copy Source Library—Disk to Disk Phase

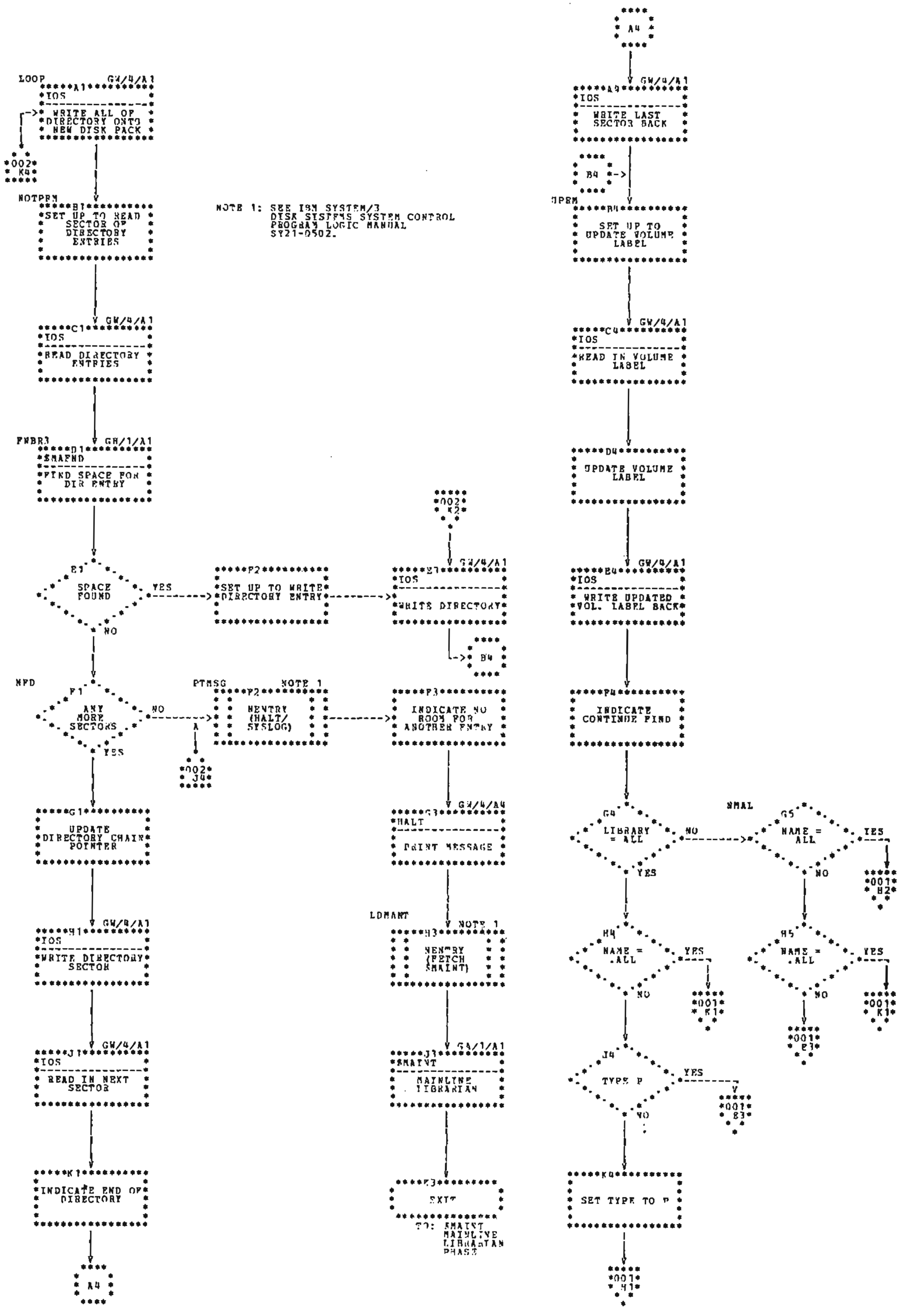
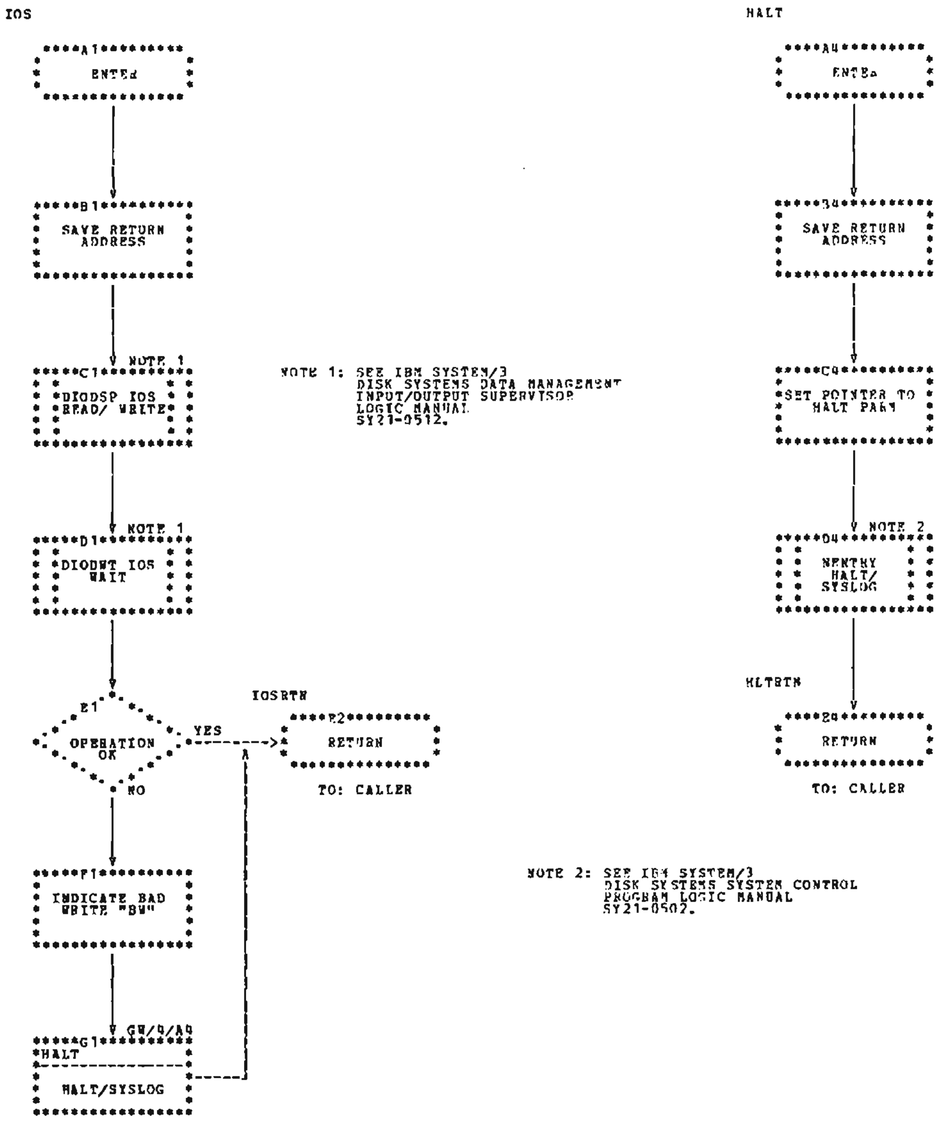


Chart GW (Part 3 of 4). Copy Source Library–Disk to Disk Phase



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS DATA MANAGEMENT INPUT/OUTPUT SUPERVISOR LOGIC MANUAL SY21-0512.

NOTE 2: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502.

Chart GW (Part 4 of 4). Copy Source Library–Disk to Disk Phase

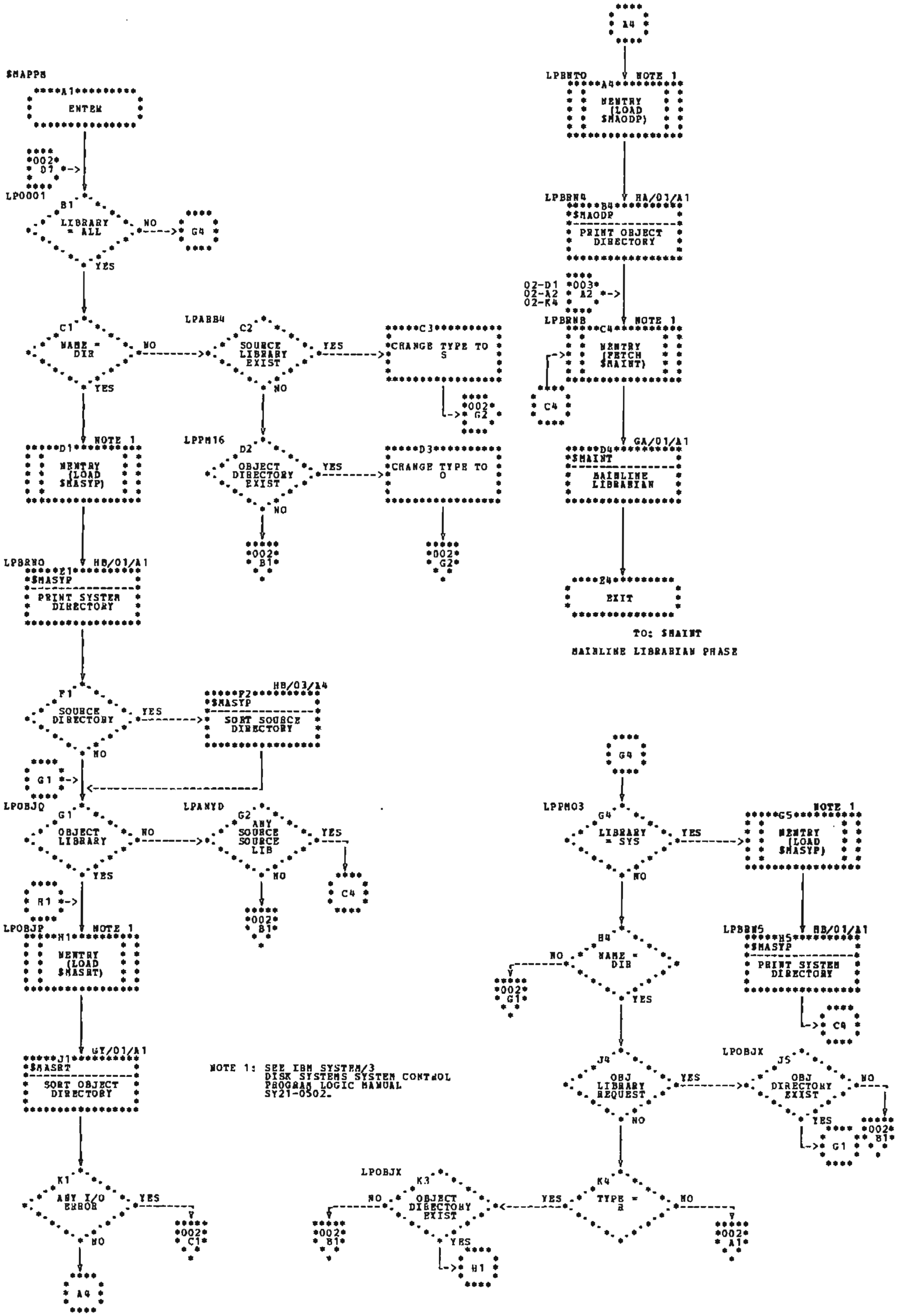
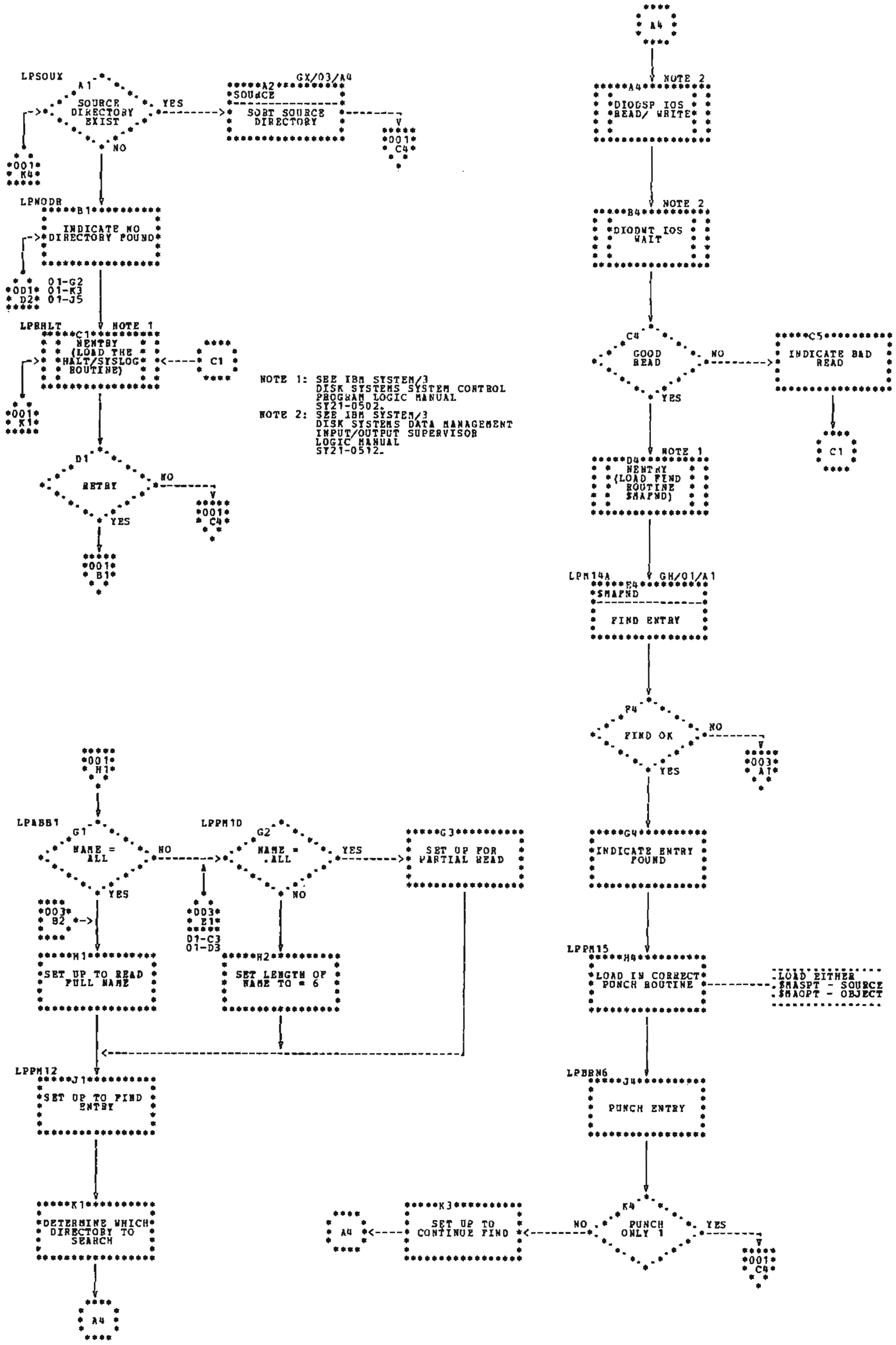


