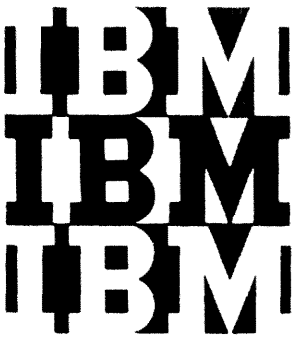


File No. S360-32
Form Y28-2039-3

Program Logic

IBM System/360 Time Sharing System

Independent Utilities

Program Logic Manual

This publication describes the internal logic of the IBM System/360 Time Sharing System (TSS/360) Independent Utility Programs.

Program logic manuals are intended for use by IBM customer engineers involved in program maintenance and by system programmers involved in altering program design. Program logic information is not necessary for program operation and use.

PREFACE

This publication describes the internal logic of the TSS/360 Independent Utility Programs:

Direct-Access Storage Device Initialization (DASDI) program,

Dump/Restore (DASDDR) program,

Direct-Access Print (DADUMP) program,

System/360 Model 67 Core Dump program,

System/360 VAM 2 Utility Program.

Each program is a self-contained module, having its own I/O supervisory routines. Accordingly, each section of this publication is devoted to one program.

The manual is intended for IBM service and maintenance personnel, and requires familiarity with overall TSS/360 techniques and terminology, as contained in:

IBM System/360 Time Sharing System: Concepts and Facilities, Form C28-2003

IBM System/360 Time Sharing System: Independent Utilities, Form C28-2038

IBM System/360 Time Sharing System: System Messages, Form C28-2037

IBM System/360 Time Sharing System: System Control Blocks Program Logic Manual, Form Y28-2011

IBM System/360 Time Sharing System: System Generation and Maintenance Program Logic Manual, Form Y28-2015

Fourth Edition (January 1970)

This is a major revision of, and makes obsolete, Form Y28-2039-2. This edition reflects changes released for Version 6, Modification 0, of IBM System/360 Time Sharing System, including the VAM 2 Utility Program, and changes to Dump/Restore.

This edition is current with Version 6, Modification 0, and remains in effect for all subsequent versions or modifications of IBM System/360 Time Sharing System unless otherwise indicated. Significant changes will be provided in new editions or Technical Newsletters. Before using this publication in connection with the operation of IBM systems, refer to the latest edition of IBM System/360 Time Sharing System: Addendum, Form C28-2043, for the editions of publications that are applicable and current.

Changes to the text are indicated as follows:

Symbol	Meaning
• (next to a page number or caption of an illustration)	The page (or illustration) has been revised and should be reviewed in its entirety.
| (in the margin at left of text)	Adjacent text contains a change. (This symbol is not used if a bullet appears next to the page number.)

This publication was prepared for production using an IBM computer to update the text and to control the page and line format. Page impressions for photo-offset printing were obtained from an IBM 1403 Printer using a special print chain.

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

A form is provided at the back of this publication for reader's comments. If the form has been removed, comments may be addressed to the IBM Corporation, Time Sharing System/360 Programming Publications, Department 643, Neighborhood Road, Kingston, N.Y. 12401

© Copyright International Business Machines Corporation 1967, 1968, 1969, 1970

CONTENTS

INTRODUCTION 1

SECTION 1: DIRECT-ACCESS STORAGE DEVICE INITIALIZATION (DASDI)
PROGRAM - MODULE CEBDI 2
Operation 3
DASDI I/O Supervisory Routines 3
Initialization Routine 3
START Routine 3
CKINPUT Routine 5
ATTN Routine 5
I/O Routines 5
SYSIN Routine 5
STARTIO Routine 5
CKCSW Routine 5
Control Statement Analysis 5
Volume Label Checking 7
GEN00 Routine 7
Format Routines 7
ADTOLIST Routine 7
CONSTR1 Routine 8
CONSTR2 Routine 9
FMTVTOC Routine 9
WRTVTOC Routine 9
ENTPOIT1 Routine 9
VAM2301 Routine 10
DASDIPAT Routine 10
Termination Routines 10
End-of-Job (EOJ) Routine 10
GETALTYZ Routine 11
TRKPRNT Routine 11
Message Routines 11
Message Output (PRNT) Routine 11
I/O Error Print (IOPRNT) Routine 11
Write-to-Operator (WTO) Routine 11

SECTION 2: DUMP/RESTORE (DASDDR) PROGRAM - MODULE CEBDR 15
Program Operation 15
DUMPED DATA FORMAT (SAM or VAM 1) 15
2311 to 2311 Dump 15
2314 to 2314 Dump 15
2311 or 2314 to Tape 15
DUMPED DATA FORMAT (VAM 2) 16
2311 to 2311, 2311 to 2314, 2314 to 2314 Dump 16
2311 or 2314 to Tape 16
Program Flow 16
Initialization Routines 16
Start Routine 16
CKINPUT Routine 17
ATTN Routine 17
Control Statement Analysis Routines 17
CLRSCAN Routine 17
Processing Routines 20
START1 Routine 20
BEG1 Routine 20
HAVEADDR Routine 20
TOTAPE Routine 20
TOFM2311 Routine 21
FRMTAPE Routine 21
CKVOLLBL Routine 21
BLDMAP Routine 21
BUILDCCW Routine 21
MODTKADF Routine 22

ANALSENS Routine 22
CNGRDCKD Routine 22
PREPRWRT Routine 22
TSTWRTSP Routine 22
DMPDASD Routine 22
DMPDTA Routine 22
PUTMAP Routine 22
RSTRTAPE Routine 22
MODTKADT Routine 22
WRTDISK Routine 22
WRTDISKY Routine 23
CHTOSKP Routine 23
WRTCHECK Routine 23
CHANGCYL Routine 23
CKALT Routine 23
CKALT1 Routine 23
GETCCW Routine 23
BADPAGE Routine 23
RLOCAT 24
ENDREEL Routine 24
EOJ1 Routine 24
EOJA Routine 24
ENDOFJOB Routine 24
EOJAA Routine 24
DADSM Routine 25
RBLDDSCB Routine 25
TAPEEOF Routine 25
VAM2DUMP Routine 25
CNVRELPG Routine 25
CLRERPG Routine 25
VAM2REST Routine 26
RELOCATE Routine 26
V2UNEXCP Routine 26
V2WRTCHK Routine 26
VAM 2 EOF Routines 26
DASDDR I/O Supervisor Routines 26
SYSIN Routine 26
STARTIO Routine 26
CKCSW Routine 27
Message Routines 27
Message Output (PRNT) Routine 27
I/O Error Print (IOPRNT) Routine 27
Write-to-Operator (WTO) Routine 27
Error Messages 27
Diagnostic Messages 27

SECTION 3: DIRECT-ACCESS PRINT (DADUMP) PROGRAM - MODULE CEBDP . . . 33
Program Operation 33
DADUMP Routines 33
Initialization Routines 33
Start Routine 33
Console Routine 33
ATTN Routine 34
TYPEINPT Routine 34
Job Statement Analysis Routines 35
CLRSCAN Routine 35
SCANCARD Routine 35
MSGPRINT Routine 35
Processing Routines 36
VRECOVR Routine 36
Extent Printer Routine (LISTTEST) 37
PRNTDASD Routine 37
FORMAT Routine 37
Heading Routine (HEADER) 37
Job Fulfillment Routines 38
Print Routine (PRINT) 38
DADUMP I/O Supervisory Routines 38
STARTIO Routine 38

CKCSW Routine 38
Message Routines 38
IOPRINT Routine 38
Error Messages 38
Diagnostic Messages 39

SECTION 4: SYSTEM/360 MODEL 67 CORE DUMP PROGRAM - MODULE CEBCP . . . 46
Program Operation 46
Initialization 46
CPU Dump Routine 47
COREDUMP Routine 47

SECTION 5: VAM 2 UTILITY PROGRAM (VAM2UT)--MODULE CEBVP 52
Operation 52
Initialization Routines 53
START Routine 54
ATTN Routine 54
Command Processing Routines 54
FOUNDSCB Routine 54
CKBUF2 Routine 54
COMRTN Routine 54
Processing Routines 54
ASKAGAIN Routine 56
BUILDSCB Routine 56
CKAGAIN Routine 56
CHKDIS Routine 56
DISKONN Routine 56
DOVL Routine 56
DOVLP Routine 56
DSCBDUMP Routine 56
DSCBDUM1 Routine 56
FINDCSD Routine 56
FIXCHK Routine 56
GETDEV Routine 57
GETDSNAM Routine 57
GOWRITE1 Routine 57
MODSAVE1 Routine 57
MODUMP Routine 57
MOVEREL Routine 57
PAGEOFF1 Routine 57
PATCH Routine 57
PATCHER Routine 57
PATDUMP Routine 57
PATDUMP1 Routine 57
PODSDUMP Routine 57
RDPOD Routine 57
READPAGE Routine 57
PMDPATCH Routine 57
READPMD Routine 58
READPREL Routine 58
SCAN Routine 58
SETDIN Routine 58
SETDISK Routine 58
SETDOUT Routine 58
SRCHPMD Routine 58
SRCHPOD Routine 58
VAMCK Routine 58
WRITPREL Routine 58

APPENDIX A: COMMON INTERRUPTION AND I/O ERROR HANDLING 75
I/O Error Routines 75
Error Processing Routines 76

APPENDIX B: UNIT CONTROL BLOCKS (UCB) 77
Unit Control Blocks (UCB) 77
Unit Control Block Format 77
Device Type Table 77
Error Indicator Code Format 77

APPENDIX C: COMMON UTILITY CONTROL BLOCKS 78

INDEX 79

ILLUSTRATIONS

FIGURES

Figure 1. Image of Track Zero 9
Figure 2. Dumping and Restoring a Direct-Access Track (SAM and VAM 1) 16
Figure 3. VAM 2 Dump Tape Format 17
Figure 4. Entering Information Into CNTRLBUF 21
Figure 5. PSA Table Format 47
Figure 6. DSCBREL 56

TABLES

Table 1. DASDI Routines 4
Table 2. Keyword Parameters 6
Table 3. ADTOLIST Identifying Code 8
Table 4. 2314 Page/Track Relationship 10
Table 5. 2311 Page/Track Relationship 10
Table 6. 2301 Page/Track Relationship 11
Table 7. DASDDR Routines (Part 1 of 3) 18
Table 8. DADUMP Routines 34
Table 9. SCANSW Bits 36
Table 10. The CALL Command 54
Table 11. The SET Command 54
Table 12. The DUMP Command 55
Table 13. The DISPLAY Command 55
Table 14. The PATCH Command 55

CHARTS

Chart AA. DASDI Routine (Part 1 of 3) 12
Chart BB. DUMP-RESTORE Routine (Part 1 of 5) 28
Chart CC. DADUMP Routine (Part 1 of 6) 40
Chart DD. CORE DUMP Routine (Part 1 of 3) 49
Chart EE. VAM2UT Routines (Part 1 of 15) 59

INTRODUCTION

The TSS/360 independent utility programs support the IBM System/360 Time Sharing System. The programs, operating independently of TSS/360, perform their own interrupt processing and input/output functions. The programs are:

Direct-Access Storage Device Initialization (DASDI): Sets the initial format for disk and drum storage units for TSS/360 use. Specifically, the program initializes:

1. 2311 disk storage drive in either of three formats: sequential access method (SAM), virtual access method (VAM 1); or VAM 2.
2. 2314 direct-access storage facility in either SAM or VAM 1 or VAM 2 format:
3. 2301 drum storage in modified VAM (4096 byte) format.

In addition, DASDI assigns alternate tracks (GETALT function) to the SAM formatted 2311 or 2314, if necessary, or makes defective pages unavailable for VAM 2 formatted disks. DASDI is used to prepare direct-access devices for TSS/360 utilization.

Dump/Restore (DASDDR): Dumps and restores the contents of a direct-access volume. DASDDR dumps the data contents of a volume from:

2311 and 2314 disk storage to magnetic tape,

2311 disk storage to 2311 disk storage,

2314 disk storage to 2314 disk storage,

2311 disk storage to 2314 disk storage (VAM 2 format only).

Data is restored from:

2400 magnetic tape to 2311 and 2314 disk storage.

Direct-Access Device Print (DADUMP): Prints the contents of direct-access devices 2311 and 2314 disk storage, and 2301 drum. DADUMP is used to obtain a printout of the drum or disk and may be used for checking and/or debugging purposes.

System/360 Model 67 Core Dump: Provides a means of printing all information pertinent to each CPU of the system and the contents of main storage. Addressable core storage is computed dynamically. The program has two functions:

1. CPU dump: lists in special format the register contents and the PSAs for each CPU in the system; and
2. Core dump: lists the contents of main storage and the protection keys.

The core dump program is used primarily for program checkout and maintenance.

VAM 2 Utility Program (VAM2UT): dumps, displays, and patches VAM 2 format disk volumes; it is entered in TSSS format from either the operator console or the 2540 card reader.

The TSS/360 independent utility programs run on any standard model of System/360 having sufficient main storage and the required device configuration. The core dump program operates on a model 67 only. Since the programs are independent of TSS/360 they are controlled and loaded from the console as card decks or card images on tape.

Control statements for the requested program can follow the last card or card image of the program, or can be entered on a separate input device. Control statements are used to specify program parameters.

The independent utility programs, except core dump, produce error and diagnostic messages on a user-assigned I/O device. Core dump program communicates through control panel lights. The error messages describe error conditions associated with the programs. Diagnostic messages describe and locate faulty conditions associated with the hardware. Error and diagnostic messages for TSS/360 independent utility programs are listed in System Messages.

SECTION 1: DIRECT-ACCESS STORAGE DEVICE INITIALIZATION (DASDI) PROGRAM - MODULE CEBDI

The Direct-Access Storage Device Initialization (DASDI) program, module CEBDI, initializes the IBM 2311 disk storage drive and the IBM 2314 direct-access storage facility in one of three formats: Sequential Access Method (SAM) or Virtual Access Method (VAM 1 and VAM 2). DASDI initializes the IBM 2301 Drum Storage in page (4096-byte) format.

ATTRIBUTES: DASDI operates outside and in support of TSS/360 as a privileged, nonrelocatable, serially-reusable program.

RESTRICTIONS: None.

HARDWARE CONFIGURATION REQUIREMENTS: The DASDI program runs on any standard model of System/360 with sufficient main storage and the required device configuration. The program is loaded as card deck or as card images on tape and is controlled from the console. DASDI machine requirements are:

Main Storage	
to initialize a 2311	65,536 bytes
to initialize a 2301 or 2314	131,072 bytes
Program Input Device	2540 or 2400 (9 track)
Control Statement Input Device	2540 or 2400 (9 track)
Message Output Device	1403, 1052 or 2400 (9 track)
Other	2311, 2314, or 2301

PROGRAM INITIATION AND OPERATION: The first four cards on the object deck are absolute loader cards, which load the program beginning at location 0. The first eight bytes become a PSW that branches control to location START.

START initializes the general registers, establishes base registers, sets up the PSWs, and then places the program in a wait state. When the wait state is entered, the program is set up to accept interrupts from either the console keyboard or the external interrupt key. DASDI is not interruption driven, but, when necessary, will enable I/O interruptions. If, as is usual, the interruption comes from the 1052 keyboard, the DASDI program causes the following message to be typed out:

```
CEB105A   DEFINE   INPUT   DEVICE.
```

The operator then indicates where the control cards are to be read from:

```
input=xxxx cuu
```

where xxxx is the device type (2540 or 2400), and cuu is the channel and unit number of the physical device address. If the 1052 is unavailable, the information may be entered via the console by storing into locations 0110 and 0111 (hexadecimal) one of the following:

```
xcuu
```

where x may be:

0 for 2540 card reader;
2 for a 2400 tape (9 track);

and where cuu is the channel and unit number of the physical device address.

For each method, DASDI fields the interruption and, after checking the validity of the input device, reads the control statements and begins processing.

For the 2311 and 2314 disk storage units, the information placed on cylinder 0, track 0 reflects the description of the volume to be initialized. Records 1 and 2 are used for IPL records. Record 3 contains the standard volume label, and records 4 through 10 provide space for up to seven user labels, as requested by appropriate control statements. If IPL text information is requested, and if text cards are supplied, up to 3600 bytes of text are inserted on cylinder 0, track 1. "Text" refers to cards punched in EBCDIC format; that is, the normal output of the BOS/360 or OS/360 assembler. If the IPL option is requested, but no text cards are supplied, DASDI inserts on the disk a pre-assembled TSS Startup Prelude program, which is used for the initial stages of system generation. The Startup Prelude description is found in System Generation and Maintenance.

The VTOC (for VAM 1) is placed as specified in the control statement with certain restrictions:

1. The VTOC may not be written on cylinder 0, track 0.
2. It must not be placed on track 1, if the IPL option is exercised.

3. It must not be placed on cylinder 199, if the pack is to be initialized in VAM format (this cylinder is used for the standard error retry procedures).

4. For SAM-formatted devices, the VTOC must not be placed on any alternate track.

If the VTOC extent specified does not end on a cylinder boundary, the remainder of the cylinder is padded in 140-byte records of zeros.

For SAM volumes, the remainder of each track following record zero, which has no data written on it, is erased. VAM volumes have 4096-byte records of zeros written on the track. Since the maximum size record for a 2311 device is 3625 bytes per track, the page-size record is written as a 3625-byte record on one track; the overflow bit is set in the flag byte. The remainder of the page is written on the next track. The remainder of this track has less than 3625 bytes left for use, and this is written in a similar manner, the overflow bit always being set and the combined records on successive tracks amounting to 4096 bytes. See Tables 4, 5, and 6 for track/page relationship.

Initialization of the 2301 drum is done in a modified VAM format. This is the formation of 4096-byte records alternated with 246-byte dummy records. Each 4096-byte record represents a VAM page, with nine pages for each even-odd pair of tracks. Unlike the disk units, the drum has no IPL, VTOC, or label records written on it. There is, however, a bit map at the beginning of the drum. This bit map is a modified version of the auxiliary storage allocation table (ASAT) as described in System Control Blocks PLM. Tracks 196 and 197 are marked unavailable due to the error retry requirements. The page which overflows from track 198 to track 199 is unavailable due to hardware restrictions.

OPERATION

The absolute loader gives control to DASDI at the START entry point. Control cards are read and analyzed to determine the requested function: initialization, obtaining an alternate track (GETALT), or clearing the PAT (VAM 2 only). The initialization routine is entered at either VOLCHK, for checking volume labels, or GEN00. The GETALT routine is entered at GETALTYZ. The PAT CLEAR routine is entered at DASDIPAT.

The initialization routine performs the following functions:

- Detect and list defective tracks.
- Write standard home address and record zero for each track.
- Write standard VTOC (SAM and VAM 1) or standard PAT (VAM 2).
- Write the optional IPL program.

SAM format devices have alternate tracks assigned to replace defective tracks. VAM format operation has defective tracks designated as unavailable. The alternate track (GETALT function) routine for SAM format devices performs the following:

- Selects first available alternate from VTOC.
- Chains the defective primary track to the alternate.
- Updates the VTOC availability list.

A normal DASDI exit results in a wait PSW (Es in the console lights), and an end-of-job message to the operator. Error conditions are noted by a PSW and the appropriate error message.

The DASDI routines are summarized in Table 1 and Chart AA.

DASDI I/O SUPERVISORY ROUTINES

The DASDI I/O supervisory routines provide these functions:

1. Program initialization and termination.
2. I/O processing: control, interruption, error.
3. Message generation.

INITIALIZATION ROUTINE

START Routine

START is given contol on entry to the program. This routine:

- Clears general registers.
- Establishes base registers.
- Establishes DASDI PSWs.
- Sets SEREP flags in the old machine check PSW.
- Enters wait state pending SYSIN device definition.

Table 1. DASDI Routines

Type	Name	Function
Initialization Routines	START	Clears registers and establishes DASDI PSWs.
		Sets up base registers, clears registers.
	CKINPUT	Sets up interrupt PSW, and resets switches; sets up SYSIN device from console input.
	ATTN	Prints messages to operator requesting input device definition.
I/O Routines	SYSIN	Stores buffer address in CCW and sets a read command.
	STARTIO	Controls all I/O operations performed by utility programs.
	CKCSW	Examines and classifies I/O interrupts.
Control Statement Analysis Routines	CLRSCAN	Performs control statement analysis, housekeeping functions on program switches and buffer areas, and links to RDCARD routine.
	RDCARD	Scans control cards and identifies the fields.
	CKVOLLBL	Matches control card specified ID with volume on disk pack.
	GEN00	Builds a pattern of hexadecimal 00 and 55 in core storage used to write out on disk for surface analysis.
Formatting	ADTOLIST	Accumulates addresses of defective tracks and produces listings of these addresses.
	CONSTR1	Verifies user labels, track conditions, format. Prepares to write track 00.
	CONSTR2	Writes track 00.
	FMTVTOC	Handles the requirements for building the VTOC for SAM-initialized devices.
	WRTVTOC	Writes the DSCBs constructed by FMTVTOC at the user-designated location.
	ENTPOIT1	Formats the 2311 (if entered at VAM2311) or the 2314 (if entered at VAM2314) in 4096-byte records and constructs the user-requested VTOC for VAM 1 format.
	VAM2301	Handles the patterning of the 2301 device in a modified VAM format.
	DASDIPAT	Removes data set indications from the PAT (VAM 2 only).
Termination	EOJ	Prints the end-of-job message, checks for the LASTCARD, and places the CPU in a terminating wait state.
	GETALTYZ	Sets Entry to process GETALT request, tests volume label checking switch.
	TRKPRNT	Causes the printing of a message, containing the addresses of the defective track and its assigned alternate.
Message Routines	PRINT	Writes messages using specified message output device.
	IOPRINT	Prints error messages, saves channel, CCW-status, and two sense bytes.
	WTO	Writes messages to the operator when input device is a typewriter.

When the operator enters the device code, control is passed to CKINPUT or ATTN as shown in the following decision table.

SYSIN = CONSOLE	Y	
SYSIN = TYPEWRITER		Y
CKINPUT	X	
ATTN		X

CKINPUT Routine

If SYSIN is the console, CKINPUT receives control following an external interruption. On entry, SYSIN device is tested and the appropriate processing routine is entered to fill in the UCB with the device code.

CKINPUT then isolates the channel number for a validity check. If the channel number is greater than 6, an error is indicated and the system enters a wait state. Assuming the channel number is not greater than 6, the SYSIN channel address is entered in the SYSIN UCB. Control then passes to CLRSCAN for control statement analysis.

ATTN Routine

When the SYSIN device is defined as a typewriter, ATTN is entered via an I/O interruption. On entry, the system mask is set to allow interruptions.

ATTN inserts the device type and channel number in the operator UCB (UCBOPR) and enters the write-to-operator routine (WTO) at location OPPRNT. WTO issues a message requesting the operator to enter the DASDI input device.

On return from WTO, ATTN loads parameter registers with the typewriter UCB and Read CCW. STARTIO is entered to read the input-device definition from the typewriter.

When control is returned from STARTIO, DEVLUP routine is entered with the device address to test for a valid device. If the device is invalid, a message is assigned and the system enters the wait state. If the device is valid, ATTN places the device address in the SYSIN UCB and passes control to CLRSCAN.

I/O ROUTINES

SYSIN Routine

SYSIN is entered by a branch-and-link instruction to the SYSIN entry point. The return address is specified in register 15; register 2 contains a buffer address.

On entry, all general registers are saved. SYSIN stores the buffer address from register 2 in the SYSIN CCW, and links to STARTIO with the addresses of the SYSIN UCB and CCW list. STARTIO reads data from the defined input device.

On return from STARTIO, registers are restored and control passes to the caller via register 15.

STARTIO Routine

STARTIO is entered at either the STARTIO or the STARTIOX entry points. Register 2 (UCBREG) contains the address of the device UCB; register 3 (CCWREG) contains the address of the CCW list.

On entry to STARTIO, all general registers are saved (STARTIOX skips this step). The routine loads the device address into register CSR4 and sets the system mask to allow I/O interruptions which are now serviced.

When all pending I/O interruptions are serviced, a test I/O loop is entered until the unit is not busy. At this point interruptions are disabled and the start I/O command is issued repeatedly until accepted. CKCSW is entered to test for I/O errors.

On exit, registers are restored and control is returned to the caller via register 9.

CKCSW Routine

CKCSW is entered to service all I/O interruptions. On entry, register 2 (UCBREG) contains the UCB address for the device which caused the interruption. CKCSW examines the status portion of the CSW to determine the type of interruption.

CONTROL STATEMENT ANALYSIS

When control statements are processed, they are checked for correct sequence, printed out on the MSG device, and switches are set. When the END statement, or an IPLTXT statement is encountered, all outstanding requests made by the control statements are honored.

GETALT statements are saved and the requests are honored upon receipt of the END statement or another GETALT request.

As each of the required and optional keyword parameters is encountered, specific areas are initialized for future analysis throughout the program. Table 2 represents a list of these areas, their respective lengths, and a description of their use.

Table 2. Keyword Parameters

Keyword	Length	Use
SEQSW	1 byte	checks for proper control card sequencing
KEYSEQ	1 byte	checks for the existence of the required keyword parameters
IPLSW	1 byte	the low-order bits indicate whether IPL has been requested and/or whether IPL text has been supplied
BRBYTE	1 byte	contains a hexadecimal code pertaining to the device type: 2311 - 08 2314 - 00 2301 - 10
VOLSECNO	1 byte	contains the volume security number obtained from the control statement VOLPASS parameter = 1 or 0.
SAVSPACE	1 byte	number of user labels
VTOCBGN	4 bytes	contains the beginning absolute track address (user-specified) of the VTOC.
VTOCEND	4 bytes	contains the number of tracks specified for the VTOC
PATADD	8 bytes	contains the relative page number of the PAT.
ROMAX	8 bytes	the last two bytes are used to specify the relative track capacity of the subject device: 2311 - 3,625 bytes 2314 - 7,294 bytes 2301 - 20,483 bytes
EXTNTAB	8 bytes	contains the extents of the VTOC and the address of the first available alternate (SAM)
VOLCHKSW	2 bytes	used to identify the action regarding the volume label checking option
BYPASS	1 byte	indicates the action to be taken regarding the program's surface analysis checking routines
ALTHOLD	6 bytes	contains the address of the first available alternate track and the total number of alternates (see EXTNTAB)
AMSW	1 byte	indicates the type of access method to be utilized in formatting the device: SAM - X'01' VAM 1 - X'02' VAM 2 - X'03'
VOLTSW	1 byte	indicates the status of the volume to be initialized: PAGING - X'40' PUBLIC - X'20' PRIVATE - X'00'
PVOLNOSW	1 byte	indicates whether a public volume is specified
PVOLHOLD	2 bytes	contains the public volume number
FLGANLSW	1 byte	indicates whether the FLAGTEST option is specified

The entry point to this routine is CLRSCAN. Housekeeping functions are first performed on program switches and buffer areas required by the program. This routine then links to the control card scan routine at RDCARD. RDCARD returns a pointer to a field and the length of that field in registers SCANADR and LENGTH, respectively, and an indication of the field type in location SWITCHRD. SWITCHRD is a one-byte switch with the following settings:

Bit	Value	Meaning
0	1	Control statement error
1	1	Bypass
2		Not Used
3	0	0 = First entry; 1 = subsequent entry
4	1	Command found
5	1	Keyword found
6	1	Parameter found
7	1	Right parenthesis found

Validity checks are then performed on the scanned data. If an error is detected in the input data, an attempt is made to print a message on the defined message output device. If the message output device is not defined, an attempt is made to issue the message, using the write to operator routine. If neither device is defined, a WAIT state is entered. When the message is successfully issued, a wait state is entered, and the program must be reinitiated and the corrected statement submitted.

At the completion of the control statement analysis routine, control is passed to either the initialization (GEN00) routine, the assigning of an alternate (GETALTYZ) routine, or clearing the PAT routine (DASDIPAT).

The initialization is entered at GEN00 if the user has specified VOLID=SCRATCH. In all other cases, the initialization is entered at VOLCHK, which passes control to CKVOLLBL to perform a validity check.

Volume Label Checking

The DASDI program compares the volume serial number of the object volume to that specified by the VOLID parameter, if both are present. If the VOLID parameter specifies SCRATCH, no comparison occurs. If a serial number is specified, and it is not equal to that in the volume label, or if the volume label is not present, this routine causes an appropriate message to be printed and terminates the program.

Entry point to the volume label checking routine in the DASDI program is at CKVOLLBL.

GEN00 Routine

The GEN00 routine sets up the surface analysis tables (TAB00 and TAB55) and determines the device and respective number of bytes per track to be tested. However, if the bypass option is selected (DADEF control card) this area is not used. Each track is tested by first checking for a good recording surface (INTALT). On disk storage, SAM format, the alternate tracks are checked first (the alternate track concept is not valid for VAM format). If the alternate track is found to be defective, it is flagged; later, FMTVTOC adjusts field six of the VTOC DSCB to indicate the number of available alternate tracks. If a primary track is found to be defective, an alternate is assigned by ASGNALT, and a message to this effect is printed. If the device to be initialized is a VAM format, the addresses of defective tracks are accumulated.

Upon encountering a defective track, control is passed to ADTOLIST, which accumulates the addresses of bad tracks. When 200 addresses are noted, they are printed out with the option to continue or terminate the job. The CPU is placed in the wait state with hexidecimal FFFFs displayed on the console. If the option to continue is selected, the buffer area is cleared and control returns to the surface analysis routine. When all tracks are checked, control passes to ADTOLIST, which prints the track addresses in the buffer.

FORMAT ROUTINES

ADTOLIST Routine

ADTOLIST is entered to accumlate and print a list of defective or unavailable tracks. On entry, registers are saved and a base register is established. ADTOLIST first executes a series of compares to determine the entry condition and the action to be taken. Branches are made accordingly.

ADTOLIST is entered by the surface analysis routines (VAM2311, VAM2314, VAM2301) to accumulate defective track addresses in LISTAREA and BUFFERME. When 200 defective track addresses are stored in BUFFERME, the list is printed and the system enters a wait state. The operator may, at this point, continue or terminate the job. Assuming the job is continued, BUFFERME is cleared for further accumulation of defective track addresses.

On exit, ADTOLIST places an ID code in register 2 which indicates the action to be taken. The codes are found in Table 3.

The tracks on a device are tested in the following manner: the home address (HA) is written followed by a maximum-length record zero consisting of a data field of identical bytes of hexadecimal 00. The track is then read and checked. The home address and a maximum-length record zero is again written, this time consisting of a data field of identical bytes of hexadecimal 55. The track is again read and checked. The home address and record zero (track descriptor record 8-byte count field followed by an 8-byte data field of zeros) are then written and they are both read again for a final test.

If this procedure is successful, the track is flagged as operative. If an error is discovered during the above routines, the steps are repeated ten more times. If an error does not occur, the track is flagged as operative. However, should an error occur during the retry phase, the track is noted as defective with the proper action taken.

If an error occurs during surface analysis, the appropriate action depends on the location, device type, access method, or a combination of these factors. When writing the home address and track descriptor record in SAM format, an error results in job termination, inasmuch as an alternate cannot be assigned. If the VAM format is employed, the corresponding pages are noted as unavailable and the job continues.

Table 3. ADTOLIST Identifying Code

ID Code	Action to be taken
1	Surface analysis error on a 2311
2	Surface analysis error on a 2314
3	Formatting finished
4	Formatting error on a 2311
5	Surface analysis error on a 2301
6	IPL requested on a 2311 and 2314
7	Error retry cylinder on a 2311
8	Error retry cylinder on a 2314
9	Error on an error retry track
10	Finished with surface analysis
11	Not used
12	Formatting error on a 2314
13	Formatting error on a 2301
14	Error retry tracks on a 2301

Regardless of the device type or the format, an error on cylinder 0, track 00, results in job termination. Also, the job is terminated if (1) an error occurs on cylinder 0, track 01 (only for the 2311 and 2314 where IPL has been requested), and (2) if an error occurs on the track designated for the VTOC.

Finally, an error occurring on cylinder 199 (VAM 1 or VAM 2) produces a message to the operator giving the option to continue or terminate the job. In the case of the 2301 drum, an error on tracks 196/197 produces the same option message. Errors occurring on any other cylinders/tracks result in either the assigning of an alternate (SAM), or making the corresponding pages unavailable (VAM 1 or VAM 2).

When all tracks are initialized, control is passed to CONSTR1, which builds an image of track zero in main storage (see Figure 1). If however, the device is the 2301 drum, control is passed to VAM2301.

CONSTR1 Routine

CONSTR1 is entered to construct and write track zero. CCWs are saved on entry.

CONSTR1 first checks to see if additional user labels are requested. If they are, YESUSER writes up to seven user-supplied additional volume labels as records four through ten. Space is allocated consisting of a 4-byte key field and an 80-byte data field for the labels. Assuming that there are no additional labels requested, NOUSER determines if IPL text is requested. If it is, LOADA reads 3600 bytes following the IPLTXT card into a buffer area (or until an END card is read). For both the 2311 and 2314, the IPL text is placed on track 1 of cylinder 0.