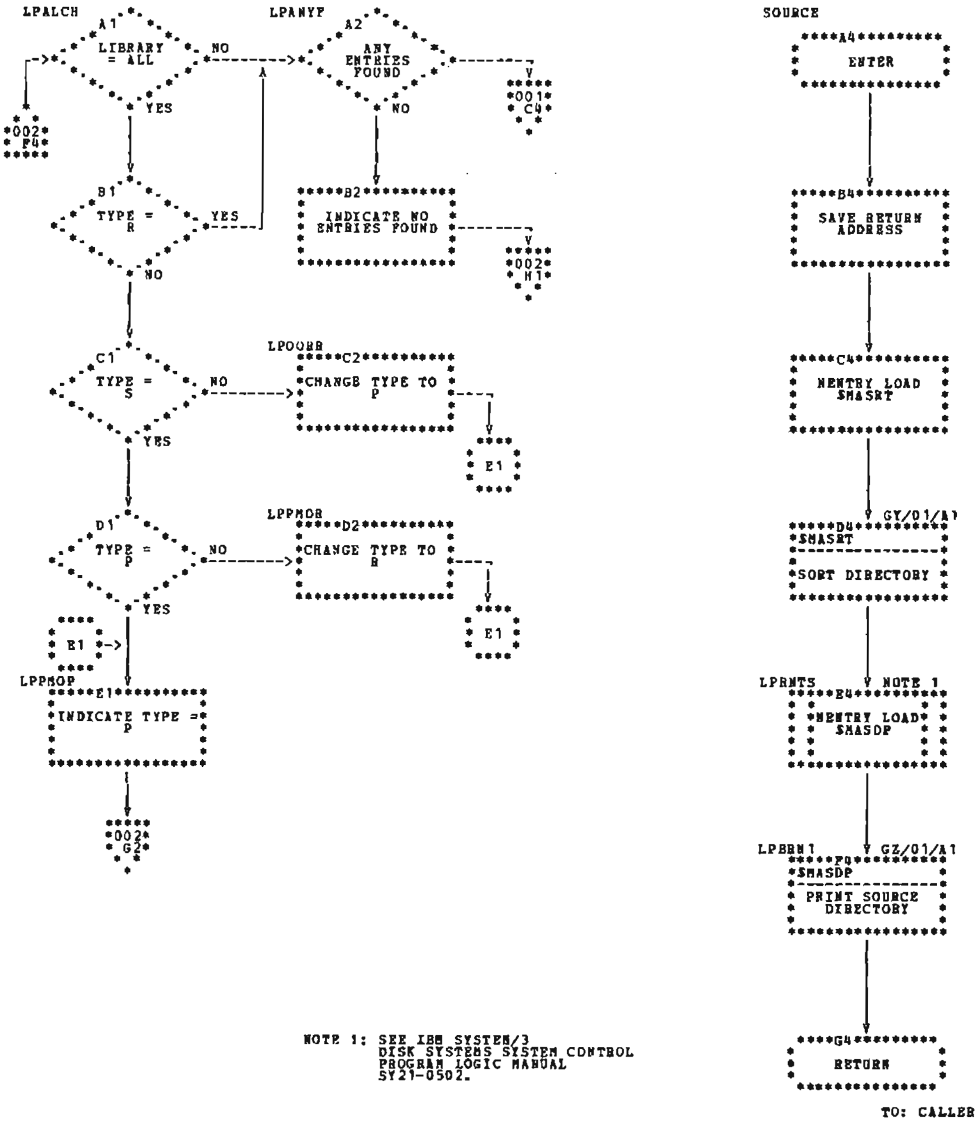
Chart GX (Part 1 of 3). Print/Punch Mainline Phase



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502.

NOTE 2: SEE IBM SYSTEM/3 DISK SYSTEMS DATA MANAGEMENT INPUT/OUTPUT SUPERVISOR LOGIC MANUAL SY21-0512.

Chart GX (Part 2 of 3). Print/Punch Mainline Phase



NOTE 1: SEE IBM SYSTEM/3
DISK SYSTEMS SYSTEM CONTROL
PROGRAM LOGIC MANUAL
SY21-0502.

Chart GX (Part 3 of 3). Print/Punch Mainline Phase

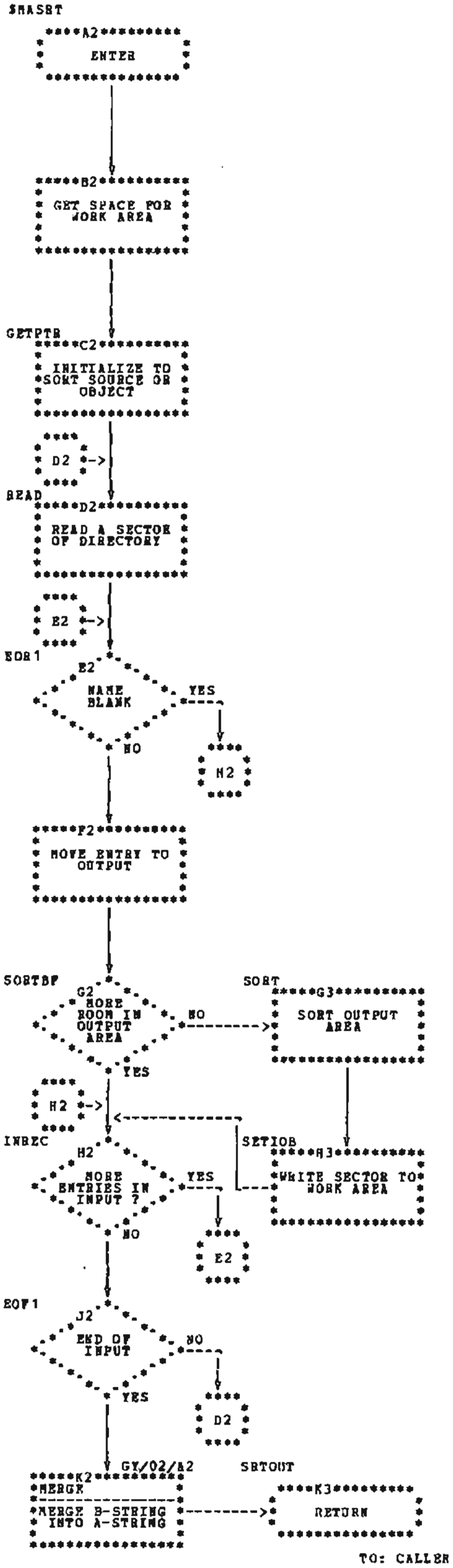


Chart GY (Part 1 of 3). Directory Sort Phase

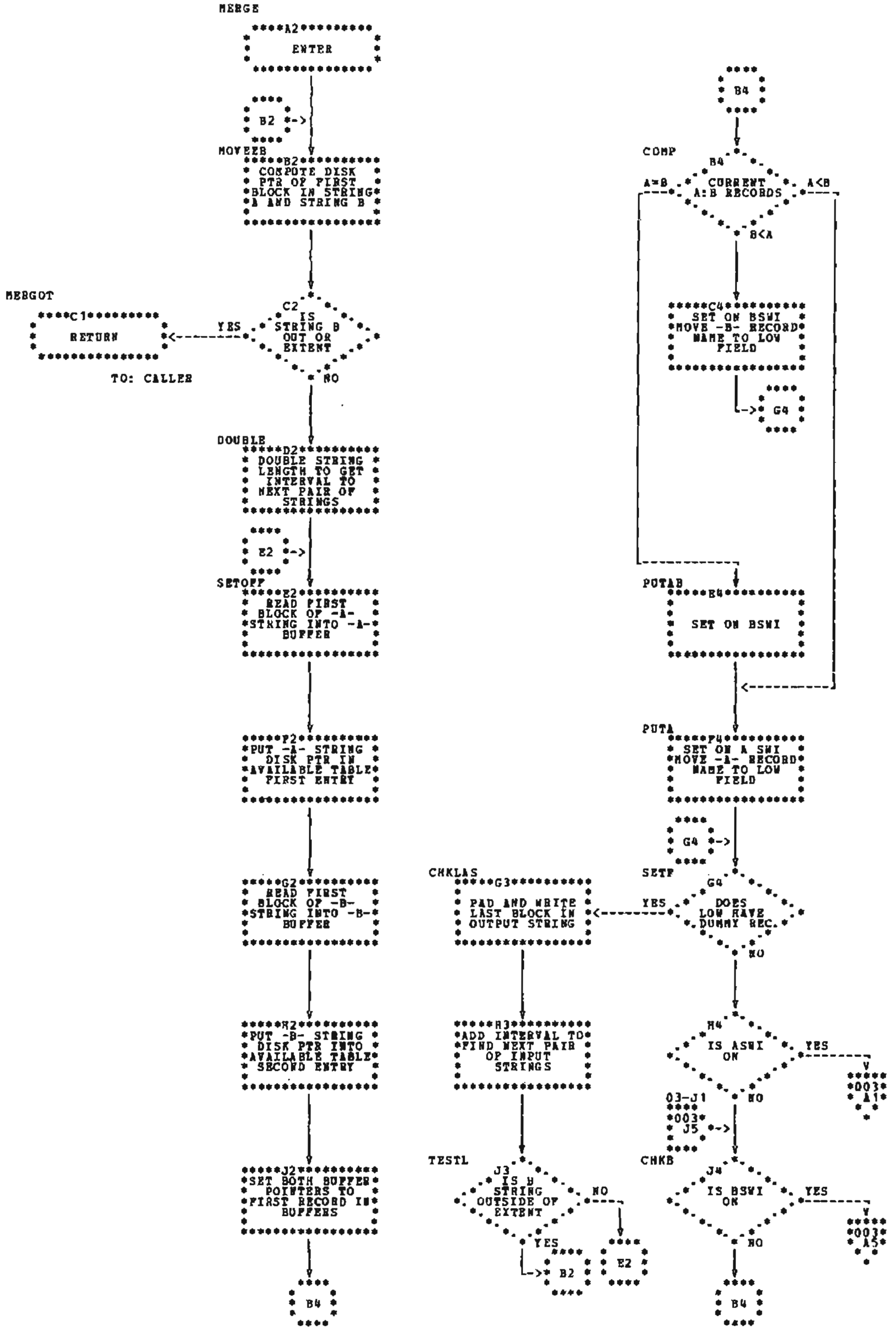

MERGE
A2
ENTER
B2
MOVEER
B2
COMPUTE DISK PTR OF FIRST BLOCK IN STRING A AND STRING B
MERGOT
C1
RETURN
TO: CALLER
C2
IS STRING B OUT OR EXTENT
YES
NO
DOUBLE
D2
DOUBLE STRING LENGTH TO GET INTERVAL TO NEXT PAIR OF STRINGS
E2
SETOFF
E2
READ FIRST BLOCK OF -A- STRING INTO -A- BUFFER
F2
PUT -A- STRING DISK PTR IN AVAILABLE TABLE FIRST ENTRY
G2
READ FIRST BLOCK OF -B- STRING INTO -B- BUFFER
H2
PUT -B- STRING DISK PTR INTO AVAILABLE TABLE SECOND ENTRY
J2
SET BOTH BUFFER POINTERS TO FIRST RECORD IN BUFFERS
B4
B4
COMP
B4
CURRENT A:B RECORDS
A=B
A<B
B<A
C4
SET ON BSWI MOVE -B- RECORD NAME TO LOW FIELD
G4
PUTAB
E4
SET ON BSWI
PUTA
F4
SET ON A SWI MOVE -A- RECORD NAME TO LOW FIELD
G4
SETF
G4
DOES LOW HAVE DUMMY REC.
YES
NO
CHKLAS
G3
PAD AND WRITE LAST BLOCK IN OUTPUT STRING
H3
ADD INTERVAL TO FIND NEXT PAIR OF INPUT STRINGS
TESTL
J3
IS B STRING OUTSIDE OF EXTENT
NO
YES
B2
E2
H4
IS ASWI ON
YES
003 A1
NO
03-J1
003 J5
CHKB
J4
IS BSWI ON
YES
003 A5
NO
B4