If the IPL function is selected, records one and two are written on track zero as an IPL bootstrap program and a program to load the IPL initialization program. If the IPL function is not selected, record one is written as a program to set a wait state in the CPU in case the volume is loaded for execution. In either case, record two is written as an IPL bootstrap. Since record one will set the wait state, in a non-IPL situation there is no danger of executing record two.

The user may, at his discretion, submit text cards for an IPL option, which is placed on cylinder 0, track 1, or he may

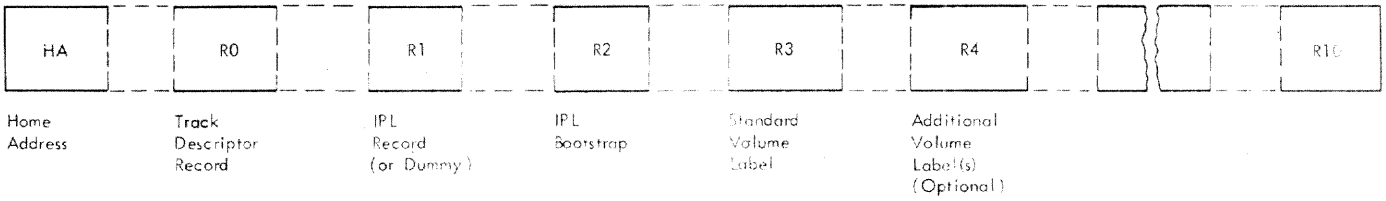


Figure 1. Image of Track Zero

request the IPL option without supplying text cards. In the latter case, a special program (Startup Prelude), which is used when the TSS/360 is first generated, is written on cylinder 0, track 1. This program is incorporated in DASDI. In either case, the user must supply the IPLTXT control statement.

CONSTR2 Routine

After the information on track zero is constructed by CONSTR1, CONSTR2 performs the necessary functions to write the track. WRTIPL writes the user-supplied IPL initialization program, if requested. The IPL program is written on cylinder zero, track one.

At this point the device type/ access method becomes important. With either the 2311 or 2314, SAM format, the program sets the format for the VTOC at FMTVTOC. If it is a 2311/2314 VAM 1 or VAM 2, control is passed to the ENTPOIT1 routine. The 2301 is handled by a separate routine as it has special requirements for its format.

FMTVTOC Routine

FMTVTOC is entered by a branch instruction from CONSTR2 to format SAM VTOC DSCBs. On entry, all general registers are saved. The following are immediately stored in the VTOC:

- First alternate track and count
- VTOC indicator
- Device size
- VTOC extents
- Highest prime CCHHR address

FMTVTOC then calculates the beginning CCHHR of the VTOC and the device type (2311 or 2314 disk device). Cylinder zero tracks are freed, and the VTOC is constructed. The EXTNT routine is entered to insert the VTOC extents. All data set extents are checked and the VTOC is written using WRTVTOC.

WRTVTOC Routine

WRTVTOC is entered by a branch statement to the WRTVTOC entry point. A loop is entered to build all CCWs and count fields necessary to write DSCBs constructed by FMTVTOC. After all DSCBs have been written, a string of 140-byte dummy DSCBs are written to pad the cylinder.

Upon completion, control is passed to the end-of-job routine at EOJ.

ENTPOIT1 Routine

The routine is entered from CONSTR2 and is responsible for setting the format for the 2311/2314 in a VAM 1 or VAM 2 4096-byte page format as well as constructing the user-requested VTOC. If IPL is requested, page 1 is not available for use. However, if IPL text is not requested, page 1 (starting on track 1) is available. As each track is processed, it must first be tested to see if it has successfully been surface-analyzed. ADTOLIST passes the address of a table, LISTAREA, which contains the addresses of those tracks that are defective. The VTOC location must be determined before the format is set.

The format technique involves the use of chained commands in a five-track cyclic pattern. When processing pages in VAM 1 or VAM format 2, which overflow, the first portion of the record is written using a write special count, key, and data command. The second portion of the overflow record is then written with the rest of the track. The second track is then verified. If both tracks are good the first portion of the record is rewritten using a write special count, key, and data to set the overflow bit in the flag byte of the count field. The overflow page is then verified for correct format. After the pattern is generated, the 4096-byte records are placed on the device. Before completing the format for the entire disk, control again goes to ADTOLIST, which now prints the addresses of defective tracks with the continue/ terminate option.

The VTOC (for VAM 1), which contains the available pages rather than relative track

addresses, is finalized and placed in the user-specified location. Dummy DSCBs are created to fill the entire cylinder. See Tables 4 and 5 for page/track relationship.

VAM2301 Routine

This specialized routine handles the patterning of the 2301 device in a modified VAM format. Within this context, page zero contains a bit map of the paging drum. The bit map uses the same format as the auxiliary storage allocation table (ASAT), but without the 16 byte ASAT header. The ASAT is described in System Control Blocks PLM.

For all defective tracks noted during surface analysis or format processing, the pages associated with that track are made unavailable. Track zero, record one, contains a bit map consisting of 144 contiguous bytes with the first 100 bits representing the pages of a slot. A zero bit indicates availability, a one bit unavailability. For each defective track encountered, a minimum of five pages is made unavailable (the four full-size pages and one overflow page).

See Table 6 for 2301 page/track relationship.

The modified VAM format begins on track zero, record two, and consists of odd-numbered records of 4096 bytes and even-numbered records of 246 bytes (dummy records). On even-numbered tracks, there are four page-sized records interspersed with four dummy records. The last record is a half-page and overflows to the next track. The odd-numbered tracks begin with the corresponding latter half of the overflow

Table 4. 2314 Page/Track Relationship

Track	Bytes/Record	Page No.
0	4,096	0
0	2,920	1
1	1,176	
1	4,096	2
1	1,592	3
2	2,504	
2	4,096	4
2	207	5
3	3,889	
3	3,136	6
4	960	
4	4,096	7
4	(1,187 unused bytes on track 4)	

Note: This pattern is repeated for 20 tracks, 32 pages per cylinder.

Table 5. 2311 Page/Track Relationship

Track	Bytes/Record	Page No.
0	3,625	0
1	471	0
1	3,069	1
2	1,027	1
2	2,486	2
3	1,610	2
3	1,875	3
4	2,221	3
	(1,234 unused bytes on track 4)	
5	3,625	4
6	471	4
6	3,069	5
7	1,027	5
7	2,486	6
8	1,610	6
8	1,875	7
9	2,221	7
	(1,234 unused bytes on track 9)	

pages (2048 bytes), and the rest of the track consists of interspersed dummy and page-size records.

The routine communicates with ADTOLIST to properly accommodate defective tracks. Upon completing the format processing and the status of the bit map, the routine passes control to EOJ.

DASDIPAT Routine

This routine is called when a PAT Clear function is requested. DASDIPAT checks the volume label to determine if the disk is a 2311 or a 2314; then reads the PAT page(s). It now removes all data set indications from the PAT, while leaving error pages flagged. The PAT is written back to the disk, and the job is terminated. Control is passed to PATWRITE.

TERMINATION ROUTINES

End-of-Job (EOJ) Routine

The EOJ routine prints the end-of-job message using the PRNT routine; checks for the LASTCARD (if operating within a stacked job environment); writes tape EOF and rewinds the tape for SYSOUT tape (TAPEEOF routine); and places the CPU in a terminating wait state.

Table 6. 2301 Page/Track Relationship

Track No.	Record No.	Bytes/Record	Page No.
0	1	4,096	0
0	2	246	
0	3	4,096	1
0	4	246	
0	5	4,096	2
0	6	246	
0	7	4,096	3
0	8	246	
0	9	2,048	
1	1	2,048	4
1	2	246	
1	3	4,096	5
1	4	246	
1	5	4,096	6
1	6	246	
1	7	4,096	7
1	8	246	
1	9	4,096	8

GETALTYZ Routine

Control is passed to this routine from CLRSCAN upon encountering a GETALT control card. The volume label checking switch is tested and, if requested, control is passed to CKVOLLBL.

If the track check "bypass" is not selected, track analysis on the user-specified track is performed. If the track is found to be operative, a message to that effect is printed (or displayed) and the next GETALT request is processed.

If the track check bypass is selected, or if the track is found to be defective, control passes to ASGNALT (SAM format). This routine flags the given track as defective and assigns it an alternate (if it is a primary track). If the defective track is an alternate, it is flagged as such; if the given alternate track has been assigned to the primary, an operative alternate is assigned to the primary. If the format is VAM 2, the PAT is read, defective pages are flagged, and the PAT is written.

TRKPRNT Routine

The TRKPRNT routine causes a message to be printed, stating the address of each defective track and its assigned alternate. Routine GETALT4 decrements field 6 of the VTOC to reflect the fact that one less alternate track is available, and increments field five to point to the next available alternate track.

Control is then given to ALTRTN which repeats the process for another user-specified GETALT request. If none exists, it encounters a valid END statement, initiates a normal end-of-job, and sets the CPU to the wait state.

MESSAGE ROUTINES

Message Output (PRNT) Routine

PRNT is entered at the PRNT entry point to write a message to the UCBSYO-defined message output device. On entry, registers are saved and SYSOUT is entered to print one line. On return from SYSOUT, registers are restored and return is made to the calling routine through register 9.

I/O Error Print (IOPRNT) Routine

When there is a hardware error, or a situation where intervention is required, information from the UCB and the command code from the last I/O instruction is printed in addition to an error message. In these cases, IOPRNT is called. IOPRNT saves the channel-byte and unit-byte, the CSW status bytes, two sense bytes, and command code. The I/O PRNT routine prints these bytes along with the error messages by going to OPPRNT (at OPPRN1).

Write-to-Operator (WTO) Routine

WTO routine is entered via a branch through register 15 to the OPPRNT entry point. WTO writes a message to the operator when the MSG device is defined as a typewriter.

On entry, register 1 contains the message length; register 2, the message address. WTO completes the OPRCCW and enters STARTIO with the addresses of UCBOPR and OPRCCW.

On return from STARTIO, control is returned to the user via register 15.

Error Messages: Error messages are listed in System Messages.

Diagnostic Messages: Diagnostic messages appear in the following format:

number (16-byte text) cuuxxssssyyyy

where c is the channel of the device in error, uu is the unit, xx is the command code, ssss are the status bytes from the channel status word, and yyyy are the sense bytes.

Message texts are listed in System Messages.

Chart AA. DASDI Routine (Part 1 of 3)

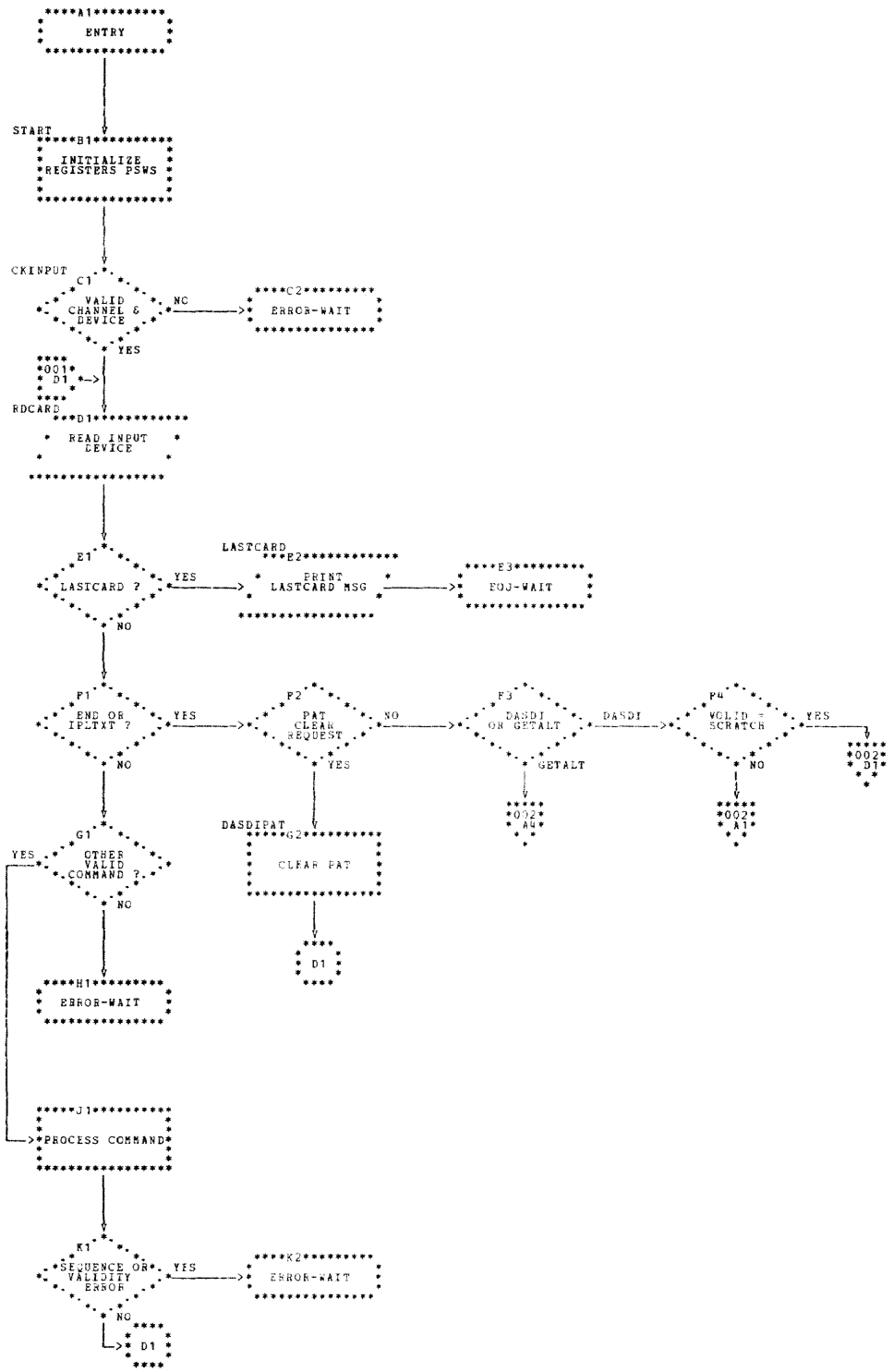


Chart AA. DASDI Routine (Part 2 of 3)

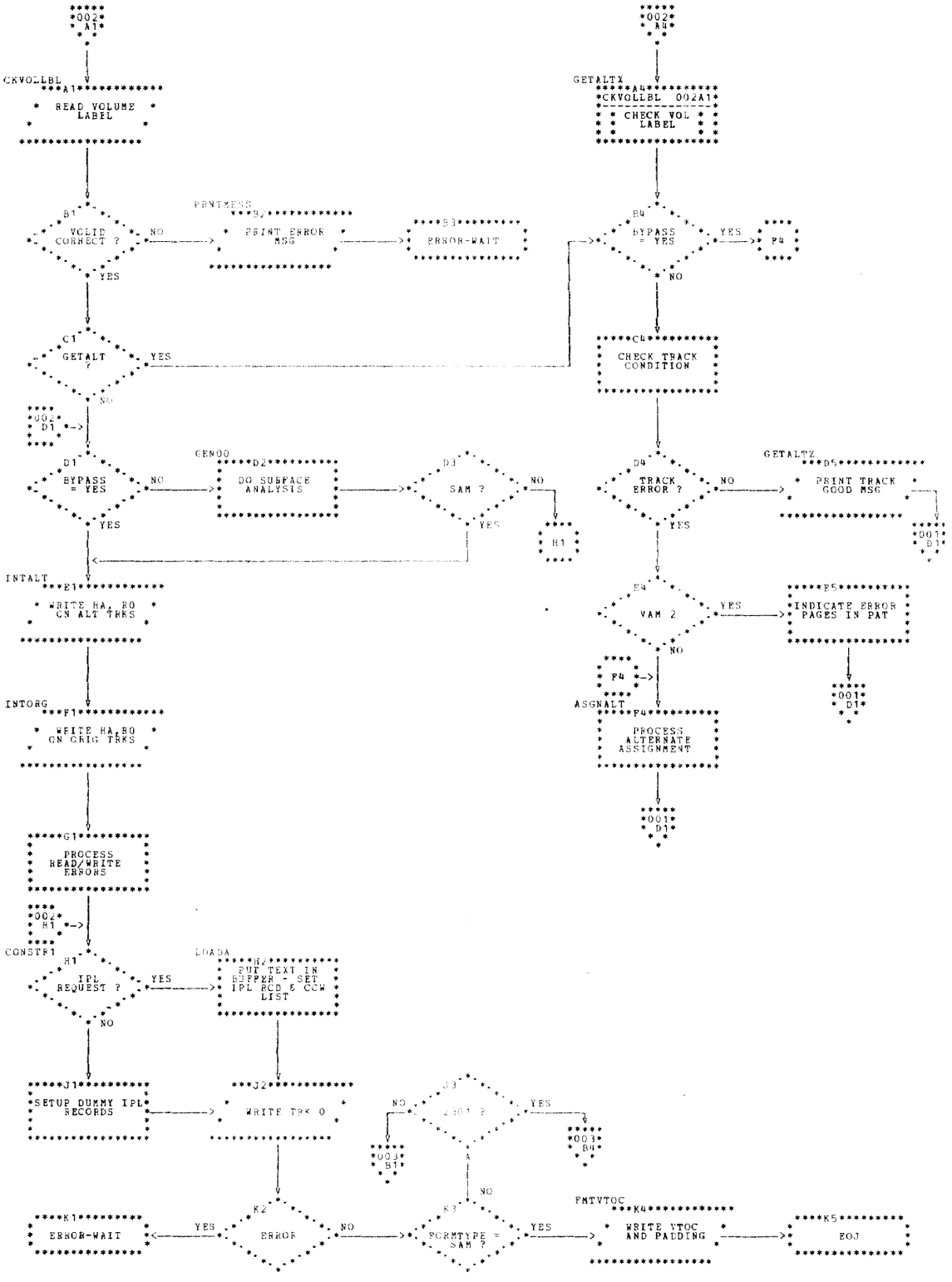
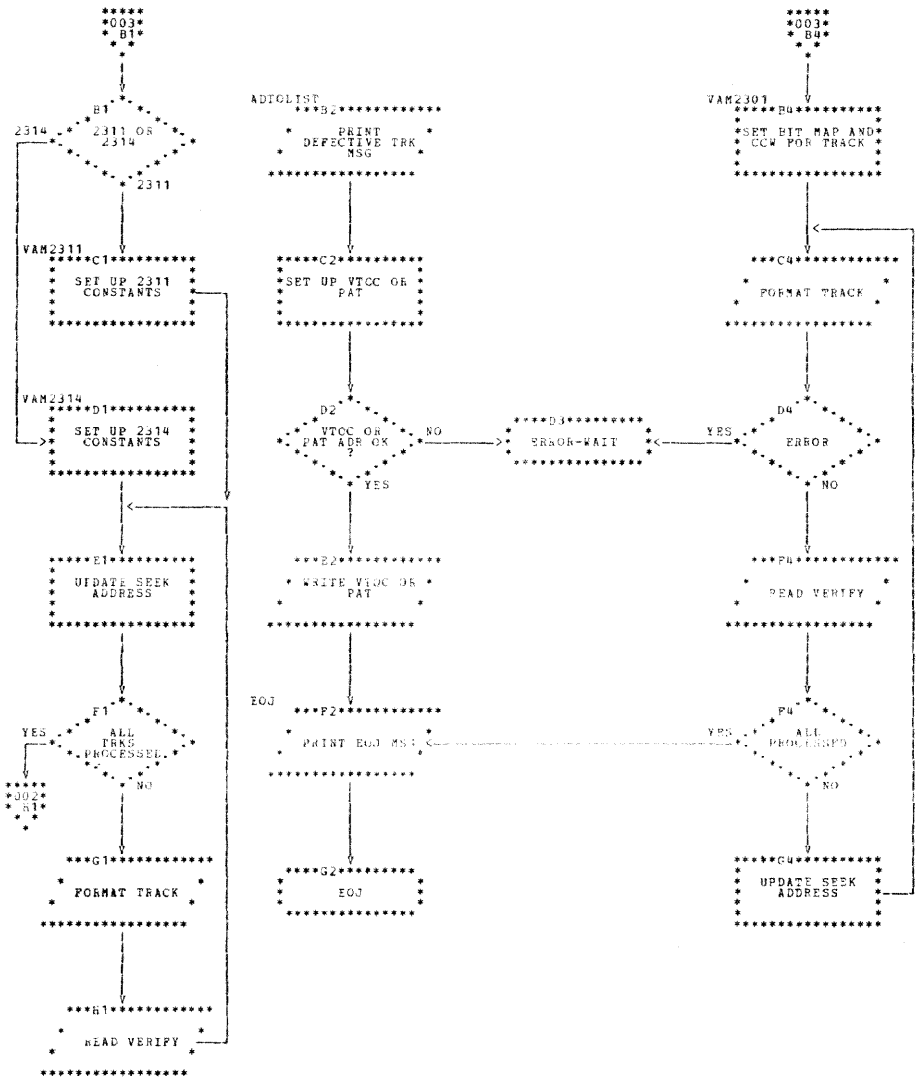


Chart AA. DASDI Routine (Part 3 of 3)



SECTION 2: DUMP/RESTORE (DASDDR) PROGRAM - MODULE CEBDR

The Dump/Restore (DASDDR) program, module CEBDR, dumps and restores data on direct-access volumes on IBM 2311 disk storage device, and 2314 direct-access storage facility. The data contents of the direct-access volume, except for the home address, can be dumped onto another IBM 2311 or 2314 disk storage volume, or onto a magnetic tape, to be restored onto a direct-access volume that resides on the same type of device as the source volume. The volume onto which data is to be dumped or restored must have been initialized using the TSS/360 DASDI program (Section 1). DASDDR is useful for preparing transportable copies and backup copies of direct-access volume contents.

ATTRIBUTES: DASDDR operates outside and in support of TSS/360 as a privileged, disabled, non-relocatable, non-reusable program.

RESTRICTIONS: The following assumptions and restrictions apply:

1. DASD contents always refer to the data content exclusive of the home address and track descriptor record count field.

2. The 2311 or 2314 to be used for dump/restore must have been previously initialized with home addresses and track descriptor records and, in the case of SAM format, have bad tracks flagged with alternates assigned.

3. Partial 2311 or 2314 contents may be dumped/restored between contiguous track limits (SAM or VAM 1 only). However, the program makes no attempt to update the volume table of contents (VTOC) or provide pointers to data. Selection of a partial dump/restore places upon the user the responsibility for any and all required updates.

4. The program makes no provision to restore to a track address other than that from which the data contents were dumped. The restore must be to the same address limits used in dumping the original 2311 or 2314. The program safeguards this requirement by providing the restore limits required.

HARDWARE CONFIGURATION REQUIREMENTS: The DASDDR program runs on a System/360 Model 67 in non-relocatable PSW mode, or on any System/360 with sufficient main storage and required device configuration:

Main Storage	65,536 bytes
Program Input Device	2540, 2400 (9 track)
Control Statement Input Device	2540, 2400 (9 track)
Message Output Device	1403, 1052, 2400 (9 track)
Other	2311 or 2314; or 2400 (9 track) for intermediate storage

PROGRAM OPERATION

DUMPED DATA FORMAT (SAM OR VAM 1)

The format of dumped data depends on the device configuration of the dump: 2311 to 2311, 2314 to 2314, or either 2311 or 2314 to tape.

2311 to 2311 Dump

Data from the input 2311 is copied, record for record, and track for track. For this reason, a restore from 2311 to 2311 is not provided, but can be effected by another dump.

2314 to 2314 Dump

Data from the input 2314 is copied, record for record, and track for track. For this reason a restore from 2314 to 2314 is not provided, but can be effected by another dump.

2311 or 2314 to Tape

The following records are written on tape for a 2311 or 2314 to tape dump (see Figure 2):

1. A limits record is written as the first record (following any labels) on each volume of tape. This record contains the addresses of the first track dumped, the last track dumped, and the first track dumped on this volume of tape.

2. A bit map record (see BLDMAP) is written next so that bad pages on the original device can be recovered when this tape is restored. (This record is used only for a VAM 1 format device.)

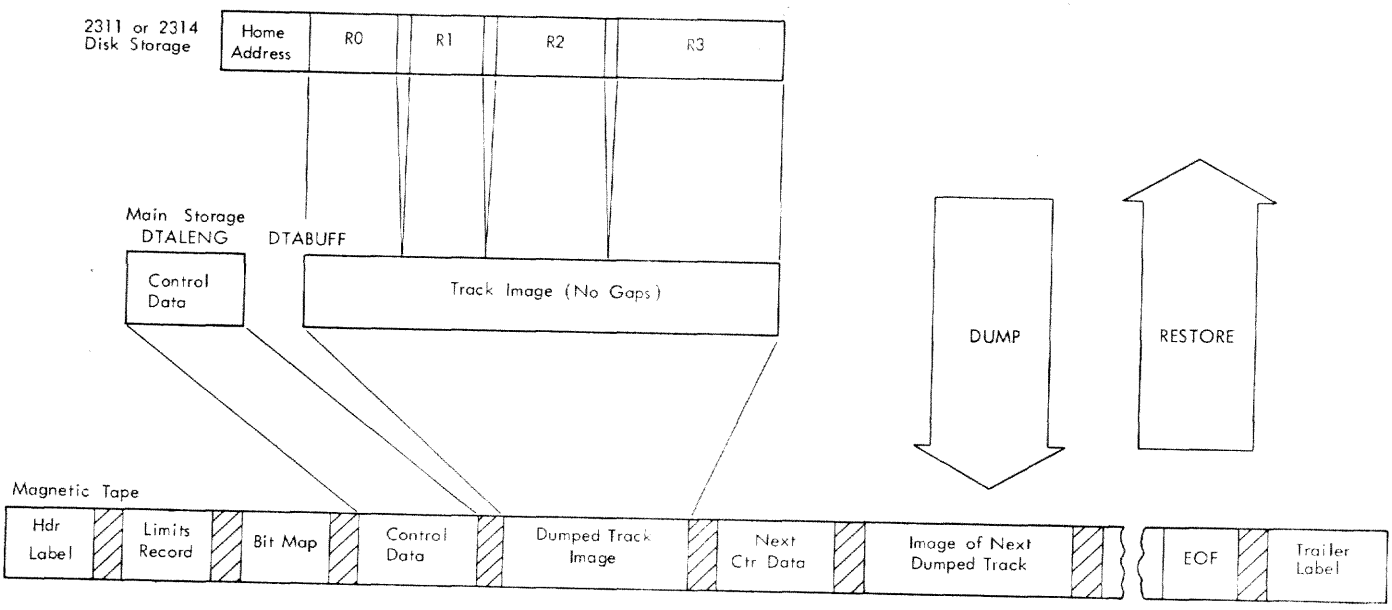


Figure 2. Dumping and Restoring a Direct-Access Track (SAM and VAM 1)

3. A control record is written for each track dumped, immediately preceding the data dumped from the track. The control record contains a channel program to be used by a subsequent restore to write one track.

DUMPED DATA FORMAT (VAM 2)

2311 to 2311, 2311 to 2314, 2314 to 2314 Dump

The following data are dumped to the output disk storage device;

1. IPL records
2. Label records
3. IPLTXT records
4. All pages in use (as indicated by the Page Assignment Table)

A restore from disk to disk is provided only by another dump.

2311 or 2314 to Tape

The following records are written on tape (see Figure 3).

1a. Unlabeled tapes -- a limits record is written as the first record on each tape volume.

1b. Labeled tapes -- TSS standard labels are written, following the volume label, on each tape volume.

2. The Page Assignment Table (PAT) of the from-device.
3. IPL, Label, and IPLTXT records.
4. Pages in use (as indicated by the PAT).

PROGRAM FLOW

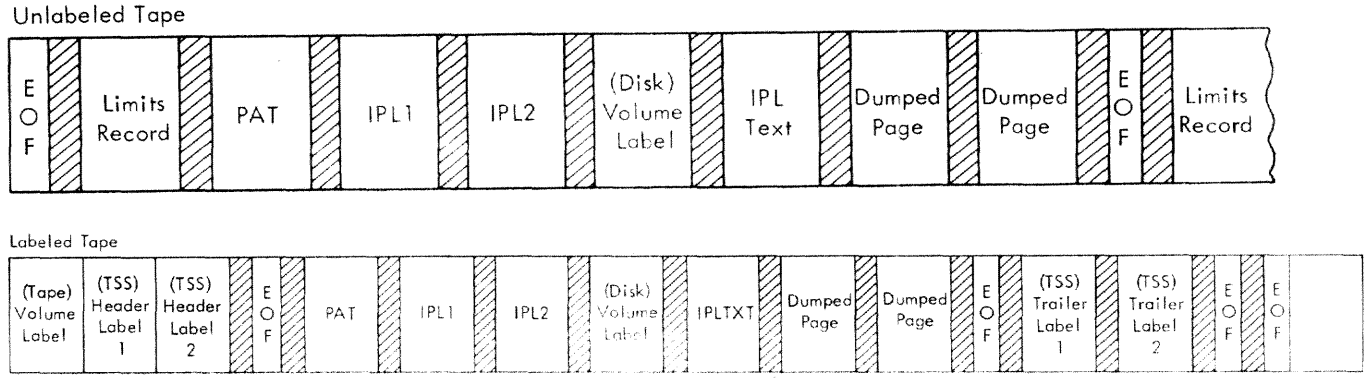
When the dump/restore program receives control after IPL, base registers 11, 12, 13, and 14 are initialized to hexadecimal 1000, 2000, 3000, and 4000, respectively. Then the SEREP flags are set in the old machine check PSW, and the wait state is entered until SYSIN is defined through either CKINPUT or ATTN routines. All I/O operations are controlled through the DASDR I/O supervisory routines, explained at the end of this section.

The dump/restore program accomplishes its purposes through the following routines summarized in Table 7 and Chart BB.

INITIALIZATION ROUTINES

Start Routine

START receives control after completion of IPL, and initializes base registers 11, 12, 13, and 14 to hexadecimal 1000, 2000, 3000, and 4000 respectively. Dump/Restore PSWs are established and the SEREP flags in the old machine check PSW are set to hexadecimal decimal 'FF'. The program then enters a wait state to allow CKINPUT or ATTN to define the SYSIN device.



• Figure 3. VAM 2 Dump Tape Format

CKINPUT Routine

CKINPUT receives control when the operator defines SYSIN through the console. Control is passed via an external interrupt generated by the interrupt key. CKINPUT first checks the entered code to verify device type (2540, 2400, or 1402).

If SYSIN is a 2540 or 2400, the SYSIN UCB (UCBSYI) is modified and the channel number is tested. If the channel number is greater than 6, an error results. If the channel number is valid, the SYSIN address is placed in UCBSYI. The system is set to receive interrupts and control is passed to CLRSCAN. If SYSIN is defined as a 1402, start is reentered for a request.

An error message results when an invalid SYSIN device is entered. The system enters a wait state until another device type is entered.

ATTN Routine

If the operator defines the input device through the typewriter, ATTN assumes control via an I/O interruption. ATTN sets up a UCBOPR and calls the write-to-operator routine at location OPPRNT, which, in turn, calls the STARTIO routine to print a message to the operator instructing him to type in the input device. The routine assumes control again, and reads the input device from the typewriter, converting lower case characters to upper case characters, if necessary. It then calls DEVLUP, which scans the DEVTAB table to ascertain that the input device is valid. If it is, the UCBSYI is then filled in with the device address which is checked for validity, and control is passed to CLRSCAN. If the input device is not valid, an error message is printed.

CONTROL STATEMENT ANALYSIS ROUTINES

CLRSCAN Routine

CLRSCAN analyzes control statements, checks them for sequence, prints them on the MSG device, and sets switches. When the END statement is encountered, all outstanding requests made on control statements are honored.

As each of the required and optional keyword parameters is encountered, switches are initialized for future use throughout the program.

RDCARD is used to identify fields in statements and to read the next statement, when necessary.

The entry point to this routine is CLRSCAN. Housekeeping functions are first performed on program switches and buffer areas required by the program. This routine then links to the control card scan routine at RDCARD. RDCARD returns a pointer to a field, and the length of the field, in registers REG1 and REG2, respectively, and an indication of the field type in location SWITCHRD. SWITCHRD is a 1-byte switch with the following settings:

Bit	Value	Meaning
0	1	Control statement error
1	1	Bypass
2		Not Used
3	0	0 = First entry; 1 = subsequent entry
4	1	Command found
5	1	Keyword found
6	1	Parameter found
7	1	Right parenthesis found

Validity checks are then performed on the scanned data. If an error is detected in the input data, an attempt is made to print a message on the defined message output device. If the message output device is not defined, an attempt is made to issue the message, using the Write to Operator routine. If neither device is defined, a wait state is entered. If the message is successfully issued, a wait state is entered, and the program must be reinitiated, and the corrected statement submitted.

At completion of the control statement analysis, control is given to the dump/restore program at START1.