Chart GY (Part 2 of 3). Directory Sort Phase

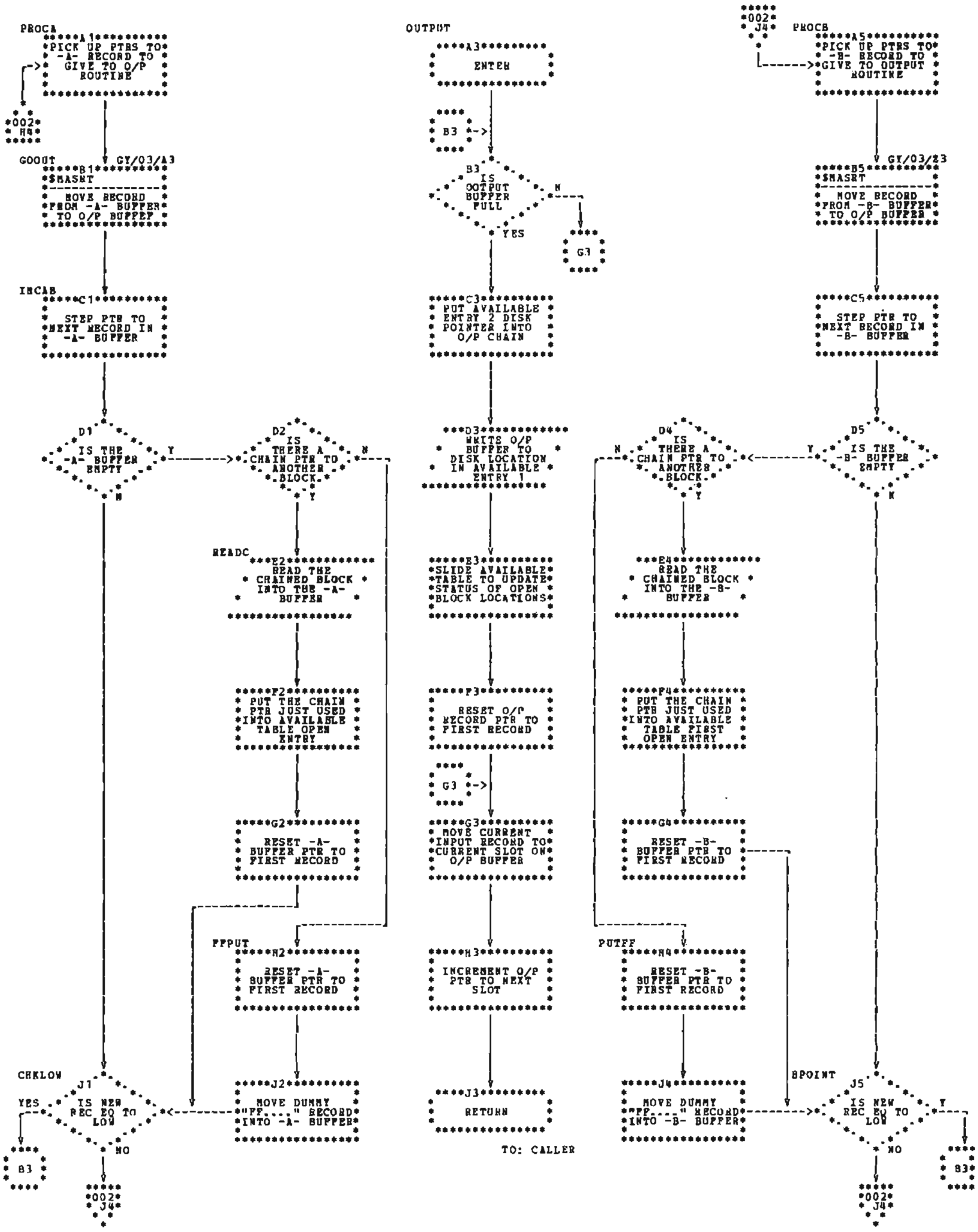

PROCA
A1
PICK UP PTRS TO -A- RECORD TO GIVE TO O/P ROUTINE
002 H4
GOOUT
GY/03/A3
B1
$HASRT
MOVE RECORD FROM -A- BUFFER TO O/P BUFFER
INCAB
C1
STEP PTR TO NEXT RECORD IN -A- BUFFER
D1
IS THE -A- BUFFER EMPTY
Y
N
D2
IS THERE A CHAIN PTR TO ANOTHER BLOCK
N
Y
READC
E2
READ THE CHAINED BLOCK INTO THE -A- BUFFER
F2
PUT THE CHAIN PTR JUST USED INTO AVAILABLE TABLE OPEN ENTRY
G2
RESET -A- BUFFER PTR TO FIRST RECORD
FFPUT
H2
RESET -A- BUFFER PTR TO FIRST RECORD
J2
MOVE DUMMY "FF...." RECORD INTO -A- BUFFER
CHKLOW
J1
IS NEW REC EQ TO LOW
YES
NO
B3
002 J4
OUTPUT
A3
ENTER
B3
B3
IS OUTPUT BUFFER FULL
N
YES
G3
C3
PUT AVAILABLE ENTRY 2 DISK POINTER INTO O/P CHAIN
D3
WRITE O/P BUFFER TO DISK LOCATION IN AVAILABLE ENTRY 1
E3
SLIDE AVAILABLE TABLE TO UPDATE STATUS OF OPEN BLOCK LOCATIONS
F3
RESET O/P RECORD PTR TO FIRST RECORD
G3
G3
MOVE CURRENT INPUT RECORD TO CURRENT SLOT ON O/P BUFFER
H3
INCREMENT O/P PTR TO NEXT SLOT
J3
RETURN
TO: CALLER
002 J4
PROCB
A5
PICK UP PTRS TO -B- RECORD TO GIVE TO OUTPUT ROUTINE
GY/03/E3
B5
$HASRT
MOVE RECORD FROM -B- BUFFER TO O/P BUFFER
C5
STEP PTR TO NEXT RECORD IN -B- BUFFER
D5
IS THE -B- BUFFER EMPTY
Y
N
D4
IS THERE A CHAIN PTR TO ANOTHER BLOCK
N
Y
E4
READ THE CHAINED BLOCK INTO THE -B- BUFFER
F4
PUT THE CHAIN PTR JUST USED INTO AVAILABLE TABLE FIRST OPEN ENTRY
G4
RESET -B- BUFFER PTR TO FIRST RECORD
PUTFF
H4
RESET -B- BUFFER PTR TO FIRST RECORD
J4
MOVE DUMMY "FF...." RECORD INTO -B- BUFFER
BPOINT
J5
IS NEW REC EQ TO LOW
Y
NO
B3
002 J4

Chart GY (Part 3 of 3). Directory Sort Phase

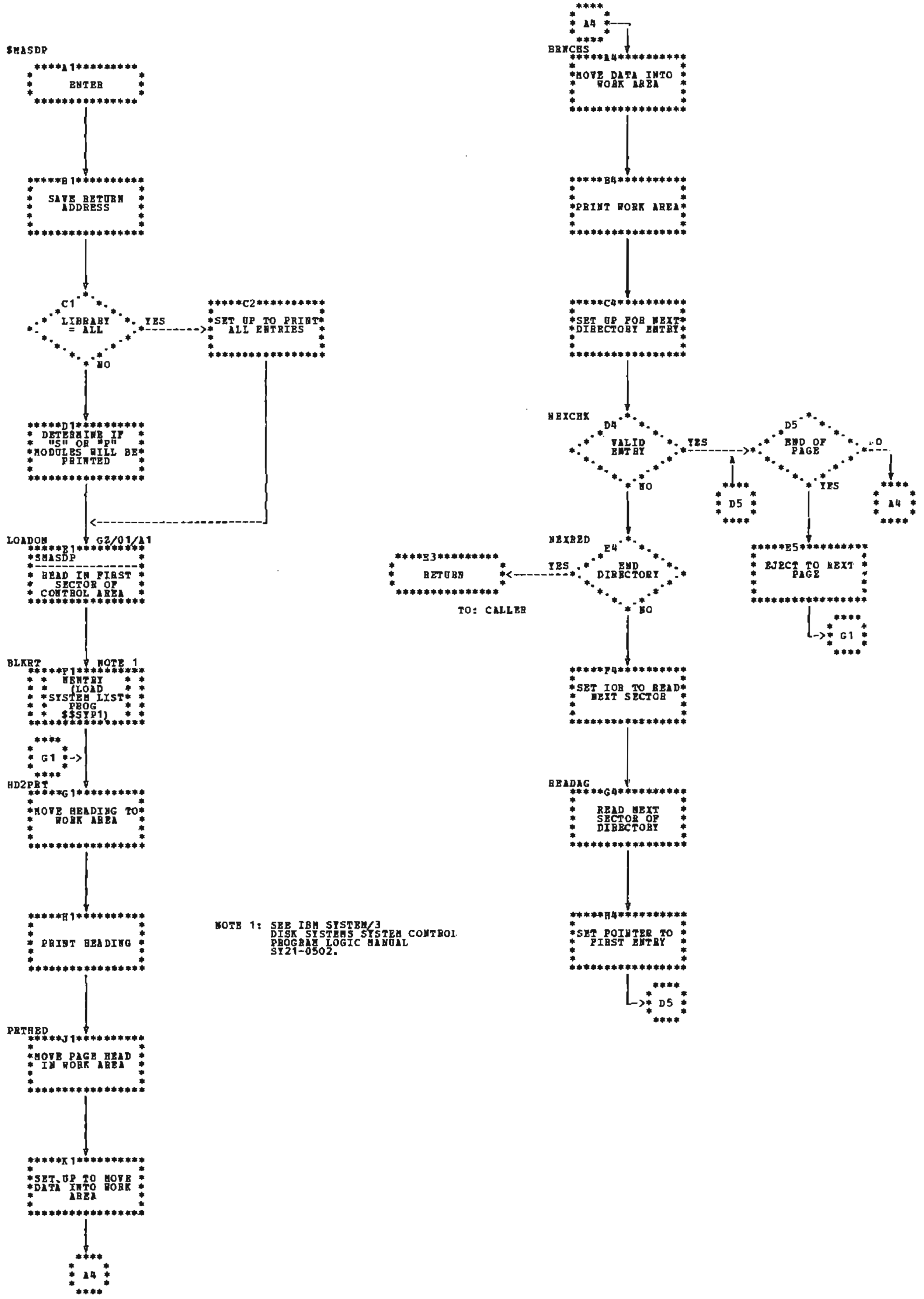

$MASDP
A1
ENTER
B1
SAVE RETURN ADDRESS
C1
LIBRARY = ALL
YES
NO
C2
SET UP TO PRINT ALL ENTRIES
D1
DETERMINE IF "S" OR "P" MODULES WILL BE PRINTED
LOADON
GZ/01/A1
E1
$MASDP
READ IN FIRST SECTOR OF CONTROL AREA
BLKRT
NOTE 1
F1
MENTRY (LOAD SYSTEM LIST PROG $$SYP1)
G1
HD2PRT
G1
MOVE HEADING TO WORK AREA
H1
PRINT HEADING
PRTHED
J1
MOVE PAGE HEAD IN WORK AREA
K1
SET UP TO MOVE DATA INTO WORK AREA
A4
NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL SY21-0502.
A4
BRNCHS
A4
MOVE DATA INTO WORK AREA
B4
PRINT WORK AREA
C4
SET UP FOR NEXT DIRECTORY ENTRY
NEXCHK
D4
VALID ENTRY
YES
NO
D5
END OF PAGE
NO
YES
D5
A4
NEXRED
E4
END DIRECTORY
YES
NO
E3
RETURN
TO: CALLER
E5
EJECT TO NEXT PAGE
G1
F4
SET IOB TO READ NEXT SECTOR
READAG
G4
READ NEXT SECTOR OF DIRECTORY
H4
SET POINTER TO FIRST ENTRY
D5

Chart GZ. Print Source Directory Phase

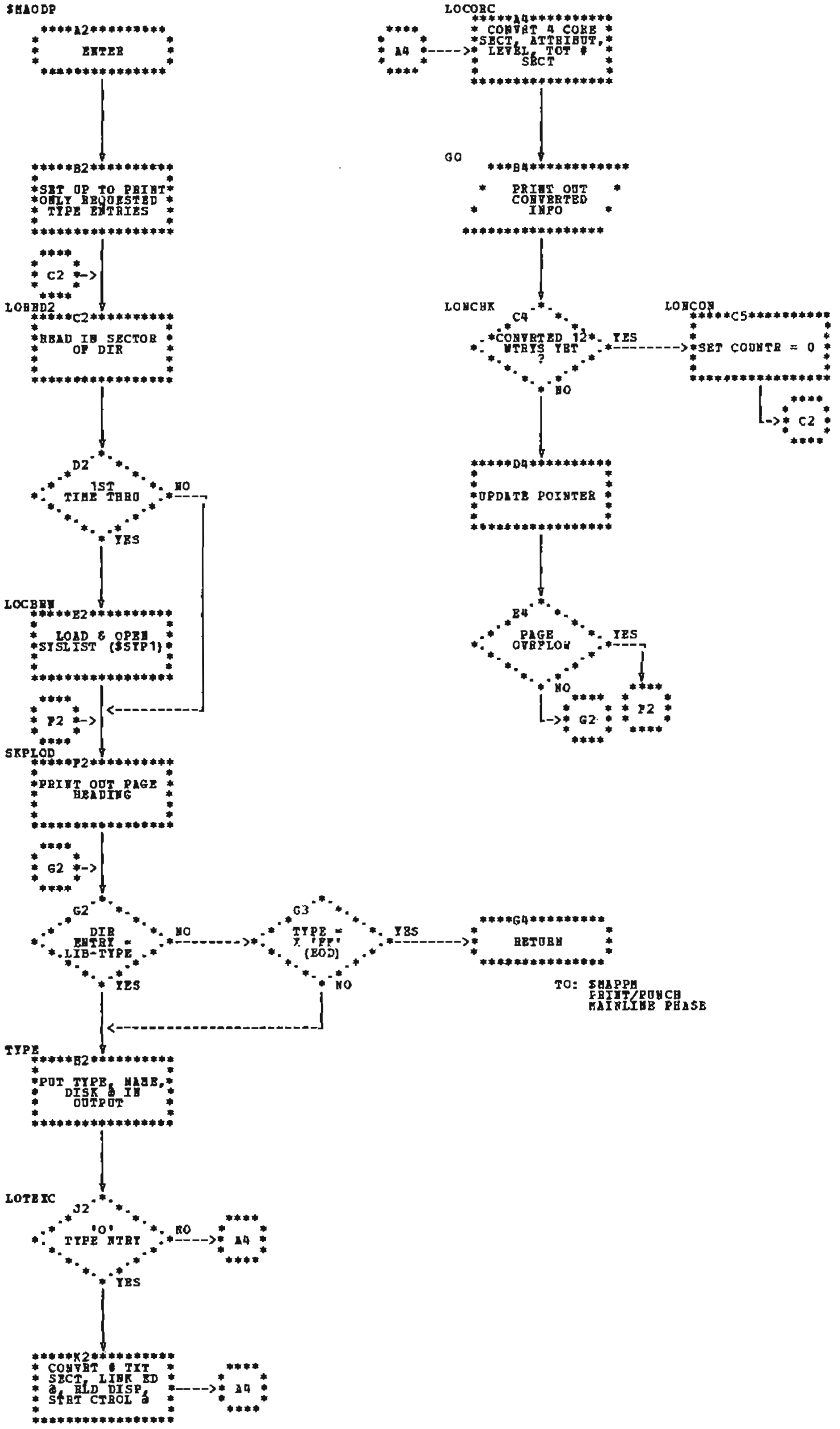
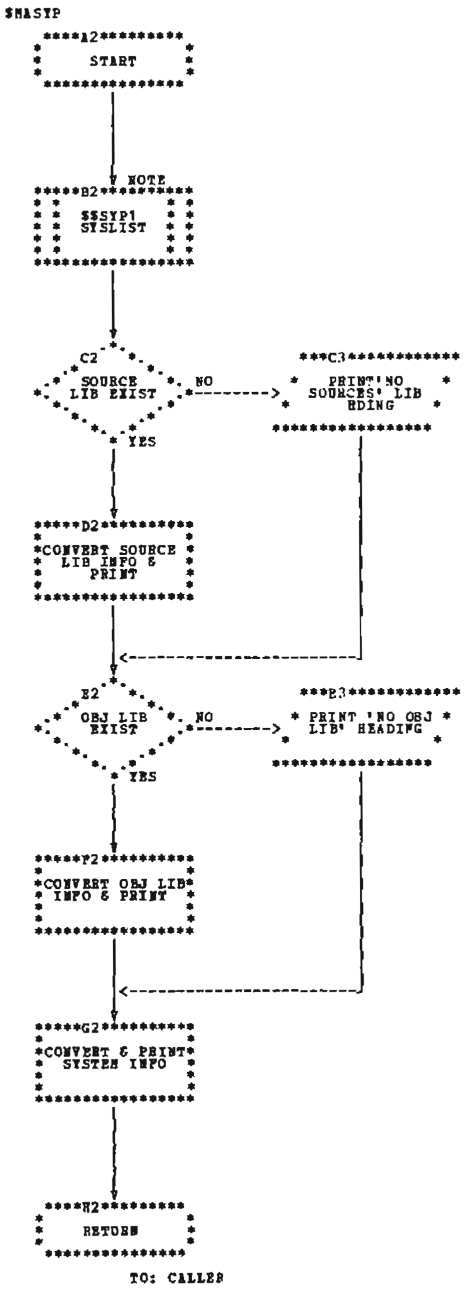


Chart HA. Print Object Directory Phase


$MASYP
A2
START
B2
NOTE
$$SYP1
SYSLIST
C2
SOURCE LIB EXIST
NO
YES
C3
PRINT'NO SOURCES' LIB HDING
D2
CONVERT SOURCE LIB INFO & PRINT
E2
OBJ LIB EXIST
NO
YES
E3
PRINT 'NO OBJ LIB' HEADING
F2
CONVERT OBJ LIB INFO & PRINT
G2
CONVERT & PRINT SYSTEM INFO
H2
RETURN
TO: CALLER

NOTE: SEE IBM SYSTEM/3
DISK SYSTEMS CONTROL
PROGRAM LOGIC MANUAL
SY21-0502.

Chart HB. Print System Directory Phase

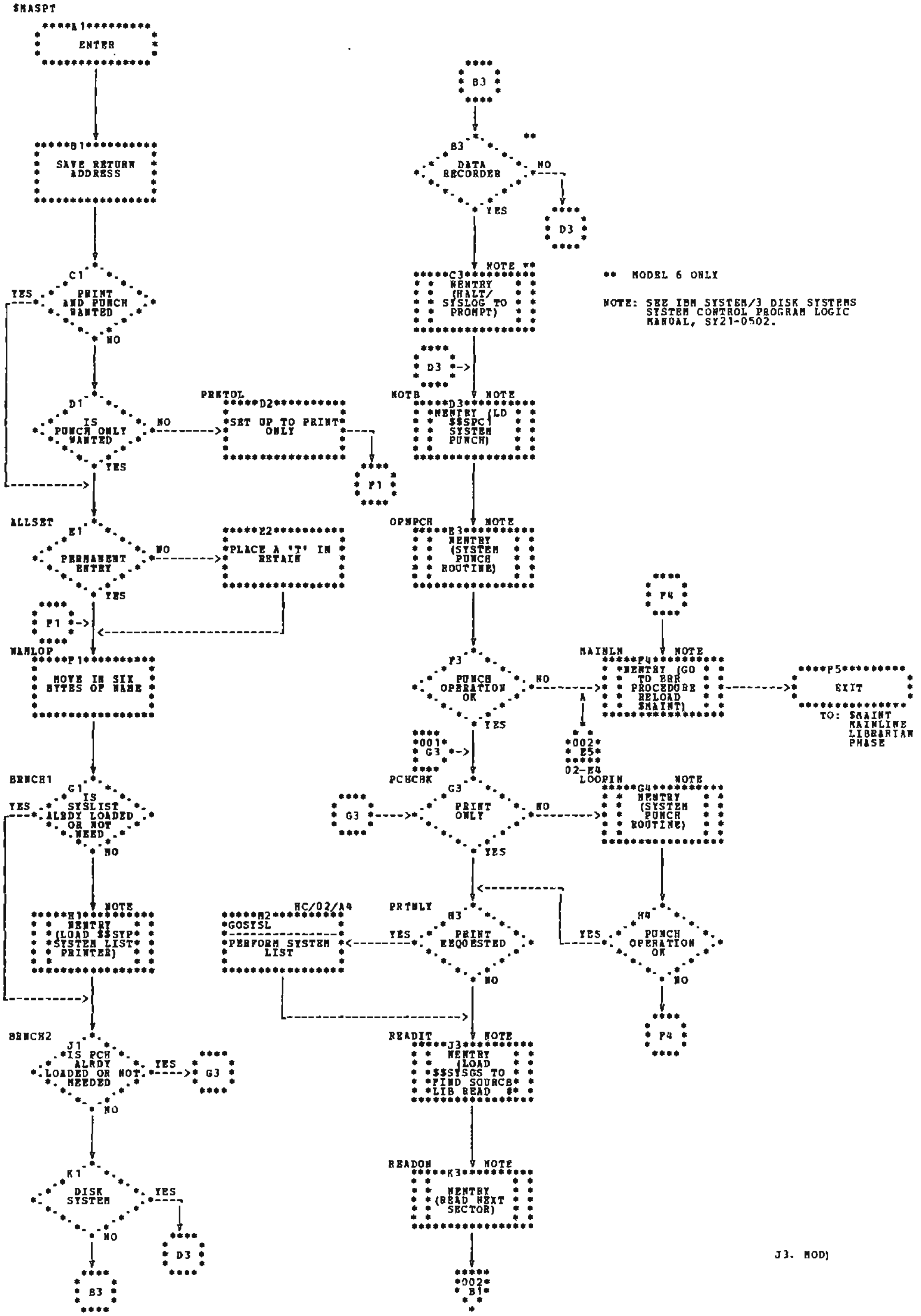
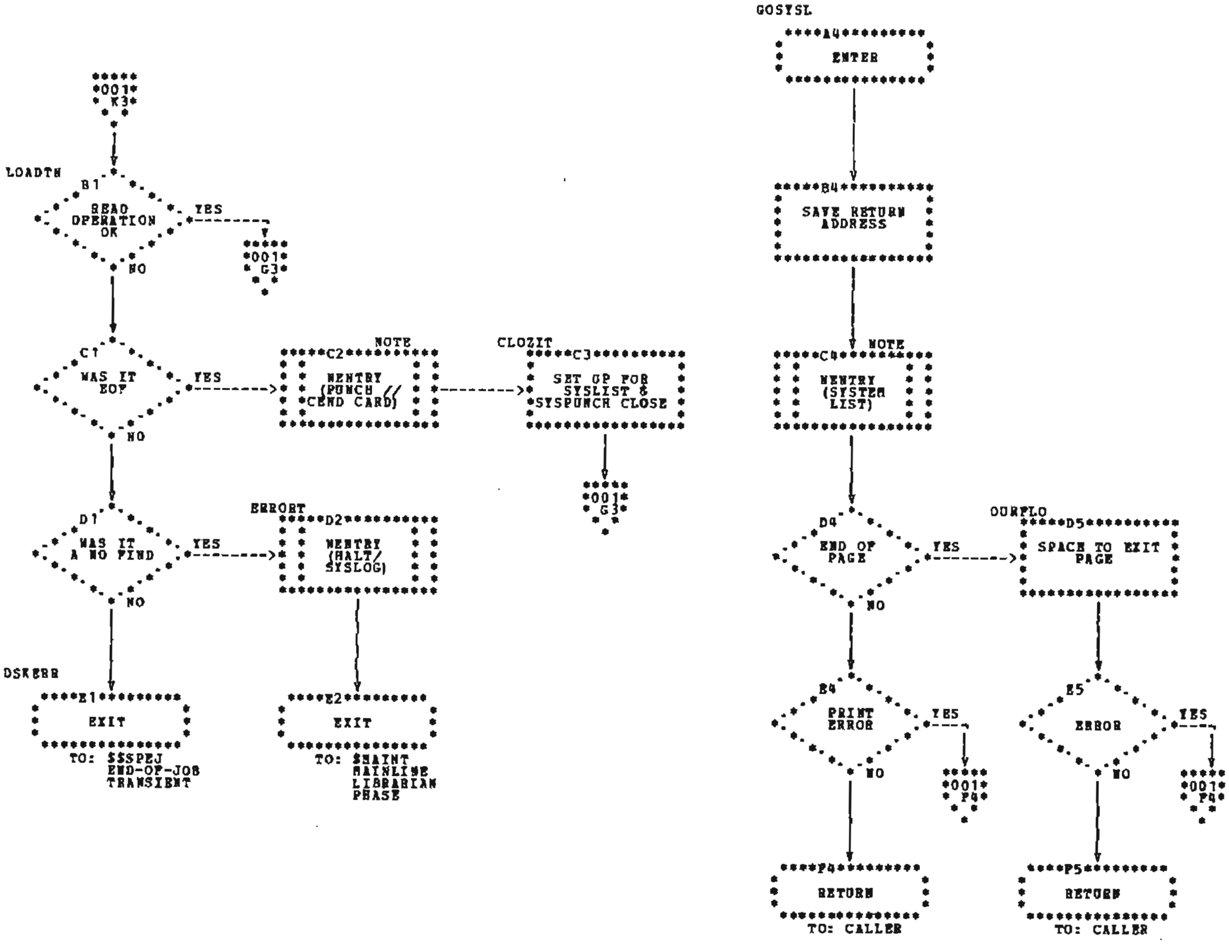
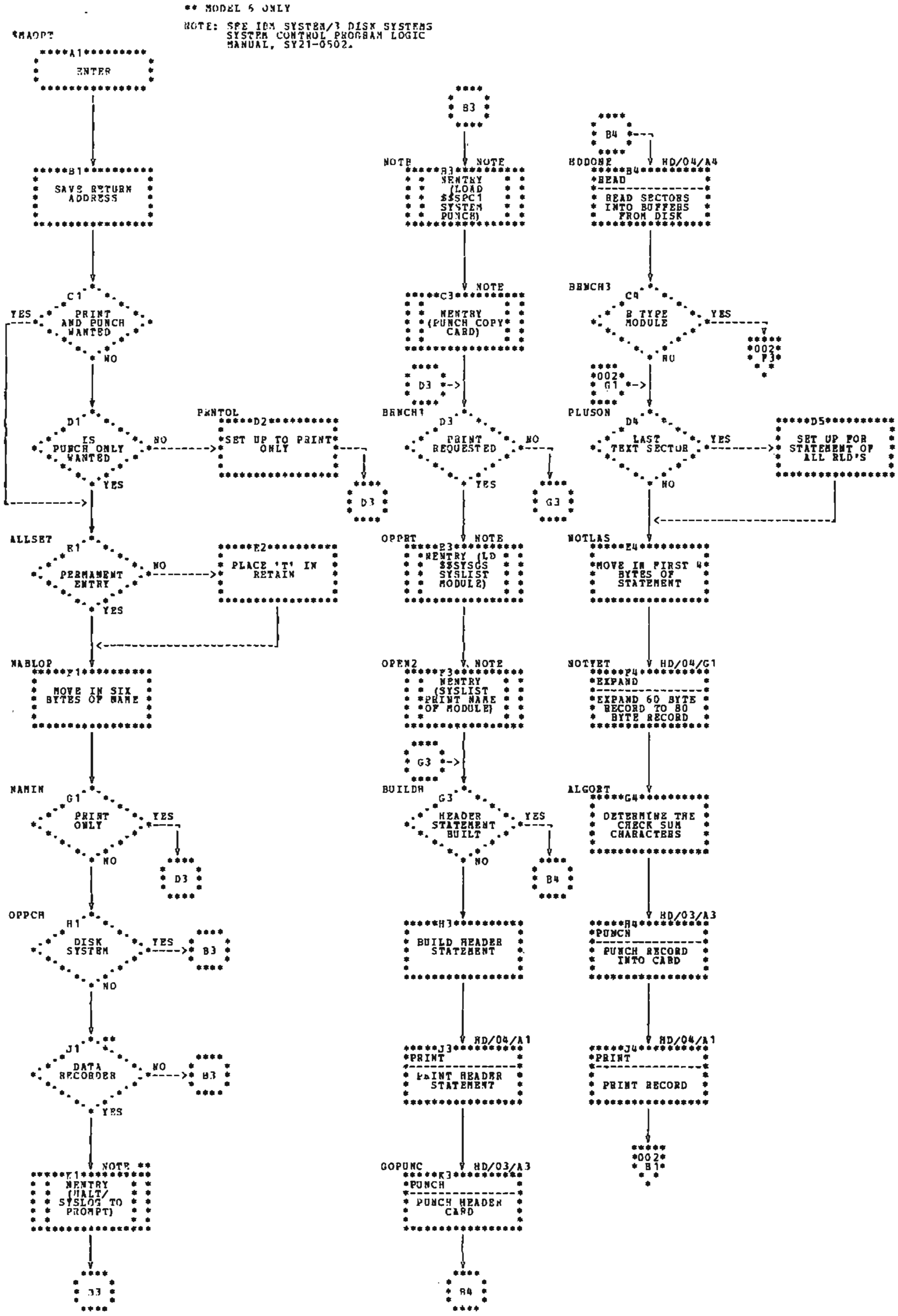