Table 7. DASDDR Routines (Part 1 of 3)

Type	Name	Function
Initialization Routines	START	Initializes base registers, and sets SEREP flags.
	CKINPUT	Checks operator-entered code, when input device is defined through the console.
	ATTN	Checks operator-entered code, when input device is defined through typewriter.
Control Statement Analysis Routine	CLRSCAN	Performs control statement analysis, and housekeeping functions on program switches and buffer areas.
Processing Routines	START1	Sets tape switches and increments the end address.
	BEG1	Restores and loads addresses, and compares to and from-device UCBs.
	HAVEADDR	Builds CNTRLBUF.
	TOTAPE	Sets tape switch and links to STARTIO to rewind the tape. Verifies density and format.
	TOFM2311	Checks volume label.
	FRMTAPE	Reads beginning and end addresses. Checks format.
	CKVOLLBL	Obtains and checks volume label, and checks for VAM or SAM format.
	BLDMAP	Builds bit map and analyzes VTOC extents and data set control blocks.
	BUILDCCW	Builds a list of Read Count CCWs to read maximum number of records in this device.
	MODTKADF	Increments track address, after RSTRTAPE, to test EOJ.
	ANALSENS	Verifies previous read count.
	CNGRDCKD	Converts read count CCWs to read CKD.
	PREPRWRT	Writes length fields in tape-write CCWs.
	TSTWRTSP	Change DASD CCWs to perform a write operation.
	DMPDASD	Executes a converted channel program during a future restore, if dump is 2311 or 2314 to tape.
	DMPDTA	Checks to see if the record to be written is noise.
	PUTMAP	Prints proper bit map on tape for 2311 or 2314.
	RSTRTAPE	Reads the control record and the record following it to restore data from tape.
	MODTKADT	Increments track address after a Read operation to test for EOJ.
	WRTDISK	Completes relocation on overflow from bad track and prevents overlaying pages used for relocation.

Table 7. DASDDR Routines (Part 2 of 3)

Type	Name	Function
Processing Routines (cont'd.)	WRTDISKY	Executes channel program by writing one track onto 2311 or 2314 for 2311-2311 or 2314-2314 dump or a restore.
	CHTOSKP	Computes length of primary and duplicate CCW lists.
	WRTCHECK	Reads back the track just written into a second buffer, and compares the contents of the two buffers to ensure a successful write.
	CHANGCYL	Updates cylinder address.
	CKALT	Checks for a bad SAM format track; places alternate track address in CCHH format.
	CKALT1	Determines if last track written is bad (SAM format).
	GETCCW	Checks whether SAM or VAM. If SAM, reads alternate track for defective primary on a RESTORE.
	BADPAGE	Interrogates the bit map to see if the defective page is available.
	RLOCAT	Relocates a data page for bad track on the to-device.
	ENDREEL	Writes data buffer record when end-of-tape.
	EOJ1	Writes trailer label on current to-volume to insure that a new volume is mounted.
	EOJA	Branch to print EOJ message. Rewind tape.
	ENDOFJOB	Prints end-of-job/reel/disk pack message.
	EOJAA	Given control at conclusion of an entire 2311-2311 or 2314-2314 dump. Updates field six of VTOC DCSB for SAM.
	DADSM	Overlays the extents with the extents shown in the bit map.
	RBLDDSCB	Updates DSCB, following page relocation.
	TAPEEOF	Routes exits at end of tape.
	VAM2DUMP	Dumps contents of a VAM 2 disk to tape or disk.
	CNVRELPG	Converts a relative page number to cylinder, track, and record number.
	CLRERPG	Clears error indicators, and updates PAT.
	VAM2REST	Restores tape contents to a VAM 2 disk.
	RELOCATE	Relocates page on a VAM 2 disk.
	V2UNEXCP	Handles unit exceptions on tape for VAM 2 dump/restore operations.
	V2WRTCHK	Compares read/write buffers to insure a successful write.
	ENDOFPAT	Writes a control record, rewinds tape, and prints EOJ message.
	PATWRITE	Writes updated PAT on disk, and prints EOJ message.

Table 7. DASDDR Routines (Part 3 of 3)

Type	Name	Function
Processing Routines (cont'd.)	TPDPLAB	For a dump, ensures tape has a TSS standard volume label or is capable of being IPLd.
	RSTPCK	For a restore, ensures tape has TSS standard labels or a control record.
I/O Routines	Data Input Routine (SYSIN)	Stores buffer address in channel command word, and sets a read command.
	STARTIO	Controls all I/O operations performed by the DASDDR program.
	I/O Interruption Analysis (CKCSW)	Examines and classifies I/O interrupts.
Message Routines	PRNT	Writes messages using the message output device defined by the MSG statement.
	IOPRNT	Prints error message, and saves channel-, unit-, CCW-status, and two sense-bytes.
	OPPRNT	Writes messages to the operator when input device is a typewriter.

PROCESSING ROUTINES

START1 Routine

START1 sets the tape switch to 0, indicating no tape is involved in the program function. If a tape is involved, the switch is later set to 1. The end address, which is the last track of the disk for an entire dump/restore, is incremented by one so that all data of the desired area are dumped. This is handled for the from-device by CHANGCYL.

BEG1 Routine

BEG1 first restores the address returned by CHANGCYL via register 15 to ENDADR. Register 15 is then loaded with the address contained in BGNADR; the incrementing factor is set to 8. Register 5, which links to the correct dump/restore routine, is loaded with the address of TOFM2311, specifying a 2311-2311 or 2314-2314 dump. The to- and from- device UCBs are then compared to see if they are for the same device type. If they are, control is given to HAVEADDR. If not, the SEQSW is tested to see if this is a dump request. If it is a dump request, register 5 is loaded with the address of TOTAPE and control is given to HAVEADDR. If it is not a dump, control is given to FRMTAPE.

HAVEADDR Routine

HAVEADDR is entered from BEG1. The routine enters information into the 8-word control data record (CNTRLBUF), as shown in Figure 4. Control is then given to the proper preliminary routine specified by BEG1 in register 5.

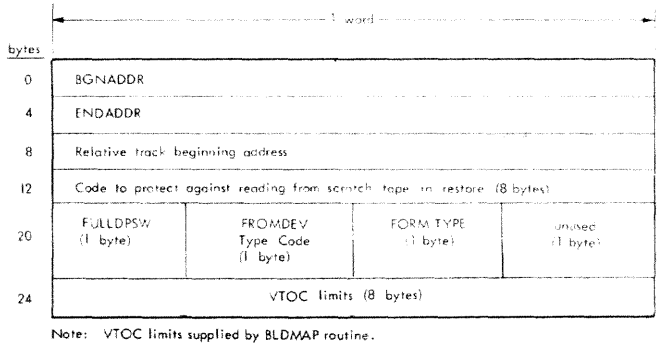
TOTAPE Routine

This routine assumes that the tape, in a tape dump, is positioned at the load point and that the begin and end addresses are available.

TOTAPE first assumes that the record is not noise. The tape switch is set on; STARTIO is entered to position the tape. The tape density is then verified.

The routine reads the first tape label. If this is a DASD loader card, RDLABELS is entered to bypass the remaining labels. If the first record is not a label, the VTOC area is cleared, and the format is verified. SAM format is processed by TODMPDTA. VAM 1 format records are checked for a bit map. When none exists, PUTMAP is entered. If the format is VAM 2, control is given to VAM2DUMP.

Exit is made to DMPSAM.



Note: VTOC limits supplied by BLDMAP routine.

Figure 4. Entering Information Into CNTRLBUF

TOFM2311 Routine

TOFM2311 is entered through register 5 by HAVEADDR. The routine immediately gives control via register 5 to CKVOLLBL, to check the volume label. On return, register 5 is loaded with the address of WRTDISK, the routine which prepares to write the disk. Control is then given to BUILDCCW.

FRMTAPE Routine

On entry, this routine gives control immediately to CKVOLLBL, to check the volume label. On return, register 5 is loaded with the address of RSTRTAPE. A check is made to see that the tape is at load point. Then the labels are bypassed, and the beginning and end addresses are read. If this is not a restore tape, or if the restore tape specifies a format type (SAM, VAM 1, VAM 2) other than that specified by the companion dump, an appropriate message is written and a wait state is entered. If the tape is correct, the routine stores the beginning and end addresses, gets the relative beginning address, and recovers FULLDPSW to see if this is a restore of a full or partial dump. If the format is VAM 2, control is passed to VAM2DUMP. Otherwise, control is then passed to RSTRTAPE via register 5.

CKVOLLBL Routine

CKVOLLBL is entered by a BAL statement using register 5 for return. The routine links to STARTIO to get a volume label. If the label is not a system volume label, control is given to PRNT to print the invalid VOLID statement. The linkage register is loaded with the address of WAITZ, which sets the CPU in a terminal wait state.

A system volume label is checked to see if it agrees with the VOLID specified. If not, PRNT is called as before. When the valid system volume label agrees with the specified VOLID, STARTIO is called to read the VTOC, and alternate track information is saved (SAM and VAM 1). If the format is VAM 1, BLDMAP is entered through the VAM entry point. If this is a VAM 2 dump, control is passed to VAM2DUMP. Otherwise, return is made through register 5 to the calling routine.

BLDMAP Routine

BLDMAP is entered through the VAM routine which verifies that the request is for a complete dump. If this is not the case, VAM immediately returns control via register 5.

When BLDMAP is entered, all registers are stored. STARTIO is then entered with the device address and CCW list to read the volume label. STARTIO is again entered to write the control record.

On return from STARTIO, all VTOC and DSCB extents are analyzed in order to set all pages in use on the from-device as unavailable. A bit map is constructed showing all pages initially available except:

- Cylinder 0 track 0 -- used for volume and user labels.
- Cylinder 0 track 1 -- used for IPL text.
- Cylinder 199 -- used for error retry.
- Entire extent of VTOC.
- Areas used by other data sets.

When completed, the bit map shows all available pages on the to-device. This map is maintained in core throughout the entire dump or restore operation.

BLDMAP uses the following subroutines:

- READDSCB -- sets all pages within DSCBs as unavailable.
- INIT2314 -- converts the head number (HH) from the VTOC extents to the proper 2314 page number.

Upon completion, exit is made to the calling routine via register 5.

BUILDCCW Routine

BUILDCCW is entered to build a list of read count CCWs to read the maximum number of records on the device. The routine loads registers 7 and 8 with the dummy read count CCW, places the address of the first CCW in registers 1 and 4, and sets the

record address to 0. Register 4 is updated to the address of the last possible CCW for one track of the device. CCWs are stored in a loop until the content of register 1 (starting address) equals the content of register 4 (stopping address). Control is passed to MODTKADF.

MODTKADF Routine

MODTKADF receives control from BUILDCCW to establish necessary addresses to read the count fields from the specified track.

CHANGCYL is called to increment the CCHH to the current track. STARTIO is called to execute the read count CCWs constructed by BUILDCCW.

ANALSENS Routine

ANALSENS tests the previous read count. If the read was invalid, control is passed to CKALT; otherwise, control is passed to CNGRDCKD.

CNGRDCKD Routine

CNGRDCKD is entered from ANALSENS to change the read count CCWs. The last used read count CCW is determined and compared with the first. If only record 0 exists, control is passed to PREPRWRT. Otherwise, the counts are used to specify the data length and the address fields in the read CKD CCWs built over the used read count CCWs. Data address fields are forced to doubleword boundaries. Control is passed to PREPRWRT.

PREPRWRT Routine

PREPRWRT calculates the length of the entire track image to be read and the CCW list to be used, and inserts these in the length fields of the tape-write CCWs. STARTIO is called to execute the read CKD CCWs constructed by CNGRDCKD. Control is passed to TSTWRTSP.

TSTWRTSP Routine

TSTWRTSP is entered at the TSTWRTSP entry point to change DASD CCWs to specify a write operation. The list of read CCWs is obtained and scanned. The data addresses in the duplicate part of this CCW list are changed to the relative addresses contained in an area called DTABUFCK. If a write special count, key, and data is necessary due to overflow, the last write CCW is changed to that format. Instead of write count, key, and data, the last CCW in the original chain becomes a write-record zero. On exit, control is passed to the specified write routine through register 5.

DMPDASD Routine

DMPDASD is entered to write the control field and CCW list from a direct-access device to a tape device. This routine simply places the proper parameters into registers 2 and 3, then links to STARTIO with the return address of DMPDTA in register 9.

DMPDTA Routine

DMPDTA receives control from STARTIO following a DMPDASD operation. This routine prepares the parameters to write the data field on tape. The count is incremented when noise is encountered. Control is passed to STARTIO with the address of BUILDCCW in the return register (register 9).

PUTMAP Routine

PUTMAP is entered to print the bit map on tape. The device is determined (2311 or 2314), and the proper bit map is indicated. Control is passed to STARTIO with the address of BUILDCCW in the return register (register 9).

RSTRTAPE Routine

RSTRTAPE is entered to read the control record (CCW list) and following data record (track image) to restore data from a tape device. The control field is first read, using STARTIO, and a switch is set to indicate whether a restore from tape, or a dump from disk is requested.

On return from STARTIO, the write-check switch (WRCHEKSW) is tested. If off, control is passed to CHTOSKP and returned to RSTRTAP2. If the switch is on, the instruction sequence continues at RSTRTAP2.

RSTRTAP2 prepares the parameter registers (2 and 3) to read the data field. Control is passed to STARTIO with a return indicated to MODTKADT.

MODTKADT Routine

MODTKADT increments the track address after the RSTRTAPE read operation to test for end-of-job (EOJ). At EOJ, control is passed to EOJ1. Otherwise, control is passed to CNGCYLT with a specified return to WRTDISK.

WRTDISK Routine

WRTDISK tests AMSW and, for SAM, gives control to WRTDISKY. OVFLPGSW is tested to determine whether a page overflowing from the previous track is being relocated. If so, PRNPG2 is called to finish the relocation. The relocated list is searched to

determine whether any relocations have been made to this track. If not, control goes to WRTDISKX for housekeeping, and then to WRTDISKY. If the track has received relocations, each record is tested, and for those which are safe, the data field, only, is written in place. At completion, control goes to CKALT1B which continues the job at the next track.

WRTDISKY Routine

WRTDISKY establishes the parameters used to write to a disk device. If the write-check switch is off, control is passed to CHTOSKP, with a return to WRTDISK1 in the WRTDISK program. Otherwise, the instruction sequence continues at the WRTDISK1 return point. STARTIO is entered to write to the disk device; the return register points to the CKALT1 routine.

CHTOSKP Routine

This routine computes the length of both primary and duplicate CCW lists; it also sets the skip bit on in all CCWs in the duplicate list. At completion, return is made through register 9.

WRTCHECK Routine

WRTCHECK is entered with a branch-and-link if write-checking is requested following a write operation to a direct-access device. On entry, registers are saved; register 9 contains the return address.

This routine compares the contents of two buffer areas:

- The buffer from which a track was written.
- The buffer into which that track was read, immediately following the write operation.

If the two buffers contain equal contents, control is passed to WRTDISK. Otherwise, the compare is retried a maximum of ten times. If the equal compare is not found, the job is terminated with an error message; the wait state is entered.

CHANGCYL Routine

CHANGCYL is entered by a BAL statement with the return address in register 9. Register 15 contains the current track address in binary, and is incremented by one. For the last track in the cylinder, the address is updated to reflect track zero of the next cylinder. Return is made using register 9.

CKALT Routine

CKALT checks for a bad track on a SAM format read (DUMP) device, places the alternate track address in CCHH form, and returns for another read try.

CKALT is entered by a branch statement to the CKALT entry point. This routine immediately tests for VAM 1 format. VAM 1 format causes a branch to the BADPAGE routine at RDBDPG. Otherwise, the bad track switch is interrogated. If the switch is off, indicating no bad track, the job is terminated with the proper error message. If the track is bad, normal instruction sequence is followed.

The home address and record 0 are read, using STARTIO, and it is determined whether the track is an alternate. If so, control is passed to CNGRDCKD to build CCWs for a read count, key, and data. Otherwise, an alternate track is found and CNGRDCKD is entered as before.

CKALT1 Routine

CKALT1 checks for a bad track on a SAM format write (RESTORE). The routine is entered at the CKALT1 entry point supplied to the calling routine in register 9.

CKALT1 first determines whether there is a bad track condition. If so, control goes to GETCCW. Otherwise, switches are reinitialized, WRTCHECK is called if write-checking is requested, and the appropriate exit is made.

GETCCW Routine

GETCCW is entered by a branch statement from CKALT1 when a bad track condition occurs. On entry, the format is tested. If VAM 1, control is passed to BADPAGE. On the first pass, the home address and record 0 of the bad track are read. Messages identifying the bad track and the alternate tracks are written. The alternate track is written using a seek CCW, with the alternate CCHH at the data address, chained to the search CCW of the original list. Return from STARTIO is specified as CKALT1, indicating that the alternate has been written. CKALT1A returns control to the mainline program.

On exit, STARTIO is given control with a return indicated, in register 9, to CKALT1.

BADPAGE Routine

BADPAGE is entered to attempt to recover an error if a bad track is found in a VAM 1 read or write. BADPAGE is entered at RDBDPG for a read, and BADPAGE for a write.

The track address is stored in DTABUFF, and device type is determined. A 2314 error is handled by the BDPG 2314 routine. BITMAP is interrogated to see whether the defective page is available. In a write error on an available page, the page is made unavailable and control is passed to TRYAGAIN to determine whether there are more pages on the track to be processed. No data has been lost. For a read error in an available page, DASDI (a Dump/Restore routine within CEBDR, not CEBDI) is called to supply a dummy track which is written to preserve VAM 1 format. Normal instruction sequence is followed.

If the page is unavailable, it is tested to see if it is part of the VTOC. If it is, the job is terminated due to loss of control data. For a data page, control goes to RLOCAT to relocate the page in case of write error. On a read, RLOCAT immediately gives control to VTOCOK and subsequent BADPAGE routines which advise the operator that the page will be lost and give him an abort-or-continue option. If the job continues, data on the page is lost and DASDI is given control.

Routines used by BADPAGE:

PGAVAIL -- determines availability of the page

VTOCCK -- terminates when VTOC is lost

TRYAGAIN (and AGAIN14) -- checks whether another page is affected by the bad track

VTOCOK -- counts lost page and saves its number

PRTMSG -- writes messages to operator about lost page(s) and goes into wait state

COMEBACK -- fields external interrupt when operator signals that job is to continue.

RLOCAT

RLOCAT relocates a data page when a bad track on the to-device is encountered.

SRCHMAP searches BITMAP to find an available page to receive the relocation, and stores the old and new page numbers in PGLST. If no page is found, or if no space remains on PGLST, VTOCOK is given control.

RBLDPG determines the position on the track of the page and reconstructs the page in its entirety in PAGBF. If it overflows to the next track, PREPOVF sets OVFLPGSW as a signal to WRTDISK that the relocation should be completed before the next track is written. RDPTI obtains the first part of a record overflowing to the bad track.

BLDNWPG retrieves the new location of the page from PGLST and converts it to CCHHR format. If the relocation is to an unrestored track, the track and record are recorded in RLOCL. If no other relocations have yet been made to the track, CALDASDI is called to format the track to assure VAM format. After the above, and if the relocation is to an already restored track, write data CCWs are constructed and executed to put the record on the disk. If the relocation is to two tracks (always for 2311, usually for 2314), a similar procedure is executed for the second track.

ENDREEL Routine

ENDREEL is entered upon encountering a unit exception on a multiple tape system. This routine is entered by a branch instruction to the ENDREEL entry point.

ENDREEL verifies that the tape is finished. If finished, the data buffer is written; STARTIO is entered with a return specified, in register 9, to ENDREEL. Otherwise, the routine writes the data buffer record, establishes switches to determine whether all tapes have been restored, and passes control to EOJ1.

EOJ1 Routine

EOJ1 is entered at end-of-volume on tape. This routine writes trailer labels on the current to-volume, and passes control to TOTAPE to insure that the new volume is mounted.

EOJA Routine

EOJA enters ENDOFJOB to print the end of job message. STARTIO is then given control to rewind the tape. On return, a branch is made, through register 5, to the proper wait routine.

ENDOFJOB Routine

ENDOFJOB prints end-of-job, or end of reel/disk pack. The routine then: stores beginning, end, and relative beginning addresses; sets the read protect switch, and determines the device type. If tape is used, it is rewound by EOJA. Otherwise, exit is made through register 5 to the proper routine.

EOJAA Routine

EOJAA is given control at the conclusion of an entire 2311-2311 or 2314-2314 dump. If the disk is in SAM format, EOJAA updates field 6 of the VTOC DSCB to reflect any alternate track assignments necessitated dur-

ing the dump. The CPU enters a wait state, and the job terminates. If the disk is in VAM 1 format, control is given to DADSM.

EOJAA is also given control after a restore. For a SAM-formatted disk, it updates field 6 of the VTOC DSCB to reflect any alternate track assignments necessitated during the entire restore. For a VAM 1-formatted disk, control is given to DADSM. No update feature is provided for either format after a partial restore.

DADSM Routine

The DADSM DSCBs are read into SAVEVTOC from the to-device. The extents are overlaid with the proper extents as shown in the bit map.

If more DADSM DSCBs are needed than were on the from-device, an unused DSCB is found in the VTOC and this new DSCB overlays the dummy DSCB and is chained to the previous DADSM DSCB.

When all extents are placed in the DADSM DSCBs, any unused extent fields are cleared and the DSCBs are put back on the to-device. After the DSCBs are written on the to-device, control is passed to RBLDDSCB.

RBLDDSCB Routine

If pages have been relocated in a full VAM 1 dump or restore, the format-A and -B DSCBs must be updated to reflect the relocations. The extents of each data set are searched for relocated pages. The format-A DSCB is searched, then chained format-B DSCBs are searched, to the end of the chain. If relocated pages are found, the extents are updated, with any new ones placed at the end of the chain. If more DSCBs are needed, they are obtained and chained.

Changed DSCBs are written out when completed. When the last relocated page is found, the end of job message is printed and the job is terminated.

TAPEEOF Routine

TAPEEOF routes EOF exits at the end of a tape. At end-of-tape, in a dump program, control is passed to ENDREEL. In a restore program, the EOJ message is prepared with exit to EOJAA. If another tape remains, PREWAIT1 rewinds the tape and requests another tape. When the tape is mounted, the external interruption passes control to FIXEX which branches to the mainline program.

VAM2DUMP Routine

VAM2DUMP is entered from TOTAPE for a VAM 2 dump to tape; from CKVOLLBL for a VAM 2 dump to disk; and from FRMTAPE for a VAM 2 restore from tape. Upon entry, it is determined whether this is the first tape in a tape operation. If not, control is passed to the proper Dump/Restore routine. Control is passed to VAM2REST for a restore operation. When a dump to tape is requested, the control record is written, the from-device PAT is read, and control is passed to CLRERPG to resolve relocation entries and remove error pages. For a disk to disk dump, CLRERPG reads the to-device PAT to determine those pages in error. At this point, in a dump to tape, the PAT is written. The IPL, volume label, user labels, and IPLTXT records are read in and written out for a dump to tape or disk. Data and DSCB pages are also read in and written out. If a page is marked as an error page on the from-device, and has a relocation entry, the relocated page is written in the original page position. For an error page on the to-device (on a disk to disk dump), or if a track condition check is detected on a write to disk, control is passed to RELOCATE to relocate the page and to build a relocation entry. When all pages are dumped, control is given to the ENDOFPAT routine.

CNVRELPG Routine

CNVRELPG is entered by a branch-and-link instruction, with return indicated via REG5. Upon entry, REG2 contains the address of the appropriate UCB which is used to determine the device type. REG4 contains the relative page number. This routine computes the cylinder, track, and record number, from a relative page number, for VAM 2 formatted direct access devices. The cylinder, track, and record number are stored at CCHH.

CLRERPG Routine

CLRERPG is entered from VAM2REST. For a dump operation, this routine resolves relocation entries and removes error pages, in the PAT. On a disk to disk dump, or on a restore, the to-device PAT is read to determine those pages in error. Error pages not in use on the from-device PAT are marked as error pages; those in use are listed in ERPGLST, which is subsequently used by CHKERR. If the operation is a 2311 to 2314 restore or dump, the PAT is expanded to two pages. On exit, control is returned to the appropriate Dump/Restore routine.

VAM2REST Routine

VAM2REST is entered, during a restore operation, by a branch instruction from VAM2DUMP. The routine restores a tape which was dumped in VAM 2 format to a direct access device. On entry, the PAT is read from tape. Control is then passed to CLRERPG, which reads the to-device PAT to find error pages. The IPL, label, and IPLTXT records are read from the tape, and written to the disk. Pages are then read from tape to disk. If there is an error page on the to-device, or if a track condition check is detected, control is passed to RELOCATE to relocate the page, build a relocation entry, and update the relocation control entry. When all pages are restored, control is passed to PATWRITE to write the updated PAT.

RELOCATE Routine

This routine, used to relocate pages on a VAM 2 formatted direct access device, is entered: if a page to be written to a disk is marked as an error page in the PAT; if a track condition check occurs while attempting to write to a disk; or, if the PAT has been expanded by going from a 2311 to a 2314. If the write operation is for an IPL, label, or IPLTXT record, the job is terminated, and a message is issued. When an error occurs on a page record, RELOCATE searches the PAT for an available page, writes the original page to that page, builds a relocation entry, and updates the relocation control entry. If there are 96 previous relocation entries, or if there are no available pages, the operator is given the choice of aborting the job, or continuing with that page lost. For a PAT page, no relocation entry is built, but the volume label pointer to the PAT is updated. For a 2314 PAT page, both PAT pages are relocated. If there are no available pages for a PAT relocation, a message is issued and the job is terminated.

V2UNEXCP Routine

This routine is entered upon receiving a unit exception from a VAM 2 tape operation. On a dump, it writes a tape mark, a multi-reel control record, and another tape mark. It then prints the mount tape message and returns to the dump routine after receiving an external interruption. On a restore, the routine reads the control record to see if this is end-of-job. If so, control is passed to PATWRITE. Otherwise, V2UNEXCP prints the mount tape message and returns to the restore routine after receiving an external interruption.

V2WRTCHK Routine

V2WRTCHK is entered immediately after execution of a VAM 2 write operation to a driect access device, if write checking is requested. This routine compares the contents of the buffer from which a record was written, with the contents of a buffer into which the same record was immediately read back after the write. If the contents do not match, the operation is re-executed and the buffers again compared. If the buffer contents do not match after ten attempts, the program is terminated, and a message issued.

VAM 2 EOF Routines

ENDOFPAT - This routine is entered on a dump to tape, when all used pages have been dumped. ENDOFPAT writes a tape mark, an end-of-job control record for a subsequent restore, and another tape mark. It rewinds the tape, prints an end-of-job message, and terminates.

PATWRITE - This routine is entered on a disk to disk dump when all used pages have been dumped, PATWRITE is also entered on a restore, when a control record is read indicating that all pages have been restored. PATWRITE writes out the new PAT, rewinds any tapes used in the operation, prints the end-of-job message, and terminates.

DASDDR I/O SUPERVISOR ROUTINES

The DASDDR I/O supervisory routines provide these functions:

1. Initialization and termination.
2. I/O processing: control, interruption processing, error handling.
3. Message generation.

SYSIN Routine

SYSIN prepares a read CCW using the buffer passed in REG2, and calls STARTIO. Return is through REG15.

STARTIO Routine

This routine controls all I/O operations performed by the DASDDR program. It is entered at STARTIO, via REG9. REG2 contains the address of the appropriate UCB. REG3 contains the address of the CCW. The device address is loaded into REG4 from the UCB. All pending interruptions are serviced at this point. An SIO instruction is issued until the command is accepted, at which time the TIO instruction is issued repeatedly until the unit is not busy. If

the unit is unavailable, the wait state is entered, and the program is terminated. Control is then given to the I/O interruption analysis routine at location CKCSW.

CKCSW Routine

All I/O interruptions are serviced by this routine. REG2 contains the address of the appropriate UCB for the device which caused the interruption. This routine examines the nature of the I/O interruptions and classifies them into any one of six types of interruptions. The type of interruption is determined by scanning the status portion of the CSW.

MESSAGE ROUTINES

Message Output (PRNT) Routine

PRNT is entered at SYSOUT. This routine writes messages using the message output device defined by UCBSYO. (The MSG statement stored the device in UCBSYO.) Register GR2 contains the message address. The appropriate device CCW containing the address of the message is constructed. Pointers to the UCBSYO and SYOCCW, respectively, are stored in registers REG2 and REG3. These parameters are then passed to the STARTIO routine where the messages are printed. On return, the PRNT returns to the calling routine.

Within the I/O package, SYSOUT is always called by the PRNT routine. This routine, called by a BAL REG9,PRNT instruction, enters SYSOUT with the address of the message print buffer in REG2. SYSOUT returns control to the PRNT routine via REG15. PRNT, in turn, returns to the calling program through REG9.

I/O Error Print (IOPRNT) Routine

When there is a system error or a situation where intervention is required, information from the UCB plus the command code from the last I/O instruction must be printed out together with an error message. In these cases, IOPRNT is called. IOPRNT extracts the necessary information, unpacks and translates it, and moves it into the input buffer, and then goes to PRNT.

Write-to-Operator (WTO) Routine

This routine writes messages to the operator only if the input device has been defined as a typewriter (UCBOPR). REG1 contains the length of the message; REG2, the address of the message. The OPRCCW is filled in accordingly. Control passes to STARTIO with pointers to UCBOPR and OPRCCW where a message will be printed on the typewriter. Control returns to the calling routine through register 15.

Error Messages

Error messages are listed in System Messages.

Diagnostic Messages

Diagnostic messages appear in the following format:

number (16-byte text) cuuxxssssyyyy

where c is the channel of the device in error, uu is the unit, xx is the command code, ssss are the status bytes from the channel status word, and yyyy are the sense bytes.

Message texts are listed in System Messages.