Chart HC (Part 1 of 2). Print/Punch Source Library Phase



NOTE: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

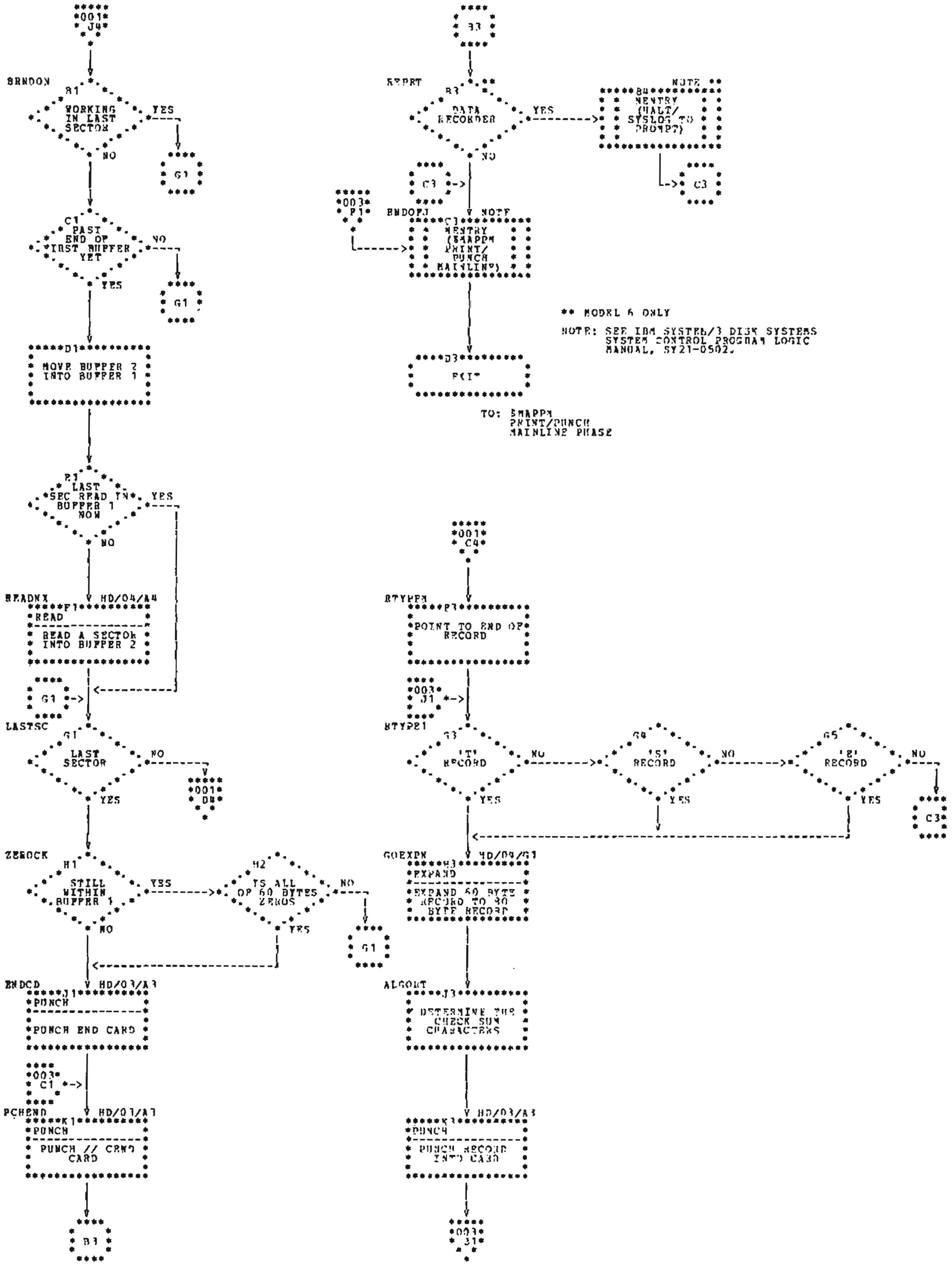
Chart HC (Part 2 of 2). Print/Punch Source Library Phase



** MODEL 6 ONLY

NOTE: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

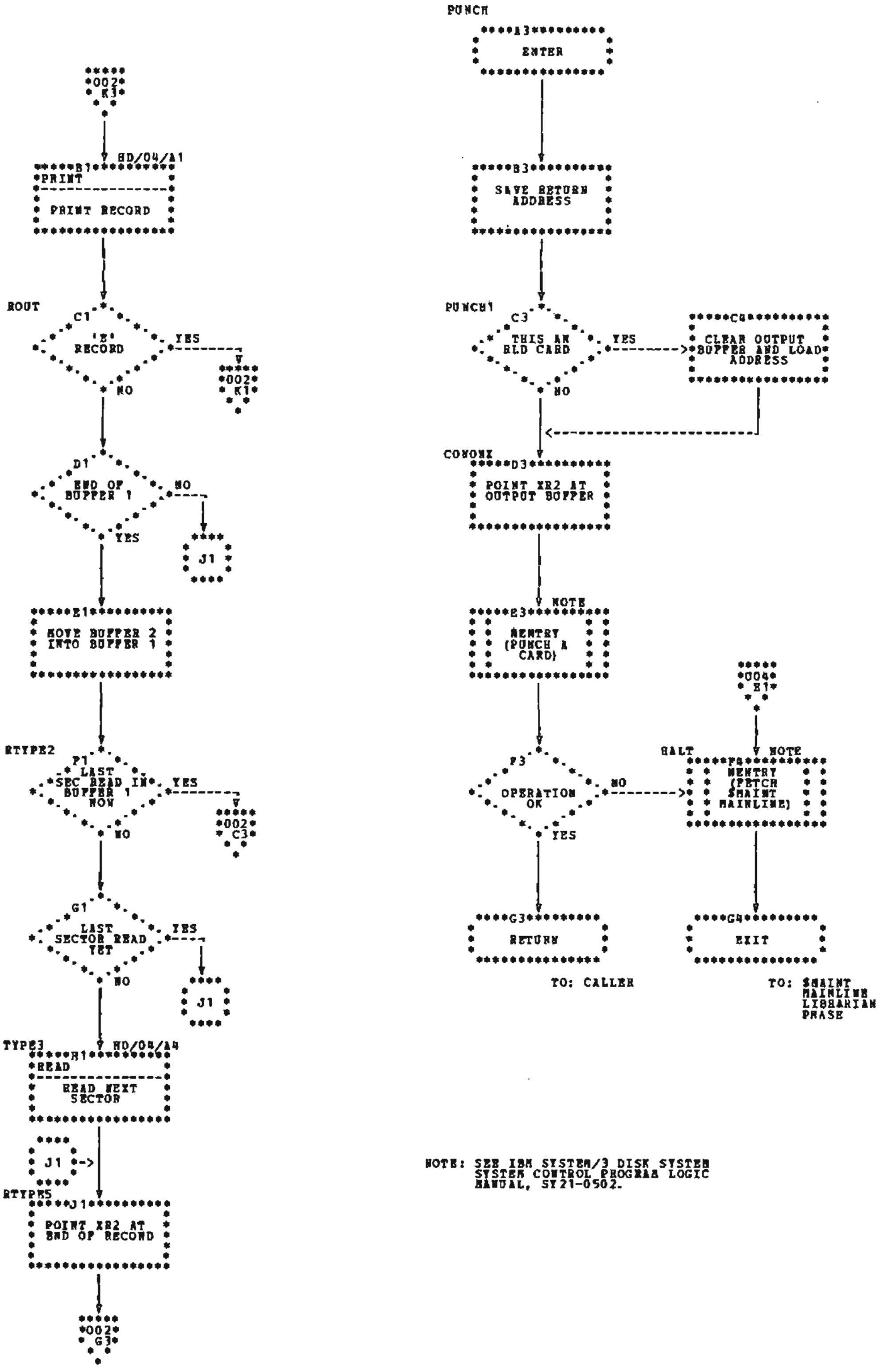
Chart HD (Part 1 of 4). Print/Punch Object Library Phase



** MODEL 6 ONLY

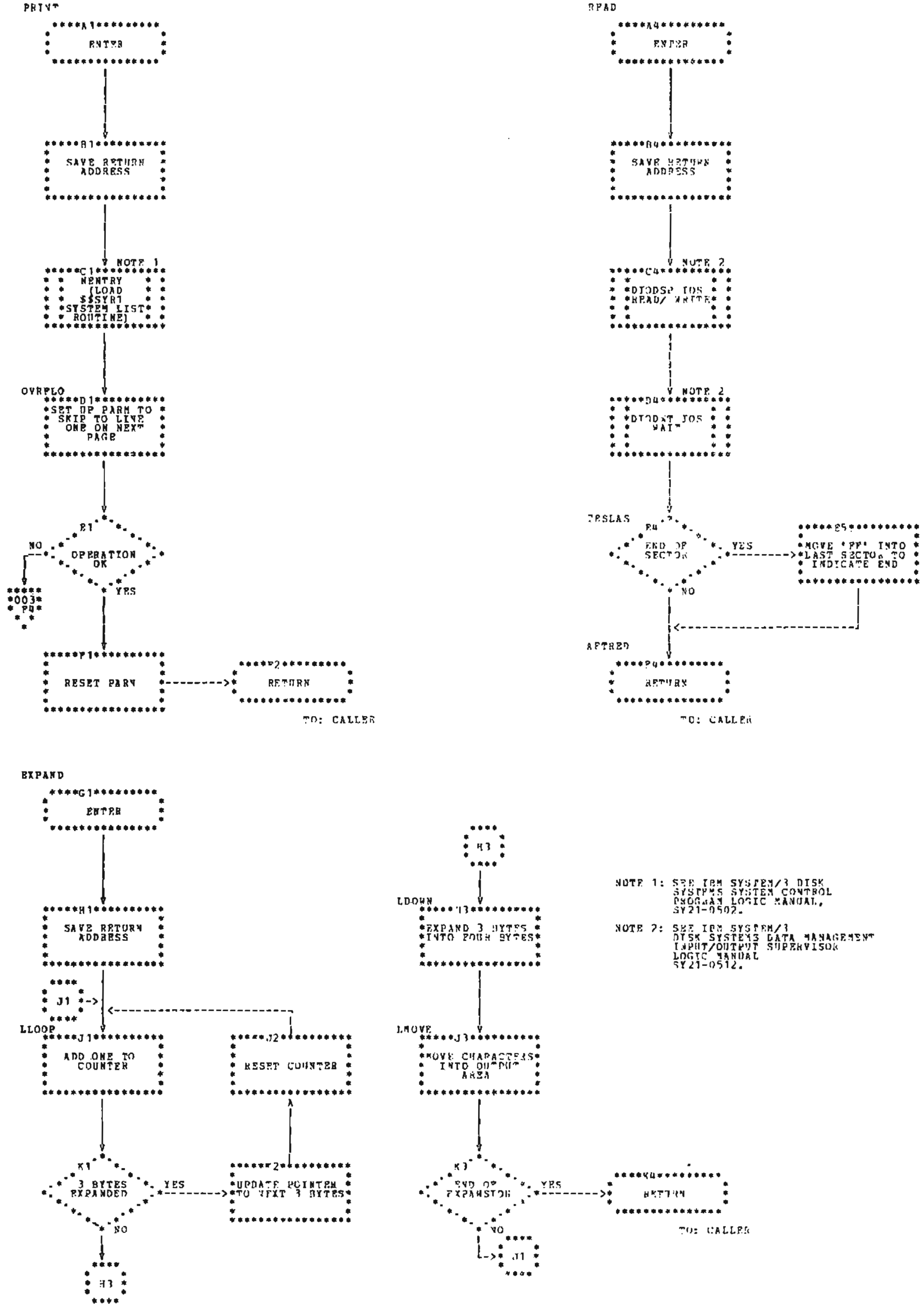
NOTE: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart HD (Part 2 of 4). Print/Punch Object Library Phase



NOTE: SEE IBM SYSTEM/3 DISK SYSTEM SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

Chart HD (Part 3 of 4). Print/Punch Object Library Phase



NOTE 1: SEE IBM SYSTEM/3 DISK SYSTEMS SYSTEM CONTROL PROGRAM LOGIC MANUAL, SY21-0502.

NOTE 2: SEE IBM SYSTEM/3 DISK SYSTEMS DATA MANAGEMENT INPUT/OUTPUT SUPERVISOR LOGIC MANUAL SY21-0512.

Chart HD (Part 4 of 4). Print/Punch Object Library Phase

Section 4. Data Area Formats

The data areas discussed in this section are used by more than one phase in the Library Maintenance program.

CONTROL AREA (CAREA)

This is a 256-byte area immediately following the supervisor that is built by the Mainline Librarian phase ($MAINT) and Allocate Syntax Scan phase ($MAALS). This area is used to supply the rest of the Library Maintenance program with the parameters necessary for their operation. The control area has five sections:

1. Library control area

2. From control area (source section)

3. From control area (object section)

4. To control area (source section)

5. To control area (object section)

Figure 7-17 shows the format of CAREA.

Segment Name	Bytes	Contents
VERB	1	X'80' = Copy X'40' = Delete X'20' = Allocate X'10' = Rename
FROM –	1	X'A8' = F1 X'B8' = F2 X'A0' = R1 X'B0' = R2 X'40' = Reader
LIBRARY –	1	X'E2' = Source statement X'D7' = Procedure X'D6' = Executable object code X'D9' = Subroutines in nonexecutable object form X'FF' = All libraries X'F0' = System (system directory print only)
NAME*	6	Module name (6 bytes) or Name.ALL (1-5 bytes) (6th byte = length) Name (partial name is left adjusted) .X'0n' (length of partial name is in rightmost byte) Example: Name – $$$.ALL = $$$ƀƀ03 If Name = ALL, X'08' is in the leftmost byte If Name = DIR (directory), X'04' is in the leftmost byte If Name = SYS (system), X'02' is in the leftmost byte
	2	Existing source library size (reallocate)
	1	Scheduler work area size (reallocate)
	1	Roll-in/roll-out size (reallocate)
	2	Existing object library size (reallocate)
TO –	1	X'A8' = F1 X'B8' = F2 X'A0' = R1 X'B0' = R2 X'08' = Print X'04' = Punch X'02' = Print/Punch
RETAIN –	2	X'D3nn' = Level where nn is the level (1-99) specified of the control card X'80' = Permanent X'00' = Temporary X'D9' = Replace
NEWNAME*	6	Module name
SOURCE	2	X'nnnn' = Size X'FF' = No change X'F0' = Deallocate X'F8' = Reallocate Leftmost byte
OBJECT	2	X'nnnn' = Size X'FF' = No change X'F0' = Deallocate X'F8' = Reallocate Leftmost byte

* Either these six bytes or the six bytes following are in storage.