Chart BB. DUMP-RESTORE Routine (Part 1 of 5)

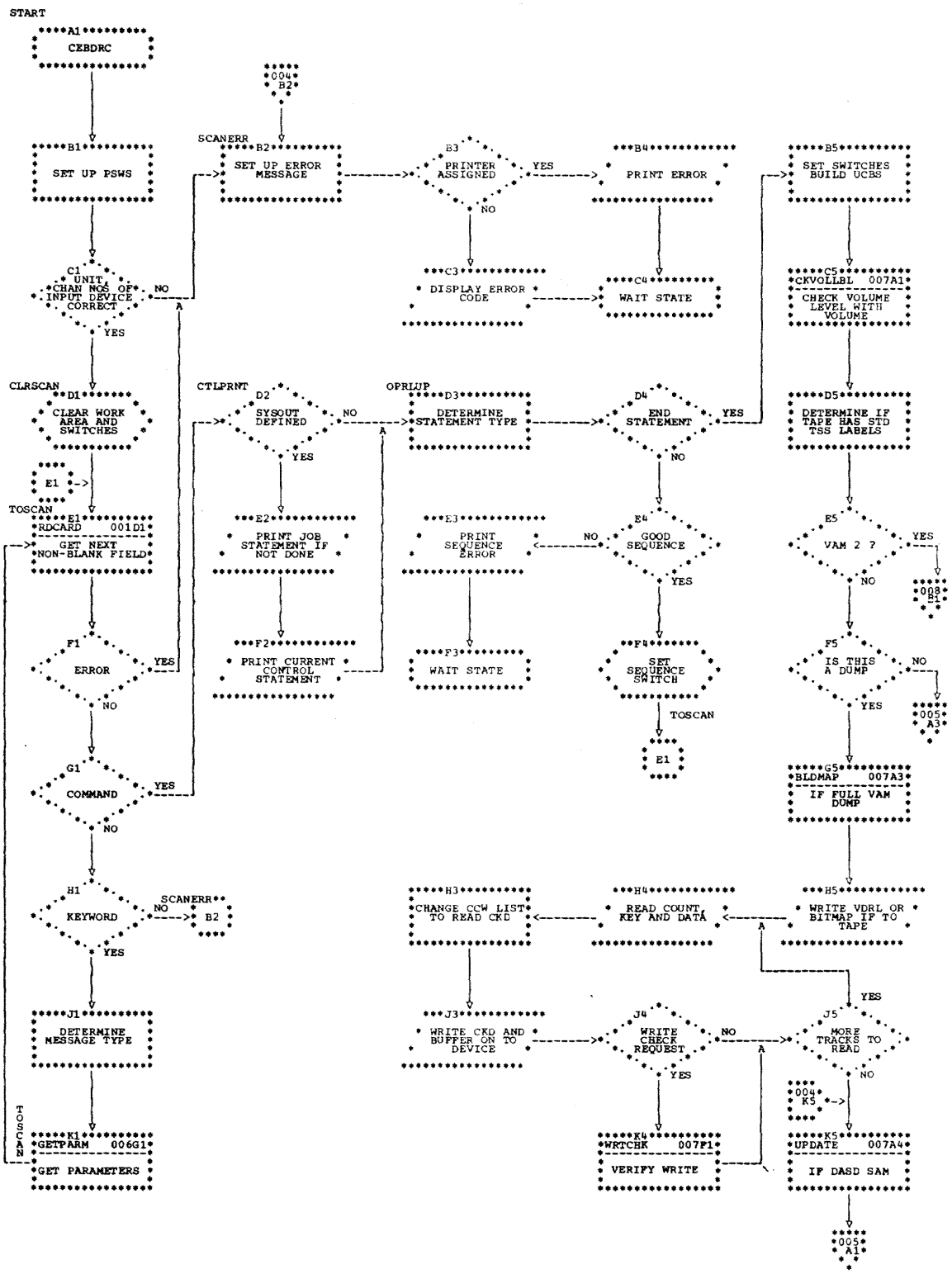


Chart BB. DUMP-RESTORE Routine (Part 2 of 5)

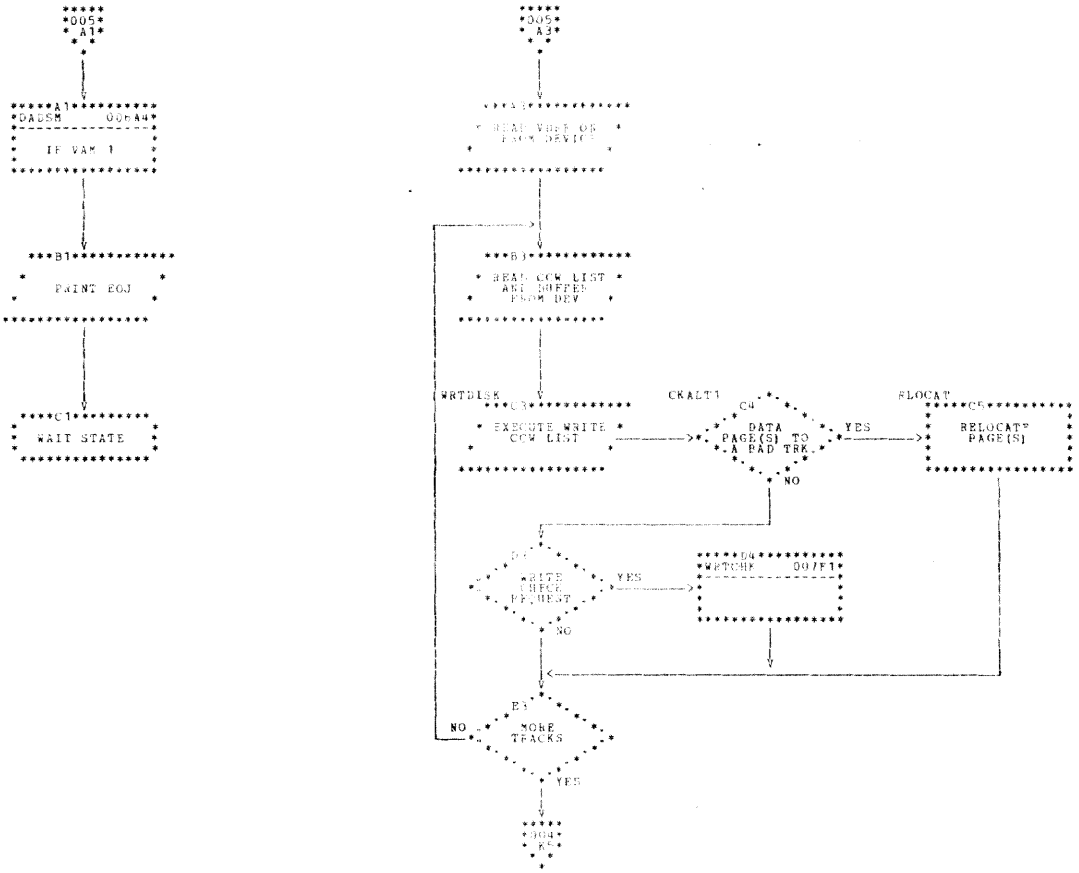


Chart BB. DUMP-RESTORE Routine (Part 3 of 5)

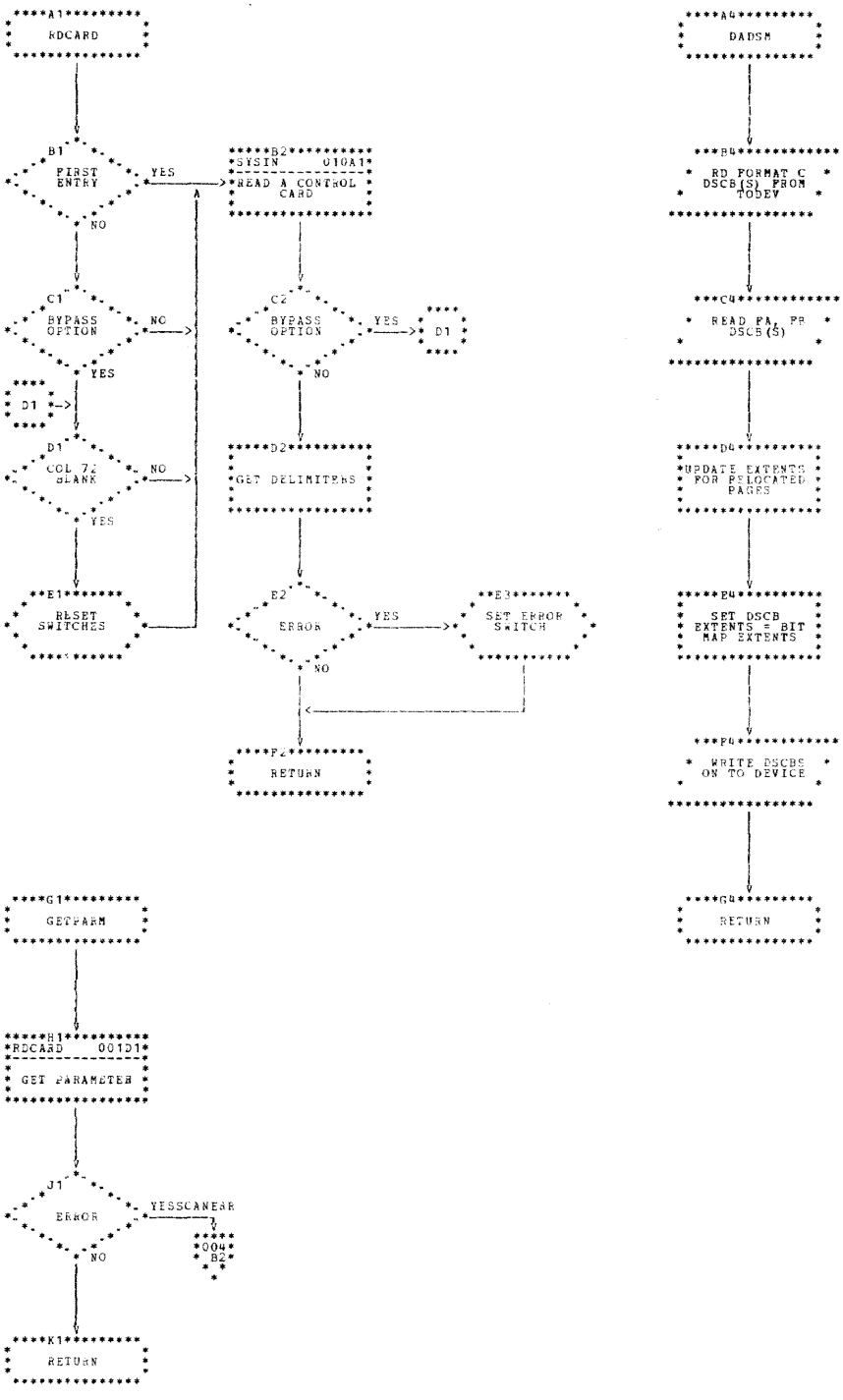


Chart BB. DUMP-RESTORE Routine (Part 4 of 5)

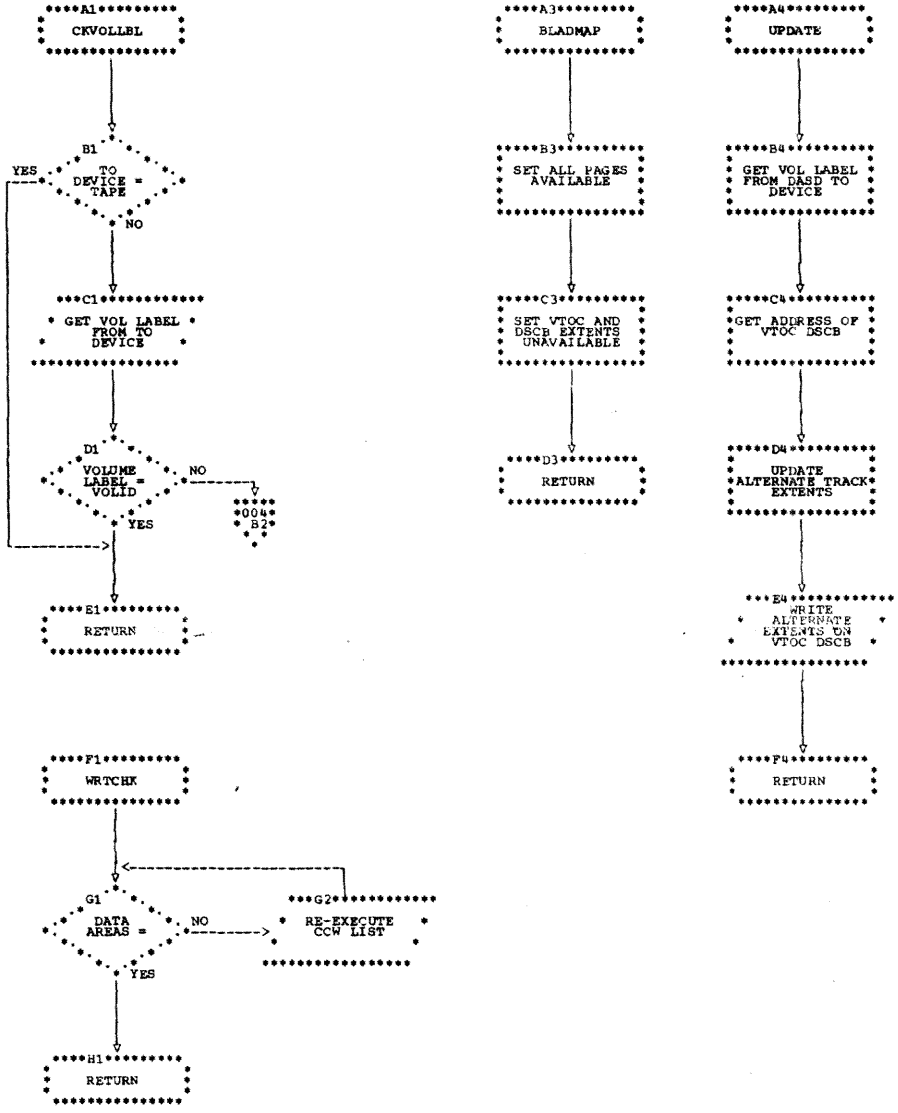
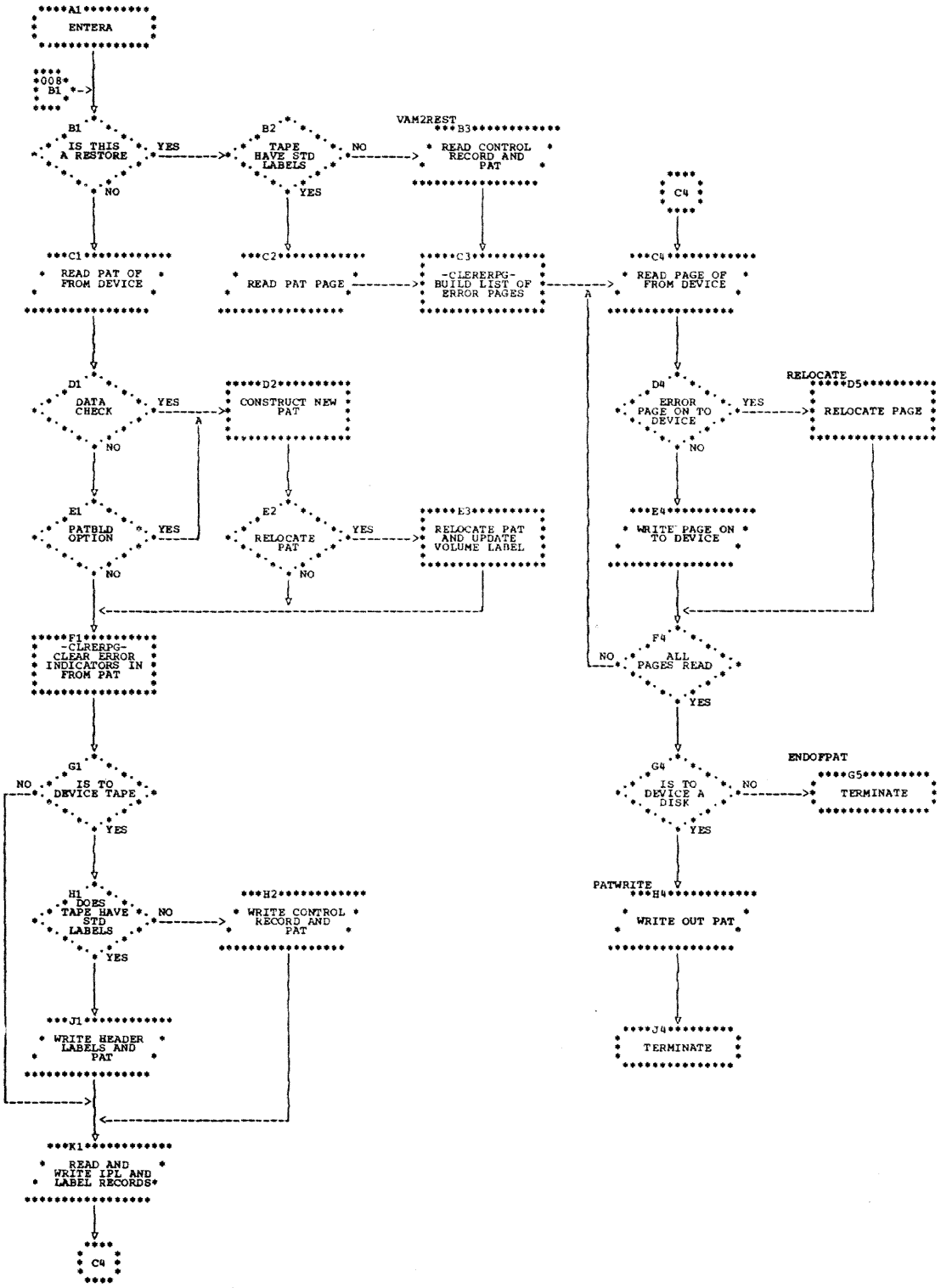


Chart BB. DUMP-RESTORE Routine (Part 5 of 5)



The direct-access print (DADUMP) program, module CEBDP, prints the contents of direct-access devices.

ATTRIBUTES: DADUMP operates outside and in support of TSS/360 as a privileged, disabled, non-relocatable, serially reusable program.

RESTRICTIONS: None.

HARDWARE CONFIGURATION REQUIREMENTS: The DADUMP program runs on a System/360 Model 67 in standard PSW mode, or on any System/360 with sufficient main storage and required device configuration:

Main Storage: 34,000 bytes (2314) 47,200 bytes (2301)

Program Input Device: 2540, 2400 (9 track)

Control Statement/Input Device: 2540, 2400, 1052

Message Output Device: 1052, 1403

Other: 2301, 2311, 2314

PROGRAM OPERATION

The DADUMP program first determines the control statement input device, and reads and analyzes control statements. The job analysis routine determines from the BEGIN statement the starting track address, converts it to BBCCHH and puts it in TRACKNO to be used by the direct-access device input to seek the proper track.

The program then reads a track from a direct-access device through its input routine (VRECOVR) and puts it into a buffer. It builds an address table (RECTABLE) whose entries point to the records of the track. The output routine LISTTEST then uses VRECTAB to print the records of the track.

Finally, the program prints the contents of the track to the output device: The number of tracks to be printed out on any one extent is put into SAVREC by the job analysis routine to be used by the output routine. TRACKSAV must have the beginning track in extended binary form as a decimal track number.

The program determines the number of records in the track by seeking the track and setting the file mask to prohibit leaving the track. The channel program continues with a search home address followed by a string of read count commands with the multiple track bit on. Upon reading the count of the last record, the hardware attempts to switch tracks, which causes a file mask violation. The interruption enables the program to (1) compute the number of records in the track and (2), if the last record was an overflow record, to set the overflow-incomplete flag in the sense data to on. The key length and data length for each record are also available from the read count commands.

The program then seeks the track and reads the home address, record zero, the count field, key field and data fields of each record. The information is converted to a printable format and printed. The program terminates and enters the wait state when the upper track limit is reached. If there are more tracks, the program proceeds similarly for the next track.

DADUMP ROUTINES

The DADUMP program accomplishes its function through the following routines, summarized in Table 8 and Chart CC.

INITIALIZATION ROUTINES

Start Routine

Control is given to location START when the utility is loaded into core. Base registers are set up, registers are cleared, interrupt PSWs are set up and program switches are reset. The program is put in a wait state by means of an LPSW. The operator then defines the input device either through the console or typewriter. The proper input routine is given control depending on device type.

Console Routine

CONSOLE assumes control via an external interruption, generated by pressing the interruption key, if the operator defines the input device through the console. CONSOLE then checks the code to verify that the input device is 2540, 2400, or 1052 and that the channel number is not greater than six. If these conditions are met, the UCBIN is filled in with the address of the input device, and control is given to the control card analysis (CLRSCAN) routine. If an error is detected in the coded infor-

Table 8. DADUMP Routines

Type	Name	Function
Initialization Routines	START	Sets up base registers, clears register, sets up interrupt PSW, and resets switches.
	CONSOLE	Checks operator-entered code to verify that the input device is valid.
	ATTN	Prints message to operator requesting input device definition.
Job Statement Analysis Routines	TYPINPT	Tests input device. Enter STARTIO to read one record.
	CLRSCAN	Performs control statement analysis, and housekeeping functions on program switches and buffer areas.
	SCANCARD	Scans control cards and identifies the fields.
	MSGPRINT	Prints a message to the device specified by MSG statement.
Processing Routines	VRECOVR	Pseudo-reads a track from a direct-access device; notes overflow and all errors found.
	LISTTEST	Controls the reading and printing of specified tracks of a direct-access device.
	PRNTDASD	Controls the printing of the contents of a track of a direct-access device.
	FORMAT	Formats a line of output (132 bytes) according to parameters given.
	HEADER	Formats the heading at the beginning of each new page.
Job Fulfillment Routine	PRINT	Writes a line of output to the output device, which can either be a printer or typewriter.
I/O Routines	I/O Control	Controls all I/O operations performed by the utility program.
	I/O Interruption Analysis	Examines and classifies I/O interrupts.
Message Routine	IOPRNT	Prints error messages, saves channel-and-unit, addresses, CCW status, and two sense bytes.

mation, an error message is displayed in the console lights by means of an LPSW.

ATTN Routine

ATTN assumes control via an I/O interruption generated if the input device is defined through a typewriter. On entry, the system mask is set to enable interruptions.

ATTN constructs the UCBMSG and enters the write-to-operator (WTO) routine at MSGPRINT. MSGPRINT instructs the operator to enter the input device type. On return from WTO, ATTN reads the input device, converting lower case characters to upper case if necessary. Parameters are then passed to DEVLUP to scan FRMTABLE to test for a valid device. ATTN routine fills in the UCBIN with the valid device address and passes control to CLRSCAN.

When an invalid input device is specified, an error message is displayed in the instruction counter portion of the console lights.

TYPEINPT Routine

On entry, general registers are saved and the control statement input device is

tested. If tape or card reader, the system mask is set to enable interruptions. The routine passes control to STARTIO to read one record. Control is returned to the calling routine.

If the typewriter is defined as the input device, the information pertaining to one extent is read; control is passed to RECOVER to read the specified track.

JOB STATEMENT ANALYSIS ROUTINES

CLRSCAN Routine

CLRSCAN performs control statement analysis. It first does housekeeping functions on program switches and buffer areas required by the program. The routine then links to the control card scan routine at SCANCARD. If SCANCARD indicates an error when it returns control, an error message is printed.

If the field is flagged as a command by SCANCARD, the routine searches CMDTABLE for a matching valid command, and goes to the specified routine to process it. Keywords are processed in the same way using KEYTABLE. If the keyword is valid, the routine that processes that keyword calls SCANCARD to obtain the parameter and checks for validity. Parameters specifying I/O devices are checked for validity by FRMTABLE.

When the control statements are processed, they are checked for correct sequence, printed out, and switches are set. When the END statement is encountered, all outstanding requests made by the control statements are honored.

Each PRINT statement is honored as it is encountered. After the PRINT statement is honored, the CLRSCAN routine is reentered, and the next control statement is read.

Error Conditions: Command or keyword field returned by SCANCARD is too long or cannot be found in the appropriate table.

An invalid parameter is given following a keyword.

A required parameter is not present when the END card is encountered and processed.

SCANCARD Routine

SCANCARD scans a control card (and any continuation cards present) and identifies the fields. The card is read, by READCARDW, from the system input device into the routine's word area (SCANSW).

Any non-blank name field is ignored. If an error, command, keyword, parameter, or right parenthesis is encountered, a switch is set in the first byte of the work area. If the remaining portion of a statement is to be skipped, the bypass switch is set in the same byte. When control returns to the calling routine, register 1 contains the address of the first byte of the field and register 2 the length of the field. SCANSW is laid out as follows:

byte	0	:	Scan switches
bytes	1-80	:	Input buffer
byte	81	:	Entry control switches
byte	82	:	Current delimiter switch
byte	83	:	Previous delimited switch
byte	127	:	Exit control switches
bytes	128-129	:	Parenthesis counter
bytes	136-139	:	Save area for pointer to next column to be scanned
bytes	144-191	:	Save area for registers 3 through 14

The individual bits of the switches are described in Table 9.

When control exits from SCANCARD, the partial parameter and command control switches are set to "off".

Error Conditions:

- No command before column 71 of a card.
- First non-blank field after the name field not terminated by a blank.
- Unpaired parentheses.
- Partial parameter field not terminated by the end of the card it is continued on.
- Keyword with a length of zero bytes.

MSGPRINT Routine

MSGPRINT is entered at the MSGPRINT entry point to write a line to the message device. On entry, registers are saved. At this point the message UCB (UCBMSG) is interrogated for a defined write-to-operator (WTO) device. If WTO is not defined, an error wait is entered. Otherwise, the count and address are stored in the CCW and STARTIO is entered to issue a message to the operator. On exit, registers are restored; control is returned to the user via register 9.

Table 9. SCANSW Bits

Bit	Scan Switch	Entry Control Switch	Exit Control Switch	Delimiter Switches
0	scan error	not used	condition code 0	delimiter is an =
1	bypass	continue	condition code 1	delimiter is a ,
2	not used	parameter follow	delimiter in column 71	delimiter is a (
3	first entry if 0	partial parameter	not used	delimiter is a)
4	command found	not used	column 72	delimiter is a blank
5	keyword found	command control	column 72 non-blank	not used
6	parameter found	not used	parameter length greater than 0	not used
7	right parenthesis found	not used	parameter length equal to 0	not used

PROCESSING ROUTINES

VRECOVR Routine

VRECOVR is entered at either the VRECOVR or RECOVER entry points. On entry the CCW list is initialized as are switches and RECTABLE. VRECOVR then saves the program check PSW, and loads it with a return address. CLEARCOR is then entered to clear core storage, beginning at the location specified in register 9.

The CLEARCOR routine loops to clear core until a program check is generated. At this time the PSW, loaded earlier, gives control to the instruction following the branch to CLEARCOR.

On return, parameters are set and READ-DASD is entered to read the home address, record 0, and record 1 count. The track preceding the requested extent is tested for overflow. If overflow is indicated, OVRFLOSW contains X'10', otherwise, X'00'. If, at this point, a bad track is discovered, the alternate is checked.

ERROR CHECKS: All check bits in the status byte are considered ABEND conditions except unit check, unit exception, and incorrect length.

UNIT CHECKS: The following form conditions are common to all supported direct-access devices (2311, 2314, 2301). All sense byte error checks are considered ABEND conditions except those listed.

1. Intervention Required. A message is printed indicating intervention is required. The PSW is loaded with the wait bits on.

2. Data Check. If it occurs in the home address, in the key or data fields, the fact is only noted in RECTABLE. If found in the count field as well as being noted in RECTABLE, a data field the length of the rest of the track is read and printed immediately. Control is returned to the main part of the routine with the bad record skipped.

3. File Protection. This means that the end of the track has been reached. The routine in control after this error occurs is a housekeeping routine which completes the address table before control is passed to the format-print routine.

ERROR CHECKS: Overflow. This error check occurs in conjunction with the file protection error. The only difference in control is one instruction, which turns on an overflow switch. Control goes to the field protection error check routine. At the end of the file protection-overflow error check routine, two switches are set to indicate that this track is overflow.

For 2301 Only: The following two conditions are handled uniquely for 2301. All sense byte error checks are considered ABEND conditions except those listed.

MISSING ADDRESS MARKER: A missing address marker is handled as a data check in the count field.

TRACK CONDITION 0: When this condition is found, it is ignored; i.e., control is returned without indicating the error.

UNIT EXCEPTION: Occurs when READDATA finds no data. If no data was expected, READDATA is modified to read count. Control continues as usual. If data is expected, an error message is issued. RECTABLE is completed and LISTTEST is given control.

If unit exception occurred because of a defective track, the head is positioned on the alternate track, an appropriate message is printed, and control is returned to VRECOVR to read the new track.

Incorrect Length: This condition is ignored; i.e., control is returned without noting any error.

Extent Printer Routine (LISTTEST)

LISTTEST is entered after the first extent has been processed by VRECOVR. Control is immediately passed to PRNTDASD. When control returns from PRNTDASD, it passes to either one of the several points, as indicated by the decision table:

Last track on volume	N	N	N	N	Y	Y	Y
Extent finished	N	Y	Y	Y			
JCL from console			Y	N	N	Y	
End card processed		Y	N	N	N	N	Y
Branch to VRECOVR	X						
WAIT (operator)		X					
Branch to ANYMORE			X			X	
Branch to TOSCAN				X	X		

On return, LISTTEST tests the track to see if it is the last of the volume. If not, the track addressed is incremented and return is made to VRECOVR. Otherwise, the message "END OF VOLUME" is printed and exit is made to TESTINPT.

PRNTDASD Routine

PRNTDASD is entered to print the track for visual inspection. On entry, all general registers are saved. The routine then initializes record number and byte count; zeros the line count; and prints the heading using the PRINT routine.

VCOUNTY is entered to restore switches if any of the following have occurred:

- Bad record 0
- Bad count
- Missing address marker

The next record is read and checked for an error flag. If one occurs, an error message results. Otherwise, the record is checked for overflow. If overflow exists, a message is printed and FORMAT is entered. If no overflow occurs, FORMAT is entered but no message is issued.

On return from FORMAT, the records are processed using FORMAT when necessary. Switches and buffers are reset and another record is read.

On exit, registers are restored and control is returned via register 9.

FORMAT Routine

FORMAT is entered at the FORMAT entry point. This routine converts and prints a line of output (132 bytes).

On entry, FORMAT expects registers 3-5 to contain:

3 - number of bytes
4 - address of output buffer
5 - address of next record

Register 1 contains the return address.

If the record is a home address (FORMATSW=X'FF') it is unpacked. FORMAT loads the home address location and the next available position in the print buffer into registers 4 and 5 in the save area. Registers are restored and control returns through register 1.

When FORMATSW=X'0F', record zero count area is unpacked, but not printed. Registers are restored and control is returned via register 1.

If neither of the above cases apply, lines are converted until full. When a line becomes full, a test is made for record zero (FORMATSW=X'F0'). A data area is tested for identical lines; record zero is not. In any case, control passes to PRINT to print the line.

On return from PRINT, control is passed to the proper routine via register 9.

Heading Routine (HEADER)

HEADER is entered at the HEADER entry point with the return address specified in register 9. On entry, all general registers are saved (restored on exit).

HEADER determines device address (CUU) and device type. FRMTABLE is searched for a useable device address. Separate routines are used to place a disk or drum device type in the heading. MOVEBUF is entered to load the print buffer with the header address, track address, and proper page. Page numbers are updated in decimal and converted to EBCDIC, MOVEBUF passes control to the PRINT routine to print the header.

On return from PRINT, the overflow condition is tested. If overflow occurs, the proper message is issued. In any case, control returns to the caller through register 9.

The logic of the HEADER routine is as follows:

Beginning of track	N	Y	Y	Y
Beginning of extent	N	N	N	Y
Previous track was overflow		N	Y	
Set up header				X
Update header	X	X		
Eject and print line 1	X	X		X
Format, print line 2, Skip 1				
		X		X
Skip 2	X			
Start overflow record			X	

JOB FULFILLMENT ROUTINES

Print Routine (PRINT)

PRINT writes a line of output to the output device, which can be either a printer or typewriter.

If the output device is undefined, a message is issued and the job terminated. If the output device is the typewriter, the message is printed, and the job is terminated. If the output is on tape, the tape is positioned, the line written, and control returned.

If the eject switch is on, a page is ejected, the line count is set to zero, and the eject switch turned off. Registers are restored and an unconditional branch taken to HEADER.

If the eject switch is off, the line is printed, and the line count is incremented and checked. If it indicates a full page, PRINT proceeds as if the eject switch were on. Otherwise, control is returned. The buffer is blanked after each line of output.

DADUMP I/O SUPERVISORY ROUTINES

The DADUMP I/O supervisory routines perform these functions:

1. Initialization and termination.
2. I/O processing (control, interruption processing, error handling).
3. Message generation.

STARTIO Routine

This routine controls all I/O operations performed by the utility program. It is entered at STARTIO, always by way of LINK9. Register 2 (UCBREG) contains the address of the appropriate UCB. Register 3 contains the address of the CCW. The device address is loaded into register CSR4 from the UCB whose address is in register UCBREG. All pending interruptions are serviced at this point with the use of an SSM instruction and the proper interrupt returns.

When all pending I/O interruptions are serviced, an SIO instruction is issued until the command is accepted, at which time the TIO instruction is issued repeatedly until the unit is not busy. If the unit is unavailable, the wait state is entered, and the program is terminated. Control is then given to the I/O Interruption Analysis routine at location CKCSW.

CKCSW Routine

All I/O interruptions cause control to be given to this routine at location CKCSW. Register UCBREG contains the address of the appropriate UCB, the device which caused the interrupt. This routine examines the nature of the I/O interruptions and classifies them according to the six types of interruptions. The type of interruption is determined by scanning the status portion of the CSW.

MESSAGE ROUTINES

IOPRINT Routine

IOPRINT is entered when there is a system error or a situation where intervention is required. Information from the UCB, and the command code from the last I/O instruction, is printed out in addition to an error message. IOPRINT saves the channel and unit addresses, CCW status, and two of the sense bytes. These are printed with the error messages by MSGPRINT.

Error Messages

Error messages are listed in System Messages.

Diagnostic Messages

Diagnostic messages appear in the following format:

number (16-byte text) cuuxxssssyyyy

where c is the channel of the device in error, uu is the unit. xx is the command code, ssss are the status bytes from the channel status word, and yyyy are the sense bytes.

Message texts are listed in System Messages.

Chart CC. DADUMP Routine (Part 1 of 6)

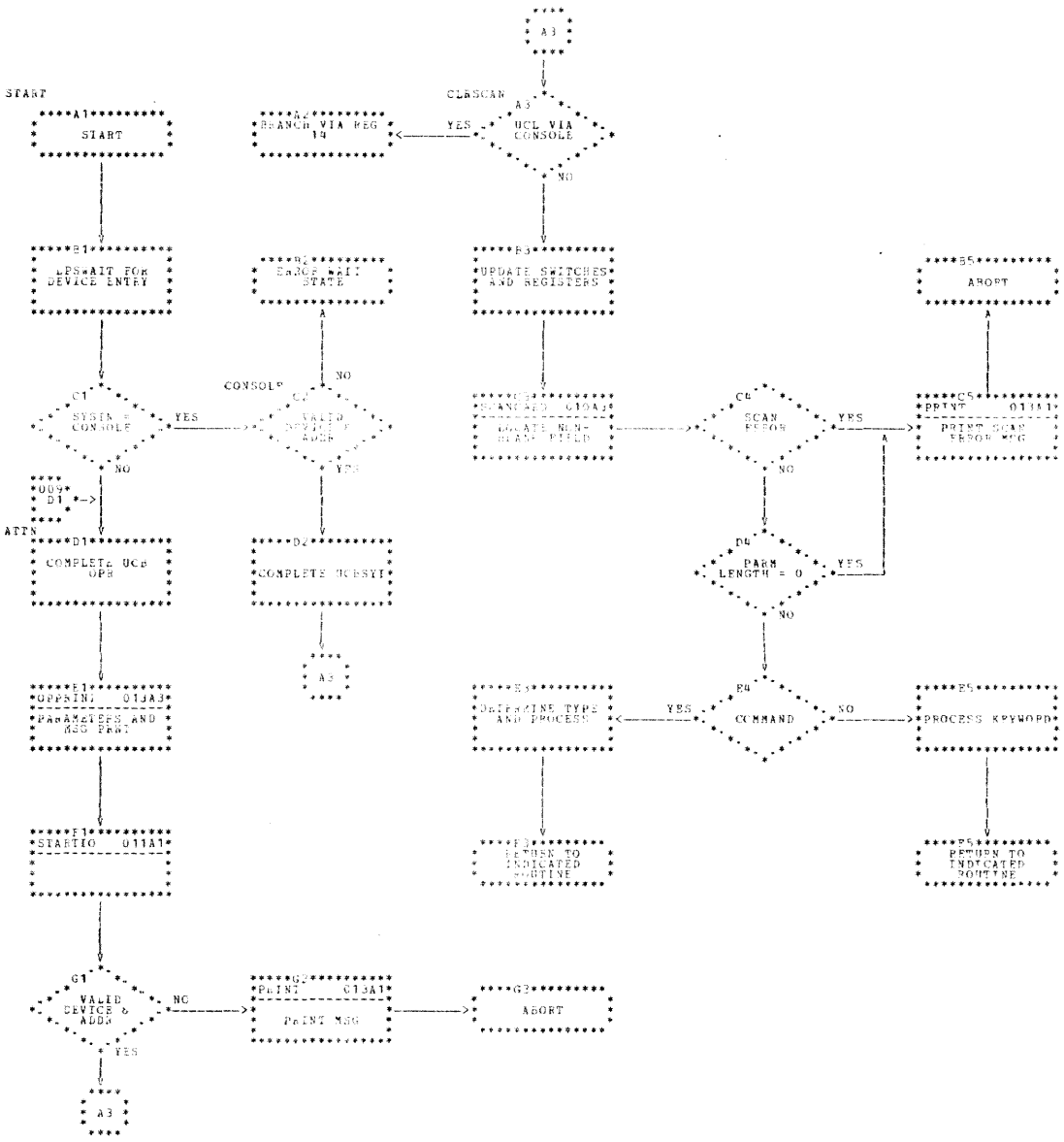


Chart CC. DADUMP Routine (Part 2 of 6)

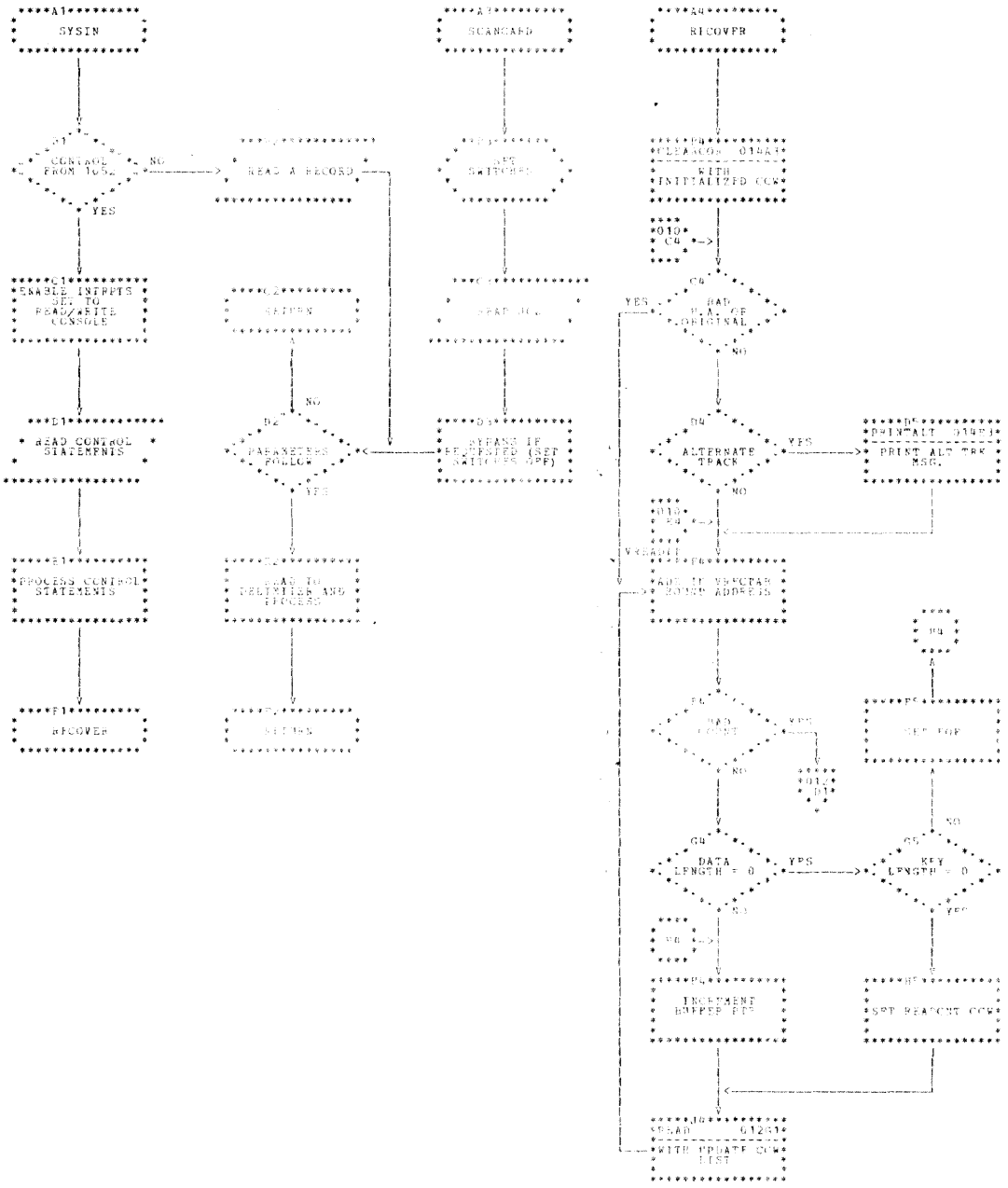


CHART CC. DADUMP Routine (Part 3 of 6)

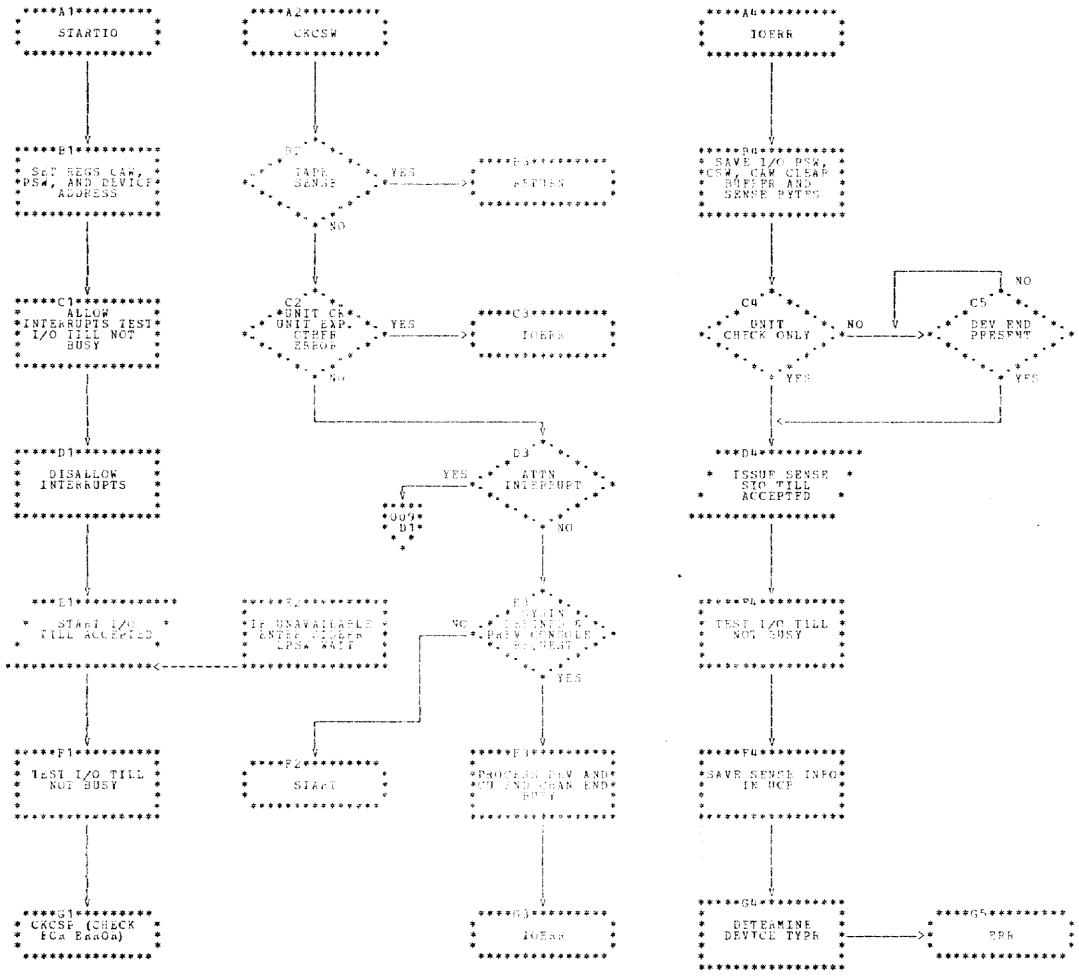


Chart CC. DADUMP Routine (Part 4 of 6)

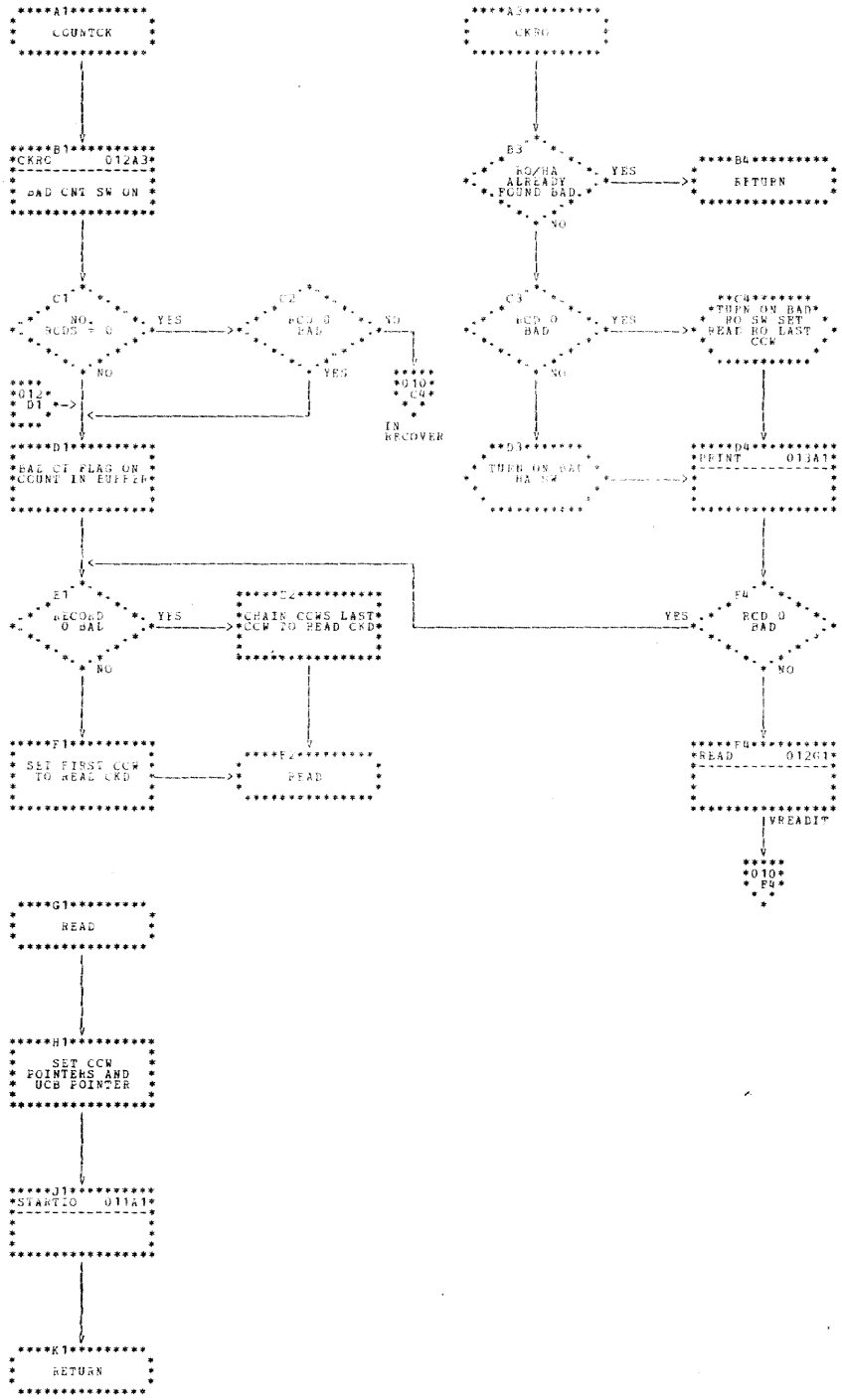


Chart CC. DADUMP Routine (Part 5 of 6)

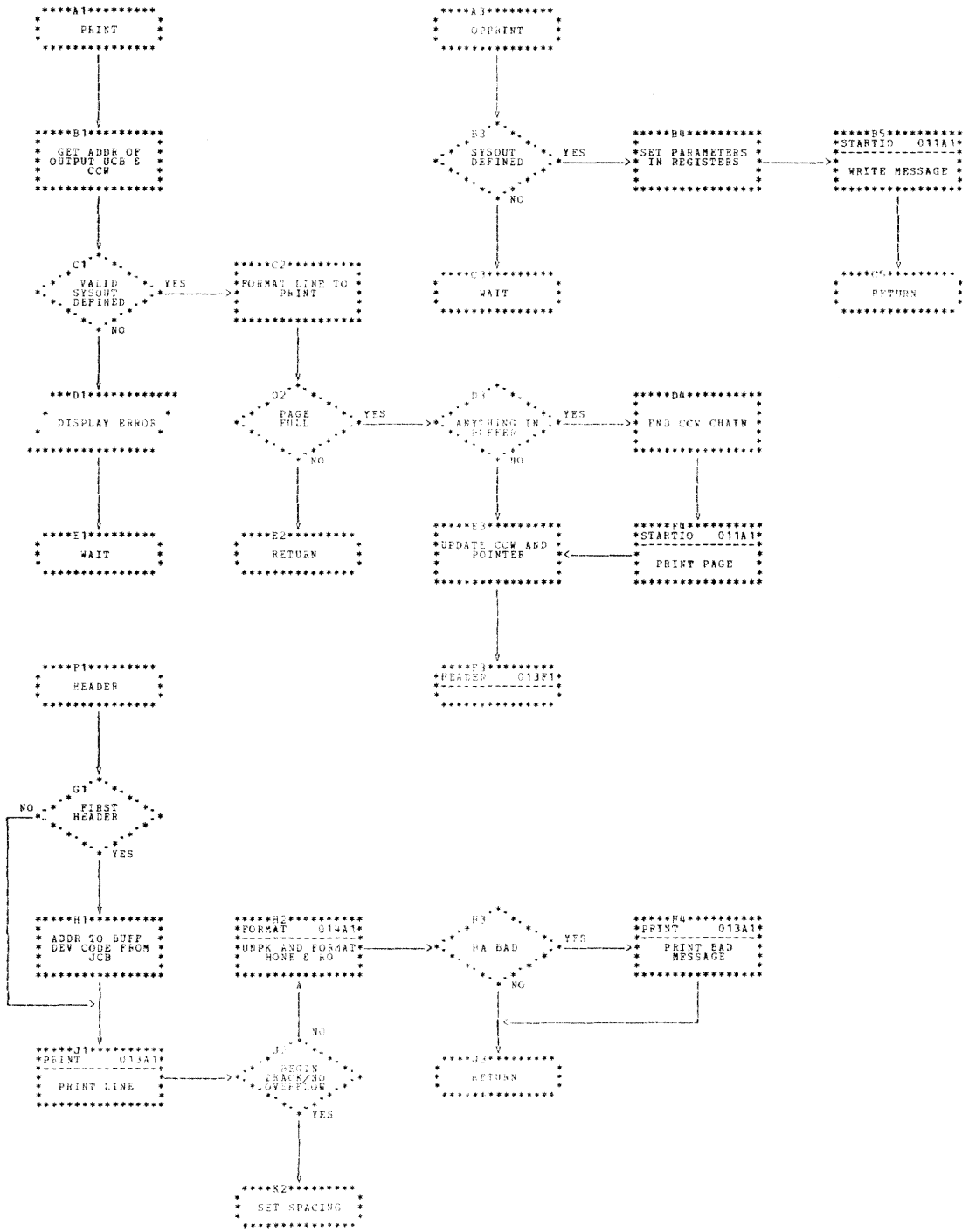
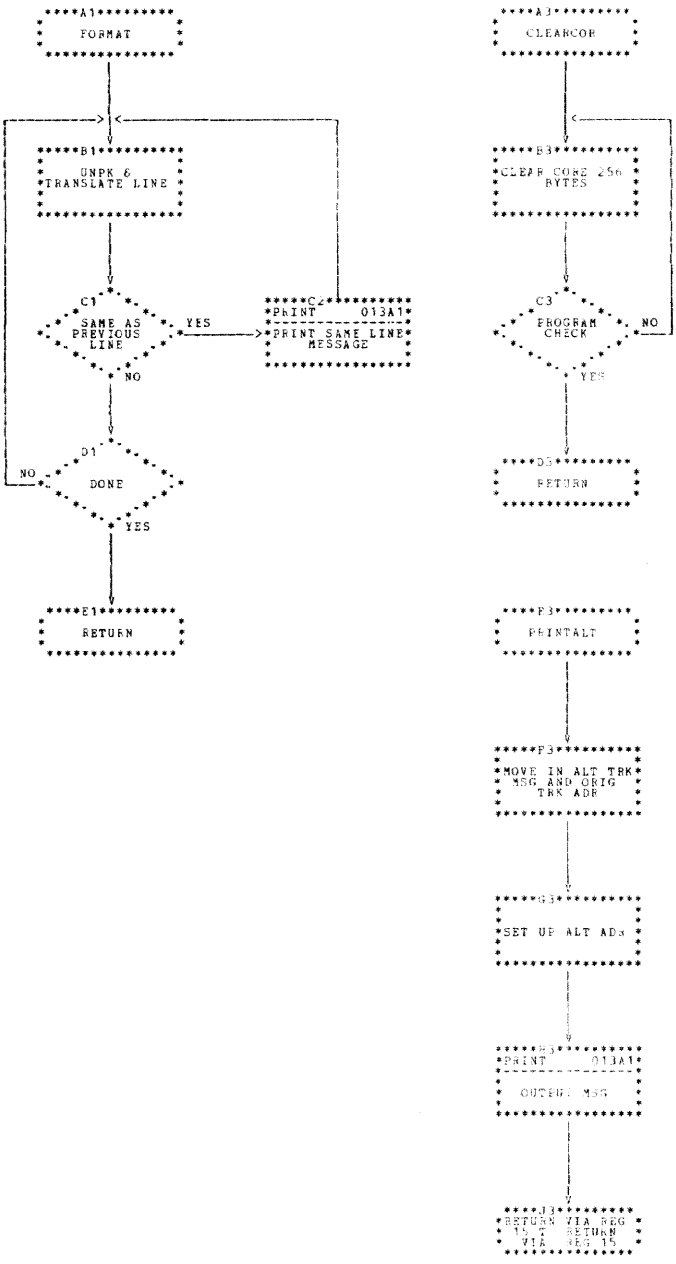


Chart CC. DADUMP Routine (Part 6 of 6)



SECTION 4: SYSTEM/360 MODEL 67 CORE DUMP PROGRAM - MODULE CEBCP

The System/360 Model 67 core dump program, module CEBCP, prints in special format the contents of registers and PSA areas for each central processing unit in the system, and prints the contents of main (core) storage with protection keys. The number of CPUs in the system is communicated via control cards. Addressable core storage is computed dynamically.

ATTRIBUTES: The core dump program operates outside and in support of TSS/360-67 as a privileged and nonreusable program.

ASSUMPTIONS AND RESTRICTIONS: The core dump program operates only on System/360 Model 67. PSA information is given for all CPUs in the system; however, if for any reason it is not included, processing continues as if there were only one CPU in the system. Locations 0-23 are destroyed by IPL. No error messages are issued to the operator.

PROGRAM OPERATION

The core dump program (Chart DD) is divided into three sections of code: (1) the IPL record, which loads in the first card of (2), the bootstrap loader, which continues to load itself and (3), the core dump program, into contiguous locations.

The IPL Record, containing a PSW, CCW, and eight unused bytes, is punched in columns 1-24 of the first card. The instruction address of the PSW is the same as the data address in the CCW, and both point to an available double-word location in core storage. This record overlays locations 0-23 at IPL time; therefore, the contents should be manually recorded by the operator before IPL.

The bootstrap loader consists of one or more 80-byte records; the first card is loaded by the IPL record, and the others are loaded by the first loader record. The loader then reads a program card into a buffer, analyzes the card image, and, if a TXT card, loads the determined number of text bytes immediately following the buffer in contiguous core locations. A counter is maintained to show the next immediate byte. If the card is found to be an END card, control is given to the dump program at the address calculated from this card. All general registers of the IPLd CPU are saved by this section.

CPU information contains the contents of control registers, floating-point registers, general registers, CAW, CSW, timer, interrupt codes, and the old PSWs. The PSWs are printed in hexadecimal format.

The primary and alternate PSA areas are printed for each CPU if they are addressable. For the IPLd CPU, the interrupt codes are not available since they reside in bytes 0-23 at IPL time. The primary PSA is printed first for all CPUs, followed by the alternate.

Each CPU printout begins on a new page with appropriate headings. The PSA areas are printed with their actual storage addresses (not location zero unless prefixing is inactive).

Following the CPU information, the contents of addressable core storage are printed. On the first line of each 2048 bytes of core, the storage protection key is printed. Thirty-two bytes are printed on each line. A line is printed indicating PSA areas. Consecutive lines containing the same information are printed only once.

INITIALIZATION

This group of routines saves all registers of the IPLd CPU; reads control cards; constructs a table of primary and alternate PSA addresses for all CPUs in the system; and, when the IPLd CPU is identified, passes control to the CPU dump section.

The FIRST routine stores general, floating-point, and multiple control registers of the IPLd CPU; saves old and new program PSWs; and constructs the channel command word (CCW) and channel address word (CAW) for reading control cards. Control is then passed to the READ routine.

The READ routine tests for a PFX card. Upon encountering a PFX card, control is passed to BUILDRTN to construct the PSA table.

BUILDRTN constructs the PSA table (Figure 5) using the CPU number as an index. Primary PSAs are analyzed followed by the alternates. At this time, an availability status is set "on" under the assumption that the PSA exists and is available to the system.

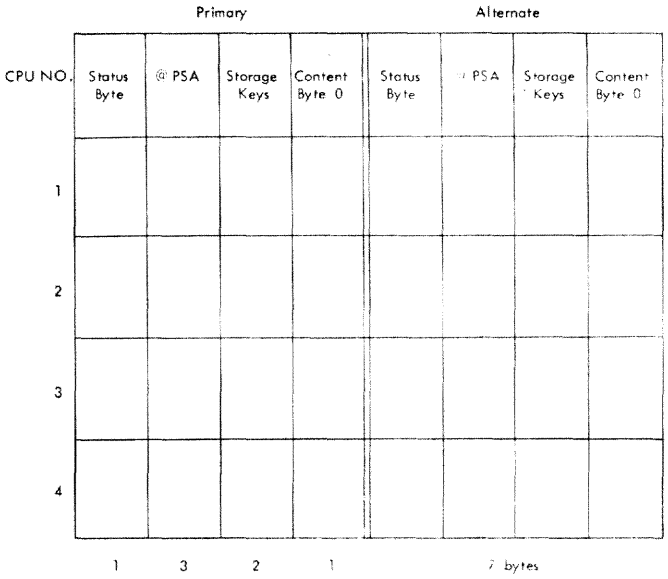


Figure 5. PSA Table Format

Since the card format makes provisions for the inclusion of information for two CPUs, the card image is scanned and the same procedure is followed as mentioned above, in building the PSA table. There can be a maximum of four prefix recordings if one card is used for each CPU; or a maximum of two prefix records if information for two CPUs is included on a record.

When an output control record (OPT card) is encountered, the PFXSW routine is given control. A test is made to see if a prefix record was read. If there was no prefix record, ONLYONE routine assumes that there is only one CPU in the system with a primary PSA starting at location 0 and no alternate PSA. If there was a prefix record, the PSA table is scanned and a last entry bit is marked for the last CPU prefix in the table.

After the OPT is analyzed, CONTINUE saves both storage keys for each PSA page in the PSA table, as well as the contents of byte 0 for all PSAs. If a program interruption is caused by an addressing exception, the PSA is considered not available and the availability bit is turned off in the status byte. If the PSA is available, zeros are stored in location zero. This procedure is continued until each PSA has been tested for availability. Now, non-zeros are stored in the IPLd CPUs location 0. The program next determines its own CPU identity by finding the address in the table which points to the non-zero byte.

Once the IPLd CPU is determined, its ID and prefixes are stored in the CPU dump area.

CPU Dump Routine

The CPU dump routine dumps all register contents and the contents of both PSAs pertaining to each CPU. Since the registers of the IPLd CPU are already saved, this CPU is dumped first.

The primary PSA of the CPU to be dumped is tested for availability. If it is available, locations 0-127 are formatted and printed, after all register contents have been formatted and printed. The entire PSA page is then dumped. The same procedure is followed for the alternate PSA of the CPU.

When the alternate PSA is printed, the CPU number is incremented by one and the CPU count is compared with the IPLd CPU number. If equal, the count is incremented by one. The new CPU number is stored in the CPU dump area.

The availability of the prefixes for the other CPUs is tested using the CPU number as an index. If the primary is not available, the alternate is tested for availability. If the alternate is not available, a message is printed to the effect that the CPU is not available. In either case, if available, locations 0-7 of the PSA are saved, a PSW is inserted in these locations, and the externally started CPU is directed to a routine that will save its registers. The CPU done bit is turned off before the other CPU is externally started.

While the externally started CPU saves its register contents, restores its location 0-7, and sets the CPU done bit on, the IPLd CPU loops and tests the CPU done bit for the "on" condition. When the on condition is recognized, the procedure is repeated for the next CPU in the system. If the on condition is not met within the required time, it is assumed to be non-responsive and a message is printed out.

This procedure is repeated for up to four CPUs or until the last entry bit is encountered. A test is made for "last entry" after each PSA has been handled.

COREDUMP Routine

The COREDUMP routine starts dumping from location hexadecimal 1000. If the location addressed results in an addressing exception, NOTAVAIL increments the dump pointer to the next 256K boundary. Dumping is continued until a test shows that all possible core has been handled. The COREDUMP rou-

tine formats and prints one page (storage) at a time using the PAGEPRNT routine.

The PAGEPRNT routine prints 4096 bytes of main (core) storage beginning at the address passed as a parameter in general register 4. The storage key is printed for each 2048 bytes. This routine prints 32 bytes of core storage per line.

The TESTDUP routine compares adjacent 32-byte areas of core storage. Duplicate lines are flagged and maintained until a non duplicate line is encountered. At this time, a summary of duplicate lines is printed.

Chart DD. CORE DUMP Routine (Part 1 of 3)

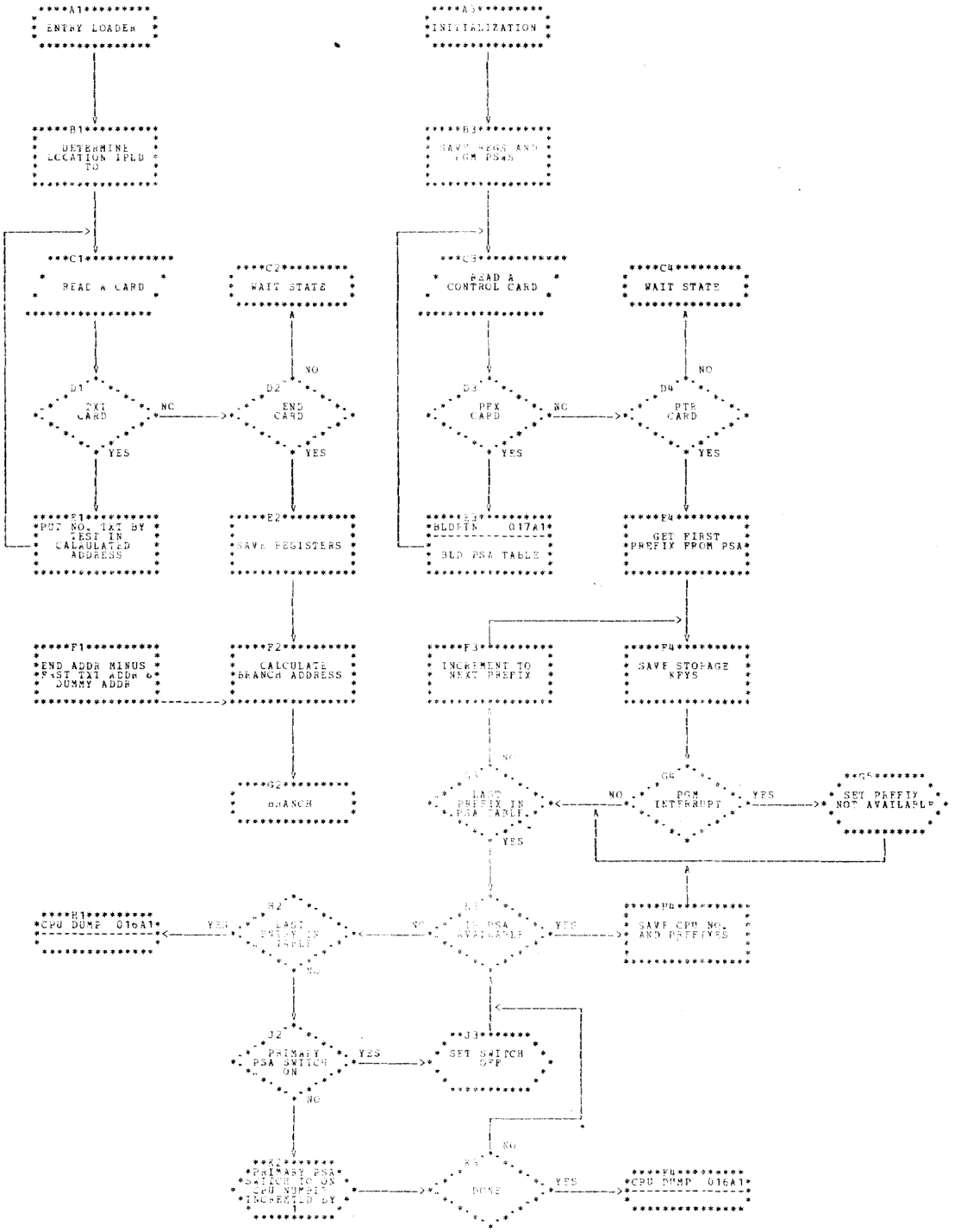


Chart DD. CORE DUMP Routine (Part 2 of 3)

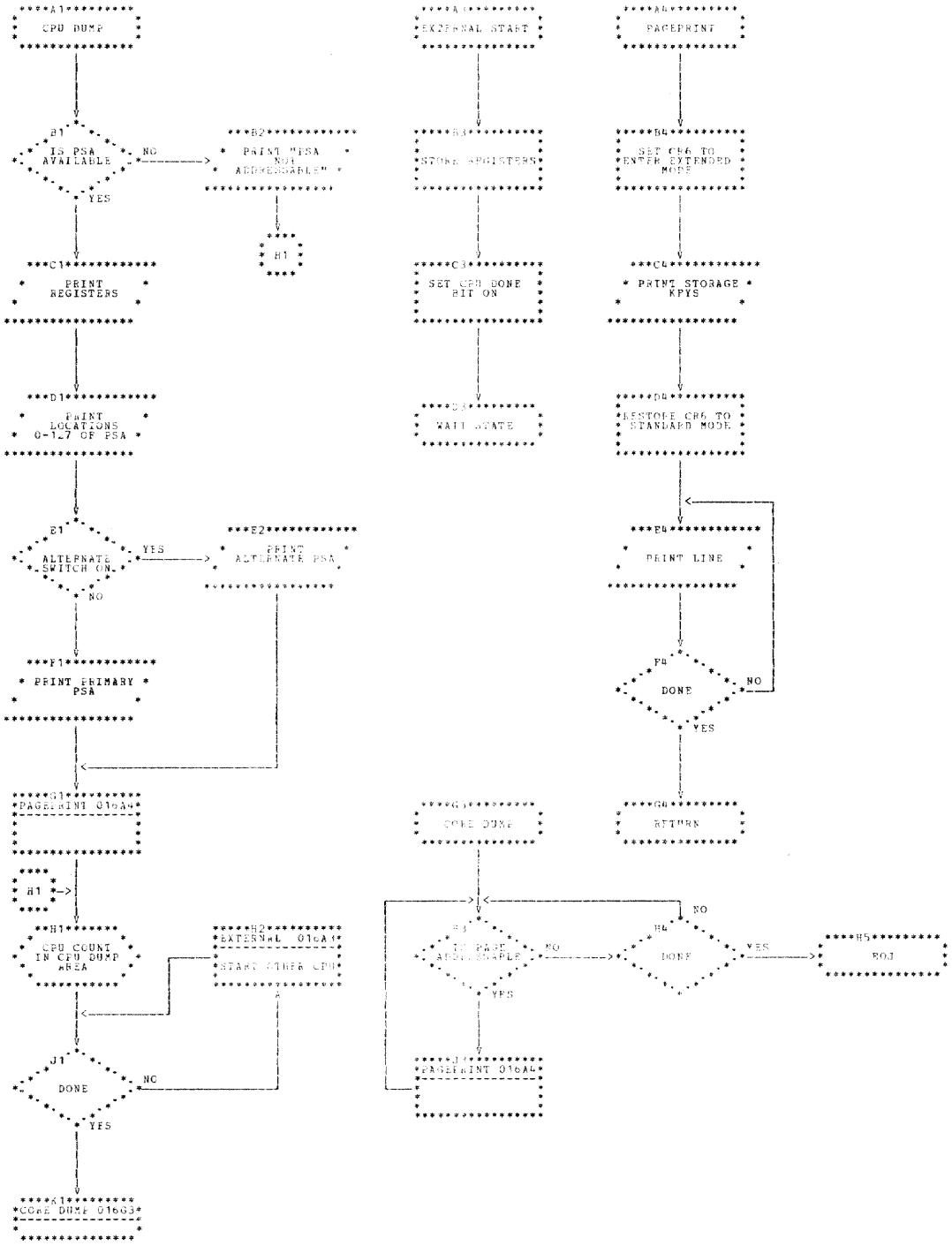
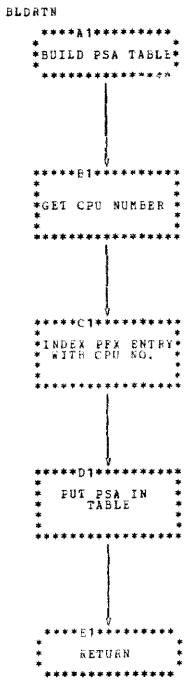


Chart DD. CORE DUMP Routine (Part 3 of 3)



SECTION 5: VAM 2 UTILITY PROGRAM (VAM2UT)--MODULE CEBVP

The VAM 2 Utility Program is a stand-alone program that allows the user to dump, display, and patch disk volumes in VAM 2 format. The program uses commands in standard Time Sharing Support System (TSSS) syntax.