Figure 7-17. Library Control Area (Part 1 of 4)

Segment Name	Bytes	Contents
SYSTEM	1	X'80' = Yes X'00' = No
WORK	1	X'A8' = F1 X'B8' = F2 X'A0' = R1 X'B0' = R2
LEFROM	0	Library From area (source)
LEFVTO	2	VTOC pointer
LEFDPN	2	Directory pointer
LEFNAS	2	Next available library sector
LEFEOL	2	End of library
LEFDRS	2	Number of directory sectors
LEFPLS	2	Number of permanent library sectors
LEFACT	2	Number of active library sectors
LEFAVL	2	Number of available library sectors
LEFRSS	12	Reserved
LEFOBJ	0	Library From area (object)
LEFDRP	2	Directory pointer
LEFEOD	2	End of directory
LEFSOL	2	Start of library
LEFAEL	2	Allocated end of library
LEFEEL	2	Extended end of library
LEFAPE	2	Number of available permanent directories
LEFATE	2	Number of available temporary directories
LEFFTD	3	First temporary directory entry
LEFNAT	3	Next available temporary directory entry
LEFNAL	2	Next available library sector for permanent entries
LEFALT	2	Next available library sector for temporary entries
LEFASP	2	Number of available library sectors for permanent entries
LEFAST	2	Number of available library sectors for temporary entries
LEFALS	2	Number of active library sectors
LEFAOP	2	Number of active O-type permanent sectors
LEFARP	2	Number of active R-type permanent sectors

Figure 7-17. Library Control Area (Part 2 of 4)

Segment Name	Bytes	Contents
LEFRSO	1	Reserved
LEFARR	2	Address of roll-in/roll-out
LEFSRR	1	Size of roll-in/roll-out
LEFASW	2	Address of scheduler work area
LEFSSW	1	Size of scheduler work area
LEFBOL	2	Beginning of libraries
LEFEND	2	End of libraries
LETROM	0	Library To area (source)
LETVTO	2	VTOC pointer
LETDPN	2	Directory pointer
LETNAS	2	Next available library sector
LETEDL	2	End of library entries
LETDRS	2	Number of directory entries
LETPLS	2	Number of permanent library sectors
LETACT	2	Number of active library sectors
LETAVL	2	Number of available library sectors
LETRSS	12	Reserved
LETOBJ	2	Library To area (object)
LETDRP	2	Directory pointer
LETEOD	2	End of directory
LETSOL	2	Start of library
LATAEL	2	Allocated end of library
LETEEL	2	Extended end of library
LETAPE	2	Number of available permanent directories
LETATE	2	Number of available temporary directories
LETFTD	3	First temporary directory entry
LETNAT	3	Next available temporary directory entry
LETNAL	2	Next available library sector for permanent entries
LETALT	2	Next available library sector for temporary entries
LETASP	2	Number of available library sectors for temporary

Figure 7-17. Library Control Area (Part 3 of 4)

Segment Name	Bytes	Contents
LETAST	2	Number of available library sectors for temporary entries
LETALS	2	Number of active library sectors
LETAOP	2	Number of active O-type permanent sectors
LETARP	2	Number of active R-type permanent sectors
LETRSO	1	Reserved
LETARR	2	Address of roll-in/roll-out
LETSRR	1	Size of roll-in/roll-out
LETASW	2	Address of scheduler work area
LETSSW	1	Size of scheduler work area
LETBOL	2	Beginning of libraries
LETEND	2	End of library
LEDIRE	3	Cylinder, sector, and displacement of directory entry
LEFIND**	2	Address of Find phase
LEDENT**	21	Directory entry
LEFSIZ**	2	Size of least fit area
LESDEN***	13	Source directory entry to copy
LESADR****	2	Address of source on work pack
LEDADR****	2	Address of object on work pack

** Used by $MAOLH
*** Used by $MACSL
**** Used by $MAAMW

Figure 7-17. Library Control Area (Part 4 of 4)

OBJECT LIBRARY DIRECTORY

The directory contains entries that point to each object program or routine in the object library. Each entry in the directory is 21 bytes in length. Each sector of the directory contains 12 entries and four unused bytes. Figure 7-18 shows the format of each directory entry.

Bytes	Contents	Description
1	Type	Entry type (O or R)
6	Name	Name of the object program or routine
2	Disk address	Cylinder/sector address of the library entry
1	Number of text sectors	Type O — number of text sectors in the library entry Type R — not used
2	Link edit address	Type O — storage address to which the library entry has been link edited Type R — not used
1	Displacement of relocation directory	Type O — displacement (in bytes) of the relocation directory in the last sector of text Type R — not used
2	Starting control address	Type O — storage address at which program execution will begin Type R — not used
1	Storage size	Amount of storage (in sectors) required to run the program
2	Attributes	Byte 1,Bit 0 0=Temporary entry 1=Permenant entry Bit 1 1=Inquiry Bit 2 1=Inquiry evoking Bit 3 1=Must run dedicated Bit 4 1=Requires source information Bit 5 1-Deferred mounting allowed Bit 6 1=No relocation directories Bit 7 1-PTF applied Byte 2 Reserved
1	Level	Version and modification level of this entry
2	Total number of sectors	Total number of sectors in the library entry

Figure 7-18. Object Library Directory

SOURCE LIBRARY DIRECTORY

The directory contains entries that point to each source program or procedure in the source library. Each entry in the directory is 13 bytes in length. Each sector of the directory contains 19 entries and 9 bytes that point to the cylinder and sector of the next sector in the directory. Figure 7-19 shows the format of each directory entry.

Bytes	Contents	Description
1	Type	Indicates whether entry is source program or procedure
6	Name	Name of the source program or procedure
2	Address of first sector in chain	Cylinder/sector address of first sector in chain
2	Address of last sector in chain	Cylinder/sector address of last sector in chain
2	Number of sectors (attribute)	Number of sectors in chain. The high order bit indicates: 0 = Permanent entry 1 = Temporary entry

Figure 7-19. Source Library Directory

Appendix A. Directory

Disk Initialization Program

Phase Name	*Descriptive Name*	*Entry Point*	*Synopsis*
$INIT	Root phase	IROOT	Loads phases in storage; maintains table of constants.
$INIT1	Control Statement Analysis phase	IDSYN	Reads control statements; converts control statements from decimal to hexadecimal; checks control statements for validity and sequence; builds two parameter tables.
$INIT2	Mainline phase	IMAIN	Obtains parameter table; reads volume label; checks for initialization validity; initializes alternate track information; determines program level; passes parameter to Surface Analysis phase; saves and restores program protection; assigns alternate tracks; verifies sector ID fields; writes and updates volume label.
$INSUR	Surface Analysis phase	SURFAN	Writes and verifies sector addresses; provides error recovery procedure; updates alternate track information; passes return code to Mainline phase.
$INMSG	Print phase	ALMSG	Log results of alternate track assignment; prints addresses of tracks; logs results of initialization.

Alternate Track Assignment Program

Phase Name	*Descriptive Name*	*Entry Point*	*Synopsis*
$ALT	Root phase	AROOT	Loads phases; maintains constants.
$ALT01	Control Statement Analysis phase	ASASYN	Reads and logs control statements; checks statements for validity and sequence; converts statement operands from decimal to hexadecimal; builds table of constants and passes it to Root phase.
$ALT02	Mainline phase	AMAIN	Reads volume label; performs conditional assignment, unconditional assignment, or unconditional cancel of prior assignment.

Phase Name	*Descriptive Name*	*Entry Point*	*Synopsis*
$ALT03	Data Transfer phase	ADATX	Transfers data from primary to secondary track; returns data to primary track; passes table of messages to Print phase.
$ALSUR	Surface Analysis phase	SURFAN	Writes and verifies sector addresses and test patterns; provides error recovery procedure; indicates defective track; performs surface analysis; writes ID field on defective track to show assignment; passes return code to Mainline phase.
$ALMSG	Print phase	ALMSG	Prints results of successful assignment and track addresses; prints results of unconditional assignment, prints results of unconditional cancel of prior assignment.
$ALSEP	Sector Print phase	ALSEP	Prints address of sector in error and data on that sector.

Alternate Track Rebuild Program

Phase Name	*Descriptive Name*	*Entry Point*	*Synopsis*
$BUILD	Mainline phase	RGOTO	Initializes constants; builds IOB; reads control statements and checks for sequence; determines if statement is control or data.

File and Volume Label Display Program

Phase Name	*Descriptive Name*	*Entry Point*	*Synopsis*
$LABEL	Root and Syntax Scan phase	FILSTN	Reads control statements; checks statement validity and sequence; prints statements if logging device is on; builds and passes table of constants; loads and passes control to Mainline phase.
$LABL1	Mainline phase	DISPLY	Reads volume label; converts and displays volume label or file information.

File Delete Program

Phase Name	*Descriptive Name*	*Entry Point*	*Synopsis*
$DELET	Root phase	FLBRS	Provides save area for keyword parameters; loads and passes control to the Control Statement Analysis and Mainline phases.
$DELT1	Control Statement Analysis phase	FILSYN	Reads control statements and checks their validity; logs statements if device is on; builds keyword parameter table.
$DELT2	Mainline phase	FDEL	If control statement reads LABEL-VTOC, all format 1s with part retain type are read and given an S retain type and format 1s are written in scheduler work area; if statement does not read LABEL-VTOC, specified format 1s are read and given S retain type and format 1s are written in scheduler work area.

Disk Copy/Dump Program

Phase Name	*Descriptive Name*	*Entry Point*	*Synopsis*
$COPY	Root phase	CROOT	Calls phases into storage; maintains table of constants.
$COSYN	Control Statement Analysis phase	CSBEG	Reads control statements and checks for errors and sequence; builds table of constants; passes constants to Root phase.
$COPIN	Main Data Transfer phase (copy pack)	COPINT	Determines I/O area size and pack size; changes last length of IPL label to equal R or F; copies tracks 0 and 1; determines if output volume has active files. Copies data from cylinder 4 to upper limit; reads disk pack by half track, full track, or full cylinder.
$COPIM	Intermediate phase (copy pack)	COPIM	Loads $COPIN to copy cylinder 0; copies remainder of pack; gets space on the fixed disk on drive on (F1) to use as temporary file; halts and issues messages.
$COFIN	Initialization phase (copy pack)	COFI	Determines type of file requested and storage size of program; gets record length and passes it to Root phase.
$CONDP	Simple Copy Files phase (copy files)	CONDP	Copies an indexed sequential or consecutive file that does not need printing, deleting or reorganizing.
$COREC	Copy Non-Indexed Files phase (copy files)	COREC	Copies a consecutive or direct file; deletes a specified record; prints a specified record.

Phase Name	*Descriptive Name*	*Entry Point*	*Synopsis*
$COCIO	Non-Reorganization Indexed Sequential Files phase (copy files)	COINDX	Copies an indexed sequential file; prevents reorganization; allows deleting and printing records; loads Print phase.
$COREI	Reorganization Indexed Sequential Files phase (copy files)	COINDX	Copies an indexed sequential file; allows reorganizing, deleting, and printing records.
$COFIM	Intermediate Control phase (copy files)	CFBEG	Receives and opens DTFs with correct attributes; determines if enough storage is allowed; preallocates space on F1 and opens F1 DTF as consecutive output.
$COIRY	Intermediate Reorganization phase (copy files)	CFBEG	Copies and reorganizes index sequential files from a pack on the removable disk on drive one (R1) to R1; allows deleting from output file; allows printing with output on disk.
$COIRN	Intermediate Non-Reorganization phase (copy file)	CFBEG	Copies an indexed sequential file R1 to R1; allows records to be deleted from output file.
$CONOI	Intermediate Consecutive phase (copy files)	CFBEG	Copies a non-indexed file R1 to R1; allows deleting from the output file and printing with output on disk.
$COPRL (Disk System) $COPRM (Model 6)	Print phase (copy files)	COPRT	Prints record with its key number; indicates if record is deleted; prints a scale on the page; prints record count.
$COMVD	Copy Multi-Volume Direct Files phase	COREC	Copies a multi-volume direct file and allows printing.
$COIMD	Intermediate Multi-Volume Direct phase	CFBEG	Copies multi-volume direct files and allows printing.
$COEOJ	End of Job phase	COEOJ	Puts the correct volume sequence number and file type in the output if copying only one volume of a multi-volume file.

Phase Name	*Descriptive Name*	*Entry Point*	*Synopsis*

Library Maintenance Program

Phase Name	*Descriptive Name*	*Entry Point*	*Synopsis*
$MAINT	Mainline Librarian phase	$MAINT	Reads control statements; builds control areas; passes control to the appropriate control statement processor.
$MAALS	Allocate Syntax Scan phase	$MAALS	Checks the parameters on the // ALLOCATE control statement and determines if the specified functions can be performed.
$MAAMN	Allocate Mainline phase	$MAAMN	Passes control to routines that allocate libraries; performs deallocation functions.
$MAAOL	Allocate Object Library phase	$MAAOL	Builds the object section of the system directory; allocates a scheduler work area.
$MAASL	Allocate Source Library phase	$MAASL	Builds the source section of the system directory.
$MATAD	Track Allocate/ Deallocate phase	$MATAD	Updates the volume label.
$MAAMW	Reallocate Mainline phase	$MAAMW	Reallocates the source/object libraries by calling the appropriate routines.
$MAFND	Find phase	$MAFND	Locates a specified routine in either the source or object library.
$MAACO	Copy Object Library phase	$MAACO	Copies the object library to the specified work pack.
$MAACS	Copy Source Library phase	$MAACS	Copies the source library to the specified work pack.
$MARCO	Copy Work Pack to Object Library phase	$MARCO	Copies the object library from the work pack to the reallocated object library.
$MARCS	Copy Work Pack to Source Library phase	$MARCS	Copies the source library from the work pack to the reallocated source library.
$MAREM	Rename Control Statement Processor phase	$MAREM	Replaces specified name with new name; deletes specified name.
$MADEM	Delete Mainline phase	$MADEM	Deletes the specified source or object library entries; updates the system directory.

Phase Name	*Descriptive Name*	*Entry Point*	*Synopsis*
$MACRS	Copy Reader to Source Library phase	$MACRS	Determines if the requested name is unique to the source library; determines if there is enough space for the program to be copied; copies the program from the reader.
$MADRV	Copy Object Library – Disk to Disk phase	$MADRV	Copies specified library on the specified disk.
$MACSY	Copy System Mainline phase	$MACSY	Copies the following from one system disk pack to another: (1) Initial Program Load (IPL) Bootstrap program, (2) system configuration record, and (3) Nucleus Initialization program (NIP).
$MAOLH	Object Library Handler phase	$MAOLH	Loads an entry into the object library from either the reader or from another object library; checks for available space; dynamically allocates space when needed.
$MALOG	Update PTF Log Sector phase	$MALOG	Updates PTF log sector after an IBM program is copied from disk to disk to reflect current PTFs against that program.
$MACRR	Copy Reader to Object Library (R. Entries) phase	$MACRR	Builds an R. entry in the object library.
$MAEXE	Readjust Extended End of Object Library phase	$MAEXE	Determines the number of tracks that are needed in addition to the allocated number of tracks in order to contain the object library; updates the volume label, format 5 directory, and system directory.
$MACRO	Copy Reader to Object Library (O. Entries) phase	$MACRO	Builds an O. entry in the object library.
$MACSL	Copy Source Library – Disk to Disk phase	$MACSL	Copies the specified source library from one disk to another.
$MAPPM	Print/Punch Source Library phase	$MAPPM	Calls other phases to print and/or punch libraries and/or directories.
$MASRT	Directory Sort phase	$MASRT	Sorts the specified directory before it is printed.
$MASDP	Print Source Directory phase	$MASDP	Prints the contents of the source library.
$MAODP	Print Object Directory phase	$MAODP	Prints the contents of the object library.