ATTRIBUTES: VAM2UT operates outside and in support of TSS/360 as a privileged, non-relocatable, serially - reusable program.

RESTRICTIONS: as follows:

1. A maximum of 25 volumes are considered for a multi-volume data set.
2. A maximum of 2000 pages are allowed for a data set.
3. Nine pages (36K) of main storage are required to patch a 2314, using the $VAM command.
4. A member may have a maximum of 40 pages.

HARDWARE REQUIREMENTS: VAM2UT is a tape-resident module that can operate on any IBM System/360 with the required device configuration, as listed below.

Main Storage	32,000 bytes
Program Input Device	2400,2540
Command Input Device	2540,1052
Program Output Device	1403,1052
Message Output Device	1052
Other	2311,2314

PROGRAM INITIATION AND OPERATION: The absolute loader, which precedes the program, loads the program at location zero. The first eight bytes become a psw that passes control to location START.

START initializes the general registers, establishes the base registers, sets up the PSWs, determines the available buffer storage, and puts the program into the wait state with I/O interruptions enabled. The initial attention interruption, from the console, defines the operator console as SYSIN. The program leaves the wait state; it branches to the I/O status check routine, (CKCSW in STARTIO), to the I/O attention interruption handler, and to FOUNDSCB, which is the main communications routine. At this point the operator receives an initialization message from ATTN, and a dollar sign ($) from FOUNDSCB, as a prompting message. He may now either transfer control to the card reader via the CALL command, or continue to enter commands from the console. VAM2UT has no normal-completion exit: rather, it will enter a wait state. If an error occurs, a message will precede the wait state.

OPERATION

The absolute loader gives control to VAM2UT at the START entry point. PSWs and base registers are established; the available core pages are recorded. START sets up a dummy UCB area, and enters the wait state with I/O interruptions enabled. An attention interruption from the operator's console causes a UCB to be built for the console, which defines that device as SYSIN, and causes control to pass to the ATTN routine which prints the identifying message at the console. FOUNDSCB next prints the $ on the console; it enters the wait state for further input.

Commands are now analyzed, and control is passed to validate the parameters.

- CALL -- SYSIN is set to receive the next command from the card reader.
- RUN -- SYSIN is set to receive the next command from the operator's console.
- SET -- The second-level keyword is found and handled as follows:

  $DS
  : At NEXTDSCB the PAT is searched to find DSCB pages; the E DSCB slots are searched for data set name. BUILDSCB constructs a table of pointers to the DSCBs for the data set. Control then returns to FOUNDSCB to accept the next command.

  $DISK
  : SETDISK builds a UCB, reads in the PAT, indicates that the disk is defined, and returns control to FOUNDSCB to accept the next command.

  $DOUT
  : SETDOUT sets the UCB to define the printer, and returns control to FOUNDSCB.

- DUMP -- The second-level keyword is found and handled as follows:

  $PAT
  at READPAT, processing is initiated to read the PAT at T311. The PAT is then printed, using the DUMP routine, and control is returned to FOUNDSCB to accept the next command.

  $DSCB
  DSCBDUMP reads the PAT for the specified DSCBs, and gives control to DUMP to print them. Control is returned to FOUNDSCB to get the next command.

  $VAM
  VAMCK finds the specified page number and passes it to READPAGE to read the page. VDUMP prints the page, using DUMP, and returns control to FOUNDSCB to accept the next command.

  $DS
  PODSDMP ensures that the disk has been defined (via SET $DISK), and that a data set name exists. The pages are read by the READ routine, and printed by DUMP. Control is returned to FOUNDSCB to accept the next command.

  $POD
  Processing is the same as for $DS, but only the POD pages are read and printed.

  $PMD
  RDPOD reads in the POD; SRCHPOD gets the PMD address from the POM. MODUMP passes the PMD length to DUMP, which prints the PMD. Control is returned to FOUNDSCB to accept the next command. If $VM or $RM is entered as a second-level keyword to the DUMP routine, the PMD is printed with its member pages.

- DISPLAY -- The second-level keyword is found and handled as follows:

  $VL
  DSAVE initiates processing of the offset, length and type attributes. DOVL then reads the volume label using the READ routine, and displays it using OPPRNT. Control is returned to FOUNDSCB to accept the next command.

  $VAM
  VAMCK passes the page number to DSAVE which processes the offset, length, and type attributes. READPAGE reads the requested page, via OPPRNT, and returns control to FOUNDSCB to accept the next command.

  $PMD
  RDPOD reads the POD, using the READ routine. SRCHPOD finds the CSECT name, and SRCHPMD locates the control section dictionary. PATCHER then displays the PMD using the OPPRNT routine, and returns control to FOUNDSCB to accept the next command. If $VM or $RM is specified, the PMD and its member pages are displayed.

  CEIAP
  READPREL reads the prelude, using the READ routine, and passes control to PATCHER to verify the CSECT length. READIN sets up the prelude for BCTR1 to display it via OPPRNT. Control is returned to FOUNDSCB to accept the next command.

- PATCH -- processing is the same as for the DISPLAY command. The user is then prompted to see if a patch is to be made. The patch is made according to keyword type:

  $VAM
  The CHECKSUM is recalculated for a DSCB page. GOWRITE writes the page, and control is returned to FOUNDSCB to accept the next command.

  $VL
  DOVLP sets up the label and passes control to WRITE, which writes the label and returns control to FOUNDSCB to accept the next command.

  CEIAP
  The prelude length is calculated and passed to WRITE. WRITE patches the prelude on the disk, and returns control to FOUNDSCB to accept the next command.

  $PMD
  GOWRITE sets up the page; WRITE patches the PMD and returns control to FOUNDSCB.

The VAM2UT routines are summarized in Chart EE.

INITIALIZATION ROUTINES

START Routine

START receives control from the absolute loader, and performs tne following:

- sets up base registers.
- sets up PSWs.
- records buffer space available.
- enters the WAIT state pending an interruption.

ATTN Routine

ATTN receives control following an attention interruption from the operator console; it prints an identifying message on the console, and it passes control to FOUNDSCB.

COMMAND PROCESSING ROUTINES

FOUNDSCB Routine

FOUNDSCB prints a $ on the console, asking for input, and if the input comes from the card reader, it prints the card image on the operator console. Control now passes to CKBUF2 for command and parameter analysis.

CKBUF2 Routine

CKBUF2 scans the input for an acceptable command and, having found one, sets an identifying flag for COMRTN.

COMRTN Routine

COMRTN, entered at COMRTN1, analyzes all second level commands keywords. Here, syntax is checked and the decision is made as to which processing path to follow (see "OPERATION").

PROCESSING ROUTINES

The VAM2UT processing routines are logically organized by command/keyboard combinations, and by the program flow that follows from each combination. The RUN command re-enters the ATTN routine to set SYSIN= the card reader. All other commands are shown in tables 10 through 14, referenced to the possible parameters, and showing the logical order of processing routines for each command/keyword combination. Following the tables, the processing routines are summarized in alphabetical order.

Table 10 shows the CALL command.
Table 11 shows the SET command.
Table 12 shows the DUMP command.
Table 13 shows the DISPLAY command.
Table 14 shows the PATCH command.

Table 10. The CALL Command

CALL	
Routines Called	GETDEV UCBBUILD SCAN FOUNDSCB

Table 11. The SET Command

SET	$DS	$DISK	$DOUT
Routines Called	GETDSNAM GETDEV1 MOVEREL NEXTDSCB BUILDSCB FOUNDSCB	SETDISK GETDEV UCBBUILD SCAN GET DEV1 READ CKAGAIN MOVEREL READ DISKONN FOUNDSCB	SETDOUT GETDEV UCBBUILD SCAN FOUNDSCB

Table 12. The DUMP Command

DUMP	$PAT	$DSCB	$VAM	$DS	$POD	$PMD
Routines Called	PATCHER	DSCBDUMP	VAMCK	DSSET	PODSET	SETPMDSW
	GETDSO	FOUNDSCB	UCBBUILD	VAMCK	VAMCK	MODCHK
	GETDEV1	_____	SCAN	MAKETAB	MAKETAB	MODSAVE
	MOVEREL	.ALL/.E	RDPOD	READ	READ	RDPOD
	READ	DSCBDUMP	DUMP	DUMP	DUMP	READ
	PATCHER	PATCHER	READ	FOUNDSCB	FOUNDSCB	SRCHPOD
	DUMP	GETDSO	DUMP			MAKETAB
	SCAN	GETDEV1	SCAN			SRCHPMD
	PRINT	MOVEREL	PRINT			SCANCSD
	FOUNDSCB	READ	FOUNDSCB			READ
		CNVRELPG				FINDCSD
		DSCBDUMP				PMDPATCH
		READ				WANTPAT
		CNVRELPG				PATCHER
		READ				MODUMP
		DSCBDUMP				READ
		PRINT				DUMP
		DSCBDUMP				FOUNDSCB
		DUMP				
		PRINT				
		FOUNDSCB				

Table 13. The DISPLAY Command

DISPLAY	$VL	$VAM	$PMD	CEIAP
Routines Called	MODSAVE1	VAMCK	MODSAVE1	MODSAVE1
	CKDIS	MODSAVE1	CKDIS	CKDIS
	SCAN	CKDIS	SCAN	SCAN
	RDPOD	SCAN	RDPOD	READPREL
	DOVL	READPAGE	SRCHPMD	PATCHER
	FOUNDSCB	PATCHER	READPMD	FOUNDSCB
		PAGEOFF1	PMDPATCH	
		FOUNDSCB	PATCHER	
			FOUNDSCB	

Table 14. The PATCH Command

PATCH	$VAM	$VL	CEIAP	$PMD
Routines Called	VAMCK	MODSAVE1	MODSAVE1	MODSAVE1
	MODSAVE1	PATCH	PATCH	PATCH
	PATCH	SCAN	SCAN	SCAN
	SCAN	DOVL	READPREL	RDPOD
	READPAGE	ASKAGAIN	PATCHER	SRCHPOD
	PATCHER	DOVLP	ASKAGAIN	SRCHPMD
	ASKAGAIN	FOUNDSCB	WRITPREL	READPMD
	PATDUMP		FOUNDSCB	FINDCSD
	CKAGAIN			PMDPATCH
	FIXCHK			ASKAGAIN
	GOWRITE1			GOWRITE1
	FOUNDSCB			FOUNDSCB

ASKAGAIN Routine

ASKAGAIN uses OPPRNT to prompt the operator either to continue with the PATCH operation, or to cancel it following the display. A "continue" reply causes control to pass to the patching routine for the particular keyword. A "stop" reply passes control to FOUNDSCB to prompt for the next command.

BUILDSCB Routine

BUILDSCB builds the DSCBREL table (see Figure 6), which consists of a series of fullword DSCB pointer entries: the first byte contains the DSCB slot number; the second byte contains the relative volume number; the second half-word contains the relative page number. After one entry is completed for each DSCB in the data set, exit is made to FOUNDSCB.

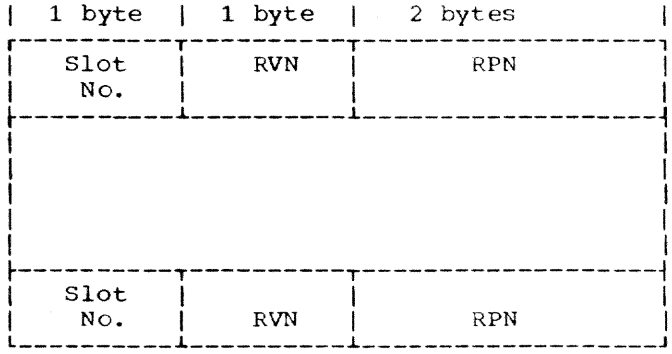


Figure 6. DSCBREL

CKAGAIN Routine

CKAGAIN is entered to read in the PAT (for a 2314, the second PAT page is read first) via READ. If RELTAB (the Relocation Entry Table) has not yet been built, MOVEREL is called to build it. In any case, control is then passed to the next processing routine, as shown in the following table.

$VAM		X		
$PAT	X			
$DSCB			X	
$DISK				X
PATDUMP1	X			
FIXCHK		X		
DSCBDUM1			X	
DISKONN				X

CHKDIS Routine

CHKDIS validates the syntax for a DISPLAY command, sets the display length in DOSAREA, and branches to SCAN for EBCDIC to binary conversion. On return from SCAN, CHKDIS passes control to RDPOD.

DISKONN Routine

DISKONN flags the disk as "defined" following a SET $DISK operation. Control is then passed to FOUNDSCB.

DOVL Routine

DOVL is entered when a volume label is being patched. First, GETDEV2 reads the volume label into core. Then, the patch is moved from its buffer to REVPAT. Finally, the volume label is displayed, via OPPRNT, at the console. If a PATCH was specified, control is passed to ASKAGAIN. Otherwise, exit is made to FOUNDSCB.

DOVLP Routine

DOVLP writes the patch to a volume label area on disk. First, SCAN is called for any EBCDIC to binary conversion; then, WRTVLBL calls WRITE to write the patch. Exit is made to FOUNDSCB.

DSCBDUMP Routine

DSCBDUMP is entered to dump the DSCBs to the 1403 printer. If all DSCBs are to be dumped, or if no data set was specified, control is passed to READPAT; control goes through READPAT, and CKAGAIN, to DSCBDUM1 where all DSCBs on the volume are dumped. When a dump is requested for all format E DSCBs, or for the DSCBs for a given data set, DSCBDUMP ejects a page on the printer, reads a DSCB page into core, and dumps either all format E DSCBs, or all DSCBs for one data set. Exit is made to FIXCCW.

DSCBDUM1 Routine

DSCBDUM1 is entered from CKAGAIN to dump all DSCBs on a volume. Each DSCB entry in the PAT is examined, and DSCB pages are read into core. Then, either the specified slots, or the whole pages, are dumped to the printer. Exit is made to FIXCCW.

FINDCSD Routine

FINDCSD searches the PMD for a control section dictionary (CSD), and records the number of definitions within that CSD. Exit is made to SCANCSD.

FIXCHK Routine

FIXCHK calls CHKSUM to perform checksum operations on DSCB slots for patch process-

ing. Exit is made to GOWRITE1 to write the patch.

GETDEV Routine

GETDEV is entered, for a SET $DISK operation, to build the unit control block (UCB) for disk. Public packs, where required are mounted, as are packs for data sets that span more than one volume. The routine is initiated by a VAM2UT message to the operator. GETDEV2 reads the volume label; GETDEV3 puts the PAT page number and VOLID in the UCB and marks that phase of the operation as "complete". Exit is made to CKAGAIN.

GETDSNAM Routine

GETDSNAM retrieves the dsname, sets up to read the PAT, and passes control to CKAGAIN.

GOWRITE1 Routine

GOWRITE1 calls WRITE to write the PAT page(s). Control is then passed to FIXCCW.

MODSAVE1 Routine

MODSAVE1 reads the offset, length, and type attributes from a command parameter; it saves the EBCDIC displacement of the area to be displayed or patched, and passes control to CKDIS for a display operation, or PATCH for a patch.

MODUMP Routine

MODUMP calculates the number of PMD pages to dump; it calls READ to read in the pages; it calls DUMP to dump them to the printer. Control is then passed to FIXCCW.

MOVEREL Routine

MOVEREL is entered to build a relocation table of full-word entries for relocated pages. Control is then passed to CKAGAIN2, within CKAGAIN.

PAGEOFF1 Routine

PAGEOFF1 calculates the number of pages involved in the patch, or display. The number of bytes involved is calculated (and is doubled for a hex operation); OPPRNT displays the patch at the console. If a display was requested, control is passed to FIXCCW; for a patch, ASKAGAIN receives control.

PATCH Routine

PATCH validates the patch and its length; it moves the patch to a patch area; and it passes control to SCAN.

PATCHER Routine

PATCHER reads in the page(s) to patch, and sets up a message which displays the area to be patched at the console. Exit is made to PAGEOFF1 to display the patch and write it to disk.

PATDUMP Routine

PATDUMP sets the $PAT flag, and enters GETDSO in the GETDSNAM routine.

PATDUMP1 Routine

PATDUMP1, entered from CKAGAIN, dumps the PAT to the printer, and exits to FOUNDSCB.

PODSDUMP Routine

PODSDUMP calculates the number of pages to dump; it builds an external page table (via MAKETAB) for the data set. At PDMP3, the pages are read and dumped using the extents table as an index. Exit is made to FIXCCW.

RDPOD Routine

RDPOD initially calls MAKETAB to build a table of POD pages; it then reads the first page of the POD. Exit is made to SRCHPOD.

READPAGE Routine

READPAGE reads a specified page and, if required, dumps it to the printer. Exit is made to FIXCCW.

PMDPATCH Routine

PMDPATCH is entered from FINDCSD for a $PMD operation; it restores the page table (via REFIX), and, if a DUMP or DISPLAY is requested, exits immediately to PATCHER. For a PATCH operation, PMDPATCH asks the operator for the type of patch (length or attributes), and uses SYSIN to read the reply. WANTATRB calculates the address of the CSECT and CSD. For an attribute change, OPPRNT displays the attributes and requests the changes; SYSIN reads the change. PMDPATCH then makes the attribute change and, at DONELOOP, allows the operator to re-enter the program for further changes, or to terminate the program (which causes control to pass to ASKAGAIN). For a length change, CHGLEN calls SCAN to convert the CSECT length to binary; it calls OPPRNT to display the old length and to request the change. SYSIN reads the reply; SCAN converts it. CHGLEN makes the change and exits to ASKAGAIN.

READPMD Routine

READPMD reads the program module dictionary (PMD) into core, and then exits to FINDCSD.

READPREL Routine

READPREL reads the prelude from disk into core. If a dump is requested, the prelude is dumped to the printer via DUMP, and control passes to FIXCCW; for a patch or display, the routine exits to PATCHER.

SCAN Routine

SCAN converts any valid EBCDIC character to binary notation, and returns control to the caller. The BITOHEX subroutine converts binary notation to EBCDIC characters.

SETDIN Routine

SETDIN positions the SYSIN UCB buffer, and calls UCBBUILD to complete the UCB. SETDIN defines SYSIN, in the UCB, as the card reader and it passes control to FOUNDSCB.

SETDISK Routine

SETDISK sets up a buffer for initializing the disk; it flags the operation as $DISK; and it exits to GETDEV.

SETDOUT Routine

SETDOUT defines the printer as SYSOUT, in the SYSOUT UCB, and exits to FOUNDSCB.

SRCHPMD Routine

SRCHPMD reads the module's PMD into core; it determines the displacement of each CSD. SCANCSD, entered from SRCHPMD, finds the required CSECT or entry point definition in the CSD. Any Q-cons found during this PMD search are ignored. The routine exits to READPMD.

SRCHPOD Routine

SRCHPOD searches the POD for the member or alias specified in the current command. The member name or alias is hashed; pages are read in as needed. When the member is found, MAKETAB is called to make an external page table for the data set. Exit is made to SRCHPMD for PMD operations, and to MODUMP for other dump requests.

VAMCK Routine

VAMCK validates the syntax for a $VAM operation; it calls UCBBUILD to build and scan a UCB; it calls GETDEV to validate the device. SCAN converts the page number to binary. The routine exits to RDPOD for a dump; otherwise, control passes to MODSAVE1.

WRITPREL Routine

WRITPREL is entered to write the prelude; it calls WRITE to perform the actual write operation; and it exits to FIXCCW.

Chart EE. VAM2UT Routines (Part 1 of 15)

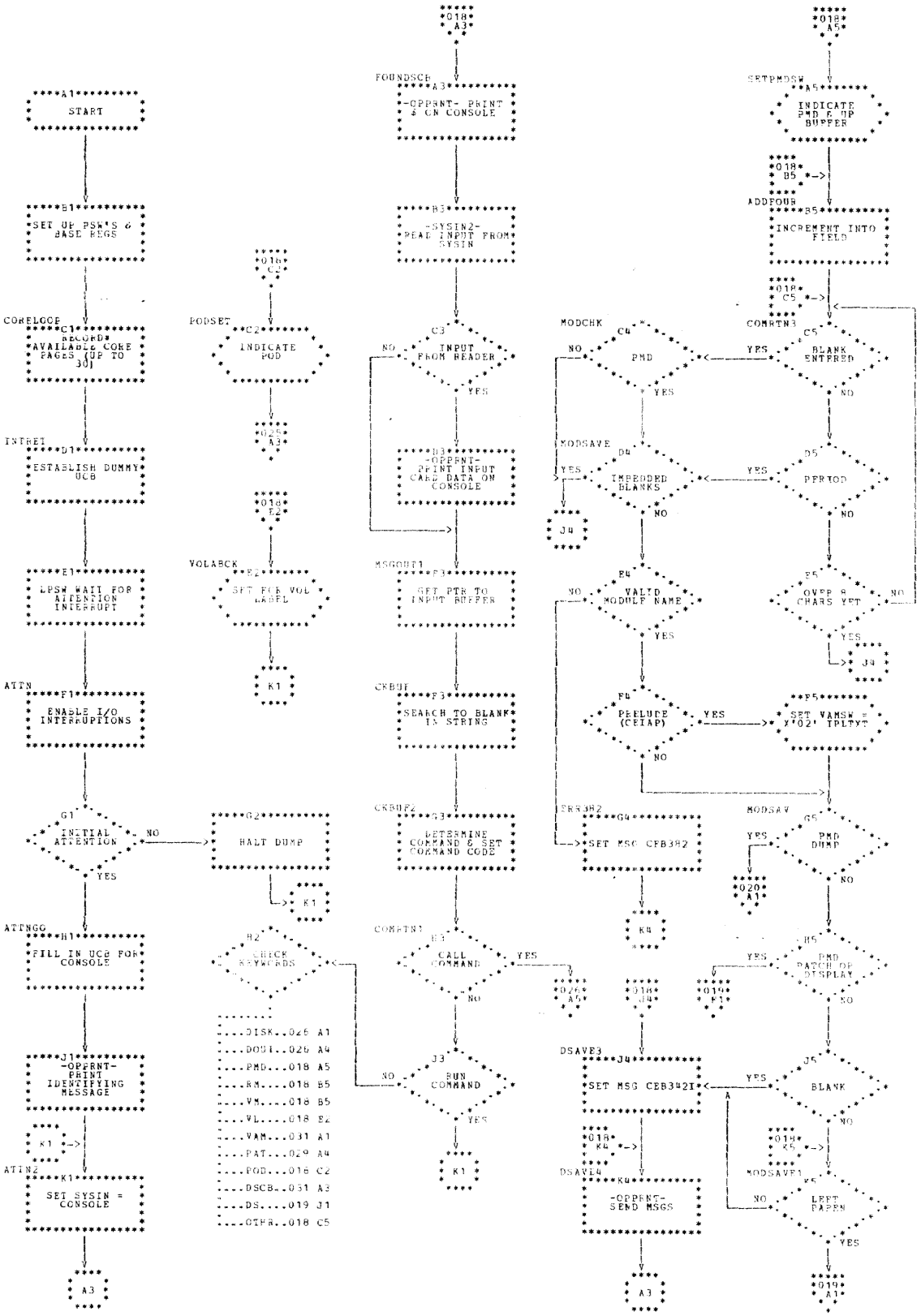


Chart EE. VAM2UT Routines (Part 2 of 15)

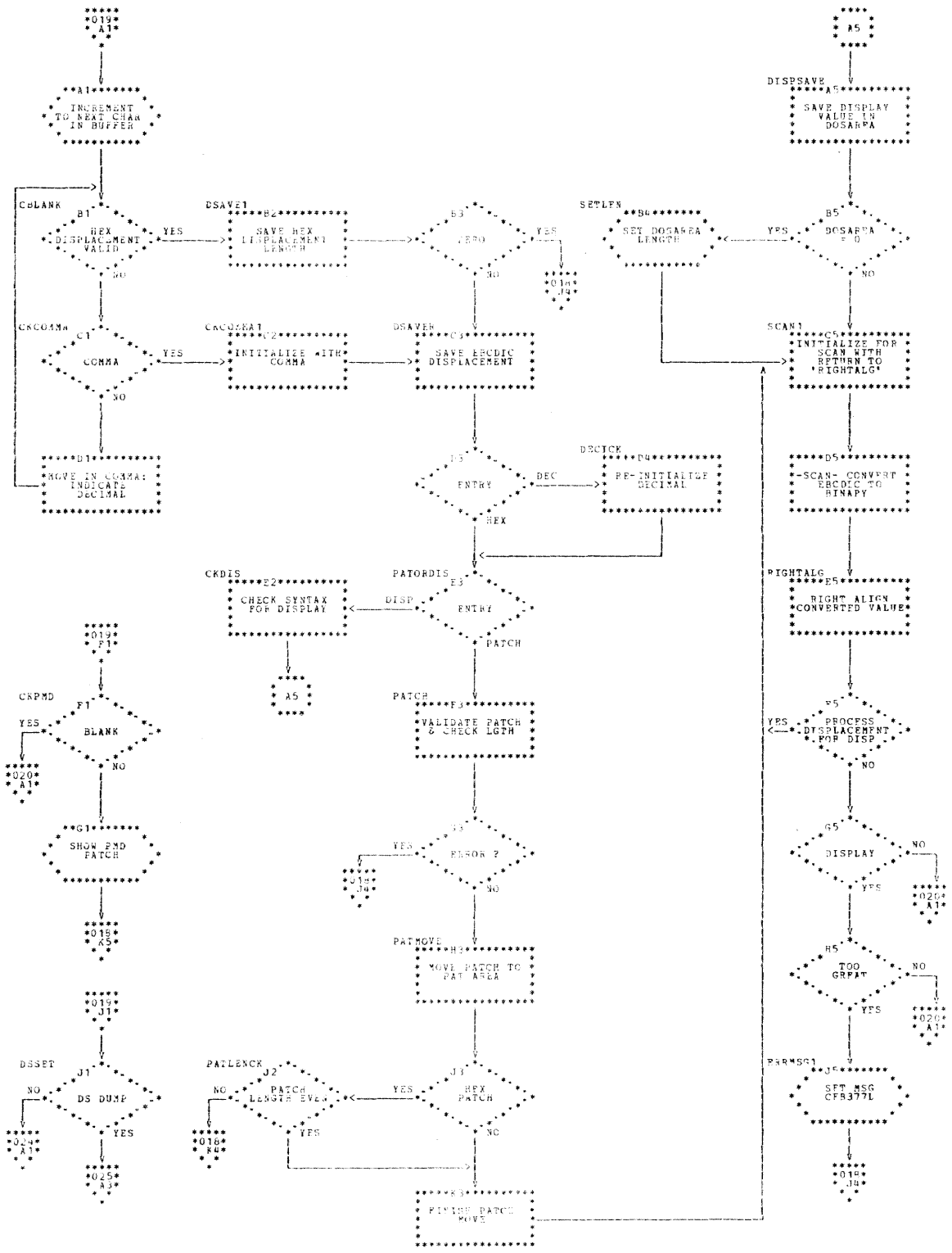


Chart EE. VAM2UT Routines (Part 3 of 15)

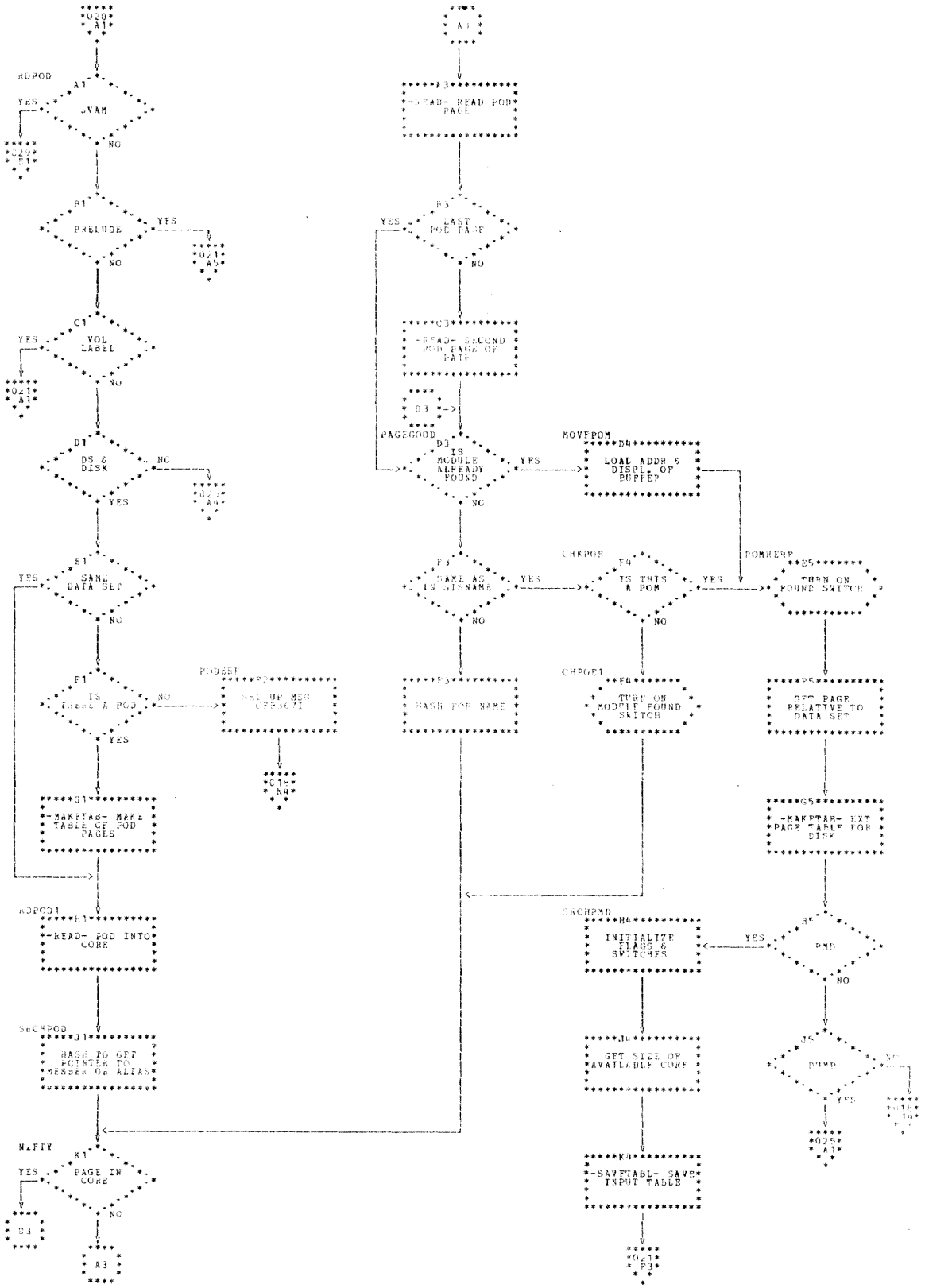


Chart EE. VAM2UT Routines (Part 4 of 15)

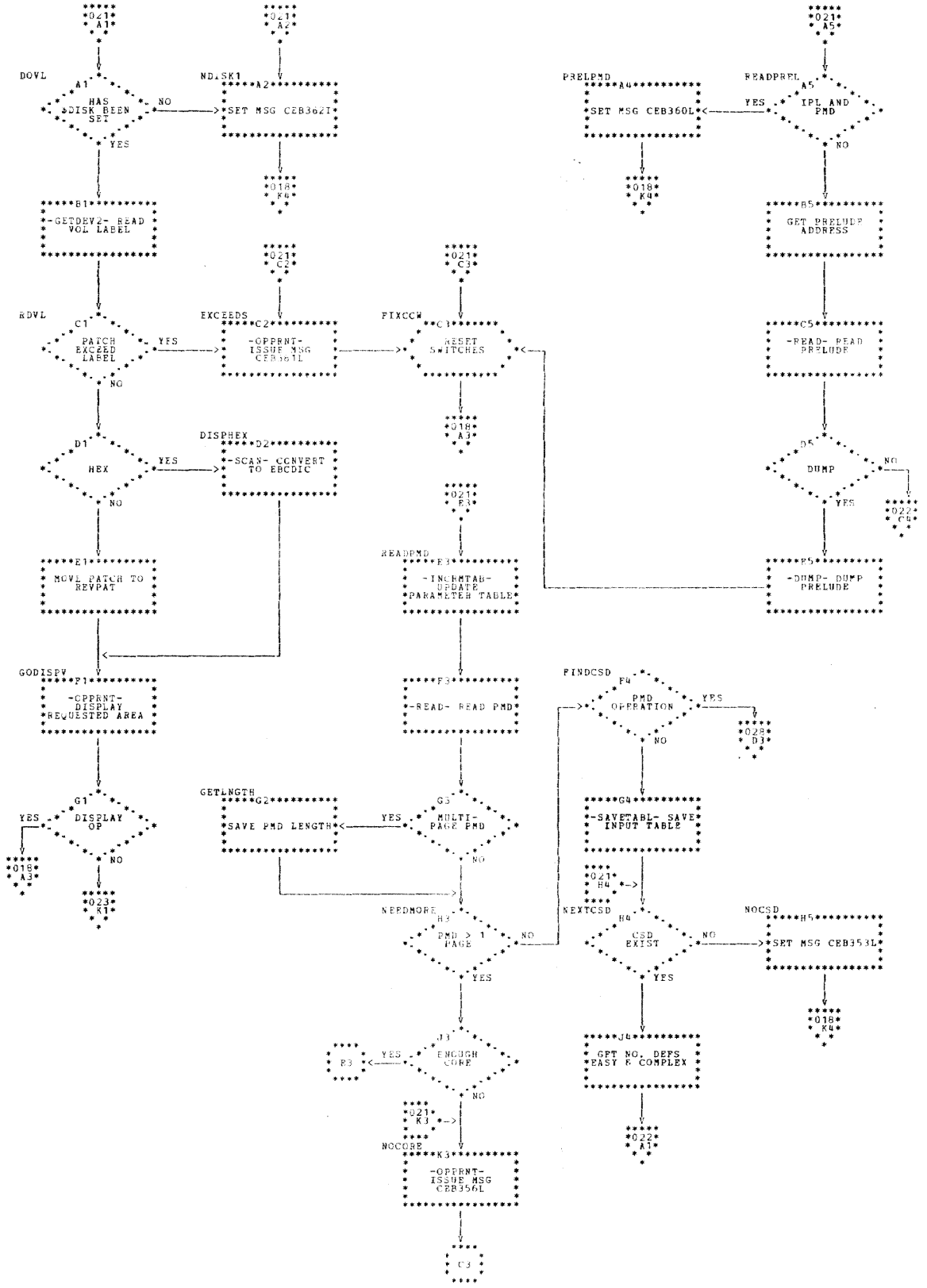


Chart EE. VAM2UT Routines (Part 5 of 15)

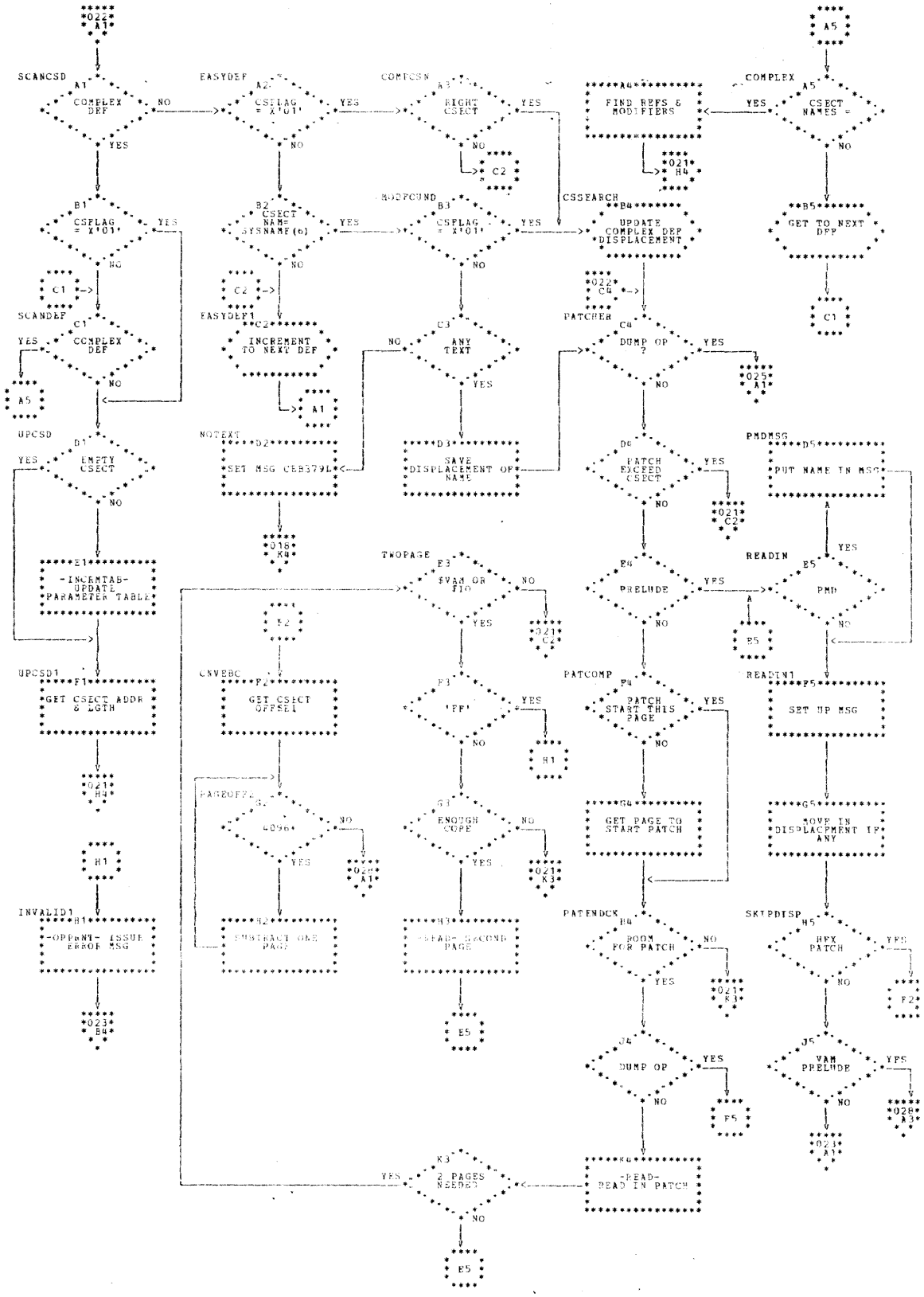


Chart EE. VAM2UT Routines (Part 6 of 15)

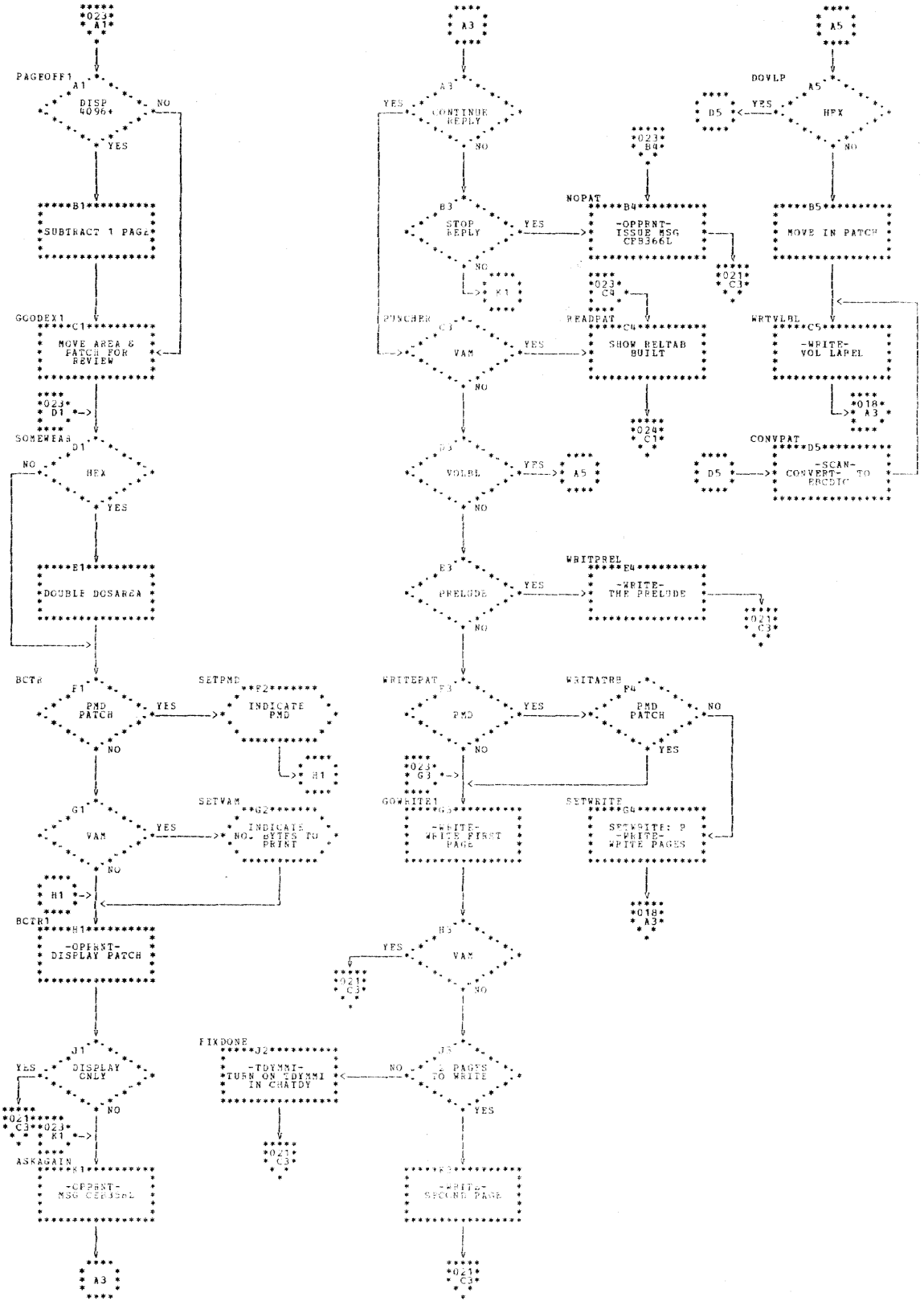


Chart EE. VAM2UT Routines (Part 7 of 15)

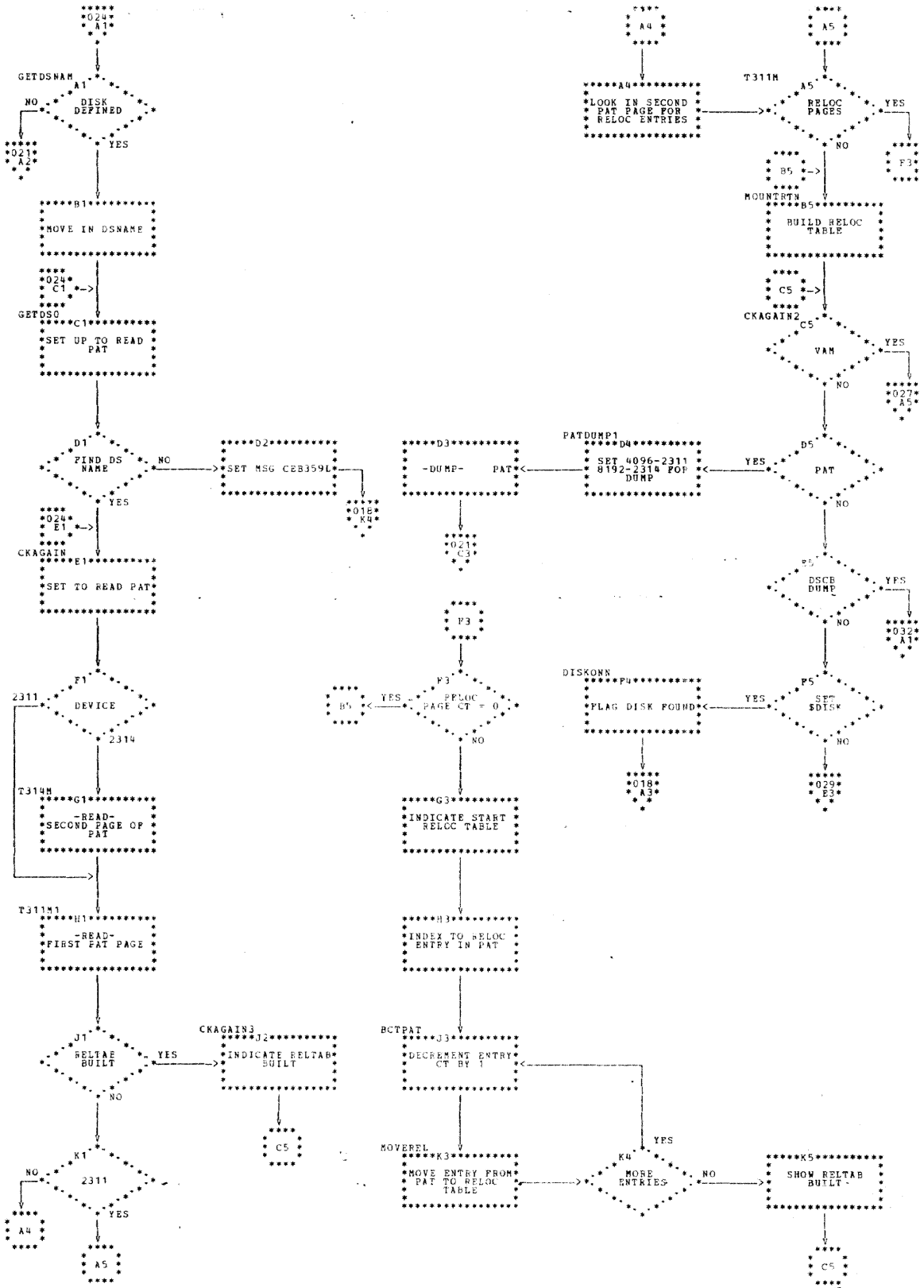


Chart EE. VAM2UT Routines (Part 8 of 15)

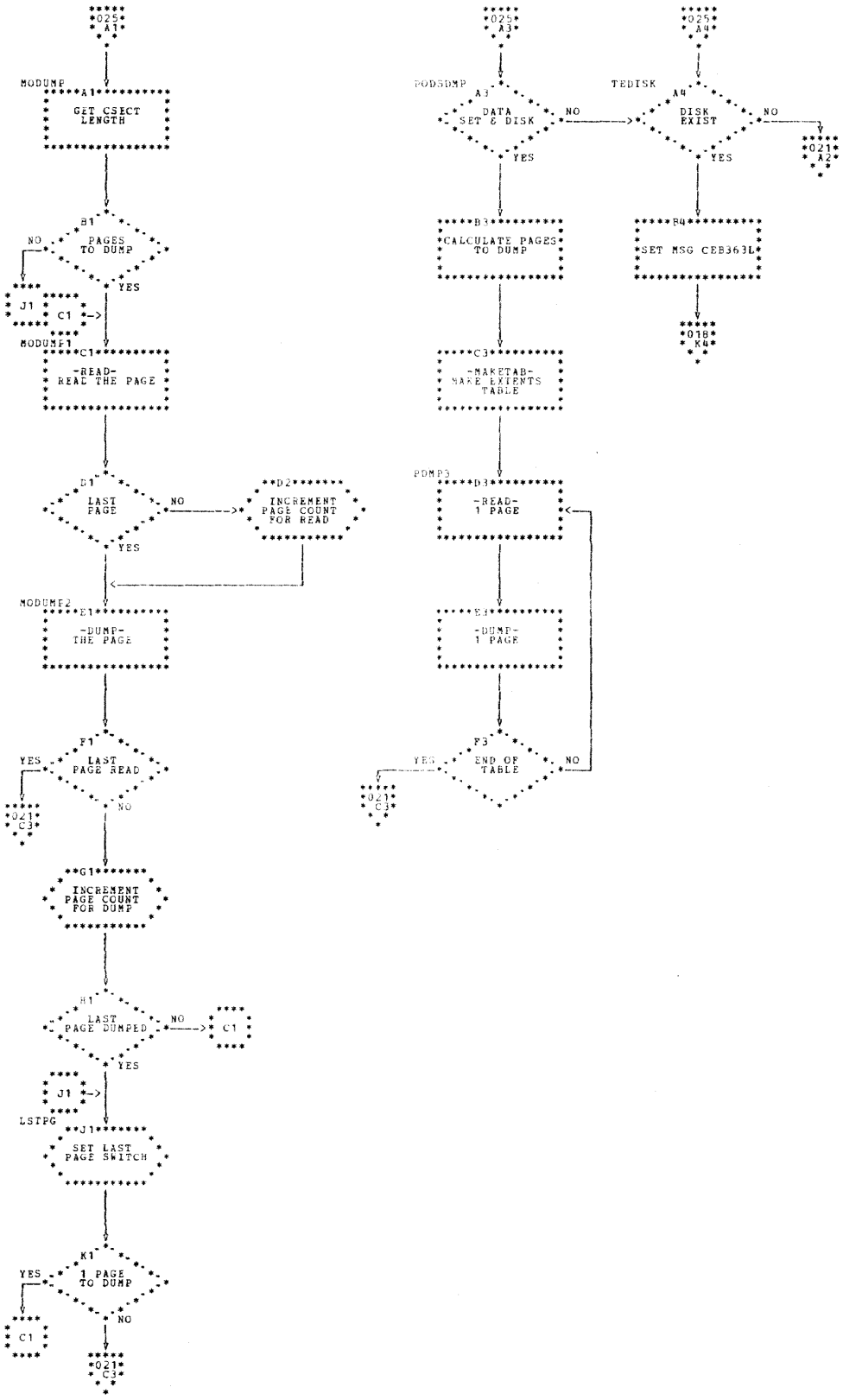


Chart EE. VAM2UT Routines (Part 9 of 15)

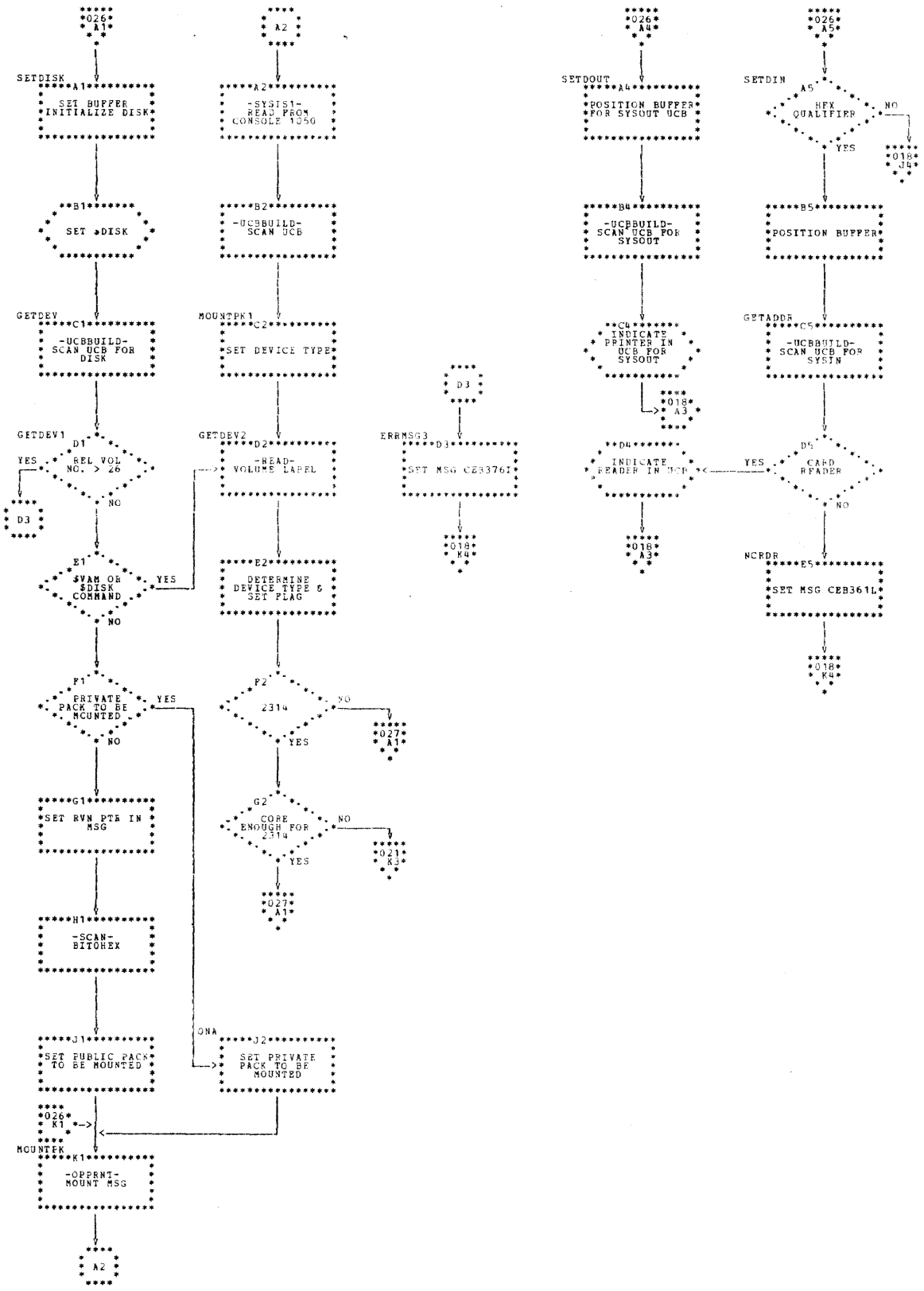


Chart EE. VAM2UT Routines (Part 10 of 15)

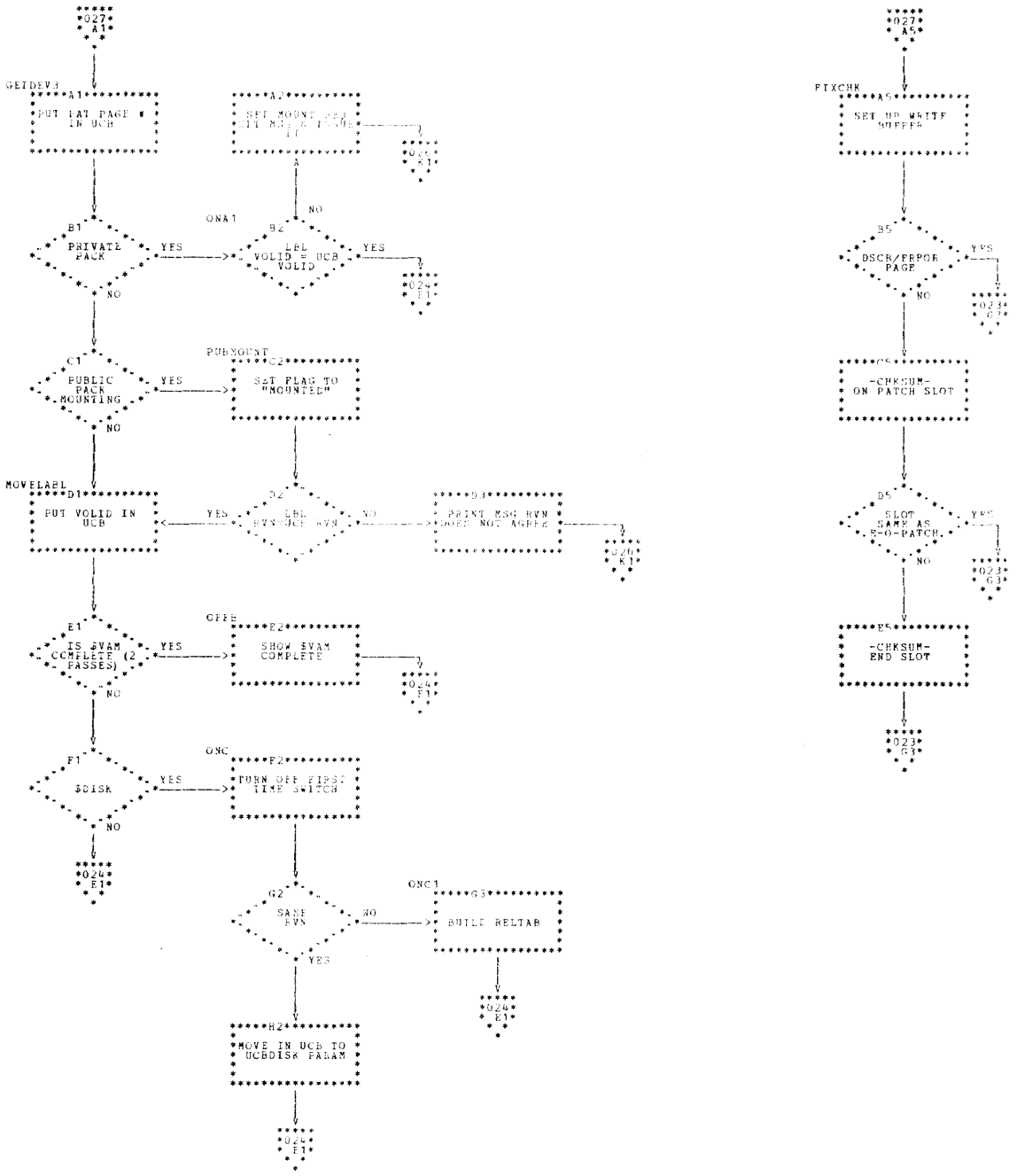


Chart EE. VAM2UT Routines (Part 11 of 15)

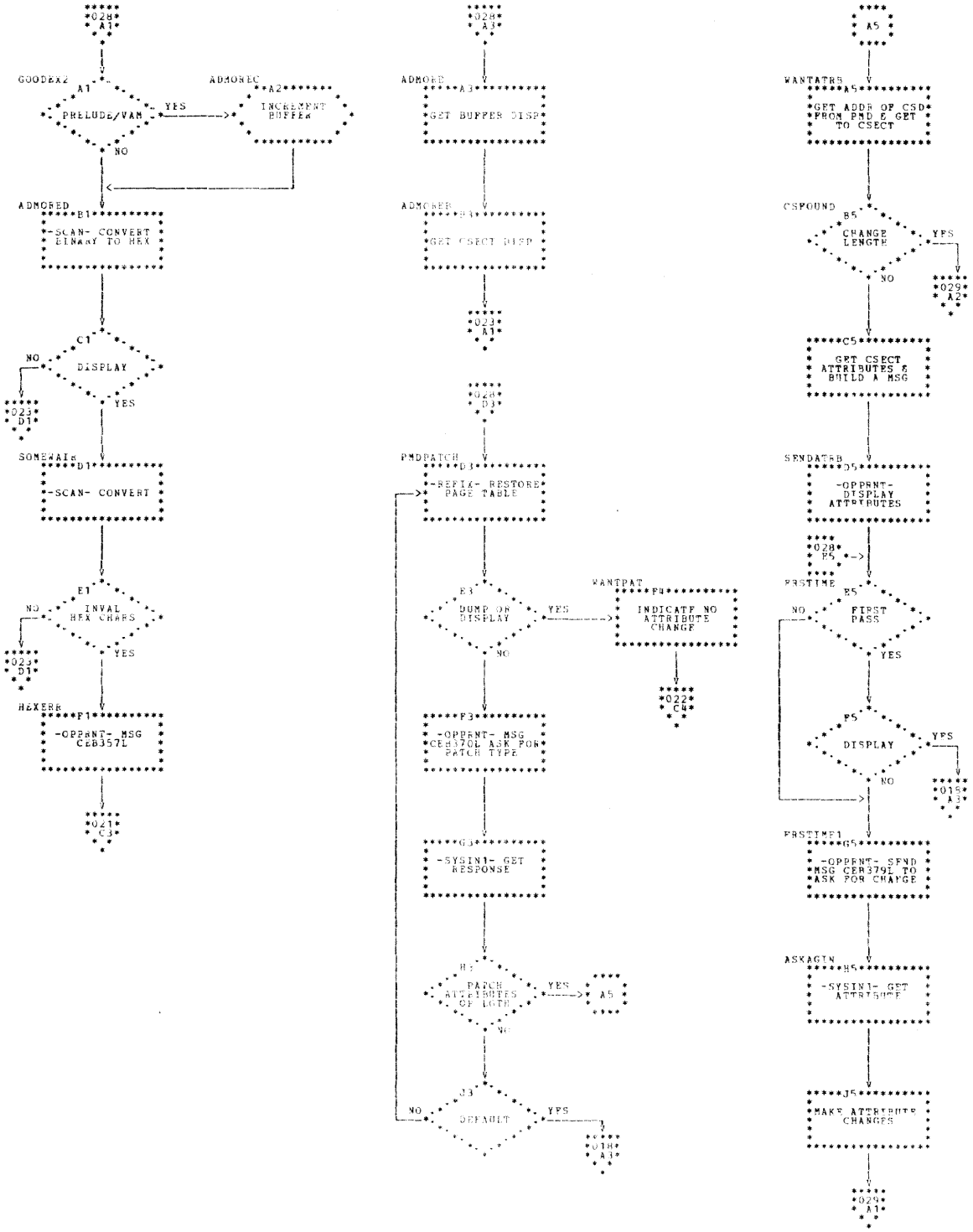


Chart EE. VAM2UT Routines (Part 12 of 15)