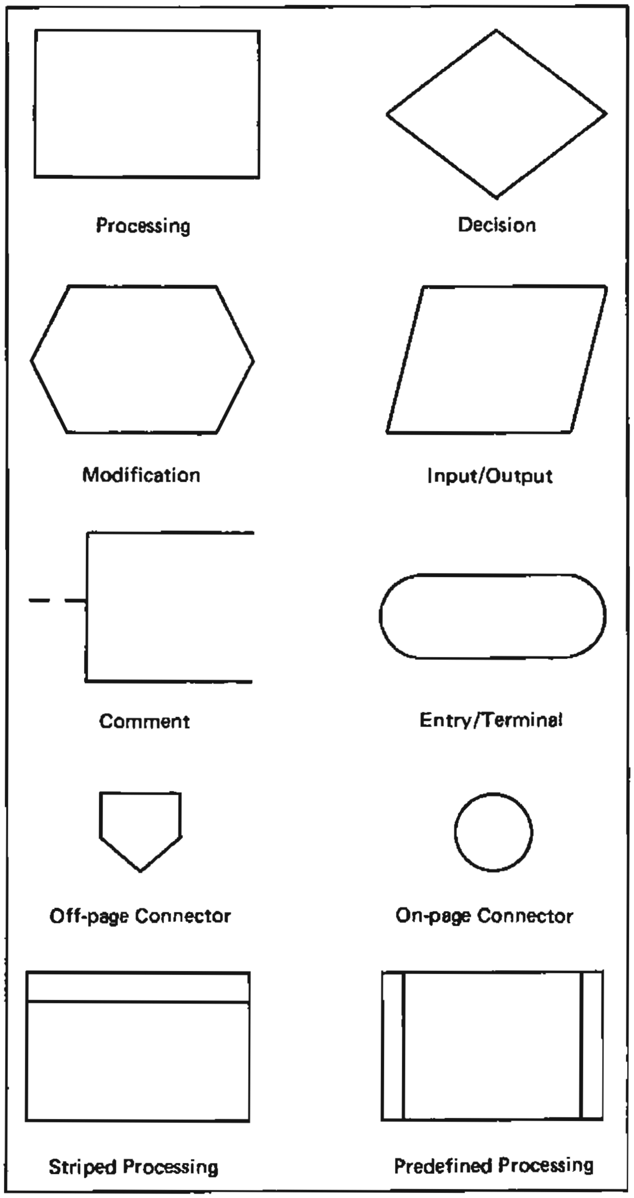
Phase Name	*Descriptive Name*	*Entry Point*	*Synopsis*
$MASYP	Print System Directory phase	$MASYP	Prints the contents of the system directory.
$MASPT	Print/Punch Source Library phase	$MASPT	Prints and/or punches modules from the source library.
$MAOPT	Print/Punch Object Library phase	$MAOPT	Prints and/or punches modules from the object library.

Appendix B. Flowcharting Techniques

Flowcharts in this PLM are identified in this manner:

- A flowchart that consists of only one page is identified with a chart ID of, for example, AA.
- A flowchart that consists of multiple pages with a chart ID of AA is identified as: first page = AA-01, second page = AA-02, and so on.
- A sequence of related flowcharts are identified as: first flowchart = AA, second flowchart = AB, third flowchart = AC, and so on.
- A sequence of flowcharts, each flowchart having multiple pages, is identified as: first flowchart with multiple pages = AA-01, AA-02, and so on; second flowchart = AB-01, AB-02, and so on; continue through the sequence of flowcharts.

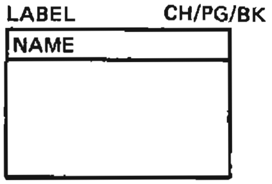
The flowcharting symbols used are:



Most of the symbols are self-explanatory, but some need more explanation.

1. The striped processing block indicates the entry of a routine flowcharted in this PLM.

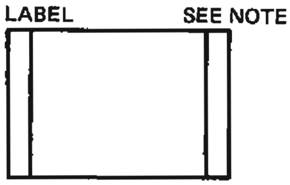
Example:



CH/PG/BK indicates the chart ID, page, and block identification of the flowcharted routine.

2. The predefined processing block indicates a routine flowcharted and/or described in another PLM.

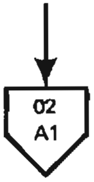
Example:



Note: See "PLM title and form number"

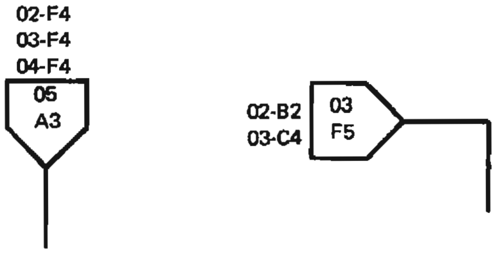
3. Off-page connectors are used to reference between different pages of the same chart ID. Off-page connectors leaving a page contain the page number and block number of their destination.

Example:



Off-page connectors entering a page contain the page number and block number of their origin. If the entry point referenced by the off-page connector is referenced from more than one origin, all origins are given. The origins are listed in alphameric order with the last reference contained within the block.

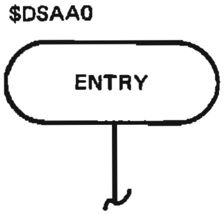
Example:



On-page connectors contain the location of a block on the same page. On-page connectors always contain the location of the destination block.

4. The label in the upper lefthand corner just above the entry symbol is the entry point in the listing for that part of the program.

Example:



Alternate Track Assignment

$$RDS1 (*see* Keyword Syntax Scan routine)
$ALT (*see* Root phase)
$ALT01 (*see* Control Statement Analysis phase)
$ALT02 (*see* Mainline phase)
$ALT03 (*see* Data Transfer phase)
$ALMSG (*see* Print phase)
$ALSUR (*see* Surface Analysis phase)

AATCDE track activity code
  contents of 2-28
  use of 2-11
ABER table
  description of 2-26
  use of 2-10, 2-11
ABFAB data buffer address area 2-28
AIOB control block
  contents of 2-25
  use of 2-5
ALTIN table
  contents of 2-27
  use of 2-5
AMSGT table
  contents of 2-28
  use of 2-5
AMTAB table 2-26
ARCDE return code
  contents of 2-29
  use of 2-11
ARDTC return code
  contents of 2-29
  use of 2-6, 2-11
ASCTAB table 2-26
ASCTBK table
  description of 2-26
  use of 2-5, 2-10
ASIOAR buffer
  contents of 2-28
  use of 2-5
ASLTA table 2-27
ASOTLY table 2-27
ASSC table
  contents of 2-26
  use of 2-5
ASSCOT buffer
  contents of 2-28
  use of 2-5
assignment, definition
  conditional 2-3
  unconditional 2-3
  unconditional cancel of prior assignment 2-3
ASSMG table 2-27
ASTAB table
  contents of 2-28
  use of 2-5
ATBLE table
  contents of 2-27
  use of 2-5
ATPDL table
  contents of 2-28
  use of 2-5

buffers, format of 2-28

cancel of prior assignment, unconditional
  description of 2-3
  operation of 2-6
conditional assignment
  description of 2-3
  operation of 2-5
control blocks, format of 2-25
Control Statement Analysis phase ($ALT01)
  flowchart 2-14
  functions 2-10
  use of 2-5

data area formats 2-25
Data Transfer phase ($ALT03)
  flowchart 2-19
  functions 2-12
  use of 2-5, 2-6
defective track, possible actions to take 2-6
directory A-1

EOJ transient 2-6
error recovery procedure
  during data transfer 2-6
  during surface analysis 2-12

FCS work area 2-29
flowcharting techniques B-1

Halt/Syslog transient 2-6

ID fields
  setting 2-5
  use of 2-3
  writing 2-5, 2-12
introduction 2-3
IOADDR buffer
  contents of 2-28
  use of 2-5, 2-13

job scheduler work area, use of 2-10

Keyword Syntax Scan routine ($$RDS1), use of 2-5

Mainline phase ($ALT02)
  flowchart 2-16
  functions 2-11
  use of 2-5, 2-6

operation diagram 2-7

Print phase ($ALMSG)
  flowchart 2-23
  function 2-13
  use of 2-5, 2-11

- PWK1 work area 2-29

- RIDFCS work area 2-29
- Root phase ($ALT)
  - functions 2-10
  - use of 2-5, 2-6

- Sector Print phase ($ALSEP)
  - flowchart 2-24
  - functions 2-13
  - use of 2-5, 2-12
- SIOB control block
  - contents of 2-25
  - use of 2-5
- storage map, alternate track assignment 2-9
- Surface Analysis phase ($ALSUR)
  - flowchart 2-21
  - functions 2-12
  - use of 2-5, 2-6
- SYSIN, use of 2-10
- system requirements 2-3

- tables, format of 2-26

- unconditional assignment
  - introduction 2-3
  - operation of 2-5
- unconditional cancel of prior assignment
  - introduction 2-3
  - operation of 2-6

- volume label 2-3, 2-11

- work areas, format of 2-29

Alternate Track Rebuild

- $$RDS1 (*see* Keyword Syntax Scan routine)
- $BUILD (*see* Mainline phase)
- // PATCH control statement 3-3
- // REBUILD control statement 3-3

- buffers, format of 3-8

- control blocks, format of 3-7
- control statements 3-5

- data area formats 3-7
- data statement 3-5
- directory A-2

- EOJ transient, use of 3-5

- flowcharting techniques B-1

- Halt/Syslog transient, use of 3-5

- introduction 3-3

- Keyword Syntax Scan routine ($$RDS1), use of 3-5

- Mainline phase ($BUILD)
  - flowchart 3-6
  - functions 3-5

- RA conversion table 3-8
- RADDS address table
  - contents of 3-8
  - use of 3-5
- RBUFFO disk input/output area 3-8
- RCARD input buffer 3-8
- RIOB control block
  - contents of 3-7
  - use of 3-5
- RSYOB output buffer 3-8
- RTAB parameter table
  - contents of 3-8
  - use of 3-5

- storage map 3-5
- system requirements 3-3

- tables, format of 3-8

Disk Copy/Dump

- $$RDS1 (*see* Keyword Syntax Scan routine)
- $COCIO (*see* Non-Reorganization Indexed Sequential Files phase)
- $COEOJ (*see* End-of-Job phase)
- $COFIM (*see* Intermediate Control phase)
- $COFIN (*see* Initialization phase)
- $COIMD (*see* Intermediate Multi-Volume Direct phase)
- $COIRN (*see* Intermediate Non-Reorganization phase)
- $COIRY (*see* Intermediate Reorganization phase)
- $COMVD (*see* Copy Multi-Volume Direct Files phase)
- $CONDP (*see* Simple Copy Files phase)
- $COPIM (*see* Intermediate phase)
- $COPIN (*see* Main Data Transfer phase)
- $COPRL (*see* Print phase)
- $COPRM (*see* Print phase)
- $COPY (*see* Root phase)
- $COREC (*see* Copy Non-Indexed Files phase)
- $COREI (*see* Reorganization Indexed Sequential Files phase)
- $COSYN (*see* Control Statement Analysis phase)

- CDCTL table 6-50
- CDFIOB control block 6-50
- CDIBUF I/O buffer 6-51
- CDIIOB control block 6-50
- CDOIOB control block 6-50
- CDPIBF I/O buffer 6-51
- CDPIOB control block 6-50
- CDPMBF I/O buffer 6-51
- CERRTB table 6-51
- CFERR table 6-52
- CFIN control block 6-51
- CFIOB control block 6-51
- CFOUT control block 6-51
- CFVLBF buffer 6-54
- CF1DTF control block 6-53
- CHTTB table 6-53
- CNOTAB table 6-54
- COIEND buffer 6-54
- common data areas 6-47
- common functions
  - general diagram 6-6
  - operation of 6-5
  - phase descriptions 6-11
- CONEND record area 6-54

Control Statement Analysis phase ($COSYN)
- flowchart 6-22
- functions 6-11
- use of 6-5

COPHLT table 6-50
copy file data areas 6-51
copy files function
- operation diagram 6-9
- operation of 6-8
- phase descriptions 6-14
- storage maps 6-15

Copy Multi-Volume Direct Files phase ($COMVD)
- flowchart 6-43
- functions 6-20
- operation of 6-8

Copy Non-Indexed Files phase ($COREC)
- flowchart 6-32
- functions 6-16
- operation of 6-8

copy pack data areas 6-50
copy pack function
- operation diagram 6-7
- operation of 6-5
- phase description 6-12
- storage maps 6-12

CORHLT table 6-52
CPBUF buffer 6-54
CPLIST table 6-54
CPWRK conversion area 6-54
CRIHLT table 6-53
CSCARD input area
- description of 6-49
- use of 6-5

CSCTL table
- contents of 6-47
- use of 6-5

CSCTLP table
- contents of 6-48
- use of 6-5

CSERRB table
- contents of 6-48
- use of 6-5

CSFILT table
- contents of 6-49
- use of 6-5

CSPKTL table
- contents of 6-49
- use of 6-5

CSSYNT table
- contents of 6-48
- use of 6-5

CSSYOUT output buffer 6-49
CSYSIN table 6-49

data area formats 6-47
- common 6-47
- copy file 6-51
- copy pack 6-50

directory A-3

End-of-Job phase ($COEOJ)
- flowchart 6-45
- functions 6-21

flowcharting techniques B-1

Initialization phase ($COFIN)
- flowchart 6-30
- functions 6-14

Intermediate Consecutive phase ($CONOI)
- flowchart 6-38
- functions 6-19
- operation of 6-8

Intermediate Control phase ($COFIM)
- flowchart 6-36
- functions 6-18
- operation of 6-8

Intermediate Multi-Volume Direct phase ($COIMD)
- flowchart 6-38
- function 6-38
- operation of 6-8

Intermediate Non-Reorganization phase ($COIRN)
- flowchart 6-38
- functions 6-19
- operation of 6-8

Intermediate phase ($COPIM)
- flowchart 6-28
- functions 6-12
- operation of 6-5

Intermediate Reorganization phase ($COIRY)
- flowchart 6-38
- functions 6-18
- operation of 6-5

introduction 6-3

Keyword Syntax Scan routine ($$RDS1), use of 6-5

Main Data Transfer phase ($COPIN)
- flowchart 6-26
- function 6-12
- operation of 6-5

method of operation 6-5

Non-Reorganization Indexed Sequential Files phase ($COCIO)
- flowchart 6-34
- functions 6-17
- operation of 6-8

operation diagram
- COPY FILE 6-9
- COPY PACK 6-7

Print phase ($COPRL–Disk System; $COPRM–Model 6)
- flowchart 6-41
- functions 6-20

Reorganization Indexed Sequential Files phase ($COREI)
- flowchart 6-34
- functions 6-17
- operation of 6-8

Root phase ($COPY)
- functions 6-11
- operation of 6-5

Simple Copy Files phase ($CONDP)
- flowchart 6-31
- functions 6-14
- operation of 6-8

storage maps
- copy file function 6-15
- copy pack function 6-12

system requirements 6-3

Disk Initialization

$$RDS1 (*see* Keyword Syntax Scan routine)
$INIT (*see* Root phase)
$INIT1 (*see* Control Statement Analysis phase)
$INIT2 (*see* Mainline phase)
$INMSG (*see* Print phase)
$INSUR (*see* Surface Analysis phase)
// END control statement 1-9
// UIN control statement 1-3
    checked 1-8
// VOL control statement 1-9