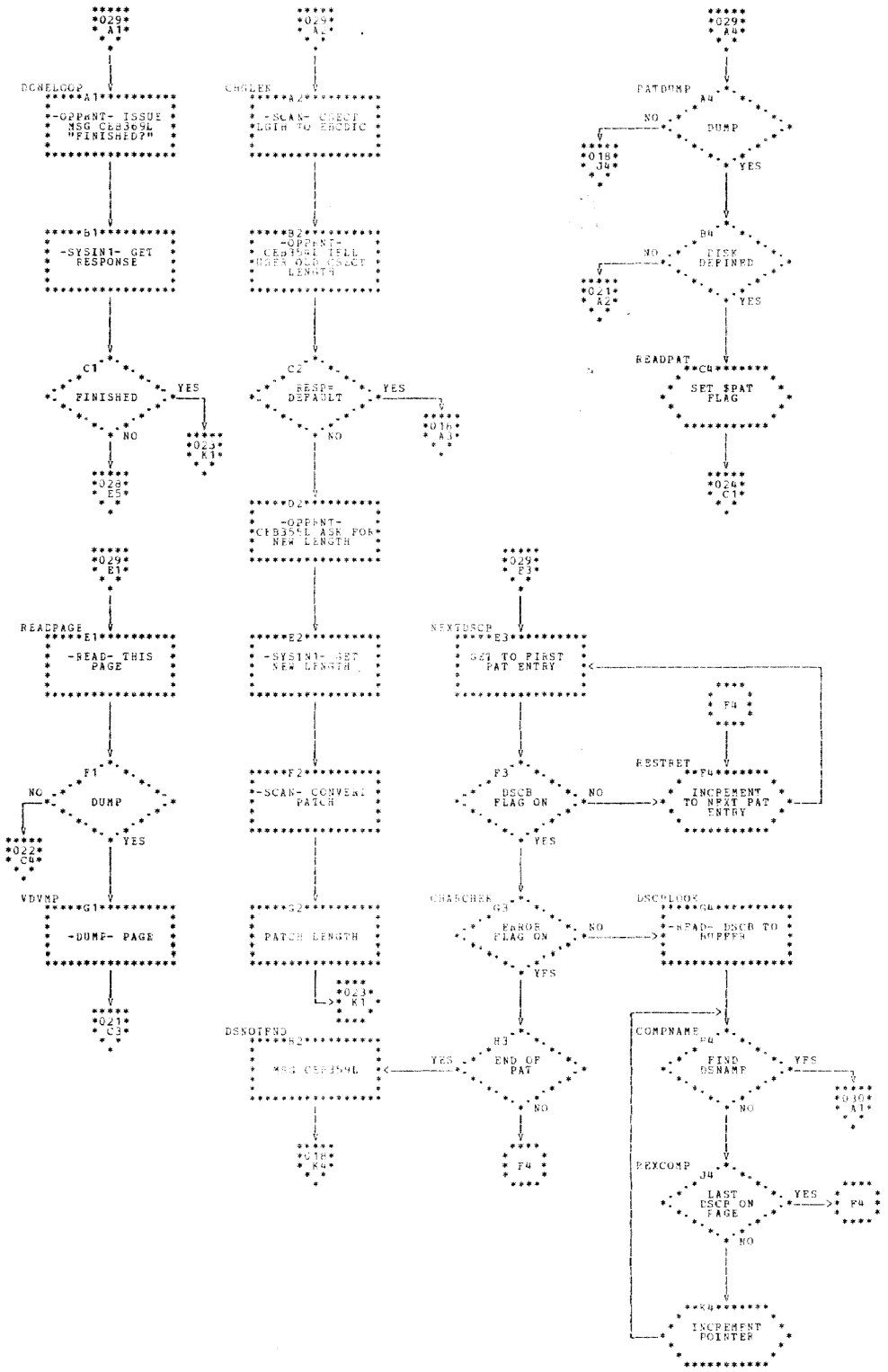


Chart EE. VAM2UT Routines (Part 13 of 15)

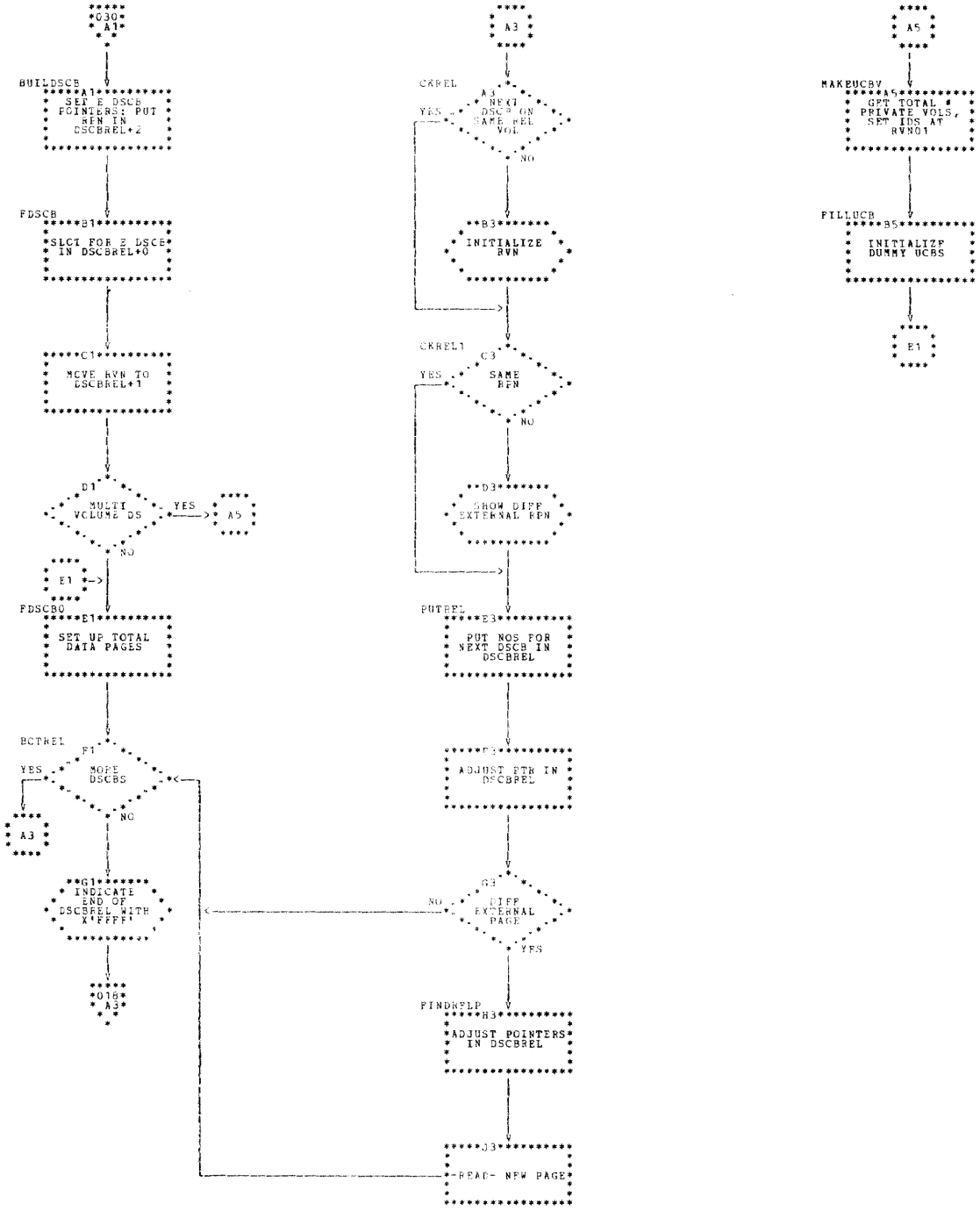


Chart EE. VAM2UT Routines (Part 14 of 15)

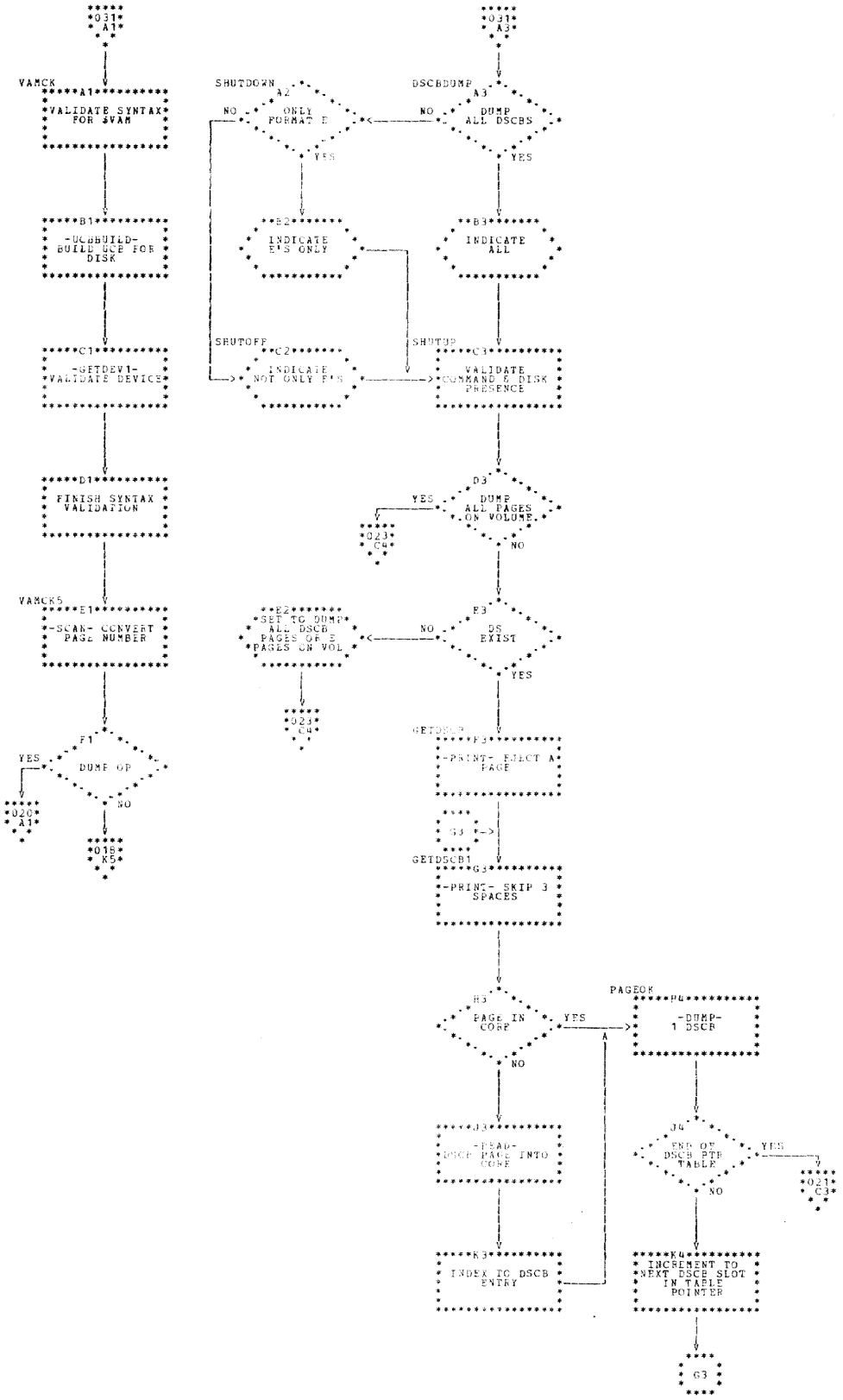
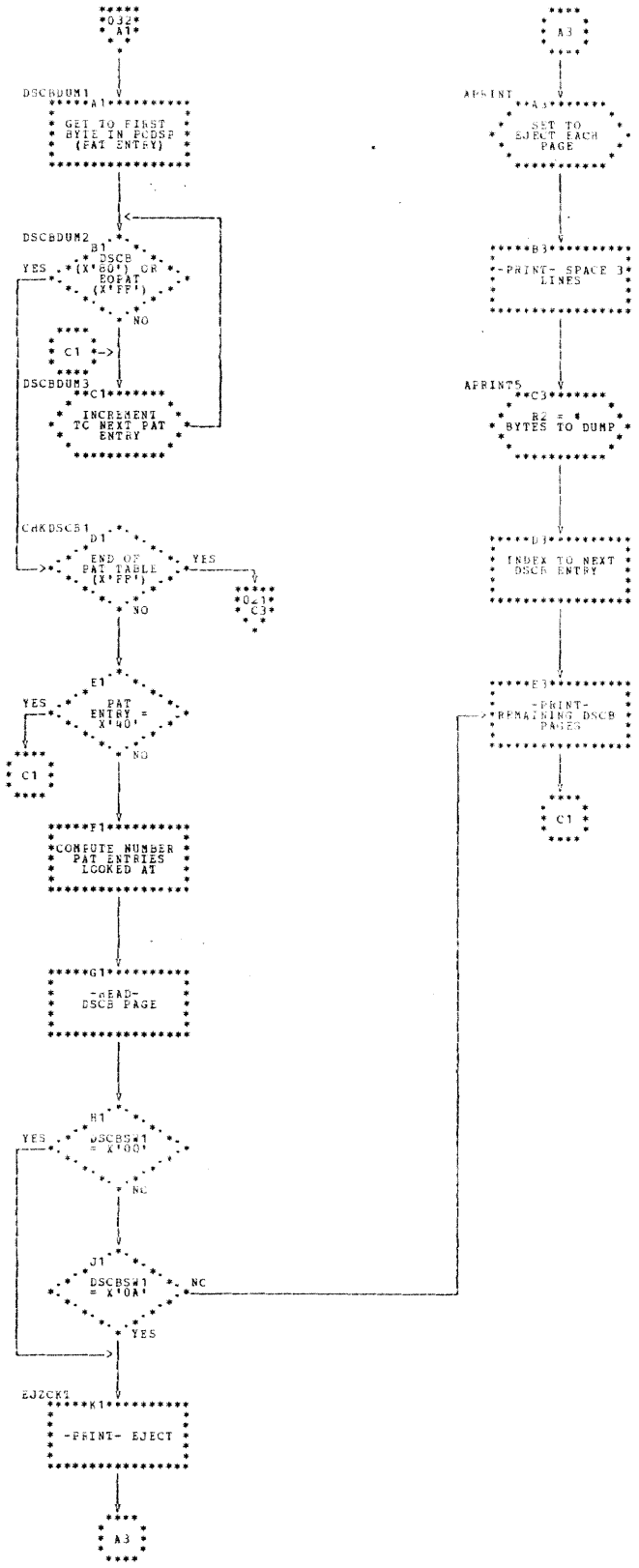


Chart EE. VAM2UT Routines (Part 15 of 15)



The six I/O interruption types are:

Type	Bits	Functions
Error	38-47	Control is given to IOERR, if any of these bits are on.
Attention	32	Control is given to ATTN if this bit is on. This request is honored.
Busy	35	SIO is reissued, if this bit is on.
Device End	37	Control is given IORTRN.
Control Unit End	34	SIO is reissued if this bit is on.
Channel End	36	TIO is reissued until a device-end condition results, if this bit is on.

IOERR: The CSW, PSW and CAW are saved by storing them in the UCB whose address is in REG2. The device address is taken from the UCB and put into REG4. A sense CCW address is loaded into REG5, and stored in the CAW in low core. If the error is a unit check only, or if a device end condition is present with any error, control is given to SENCHK; otherwise control is given to TYPECHK.

IORTRN: Here, control is returned via REG9.

SENCHK: The device address is entered into an SIO instruction from REG4. The SIO instruction, which executes the sense command addressed by the CAW that was stored in the IOERR routine, is issued until it is accepted. The sense and status bytes from the sense CCW are then stored in the UCB whose address is in REG2, before control is given to TYPECHK. All errors result in a message to the console and the program enters a wait state.

TYPECHK: Control is received either from IOERR or SENCHK. The device type causing the interruption is determined by checking the unit reference byte in the UCB whose address is in REG2. Depending upon the device type interruption, control is given to any one of the following locations:

Device	Location
2311, 2314	ERR100
2400	ERR300
1403	ERR400
1052	ERR500
2340	ERR600
2301	ERR700

I/O ERROR ROUTINES

Depending on the device type, control is given to certain locations as illustrated under "TYPECHK". At each location, there is the instruction BAL ERRLINK, ERRTEST followed by a table of 2-byte entries. This instruction loads the address of the next instruction (the table) into ERRLINK, and gives control to the common I/O error checking routine whose entry point is ERRTEST. ERRTEST uses the 2-byte table to interrogate sense and/or status bits. The first byte of the 2-byte table is examined to determine whether or not a sense byte or a status byte is to be tested. Once this is determined, the correct sense information bytes are obtained from the UCB whose address is in REG2.

One bit at a time is checked. If the bit tested is not set, an increment to the address of a 2-byte table is made to locate the next table entry. This provides the next bit to be checked. If the bit is set, the type of error is determined, and control is given to location A + B where:

A = the address of the first byte of the current table entry.

B = the value of the second byte of the current table entry.

That is, the second byte is a displacement of the address of the first byte. This enables the routine to get to a location and give control to it. This location within the ERRxxx routine that called the common I/O error checking routine, processes the error or gives control to any number of locations (depending on the device and type of error) where these errors may be processed. These locations are found in one of the following routines.

Error Processing Routines

- Routines common to DASD devices
- Routines common to tape devices
- All I/O error retries
- Operator alert routines common to all I/O devices
- Indexing factor to error message table
- IOPRNT error message table

UNIT CONTROL BLOCKS (UCB)

Each utility program contains one eight-word UCB for each device in use. UCBs serve two primary functions: they contain information needed for the STARTIO routine, and they also have information needed for error checking routines. Unit control blocks are also a source of user information since they are printed out along with an error message for certain types of errors. The unit control block format is as follows:

UNIT CONTROL BLOCK FORMAT

Byte	Function
00	Unit reference number
01	Unused
02-03	Device address
04	CAW protect
05-07	CAW
08-15	Interruption PSW
16-23	Interruption CSW
24-31	Sense bytes

Device Type Table

Code	Device
X'07'	1052
X'0A'	1403
X'01'	2311
X'02'	2314
X'03'	2400
X'05'	2301
X'06'	2400
X'08'	2540
X'0B'	2540

Error Indicator Code Format

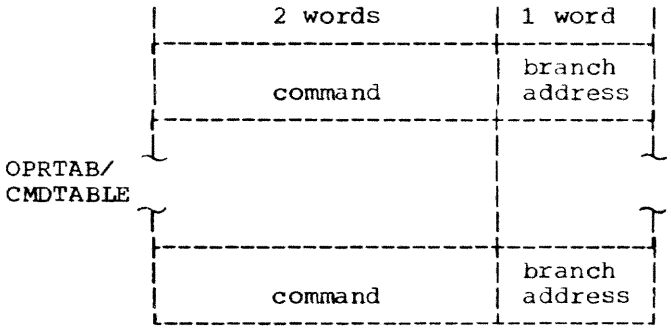
In the event a message output device has not been specified by the MSG statement, the following error indicators are displayed in the operator console storage address lights. The error indicator code following structure:

Indicator Code	Explanation
3000	General control statement error 1. Improper sequence 2. Improper format
4xxx	DASD unavailable
5xxx	Spare
6xxx	Spare
7xxx	DASD error or unit not ready
8xxx	Input parameter insertion incorrect
9xxx	Input device unavailable
Axxx	Input device error or not ready
Bxxx	Output device unavailable
Cxxx	Output device error or not ready
Dxxx	Improper unit address specified
E000	Improper system output file statement
F000	Normal program completion

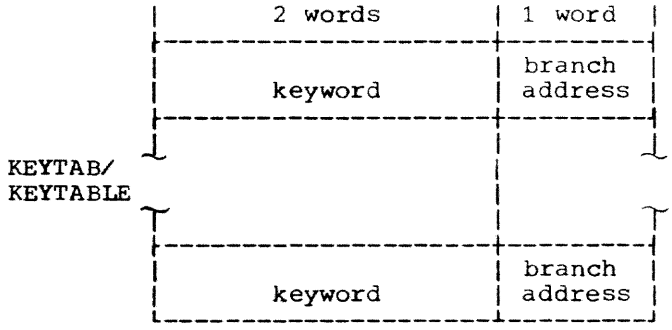
Note: xxx indicator is channel- and unit-interface address of error device.

The CPU is placed in the wait state.

OPRTAB and CMDTABLE consist of a three-word entries aligned on word boundaries. The first eight bytes are a left-adjusted character constant command. The third word is an unconditional branch to the command-processing routine.



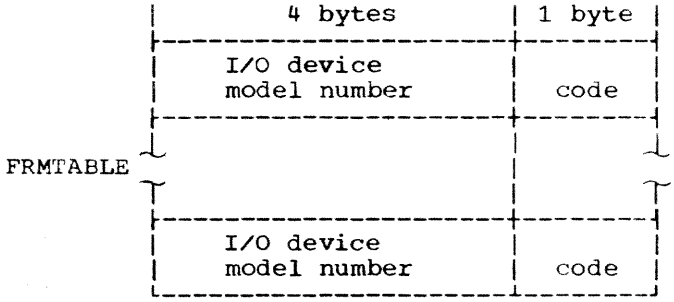
KEYTAB and KEYTABLE consist of three-word entries aligned on word boundaries. The first eight bytes are a left-adjusted character constant keyword. The third word is an unconditional branch to the keyword-processing routine.

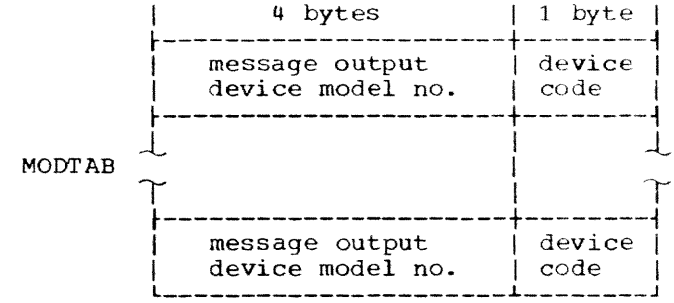


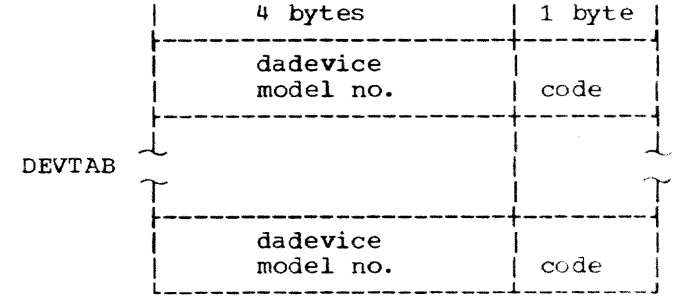
FRMTABLE, MODTAB, and DEVTAB are five-byte entries. The first four bytes are a character constant I/O device model number:

FRMTABLE - I/O device
MODTAB - message output device
DEVTAB - direct access device

The fifth byte is a device type code used by the utility I/O routines.







The index for this manual is organized individually by module ID. Refer to the specific module for index entries to module routines, register usage, tables, etc. Use the following list to cross reference the module name to the module ID.

System/360, Model 67 Core Dump - Core Dump

Direct-Access Storage Device Dump/Restore - DASDDR

Direct-Access Storage Device Initialization - DASDI

Direct-Access Storage Device Print - DADUMP

Core Dump (CEBDP) 46
  alternate PSA 46
  assumptions 46
  attributes 46
  bootstrap loader 46
  BUILDRTN routine 46

  CONTINUE routine 47
  COREDUMP routine 47
  CPU information saved 46
  CPU dump routine 47

  duplicate lines 48

  END card 46

  FIRST routine 46

  initialization 46
  IPL record 46

  NOTAVAIL routine 47

  ONLYONE routine 47
  operation 46
  OPT card 47

  PAGEPRNT routine 48
  PFXSW routine 47
  prefix records 47
  primary PSA 46
  PSA table 46,47
  PSW instruction address 46

  READ routine 46
  restrictions 46

  TESTDUP routine 48
  TXT card 46

DADUMP (CEBDP) 33
  attributes 33
  ATTN routine 34

BBCCHH 33
BEGIN statement 33

CKCSW routine 38
CLEARCOR 36
CLRSCAN routine 35
CMDTABLE 35
CONSOLE routine 33
CSR4 register 38

DEVLUP subroutine 34

end of volume 37
END statement 35
error checks 36

FORMAT routine 37
FORMATSW 37
FRMTABLE 34,38

hardware requirements 33
HEADER routine 37

initialization routines 33
interruption PSW 33
I/O supervisory routines 38

Job Fulfillment routines 38
Job Statement Analysis routines 35

KEYTABLE 35

LISTTEST routine 37

message routines 38
missing address marker 37
MOVEBUF subroutine 38
MSGPRINT routine 35

operation 33
overflow error 36
OVRFLOSW 36

PRINT routine 38
PRINT statement 35
PRNTDASD routine 37
processing routines 36
program check PSW 36

READDATA subroutine 37
RECOVER entry point 36
RECTABLE 33
restrictions 33
routines 33,34

SAVREC 33
SCANCARD routine 35
SCANSW 35
START routine 33
STARTIO routine 38

TESTINPT 37
track condition 0 37
TRACKNO 33
TYPEINPT routine 34

UCBIN 33
UCBMSG 33,35

VCOUNTY subroutine 37
VRECOVR routine 36

2301 error 36

DASDDR (CEBDR) 15
AGAIN14 24
ANALSENS routine 22
ATTN routine 17
attributes 15

BADPAGE routine 23
bad track SAM dump 23
bad track SAM restore 23
bad track switch 23
BDPG 2314 routine 24
BEG1 routine 20
BGNADR 20
BITMAP 24
BITMAP record 21
bit map 21
BLDMAP routine 21
BUILDCCW routine 21
bypass switch (RDCARD) 17

CHANGCYL routine 23
CHTOSKP routine 23
CKALT routine 23
CKALT1 routine (DASDDR) 23
CKCSW 27
CKINPUT routine 17
CKVOLLBL routine 21
CLRERPG 25
CLRSCAN 17
CNVRELPG 25
COMEBACK 24
control field 22
control record 16,22
control statement analysis routines 17
control statement error 17
CNGRDCKD routine 22
CNTRLBUF (control data record) 20,21

DADSM routine 25
DASD CCW 22
data field 22
define SYSIN 17
DEVLUP 17
DEVTAB 17
DMPDASD routine 22
DMPDTA routine 22
DSCB extent analysis 25
DTABUFCK 22

END statement 17
ENDADR 20
ENDOFJOB routine 24
ENDOFPAT 26
end-of-disk pack 24
end-of-job 24
end-of-reel 24
ENDREEL routine 24
EOJ (end-of-job) 22
EOJA routine 24
EOJAA routine 24
EOJ1 routine 24

FRMTAPE routine 21
FULLDPSW switch 21

GETCCW routine 23

hardware requirements 15
HAVEADDR routine 20

initialization routines 16
INIT 2314 subroutine 21
IOPRNT (I/O error print) routine 27
I/O interruption classification 27
I/O supervisory routines 26

limits record (2311 or 2314 to tape) 15

message routines 27
MODTKADF routine 22
MODTKADT routine 22
MSGCCW 27
multiple tape unit exception 24

operation 15

PATWRITE 26
PGAVAIL 24
purpose 15
PUTMAP routine 22
PREPRWRT routine 22
PRNT (message output) routine 27
processing routines 20
program flow 16
PRTMSG 24

RBLDDSCB 25
RDBDPG 23
RDCARD routine 17
RDLABELS subroutine 20
read protect switch 24
READDSCB subroutine 21
restrictions 15
RLOCAT 24
routines 18,19,20
RSTRTAPE routine 22
RSTRTAP2 routine 22

START routine 16
START1 routine 17
STARTIO routine 26
SWITCHRD work area 17
SYOCCW 27
SYSIN UCB (UCBSYI) 17
SYSIN routine 26
SYSOUT entry point 27

TAPEEOF 25
tape switch 20
TOFM2311 routine 21
TOTAPE routine 20

TRYAGAIN 24
TSTWRTSP routine 22

UCBMSG 27
UCBOPR 17
UCBSYI 17
UCBSYO 27
V2UNEXCP 26
V2WRTCHK 26
VAM 2 16
VAM2DUMP 25
VAM2REST 26

volume label 21
VTOCCK 24
VTOCOK 24

WAITZ routine 21
write-to-operator (WTO) routine 27
WRTCHECK routine 23
WRCHEKSW (write-check-switch) 23
WRTDISK routine 22
WRTDISKX 23
WRITDISKY 23
WRTDISK1 entry point 23
WTO (write-to-operator) routine 27

DASDI (CEBDI) 2
ADTOLIST 7
absolute loader 2
alternate track (GETALT) 3
attributes 2
ATTN routine 5

BITMAP (VAM 2301) 10

Clearing the PAT 10
control statement analysis 5
CSR4 5
CKCSW routine 5
CKINPUT routine 5
CKVOLLBL-DASDI 7
CLRSCAN 7
CONSTR1 8
CONSTR2 9

DADEF control card (GEN00) 7
DASDIPAT 10

END statement 11
ENTPOIT1 routine 9
EOJ 10
error processing 7
exit 3
EXTNT 9

FMTVTOC 9
formatting routines 7

GEN00 7
GETALTYZ 11
GETALT4 11

hardware requirements 2
HEX'00' 8
HEX'55' 8

initialization 3
initiation 2
INTALT 7
I/O interruptions 5
IOPRNT 11
I/O Supervisory routines 3
IPL 2

KEYWORD parameters 6

LASTCARD (EOJ) 10
LENGTH 5
LISTAREA 7
LOADA 8

Message routines 11

OPPRNT 11
OPPRN1 entry point 11
OPRCCW 11

PAT 2
PAT CLEAR 10
PRNT 11
purpose 2

RDCARD 7
restrictions 2
routines 4

SCANADR 7
STARTUP prelude 2
START routine 3
STARTIO routine 5
STARTIOX entry point 5
SYSIN routine 5
SYSOUT tape (EOJ) 10
SYSOUT message device 11

TAB00 7
TAB55 7
tapeEOF (EOJ) 10
Termination routines 10
TRKPRNT 11

VTOC 2
VAM 2 2,3
VAM 2301 10
Volume Label Checking 7

WRTVTOC 9
WTO (write-to-operator) 11

VAM2UT (CEBVP) 52
$ (dollar sign) 54
$PAT flag 57

absolute loader 52
alias 58
ASKAGAIN routine 56
ATTN routine 54
attributes 52
attributes change 57
offset, length, type 57
available buffer storage 52,54

binary conversion 56
BITOHEX routine 58
buffer
  available storage 52
  SYSIN UCB 58
BUILDSCB routine 56

CALL command 52
CEBVP (see VAM2UT) 52
change attributes 57
change length 57
CHGLEN 57
CHKDIS 56
CHKSUM 56
CKAGAIN routine 56
CKCSW routine 52
Command Processing routin s 54
  FOUNDSCB 54
  CKBUF2 54
  COMRTN 54
commands 52
  CALL 52
  RUN 52
  SET 52
  DUMP 53
  DISPLAY 53
  PATCH 53
COMRTN1 entry point 54
configuration 52
continue reply 56
Control Section Dictionary (CSD) 56
convert to binary 56
CSD address 57
CSECT address 57

data sets span more than
  one volume 57
define SYSIN 52
define SYSOUT 58
defined disk 56
device configuration 52
DISKONN routine 56
DISPLAY command 53
display patch 57
dollar sign ($) 54
DONELOOP 57
DOSAREA 56
DOVL routine 56
DOVLP routine 56
DSCB pointer entries 56
DSCB slot number 56
DSCBDUM1 routine 56
DSCBDUMP routine 56
DSCBREL table 56
dsname retrieved 57
dummy UCB 52
DUMP command 53
dump DSCBs 56
  E DSCBs 56
  DSCBs for data set 56
dump PAT 57
dump PM 57 D
dump prelude 58

enabled interruptions 22
exit 52
external page table 57

FINDCSD routine 56
FIXCHK routine 56
format E DSCB dump 56
FOUNDSCB routine 54

GETDEV routine 57
GETDEV2 57
GETDEV3 57
GETDSNAM routine 57
GETDSO 57
GOWRITE1 routine 57

hardware requirements 52
hash member name/alias 58

I/O attention handler 52
I/O interruptions received 52
I/O status check (CKCSW) 52
initial attention interruption 52
initialization routines 53
  START 54
  ATTN 54
  initiation 52
  interruption
  initial 52
  to cancel 56
  enabled 52

keywords 54

length change 57
loader 52

MAKETAB 57
maximum pages for multi
  volume data set 52
maximum volumes 52
member length 52
member name 58
MODSAVE1 routine 57
MODUMP routine 57
MOVEREL routine 57

offset, length, type 57
operation 52
OPPRNT routine 57,56

PAGEOFF1 routine 57
pages, maximum 52
PAT
  dump 57
  read 56
  write page 57
patch a 2314 52
patch a disk 52
PATCH command 53
PATCH routine 57
PATCHER routine 57
PATDUMP routine 57
PATDUMP1 routine 57
PDMP3 57
PMD pages to dump 57
PMD read 58
PMDPATCH routine 57
POD page table 57
POD, read 57
PODSDUMP routine 57
prelude

read 58
write 58
processing routines 54
public pack mounting 57

RDPOD routine 57
read PAT 56
read patch 57
read PMD 58
read POD 57
read prelude 58
READPAGE routine 57
READPMD routine 58
READPREL routine 58
REFIX 57
relocated pages 57
relocation table 56,57
RELTAB 56
restore page table 57
restrictions 52
retrieve dsname 57
REVPAT 56
routines
  initialization 53
  command processing 54
  processing 54
RUN command 52

SCAN routine 58
second level commands 54
SET $DISK operation 57
SET command 52
SETDIN routine 58
SETDISK routine 58
SETDOUT routine 58
SRCHMPD routine 58
SRCHPOD routine 58
START routine 54
stop reply 56
syntax
  $VAM 58
  check
  TSSS 52
SYSIN
  define 52,54
  operator console 54
SYSIN UCB buffer 58
SYSOUT=printer 58

tape resident 52
transfer control to reader 54
TSSS syntax 52

VAMCK routine 58
volume label patch 56
volume label 56
  write to disk 56
volumes, maximum 52

WANTATRB routine 57
write PAT page 57
write prelude 58
write volume label 56
WRITPREL routine 58
WRTVLBL routine 56

TSS Utilities Printed in U.S.A. Y28-2039-3

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
(USA Only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)