ALTIN table
    contents of 1-25
    initialization of 1-9
    updating of 1-10
    use of 1-5

buffers, format of 1-25

capacity
    cylinder 1-3
    disk 1-3
    pack 1-9
clear 1-3
control blocks, format of 1-21
Control Statement Analysis phase ($INIT1)
    flowchart 1-12
    functions 1-8
    use of 1-5
cylinder capacity 1-3

data area formats 1-21
directory A-1
disk capacity 1-3

EOJ transient, use of 1-5, 1-10
error recovery procedure 1-5

FCS work area, description of 1-26
flowcharting techniques B-1

halts, PID keyword 1-8
Halt/Syslog transient 1-5

IBFAD write buffer address area
    description of 1-25
    use of 1-5
ICSTL table
    contents of 1-22
    use of 1-5
ID field format 1-3
IERTAB table 1-24
IMSGT table
    description of 1-24
    use of 1-5
introduction 1-3
IOADDR write buffer
    description of 1-25
    use of 1-5
IOB control block
    contents of 1-21
    use of 1-5
IOBV control block
    contents of 1-21
    use of 1-5
IOSV save area
    contents of 1-26
    use of 1-5
IPL sectors 1-10
IRCDE return code
    contents of 1-26
    use of 1-5
ISCTBK table
    contents of 1-22
    use of 1-5
ISCTLO table
    contents of 1-24
    use of 1-5
ISIO read buffer
    description of 1-25
    use of 1-8
ISMGT table 1-10
ISSCAN table
    contents of 1-23
    use of 1-5
ISSCOT output buffer
    description of 1-25
    use of 1-5
ISSYSM table
    contents of 1-24
    use of 1-5
ISUNTB table
    contents of 1-23
    use of 1-5
ISVBG table
    contents of 1-22
    use of 1-5
ISVLTB table
    contents of 1-23
    use of 1-5

Keyword Syntax Scan routine ($$RDS1), use of 1-5

Mainline phase (INIT2)
    flowchart 1-14
    functions 1-9
    use of 1-5
method of operation 1-5

OBR/SBR sectors 1-3, 1-9
operation diagram 1-6

pack capacity 1-9
PID keyword, halts 1-8
primary initialization 1-3
Print phase ($INMSG)
    flowchart 1-20
    functions 1-11
    use of 1-5
program portection sectors 1-3, 1-9

RIDFCS work area 1-26
Root phase ($INIT)
    functions 1-8
    use of 1-5

- secondary initialization 1-3
- SIOB control block 1-21
- storage map, disk initialization program 1-7
- Surface Analysis phase ($INSUR)
  - flowchart 1-18
  - functions 1-10
  - use of 1-5
- system requirements 1-3

- tables, format of 1-22
- TBLE table
  - contents of 1-25
  - use of 1-5

- VOLS write buffer
  - contents of 1-25
  - use of 1-5
- volume label
  - use of 1-9
  - writing of 1-5
- volume table of contents (VTOC) 1-3, 1-10

- work areas, format of 1-26

File Delete

- $$RDS1 (*see* Keyword Syntax Scan routine)
- $DELET (*see* Root phase)
- $DELT1 (*see* Control Statement Analysis phase)
- $DELT2 (*see* Mainline phase)

- buffers, format of 5-18

- Control Statement Analysis phase ($DELT1)
  - flowchart 5-10
  - functions 5-8
  - use of 5-5

- data area formats 5-15
- DATA-YES parameter 5-5, 5-9
- DATE option 5-9
- directory A-3

- END statement 5-5
- EOJ transient, use of 5-5

- FFIO I/O buffer 5-18
- flowcharting techniques B-1
- FMHSP table 5-17
- FMOTL table 5-16
- format 1's, reading 5-9
- FPRNT table 5-16
- FRCTBK (FSCTBK) table
  - contents of 5-15
  - use of 5-5, 5-8
- FRMNTB table 5-17
- FSBUF input buffer
  - description of 5-18
  - use of 5-5
- FSCTBK table, use of 5-5
  - *also see* FRCTBK table
- FSCTLO table 5-18
- FSFLG table 5-16
- FSHTB table 5-17
- FSJWTB table 5-16
- FSSCAN table 5-15
- FSSCOT output buffer
  - description of 5-18
  - use of 5-5
- FSSCTB table 5-17
- FSSTB table 5-15
- FSSYSM table 5-18
- FSWRK work area 5-18
- FVIO I/O buffer 5-18
- FVTAB table 5-16
- F1's, reading (*see* format 1's, reading)

- Halt/Syslog transient, use of 5-5

- introduction 5-3

- Job Scheduler Work Area Put routine, use of 5-5
- Job Scheduler Work Area Read routine, use of 5-16

- Keyword Syntax Scan routine ($$RDS1), use of 5-5

- LABEL-VTOC statement 5-5

- Mainline phase ($DELT2)
  - flowchart 5-12
  - functions 5-9
  - use of 5-5
- method of operation 5-5

- operation diagram 5-6

- P retain type (*see* retain type, P)
- PARMA table 5-16
- PARMB table 5-17

- REMOVE statement 5-3, 5-5
- retain type
  - P 5-5
  - S 5-5
  - T 5-5
- Root phase ($DELET)
  - functions 5-8
  - use of 5-5

- S retain type (*see* retain type, S)
- SCRATCH statement 5-3, 5-5
- storage map 5-7
- system requirements 5-3

- T retain type (*see* retain type, T)
- tables, format of 5-15

- VTOC Read/Write routine 5-3, 5-5

File and Volume Label Display

- $$RDS1 (*see* Keyword Syntax Scan routine)
- $LABEL (*see* Root and Syntax Scan phase)
- $LABL1 (*see* Mainline phase)

- buffers, format of 4-16

data area formats 4-13
directory A-2
DISPLAY statement 4-5

EOJ transient, use of 4-5

FDBUF print buffer 4-5
FDHSP table 4-15
FDLIST table 4-15
FDOTL table 4-15
FIOA read buffer
  description of 4-16
  use of 4-5
FIOA2 read buffer
  description of 4-16
  use of 4-5
flowcharting techniques B-1
FRCTBK table
  building of 4-8
  description of 4-13
  use of 4-5, 4-8
FRMNTB table 4-14
FSBUF input buffer
  description of 4-16
  use of 4-5
FSCTLO table 4-14
FSDTB table 4-13
FSFLG table 4-13
FSJWTB table 4-14
FSSCAN table
  contents of 4-13
  use of 4-5
FSSCOT scan output area
  description of 4-16
  use of 4-5
FSSCTB table 4-15
FSSYSM table 4-14
FVPARM table 4-15

Halt/Syslog transient, use of 4-5

ID, pack 4-5
introduction 4-3

Keyword Syntax Scan routine ($$RDS1), use of 4-5

LABEL-VTOC, control statement 4-8

Mainline phase ($LABL1)
  flowchart 4-10
  functions 4-8
method of operation 4-5

operation diagram 4-6

pack ID (*see* ID, pack)

Root and Syntax Scan phase ($LABEL)
  flowchart 4-9
  functions 4-8
  use of 4-5

storage map 4-7
system requirements 4-3

tables, format of 4-13

volume table of contents (VTOC), use of 4-5

Library Maintenance

$$OXRF (*see* Transient Resolver phase)
$$RDS1 (*see* Keyword Syntax Scan routine)
$$STAI (*see* Allocation Initiator phase)
$$SYSP (*see* Source Put phase)
$$TMEJ (*see* End of Job Deallocation phase)
$MAACO (*see* Copy Object Library phase)
$MAACS (*see* Copy Source Library phase)
$MAALS (*see* Allocate Syntax Scan phase)
$MAAMN (*see* Allocate Mainline phase)
$MAAMW (*see* Reallocate Mainline phase)
$MAAOL (*see* Allocate Object Library phase)
$MAASL (*see* Allocate Source Library phase)
$MACRO (*see* Copy Reader to Object Library (O. Entries) phase)
$MACRR (*see* Copy Reader to Object Library (R. Entries) phase)
$MACRS (*see* Copy Reader to Source Library phase)
$MACSL (*see* Copy Source Library–Disk to Disk phase)
$MACSY (*see* Copy System Mainline phase)
$MADEM (*see* Delete Mainline phase)
$MADRV (*see* Copy Object Library–Disk to Disk phase)
$MAEXE (*see* Readjust Extended End of Object Library phase)
$MAFND (*see* Find phase)
$MAINT (*see* Mainline Librarian phase)
$MALOG (*see* Update PTF Log Sector phase)
$MAODP (*see* Print Object Directory phase)
$MAOLH (*see* Object Library Handler phase)
$MAOPT (*see* Print/Punch Object Library phase)
$MAPPM (*see* Print/Punch Mainline phase)
$MARCO (*see* Copy Work Pack to Object Library phase)
$MARCS (*see* Copy Work Pack to Source Library phase)
$MAREM (*see* Rename Control Statement Processor phase)
$MASDP (*see* Print Source Directory phase)
$MASPT (*see* Print/Punch Source Library phase)
$MASRT (*see* Directory Sort phase)
$MASYP (*see* Print System Directory phase)
$MATAD (*see* Track Allocate/Deallocate phase)
// ALLOCATE control statement 7-5
// COPY control statement 7-15
// DELETE control statement 7-15
// END control statement 7-16
// RENAME control statement 7-5

// COPY control statement 7-8
// DELETE control statement 7-8
// END control statement 7-9
// RENAME control statement 7-5

Allocate Mainline phase ($MAAMN)
  flowchart 7-44
  functions 7-14
  operation of 7-5
  storage map 7-14
Allocate Object Library phase ($MAAOL)
  flowchart 7-45
  functions 7-14
  operation of 7-5

Allocate Source Library phase ($MAASL)
  flowchart 7-46
  functions 7-15
  operation of 7-5
Allocate Syntax Scan phase ($MAALS)
  flowchart 7-38
  functions 7-13
  operation 7-5
  storage map 7-13
allocation of libraries 7-3, 7-5
Allocation Initiator phase ($$STAI) 7-5

control area (CAREA)
  building of 7-11
  contents of 7-105
  use of 7-5
control statements
  description of 7-3
  operation of 7-5
copy all libraries 7-8
copy O. entries or R. entries to object libraries 7-8
Copy Object Library–Disk to Disk phase ($MADRV)
  flowchart 7-64
  functions 7-22
  operation of 7-8
  storage map 7-22
Copy Object Library phase ($MAACO)
  flowchart 7-52
  functions 7-18
  operation of 7-5
copy reader to object library (O. entries) 7-8
copy reader to object library (R. entries) 7-8
Copy Reader to Object Library (O. Entries) phase ($MACRO)
  flowchart 7-82
  functions 7-26
  operation of 7-9
  storage map 7-28
Copy Reader to Object Library (R. Entries) phase ($MACRR)
  flowchart 7-78
  functions 7-25
  operation of 7-8
  storage map 7-25
copy reader to source library 7-8
Copy Reader to Source Librarv phase ($MACRS)
  flowchart 7-63
  functions 7-21
  operation of 7-8
  storage map 7-21
Copy Source Library–Disk to Disk phase ($MACSL)
  flowchart 7-85
  functions 7-28
  operation of 7-9
  storage map 7-28
Copy Source Library phase ($MAACS)
  flowchart 7-54
  functions 7-18
  operation of 7-5
copy source library to source library 7-8
copy system 7-8
Copy System Mainline phase ($MACSY)
  flowchart 7-71
  functions 7-24
  operation of 7-8
Copy Work Pack to Object Library phase ($MARCO)
  flowchart 7-57
  functions 7-19
  operation of 7-5
Copy Work Pack to Source Library phase ($MARCS)
  flowchart 7-58
  functions 7-19
  operation of 7-5
copying of libraries 7-3, 7-9

data area formats 7-105
Delete Mainline phase ($MADEM)
  flowchart 7-61
  functions 7-20
  operation of 7-15
  storage map 7-20
deletion of libraries 7-3, 7-8
directory, format of A-5
Directory Sort phase ($MASRT)
  flowchart 7-92
  functions 7-30
  operation of 7-9

end of job 7-9
End of Job Deallocation phase ($$TMEJ) 7-9
END statement 7-27
EOJ transient 7-9

Find phase ($MAFND)
  flowchart 7-50
  functions 7-16
  operation of 7-5
flowcharting techniques B-1

hardware requirements 7-3
header statement 7-27

introduction 7-3
IPL Bootstrap program, copying 7-8

Keyword Syntax Scan routine ($$RDS1) 7-5

LEBUFR input buffer 7-5
LELUXP keyword parameter list 7-5
LERETO data area 7-5

Mainline Librarian phase ($MAINT)
  flowchart 7-33
  functions 7-11
  operation of 7-5
  storage map 7-12
method of operation 7-5

NIP, copying of 7-8

object library directory 7-110
Object Library Handler phase ($MAOLH)
  flowchart 7-73
  functions 7-24
  operation of 7-8
operation diagram 7-6
operation of
  ALLOCATE control statement 7-5
  COPY control statement 7-8
  DELETE control statement 7-8
  END control statement 7-9
  RENAME control statement 7-5

Print Object Directory phase ($MAODP)
  flowchart 7-96
  functions 7-31
  operation of 7-9
print/punch directory or library 7-8
Print/Punch Mainline phase ($MAPPM)
  flowchart 7-89
  functions 7-29
  storage map 7-29
Print/Punch Object Library phase ($MAOPT)
  flowchart 7-100
  functions 7-32
  operation of 7-9
Print/Punch Source Library phase ($MASPT)
  flowchart 7-98
  functions 7-32
  operation of 7-9
Print Source Directory phase ($MASDP)
  flowchart 7-95
  functions 7-30
  operation of 7-9
Print System Directory phase ($MAASYP)
  flowchart 7-97
  functions 7-31
  operation of 7-16

Readjust Extended End of Object Library phase ($MAEXE)
  flowchart 7-81
  functions 7-26
  operation of 7-8

Reallocate Mainline phase ($MAAMW)
  flowchart 7-49
  functions 7-16
  operation of 7-5
  storage map 7-17
relocation directory statement 7-27
Rename Control Statement Processor phase ($MAREM)
  flowchart 7-59
  functions 7-20
  operation of 7-5
  storage map 7-20
renaming of libraries 7-3, 7-5

source library directory 7-110
Source Put phase ($$SYSP) 7-8
system configuration record, copying 7-8
system requirements 7-3

text statement 7-27
Track Allocate/Deallocate phase ($MATAD)
  flowchart 7-47
  functions 7-15
  operation of 7-5
Transient Resolver phase ($$OXRF) 7-9

Update PTF Log Sector phase ($MALOG)
  flowchart 7-77
  functions 7-25
  operation of 7-8

IBM System/3 Printed in U.S.A. SY21-0503-1

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)